Eurocomp

LGP 21

Unterprogramm-Handbuch

Eurocomp GmbH
Elektronische Rechenanlagen
Minden/Westf.

Eurocomp

PROGRAMM-BESCHREIBUNG LGP21

EUROCOMP GMBH · ELEKTRONISCHE RECHENANLAGEN · MINDEN (WESTF)

LGP-21 Unterprogramme

✓ vorhanden

Inhaltsverzeichnis

Datenausgabe 3	J3-10.2 ✓
Datenausgabe 4	J3-10.3 ✓
Datenausgabe 5	J3-10.4 ✓
Datenausgabe 6	J3-10.5 ✓
Hexadezimale Ausgabe 2	J4-10.1 ✓
Hexadezimale Ausgabe 3	J4-10.2 ✓
Hexadezimale Ausgabe 4	J4-10.3 ✓
Alphanumerische Ausgabe 1	J4-11.0 ✓
Alphanumerische Ausgabe 2	J4-11.1 ✓
Hexadezimale Eingabe 3	J5-10.0 ✓
Richtung oder Winkel Eingabe und Ausgabe 1	J9-11.0 ✓
T r a c e 1	K1-10.0 ✓
Speicherausdrucken 1	K2-10.0 ✓
Speicherausdrucken 2	K2-10.1 ✓
Suchprogramm 1	K3-10.0 ✓
Programmtestprogramm 1	K9-10.0 ✓
Sortieren 1	L1-10.0 ✓
Sortieren 2	L1-10.1 ✓
Sortieren 3	L1-10.2 ✓
Sortieren 4	L1-10.3 ✓
Binärisieren ganzer Zahlen 1	L3-10.0 ✓
Umspeichern 1	L4-10.0 ✓
Streifendoppler 1	L4-11.0 ✓
Tabellenabsuchen 1	L7-10.0 ✓
Schleifenzähler 1	L8-10.0 ✓

SINUS-COSINUS 1	B1-10.0

VERWENDUNGSZWECK

B1-10.0 berechnet den Sinus oder Cosinus eines im Gradmaß gegebenen Winkels. Ist der gegebene Winkel größer als 360°, so reduziert das Unterprogramm ihn auf das entsprechende Äquivalent zwischen 0° und 360°, bevor es den Funktionswert errechnet. Zur Approximation wird ein Polynom 9. Grades benutzt.
B1-10.0 arbeitet als Unterprogramm und wird durch eine Aufruffolge des Hauptprogrammes angesprochen. Beim Sprung in das Unterprogramm muß der Winkel im Akkumulator stehen, und die Rückkehradresse muß im Unterprogramm gespeichert sein. Bei der Rückkehr in das Hauptprogramm steht der Funktionswert einschließlich Vorzeichen im Akkumulater.

SPEICHERBEDARF	1 Spur
Benutzte Zwischenspeicher	Die Sektoren 2, 4, 5, 6, 7, 45 der Spur 63

EINGABE

Die Eingabevariable für B1-10.0 ist ein Winkel Θ, dessen Funktionswert berechnet werden soll. Θ muß im Akkumulator@9 stehen, wenn der Sprung in das Unterprogramm erfolgt. Der Winkel muß im Gradmaß vorliegen. Ist der Winkel größer als 360°, so wird er vom Unterprogramm auf sein Äquivalent zwischen 0° und 360° reduziert.

AUFRUFFOLGE

Für Sinus und Cosinus sind verschiedene Aufruffolgen vorgesehen. Bei einem Durchlauf des Unterprogrammes kann nur ein Funktionswert errechnet werden.

1. SINUS

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Θ@9
α	R	L_o + 49	Rückkehradresse in das Unterprogramm setzen.
+1	U	L_o	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr zum Hauptprogramm.

Der Befehl in α-1 bringt Θ@9 in den Akkumulator. (Jeder Befehl der dies tut, ist zulässig.) Der Befehl in α speichert die Rückkehradresse (α+2 im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

2. COSINUS

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe ϑ@9
∝	R	L_o + 49	Rückkehradresse in das Unterprogramm setzen.
+1	U	L_o + 4	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr zum Hauptprogramm.

Die Aufruffolge für den Cosinus ist identisch mit der für den Sinus, mit Ausnahme des Befehls in ∝+1. Dieser Befehl bewirkt einen Sprung nach L_o + 4, der Anfangsadresse des Cosinus, wobei L_o Anfangsspeicheradresse des gesamten Unterprogrammes ist.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht der gewünschte Funktionswert einschließlich des Vorzeichens im Akkumulator @1.

BEDIENUNG

1. Genauigkeit: Kein Fehler ist größer 5×10^{-7}.
2. Zeitbedarf: ca. 775 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine
5. Überlauf:
   Wird beim Eingang in das Unterprogramm nicht beachtet und beim Ausgang nicht zurückgesetzt.

SINUS-COSINUS 2	B1-10.1

VERWENDUNGSZWECK

Das Programm B1-10.1 berechnet den Sinus oder Cosinus eines im Bogenmaß gegebenen Winkels. Der absolute Berrag des Winkels Θ darf im Bogenmaß nicht größer als 7,9999999 sein. Vor der Berechnung der Funktion reduziert das Unterprogramm den Winkel Θ auf einen Winkel β, dessen absoluter Betrag im Bogenmaß $\leq 1{,}75$ ist. Zur Berechnung der Funktion Θ wird die Taylor-Reihe bis zum 14. Grad benutzt. Das Programm B1-10.1 arbeitet wie ein Unterprogramm und wird mittels einer Aufruffolge des Hauptprogrammes angesprochen. Beim Sprung in das Unterprogramm muß das Argument im Akkumulator stehen. Die Aufruffolge muß die Rückkehradresse einsetzen. Bei der Rückkehr in das Hauptprogramm steht der Funktionswert im Akkumulator.

SPEICHERBEDARF — 1 Spur

Benutzte Zwischenspeicher — Die Sektoren 8 und 49 der Spur 63

EINGABE

Die Eingabevariable für "Sinus-Cosinus 2" ist ein Winkel Θ, dessen Funktionswert zu berechnen ist. Θ muß im Akkumulator@3 stehen mit der Bedingung $|\Theta| \leq 7{,}9999999$. Das Unterprogramm reduziert Θ auf einen äquivalenten Winkel β mit $|\beta| \leq 1{,}75$ @ 1.

AUFRUFFOLGE

Für Sinus und Cosinus sind verschiedene Aufruffolgen vorgesehen. Bei einem Durchlauf des Unterprogrammes kann nur der Wert einer Funktion errechnet werden.

1. SINUS

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Θ @ 3
α	R	L_o + 16	Speichere die Rückkehradresse im Unterprogramm.
+1	U	L_o	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr aus dem Unterprogramm.

Der Befehl in α-1 bringt Θ @ 3 in den Akkumulator. (Jeder Befehl, der dies bewirkt, ist zulässig.) Der Befehl in α speichert die Rückkehradresse (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

2. COSINUS

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bring θ@3
α	R	L_o + 16	Speichere die Rückkehradresse im Unterprogramm.
+1	U	L_o + 45	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Die Aufruffolge für den Cosinus ist identisch mit der für den Sinus, mit Ausnahme des Befehls in α+1. Dieser Befehl bewirkt einen Sprung nach L_o +45 des Unterprogrammes, wobei L_o die Anfangsadresse des Unterprogrammes ist.

AUSGABE

Bei der Rückkehr steht der entsprechende Funktionswert mit dem algebraischen Vorzeichen im Akkumulator @1.

BEDIENUNG

1. Genauigkeit: Der Fehler im Bogenmaß ist kleiner als 8×10^{-9}.
2. Zeitbedarf: ca. 1200 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.
5. Überlauf:
   Wird nicht beachtet und auch nicht zurückgesetzt.

ARCUSTANGENS 1	B1-11.0

VERWENDUNGSZWECK

B1-11.0 berechnet den Arcustangens einer beliebigen Zahl, die kleiner als 512 ist. Der Hauptwert, mit algebraischem Vorzeichen, ist im Gradmaß ausgedrückt. Zur Approximation wird ein Polynom 15. Grades benutzt.
B1-11.0 arbeitet wie ein Unterprogramm, und wird durch eine Aufruffolge des Hauptprogrammes angesprochen. Beim Sprung in das Unterprogramm muß das Argument im Akkumulator stehen, und die Rückkehradresse muß im Unterprogramm gespeichert sein. Bei der Rückkehr in das Hauptprogramm steht der Arcustangens im Akkumulator.

SPEICHERBEDARF	1 Spur
Benutzte Zwischenspeicher	Die Sektoren 4, 5, 6, 7, 8, 9, 10, 13, 50 und 51 der Spur 63

EINGABE

Die Eingabevariable für B1-11.0 ist die Zahl N, deren Arcustangens berechnet werden soll. Beim Sprung in das Unterprogramm muß N @ 9 im Akkumulator stehen.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe N @ 9
α	R	L_o + 51	Speichere Rückkehradresse im Unterprogramm.
+1	U	L_o	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α-1 bringt N @ 9 in den Akkumulator. (Jeder Befehl, der dies bewirkt, ist zulässig.) Der Befehl in α speichert die Rückkehradresse (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht der Hauptwert des Arcustangens im Gradmaß mit Vorzeichen im Akkumulator @ 9. Der absolute Wert variiert zwischen 0^o und $89{,}90^o$.

BEDIENUNG

1. Genauigkeit: Kein Fehler ist größer als 5×10^{-7} Grad.
2. Zeitbedarf: 950 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.

5. Überlauf:
   Wird weder beim Eingang beachtet, noch beim Ausgang zurückgesetzt.

BEMERKUNGEN

Nach Wunsch kann eine Programmänderung vorgenommen werden, die bewirkt, daß 90^{o} (absolut genommen) besser angenähert werden. Man ändere den Inhalt der Speicherzellen $L_o + 56$ und $L_o + 59$.

Speicherplatz	Inhalt (alt)	Inhalt (neu)
$L_o + 56$	1@9	1@q>9
$L_o + 59$	2@9	2@q>9

Das Argument muß dann beim Sprung in das Unterprogramm bei diesem "q" stehen, das größer als 9 ist.

VEKTORIELLE BESTIMMUNG DES ARCUSTANGENS 1

B1-11.1

VERWENDUNGSZWECK

Das Programm B1-11.1 berechnet θ so, daß bei gegebenem x und y gilt:

$$y = (\sqrt{x^2 + y^2}) \sin \theta$$

$$x = (\sqrt{x^2 + y^2}) \cos \theta$$

Man erhält in allen vier Quadranten Lösungen. Die Werte x und y müssen beim gleichen "q" gegeben sein. Das Ergebnis stellt ein Teilergebnis dar, das in Grad oder Bogengrad umwandelbar ist.
Das Programm arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Beim Sprung in das Unterprogramm muß y im Unterprogramm gespeichert worden sein, die Rückkehradresse muß eingesetzt worden sein, und x muß im Akkumulator stehen. Bei der Rückkehr in das Hauptprogramm steht das Ergebnis im Akkumulator.

SPEICHERBEDARF — 2 Spuren, 17 Sektoren

Benutzte Zwischenspeicher — keine

EINGABE

Die Eingabevariablen für das Programm sind die beim gleichen "q" stehenden Werte x und y. Der Wert y muß sich im Unterprogramm befinden, und x muß im Akkumulator stehen, wenn ein Sprung in das Unterprogramm erfolgt.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-3	B	[ ]	Bringe y in den Akkumulator.
-2	C	L_o + 34	Speichere y im Unterprogramm.
-1	B	[ ]	Bringe x in den Akkumulator.
α	R	L_o + 10	Speichere Rückkehradresse.
+1	U	L_o	Sprung in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α-3 bringt y in den Akkumulator, und der Befehl in α-2 speichert y im Unterprogramm. (Alle Befehle, die y in L_o + 34 des Unterprogrammes speichern, können benutzt werden.) Der Befehl in α-1 bringt x in den Akkumulator, das beim gleichen "q" wie y stehen muß. (Jeder Befehl, der dies bewirkt, kann benutzt werden.) Der Befehl in α speichert die Rückkehradresse im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. Speicherzelle α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht das Ergebnis bei "q" = 1 im Akkumulator. Zur Umwandlung ins Gradmaß multipliziere man diesen Wert mit 180; zur Umwandlung ins Bogenmaß multipliziere man den Wert mit π.

BEDIENUNG

1. Genauigkeit:
   Kein Fehler überschreitet $\pm$ 4 x 10^{-8} im Teilergebnis.
2. Zeitbedarf: ca. 1160 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.
4. Es sind keine Eingabe-/Ausgabeeinheiten erforderlich.
5. Überlauf:
   Beim Eingang wird Überlauf nicht beachtet, noch wird er beim Ausgang zurückgesetzt.

ARCUSSINUS-ARCUSCOSINUS 1	B1-12.0

VERWENDUNGSZWECK

Das Programm B1-12.0 berechnet entweder den Arcussinus oder den Arcuscosinus einer gegebenen Zahl. Für die Zahl gilt $-1 \leq N \leq 1$. Der Hauptwert mit algebraischem Vorzeichen wird im Gradmaß ausgedrückt. Zur Approximation wird ein Polynom 7. Grades benutzt.
B1-12.0 arbeitet wie ein Unterprogramm und wird durch eine Aufruffolge des Hauptprogrammes angesprochen. Beim Sprung in das Unterprogramm muß das Argument im Akkumulator stehen, und die Rückkehradresse muß im Unterprogramm gespeichert worden sein. Bei der Rückkehr in das Hauptprogramm steht der gewünschte Funktionswert im Akkumulator.

SPEICHERBEDARF	2 Spuren, 32 Sektoren
Benutzte Zwischenspeicher	Die Sektoren 12, 16, 18, 19, 20, 21, 23, 24 und 28 von Spur 63

EINGABE

Die Eingabevariable für B1-12.0 ist die Zahl N, deren Funktionswert errechnet wird. Für N gilt $-1 \leq N \leq 1$. Beim Sprung in das Unterprogramm muß N @ 1 im Akkumulator stehen.

AUFRUFFOLGE

Für Arcussinus und Arcuscosinus sind verschiedene Aufruffolgen vorgesehen. Bei einem Durchlauf des Unterprogrammes kann nur der Wert einer Kofunktion errechnet werden.

1. ARCUSSINUS

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe N @ 1.
α	R	L_o + 21	Speichere die Rückkehradresse im Unterprogramm.
+1	U	L_o	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α-1 bringt N @ 1 in den Akkumulator. (Jeder Befehl, der dasselbe bewirkt, ist zulässig.) Der Befehl in α speichert die Rückkehradresse (α+2) in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. In α+2 steht der Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

2. ARCUSCOSINUS

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe N @ 1.
α	R	L_o + 21	Speichere die Rückkehradresse im Unterprogramm.
+1	U	L_o +211	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Die Aufruffolge für den Arcuscosinus ist identisch mit der für den Arcussinus, mit Ausnahme des Befehls in $\alpha+1$. Dieser Befehl bewirkt einen Sprung zur Anfangsadresse L_o + 211 des Arcuscosinusprogrammes. Dabei ist L_o die Anfangsadresse des gesamten Unterprogrammes.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht der Hauptwert der gewünschten Kofunktion im Akkumulator. Er steht @ 9 im Bogenmaß mit Vorzeichen. Der Absolutwert des Arcussinus variert zwischen -90^o und $+90^o$, der Absolutwert des Arcuscosinus zwischen 0^o und $+180^o$.

BEDIENUNG

1. Genauigkeit:
   Kein Fehler ist größer als $1{,}2 \times 10^{-6}$ im Gradmaß.
2. Zeitbedarf: ca. 2550 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0
      und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.
5. Überlauf:
   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.
6. Fehlerstop:

Speicherplatz	Bedeutung und Behebung
L_o + 161	Unterprogramm soll die Wurzel aus x berechnen, wobei x negativ ist.

BEMERKUNG

Da zur Berechnung des Arcussinus bzw. des Arcuscosinus die Berechnung einer Quadratwurzel erforderlich ist, so wurde das Programm dafür in das Unterprogramm aufgenommen. Es kann unabhängig durch folgende Aufruffolge benutzt werden:

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe eine Zahl bei geradem "q"
α	R	L_o + 150	Speichere Rückkehradresse.
+1	U	L_o + 100	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in $\alpha-1$ bringt das Argument bei geradem "q" in den Akkumulator. (Jeder Befehl, der dasselbe bewirkt, kann benutzt werden.) Der Befehl in α speichert die Rückkehradresse ($\alpha+2$) in das Unterprogramm. Der Befehl in $\alpha+1$ bewirkt einen Sprung nach L_o + 100, der Anfangsadresse der Quadratwurzel, wobei L_o die Anfangsadresse von B1-12.0 ist. In $\alpha+2$ steht der Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird. Die Quadratwurzel steht im Akkumulator. Steht das Argument N @ q_N dann steht $\sqrt{N}$ @ $\frac{q_N}{2}$.

EXPONENTIALFUNKTION 1	B3-10.0

VERWENDUNGSZWECK

Das Programm B3-10.0 berechnet die Exponentialfunktion K^x, wobei $K = 2$, e oder 10 ist und für x gilt: $-1 \leq x \leq 1$.
Exponentialfunktion 1 arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Beim Sprung in das Unterprogramm muß x im Akkumulator stehen, und die Rückkehradresse muß im Unterprogramm gespeichert worden sein. Bei der Rückkehr in das Hauptprogramm steht der Wert von K^x im Akkumulator.

SPEICHERBEDARF 63 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariable für das Programm ist der Exponent x. x muß der Umgleichung $-1 \leq x \leq 1$ genügen und muß sich im Akkumulator bei "q"=1 befinden, wenn der Sprung in das Unterprogramm erfolgt.

AUFRUFFOLGE

Für jeden Exponentialwert (zur Basis 2, e oder 10) existiert eine spezielle Aufruffolge. Diese sind im folgenden gelistet und erklärt.

1. 2^x

Speicherzelle	Befehl	Adresse	
-1	B	[ ]	Bringe x @ 1.
α	R	$L_o + 9$	Speichere Rückkehradresse in das Unterprogramm.
+1	U	L_o	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

2. e^x

Speicherzelle	Befehl	Adresse	
-1	B	[ ]	Bringe x @ 1.
α	R	$L_o + 9$	Speichere Rückkehradresse in das Unterprogramm.
+1	U	$L_o + 2$	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

3. 10^x

Speicherzelle	Befehl	Adresse	
-1	B	[ ]	Bringe x @ 1.
α	R	$L_o + 9$	Speichere Rückkehradresse in das Unterprogramm.
+1	U	$L_o + 3$	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α-1 bringt x @ 1 in den Akkumulator. (Jeder Befehl, der dies bewirkt, ist zulässig.) Der Befehl in α speichert die Rückkehradresse (α+2) in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung zur jeweils angegebenen Stelle des Unterprogrammes, wo die entsprechenden Funktionswerte errechnet werden: Anfangsadresse L_o für 2^x; L_o + 2 für e^x; L_o + 3 für 10^x.

α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht der gewünschte Funktionswert bei "q" = 4 im Akkumulator.

BEDIENUNG

1. Genauigkeit:
   Kein Fehler ist größer als 5×10^{-8}.
2. Zeitbedarf: ca. 775 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.
5. Überlauf:
   Wird vom Programm weder beim Eingang beachtet noch zurückgesetzt.

BEMERKUNG

Zur Bestimmung der Exponentialfunktion eines Argumentes "x" dessen absoluter Wert größer als 1 ist, benutze man die Relation $K^{x.xx}=(K^x)\cdot(K^{\cdot xx})$; wobei $K^x=K\cdot K\cdot K...K$, x mal K, und der Wert $K^{\cdot xx}$ vom Unterprogramm berechnet wird. Z.B. $e^{3,476}= (e\cdot e\cdot e)\cdot (e^{\cdot 476})$. Das duale "q" von $K^{x.xx}$ kann bestimmt werden durch:

$$4 + \sum_{i=1}^{x} (q \text{ von } K)_i$$

oder

$$4 + (q \text{ von } K)\cdot x$$

wobei K bei jedem beliebigen dem Programmierer passenden "q" stehen kann.

LOG 1	B3-11.0

VERWENDUNGSZWECK

Das Programm B3-11.0 berechnet den Logarithmus von einer positiven Zahl zur Basis 2, e oder 10. Als Näherungsverfahren wird ein Polynom 7. Grades benutzt.
B3-11.0 arbeitet wie ein Unterprogramm und wird durch eine Aufruffolge im Hauptprogramm angesprochen. Beim Sprung ins Unterprogramm muß das Argument im Akkumulator stehen. Die zwei Speicherplätze, die nach dem Sprungbefehl folgen, müssen das "q" des Arguments und die gewünschte Basis des Logarithmus enthalten. Bei der Rückkehr ins Hauptprogramm steht der Logarithmus im Akkumulator.

SPEICHERBEDARF 1 Spur, 58 Zellen

Benutzte Zwischenspeicher keine

EINGABE

Die erforderliche Eingabe von "Log 1" besteht aus: 1. N, der Zahl, deren Logarithmus berechnet werden soll, 2. Q, ist das "q" von N und 3. K, die gewünschte Basis des Logarithmus.
N muß eine positive Zahl, größer als Null sein, und muß beim Sprung ins Unterprogramm mit einem positiven "q" im Akkumulator stehen.
Das Q muß zwischen $0 \leq Q \leq 31$ liegen. In der Aufruffolge steht das Q in dem Sektorteil eines Z-Befehles, der unmittelbar nach dem Sprungbefehl folgt.
K, die gewünschte Basis des Logarithmus, steht im Sektorteil eines Z-Befehls.
Für log N zur Basis 2 ist K = 00;
für log N zur Basis e ist K = 01;
für log N zur Basis 10 kann K jeden beliebigen Wert außer 00 und 01 annehmen. Der Z-Befehl, der das K enthält, ist der letzte in der Aufruffolge.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe N bei einem positiven "q".
α	R	$L_o + 24$	Halte die Adresse von Q im Unterprogramm.
+1	U	L	Sprung ins Unterprogramm.
+2	Z	08QQ	"q" von N.
+3	Z	000K	Angabe der gewünschten Basis des Logarithmus.
+4	u.s.w.		Rückkehr ins Hauptprogramm.

Die Aufruffolge zeigt, daß nach dem Befehl in Speicherzelle α-1 N mit einem positiven "q" im Akkumulator steht. (Jeder Befehl, nach dessen Ausführung N im Akkumulator steht, ist zulässig.) Der Befehl in Speicherzelle α speichert die Adresse (α+2) des Z-Befehles, der Q enthält, im Unterprogramm.

Der Befehl in der Zelle α+1 bewirkt einen Sprung nach L_8, der Anfangsadresse des Unterprogrammes. Der Befehl in der Speicherzelle α+2 enthält Q bei einem "q" von 29. Der Befehl in der Zelle α+3 enthält K @ 29. Die Speicherzelle α+4 enthält den nächsten Befehl des Hauptprogrammes, der nach dem Rückkehrsprung ins Hauptprogramm ausgeführt wird.

AUSGABE

Nach dem Rückkehrsprung ins Hauptprogramm steht der Logarithmus bei der gewünschten Basis, bei einem "q" von 6 im Akkumulator.

BEDIENUNG

1. Genauigkeit:
   Ein eventueller Fehler ist kleiner als 3×10^{-8}.
2. Zeitbedarf: ca. 1350 ms.
3. Programmeingabe:
   Der Programmstreifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.
5. Überlauf:
   Wird beim Sprung ins Unterprogramm nicht berücksichtigt und beim Verlassen nicht zurückgesetzt.
6. Fehlerstops:

Speicherplatz	Bedeutung und Behebung
$L_o + 8$	N ist Null oder negativ. Der Akkumulator enthält den Wert N - (1@30). Um fortzufahren, springe von Hand.

QUADRATWURZEL 1	B4-10.0

VERWENDUNGSZWECK

Das Programm B4-10.0 berechnet die Quadratwurzel einer positiven Zahl. Die Zahl muß im Akkumulator stehen bei geradem "q". Das Programm benutzt das Newton'sche Verfahren zur Lösung der Gleichung

$$0 = x^2 - N$$

mittels der wiederholten Approximationen

$$x_{i+1} = x_i + [-1/2][-N/x_i + x_i]$$

Das Programm arbeitet wie ein Unterprogramm und wird durch eine Aufruffolge des Hauptprogramms angesprochen. Beim Sprung ins Unterprogramm muß das Argument im Akkumulator stehen, und die Rückkehradresse im Unterprogramm gespeichert sein. Bei der Rückkehr ins Hauptprogramm steht die Quadratwurzel im Akkumulator.

SPEICHERBEDARF 51 Zellen

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariable für "Quadratwurzel 1" ist die positive Zahl N, deren Quadratwurzel berechnet wird.
N muß beim Sprung ins Unterprogramm bei geradem "q" im Akkumulator stehen.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe N bei geradem "q".
α	R	L_o + 50	Speichere die Rückkehradresse im Unterprogramm.
+1	U	L_o	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt N bei geradem "q" in den Akkumulator. (An dieser Stelle kann jeder Befehl benutzt werden, der das Gleiche bewirkt.) Der Befehl in α speichert die Rückkehradresse (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. In α+2 steht der Befehl des Hauptprogramms, der nach Durchlaufen des Unterprogramms als erster ausgeführt wird.

AUSGABE

Bei der Rückkehr ins Hauptprogramm steht die Quadratwurzel des Argumentes N im Akkumulator bei $q_{N/2}$. N stehe beispielsweise @24, dann steht $\sqrt{N}$@12.

BEDIENUNG

1. Genauigkeit:
   Kein Fehler ist größes als 2^{-30}.
2. Zeitbedarf: ca. 775 ms.
3. Programmeingabe:
   Der Streifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.
5. Überlauf:
   Wird nicht beachtet, und bei der Rückkehr nicht zurückgesetzt.
6. Fehlerstops:

Speicherplatz	Bedeutung und Behebung
$L_o + 24$	N ist negativ. Nach START wird der Akkumulator gelöscht, und es erfolgt Rückkehr ins Hauptprogramm.

N-te WURZEL	B4-11.0

VERWENDUNGSZWECK

B4-11.0 berechnet die n-te Wurzel beliebiger nicht negativer Werte. Der Wert n muß eine ganze Zahl sein mit $2 \leq n \leq 20$. Die Ergebnisse werden unter Benutzung eines Iterationsverfahren berechnet nach der Formel:

$$y_{i+1} = y_i + \frac{d - y_i^n}{n y_i^{n-1}}$$

wobei y_i die i-te Näherung der Wurzel ist.

B4-11.0 arbeitet wie ein Unterprogramm und wird durch eine Aufruffolge des Hauptprogramms angesprochen. Beim Sprung ins Unterprogramm muß das Argument im Akkumulator stehen; und die Adresse von n muß dem Unterprogramm angezeigt worden sein. Die auf den Sprungbefehl folgende Speicherzelle muß den Wert von n enthalten. Bei der Rückkehr ins Hauptprogramm steht die gewünschte Wurzel im Akkumulator.

SPEICHERBEDARF — 1 Spur

Benutzte Zwischenspeicher — Die Sektoren 1, 2, 4, 5, 44, 53, 54 und 60 der Spur 63

EINGABE

Die Eingabevariablen für B4-11.0 sind die nicht verschwindenden ganzen Zahlen A, deren Wurzel errechnet wird und der Wert n, der angibt, die wievielte Wurzel errechnet werden soll. Für n gilt: $2 \leq n \leq 20$. Beim Sprung ins Unterprogramm muß A bei einem durch n teilbaren "q" im Akkumulator stehen.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe A @ q_a, q_a muß durch n teilbar sein.
α	R	L_o+16	Adresse von n im Unterprogramm speichern.
+1	U	L_8	Eingang ins Unterprogramm.
+2	Z	08_{nn}	n als dezimale Sektoradresse.
+3	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt den Wert A, der bei einem durch n teilbaren "q" steht, in den Akkumulator. (Jeder Befehl, der dieses bewirkt, kann benutzt werden.) Der Befehl in α speichert die Adresse (α+2), die den Befehl mit n enthält, in das Unterprogramm.

Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Der Befehl in α+2 enthält n als dezimale Sektoradresse. Zelle α+3 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogramms als erster ausgeführt wird.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht $\sqrt{A}\,@\,\frac{q_a}{n}$ im Akkumulator; war z.B. n-3 und q_a = 12 so steht $\sqrt{A}\,@\,4$.

BEDIENUNG

1. Genauigkeit:
   Ein eventueller Fehler erstreckt sich nur auf die neunte Dezimalstelle des Ergebnisses.
2. Zeitbedarf:
   Zwischen 2 und 60 Sekungen; die Zeit liegt höher, wenn man Wurzeln hohan Grades von kleinen Werten errechnet.
3. Erforderliche Eingabe-/Ausgabeeinheiten: Keine.
4. Fehlerstops:

Speicherplatz	Bedeutung und Behebung
L_o + 61	Die Wurzel ist ein imaginärer Wert; d.h. eine gerade Wurzel einer negativen Zahl. START bewirkt Löschen des Akkumulators und Rückkehr ins Hauptprogramm.

5. Überlauf:
   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

MATRIZEN PROGRAMME 1	D1-10.0

In dem Programm D1-10.0 sind fünf Programme zusammengefaßt, die alle das Gleitkomma Interpretiersystem 1 (H1-11.0) benutzen.

Es sind folgende Programme:

Programm Nummer	SC Nummer	Titel
D1-11.0	2044	Matrix-Inversion 1
D1-12.0	2045	Matrix-Vektor-Multiplikation 1
D1-13.0	2046	Matrizen-Multiplikation 1
D1-14.0	2047	Matrizen-Addition/Subtraktion 1
D1-15.0	2048	Matrix-Transponierung 1

MATRIX-INVERSION 1	D1-11.0

VERWENDUNGSZWECK

Benutzt das Gleitkomma Interpretiersystem 1, H1-11.0.

Das Programm D1-11.0 ersetzt die Elemente einer quadratischen Matrix A durch die Elemente ihrer Inversen, $A^{-1}=B$. Die Eingabe besteht aus den Elementen einer quadratischen Matrix, deren Rang größer als 1 ist. Die Daten müssen in Gleitkommaformat angegeben werden. Die Anzahl der Elemente darf den verfügbaren Speicherraum nicht überschreiten.
"Matrix-Inversion 1" arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogramms angesprochen wird. Vor der Aufruffolge muß für Ausgang aus dem Interpretiersystem gesorgt werden. Vor dem Sprung ins Unterprogramm muß die Anfangsadresse des Interpretiersystems im Unterprogramm gespeichert werden. Die Aufruffolge muß den Rang und die Anfangsadresse der Matrix liefern. Bei der Rückkehr ins Hauptprogramm steht im Speicher anstelle der Originalmatrix ihre Inverse.

SPEICHERBEDARF 2 Spuren

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariablen sind die Elemente der N x N Matrix, die in aufeinanderfolgenden Zellen zeilenweise gespeichert werden, wobei das erste Element der zweiten Zeile unmittelbar auf das letzte Element der ersten Zeile folgt. Diese Daten müssen sich in dem für das Gleitkomma Interpretiersystem 1 zulässigen Format befinden. Die Matrix darf nicht singulär sein. Die Anzahl der Elemente darf den verfügbaren Speicherraum nicht überschreiten.
Obwohl für die Matrix nur N^2 Sektoren erforderlich sind, sind insgesamt $N^2 + N$ aufeinanderfolgende Speicherzellen, beginnend in M_o, notwendig.
Bevor der Sprung zur Matrix-Inversion erfolgt, muß die Anfangsadresse des Interpretiersystems in L_o+163 gespeichert werden, wobei L_o die Anfangsadresse der "Matrix-Inversion 1" ist. Dies kann der Bediener von Hand unmittelbar nach Einlesen des Unterprogramms tun, oder es kann ins Hauptprogramm einprogrammiert werden.

AUFRUFFOLGE

Die Aufruffolge muß folgende Informationen liefern:

1. N Rang der Matrix bei q=1 mit N>1.
2. M_o Anfangsadresse der Matrix in dezimaler Spur-/Sektorschreibweise.

N und M_o werden in einem Wort angegeben. Ist N größer als 15, so muß das Wort in hexadezimaler Schreibweise angegeben werden.
Es folgt ein Muster einer Aufruffolge. L_o ist die Anfangsadresse der "Matrix-Inversion 1".

Speicherplatz	Befehl	Adresse	
-1	XE	0000	Ausgang aus dem Interpretiersystem.
α	R	L_o+14	Adresse des Schlüsselwortes ins Unterprogramm einsetzen.
+1	U	L_o	Eingang ins Unterprogramm.
+2	(N)	M_o^o	Angabe von N @ 15 und M_o @ 29.
+3	u.s.w.		Rückkehr ins Hauptprogramm.

Der E-Befehl in α-1 ist nur notwendig, wenn der davorliegende Befehl des Hauptprogramms ein Pseudobefehl des Interpretiersystems ist.

AUSGABE

Das Ergebnis der Matrix-Inversion 1 ist die invertierte N x N Matrix B, die zeilenweise gespeichert ist, beginnend mit dem Element b_{11} in M. Die Elemente von B sind zeilenweise in aufeinanderfolgenden Speicherzellen gespeichert, d.h., das Element b_{21} folgt unmittelbar auf das Element b_{1n}, u.s.w.

BEDIENUNG

1. Programmeingabe
   Das Programm liegt in zwei Fassungen vor:
   a) willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
2. Eingabe-/Ausgabeeinheiten sind nicht erforderlich.
3. Überlauf
   Stört beim Eingang nicht und wird beim Ausgang nicht zurückgesetzt.
4. Fehlerstops

Speicherzelle	Bedeutung	Behebung
L_o + 759 (des Interpretiersystems)	Die Matrix ist singulär	Nicht fortfahren.

BEMERKUNG

Die Matrix-Inversion 1 benutzt das Interpretiersystem. Vor der Rückkehr in das Hauptprogramm ist aber für Ausgang aus dem Interpretiersystem gesorgt.

MATRIX-VEKTOR-MULTIPLIKATION 1	D1-12.0

VERWENDUNGSZWECK

Das Programm benutzt das Gleitkomma Interpretiersystem 1, H1-11.0. Das Programm D1-12.0 multipliziert eine Matrix mit einem Spaltenvektor und speichert den Produktvektor auf vorgegebenen Speicherzellen. Die Eingabe besteht aus einer Matrix mit i Zeilen und j Spalten und aus einem Spaltenvektor mit j Elementen. Die Daten müssen in Gleitkommaformat geliefert werden. Die Matrix braucht nicht quadratisch zu sein. Die Werte i und j müssen kleiner als 64 sein. Die Anzahl der Elemente von Matrix, Vektor und Produktvektor darf den verfügbaren Speicherraum nicht überschreiten.
"Matrix-Vektor-Multiplikation 1" arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogramms angesprochen wird. Vor der Aufruffolge muß für Ausgang aus dem Interpretiersystem gesorgt werden.Die Aufruffolge muß liefern (1) die Anfangsadresse des Interpretiersystems, (2) die Größe und Anfangsadresse der Matrix, (3) die Anfangsadresse des Spaltenvektors und (4) die Anfangsadresse für den Produktvektor. Bei der Rückkehr ins Hauptprogramm ist der Produktvektor auf den angegebenen Speicherzellen gespeichert.

SPEICHERBEDARF 1 Spur, 32 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariablen sind:

1. Die Elemente einer Matrix, mit i Zeilen und j Spalten, die zeilenweise in aufeinanderfolgenden Zellen gespeichert sind, d.h., das erste Element der zweiten Zeile folgt unmittelbar auf das letzte Element der ersten Zeile.
2. Die Elemente eines Spaltenvektors in aufeinanderfolgenden Speicherzellen. Die Anzahl der Elemente des Spaltenvektors muß gleich sein der Anzahl der Spalten der Matrix.

Die Daten müssen in dem für das Gleitkomma-Interpretiersystem zulässigen Format stehen. Jede der Größen i und j muß kleiner als 64 sein; sie brauchen nicht gleich zu sein. Die Gesamtanzahl der Elemente von Matrix, Vektor und Produktvektor darf den verfügbaren Speicherraum nicht überschreiten.

AUFRUFFOLGE

Die folgenden Informationen müssen von der Aufruffolge in der angegebenen Reihenfolge geliefert werden:

1. F_o, die Anfangsadresse des Interpretiersystems in dezimaler Spur-/Sektorschreibweise.
2. M_o, die Anfangsadresse der Matrix in dezimaler Spur-/Sektorschreibweise.

3. Die Größe der Matrix, mit dem Wert i, als dezimale Spuradresse und dem Wert j, als dezimale Sektoradresse.

4. V_o, die Anfangsadresse des Spaltenvektors in dezimaler Spur-/Sektorschreibweise.

5. P_o, die Anfangsadresse für den Produktvektor in dezimaler Spur-/Sektorschreibweise.

Es folgt ein Muster für eine Aufruffolge. L_o ist die Anfangsadresse von "Matrix-Vektor-Multiplikation 1".

Speicherplatz	Befehl	Adresse	
-1	XE	0000	Ausgang aus dem Interpretiersystem.
α	R	L_o	Speichere Adresse für Schlüsselwort des Interpretiersystems ins Unterprogramm.
+1	U	L_o	Sprung ins Unterprogramm.
+2	Z	F_o	Anfangsadresse des Interpretiersystems.
+3	Z	M_o	Anfangsadresse der Matrix.
+4	Z	XXYY	Größe der Matrix, wobei XX die Anzahl der Zeilen ist und YY die Anzahl der Spalten.
+5	Z	V_o	Anfangsadresse des Spaltenvektors.
+6	Z	P_o	Anfangsadresse des Produktvektors.

Der E-Befehl in α-1 ist nur notwendig, wenn der Befehl davor ein Pseudobefehl des Interpretiersystems ist.

AUSGABE

Die Elemente des Produktvektors werden, beginnend in P_o, in aufeinanderfolgenden Zellen gespeichert.

BEDIENUNG

1. Programmeingabe:
   Das Programm liegt in zwei Fassungen vor:
   a) willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
2. Eingabe-/Ausgabeeinheiten sind nicht erforderlich.
3. Überlauf:
   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEMERKUNG

Das Unterprogramm benutzt das Interpretiersystem für alle Gleitkommaoperationen. Vor der Rückkehr ins Hauptprogramm ist aber für Ausgang aus dem Interpretiersystem gesorgt.

MATRIZENMULTIPLIKATION 1	D1-13.0

VERWENDUNGSZWECK

Das Programm benutzt das Gleitkomma Interpretiersystem 1, H1-11.0.

Das Programm D1-13.0 berechnet die Produkt-Matrix C des Ausdrucks AB=C. Das Produkt der nXm Matrix $[a_{ij}]$ und der mXp Matrix $[b_{ij}]$ ist die nXp Matrix $[c_{ij}]$. Die Eingabe besteht aus den Elementen der beiden Matrizen A und B. Die Daten müssen im Gleitkommaformat angegeben werden. Die Größen i und j müssen beide kleiner als 64 sein. Die Gesamtzahl der Elemente der Matrizen A und B und der Produkt-Matrix C dürfen den verfügbaren Speicherraum nicht überschreiten.

"Matrizenmultiplikation 1" arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogramms angesprochen wird. Vor der Aufruffolge muß für Ausgang aus dem Interpretiersystem gesorgt werden. Die Aufruffolge muß liefern: (1) die Anfangsadresse des Interpretiersystems, (2) die Größe und Anfangsadresse der Matrix A, (3) die Größe und Anfangsadresse der Matrix B und (4) die Anfangsadresse des Bereichs für die Produkt-Matrix. Bei der Rückkehr ins Hauptprogramm ist die Produkt-Matrix auf den vorgegebenen Zellen gespeichert.

SPEICHERBEDARF 2 Spuren, 50 Dektoren

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariablen sind:

1. Die Elemente der Matrix A, mit i Zeilen und j Spalten, zeilenweise in aufeinanderfolgenden Zellen gespeichert.
2. Die Elemente der Matrix B, mit i Zeilen und j Spalten, zeilenweise in aufeinanderfolgenden Zellen gespeichert. Die Zeilenzahl von B muß gleich sein der Spaltenzahl von A.

Die Elemente müssen zeilenweise gespeichert werden. D.h., das erste Element der zweiten Zeile folgt in allen Matrizen unmittelbar auf das letzte Element der ersten Zeile.
Die Daten müssen in dem für das Interpretiersystem zulässigen Format stehen. Die Werte i und j müssen beide kleiner als 64 sein.
Die Gesamtzahl der Elemente darf den verfügbaren Speicherraum nicht überschreiten.
Obwohl nur (nXm)+(mXp) Sektoren für die Matrixelemente erforderlich sind, so beträgt der Speicherbedarf doch (nXm)+(mXp)+(nXp) Zellen.

AUFRUFFOLGE

Die Aufruffoge muß folgende Informationen in der angegebenen Reihenfoge liefern:

1. F_o, die Anfangsadresse des Interpretiersystems in dezimaler Spur-/Sektorschreibweise.

2. A_o, die Anfangsadresse der Matrix A in dezimaler Spur-/Sektorschreibweise.

3. Die Größe der Matrix A, mit dem Wert i als dezimaler Spuradresse und mit dem Wert j als dezimaler Sektoradresse.

4. B_o, die Anfangsadresse der Matrix B, in dezimaler Spur-/Sektorschreibweise.

5. Die Größe der Matrix B, mit i als dezimale Spuradresse und j als dezimale Sektoradresse.

6. C_o, die Anfangsadresse des Speicherbereichs für die Produkt-Matrix in dezimaler Spur-/Sektorschreibweise.

Es folgt ein Muster für eine Aufruffolge. L_o ist die Anfangsadresse von "Matrizenmultiplikation 1".

Speicherplatz	Befehl	Adresse	
-1	XE	0000	Ausgang aus dem Interpretiersystem.
α	R	L_o	Speichere die Adresse des Schlüsselwortes für das Interpretiersystem im Unterprogramm.
+1	U	L_o	Eingang ins Unterprogramm.
+2	Z	F_o	Anfangsadresse des Interpretiersystems.
+3	Z	A_o	Anfangsadresse der Matrix A.
+4	Z	XXYY	Größe der Matrix A, wobei XX die Anzahl der Zeilen und YY die Anzahl der Spalten ist.
+5	Z	B_o	Anfangsadresse der Matrix B.
+6	Z	XXYY	Größe der Matrix B, wobei XX die Anzahl der Zeilen und YY die Anzahl der Spalten ist.
+7	Z	C_o	Anfangsadresse der Produkt-Matrix.
+8	u.s.w.		Rückkehr zum Hauptprogramm.

Der E-Befehl in α-1 ist nur notwendig, wenn der Befehl davor ein Pseudobefehl des Interpretiersystems ist.

AUSGABE

Die Ausgabe des Programms "Matrizenmultiplikation 1" ist die Produkt-Matrix C, mit i Zeilen und j Spalten, die zeilenweise in aufeinanderfolgenden Zellen gespeichert sind; das Element C_{11} liegt in Speicherzelle C_o. Die Elemente der Matrix C sind zeilenweise in aufeinanderfolgenden Speicherzellen gespeichert; d.h., das Element C_{21} folgt unmittelbar auf das Element C_{1n}, u.s.w.

BEDIENUNG

1. Programmeingabe:
   Das Programm liegt in zwei Fassungen vor:
   a) willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
2. Eingabe-/Ausgabeeinheiten sind nicht erforderlich.
3. Fehlerstops:

Speicherplatz	Bedeutung	Behebung
L_o + 241	Spaltenzahl von A $\neq$ Zeilenzahl von B.	Nicht starten. Daten müssen berichtigt werden.

4. Überlauf:
   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEMERKUNG

Das Unterprogramm benutzt das Interpretiersystem. Vor der Rückkehr ins Hauptprogramm ist aber für Ausgang aus dem Interpretiersystem gesorgt.

MATRIX-ADDITION UND SUBTRAKTION 1	D1-14.0

VERWENDUNGSZWECK

Das Programm benutzt das Gleitkomma Interpretiersystem, H1-11.0.

Das Programm D1-14.0 addiert oder subtrahiert zwei Matrizen A und B und speichert die Ergebnismatrix auf den vorgegebenen Speicherbereich. Die Eingabe besteht aus den Elementen der beiden Matrizen, von denen jede i Zeilen und j Spalten hat. Die Daten müssen im Gleitkommaformat angegeben werden. Die Matrizen brauchen nicht quadratisch zu sein. Die Werte i und j müssen beide kleiner als 64 sein. Die Gesamtzahl der Elemente der beiden Matrizen A und B und der Ergebnismatrix dürfen den verfügbaren Speicherraum nicht überschreiten.

Das Programm arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogramms angesprochen wird. Vor der Aufruffolge muß für Ausgang aus dem Interpretiersystem gesorgt werden. Die Aufruffolge muß liefern: (1) die Anfangsadresse des Interpretiersystems, (2) die Größen und Anfangsadressen der Matrizen, (3) die Art der gewünschten Rechenoperation, und (4) die Anfangsadresse des Speicherbereichs für die Ergebnismatrix. Bei der Rückkehr ins Hauptprogramm steht die Ergebnismatrix auf dem vorgegebenen Speicherbereich.

SPEICHERBEDARF 1 Spur

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariablen sind die Elemente der beiden Matrizen A und B, von denen jede i Zeilen und j Spalten hat. Die Elemente der Matrizen müssen entweder bei beiden zeilenweise oder bei beiden spaltenweise in aufeinanderfolgenden Zellen gespeichert werden: das erste Element der zweiten Zeile folgt unmittelbar auf das letzte Element der ersten Zeile, wenn die Elemente zeilenweise gespeichert sind; oder das erste Element der zweiten Spalte folgt unmittelbar auf das letzte Element der ersten Spalte, wenn die Elemente spaltenweise genommen werden.
Die Daten müssen in dem für das Gleitkomma-Interpretiersystem zulässigen Format stehen. Die Matrizen müssen gleich groß sein, brauchen aber nicht quadratisch zu sein. Die Werte i und j müssen beide kleiner als 64 sein. Die Gesamtzahl der Elemente der Matrizen darf den verfügbaren Speicherraum nicht überschreiten.
Obwohl nur 2(ij) Sektoren für die Elemente der Eingabematrizen erforderlich sind, so sind doch insgesamt 3(ij) Speicherzellen erforderlich.

AUFRUFFOLGE

Die folgenden Informationen müssen durch die Aufruffolge in der angegebenen Reihenfolge geliefert werden:

1. F_{8}, die Anfangsadresse des Interpretiersystems in dezimaler Spur-/Sektorschreibweise.

2. A_o, die Anfangsadresse der Matrix A in dezimaler Spur-/Sektorschreibweise.

3. Die Größe der Matrizen, durch den Wert i als dezimaler Spuradresse und durch den Wert j als dezimaler Sektoradresse.

4. Der Befehl A oder S, je nachdem ob es sich um Addition oder Subtraktion handelt. Das Befehlssymbol steht im Befehlsteil. Im gleichen Wort wird die Anfangsadresse der Matrix B als dezimale Spur-/Sektoradresse angegeben.

5. C_o, die Anfangsadresse des Speicherbereiches für die Ergebnismatrix in dezimaler Spur-/Sektorschreibweise.

Es folgt ein Muster für eine Aufruffolge, L_o ist die Anfangsadresse von "Matrix-Addition und Subtraktion 1".

Speicherplatz	Befehl	Adresse	
-1	XE	0000	Ausgang aus dem Interpretiersystem.
α	R	L_o	Speichere Adresse für das Schlüsselwort des Interpretiersystems ins Unterprogramm.
+1	U	L_o	Eingang ins Unterprogramm.
+2	Z	F_o	Anfangsadresse des Interpretiersystems.
+3	Z	A_o	Anfangsadresse der Matrix A.
+4	Z	XXYY	Größe der Matrizen, wobei XX die Zeilenzahl und YY die Spaltenzahl ist.
+5	(A oder S)	B_o	Addition oder Subtraktion und Anfangsadresse der Matrix B.
+6	Z	C_o	Anfangsadresse des Speicherbereichs für die Ergebnismatrix.
+7	u.s.w.		Rückkehr zum Hauptprogramm.

AUSGABE

Die Matrix C_{ij}, wird im Gleitkommaformat in aufeinanderfolgenden Zellen, beginnend in C_o, gespeichert.

BEDIENUNG

1. Programmeingabe:
   Der Programmstreifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

2. Eingabe-/Ausgabeeinheiten sind nicht erforderlich.

3. Überlauf:
   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEMERKUNGEN

Das Programm benutzt das Interpretiersystem für alle Gleitkommaoperationen. Vor der Rückkehr ins Hauptprogramm wird aber für Ausgang aus dem Unterprogramm gesorgt.
Die Ergebnismatrix kann auf die Zellen gelegt werden, die von den Matrizen A oder B besetzt sind. Wird ein anderer Speicherbereich genommen, so darf er sich weder mit A noch mit B überschneiden.

MATRIZENTRANSPONIERUNG 1	D1-15.0

VERWENDUNGSZWECK

Das Programm D1-15.0 stürzt eine quadratische Matrix, die im Speicher steht, und speichert die Transponierte auf vorgegebene Zellen. Die Eingabe besteht aus den Elementen einer quadratischen Matrix, deren Rang kleiner als 32 ist. Es sind sowohl Festkomma als auch Gleitkommadaten möglich. Die Gesamtzahl der Elemente darf den verfügbaren Speicherbereich nicht überschreiten.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogramms angesprochen wird. Vor der Aufruffolge muß für Rückkehr aus dem Interpretiersystem gesorgt werden. Die Aufruffolge muß liefern, 1. den Rang und die Anfangsadresse der Matrix, und 2. die Anfangsadresse für die Transponierte. Bei der Rückkehr in das Hauptprogramm ist die Transponierte auf den angegebenen Zellen gespeichert.

SPEICHERBEDARF 1 Spur, 58 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariablen sind die Elemente einer quadratischen Matrix, die zeilenweise oder spaltenweise in aufeinanderfolgenden Zellen gespeichert sind: das erste Element der zweiten Zeile folgt unmittelbar auf das letzte Element der ersten Zeile bei zeilenweiser Speicherung; das erste Element der zweiten Spalte folgt unmittelbar auf das letzte Element der ersten Spalte bei spaltenweiser Speicherung. Die Daten können in Festkommaformat oder im Gleitkommaformat, das für das Gleitkomma Interpretiersystem 1, H1-11.0 zulässig ist, angegeben werden. Die Gesamtzahl der Elemente darf den verfügbaren Speicherraum nicht überschreiten.
Obwohl für die Matrixelemente nur n^2 Sektoren erforderlich sind, werden doch maximal $2(n^2)$ Zellen benötigt.

AUFRUFFOLGE

Die Aufruffolge muß die folgenden Informationen in der angegebenen Reihenfolge liefern:

1. N den Rang der Matrix bei q=15. Für N gilt: $2 \leq N \leq 31$.
2. M_o die Anfangsadresse der Matrix in dezimaler Spur-/Sektorschreibweise.
3. T_o die Anfangsadresse des Speicherbereiches für die Transponierte in dezimaler Spur-/Sektorschreibweise.

N und M_o werden im gleichen Wort angegeben. Wenn N größer als 15 ist, muß das Wort in hexadezimaler Schreibweise angegeben werden. Die Adressen M_o und T_o dürfen gleich sein; soll aber die Transponierte auf einem anderen Speicherbereich liegen, so darf dieser sich nicht mit dem Bereich der Ausgangsmatrix überschneiden.

Es folgt das Muster einer Aufruffolge. L_o ist die Anfangsadresse von "Matrizentransponierung 1".

Speicherplatz	Befehl	Adresse	
-1	XE	0000	Ausgang aus dem Interpretiersystem.
α	R	L_o	Adresse des Schlüsselwortes für Rang und Anfangsadresse der Matrix im Unterprogramm speichern.
+1	U	L_o	Eingang ins Unterprogramm.
+2	(N)	M_o^o	N @ 15 und Anfangsadresse der Matrix.
+3	Z	T_o	Anfangsadresse für die Transponierte.
+4	u.s.w.		Rückkehr zum Hauptprogramm.

Der Befehl in α-1 ist nur erforderlich, wenn der vorherige Befehl ein Pseudobefehl des Interpretiersystems ist.

AUSGABE

Die Elemente der Transponierten ab T_o gespeichert. Stand die Originalmatrix zeilenweise, so stehen in der Transponierten die Spalten zeilenweise; stand die Originalmatrix spaltenweise, so stehen in der Transponierten die Zeilen spaltenweise.

BEDIENUNG

1. Programmeingabe:
   Der Programmstreifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.
2. Eingabe-/Ausgabeeinheiten sind nicht erforderlich.
3. Überlauf:
   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEMERKUNG

Das Unterprogramm ruft das Gleitkomma Interpretiersystem 1 nicht auf.

INTERPRETIERSYSTEM FÜR KOMPLEXE OPERATIONEN	H1-10.0

VERWENDUNGSZWECK

Das Programm erlaubt dem Benutzer algebraische Ausdrücke der Form (a+bi) aus dem Bereich der Komplexen Zahlen so zu verwenden, als handelte es sich um reelle Zahlen. Auch Links- und Rechtsschiften ist mit dem Programm möglich. Schließlich gestattet es die Änderung von Befehlen und Tests auf die Endadresse ohne das interpretierende Verfahren zu verlassen.

SPEICHERBEDARF 3 Spuren

Benutzte Zwischenspeicher keine

EINGABE

Von jedem Datenelement muß Real- und Imaginärteil beim gleichen "q" in aufeinanderfolgenden Zellen gespeichert werden; d.h., der Realteil in Zelle ttss und der Imaginärteil in Zelle ttss+1.
Da jede komplexe Zahl zwei Zellen belegt, sind simulierte Register vorgesehen, die die Ausführung der Befehle erleichtern. Diese Register sind der simulierte Akkumulator in Zelle L_o+33 und das Adressregister in L_o+219, wobei L_o die Anfangsadresse des Unterprogrammes ist. Zusätzlich zu diesen Registern zeichnet das Unterprogramm den Inhalt des Maschinenakkumulators in Zelle L_o+59 auf.
15 Befehle für komplexe Operationen werden vom Unterprogramm interpretiert. Bei den arithmetischen Befehlen bezieht sich ttss auf den Realteil der komplexen Größen, aber beide Zahlenteile werden vom Programm verarbeitet. Bei anderen Befehlen ist ttss eine Standard-Einwortadresse. Nur die erläuterten Befehle dürfen in einer komplexen Operationenfolge stehen. Die Register, auf die sie sich beziehen, sind die simulierten Register.

1. ARITHMETISCHE BEFEHLE

Bttss Bringe
Die Inhalte von ttss und ttss+1 ersetzen die Inhalte des Pseudoakkumulators. Der Speicherinhalt bleibt unverändert.

Attss Addiere
Die Inhalte von ttss und ttss+1 plus die Inhalte des Pseudoakkumulators ersetzen die Inhalte des Pseudoakkumulators. Der Speicherinhalt bleibt unverändert.

Sttss Subtrahiere
Die Inhalte des Pseudoakkumulators minus die Inhalte von ttss und ttss+1 ersetzen die Inhalte des Pseudoakkumulators. Der Speicherinhalt bleibt unverändert.

Mttss Multipliziere
Die Inhalte des Pseudoakkumulators multipliziert mit den Inhalten von ttss und ttss+1 ersetzen die Inhalte des Pseudoakkumulators. Die Speicherinhalte bleiben unverändert.

Dttss Dividiere
Die Inhalte des Pseudoakkumulators dividiert durch die Inhalte von ttss und ttss+1 ersetzen die Inhalte des Pseudoakkumulators. Der Speicherinhalt bleibt unverändert.

Httss Halte
Die Inhalte des Pseudoakkumulators ersetzen die Inhalte von ttss und ttss+1. Der Inhalt des Pseudoakkumulators bleibt unverändert.

Cttss Lösche
Die Inhalte des Pseudoakkumulators ersetzen die Inhalte von ttss und ttss+1, und der Pseudoakkumulator wird gelöscht.

2. HILFSBEFEHLE

XROOnn Rechtsschiften
Schifte die Inhalte des Pseudoakkumulators nn Dualstellen nach rechts. $0 \leq nn \leq 10$.

XPOOnn Linksschiften
Schifte die Inhalte des Pseudoakkumulators nn Dualstellen nach links. $0 \leq nn \leq 10$.

Uttss Unbedingter Sprung
Der nächste auszuführende Befehl steht in Zelle ttss. Dieser Sprungbefehl kann nicht benutzt werden um das Programm zu verlassen; d.h. ttss muß einen "komplexen Befehl" enthalten.

XE0000 Ausgang
Ausgang aus dem Interpretiersystem. Der auf XE0000 folgende Befehl wird als gewöhnlicher Maschinenbefehl ausgeführt.

3. BEFEHLE ZUR ADRESSENÄNDERUNG

Ettss Adressregister setzen
Der Adressteil des Wortes in ttss wird in das Adressregister gesetzt. Der Speicherinhalt bleibt unverändert.

Ittss Adressenerhöhung
Erhöhe den Inhalt des Adressregisters um den Wert ttss.

Yttss Adresse einsetzen
Speichere den Inhalt des Adressregisters in den Adressteil des Wortes in ttss. Das Adressregister und der Rest des Speichers bleibt unverändert.

Zttss Teste auf Null und überspringe
Subtrahiere den Inhalt des Adressregisters vom Wert ttss. Ist das Ergebnis Null, so wird der nächste Befehl übersprungen; d.h. der übernächste Befehl wird ausgeführt. Ist das Ergebnis ungleich Null, so wird der nächste Befehl ausgeführt.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
∝	R	L_o	Eingang in das Unterprogramm
+1	U	L_o	
+2			Komplexe Befehle
.			
.			
.			
+n	XE	0000	Ausgang aus dem Unterprogramm
+n+1	u.s.w.		

INTERPRETIERSYSTEM FÜR KOMPLEXE OPERATIONEN	H1-10.0

Der Befehl in α speichert die Adresse (α+2) des ersten komplexen Befehles im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. Die Zellen α+2 bis α+n enthalten die Folge von komplexen Befehlen; der Befehl in α+n bewirkt Ausgang aus dem Interpretiersystem. Der Befehl in α+n +1 ist der Maschinenbefehl des Hauptprogrammes, der nach der Interpretation als erster ausgeführt wird.

BEDIENUNG

1. ZEITBEDARF

Die folgende Tabelle gibt die Maximalzeiten an, die für die Ausführung der jeweiligen komplexen Operation erforderlich sind.

Befehl	Trommelumdrehungen	Millisekunden
B	11	561
A	14	714
S	14	714
M	22	1122
D	41	2091
H	14	714
C	14	714
R	14	714
P	17	1167
U	8	408
E(Ausgang)	6	306
E(Adresse setzen)	9	459
I	6	306
Y	8	408
Z($\neq$)	11	561
Z(=)	13	663

2. PROGRAMMEINGABE

Der Streifen enthält das Programm in zwei Fassungen:
a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
b) willkürlich verlegbar dezimal in Kodierungsblattformat.

3. ERFORDERLICHE EINGABE-/AUSGABEEINHEITEN

Weder Eingabe- noch Ausgabeeinheiten sind erforderlich.

4. FEHLERSTOPS

Speicherzelle	Bedeutung	Behebung
L_o + 122	Negativer Befehl	nicht fortfahren
L_o + 135	N - Befehl	nicht fortfahren
L_o + 154	T - Befehl	nicht fortfahren

5. ÜBERLAUF

Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEMERKUNG

Der erste komplexe Befehl darf kein XE0000 sein, da kein Ausgang erfolgen würde.

GLEITKOMMA-INTERPRETIERSYSTEM 1	H1-11.0

EINFÜHRUNG

Das Gleitkomma-Interpretiersystem 1 erlaubt es dem Programmierer, Berechnungen im Gleitkomma gegenüber Berechnungen im Festkomma zu bevorzugen. Die Daten werden dem System geliefert als 7-ziffrige ganze Zahlen multipliziert mit einer Zehnerpotenz; das Interpretiersystem skaliert die Daten automatisch während der Rechnung und liefert die Ergebnisse als 7-ziffrige ganze Zahlen, multipliziert mit einer Zehnerpotenz. Obwohl es sich grundsätzlich um ein Ein-Wort-Interpretiersystem handelt, so sind doch zwei Worte nötig für jede Zahl während der Rechnungen und für die Ausgabe. Mantisse und Exponent werden automatisch getrennt und wieder zusammengesetzt.

Eine genaue Kenntnis der Größenordnung der Variablen eines Problems ist nicht erforderlich. Das System verfügt über 33 Befehle, wodurch ebenfalls Leichtigkeit und Genauigkeit der Programmierung verbessert sind.

Das Interpretiersystem besteht aus verschiedenen Unterprogrammen - Dateneingabe, Datenausgabe, Berechnung von Funktionswerten - die als Programmpaket oder einzeln benutzt werden können. Die Unterprogramme werden durch bestimmte Befehle aufgerufen, so daß keine Aufruffolgen nötig sind; das Interpretiersystem selbst und das Unterprogramm "Festkomma-Gleitkomma-Umwandlung und umgekehrt" werden durch eine Aufruffolge angesprochen.

Die vorliegende Schrift enthält Beschreibungen der folgenden Unterprogramme und spezielle Eigenheiten des Interpretiersystems:

INTERNES ZAHLENFORMAT

Jede Gleitkommazahl wird im Speicher in einer einzigen Zelle gehalten als Mantisse mit Exponent. Jede Mantisse wird in dualer Schreibweise bei "q" = 0 gehalten und besteht aus algebraischem Vorzeichen und 24 Bits. Das meistbedeutende Bit hat den Wert 2^{-1}, d.h., alle gespeicherten Mantissen sind normalisiert.
Der Exponent wird bei "q" = 30 gehalten und besteht aus algebraischem Vorzeichen und 5 Bits. Der Exponent ist die Potenz zur Basis2, mit der die Mantisse multipliziert werden muß, um den wirklich dargestellten Zahlenwert zu erhalten. Kurz ausgedrückt heißt das, daß die Faktoren einer Gleitkommazahl diese Zahl darstellen als

$$Z = M \cdot 2^{E}$$

Dabei bedeutet: Z = Zahl
M = Mantisse
E = Exponent

In Gleitkommaform würde 3,75 im Rechner dargestellt als

```
0111100000000000000000000000100
^                             ^
q = 0                         q = 30
```

Wobei: M = 0,9375
E = 2
$Z = 0{,}9375 \cdot 2^{2} = 3{,}7500$ ist.

Mantisse und Exponent werden vom System als zwei verschiedene Zahlen behandelt. Daher darf eine negative Mantisse und ein positiver Exponent im gleichen Wort enthalten sein. D.h., die Teile einer Zahl werden unabhängig voneinander komplementiert. Beispielsweise würde -3,75 dargestellt als

```
1000100000000000000000000000100
^                             ^
q = 0                         q = 30
```

Wobei: M = -0,9375
E = 2
$Z = -0{,}9375 \cdot 2^{2} = -3{,}7500$ ist.

Als weiteres Beispiel würde 0,46875 in Gleitkommaform dargestellt sein als

```
0111100000000000000000000001111110
^                             ^
q = 0                         q = 30
```

Wobei: M = 0,9375
E = -1
$Z = 0{,}9375 \cdot 2^{-1} = 0{,}46875$ ist.

Im Gleitkomma-Akkumulator oder im Multiplikationsregister sind die Zahlen so eingerichtet, daß sie zwei Zellen belegen, wobei die Mantisse eine Stelle nach rechts geschiftet ist. Dennoch wird die Beziehung zwischen jeder Mantisse und ihrem Exponenten aufrechterhalten.

INTERNE REGISTER

Obwohl im Speicher jede Zelle eine Mantisse und deren Exponenten enthält, sind zwei Zellen erforderlich, um bei Zwischenrechnungen und bei der Ausgabe eine Zahl aufzunehmen: nämlich eine Zelle für die Mantisse und eine für den Exponenten. Im Interpretiersystem sind ein spezieller Akkumulator, ein Multiplikationsregister, ein Adressenregister und ein Zählregister

eingerichtet, damit die Daten dem System zur Verarbeitung aus dem Speicher zugeführt werden können. Akkumulator und Multiplikationsregister belegen jeweils zwei Zellen; Adressenregister und Zählregister jeweils nur eine. Alle Daten bleiben in Gleitkommaform. Adressen können modifiziert werden, ohne daß das Interpretiersystem verlassen wird.
Die Inhalte aller Register sind anfangs Null; d.h., sie sind Null, wenn das Programm gerade erst eingelesen wurde und noch nicht benutzt ist. Die Inhalte aller Register, jedoch nicht des Zählers, sind anfangs gesetzt und werden nur durch spezielle Befehle verändert. Daher kann das Interpretiersystem verlassen und neu aufgerufen werden, ohne daß die Inhalte der Register verändert werden. Der Zähler wird bei jedem Eingang ins Interpretiersystem gesetzt und wird bei der Ausführung eines jeden Befehls erhöht.
Diese speziellen Register werden im einzelnen in diesem Abschnitt erläutert. Alle erwähnten Befehle werden unter "Gleitkommabefehle" erwähnt.

1. Der Gleitkomma-Akkumulator

   Der Gleitkomma-Akkumulator hält Zwischenergebnisse. Obwohl die Mantisse im Speicher normalisiert stand, wird sie im Gleitkomma-Akkumulator um eine Dualstelle nach rechts geschiftet. D.h., die Mantisse steht bei q = 0, wobei das meistbedeutende Bit um eine Stelle nach rechts geschiftet ist; der Exponent steht bei q = 30. Beide Zahlenteile haben ihr zugehöriges algebraisches Vorzeichen. (Siehe "Internes Zahlenformat".) Die Inhalte des Akkumulators werden nur durch arithmetische Operationen geändert, durch die Befehle: "Löschen","Vertauschen", "Vorzeichen setzen" oder bei der Berechnung eines Funktionswertes.

2. Das Multiplikationsregister

   Das Multiplikationsregister hält den Multiplikanden bei den Befehlen für "Multiplizieren" und beim Befehl für "kumulative Multiplikation". Die Mantisse steht bei q = 0, wobei das meistbedeutende Bit um eine Stelle nach rechts verschoben ist; der Exponent steht bei q = 30. Beide Teile sind mit ihrem algebraischen Vorzeichen versehen. Die Inhalte dieses Registers werden nur durch die Befehle "Setzen" und "Vertauschen" geändert.

3. Das Adressenregister

   Das Adressenregister enthält eine einzige Zelle in Spur-/Sektorschreibweise bei q = 29. Dieses Register wird benutzt, um den Operandenadressteil von Worten des Speichers zu verändern. Dabei wird der Befehl "Adresse speichern" benutzt. Außerdem findet dieses Register Verwendung beim "Test auf Null". Der Inhalt des Registers kann nur verändert werden durch einen Befehl "Adresse setzen" oder "Adresse erhöhen".

4. Das Zählregister

Das Zählregister enthält die Adresse des Gleitkommabefehls, der gerade ausgeführt wird und zeigt die Adresse des nächsten auszuführenden Befehls an. Bei der Ausführung des Befehls "Rückkehr" wird der Inhalt des Adressteils der Zelle, auf den sich der Rückkehrbefehl bezieht, ersetzt durch den Inhalt des Zählregisters plus 2. Der Inhalt des Registers wird zu Anfang durch die Aufruffolge des Hauptprogramms gesetzt und bei der Ausführung eines jeden Befehles um 1 erhöht. Diese Arbeitsweise wird nur durch einen Sprungbefehl unterbrochen, indem dann der Registerinhalt neu gesetzt wird.

GLEITKOMMABEFEHLE

Dieses Interpretiersystem arbeitet mit 33 Pseudobefehlen. Jeder Befehl hat einen Operationsteil und einen Operandenadressteil. Einige der Operandenadressen haben gegenüber Festkommabefehlen abweichende Bedeutung, z.B. die Nulladressen. Die Gleitkommabefehle sind auf den folgenden Seiten gelistet und erläutert. Die Buchstaben ttss stehen für eine Adresse in dezimaler Spur- und Sektorschreibweise.
Alle erwähnten Register sind die simulierten Gleitkommaregister des Interpretiersystems. (Das M Register ist das Multiplikationsregister.) Alle Gleitkommabefehle werden ausgeführt, ohne daß das Interpretiersystem verlassen wird.

1. Arithmetische Befehle

B ttss Bringe

Der Inhalt von Zelle ttss wird in den Akkumulator gebracht. Der Speicherinhalt wird bei der Operation nicht verändert.

A ttss Addiere

Der Inhalt des Akkumulators plus der Inhalt der Zelle ttss wird zum Inhalt des Akkumulators. Der Speicherinhalt wird bei der Operation nicht geändert.

S ttss Subtrahiere

Der Inhalt des Akkumulators minus der Inhalt der Zelle ttss wird zum Inhalt des Akkumulators. Der Speicherinhalt wird bei der Operation nicht verändert.

D ttss Dividiere

Der Inhalt des Akkumulators dividiert durch den Inhalt der Zelle wird zum Inhalt des Akkumulators. Der Speicherinhalt wird bei der Operation nicht verändert.

P ttss M Register setzen

Der Inhalt der Zelle ttss wird zum Inhalt des M Registers. Der Speicherinhalt wird bei der Operation nicht verändert.

M ttss Multiplizieren

Der Inhalt des M Registers multipliziert mit dem Inhalt der Zelle ttss wird zum Inhalt des Akkumulators. Der Speicherinhalt und der Inhalt des M Registers werden bei der Operation nicht verändert.

N ttss Kumulative Multiplikation

Der Inhalt des M Registers wird multipliziert mit dem Inhalt der Zelle ttss, und das Produkt wird zum Akkumulatorinhalt addiert (Ergebnis im Akkumulator). Die Inhalte des Speichers und des M Registers werden bei der Operation nicht verändert.

D 000y Rechtsschiften

Der Inhalt des Akkumulators, durch 2^y dividiert, wird zum Inhalt des Akkumulators. Die Operandenadresse muß sein 000y, wobei für y gilt: $0 \leq y \leq 9$. Der Inhalt des Akkumulators bleibt in Gleitkommaform.

M 000y Linksschiften

Der Inhalt des Akkumulators, multipliziert mit 2^y, wird zum Inhalt des Akkumulators. Die Operandenadresse hat das Aussehen 000y, wobei für y gilt: $0 \leq y \leq 9$. Der Inhalt des Akkumulators bleibt in Gleitkommaform.

H ttss Halte

Der Inhalt des Akkumulators wird zum Inhalt der Zelle ttss. Der Inhalt des Akkumulators wird bei der Operation nicht verändert.

C ttss Lösche

Der Inhalt des Akkumulators wird zum Inhalt der Zelle ttss, und der Akkumulator wird mit Nullen gefüllt.

2. Sprungbefehle

U ttss Unbedingter Sprung

Die Adresse ttss wird zum Inhalt des Zählregisters. Daher ist der nächste Befehl, der ausgeführt wird, der in Zelle ttss stehende. Dieser Befehl darf nicht zum Verlassen des Interpretiersystems benutzt werden.

T ttss Teste auf Minus

Die Adresse ttss wird zum Inhalt des Zählregisters, wenn der Inhalt des Akkumulators negativ ist; in diesem Fall steht der nächste ausgeführte Befehl in Zelle ttss. Andernfalls wird der auf den Sprungbefehl folgende Befehl ausgeführt. Dieser Befehl darf nicht zum Verlassen des Interpretiersystems benutzt werden.

800T ttss Springe

Die Adresse ttss wird zum Inhalt des Zählregisters, wenn der Inhalt des Akkumulators negativ ist oder wenn PST eingeschaltet ist; der nächste ausgeführte Befehl steht dann in Zelle ttss. Andernfalls wird der auf den Sprungbefehl folgende Befehl ausgeführt. Dieser Befehl darf nicht benutzt werden, um das Interpretiersystem zu verlassen.

3. Befehle zur Adressenmodifikation

E ttss Adresse setzen

Der Adressteil des Befehls in Zelle ttss wird zum Inhalt des Adressregisters. Der Speicherinhalt wird bei der Operation nicht verändert.

I ttss Adressenerhöhung

Der Inhalt des Adressregisters wird erhöht um den Wert ttss. Der Befehl kann auch zur Erniedrigung des Adressregisterinhaltes benutzt werden, indem man das Komplement des Wertes ttss bildet.

Y ttss Adresse speichern

Die im Adressregister enthaltene Adresse wird zum Adressteil des Befehls in Zelle ttss. Der Inhalt des Adressregisters und der Rest der Zelle ttss werden von der Operation nicht verändert.

Z ttss Teste auf Null

Der Wert ttss wird vom Inhalt des Adressregisters subtrahiert. Ist das Ergebnis ungleich Null, so wird der nächstfolgende Befehl ausgeführt. Ist das Ergebnis Null, so wird der nächste folgende Befehl übersprungen und der zweite folgende (übernächste) wird ausgeführt.

R ttss Rückkehr

Der Inhalt des Zählregisters plus 2 wird zum Adressteil des Befehls in Zelle ttss. Der Inhalt des Zählers und der Rest der Zelle ttss werden bei der Operation nicht verändert.

4. Hilfsbefehle

Die Operandenadresse der folgenden Befehle muß Null sein.

U 0000 Vertausche

Die Inhalte vom M Register und Akkumulator werden vertauscht.

B 0000 Setze Pluszeichen

Der Inhalt des Akkumulators wird positiv gemacht, wenn er negativ ist. Wenn der Akkumulatorinhalt schon positiv ist, wird der Befehl übergangen.

T 0000 Setze Minuszeichen

Der Akkumulatorinhalt wird negativ gemacht, wenn er positiv ist. Wenn der Inhalt schon negativ ist, wird der Befehl übergangen.

Y 0000 Vorzeichenumkehr

Das Komplement des Akkumulatorinhalts wird zum Akkumulatorinhalt.

Z 0000 Stop

Die Rechnung hält an, wenn PS-16 ausgeschaltet ist. "Start" drücken bewirkt Fortsetzung der Operation in der nächsten Zelle. Ist PS-16 eingeschaltet, so wird der Stop übergangen.

E 0000 Ausgang

Ausgang aus dem Interpretiersystem. Der nächstfolgende Befehl wird als Maschinenbefehl ausgeführt.

5. Eingabe-/Ausgabebefehle

Die Operandenadresse der folgenden Befehle muß Null sein.

I 0000 Eingabe

Lese und speichere Daten in Gleitkommaform zur Benutzung durch das Hauptprogramm. Die Anfangsadresse wird auf dem Datenstreifen angegeben.

P 0000 Ausgabe

Gebe den Inhalt des Akkumulators in Gleitkommaschreibweise aus. Der Inhalt des Akkumulators wird bei der Operation nicht verändert.

6. Befehle zur Berechnung von Funktionswerten

Der Adressteil der folgenden Befehle muß Null sein.

R 0000 Quadratwurzel

Die Quadratwurzel aus dem Inhalt des Akkumulators wird zum Inhalt des Akkumulators.

S 0000 Sinus

Der Sinus aus dem Akkumulatorinhalt wird zum Akkumulatorinhalt. Das Argument muß in Bogenmaß vorliegen.

C 0000 Cosinus

Der Cosinus des Akkumulatorinhalts wird zum Akkumulatorinhalt. Das Argument muß in Bogenmaß vorliegen.

A 0000 Arcustangens

Der Arcustangens aus dem Akkumulatorinhalt wird zum Akkumulatorinhalt. Das Ergebnis steht im Bogenmaß.

N 0000 Natürlicher Logarithmus

Der Log zur Basis "e" des Akkumulatorinhalts wird zum Akkumulatorinhalt.

H 0000 Exponentialfunktion

Die Größe e^x wird zum Akkumulatorinhalt, wobei der Wert x der ursprüngliche Inhalt des Akkumulators ist.

DATENEINGABE UND AUSGABE 1, Programm J9-10.0

1. Dateneingabe

"Dateneingabe und Ausgabe 1" führt Daten in den Rechner ein zur Benutzung durch ein Hauptprogramm. Dezimalzahlen werden gelesen, in duale Gleitkommaform umgewandelt und auf den angegebenen Zellen in der erforderlichen Weise gespeichert. Dieses Unterprogramm wird aufgerufen durch einen I0000 Befehl des Hauptprogramms, und es erfolgt ein Rücksprung zu der auf den Eingabebefehl folgenden Zelle, wenn alle Daten gespeichert worden sind.

Die Eingabedaten müssen die folgenden Ausdrücke in der beschriebenen Reihenfolge und Form enthalten:

a. Eingabeschlüsselwort

Der erste Ausdruck in jeder Gruppe von Daten muß ein Wort sein, das die Anzahl der Dezimalstellen P eines jeden Wortes der Gruppe und die Anfangsadresse ttss der Gruppe angibt. Der Wert P muß als zweiziffrige Dezimalzahl mit algebraischem Vorzeichen angegeben werden, wobei gilt $-03 \leq P \leq +16$. Die Anfangsadresse muß eine absolute Adresse in dezimaler Spur-/Sektorschreibweise sein. Ein positiver P Wert zeigt an, daß die Kommastelle nach links verschoben liegt; ein negativer, daß sie nach rechts verschoben liegt.

b. Daten

Die Daten werden ausgedruckt als ganze Dezimalzahlen mit bis zu 7 Ziffern plus Vorzeichen. Das Vorzeichen und linksseitige Nullen können weggelassen werden, wenn die Zahl positiv ist; bei negativen Zahlen müssen die linksleitigen Nullen zwischen Vorzeichen und Zahl jedoch angegeben werden. Für eine Null braucht nur ein Stopcode (') geschrieben zu werden.

Beispiel: Das Eingabeschlüsselwort ist +032400', die Zahl 16,192 würde angegeben als 16192'.

c. Schlüsselwort für Ende des Einlesens und Minusnullwort

Das Ende einer Gruppe wird angegeben durch das Wort -0000000'. Dieses Wort wird nicht vom Unterprogramm gespeichert.
Das Ende des Einlesens wird angezeigt durch einen zweiten Stopcode (') nach dem Minusnullwort; d.h., -0000000''.
Wenn das Minusnullwort gelesen ist, liest das Programm das nächste Eingabeschlüsselwort, wandelt die Daten jetzt entsprechend dem neuen P Wert um und speichert sie beginnend mit der neuen Anfangsadresse. Wenn der Stopcode für das Ende des Einlesens gelesen ist, erfolgt ein Wagenrücklauf und Rückkehr ins Hauptprogramm.

Das folgende Beispiel zeigt einen typischen Datensatz:

Eingabe-schlüssel-wort			Stop	Dezimal-zahlen m. Vorzeichen	Stop	
±	P	Zelle		±		WR
+	07	2004	'	1090000	'	
				120000	'	X
				340000	'	
				560000	'	X
				780000	'	
				900000	'	X
				-0000000	'	
+	03	2010	'	-4320,001	'	X
				-0000,021	'	
				3,470	'	X
					'	
				10	'	X
				-0000000	'	
					'	X

Die ersten sechs Dezimalzahlen werden folgendermaßen eingelesen:

Speicherzelle	Inhalt
2004	,109
2005	,012
2006	,034
2007	,056
2008	,078
2009	,09

Der zweite Satz von fünf Datenworten würde folgendermaßen gespeichert:

Speicherzelle	Inhalt
2010	-4320,001
2011	-,021
2012	3,47
2013	0
2014	,01

Das Minusnullwort -0000000' beendet jede Gruppe und bewirkt das Lesen der nächsten. Alle Mantissen werden bei q = 0 mit 24 Bits Länge gehalten. Alle Exponenten werden im gleichen Wort wie die Mantisse bei q = 30 gehalten. Der Extrastopcode hinter dem zweiten Minusnullwort beendet die Eingabe.

2. Datenausgabe

"Dateneingabe und Ausgabe 1" liefert den Inhalt des Gleitkomma-Akkumulators als eine mit Vorzeichen versehene, siebenstellige, dezimale Mantisse und einen mit Vorzeichen versehenen, zweistelligen, dezimalen Exponenten. Nach der Ausgabe erfolgt ein Tabulatorsprung, und ein Sprung zu der auf den P0000 Befehl folgenden Zelle. Der Befehl P0000 hatte das Unterprogramm aufgerufen.

Der Ausdruck hat die Form

.XXXXXXX± EE±

wobei .XXXXXXX± die Mantisse ist
EE± der Exponent zur Basis 10 ist.

Das folgende Beispiel zeigt eine numerische Ausgabe und ihre übliche Darstellung:

Maschinenausdruck	Übliche Darstellung
,5060000- 04	-5060,000
,0937500- 02-	-,000937500
,1000000- 06-	-,0000001000000
,1000000- 10	$,1 \cdot 10^{10}$

SINUS-COSINUS 3, Programm B1-10.2

"Sinus-Cosinus 3" berechnet den Sinus oder den Cosinus eines Argumentes, das im Bogenmaß in Gleitkommaform vorliegt. Das Ergebnis steht bei der Rückkehr ins Hauptprogramm im Gleitkomma-Akkumulator. Die Rückkehr erfolgt zu der auf den Befehl S0000 oder C0000 folgenden Zelle, je nachdem, welches Unterprogramm aufgerufen wurde.

ARCUSTANGENS 2, Programm B1-11.2

"Arcustangens 2" berechnet den Arcustangens eines im Gleitkomma-Akkumulator stehenden Arguments. Das Ergebnis steht im Bogenmaß im Gleitkomma-Akkumulator, wenn Rückkehr zu der auf den A0000 Befehl folgenden Zelle des Hauptprogramms erfolgt.

LOGARITHMUS 2, Programm B3-11.1

"Logarithmus 2" berechnet den Logarithmus zur Basis "e" einer im Gleitkomma-Akkumulator stehenden Zahl. Das Ergebnis steht im Gleitkomma-Akkumulator, wenn Rückkehr zu der auf den N0000 Befehl folgenden Zelle des Hauptprogramms erfolgt.

EXPONENTIALFUNKTION 2, Programm B3-10.1

"Exponentialfunktion 2" berechnet die Größe e^x, wobei x der Inhalt des Gleitkomma-Akkumulators ist. Das Ergebnis steht im Gleitkomma-Akkumulator, wenn Rückkehr zu der auf den H0000 Befehl folgenden Zelle des Hauptprogramms erfolgt.

QUDRATWURZEL 2, Programm B4-10.1

"Quadratwurzel 2" berechnet die Quadratwurzel einer positiven Zahl im Gleitkomma-Akkumulator. Das Ergebnis steht im Gleitkomma-Akkumulator, wenn Rückkehr zu der auf den R0000 Befehl folgenden Zelle des Hauptprogramms erfolgt.

AUFRUFFOLGE

Das "Gleitkomma-Interpretiersystem 1" wird aufgerufen durch eine Aufruffolge des Hauptprogramms. Vor dem Sprung in das System

muß dem System die Anfangsadresse der zu interpretierenden Gleitkommabefehle geliefert werden. Die Gleitkommabefehle müssen unmittelbar auf den Sprungbefehl folgen. Im folgenden wird ein Beispiel dafür geliefert.

Speicherzelle	Befehl	Adresse	
α	R	L_o	Anfangsadresse der Gleitkommabefehle speichern.
α+1	U	L_o	Eingang ins Interpretiersystem.
α+2			]
.			] Gleitkommabefehle
.			]
α+n	XE	0000	Ausgang aus dem Interpretiersystem.
α+n+1			Rückkehr zu Festkommabefehlen.

Die Zellen α und α+1 enthalten die Befehle R und U, die die Adresse (α+2) ins Interpretiersystem speichern und einen Sprung nach L_o, der Anfangsadresse des Interpretiersystems, bewirken. Die Zellen α+2 bis α+n enthalten die Gleitkommabefehle. Ein Ausgang aus dem Interpretiersystem ist erforderlich, wenn wie hier in α+n+1 Festkommabefehle (Maschinenbefehle) folgen.

FESTKOMMA-GLEITKOMMA-UMWANDLUNG und umgekehrt, L3-12.0

Das Programm hat zwei Funktionen. Es wandelt eine Festkommazahl in eine Gleitkommazahl der Form um, wie sie für das "Gleitkomma-Interpretiersystem 1" definiert ist; oder wandelt eine Gleitkommazahl in eine Festkommazahl um. Dieses Programm ist kein grundlegender Bestandteil des Interpretiersystems; d.h., es ist ein Festkommaunterprogramm, dazu bestimmt, Daten umzuwandeln, die für das Interpretiersystem bestimmt sind oder die das Interpretiersystem liefert. Vor der Aufruffolge für dieses Unterprogramm muß das Interpretiersystem verlassen werden. Die umzuwandelnde Zahl muß im Elementarakkumulator stehen und ihr "q" muß von der Aufruffolge geliefert werden.
Eine Gleitkommazahl darf nicht zu groß sein, um bei einem bestimmten "q" gehalten zu werden, wenn sie zurückgewandelt wird. Eine Festkommazahl darf keinen Exponenten größer als 31 haben, wenn sie umgewandelt wird.
Bei der Rückkehr ins Hauptprogramm steht die umgewandelte Zahl im Maschinenakkumulator. Die Zahl steht entweder in Festkommaform beim von der Aufruffolge gelieferten "q", wenn zurückgewandelt wurde, oder in Gleitkommaform entsprechend dem angegebenen "q", wenn umgewandelt wurde.
Die entsprechenden Aufruffolgen werden erläutert. Die Erste ist für das Umwandeln, die Zweite für das Zurückwandeln einer Zahl. Während eines Programmdurchlaufes kann nur eine Operation bezüglich einer Zahl durchgeführt werden.

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Festkommazahl in den Akkumulator.
α	R	L_o+25	Adresse von "q" speichern.
+1	U	L_o	Eingang ins Unterprogramm.
+2	Z	0^8qq	"q" der Zahl angeben.
+3	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt die Festkommazahl in den Akkumulator. (Jeder Befehl, der dies bewirkt, kann benutzt werden.) Der Befehl in α speichert die Adresse (α+2) des Wertes "q" ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse dieses Unterprogramms. Der Befehl in α+2 enthält das "q" der Festkommazahl als dezimale Sektoradresse. Nach α+3 erfolgt Rückkehr aus dem Unterprogramm.

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Gleitkommazahl in den Akkumulator.
α	R	L_o+148	Adresse von "q" speichern.
+1	U	L_o+122	Eingang ins Unterprogramm.
+2	Z	0^8qq	Angabe des "q" der Zahl.
+3	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt die Gleitkommazahl in den Elementarakkumulator. (Jeder Befehl, der dies bewirkt, ist zulässig.)
Der Befehl in α speichert die Adresse (α+2) des Wertes "q" ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung zur angemessenen Zelle, L_o+122 dieses Unterprogramms. Der Befehl in α+2 enthält das "q", das die zurückgewandelte Zahl haben soll, als dezimale Sektoradresse. Nach α+3 erfolgt Rückkehr aus dem Unterprogramm.
In diesem Unterprogramm gibt es zwei besondere Programmstops. Diese werden unten gelistet und erklärt. L_o ist die Anfangsadresse dieses Unterprogramms.

Speicherzelle	Bedeutung und Behebung
L_o + 152	Die Zahl ist zu groß für das angegebene "q" beim Zurückwandeln. Bei "Start" wird der Akkumulator gelöscht, und es erfolgt Rückkehr ins Hauptprogramm.
L_o + 262	Die Zahl verlangt beim Umwandeln einen Exponenten größer als 31. Bei "Start" wird der Akkumulator gelöscht, und es erfolgt Rückkehr ins Hauptprogramm.

Man beachte, daß dieses Unterprogramm unabhängig benutzt wird. Eine Eingabe- oder Ausgabeeinheit ist nicht erforderlich. Der Speicherbedarf wird unten erläutert.

SPEICHERBEDARF

Im eigentlichen Interpretiersystem eingeschlossen ist nur "Quadratwurzel 2". Die anderen Unterprogramme liegen einzeln vor. Sie können nach Erfordernis teilweise oder alle benutzt werden. Die Unterprogramme sind im folgenden mit ihren relativen Anfangsadressen gelistet, wobei L_o die Anfangsadresse des Interpretiersystems ist.

GLEITKOMMA-INTERPRETIERSYSTEM 1	H1-11.0

Unterprogramm	Füllschlüsselwort	Modifikatorschlüsselwort	Speicherbedarf
Interpretiersystem	L_o	L_o	10 Spuren
Dateneingabe u. Ausgabe 1	L_o+1000	L_o+1000	7 Spuren
Sinus-Cosinus 3	L_o+1700	L_o	2 Spuren, 32 Sekt.
Arcustangens 2	L_o+1932	L_o	1 Spur, 32 Sekt.
Logarithmus 2	L_o+2100	L_o	1 Spur
Exponentialfunktion 2	L_o+2200	L_o	2 Spuren
Festkomma-Gleitkomma-Umwandlung u. umgekehrt 1	beliebig		3 Spuren

Spur 63 wird von verschiedenen Teilen des Systems als Zwischenspeicher benutzt, ausgenommen "Festkomma-Gleitkomma-Umwandlung und umgekehrt 1", das keine Zwischenspeicher benötigt. Daher sollte kein Teil des Systems in Spur 63 gespeichert werden.

BEDIENUNG

1. Programmeingabe

Das "Gleitkomma-Interpretiersystem 1" liegt in zwei Streifen vor. Der eine Streifen enthält das Programm in willkürlich verlegbarer hexadezimaler Form, zulässig für Programmeingabe 1 (J1-10.0); der andere Streifen enthält das Programm in willkürlich verlegbarer dezimaler Form in Kodierungsblattformat. Jeder Streifen enthält das Interpretiersystem und alle Unterprogramme in der unter "Speicherbedarf" angegebenen Reihenfolge. Es wird vorausgesetzt, daß Programmeingabe 1 in 0000 beginnt.
Ist PS-32 eingeschaltet, so kann der willkürlich verlegbare hexadezimale Programmstreifen eingelesen werden ohne Zwischenbedienung bis " Festkomma-Gleitkomma-Umwandlung und umgekehrt" erreicht ist. Ist PS-32 ausgeschaltet, so hält der Rechner nach jedem eingelesenen Unterprogramm an. (Zwischen Interpretiersystem und "Dateneingabe und Ausgabe 1" ist kein Stop eingebaut.) In jedem der beiden Fälle ist die Eingabe eines neuen Modifikatorschlüsselwortes und eines Schlüsselwortes für die Auswahl der Eingabeeinheit erforderlich, wenn "Festkomma-Gleitkomma-Umwandlung und umgekehrt" eingelesen werden soll. Dieses Programm darf nie mit dem gleichen Modifikator eingelesen werden, wie irgend ein Teil des Interpretiersystems.
Der willkürlich verlegbare dezimale Streifen hat einen eingebauten Stop hinter dem Interpretiersystem und hinter jedem Unterprogramm.

In jedem Falle muß ein neuer Modifikator und ein neues Auswahlschlüsselwort jür jedes einzulesende Unterprogramm eingegeben werden.
Bei jedem Streifen muß bei einem aus "PS-32 aus" resultierender Stop "Start" gedrückt werden, um zum Zwecke weiterer Eingabe zur Programmeingabe 1 zurückzukehren. Geschieht der Stop bei "PS-32 ein", so hält der Rechner auf einem Eingabebefehl der Programmeingabe 1.

2. Erforderliche Eingabe-/Ausgabeeinheiten.

   Die Lochstreifenschreibmaschine wird für Eingabe und Ausgabe ausgewählt.

3. Programmsprungtasten.

Taste	Funktion
PS-16	EIN - Verhindert den Stop bei dem Befehl Z0000. AUS - Läßt einen Stop bei dem Befehl Z0000 zu.
PST	EIN - Bewirkt einen unbedingten Sprung bei einem 800T Befehl. AUS - Bewirkt einen bedingten Sprung bei einem 800T Befehl.

4. Programmstops

   L_o ist die Anfangsadresse des Systems.

Speicherplatz	Bedeutung	Behebung
L_o+557	Der Exponent ist zu groß für einen H oder C Befehl, oder das Argument einer Quadratwurzel ist negativ. Die Adresse des auszuführenden Befehls steht im Maschinenakkumulator.	Drücke "Start" zum Fortfahren. Der H Befehl, C Befehl oder die Quadratwurzel werden nicht ausgeführt.
L_o+759	Division durch einen unzulässigen Wert.	Nicht fortfahren. Der Datenwert muß verbessert werden.
L_o+1152	Der Eingabewert hat einen zu großen Exponent.	Drücke "Start", um zum nächsten Pseudobefehl überzugehen. Anstelle der zurückgewiesenen Zahl wird eine Null gespeichert.
L_o+2005	Das Argument für einen natürlichen Logarithmus ist ≤ 0.	Drücke "Start", um mit einem Ergebnis von Null fortzufahren.
L_o+2056	Der Exponent ist für einen natürlichen Logarithmus zu groß.	Nicht fortfahren. Der Wert muß verbessert werden.

BEMERKUNGEN

Die Anfangszelle des Interpretiersystems sollte mit dem Sektor 00 einer Spur beginnen, damit die Bezugsstellen auf Spur 63 optimal liegen.
Alle Befehle mit Nulladressen haben spezielle Bedeutungen. Keine dieser Nulladressen bezieht sich auf Zelle 0000.

Das System verwendet 16 solcher Befehle, einschließlich D000y und M000y. Obwohl die Befehle D000y und M000y einen Wert y haben können zwischen 1 und 9, können sich dennoch die Befehle Dttss und Mttss nicht auf eine Zelle zwischen 0000 und 0009 beziehen.

BEISPIEL

Das folgende Beispielprogramm zeigt, wie Gleitkommabefehle für das Interpretiersystem programmiert werden können, und wie die Daten aussehen müssen, um das gewünschte Ergebnis zu erhalten.

Gegeben sei ein Polynom vierten Grades der Form

$$A_0x^4 + A_1x^3 + A_2x^2 + A_3x + A_4 = y$$

Der Wert von y ist zu errechnen und auszuschreiben.

Das Interpretiersystem sei ab 0300 gespeichert. Das Hauptprogramm sei ab 4200 gespeichert. Die Daten seien ab Zelle 4306 gespeichert.

Progr.eingabe Schlüsselwort	Stop	Zelle	Befehl Oper. Adr.	Stop	Inhalt der Adresse	Bemerkungen
;0004200	'					
/0004200	'					
		00	XR,0300	'		Eingang ins Interpretiersystem
		01	XU,0300	'		Daten einlesen
		02	XI,0000	'		Anfangsadresse setzen
		03	E,0017	'	4307	
		04	Y,0006	'	4307	
		05	B,0013	'	Null	
		06	XA,[y+1]	'	a_n	Add. n-ten Koeffiz.
		07	XI,0001	'	a(4306+n)	Adr-reg. um 1 erhö.
		08	Y,0006	'		speichern in 0006
		09	XZ,4312	'		Adressentest
		10	U,0014	'		noch nicht beendet
		11	XP,0000	'		drucke Ergebnis
		12	XE,0000	'		Ausg. aus d. System
		13	XZ,0000	'	Stop (Festkommabefehl)	
		14	XU,0000	'		vertausche
		15	XM,4306	'		multipl. mit X
		16	U,0006	'		zum nächsten Glied
		17	XZ,4307	'		Anfangskoeffizient

Datenmuster-
blatt

Eingabe-schlüssel-wort			Stop	Dezimal-zahlen m. Vorzeichen		Stop		
±	P	Zelle		±			WR	Anm.
+	07	4306	'	+	1090000	'		X
				+	120000	'	☒	A_0
				+	340000	'		A_1
				+	560000	'	☒	A_2
				+	780000	'		A_3
				+	900000	'		A_4
				-	0000000	'		
						'	☒	

BEFEHLSLISTE IN KURZFORM

BEFEHL	ERGEBNIS WENN ADRESSE ≠ 0
B	(ttss) ⟶ (Akku)
A	(Akku) + (ttss) ⟶ (Akku)
S	(Akku) - (ttss) ⟶ (Akku)
D	(Akku) ÷ (ttss) ⟶ (Akku)
P	(ttss) ⟶ (M Reg.)
M	(M Reg.) x (ttss) ⟶ (Akku)
N	(M Reg.) x (ttss) + (Akku) ⟶ (Akku)
H	(Akku) ⟶ (ttss)
C	(Akku) ⟶ (ttss); 0 ⟶ (Akku)
U	Nächster Befehl in ttss
T	Springe, wenn (Akku) negativ
-T	Springe, wenn (Akku) negativ oder wenn PST EIN
E	(Adressteil) von ttss ⟶ (Adr. Reg.)
I	ttss + (Adr. Reg.) ⟶ (Adr. Reg.)
Y	(Adr. Reg.) ⟶ (Adressteil) von ttss
Z	(Adr. Reg.) - ttss = 0? ja, überspringe; nein, überspringe nicht
R	(Zählreg.) + 2 ⟶ (Adressteil) von ttss

( ) = Inhalt von
⟶ = wird zu
Akku = Akkumulator
Adr. Reg. = Adressregister
M Reg. = Multiplikationsregister
Zählreg. = Zählregister

BEFEHL	ERGEBNIS, WENN ADRESSE = 0*
D	(Akku) $\div 2^y \longrightarrow$ (Akku)
M	(Akku) x $2^y \longrightarrow$ (Akku)
U	(Akku) $\longrightarrow$ (M Reg.); (M Reg.) $\longrightarrow$ (Akku)
B	Mache (Akku) positiv
T	Mache (Akku) negativ
Y	Wechsle Vorzeichen von (Akku)
Z	Stop, wenn PS-16 aus ist
E	Ausgang aus dem Interpretiersystem
I	Lies Gleitkommazahlen
P	Gebe Gleitkommazahlen aus
R	$\sqrt{\text{(Akku)}} \longrightarrow$ (Akku)
S	Sinus (Akku) $\longrightarrow$ (Akku)
C	Cosinus (Akku) $\longrightarrow$ (Akku)
A	Arcustangens (Akku) $\longrightarrow$ (Akku)
N	Log_e (Akku) $\longrightarrow$ (Akku)
H	e(Akku) $\longrightarrow$ (Akku)

*) Ausgenommen D000y oder M000y. $0 \leq y \leq 9$.

Eurocomp

PROGRAMM-BESCHREIBUNG LGP21

EUROCOMP GMBH · ELEKTRONISCHE RECHENANLAGEN · MINDEN (WESTF)

GLEITKOMMA-INTERPRETIERSYSTEM 2	H1-11.1

1. EINFÜHRUNG	H1-11.1

Das "Gleitkomma-Interpretiersystem 2" gestattet die Programmierung von LGP-21 Problemen in Gleitkomma-Arithmetik. Das System läßt als Daten 9-stellige ganze Dezimalzahlen zu mit zweiziffrigen dezimalen Skalierungsfaktoren; es führt alle notwendigen Skalierungen während der Rechnung aus, und gibt die Ergebnisse als 7-stellige dezimale Mantissen mit Vorzeichen aus, gefolgt von einem 2-stelligen dezimalen Exponenten.
Da das System Werte eines außergewöhnlich breiten Zahlenbereichs selbstständig verarbeitet, ist keine genaue Kenntnis des zahlenmäßigen Bereichs eines Problems notwendig. Dieses Fassungsvermögen des Systems garantiert einen hohen Grad von Genauigkeit bei den Berechnungen und bei der Ausgabe. Außerdem wurde die Zahl der Standardbefehle des Rechners durch dieses Programm auf 33 erweitert, um die Benutzung so einfach wie möglich zu gestalten.
Das "Gleitkomma-Interpretiersystem 2" ist verfügbar als ein einzelner Programmstreifen in willkürlich verlegbarer hexadezimaler Fassung, zulässig für Programmeingabe 1 (J1-10.0). Er enthält verschiedene Unterprogramme, wie Dateneingabe, Datenausgabe und eine Reihe von Funktionsunterprogrammen, die später im einzelnen beschrieben sind. Nur die für ein besonderes Problem erforderlichen Unterprogramme brauchen eingelesen zu werden, zusammen mit dem "Gleitkomma-Interpretiersystem 2"; dennoch muß dabei der Speicherbezug für das gesamte System gewahrt bleiben.
Der Programmierer wird sich überzeugen, daß die Benutzung dieses Systems eine große Kraft- und Zeitersparnis bedeutet, wenn es auf Probleme angewandt wird, wo die Zahlen größenmäßig weit streuen, so daß bei einer Festkomma-Arithmetik ein großer Programmierungsaufwand getrieben werden müßte.

Grundzüge - Das "Gleitkomma-Interpretiersystem 2" interpretiert die Elementarbefehlsstruktur des LGP-21 so, daß er wie ein "festverdrahteter" Gleitkommarechner programmiert werden kann. Das wird erreicht durch:

1. Internes Multiplikationsregister, Adressregister und Gleitkomma-Akkumulator.
2. Kumulative Multiplikation, Vorzeichenwechsel und funktionserzeugende Befehle.
3. Einen breiten Zahlenbereich. Die Eingabedaten bestehen aus 9-stelligen Dezimalzahlen mit Vorzeichen, gefolgt von einem Exponenten zwischen +99 und -99. Die Ausgabedaten bestehen aus 7-stelligen Dezimalzahlen mit Vorzeichen, gefolgt von einem Zehnerexponenten zwischen +99 und -99.
4. Einen erweiterten Rahmen für die Eingabe und Druckbefehle.

Das Programm braucht 28 Spuren einschließlich Dateneingabe (7 Spuren), Datenausgabe (4 Spuren), Gleitkommafunktionen (6 Spuren). Es bleiben also 36 Spuren (2304 Worte) für das Hauptprogramm und Datenspeicher übrig.

2. DIE REGISTER

Die Gleitkommaform benötigt zwei aufeinanderfolgende Speicherzellen zur Darstellung eines Datenwortes. Im Interpretiersystem sind daher zwei spezielle Register vorgesehen zum Transport der Daten, zur Verarbeitung der Daten und zur Ausgabe, wobei die Gleitkommaform erhalten bleibt. Diese Register sind der Gleitkomma-Akkumulator und das Multiplikationsregister, die jeweils zwei Zellen belegen. Adressenmodifikation kann ohne Verlassen des Interpretiersystems vorgenommen werden, indem man das simulierte Adressregister benutzt, das eine Zelle belegt. Ein Zählregister von einer Zelle regelt den Ablauf der Operationen des Interpretiersystems.
Der Inhalt aller Register ist Null, wenn das System gerade erst eingelesen wurde und noch nicht damit gearbeitet wurde.

Eine ausführliche Erläuterung der speziellen Register folgt. Die in diesem Zusammenhang erwähnten Befehle werden in Abschnitt 4 unter "Pseudobefehle" genau erklärt.

1. Der Gleitkomma-Akkumulator.

   Der Gleitkomma-Akkumulator hält Zwischenergebnisse und Ausgabewerte. Da die Ausgabedaten aus einer 7-stelligen dezimalen Mantisse mit Vorzeichen und einem 2-stelligen dezimalen Exponenten bestehen, werden sie intern folgendermaßen behandelt: Die Mantisse wird im Gleitkomma-Akkumulator mit 30 Bits bei $q = 0$ gehalten, wobei die erste bewertete Stelle den Wert 2^{-1} hat; d.h., die Mantisse ist immer normalisiert. Der Exponent steht bei $q = 29$. Beide Teile haben jeweilig ihr algebraisches Vorzeichen.
   Der Inhalt des Akkumulators wird nur verändert durch einen arithmetischen Befehl oder durch einen der folgenden Befehle: Lösche, Vertausche, Setze Vorzeichen; oder durch einen Funktionsbefehl.

2. Multiplikationsregister.

   Das Multiplikationsregister enthält den Multiplikanden bei Multiplikation und bei kumulativer Multiplikation. Die Mantisse ist bei $q = 0$ und der Exponent bei $q = 29$ gespeichert. Beide Teile enthalten jeweilig ihr algebraisches Vorzeichen. Der Inhalt dieses Registers wird nur durch die Befehle "Setzen" und "Vertauschen" verändert.

3. Adressenregister.

   Das Adressenregister enthält eine Adresse in Spur-/Sektorschreibweise bei $q = 29$. Dieses Register wird benutzt, um den Operandenadressteil von Speicherworten mittels des Befehls "Adresse speichern" zu modifizieren, und zur Ablaufsteuerung von Hauptprogrammoperationen beim Null-Test. Der Inhalt des Registers wird nur durch einen Befehl "Adresse setzen" oder "Adresse erhöhen" geändert.

4. Zählregister

   Das Zählregister enthält die Adresse des ausgeführten Pseudobefehls. Ebenso gibt es die Adresse des nächsten auszuführenden Befehls an. Der Inhalt plus 2 dieses Registers wird zum Operandenadressteil des Befehls, auf den sich ein Rückkehrbefehl bezieht. Der Inhalt dieses Registers wird anfangs durch die Aufruffolge gesetzt und bei jeder Befehlsausführung um 1 erhöht. Die fortschreitende Erhöhung dieses Registers kann nur durch einen Sprungbefehl geändert werden.

3. INTERNES ZAHLENFORMAT

Jede Gleitkommazahl benötigt zu ihrer vollständigen Darstellung zwei aufeinanderfolgende Speicherzellen. Die erste Zelle nimmt die Mantisse der Zahl in dualer Darstellung bei q = 0 auf. Jede Mantisse besteht aus algebraischem Vorzeichen und 30 bewerteten Dualstellen; das meistbedeutende Bit hat den Wert 2^{-1}.
Die zweite Zelle nimmt den Zweierexponenten auf, der mit der Mantisse multipliziert werden muß, um die wirklich dargestellte Zahl zu erhalten. Dieser bei q = 29 gehaltene Exponent ist eine ganze Dualzahl mit algebraischem Vorzeichen.
Zusammen betrachtet, stellen die Faktoren einer Gleitkommazahl diese Zahl dar als $Z=M\cdot 2^E$, wobei Z = Zahl, M = Mantisse, E = Exponent ist.

Beispiel: 3,75 würde im Gleitkomma im Speicher dargestellt als $,9375\cdot 2^2$ bzw. als das duale Äquivalent:

Mantisse: 01111000000000000000000000000000
$\hat{q} = 0$

Exponent: 00000000000000000000000000001000
$\hat{q} = 29$

wobei: M = 0,9375
E = 2
$Z = 0{,}9375\cdot 2^2$ ist.

Eine negative Zahl, -3,75, würde dargestellt als:

Mantisse: 10001000000000000000000000000000
$\hat{q} = 0$

Exponent: 00000000000000000000000000001000
$\hat{q} = 29$

wobei: M = -0,9375
E = 2
$Z = -0{,}9375\cdot 2^2$ ist.

Die fortlaufende Speicherung von Mantissen und Exponenten wird im folgenden gezeigt.

Speicherplatz	Inhalt
4304	Mantisse 1
4305	Exponent der Mantisse 1
4306	Mantisse 2
4307	Exponent der Mantisse 2

Man beachte:

1. Bei der Eingabe ist der dezimale Exponent beschränkt auf: $-99 \leq E \leq +99$.

2. Bei der Ausgabe ist der dezimale Exponent beschränkt auf: $-99 \leq E \leq +99$.

3. Bei Speicherung des Gleitkomma-Akkumulatorinhalts hält der Rechner nicht an, wenn für den dualen Exponenten gilt: $-333 \leq E \leq +333$.

4. PSEUDOBEFEHLE

Es folgt die Liste der 33 LGP-21 Befehle für das 'Gleitkomma-Interpretiersystem 2', und die Erklärung ihrer Funktion. Die Bezeichnung "Akkumulator" bezieht sich auf die zwei Speicherzellen des Gleitkomma-Akkumulators.

ARITHMETISCHE BEFEHLE - XXXX ist die Adresse der Mantisse einer Gleitkommazahl.	
BXXXX Bringe	Der Inhalt von Zelle XXXX wird zum Inhalt des Akkumulators.
AXXXX Addiere	Der Inhalt des Akkumulators plus der Inhalt der Zelle XXXX wird zum Inhalt des Akkumulators.
SXXXX Subtrahiere	Der Inhalt des Akkumulators minus der Inhalt der Zelle XXXX wird zum Inhalt des Akkumulators.
DXXXX Dividiere	Der Inhalt des Akkumulators dividiert durch den Inhalt der Zelle XXXX wird zum Inhalt des Akkumulators.
PXXXX Setze	Der Inhalt der Zelle XXXX wird zum Inhalt des M Registers.
MXXXX Multipliziere	Der Inhalt des M Registers multipliziert mit dem Inhalt der Zelle XXXX wird zum Inhalt des Akkumulators.
NXXXX Multipliziere kumulativ	Das Produkt aus M Registerinhalt und Inhalt der Zelle XXXX plus Akkumulatorinhalt wird zum Akkumulatorinhalt.
XD000y Dividiere durch 2^y	Der Inhalt des Akkumulators dividiert durch 2^y wird zum Inhalt des Akkumulators. Der Inhalt des Akkumulators bleibt in Gleitkommaform ($0 \leq y \leq 9$).
XM000y Multipliere mit 2^y	Der Inhalt des Akkumulators multipliziert mit 2^y wird zum Inhalt des Akkumulators. Der Inhalt des Akkumulators bleibt in Gleitkommaform ($0 \leq y \leq 9$).
HXXXX Halte	Speichere den Inhalt des Akkumulators in Zelle XXXX.
CXXXX Lösche	Speichere den Inhalt des Akkumulators in Zelle XXXX und setze den Akkumulator auf Null.

LOGISCHE ODER SPRUNGBEFEHLE - XXXX ist die Adresse eines Befehls	
UXXXX Unbedingter Sprung	Der nächste zu interpretierende Befehl ist der Zelle XXXX zu entnehmen. Dieser Befehl darf nicht zum Verlassen des Interpretiersystems benutzt werden.
TXXXX Teste Minus	Der nächste zu interpretierende Befehl steht in Zelle XXXX, wenn der Gleitkomma-Akkumulatorinhalt negativ ist. Andernfalls wird der nächstfolgende Befehl interpretiert.
800TXXXX Bedingter Sprung	Der nächste zu interpretierende Befehl steht in Zelle XXXX, wenn entweder der Akkumulatorinhalt negativ ist, oder wenn PST eingeschaltet ist. Andernfalls wird der nächstfolgende Befehl interpretiert.
EXXXX Setze Adresse	Der Adressteil der Zelle XXXX wird zum Inhalt des Adressregisters.
IXXXX Erhöhe	Der Inhalt des Adressregisters wird um XXXX erhöht. Dieser Befehl kann auch zur Erniedrigung des Adressregisters benutzt werden, wenn man das Komplement des Adressteils des IXXXX Befehls verwendet.
YXXXX Adresse speichern	Der Adressteil des Adressregisters wird zum Adressteil der Zelle XXXX.
ZXXXX Null-Test	Die Adresse (XXXX) des Befehls ZXXXX wird vom Inhalt des Adressregisters subtrahiert. Wenn das Ergebnis Null ist, wird der nächste Befehl übersprungen und der zweite folgende Befehl interpretiert. Ist das Ergebnis ungleich Null, so wird der nächste Befehl interpretiert.
RXXXX Rückkehr	Die Adresse, die den RXXXX Befehl enthält, wird um zwei erhöht und in den Adressteil der Zelle XXXX gesetzt.

HILFSBEFEHLE - der Adressteil der folgenden Befehle muß Null sein.	
U0000 Vertausche	Der M Registerinhalt und der Akkumulatorinhalt werden vertauscht.
B0000 Pluszeichen setzen	Der Inhalt des Akkumulators wird positiv gemacht, wenn er nicht schon positiv ist.
T0000 Minuszeichen setzen	Der Inhalt des Akkumulators wird negativ gemacht, wenn er nicht schon negativ ist.
Y0000 Vorzeichenwechsel	Der Inhalt des Akkumulators, multipliziert mit -1, wird zum Inhalt des Akkumulators.
Z0000 Stop	Der Rechner hält an, wenn PS-16 eingeschaltet ist. Nach "Start" wird die Rechnung mit dem nächsten folgenden Befehl fortgesetzt.
E0000 Ausgang	Ausgang aus dem Gleitkomma-Interpretiersystem. Es erfolgt Rückkehr zur ersten folgenden Zelle.
EINGABE-/AUSGABEBEFEHLE - der Adressteil der folgenden Befehle muß Null sein.	
I0000 Eingabe	Es erfolgt ein Sprung zu einem Einleseunterprogramm für Gleitkommadaten, das dezimal abgelochte Daten vom Streifen liest, in Gleitkommaform umwandelt und speichert. Der nächstfolgende Befehl wird interpretiert, wenn der Sonderstopcode vom Streifen gelesen wurde.
P0000 Ausgabe	Ausdruck des Akkumulatorinhalts. Der Akkumulatorinhalt wird nicht verändert.
FUNKTIONSBEFEHLE - der Adressteil der folgenden Befehle muß Null sein.	
R0000 Quadratwurzel	Die Quadratwurzel aus dem Akkumulatorinhalt wird zum Akkumulatorinhalt. Ein negatives Argument bewirkt einen Fehlerstop.
S0000 Sinus	Der Sinus des Akkumulatorinhalts wird zum Akkumulatorinhalt. Das Argument muß im Bogenmaß stehen.
C0000 Cosinus	Der Cosinus des Akkumulatorinhalts wird zum Akkumulatorinhalt. Das Argument muß im Bogenmaß stehen.
A0000 Arcustangens	Der Arcustangens des Akkumulatorinhalts wird zum Akkumulatorinhalt. Das Ergebnis steht im Bogenmaß.
N0000 Natürlicher Logarithmus	Der natürliche Logarithmus des Akkumulatorinhalts wird zum Akkumulatorinhalt.
H0000 Exponentialfunktion	Die Größe E^X wird zum Akkumulatorinhalt, wobei X ursprünglich im Akkumulator stand.

Anmerkung: Die arithmetischen Befehle, Sprungbefehle, Adressenmodifikationsbefehle, die Hilfsbefehle und das Quadratwurzel-Unterprogramm sind Hauptbestandteile des Interpretiersystems. Die Eingabe-/Ausgabe- und Funktions-Unterprogramme dagegen wurden extra programmiert und brauchen nur gespeichert zu werden, wenn das Hauptprogramm sie benötigt.

5. DATENEINGABE

Der Pseudoeingabebefehl verlangt Eingabe eines dezimalen Streifens der folgenden Form:

1. Einer Anfangsadresse.
2. Der Daten als mit Vorzeichen versehenen Dezimalzahlen mit ihren Exponenten.
3. Einem Schlüsselwort für jedes Gruppenende von Zahlen.
4. Einem Extrastopcode am Ende das Datenstreifens.

Die Anfangsadresse - Als erster Ausdruck in jeder Datengruppe muß die Anfangsspeicheradresse der Gruppe in dezimaler Spur-/Sektorschreibweise stehen. Dies muß eine absolute Adresse sein, gefolgt von einem Stopcode (‘). Beispiel: 4200‘ würde das Unterprogramm veranlassen, die Daten ab Zelle 4200 zu speichern.

Die Daten - Die Anfangsadresse ist gefolgt von einer oder mehreren, mit Vorzeichen versehenen ganzen Dezimalzahlen mit ihren Exponenten. Diese Daten werden in Gleitkommaform umgewandelt und in aufeinanderfolgenden Zellen gespeichert, beginnend bei der Anfangsadresse. Jede Zahl wird in zwei Worten angegeben; hinter jeder der beiden muß ein Stopcode stehen. Bei negativen Zahlen muß das Vorzeichen vor der ersten Ziffer des ersten Wortes stehen; bei positiven Zahlen darf an dieser Stelle eine Leertaste, aber kein Pluszeichen stehen.
Das erste Wort besteht aus dem Vorzeichen und den ersten sieben Ziffern der ganzen Zahl; das zweite Wort besteht aus den beiden letzten Ziffern der ganzen Zahl, gefolgt von dem Vorzeichen und den zwei Ziffern des Exponenten.
Formal sieht das so aus:

$$\underbrace{\pm d_1 d_2 d_3 d_4 d_5 d_6 d_7'}_{\text{1. Wort}} \quad \underbrace{d_8 d_9 \pm e_1 e_2'}_{\text{2. Wort}}$$

wobei $\pm d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9$ die 9-ziffrige Dezimalzahl mit Vorzeichen darstellt und $\pm e_1 e_2$ den 2-ziffrigen dezimalen Exponenten.

Die Ausdrucksweise der Daten beruht auf der Beziehung:

Dargestellte Zahl = Ganze Zahl · $10^{e_1e_2}$

mit $-99 \leq e_1e_2 \leq +99$.

Beispiel: Dem Bruch ,0000000521374209 entspricht die Schreibweise 521374209 · 10^{-16}, und er wird für die Eingabe dargestellt als die zwei Worte

5213742'09-16'

Der ganzen Zahl 5522829640000000 entspricht die Schreibweise 552282964 · 10^{+7}, und sie wird zur Eingabe dargestellt als die beiden Worte

5522829'64+07'

Der Schlußstopcode - Es gibt zwei Arten von Schlußstopcodes, die verschiedene Funktionen haben entsprechend ihrer Lage auf dem Datenblatt (Lochstreifen).
Die erste Art zeigt das Gruppenende der Daten an und bewirkt das Einlesen der nächsten Datengruppe; die zweite Art zeigt das Ende der Eingabe an und bewirkt Ausgang aus dem Leseprogramm; in diesem letzten Falle erfolgt ein Wagenrücklauf und Rückkehr ins Hauptprogramm.
Das folgende Beispiel zeigt ein Musterdatenblatt für das "Gleitkomma-Interpretiersystem 2", in dem zwei Sätze von Daten zu speichern sind. Der Stopcode am Ende der ersten Gruppe bewirkt Einlesen der Gruppe 2. Am Ende der zweiten Gruppe ist die Möglichkeit gezeigt, eine weitere Gruppe einzulesen oder das Leseprogramm zu verlassen, was von der Benutzung der Schlußstopcodes abhängt.

EINGABESCHLÜSSELWORT (Anfangsadresse)	STOP	DEZIMALZAHLEN MIT VORZEICHEN Teil 1 $d_1d_2\ldots d_7$	'	Teil 2 d_8d_9	e_1e_2	STOP	
5500	'	- 3300000	'	00	- 04	'	Gruppe 1
		3141592	'	65	- 08	'	
		- 0000592	'	34	- 11	'	
benutze nur Stopcode			'		hier keinen Stopcode benutzen	'	
2134	'	0	'			'	Gruppe 2
		30	'	50	-	'	
		750	'	00	+	'	
		- 0100000	'	00	+	'	

Minuszeichen für negative Zahlen, für positive Zahlen nur Leertaste

nur Stopcode benutzen

Stopcode hier nur benutzen, wenn Ausgang erwünscht ist

Die drei Dezimalzahlen der ersten Gruppe werden folgendermaßen gespeichert:

Speicherplatz	Inhalt (im Speicher in dualer Form)
5500	-33000,
5501	(Exponent zu -33000)
5502	3,14159265
5503	(Exponent zu 3,14159265)
5504	-,00000059234
5505	(Exponent zu -,00000059234)

Alle Exponenten werden bei q = 29 gehalten. Der Stopcode hinter der letzten Zahl der Gruppe (der in der Spalte hinter dem "Teil 1" steht) bewirkt, daß die Anfangsadresse der zweiten Gruppe (2134) gelesen wird.

Der zweite Datensatz wird folgendermaßen gespeichert:

Speicherplatz	Inhalt (gespeichert in dualer Form)
2134	Null
2135	Null
2136	30,50
2137	(Exponent von 30,50)
2138	750000
2139	(Exponent von 750000)
2140	-10000000000000
2141	(Exponent von -10000000000000)

Der zweite Datensatz wird durch einen Stopcode in der gleichen Spalte, wie bei Gruppe 1, beendet. Ist Ausgang aus dem Leseprogramm erwünscht, so muß ein zweiter Stopcode eingegeben werden, der in der Spalte hinter "Teil 2" steht.
Man beachte bitte, daß eine Zahl mit dem Wert Null eingegeben werden muß als 0' und nicht als ('), damit das Programm die Zahl vom Stopcode für das Gruppenende unterscheiden kann.

6. DATENAUSGABE

Der Ausdruck besteht aus Vorzeichen und dezimalem Punkt, gefolgt von 7 Dezimalziffern (xxxxxxx) für die Mantisse. (Bei einer negativen Zahl wird ein Minuszeichen geschrieben; anstelle eines Pluszeichens eine Leertaste.) Der Mantisse folgt eine Leertaste, das Vorzeichen des Exponenten (+ oder -) und zwei Dezimalziffern für den Exponenten. Dann erfolgt ein Tabulatorsprung und Rückkehr ins Hauptprogramm. Bei jedem Unterprogramm-Durchlauf wird eine Zahl ausgegeben. Formal sieht die Ausgabe so aus:

-.xxxxxxx±ee
↑——————^——Leertaste für +

Das entspricht der mathematischen Darstellung:

$$\pm.xxxxxxx\ (10^{\pm ee})$$

Beispiele:

Ausdruck	Zahlenwert
-.5060000+04	-5060.0000
-.9375000-02	-.00937500000
-.1000000-06	-.00000010000000
.1000000+30	$.1\cdot(10^{30})$

Für den ausgedruckten dezimalen Exponenten gilt: $-99 \leq ee \leq +99$.
Für die ausgedruckte Mantisse gilt:

$,10000000 \leq ,xxxxxxx \leq ,9999999$ oder Null.

Für den dualen Exponenten gilt:

$-333 \leq \text{Exponent} \leq +333$.

Ein dualer Exponent kleiner als -333 bewirkt den Ausdruck einer Null, und ein dualer Exponent größer als +333 bewirkt einen Bereichsfehler.

7. FUNKTIONSUNTERPROGRAMME

SINUS-COSINUS

Dieses Unterprogramm berechnet den Sinus oder Cosinus eines gegebenen Wertes der unabhängigen Variablen ("Argument"), ausgedrückt im Bogenmaß. Zur Approximation des Funktionswertes wird ein Polynom 9-ten Grades benutzt. Bevor die Rechnung durch den Pseudobefehl "S0000" (oder "C0000") eingeleitet wird, muß der Winkel im Gleitkomma-Akkumulator stehen. Der Sinus des Winkels (oder der Cosinus des Winkels) steht nach der Ausführung im Gleitkomma-Akkumulator. Die folgende Tabelle zeigt die Methode zur Berechnung der Funktionswerte in Abhängigkeit vom Exponenten des Arguments.

Exponent (dual)	Sinus X	Cosinus X
26 bis +127	Bereichsfehler	Bereichsfehler
-8 bis +25	Berechnung durch ein Polynom 9. Grades	Berechnung durch ein Polynom 9. Grades
-20 bis -9	Sinus X=X	Berechnung durch ein Polynom 9. Grades
-127 bis -21	Sinus X=0	Cosinus x=1,0

ARCUSTANGENS

Dieses Unterprogramm berechnet den Winkel, der einem gegebenen Tangens-Argument entspricht. Das Argument muß im Gleitkomma-Akkumulator stehen, wenn der Befehl "A0000" die Rechnung einleitet. Das Ergebnis steht im Bogenmaß im Gleitkomma-Akkumulator.

LOGARITHMUS UND EXPONENTIALFUNKTION

Die unabhängige Variable X muß im Gleitkomma-Akkumulator stehen, wenn das entsprechende Unterprogramm aufgerufen wird; das Ergebnis $\log_e x$ oder e^x erscheint wieder im Gleitkomma-Akkumulator. Der Logarithmus und die Exponentialfunktion werden aufgerufen durch die Pseudobefehle N0000 und H0000 jeweilig.

QUADRATWURZEL

Dieses Unterprogramm, das im Interpretiersystem eingeschlossen ist, berechnet die Quadratwurzel aus der im Akkumulator stehenden Zahl und speichert sie im Akkumulator.

8. BEDIENUNG

Eingabe - Die Eingabe besteht aus Gleitkommazahlen vom Streifen oder aus dem Speicher, oder aus in den Pseudoregistern stehenden Zahlen, die aus vorhergehenden Operationen stammen.

GLEITKOMMA-INTERPRETIERSYSTEM 2 | H1-11.1

Aufruffolge

Speicherplatz	Befehl	Adresse	
XXXX	R	L_o	
XXXX +1	U	L_o	
XXXX +2	.	.	} Gleitkommabefehle
XXXX +3	.	.	
..	.	.	
..	.	.	
..	.	.	
XXXX +n	E	0000	Ausgang
XXXX +n+1	u.s.w.		Festkommabefehle

Speicherbedarf der Unterprogramme - Wenn die Anfangsadresse von H1-11.1 definiert ist als "L_o", so müssen die Unterprogramme dieses Streifens, entsprechend der folgenden Tabelle, eingelesen werden (unter Benutzung des entsprechenden Eingabe- und Modifikatorschlüsselwortes).

Programm	Eingabe-schlüsselwort	Modifikator-schlüsselwort	Speicher-bedarf
Interpretiersystem	L_o	L_o	11 Spuren
Dateneingabe	L_o+1100	L_o	6 Spuren
Datenausgabe	L_o+1700	L_o	4 Spuren
Sinus - Cosinus	L_o+2100	L_o	2 Spuren
Arcustangens	L_o+2300	L_o	83 Sektoren
Logarithmus	L_o+2419	L_o	45 Sektoren
Exponentialfunktion	L_o+2500	L_o	2 Spuren

Benutzte Zwischenspeicher - Das Programm braucht als Zwischenspeicher die Zellen 6300 bis 6363.

Programmstops

Speicherplatz	Bedeutung	Behebung
L_o + 0947	Negatives Argument für Quadratwurzel	Drücke "Start". Es folgt Ausgang aus dem Unterprogramm mit dem Argument im Akkumulator.
L_o + 0312	Exponent ≥333 für Halte oder Lösche	Drücke "Start". Halte oder Lösche Befehle werden übergangen.
L_o + 0710	Division durch Null oder nicht gefloatete Zahl	Nicht fortfahren.

Speicherplatz	Bedeutung	Behebung
L_o + 1808	Exponent ≥ 99 (Datenausgabe)	Bei "Start" erfolgt ein "Tab" (ohne Drucken) und Ausgang zum Programm H1-11.1
L_o + 2100	(Argument für Sinus-Cosinus ≥ 2^{26}	Drücke "Start". Der S0000 oder C0000-Befehl wird übergangen.
L_o + 2426	(Argument für $\text{Log}_e X$) ≤ 0	Drücke "Start". Das Unterprogramm liefert als Ergebnis Null, und es erfolgt Rückkehr.
L_o + 2441	$\text{Log}_e X > .789629525 \times 10^{14}$ [e^{32}]	Drücke "Start". Rechnung wird mit Null im Gleitkomma-Akkumulator fortgesetzt.
L_o + 2629	(Argument für e^x) ≥ 128	Drücke "Start". H0000-Befehl wird übergangen.

Programmsprungtasten

Taste	Funktion
PS-16	AUS - Beim Z0000-Befehl erfolgt kein Stop. EIN - Beim Z0000-Befehl erfolgt ein Stop.

Genauigkeit

Sinus-Cosinus - Maximaler Fehler 5×10^{-8}.

Datenausgabe - Der maximale Fehler ist eine Einheit auf der letzten Mantissenstelle, wenn $|EXP_{10}| \leq 30$; ± 5 Einheiten auf der letzten Mantissenstelle, wenn $|EXP_{10}| \leq 99$.

Ausgang - Ausgang aus dem System erfolgt zur ersten Zelle hinter dem Befehl E0000.

ANMERKUNGEN

1. Ein Programm kann das System H1-11.1 verlassen und zu ihm zurückkehren, ohne daß der Inhalt der Pseudoregister verloren geht.
2. Die Anfangsadresse des Systems sollte der Sektor 00 einer Spur sein. Andernfalls sind viele der Adressen, die sich auf Spur 63 beziehen, nicht optimal.
3. Es ist zu empfehlen, das gesamte Programm einzulesen, und in Teilen mittels J4-10.3 oder mittels J4-10.2 auszustanzen. Dann brauchen nur die benötigten Teile wieder eingelesen zu werden und zwar entweder mittels J1-10.0 oder mittels J5-10.0; außerdem kann jede Prüfsumme verifiziert werden.
4. Befehle mit Nulladressen beziehen sich nicht auf die Zelle 0000. Vielmehr werden sie benutzt, um spezielle Pseudobefehle anzuzeigen. Das Programm benutzt 14 solche Befehle mit Nulladressen, und außerdem die zwei speziellen Schiftbefehle D000y und M000y. Die Schiftbefehle haben den Adressteil 0001 bis 0009; daher darf der Divisionsbefehl und der normale Multiplikationsbefehl diese Adressen nicht benutzen.

BEFEHL	ERGEBNIS, WENN ADRESSE XXXX ≠ 0
Z	(Adr.reg.) - (XXXX) = 0? ja→überspringe nein→überspringe nicht
B	(XXXX)⟶(Akku)
Y	(Adr.reg.)⟶Adressteil von (XXXX)
R	Speicherzelle von R plus 2⟶Adressteil von (XXXX)
I	(Adr.reg.) + (XXXX)⟶(Akku)
D	(Akku) ÷ (XXXX)⟶(Akku)
N	(M Reg.) · (XXXX) + (Akku)⟶(Akku)
M	(M Reg.) · (XXXX)⟶Akku
P	(XXXX)⟶(M Reg.)
E	Adressteil von (XXXX)⟶Adressteil des (Akku)
U	Nächsten Befehl aus XXXX nehmen
T	Sprung, wenn (Akku) negativ
-T	Sprung, wenn (Akku) negativ oder wenn PST eingeschaltet ist
H	(Akku)⟶(XXXX)
C	(Akku)⟶(XXXX); 0⟶(Akku)
A	(Akku) + (XXXX)⟶(Akku)
S	(Akku) - (XXXX)⟶(Akku)

( ) = Inhalt von ⟶ = wird zu

BEFEHL	ERGEBNIS, WENN ADRESSE = 0 *
Z	Stop (PS-16); weiter rechnen bei "Start"
B	Mache (Akku) positiv
Y	Bilde Komplement des (Akku)
R	$\sqrt{(\text{Akku})}$⟶(Akku)
I	Eingabe von Gleitkomma-Daten
D	(Akku) ÷ 2^y ⟶ (Akku)
N	Log_e (Akku)⟶(Akku)
M	(Akku) · 2^y ⟶(Akku)
P	Drucke (Akku)
E	Ausgang aus dem Programm

BEFEHL	ERGEBNIS WENN ADRESSE = 0 *
U	(Akku) → (M Reg.); (M Reg.) → (Akku)
T	Mache (Akku) negativ
-T	Springe, wenn (Akku) negativ oder wenn PST ein ist
H	$e^{(Akku)} \rightarrow$ (Akku)
C	Cosinus (Akku) → (Akku)
A	Arcustangens (Akku) → (Akku)
S	Sinus (Akku) → (Akku)

*) Mit Ausnahme der Befehle "M000y" und "D000y", mit $0 \leq y \leq 9$.

Progr.eingabe Schlüsselwort	Stop	Zelle	Befehl Oper.	Adr.	Stop	Inhalt der Adresse	Bemerkungen
:0004200	'						
/0004200	'						
		00	xR	1400	'		Eingang ins Inter-
		1	xU	1400	'		pretiersystem
		2	xI	0000	'		Koeffizienten einl.
		3	E	0018	'	4306	Anfangsadresse
		4	Y	0006	'	4306	setzen
		5	B	0013	'	Null	
		6	A	[ ]	'	a_n	
		7	xI	0002	'	a(4306+2n)	Adressenregister
		8	Y	0006	'	a(4306+2n)	um 2 erhöhen
		9	xZ	4316	'		Teste auf Ende
		10	U	0015	'		Nicht beendet
		11	xP	0000	'		Drucke Erg.; Ende
		12	xE	0000	'		Ausgang
		13	xZ	0000	'	Stop (Fes	tkomma Befehl)
		14	U	0000	'		Rückk. z. Anfangsa.
		15	xU	0000	'		(Akku) ⇄ (M Reg.)
		16	xM	4304	'		Multipl. mit x
		17	U	0006	'	Rückk. f.	nächstes Glied
		18	xZ	4306	'		Anfangskoeffizient
000XXXX		19			'		

Schlüsselwort	STOP	DEZIMALZAHLEN Teil 1 $\pm d_1 d_2 .. d_7$	'	Teil 2 $d_8 d_9 \pm e_1 e_2$	STOP
4304	'	1090000	'	00 + 09	'
		0120000	'	00 + 09	'
		0340000	'	00 + 09	'
		0560000	'	00 + 09	'
		0780000	'	00 + 09	'
		0900000	'	00 + 09	'
			'		'

UMWANDLUNG DEZIMAL-HEXADEZIMAL	H4-10.0

VERWENDUNGSZWECK

Das Programm H4-10.0 wandelt willkürlich verlegbare, dezimalkodierte Programmstreifen, die zulässig sind für Programmeingabe 1 (J1-10.0), um in willkürlich verlegbare dezimale Streifen, zulässig für Programmeingabe 1. Während der Ausgabe wird ein Protokoll geführt von (1) allen Befehlen des umzuwandelnden Programmes, das anzeigt, welche Befehle zu modifizieren sind und welche nicht, und (2) von den Speicherbereichen, auf die das Programm gelegt wird. Alle im Programm erforderlichen Korrekturen können mittels Programmeingabe 1 vor der Ausgabe gemacht werden. Der daraus resultierende willkürlich verlegbare hexadezimale Streifen kann nach der Ausgabe bezüglich seines Inhaltes und seiner Prüfsummen geprüft werden.

SPEICHERBEDARF

Das Programm H4-10.0 wird unmittelbar im Anschluß an Programmeingabe 1 gespeichert. Für Programm nebst Tabellen sind 11 Spuren und 35 Sektoren erforderlich. Zwischenspeicher werden nicht benutzt.

EINGABE

Das Programm H4-10.0 muß im Speicher sein, bevor das umzuwandelnde Programm eingegeben wird. H4-10.0 verändert die Programmeingabe 1 so, daß sie während der Eingabe des umzuwandelnden Programmes spezielle Tabellen aufstellt. Diese Tabellen regeln die anschließende Ausgabeoperation. Das Programm stellt die ursprüngliche Form von Programmeingabe 1 wieder her.
Das umzuwandelnde dezimale Programm sollte mindestens 15 Spuren oberhalb der Anfangsadresse von Programmeingabe 1 gespeichert werden, und zwar so, daß man es ausführen könnte; d.h., es sollte mit richtigem Füllschlüsselwort(;) und Modifikator(/) eingelesen werden. Alle Programmeingabeschlüsselwörter arbeiten normal.
Wenn das umzuwandelnde Programm eingelesen wird, speichert die Programmeingabe 1 ein "1" Bit in einer speziellen Tabelle (Modifikationstabelle) für jeden zu modifizierenden Befehl (das sind alle Befehle, denen kein X vorausgeht) und ein "0" Bit für jeden nicht zu modifizierenden Befehl.
Ebenso wird ein Kennzeichen in einer anderen speziellen Tabelle gehalten (Tabelle des Stanzbereichs) und zwar von jedem Speicherbereich, in den Worte gespeichert werden. Ein "Direktbefehl" (+) im umzuwandelnden Programm ändert den Speicher ohne die Modifikationstabelle oder die Tabelle des Stanzbereichs zu verändern, alle Eingabeschlüsselworte jedoch setzen diese Tabelle. D.h., wenn ein zu modifizierendes Wort in einer bestimmten Zelle gespeichert wurde, und ein "Direktbefehl" (+) speichert das Wort 123H01Q8 auf diese Zelle, so zeigt die Modifikationstabelle immer noch an, daß dieses Wort zu modifizieren ist. Nach Einlesen kann das umzuwandelnde Programm mittels Programmeingabe 1 geändert werden. Für Korrekturen ist die Taste PS-4 vorgesehen. (Siehe Programmsprungtasten).

Nach dem Sprung zum Stanzen von H4-10.0 wählt der Rechner die Schreibmaschine zur Eingabe aus und verlangt das Auswahlschlüsselwort für die Ausgabeeinheit. Drücke die Taste RECHNEN START, um den Stanzer der Schreibmaschine auszuwählen, oder gebe eine Null ein und drücke RECHNEN START, um den Schnellstanzer auszuwählen.
Nach Auswahl der Ausgabeeinheit wählt der Rechner noch einmal die Schreibmaschine zur Eingabe aus; diesmal für die Eingabe des Ausgabemodifikators. Das ist eine Zahl in dezimaler Spur-/Sektorschreibweise, die von den modifizierbaren Adressen angezogen wird. Der Modifikator muß vierziffrig sein oder kleiner. Das umzuwandelnde Programm sei beispielsweise ab Zelle 2000 gespeichert. Soll es ausgegeben werden mit Adressen relativ zu 0000, so gäbe man einen Ausgabemodifikator von 2000 ein. Sollen die Programmadressen relativ zu 0800 sein, so würde man den Modifikator 1200 eingeben.

AUFRUFFOLGE

Das Programm hat folgende fünf Eingänge:

L_o + 0600 - zur Eingabe des umzuwandelnden Programmes
L_o + 0601 - zur Ausgabe des umgewandelten Programmes
L_o + 0602 - zur Wiederherstellung von Programmeingabe 1 falls der Streifen nicht verifiziert wird.
L_o + 0603 - zum Verifizieren des Programmes
L_o + 0604 - zur Wiederherstellung von Programmeingabe 1 nach dem Verifizieren

L_o ist die Anfangsadresse von Programmeingabe 1.

AUSGABE

Wird nach der Eingabe des Ausgabemodifikators RECHNEN START gedrückt, so wird das umgewandelte Programm in willkürlich verlegbarer hexadezimaler Form ausgegeben. Die Tabelle des Stanzbereiches gibt an, welche Worte gestanzt werden, und die Modifikationstabelle gibt an, welche Befehle modifizierbar sind.
Die umgewandelten Worte werden in Gruppen von 64 oder weniger gestanzt. Jeder Gruppe geht ein Schlüsselwort für hexadezimales Füllen (M) voraus, und jeder Gruppe folgt eine Prüfsumme. Weitere Erläuterungen zum Format des hexadezimalen Streifens befinden sich in der Beschreibung zu Programmeingabe 1. Nach Beendigung der Ausgabe erfolgt ein Sprung zur Anfangsadresse von Programmeingabe 1.
Nach der Ausgabe können sowohl Inhalt als auch Prüfsummen des Ausgabestreifens verifiziert werden. Man lege dazu den erhaltenen Streifen in den Leser und springe nach L_o + 0603, wobei L_o die Anfangsadresse von Programmeingabe 1 ist. Der Rechner wählt die Schreibmaschine zur Eingabe eines Modifikators aus, der so beschaffen sein muß, daß das Programm genau dahin gespeichert wird, von wo es ausgestanzt wurde. Als dann gebe man das Auswahlschlüsselwort für die Eingabeeinheit an (falls notwendig); drücke RECHNEN START, und der Streifen wird verifiziert.

Anmerkung:
Beim Verifizieren wird der Streifen nicht wirklich gespeichert. Stattdessen wird der Inhalt des Streifens mit dem Inhalt des Speichers verglichen. Ist der Streifen nicht in Ordnung, so schreibt die Schreibmaschine ein Zeichen aus: "w" zeigt an, ein Speicherwort stimmt nicht überein; "e" zeigt an, daß eine Prüfsumme nicht stimmt.

Nach dem Verifizieren springe man nach L_o + 0604, um Programmeingabe 1 wiederherzustellen. Falls nicht verifiziert wurde, springe man zur Wiederherstellung nach L_o + 0602. (Siehe PST unter Programmsprungtasten.)

BEDIENUNG

1. PROGRAMMEINGABE

Der Streifen enthält das Programm in zwei Fassungen: willkürlich verlegbar hexadezimal, zulässig für Programmeingabe 1 und willkürlich verlegbar dezimal, in Kodierungsblattformat. Wird der willkürlich verlegbare hexadezimale Streifen eingelesen, so benutze man ein Modifikatorschlüsselwort (/), das der Anfangsadresse von Programmeingabe 1 gleich ist. Wird der dezimale Streifen benutzt, so gebe man ein Füllschlüsselwort mit L_o+0329 und ein Modifikatorschlüsselwort mit L_o ein, wobei L_o die Anfangsadresse von Programmeingabe 1 ist.

2. ERFORDERLICHE EINGABE-/AUSGABEEINHEITEN

Das Programm wählt die Lochstreifenschreibmaschine für die Eingabe aus; dennoch kann durch ein Auswahlschlüsselwort eine andere Einheit ausgewählt werden. Die Ausgabeeinheit muß vom Bediener ausgewählt werden (siehe unter "Eingabe").

3. PROGRAMMSPRUNGTASTEN

Taste	Funktion
PS-4	EIN - Verhindert Eingang in die speziellen Tabellen, wenn die Worte mit einen Füllschlüsselwort (;) eingegeben werden. Dadurch sind nach der Speicherung Korrekturen im umzuwandelnden Programm möglich, ohne die Anzahl von Tabellen zu erhöhen, da die Zellen nicht zweimal gestanzt werden. Dieser Weg darf erst gegangen werden, wenn alle zu stanzenden Bereiche eingegeben wurden. Da die Änderungen durch den "Direktbefehl" (+) die Tabellen in keiner Weise verändern, so können sie zu jeder Zeit ohne Rücksicht auf den Zustand von PS-4 vorgenommen werden.
	AUS - Gestattet das Anlegen einer speziellen Speichertafel beim Einlesen der einzelnen Worte durch Programmeingabe 1.
PST	EIN - Beim Sprung, entweder nach L_o + 0602 oder nach L_o + 0603, wird die Programmeingabe 1 auf ihren hexadezimalen Eingabeteil reduziert, d.h. die Verbindung mit dem dezimalen Eingabeteil wird zerstört.
	AUS - Beim Sprung, entweder nach L_o + 0602 oder nach L_o + 0603, werden sowohl der dezimale als auch der hexadezimale Eingabeteil von Programmeingabe 1 wiederhergestellt.

Anmerkung:
Wird ein Sprung mit eingeschaltetem PST entweder nach L_o + 0602 oder nach L_o + 0603 ausgeführt, so bewirkt ein erneuter Sprung nach L_o + 0602 oder nach L_o + 0603 mit ausgeschaltetem PST nicht die Wiederherstellung des dezimalen Teiles von Programmeingabe 1.

4. BEDIENUNG DES PROGRAMMES

a) Ist Programmeingabe 1 nicht im Speicher, so gebe man sie ein; man beginne vorzugsweise mit Spur 00 (das vollständige Programm muß vorliegen, 3 Spuren,29 Sektoren).

b) Gebe das Programm H4-10.0 in die unmittelbar auf Programmeingabe 1 folgenden Zellen ein unter Benutzung des Modifikators (L_o der Programmeingabe 1).

c) Springe nach L_o + 0600, wobei L_o die Anfangsadresse von Programmeingabe 1 ist. Das Programm H4-10.0 ändert die Programmeingabe 1 ab und bereitet ihre Tabellen vor; dann verlangt die Schreibmaschine Eingabe.

d) Gebe das umzuwandelnde Programm ein: gebe die notwendigen Schlüsselworte ein (;) und (/) und, wenn gewünscht, ein Auswahlschlüsselwort. Das Programm liest den Streifen ein, bis ein "Stop und Sprung" auftritt. (Der Sprung wird wie üblich durchgeführt.)

e) Sollen irgendwelche Änderungen im umzuwandelnden Programm vorgenommen werden, so setze man PS-4 und gebe dann die Änderung ein.

f) Springe nach L_o + 0601, wobei L_o die Anfangsadresse von Programmeingabe 1 ist. Die Schreibmaschine verlangt Eingabe.

g) Wähle die Ausgabeeinheit aus. Die Schreibmaschine verlangt wieder Eingabe.

h) Gebe den Ausgabemodifikator ein. Das umgewandelte Programm wird ausgegeben.

i) Nach der Ausgabe erfolgt ein Sprung nach L_o von Programmeingabe 1.

j) Verifizieren des hexadezimalen Streifens:
   1. Streifen in Schreibmaschinenleser einlegen und nach L_o + 0603 springen. Schreibmaschine verlangt Eingabe.
   2. Gebe ein Modifikatorschlüsselwort ein, das den Streifen auf die gleichen Zellen einliest, aus denen er ausgestanzt wurde. Der Streifen wird gelesen und verifiziert.
   3. Nach dem Verifizieren springe nach L_o + 0604, um Programmeingabe 1 wiederherzustellen (siehe PST unter Programmsprungtasten).

k) Soll der Streifen nicht verifiziert werden, so springe man nach L_o + 0602, um Programmeingabe 1 wiederherzustellen (siehe PST unter Programmsprungtasten).

5. FEHLERSTOPS

Ausschrift	Bedeutung
E	Prüfsummenfehler.
W	Streifenfehler (Speicherwort)

PROGRAMM - EINGABE 1	J1-10.0

VERWENDUNGSZWECK

"Programm Eingabe 1", J1-10.0, ist ein Speicherprogramm. Es erleichtert das Einlesen von hexadezimalen Bändern - willkürlich verlegbar oder nicht - und errechnet während des Einlesens eine Prüfsumme. Es können ebenfalls dezimale Programme, die willkürlich verlegbar sind, eingelesen und geprüft werden.
Vom Programm J1-10.0 werden folgende Funktionen ausgeführt:

1. Einlesen, binärisieren und speichern von Befehlsworten auf vorgegebene Speicherplätze.
2. Einlesen und speichern von hexadezimalen oder alphanumerischen Worten auf vorgegebene Speicherplätze ohne Binärisierung.
3. Einen Adressenmodifikator in einem Programm auf einen bestimmten Wert setzen.
4. Einlesen von dezimalen Adressen, die um die Modifikatorgröße erhöht werden.
5. Setzen des "Start fill" Zählers auf einen bestimmten Adressenwert.
6. Einlesen bestimmter Befehle und deren sofortige Ausführung.
7. Einlesen dezimaler Adressen, gefolgt von einem Sprung entweder zu diesem Speicherplatz oder zu dem Speicherplatz, der durch die Adresse plus Modifikator gegeben ist.
8. Einlesen einer dezimalen Adresse; der Inhalt dieses Speicherplatzes erscheint im Akkumulator.
9. Einlesen, binärisieren und speichern von willkürlich verlegbaren dezimalen Streifen.
10. Einlesen und speichern von hexadezimalen Streifen.
11. Auswählen einer bestimmten Eingabeeinheit.

Die hexadezimalen Streifen werden gewöhnlich von einem "hexadezimalen Ausgabe Programm" hergestellt, wie etwa das Programm J4-10.0.

SPEICHERBEDARF

Nur hexadezimale Eingabe	2 Spuren
Kombinierte Eingabe	3 Spuren, 29 Sektoren
Zwischenspeicher	keine

EINGABE

Die "Programm - Eingabe 1" enthält 12 Eingabe-Schlüsselworte, die dazu bestimmt sind, die gewünschten Funktionen auszuführen. Ihr Format und Verwendungszweck wird auf den folgenden Seiten erläutert.
Die erforderliche Eingabeeinheit - soweit eine andere als die Standardschreibmaschine benutzt wird - muß durch ein spezielles Schlüsselwort ausgewählt werden. Alle Adressen müssen in Spur-/Sektorschreibweise erscheinen.

MNNKHHHH

Das hexadezimale Füllschlüsselwort für willkürliche Verlegung liest NN hexadezimale Worte in aufsteigender Folge auf aufeinanderfolgende Speicherplätze.

Das Einlesen beginnt bei Speicherplatz HHHH plus Modifikator. Das "K" ist ein Speicherbefehl. Das Eingabe-Schlüsselwort enthält ein Kennzeichnungsbit (1@31) und jedes der NN Worte, die modifiziert werden sollen, müssen ein Kennzeichnungsbit enthalten. Der Modifikator (siehe unten) wird zum Operandenadressteil eines jeden gekennzeichneten Wortes addiert, bevor es gespeichert wird. Während alle NN Worte gespeichert werden, wird eine Prüfsumme von den Wörtern und dem Eingabe-Schlüsselwort (alle Werte ohne Modifikator) berechnet. Stimmt die berechnete Prüfsumme nicht mit der von einem Ausgabeprogramm verher auf dem Streifen gestanzten Prüfsumme überein, so erfolgt Fehlerstop. Stimmen die Prüfsummen überein, so verlangt der Rechner ein neues Eingabe-Schlüsselwort.
Z.B. sei der Modifikator 1200 (dezimal) und das Eingabe-Schlüsselwort M40K1201' (hexadezimal).
Die folgenden 64 hexadezimalen Worte werden modifiziert, soweit sie ein Kennzeichnungsbit enthalten und, beginnend mit Speicherplatz 3000, aufeinanderfolgend gespeichert.

VNNCHHHH

Das hexadezimale Füllschlüsselwort für nicht willkürlich verlegbare Streifen liest beginnend mit Speicherplatz HHHH "NN" hexadezimale Worte ein. "C" ist ein Speicherbefehl. Das Eingabe-Schlüsselwort enthält kein Kennzeichnungsbit. Bei allen hexadezimalen Worten, die etwa ein Kennzeichnungsbit enthalten, wird beim Speichern der zuletzt eingegebene Modifikator zum Operandenadressteil addiert.
Während alle NN Worte gespeichert werden, wird eine Prüfsumme vom Eingabe-Schlüsselwort und den Wörtern berechnet. Stimmt die berechnete Prüfsumme mit der vorher vom Ausgabeprogramm auf dem Streifen gestanzten Prüfsumme überein, so verlangt der Rechner ein neues Eingabe-Schlüsselwort.
Stimmen sie nicht überein, so erfolgt ein Fehlerstop.

VNNCTTSS

Z.B. das Eingabe-Schlüsselwort V40C1100' bewirkt, daß die folgenden 64 hexadezimalen Wörter, beginnend mit Speicherplatz 1700, gespeichert werden.
Beim Einlesen von hexadezimalen Streifen mit "V" oder "M" Schlüsselwörtern prüft das "Programm Eingabe" Programm jedes gespeicherte Wort, und zwar nachdem ein Wort gespeichert ist, wird der Wert im Speicher mit dem eingelesenen Wert verglichen. Stimmen die beiden Werte nicht überein, so erfolgt ein Fehlerstop.

/OOOTTSS'
Das Modifikator Eingabe-Schlüsselwort setzt den Adressenmodifikator auf die Adresse TTSS, die das Eingabe-Schlüsselwort enthält.

Bei der Eingabe von hexadezimalen Streifen wird zu allen Schlüsselworten und Worten, die ein Kennzeichnungsbit enthalten, dieser Adressenmodifikator addiert.
Bei der Eingabe von dezimalen Befehlsworten wird er bei allen Befehlen, vor denen kein "X" steht, zu den Operandenadressen addiert.
Der Modifikator bleibt solange unverändert, bis er durch ein neues Modifikator Eingabe-Schlüsselwort ersetzt wird.
Z.B. das Eingabe-Schlüsselwort /0001200' bewirkt, daß zu allen gekennzeichneten Operandenadressen und Schlüsselwörtern 1200 addiert wird.

.000TTSS'

Das absolute Halt- und Sprung-Eingabe-Schlüsselwort bewirkt, daß der Rechner anhält. Wird anschließend die Start-Taste gedrückt, erfolgt ein Sprung nach der Zelle TTSS. Falls der Schalter PS-32 auf "ON" steht, wird der Stop unterdrückt.
Z.B. das Eingabe-Schlüsselwort .0001663' würde einen Sprung nach 1663 bewirken.

.100TTSS'

Das relative Halt- und Sprung-Eingabe-Schlüsselwort bewirkt, daß der Rechner stoppt. Wird anschließend die Start-Taste gedrückt, so erfolgt ein Sprung nach der Zelle TTSS plus Modifikator. Wenn der Schalter PS-32 auf "ON" steht, wird der Stop unterdrückt.
Z.B. der Modifikator sei auf 1200 gesetzt worden, dann würde das Eingabe-Schlüsselwort .1001663' einen Sprung nach 2863 bewirken.

S000TTyy'

Das Auswahlschlüsselwort wählt diejenige Eingabeeinheit aus, die durch die dezimale Spuradresse TT angegeben wird. (yy kann eine beliebige Sektoradresse sein) Eine Liste, die für bestimmte Spuradressen die entsprechenden Eingabeeinheiten angibt, findet man im LGP-21 Programmierungshandbuch. Nach der Eingabe des Auswahlschlüsselwortes hält der Rechner an. Nach Drücken der Start-Taste liest der Rechner über die ausgewählte Eingabeeinheit weiter ein.
Springt man aber erneut an den Anfang des Eingabeprogrammes J1-10.0, so wird automatisch die Standardschreibmaschine für die Eingabe ausgewählt. Es wird nun solange über die Standardschreibmaschine eingelesen, bis ein neues Auswahlschlüsselwort eine andere Einheit auswählt.
Wenn z.B. 32 sich auf eine 2. Schreibmaschine bezieht, dann wird durch das Schlüsselwort S0003200' diese Schreibmaschine für die weitere Eingabe ausgewählt.
Die folgenden Schlüsselworte beziehen sich auf den Programmteil von J1-10.0, der die dezimale Eingabe ermöglicht.

,000TTSS'

Das dezimale Füllschlüsselwort (Start fill) veranlaßt das Eingabeprogramm, dezimale Worte in aufsteigender Folge, beginnend mit Speicherplatz TTSS, auf aufeinanderfolgende Zellen zu speichern.

Jede zulässige Adresse, gefolgt von einem Stopcode, darf benutzt werden. Vor jeder dezimalen Eingabe sollte ein dezimales Füllschlüsselwort (Start fill) stehen. Die "Start fill" Adresse wird im "Start fill" Zähler gespeichert. Diese Speicherzelle des Unterprogrammes enthält die Adresse für die dezimale Eingabe. Der Zähler wird nach dem Einlesen und Speichern eines Wortes (nicht eines Schlüsselwortes) jedesmal um 1 erhöht. Die eingelesenen Worte werden fortlaufend gespeichert, bis ein neues dezimales Füllschlüsselwort gelesen oder die Eingabe beendet wird.
Z.B. das Füllschlüsselwort ;0000400' würde eingelesene Worte beginnend mit Speicherplatz 0400 speichern.

,00000NN'

Das Schlüsselwort für die Eingabe von hexadezimalen Worten veranlaßt, daß NN hexadezimale Worte gespeichert werden; und zwar beginnend in der Speicherzelle, deren Adresse im "Start fill" Zähler steht. "NN" muß dezimal angegeben werden und kann den Wert 1 bis 63 annehmen. Die gespeicherten Worte werden nicht um dem Adressenmodifikator erhöht. Die Führungsnullen bei den hexadezimalen Worten, die eingelesen werden sollen, dürfen weggelassen werden. Nach jedem hexadezimalen Wort und Schlüsselwort muß jeweils ein Stopcode folgen. Soll eine Null eingegeben werden, ist es nicht notwendig, eine "0" zu schreiben. Es genügt dann ein Stopcode. Jedes hexadezimale Wort darf nicht aus mehr als 8 Zeichen bestehen.
Z.B. das Eingabe-Schlüsselwort ,0000003' gefolgt von den drei hexadezimalen Worten -2'JK73''- würde bewirken, daß die Werte 00000002,0000JK73 und 00000000 in die nächsten drei laufenden Zellen gespeichert werden.

A00000NN'

Das Schlüsselwort für die Eingabe von alphanumerischen Worten veranlaßt, daß NN alphanumerische Worte gespeichert werden und zwar beginnend in der Speicherzelle, deren Adresse im "Start fill" Zähler steht. Die alphanumerischen Worte werden um zwei Stellen nach links geschiftet und ohne Binärisation gespeichert. "NN" muß dezimal angegeben werden und kann den Wert 1 bis 63 annehmen.
Jedes Wort kann bis zu fünf Zeichen in der Sechs-Bit Darstellung enthalten (werden mehr als fünf Zeichen pro Wort eingegeben, so bleiben nur die letzten fünf erhalten). Das letzte gespeicherte Wort erhält ein Gruppenendkennzeichen, d.h. 1@30. Alle Zeichen außer Bandlauf, Code löschen und Stopcode können eingegeben werden, z.B. Kleinschreibung kann gespeichert werden.
Z.B. das Eingabe-Schlüsselwort A0000004' gefolgt von den vier Worten -TOTAL' UNIT'S PRO'DUCED'- würde bewirken, daß die Information TOTAL UNITS PRODUCED in vier aufeinanderfolgenden Speicherplätzen ohne Umwandlung ins Binäre gespeichert wird.

+00LTTSS'
Das Befehlseingabe-Schlüsselwort wird vom Unterprogramm wie ein Befehl behandelt. Der im Schlüsselwort enthaltene Befehl LTTSS - wobei L ein beliebiger positiver Maschinenbefehl ist und TTSS eine beliebige zulässige Adresse darstellt- wird ausgeführt, nachdem entweder Eingabe eines darauffolgenden hexadezimalen Wortes, oder Eingabe eines Stopcodes (falls kein darauffolgendes Wort benötigt wird) erfolgte.
Der Adressteil des Eingabe-Schlüsselwortes wird nicht durch den Adressenmodifikator verändert.
Jedes Schlüssel- und Eingabewort muß von einem Stopcode gefolgt sein.
Im hexadezimalen Wort dürfen linksseitige Nullen weggelassen werden.
Der Rechner kehrt zu einem Eingabebefehl zurück, ohne daß die Ausführung des Befehls eine Verzweigung bewirkte. Sprungbefehle sollten vermieden werden.
Anmerkung: Der im Eingabe-Schlüsselwort enthaltene Befehl verändert den Füllzähler nicht.
Beispielsweise würde das Eingabe-Schlüsselwort +00h1637' gefolgt von dem hexadezimalen Wort 73W08' bewirken, daß der Wert 00073W08 in der Speicherzelle 1637 abgelegt wird.

-000TTSS'

Das Eingabe-Schlüsselwort zur aufeinanderfolgenden Anzeige bewirkt, daß der Inhalt der Speicherzelle TTSS im Akkumulator erscheint, wenn der Rechner anhält. Ein Startsignal bewirkt die Rückkehr zu einem Eingabebefehl für ein neues Schlüsselwort, wenn PST ausgeschaltet ist. Ist PST eingeschaltet, so erscheint im Akkumulator der Inhalt von TTSS+i (i=1,2,...)
Bei jedem Startsignal wird i um 1 erhöht. Man kann auch die Adresse des angezeigten Wortes erhalten, wenn man auf Einzeloperationen schaltet und startet. (Im Akkumulator erscheint der B-Befehl, der die eben angezeigte Information brachte.) TTSS darf eine beliebige zulässige Adresse sein.
Das Eingabe-Schlüsselwort -0003263' würde beispielsweise direkt den Inhalt der Speicherzelle 3263 im Akkumulator anzeigen.
Jedes eingegebene Wort, dessen ganz linksgelegene vier Bits entweder einer "8" entsprechen (negativer Befehl), oder einer Null (positiver Befehl), wird als Befehl behandelt. Die Operandenadresse wird durch den Adressenmodifikator verändert, es sei denn, dem Befehl gehe ein "X" voraus. Das Unterprogramm erlaubt den Einschluß von Indexkennzeichen (von einigen Interpretiersystemen benutzt) unmittelbar vor den Befehlen. Einschluß eines "X" oder eines Indexkennzeichens verändert den Befehl im Speicher nicht. Entsprechen die ganz linksgelegenen vier Bits eines Wortes einer der Zahlen "9" bis "13", so erfolgt ein Fehlerstop(siehe Liste der Fehlerstops).
Beispiel: Der Adressenmodifikator sei 0600. Verschiedene Typen von Befehlsworten würden wie folgt gespeichert werden:

Eingabe	Speicherung
B 1234	B 1834
XB 1234	B 1234
800T 1234	800T 1834
80XT 1234	800T 1234
8B 1234	8B 1834
X8B 1234	8B 1234

Das Unterprogramm besteht aus zwei Teilen: hexadezimale Eingabe und dezimale Eingabe. Der Abschnitt mit der hexadezimalen Eingabe kann unabhängig von der dezimalen Eingabe benutzt werden, die dezimale Eingabe hängt jedoch mit der hexadezimalen zusammen, die immer gespeichert sein muß.

AUFRUFFOLGE

Das Programm wird von Hand aufgerufen durch Sprung zur Anfangsadresse (ganz gleich welcher Abschnitt benutzt wird).

BEDIENUNGSVORSCHRIFT

1. PROGRAMMEINGABE

Der Streifen enthält das Programm in willkürlich verlegbarer hexadezimaler Form mit eigenem Bootstrap. Die Bedienung erfolgt so (TR = Tastenfeld des Rechners) :
a) Programmstreifen in Lesevorrichtung der Schreibmaschine legen und alle Hebel lösen.
b) Hebel für die Betriebsarten auf "Eingabe von Hand" schalten (TR).
c) "Lesen Start" auf Schreibmaschine drücken.
d) Drücke "Löschen und Eingeben" (Fill Clear) (TR).
e) Drücke "Lesen Start".
f) Soll das Programm verlegt werden, so gebe man über die Schreibmaschine die Anfangsadresse in hexadezimaler Form ein (gewöhnlich ist die Anfangsadresse 0000); im anderen Falle unterbleibt dieser Schritt.
g) Betriebsartenschalter auf "Einzeloperation" schalten.
h) Drücke "Ausführen" (TR).
i) Wiederhole die Schritte b bis g, bis vom Reohner "normal" ausgeschrieben wird.
j) Betriebsartenschalter auf NORMAL stellen.
k) Drücke START (TR)
l) Das Programm liest automatisch weiter ein.

Anmerkung: Ist PS-32 ausgesohaltet, so hält der Reohner naoh dem Einlesen des Abschnittes mit der hexadezimalen Eingabe an. Ist PS-32 eingeschaltet oder wird START gedrückt, so wird der dezimale Eingabeabschnitt oder ein hexadezimaler Objektprogrammstreifen eingelesen.

PS-32 sollte ausgeschaltet werden, bevor der Abschnitt mit der dezimalen Eingabe völlig eingelesen ist, sonst arbeitet die Lesevorrichtung nach dem Einlesen ohne Streifen weiter.
Der Bootstrap belegt immer die Speicherzellen 0000, 0002-0008 (absolut), während der Rest auf dem Bereich steht, der für den hexadezimalen Eingabeabschnitt bestimmt ist.

2. VORSCHRIFT ZUM WIEDEREINLESEN

Der Bootstrap bleibt im Speicher erhalten, wenn das Programm nicht verlegt wurde. Zum Wiedereinlesen ohne Bootstrap verfahre man wie folgt:

a) Programm in den Leser einlegen und zwar den Bandlaufabschnitt zwischen Bootstrap und Programm.
b) Betriebsartenschalter auf "Eingabe von Hand" stellen (TR).
c) C0000 über Schreibmaschine.
d) Drücke "Löschen und Eingeben" (TR).
e) Modifikator in hexadezimaler Form schreiben (alle 8 Zeichen) oder 8 Nullen, wenn das Programm nicht verlegt werden soll.
f) Betriebsartenschalter auf Einzeloperationen stellen (TR).
g) Drücke "Ausführen" (TR).
h) Betriebsartenschalter auf "Eingabe von Hand" schalten (TR).
i) Schreibe U0008.
j) Drücke "Löschen und Eingeben" (TR).
k) Betriebsartenschalter auf "Einzeloperation" (TR).
l) Drücke "Ausführen".
m) Betriebsartenschalter auf NORMAL (TR).
n) Drücke "Start". Das Programm wird automatisch eingelesen.

3. ERFORDERLICHE EINGABE-/AUSGABEEINHEITEN

Von der Standardschreibmaschine verschiedene Eingabeeinheiten müssen durch einen Auswahlcode angezeigt werden. Ausgabeeinheiten sind nicht erforderlich.

4. DIE WIRKUNG DER PROGRAMM - SPRUNGTASTE

PS-32

AUS - Stop vor dem Sprung bei Benutzung eines Halt- und Sprungeingabeschlüsselwortes.
Stop nach Einlesen des hexadezimalen Eingabeabschnitts des Unterprogrammes.

EIN - kein Stop bei Benutzung eines Halt- und Sprungeingabeschlüsselwortes.
Kein Stop nach dem Einlesen des hexadezimalen oder dezimalen Abschnitts des Unterprogrammes.

PST

AUS - nur der Inhalt einer Speicherzelle erscheint im Akkumulator.

EIN - Aufeinanderfolgende Speicherzellen können durch fortlaufendes Starten in den Akkumulator gebracht werden.

5. FEHLERSTOP

E Beim Einlesen eines hexadezimalen Streifens zeigt dieser Buchstabe einen Prüfsummenfehler an.
Beim Einlesen eines dezimalen Streifens zeigt er einen falschen Eingabecode an.
Ein Prüfsummenfehler wird wie folgt behoben:
a) Den Streifen bis zum letzten M oder V-Eingabeschlüsselwort zurücklegen.
b) Drücke START (eine Eingabeeinheit braucht nicht neu eingelesen zu werden).
Die Behebung eines falschen Eingabecodes erfolgt so (das letzte gelesene Wort enthält den Fehler):
a) Drücke "Eingabe von Hand" auf der Schreibmaschine.
b) Stelle Betriebsartenschalter auf "Eingabe von Hand".
c) * Schreibe in hexadezimaler Form einen Sprung zur Anfangsadresse des Unterprogrammes (d.h. U L_0)
d) Drücke "Löschen und Eingeben" (TR).
e) * Stelle Betriebsartenschalter auf "Einzeloperation" (TR).
f) * Drücke "Ausführen" (TR).
* Falls das Unterprogramm in Speicherzelle 0000 beginnt, lasse man die Schritte c,e und f weg.
g) Stelle Betriebsartenschalter auf "Normal" (TR).
h) Drücke "Start" (TR). Die Lampe der Schreibmaschine leuchtet auf.
i) Richtigen Eingabecode eingeben.
j) Ist die Schreibmaschine die Eingabeeinheit, so löse man den Hebel "Eingabe von Hand", und der Streifen wird eingelesen.
Wird eine andere Eingabeeinheit benutzt, so gebe man den Auswahlcode ein und drücke den Hebel "Rechner Start", dann "Start" (TR), und der Streifen wird eingelesen.

W Dieser Ausdruck zeigt an, daß während des Einlesen eines hexadezimalen Streifens ein Protokollfehler erfolgte, (d.h. das wirklich gespeicherte Wort war dem vom Streifen gelesenen nicht gleich). Dieser Fehler wird wie ein Prüfsummenfehler behoben.

BEMERKUNG

Wenn man im Unterprogramm zwei Befehle ändert, kann man es zum Ausprüfen von hexadezimalen Streifen verwenden. Z.B. Nachdem ein hexadezimaler Streifen ausgestanzt worden ist, ändere man die zwei Befehle im Unterprogramm und lese dann den hexadezimalen Streifen ein.

PROGRAMM - EINGABE 1	J1-10.0

Das Unterprogramm liest den Streifen (ohne ihn zu speichern) und vergleicht ihn Wort für Wort mit dem Speicher. Auch die Prüfsummen werden ausgetestet. Ist der Vergleich nicht erfolgreich, so hält das Programm mit dem Fehlerstop "W" oder "E". Der Streifen ist in Ordnung, wenn er ohne Schwierigkeiten eingelesen wird.

Die zu ändernden Befehle sind:

Speicherplatz	Inhalt (alt)	Inhalt (neu)
L_o+24	H[ ]	U0025
L_o+31	Y0024	U0032

ZUSAMMENFASSUNG

Eingabeschlüsselwort	Interpretiert als
ØTTSS	Befehl - darf modifiziert werden.
XØTTSS	Befehl - darf nicht modifiziert werden.
800ØTTSS	Negativer Befehl - darf modifiziert werden.
80XØTTSS	Negativer Befehl - darf nicht modifiziert werden.
IØTTSS	Befehl mit Indexkennzeichen - darf modifiziert werden.
XIØTTSS	Befehl mit Indexkennzeichen - darf nicht modifiziert werden.
MNNKHHHH	Füllschlüsselwort für willkürlich verlegbare, hexadezimale Streifen - ließt NN hexadezimale Worte auf aufeinanderfolgende Speicherzellen ein, beginnend mit der Speicherzelle HHHH plus Modifikator. Alle Worte, deren Operandenadresse modifiziert werden soll, müssen eine 1@31 enthalten. Während der Eingabe wird eine Prüfsumme errechnet und mit der Prüfsumme auf dem Streifen verglichen.
VNNCHHHH	Füllschlüsselwort für nicht willkürlich verlegbare hexadezimale Streifen - liest NN hexadezimale Worte auf aufeinanderfolgende Speicherzellen ein, beginnend mit Speicherzelle HHHH. Alle Worte, deren Operandenadresse modifiziert werden soll, müssen eine 1@31 enthalten. Während der Eingabe wird eine Prüfsumme errechnet und mit der Prüfsumme des Streifens verglichen.
/000TTSS	Modifikator setzen - setze einen Adressenmodifikator im Speicher auf TTSS. Er wird zu allen hexadezimalen Worten addiert, die ein Kennzeichnungsbit haben, und zu den Adressen von dezimalen Befehlen, denen kein X vorausgeht.

.000TTSS	Schlüsselwort für absoluten "Stop und Sprung"- Stop: Wenn START gedrückt wird, erfolgt ein Sprung nach TTSS. Der Stop wird übergangen, wenn PS-32 eingeschaltet ist.
.100TTSS	Schlüsselwort für relativen "Stop und Sprung"- Stop: Wenn START gedrückt wird, erfolgt ein Sprung nach TTSS plus Modifikator. Der Stop wird übergangen, wenn PS-32 eingeschaltet ist.
S000TTyy	Auswahlschlüsselwort - wählt die Eingabeeinheit aus, die durch die dezimale Spuradresse TT gegeben ist (yy ist ohne Bedeutung). Siehe Liste der Schlüsselsymbole im LGP-21 Programmierungshandbuch. Nach Eingabe des Auswahlschlüsselwortes hält der Rechner an. Um fortzufahren, drücke man START.
;000TTSS	Füllschlüsselwort - liest dezimale Befehle in aufeinanderfolgende Speicherzellen ein, beginnend mit TTSS.
,00000NN	Hexadezimale Worte - liest NN hexadezimale Worte auf die folgenden NN Speicherzellen ein.
A00000NN	Alphanumerische Worte - liest NN alphanumerische Worte auf die folgenden NN Speicherzellen ein.
+00ØTTSS	Befehl - führe den Befehl ØTTSS aus, der im Schlüsselwort enthalten ist, aber erst nach Eingabe des folgenden Wortes in hexadezimaler Schreibweise, oder nach Eingabe eines Stopcodes, wenn kein weiteres Wort nötig ist.
-000TTSS	Schlüsselwort für aufeinanderfolgende Anzeige - der Inhalt der Speicherzelle TTSS erscheint im Akkumulator, wenn der Rechner hält. Nach START verlangt der Rechner ein Eingabeschlüsselwort, falls PST ausgeschaltet ist. Ist PST eingeschaltet, so erscheint nach START im Akkumulator der Inhalt von TTSS+1, und so fort. Die Adresse der im Akkumulator angezeigten Zelle erscheint im Akkumulator, wenn man auf "Einzeloperation" geht und startet.

Ø stellt einen beliebigen der 16 Maschinenbefehle dar. TT sind dezimale Zeichen für den Spurteil der Adresse, SS sind dezimale Zeichen für den Sektorteil der Adresse. NN sind dezimale Zeichen, die die Anzahl der einzulesenden Worte darstellen. HHHH bedeutet: eine hexadezimale Adresse.

PROGRAMM - EINGABE 2

J1-10.1

VERWENDUNGSZWECK

Das Programm J1-10.1 ist ein allgemeines Speicherprogramm. Es übernimmt willkürlich verlegbare dezimale Programme, hexadezimale Worte und alphanumerische Worte. Es liest auch nicht willkürlich verlegbare hexadezimale Programme, berechnet Prüfsummen und vergleicht sie mit den Prüfsummen auf dem hexadezimalen Streifen. Darüberhinaus lassen sich mit diesem Programm Speicherinhalte in den Akkumulator bringen; es können Befehle direkt ausgeführt werden, ohne gespeichert zu sein; und man kann damit "von Hand" springen.

SPEICHERBEDARF

3 Spuren

Benutzte Zwischenspeicher — keine

EINGABE

Die eingegebenen Informationen sind Schlüsselworte, die dem Programm angeben:

a) das Format der Worte, die folgen,
b) die Worte, welche zu speichern sind,
c) die Worte, die unmittelbar ausgeführt werden sollen.

Diese Eingabeschlüsselworte und die Formate der ihnen folgenden Worte, werden auf den folgenden Seiten besprochen. (Man beachte bitte, daß für jeden in der Beschreibung angeführten Stopcode (') START COMP gedrückt werden muß, wenn die Eingabe über das Tastenfeld der Schreibmaschine erfolgt.)

;000TTSS' Füllschlüsselwort (Start Fill)

Das Füllschlüsselwort gibt die Anfangsadresse TTSS für eine Reihe zu speichernder Informationen an. Dieser Wert wird im "Füllzähler", einer Speicherzelle des Unterprogrammes aufbewahrt und gibt an, in welche Zelle das nächste Wort zu speichern ist. Der Zähler wird jedesmal erhöht, wenn ein Wort (kein Schlüsselwort) eingelesen und gespeichert wird. Die einzulesenden Worte können vorliegen in Befehlsform, als hexadezimale Worte oder als alphanumerische Worte. Die Worte werden in aufeinanderfolgende Speicherzellen gebracht bis Eingabe beendet ist oder bis ein anderes Füllschlüsselwort auftritt. Die Startfilladresse muß in dezimaler Spur-/Sektorschreibweise vorliegen, gefolgt von einem Stopcode. Jede reale Adresse kann benutzt werden.
Das Schlüsselwort ;0001600' bewirkt beispielsweise, daß die folgenden Informationen in aufeinanderfolgenden Zellen, beginnend in 1600, gespeichert werden.

OTTSS' Befehl

Einen Befehl erkennt das Unterprogramm an Nullen in den Bitpositionen 0 bis 3 oder bei negativen Befehlen an einer "1" in der Vorzeichenstelle und Nullen in Bitposition 1 bis 3. Der Befehl besteht aus dem alphabetischen Befehlssymbol O, gefolgt von der Operandenadresse TTSS, die die vier dezimalen Ziffern der Spur-/Sektoradresse ausmachen.

Wird der Akkumulator vor der Eingabe gelöscht, so können linksseitige Nullen (das sind solche, die dem Befehl vorausgehen) weggelassen werden. Der gesamte Befehl wird in sein duales Äquivalent umgewandelt und in der vom Füllzähler angegebenen Zelle gespeichert.

/OOOTTSS' Modifikator

Der Adressteil (TTSS) dieses Schlüsselwortes wird im Unterprogramm in der für den Modifikator bestimmtenZelle gespeichert. Der Modifikator (d.h. der Inhalt dieser Zelle) wird zurOperandenadresse aller nacheinander eingelesenen Befehlsworte addiert,ausgenommen von den Befehlen, denen ein "X" vorausgeht. Der Modifikator ist wirksam bis ein anderes Modifikatorschlüsselwort ihn ersetzt. Jede reale Adresse in dezimaler Spur-/Sektorschreibweise kann benutzt werden und muß von einem Stopcode gefolgt sein.
Das Schlüsselwort /0001100' speichert beispielsweise den Wert 1100 in der Modifikatorzelle. 1100 wird zur Adresse aller folgenden Befehle addiert, denen kein "X" vorausgeht:

eingelesene Befehle	gespeicherte Befehle
B2611'	B3711
XB2611'	B2611
800Z0400	800Z1500
80XZ0400	800Z0400

Anmerkung

Jedem Füllschlüsselwort sollte ein Modifikatorschlüsselwort folgen. Sollen die Adressen nicht modifiziert werden, so wähle man das Schlüsselwort /0000000', da der Modifikator immer zu den Adressen modifizierbarer Befehle addiert wird.

.OOOTTSS' Halte und Springe (Stop and Transfer)

Dieses Schlüsselwort wird in zwei Schritten ausgeführt. Zuerst hält der Rechner an. Dann, nach einmaligem Drücken der START-Taste erfolgt ein Sprung nach TTSS. Jede reale Adresse in dezimaler Spur-/Sektorschreibweise kann benutzt werden. Jedem Schlüsselwort muß ein Stopcode folgen. Ist PS 32 gedrückt, bevor das Schlüsselwort eingegeben wird, so wird der Stop verhindert.
Das Schlüsselwort .0002000' bewirkt beispielsweise, daß der Rechner anhält, und daß nach einem Startsignal ein Sprung nach 2000 erfolgt.

+OOOTTSS' Direktbefehl

Dieses Schlüsselwort ermöglicht dem Programmierer, den Rechner einen Befehl direkt ausführen zu lassen. Der im Schlüsselwort enthaltene Befehl OTTSS, wobei O stellvertretend für ein Befehlssymbol steht und TTSS eine beliebige reale Adresse in dezimaler Spur-/Sektorschreibweise ist, wird ausgeführt nach der Eingabe eines hexadezimalen Wortes und/oder eines Stopcodes; d.h. im Anschluß an die Eingabe des hexadezimalen Wortes, auf das sich das Schlüsselwort bezieht, oder nur im Anschluß an die Eingabe eines Stopcodes, falls kein hexadezimales Wort benötigt wird.
Der Adressteil des Befehlsschlüsselwortes wird nicht modifiziert; auch das hinterher eingegebene hexadezimale Wort wird nicht modifiziert. Jedem Schlüsselwort und jedem hexadezimalen Wort muß ein Stopcode folgen. Führende Nullen im hexadezimalen Wort können weggelassen werden.

Der Direktbefehl +00H1637' gefolgt von dem hexadezimalen Wort 73W08' bewirkt beispielsweise, daß der Wert 00073W08 in Zelle 1637 gespeichert wird; das Wort +00U1735'' bewirkt einen Sprung nach 1735.

Anmerkung

Das Schlüsselwort für den Direktbefehl ändert der Füllzähler nicht.

,00000NN' hexadezimale Worte

Das Schlüsselwort bewirkt, daß die folgenden NN Worte als hexadezimale Worte gespeichert werden und zwar in aufeinanderfolgenden Speicherzellen entsprechend dem Stand des Füllzählers. NN muß kleiner als 64 sein. Die zu speichernden Worte müssen in hexadezimalem Format sein und werden nicht modifiziert.
;0002200', 0000003', 3W7'140Q'' bewirkt beispielsweise, daß der Speicher nach der Eingabe so aussieht:

2200	000003W7
2201	0000140Q
2202	00000000

Das Format von hexadezimalen Worten:
Ein hexadezimales Wort darf maximal acht Zeichen enthalten. Jede Kombination der Zeichen 0,1,...,9 und F,G,J,K,Q,W darf benutzt werden; Null ist der kleinste und WWWWWWWQ ist der größte Wert, den der Rechner übernehmen kann. Ist ein Wort Null, so ist nur ein Stopcode erforderlich. Linksseitige Nullen können weggelassen werden; jedem Wort muß ein Stopcode folgen.

VNNCHHHH' hexadezimales Füllen

Dieses Schlüsselwort bewirkt, daß die folgenden NN hexadezimalen Worte in aufeinanderfolgenden Zellen, beginnend mit HHHH (ein Füllschlüsselwort ist nicht nötig) gespeichert werden. Das "C" ist ein Löschbefehl. Das Schlüsselwort ist in hexadezimaler Schreibweise, und wird normalerweise von einem Programm für hexadezimale Ausgabe geliefert, wo es am Anfang eines Blockes von 64 oder weniger hexadezimalen Worten steht.
Während der Eingabe wird eine Prüfsumme errechnet aus Schlüsselwort und allen hexadezimalen Worten. Sind alle NN Worte gespeichert und ist PST aus, so wird diese Prüfsumme mit der Streifenprüfsumme am Ende des entsprechenden Blockes verglichen. Bei Gleichheit wird weiter eingelesen; bei Ungleichheit erfolgt ein Fehlerstop. Siehe unter "Fehlerstops". Ist PST eingeschaltet, so wird der Prüfsummenvergleich übersprungen.
Das Schlüsselwort V1WC2W00' bewirkt beispielsweise, daß die nächsten $1W_{16}=31_{10}$ hexadezimalen Worte gespeichert werden beginnend in Zelle $2W00_{16}=4700_{10}$.

A00000NN' Alphanumerische Worte

Das Schlüsselwort gestattet die Eingabe von alphanumerischen Informationen in 6-Bitart. Die so eingelesenen Informationen können durch Druckbefehle ohne Umwandlung direkt wieder ausgestanzt werden, oder durch ein alphanumerisches Druckprogramm. Jedes Wort, mit Ausnahme des letzten Wortes eines Blockes, muß 5 Zeichen enthalten (das letzte darf weniger enthalten) und von einem Stopcode gefolgt sein. Die Anzahl NN der zu speichernden Worte muß kleiner als 64 sein. Die NN Worte werden in aufeinanderfolgenden Zellen (bei q=29), beginnend in der dem Füllzähler entsprechenden Zelle, gespeichert.
Das Beispiel A0000005' DEPRE'SS ST'ART T'O CON'TINUE' bewirkt, daß DEPRESS START TO CONTINUE in fünf aufeinanderfolgenden Zellen gespeichert wird.

-000TTSS' Frageschlüsselwort

Das Frageschlüsselwort bringt Worte aus dem Speicher in den Akkumulator. Wenn der Rechner mit Oszilliskop ausgerüstet ist, kann der Programmierer sich die Worte ansehen, ohne daß Befehle ausgeführt werden. Es wird nur jeweils das Wort gebracht, das sich in TTSS befindet. Jedesmal wenn START gedrückt wird, erscheint das nächste Wort der steigenden Folge. Die folgende Bedienung ist auszuführen, um zur normalen Arbeitsweise zurückzukehren:
1. Wahlschalter auf "Eingabe von Hand" stellen.
2. Drücke "Eingabe und Löschen".
3. Wahlschalter auf "Normal" stellen.
4. Drücke "Start".

Das Schlüsselwort -0002003' würde beispielsweise den Inhalt von 2003 in den Akkumulator bringen. Nach "Start" erscheint der Inhalt von 2004 u.s.w. Das Verfahren kann fortgesetzt werden, bis obige Bedienung den Ablauf unterbricht.

S000TTss' Eingabeauswahl

Dieses Schlüsselwort bewirkt die Auswahl einer anderen Eingabeeinheit (als die Standardschreibmaschine für das Unterprogramm J1-10.1). Man beachte, daß die Standardschreibmaschine vom Unterprogramm ausgewählt wird beim Eingang in 0000 und nach Fehlerstops. Die Auswahlcodes und die dazugehörenden Einheiten sind im LGP-21 Programmierungshandbuch gelistet. Die Auswahlcodes müssen in dezimaler Spur-/Sektorschreibweise TTss angegeben werden, gefolgt von einem Stopcode (der Sektorteil ist dem Wert nach unwichtig). Ist eine zweite Schreibmaschine beispielsweise durch die Kennzahl 32 auswählbar, so bewirkt das Auswahlschlüsselwort S0003200', daß diese zweite Schreibmaschine für die Eingabe des Unterprogrammes ausgewählt wird.
Vom Programm zu lesende Streifen, sollten auf der Schreibmaschine ausgeschrieben werden können; d.h., nach jeweils vier bis acht Worten sollte ein Wagenrücklauf stehen.
Wird eine inkorrekte Information gelesen, so wählt das Programm die Schreibmaschine aus, schreibt "err" und hält an. Siehe unter Fehlerstops.

AUFRUFFOLGE

Das Programm wird durch folgende Handbedienung aufgerufen:
1. Drücke den Hebel "Eingabe von Hand" auf der Schreibmaschine.
2. Stelle Auswahlschalter auf "Eingabe von Hand".
3. Drücke "Füllen und Löschen".
4. Stelle Auswahlschalter auf NORMAL.
5. Drücke die START-Taste.

BEDIENUNG

1. Programmeingabe

Der Programmstreifen enthält das Programm in zwei Fassungen:
a) willkürlich verlegbar hexadezimal mit eigenem Bootstrap und
b) dezimal in Kodierungsblattformat.
Das Eingabeverfahren ist wie folgt:
a) Streifen in Schreibmaschinenleser einlegen. Man vergewissere sich, daß alle Hebel der Schreibmaschine gelöst sind.
b) Drücke E/A-Taste auf dem Rechner.
c) Stelle Wahlschalter auf "Eingabe von Hand".
d) Drücke "Lesen Start" auf der Schreibmaschine.
e) Drücke "Füllen und Lösohen" auf dem Rechner.
f) Drücke "Lesen Start" auf der Schreibmaschine.
g) Stelle Wahlschalter "Einzeloperation" (Step) auf dem Rechner.
h) Drücke "Ausführen" (Rechner)
i) wiederhole c bis h zweimal; dann fahre nach j fort.
J) Wahlschalter auf "Eingabe von Hand" stellen (Rechner)
k) Drücke "Lesen Start" (Schreibmaschine)
l) Drücke "Füllen und Löschen" (Rechner)
m) Wahlschalter auf "Einzeloperation" (Rechner)
n) Drücke "Ausführen" (Rechner)
o) Wahlschalter auf "Normal" stellen (Rechner)
p) Drücke "Start" (Rechner). Der Rest wird eingelesen, und der Rechner hält an.
q) Drücke "Eingabe von Hand" (Schreibmaschine)
r) Drücke "Start" (Rechner)
s) Die Kontrollampe der Schreibmaschine leuchtet auf und zeigt, daß der Rechner zur Eingabe bereit ist.

2. Wiederholung des Einlesens

Wenn das eingelesene Programm nach Schritt p nicht arbeiten will, oder wenn es nach dem Einlesen im Speicher zerstört wurde, kann der Bootstrap immer noch gespeichert sein. In diesem Fall kann das Programm ohne Bootstrapeingabe folgendermaßen eingelesen werden:
a) Lege den Streifen in den Leser, und zwar so, daß der Bootstrap nicht mehr gelesen wird (Bandlauf).
b) Stelle Wahlschalter auf "Eingabe von Hand" (Rechner)
c) Drücke "Lesen Start" (Schreibmaschine)
d) Drücke "Füllen und Löschen" (Rechner)
e) Stelle Wahlschalter auf "Einzeloperation" (Rechner)
f) Drücke "Ausführen" (Rechner)
g) Stelle Wahlschalter auf "Normal".
h) Drücke "Start" (Rechner). Der Streifen wird eingelesen und der Rechner hält an. Sollte der Streifen sich nicht einlesen lassen, so wiederhole man die gesamte Bootstrapprozedur.

3. Erforderliche Eingabe-/Ausgabeeinheiten.

   Soll eine andere Eingabeeinheit als die Lochstreifenschreibmaschine ausgewählt werden, so ist das Auswahlschlüsselwort (S) zu verwenden. Die erforderlichen Ausgabeeinheiten wählt das Unterprogramm selbst aus.

4. Bedeutung der Programmsprungtasten.

Schalter	Funktion
PS-32	Aus - Rechner hält vor dem Sprung an, wenn ein "Stop- und Sprung" Schlüsselwort ausgeführt wird.
	Ein - Rechner führt den "Stop- und Sprung" aus ohne anzuhalten.
PST	Aus - Bei Eingabe hexadezimaler Streifen werden die Prüfsummen verglichen.
	Ein - Bei Eingabe hexadezimaler Streifen werden die Prüfsummen nicht verglichen.

5. Fehlerstops.

Ausdruck	Bedeutung und Behebung
err	Der Rechner hat einen Fehler entdeckt. Nach der Eingabe eines hexadezimalen Füllwortes (v): a) Lege den Streifen bis zum letzten Eingabeschlüsselwort zurück. b) Drücke START (Rechner) Im Falle eines falschen oder vergessenen Wortes: a) Drücke "Eingabe von Hand" (Schreibmaschine) b) Drücke START (Rechner) c) Gebe das richtige Wort ein. d) Löse "Eingabe von Hand" (Schreibmaschine) e) Drücke START (Rechner). (Das letzte gelesene Wort enthält den Fehler).

Anmerkung: Nach einem Fehlerstop wählt das Unterprogramm die Schreibmaschine zur Eingabe aus.

BEMERKUNGEN

Das Unterprogramm kann keine Datenworte (oder Konstanten) in dezimaler Schreibweise verarbeiten.
Wenn ein Programm eingelesen wird, werden das Vorzeichen und die drei meistbedeutenden Bits eines jeden Eingabewortes untersucht und dann ein Sprung zum entsprechenden interpretierenden Teil des Unterprogrammes ausgeführt.
Die Kodierung eines Programmes kann in logisch unabhängige Gruppen zerfallen, von denen einige auch in anderen Programmen verwandt werden können; Beispiele dafür sind alle Arten von Unterprogrammen, Standard - Eingabe - und Ausgabeprogramme und Programme mathematischer Teilprobleme. Damit diese Befehlsgruppen ständig gegenwärtig sind, sollten sie auf Speicherbereiche gelegt werden, wo sie bleiben können (damit nicht zwei verschiedene Blöcke auf den gleichen Speicherbereich eingelesen werden.) Man kodiert daher jede Programmgruppe relativ zur Adresse 0000. Durch das Füllschlüsselwort (;) und den Modifikator (/) kann der Programmierer das Programm jetzt auf einen beliebigen Bereich legen, ohne daß die Beziehung der Befehle untereinander gestört würde. Befehlen einer solchen Gruppe, die nicht modifiziert werden sollen, muß ein "X" vorangehen.

Andererseits kann die Modifikation einer ganzen Befehlsgruppe vermieden werden, indem man den Modifikator Null setzt. Durch das System ist also die Verfügbarkeit des Speicherraumes nicht beschränkt.

ZUSAMMENFASSUNG

Eingabeschlüsselwort	Interpretiert als
ØTTSS	Befehl - modifizierbar.
XØTTSS	Befehl - nicht modifizierbar.
8ØØØTTSS	Negativer Befehl - modifizierbar.
8ØXØTTSS	Negativer Befehl - nicht modifizierbar.
;ØØØTTSS	Füllschlüsselwort (Start Fill) - Speichere die folgenden Worte (Modifikator ausgenommen) in aufeinanderfolgenden Zellen beginnend in TTSS.
/ØØØTTSS	Modifikator - Setze die Modifikatorzelle auf TTSS. Addiere diesen Modifikator zu allen folgenden modifizierbaren Befehlen.
.ØØØTTSS	Stop und Sprung (Stop and Transfer) - Stop (es sei denn PS-32 ist gedrückt) und Sprung nach Zelle TTSS, wenn ein Startsignal erfolgt.
+ØØØTTSS	Direktbefehl - Gestattet die unmittelbare Ausführung eines Befehls, wobei das zweite Wort als Datenwort in Frage kommt. Dieses zweite Wort muß hexadezimale Schreibweise haben.
,ØØØØØNN	Hexadezimale Worte - Die nächsten NN hexadezimalen Worte sind fortlaufend in der jeweils durch den Füllzähler definierten Zelle zu speichern. 1≤NN≤63.
VNNCHHHH	Hexadezimales Füllen - Speichere NN hexadezimale Worte beginnend in HHHH. 1≤NN≤63. Das Schlüsselwort in hexadezimale Schreibweise. Für jede Wortgruppe wird eine Prüfsumme berechnet, und, wenn PST aus ist, wird diese Prüfsumme mit der auf den Streifen nach NN Worten gestanzten Prüfsumme verglichen. Ist PST ein, so findet kein Prüfsummenvergleich statt.
AØØØØØNN	Alphanumerische Worte - Die nächsten NN Worte sind als alphanumerische 5 sechs-Bit-Zeichen lange Worte@29 in aufeinanderfolgenden Zellen zu speichern. 1≤NN≤63.

-000TTSS	Frageschlüsselwort - Bringe das Wort in TTSS in den Akkumulator. Bei Start bringe das Wort in TTSS+1 in den Akkumulator, u.s.w. Unterbrechung dieser Arbeitsweise durch Sprung von Hand nach 0000.
S000TT00	Eingabeauswahl - Benutze die Eingabeeinheit, deren Kennziffer TT ist, zur fortwährenden Eingabe durch das Unterprogramm.

Ø steht für einen beliebigen Maschinenbefehl. TT sind die Dezimalziffern für die Spuradresse, und SS sind die Dezimalziffern für die Sektoradresse. NN sind die Dezimalziffern, die die Anzahl der einzulesenden Worte angeben. HHHH ist eine Adresse in hexadezimaler Schreibweise.

DATENEINGABE 1	J2-10.0

VERWENDUNGSZWECK

J2-10.0 liest Daten in den Rechner ein zur Benutzung in einem Hauptprogramm. Die in dezimaler Schreibweise vorliegenden Daten werden eingelesen, binärisiert, zum gewünschten "q" geschiftet, und auf vorgegebene Speicherplätze gebracht. Während eines jeden Programmablaufs kann die nötige Anzahl von Dezimalzahlen umgewandelt und gespeichert werden. Alle Informationen bezüglich der Art der Umformung müssen mit den Daten eingegeben werden.

Die Dateneingabe 1 arbeitet wie ein Unterprogramm und wird mittels einer Aufruffolge des Hauptprogrammes angesprochen. Bevor ein Sprung ins Unterprogramm erfolgt, muß die Aufruffolge die Rückkehradresse im Unterprogramm speichern.
Soll nicht die Standardschreibmaschine zur Eingabe benutzt werden, so muß ein geeignetes Auswahlschlüsselwort in der Aufruffolge enthalten sein. Bei der Rückkehr ins Hauptprogramm sind alle Dezimalzahlen umgewandelt und auf den angegebenen Speicherzellen gespeichert.

SPEICHERBEDARF	4 Spuren
Benutzte Zwischenspeicher	keine

EINGABE

Zur Verarbeitung von Dezimalzahlen mittels J2-10.0 müssen folgende Informationen, jede gefolgt von einem Stopcode, in der angegebenen Reihenfolge eingegeben werden:

1. Die Anzahl von Dezimalstellen P eines jeden Datenwortes in dezimaler Form.
   Dabei gilt: $0 \leq P \leq 11$
2. Der Wert Q, der das "q" darstellt, bei welchem die Worte gespeichert werden sollen. Q besteht aus drei Zeichen einschließlich Vorzeichen. Es gilt: $-36 \leq Q \leq +46$
3. Die Anfangsadresse des Speicherbereiches in dezimaler Spur-/Sektorschreibweise.
4. Die Dezimalzahlen in Form von maximal 7 Stellen und Vorzeichen zu Beginn. Negative Zahlen müssen das Vorzeichen und die linksseitigen Nullen enthalten; bei positiven Zahlen darf beides fehlen.
5. Der Buchstabe "j", der bewirkt, daß neue P, Q und Adressenwerte eingelesen werden.

BEMERKUNG
Falls P,Q oder die Adressenfolge sich ändern, so muß ein "j" eingeführt werden, gefolgt von den gewünschten P,Q-Werten, sowie der Adresse (einschließlich derer, die gleich bleiben).

6. Der Buchstabe "w" bewirkt Rückkehr ins Hauptprogramm.

Beispiel:

```
4'+13'2318'627138'48390'j'
2'+20'3800'5134'664'-0000047'w'
```

Die Dezimalzahlen 62,7138 und 4,839 werden mit q=13 in 2318 und 2319 gespeichert. Die Zahlen 51,34, 6,64 und -,47 werden bei q=20 in 3800, 3801 und 3802 gespeichert. Dann erfolgt Rückkehr ins Hauptprogramm. Gewöhnlich wählt das Unterprogramm die Standardschreibmaschine zur Eingabe aus. Eine abgewandelte Aufruffolge zur Auswahl einer anderen Eingabeeinheit ist vorgesehen. Bei Benutzung dieser Aufruffolge muß der entsprechende Auswahlcode bei q=23 im Akku stehen, wenn der Sprung ins Unterprogramm erfolgt.

AUFRUFFOLGE

Aufruffolge 1 wählt die Standardschreibmaschine zur Eingabe aus; bei Auswahl einer anderen Eingabeeinheit benutze man Aufruffolge 2.

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
α	R	L_o+138	Rückkehradresse setzen.
+1	U	L_o	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr aus dem Unterprogramm.

Der Befehl in α speichert die Adresse (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung zum Anfang des Unterprogrammes. α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	B	[TTSS]	Bringe neuen Auswahlcode.
α	R	L_o+138	Rückkehradresse setzen.
+1	U	L_o+247	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr aus dem Unterprogramm.
.	.	.	
.	.	.	
.	.	.	
TTSS	Z	SS00	Auswahlcode für Eingabeeinheit.

Der Befehl in α-1 bringt den Auswahlcode für die neue Eingabeeinheit in den Akku. Der Befehl in α speichert die Adresse (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o+247 des Unterprogrammes.

DATENEINGABE 1	J2-10.0

Hier wird die Auswahl der Eingabeeinheit geändert und das Unterprogrammdurchlaufen, bis in L_0+138 Rückkehr nach α+2 des Hauptprogrammes erfolgt.

1. PROGRAMMEINLESEVORSCHRIFT

Der Programmstreifen enthält das Programm in zwei Arten:

a) Willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

2. ZEITBEDARF

Das Programm liest und binärisiert 35 fünfstellige Zahlen pro Minute.

3. ERFORDERLICHE EINGABE - AUSGABEEINHEITEN

Die Standardschreibmaschine wird zur Eingabe ausgewählt. Eine Ausgabeeinheit ist nicht erforderlich. Die Auswahl kann durch die entsprechende Aufruffolge geändert werden (siehe oben).

4. ÜBERLAUF

Überlauf bei der Eingabe wird nicht beachtet. Vor der Rückkehr ins Hauptprogramm wird Überlauf zurückgesetzt.

DATENEINGABE 2	J2-10.1

VERWENDUNGSZWECK

Das Programm J2-10.1 übernimmt Daten in den Rechner zur Verwendung durch ein Hauptprogramm. Die dezimalen Daten werden gelesen, binärisiert, zum gewünschten "q" geschiftet und, wenn mehr als eine Zahl auftritt, auf den vorgegebenen Zellen gespeichert. Tritt nur eine Zahl auf, so bleibt sie im Akkumulator, nach dem sie zum entsprechenden "q" geschiftet wurde. Von einem einzigen Programmaufruf her können beliebig viele Dezimalzahlen umgewandelt und gespeichert werden. Alle Informationen bezüglich der Umwandlung muß das Hauptprogramm liefern.
Das Programm arbeitet als Unterprogramm, das durch Aufruffolge des Hauptprogrammes aufgerufen wird. Vor dem Sprung in das Unterprogramm muß die Aufruffolge die Anfangsadresse der Umwandlungskontrolliste und die Rückkehradresse im Unterprogramm speichern. Bei der Rückkehr in das Hauptprogramm sind alle Daten umgewandelt und gespeichert, bzw. stehen im Akkumulator.

SPEICHERBEDARF 4 Spuren, 48 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die zu verarbeitenden Daten müssen als 7-stellige Dezimalzahlen plus Vorzeichen eingegeben werden. Bei positiven Zahlen dürfen Vorzeichen und linksseitige Nullen fehlen. Vor dem Sprung in das Unterprogramm muß das Hauptprogramm eine Umwandlungskontrolliste gespeichert haben. Die Liste belegt aufeinanderfolgende Zellen und sieht so aus:

1. N, die Anzahl von Worten in Spur-/Sektorschreibweise; d.h. mod 64.
2. L, Anfangsadresse für die Daten in dezimaler Spur-/Sektorschreibweise.
3. In einem einzigen Wort:
   a) Q, der Wert "q" als dezimale Spuradresse; für Q gilt: $-47 \leq Q \leq +36$.
   b) P, die Anzahl der Dezimalstellen als dezimale Sekteradresse; es gilt: $0 \leq P \leq 11$.

Wird "N" in einen negativen Befehl eingesetzt, so wird die beim vorigen Durchlauf gesetzte Zahl beibehalten. Wird "L" in einen negativen Befehl eingeschlossen, so wird die Adresse des vorigen Durchlaufes für die Speicherung benutzt. Alle Befehlsbits in den Befehlen mit N und L müssen "0" sein. Wird Q in einen negativen Befehl eingeschlossen, so ist sein Wert negativ. Wenn das erste Befehlsbit des Befehles, der Q und P enthält "1" ist (d.h. 1@12), so werden die beim vorigen Durchlauf gesetzten Werte benutzt.

Anmerkung: Wenn N Null ist, wird eine unbegrenzte Zahl von Datenworten gelesen, und zwar solange bis ein "w" auftritt.

Ist "N" gleich 1, so bleibt das gelesene Wort beim spezifizierten "q" im Akkumulator und wird nicht gespeichert.
Zusätzlich muß ein Befehl mit der Anfangsadresse der Umwandlungsliste gespeichert sein. Ist dieser Befehl negativ, so arbeitet das Unterprogramm mit dem entsprechenden Wert des vorigen Durchlaufes. Ist das erste Befehlsbit eine "1", so setzt das Unterprogramm die neuen Kontrollwerte ein, ohne etwas einzulesen. Daher brauchen die Vorbereitungen nur einmal getroffen zu werden, wenn die gleichen Werte für eine ganze Folge von Operationen gelten. Dieser Befehl muß im Akkumulator stehen, wenn der Sprung in das Unterprogramm erfolgt.
Soll die Eingabeeinheit gewechselt werden, so ist eine andere Aufruffolge zu verwenden (gewöhnlich wählt das Unterprogramm die Standardschreibmaschine aus). Dazu ist ein vierter Befehl in der Kontrolliste notwendig. Dieser Befehl muß den Auswahlcode für die gewünschte Einheit als dezimale Spuradresse enthalten. Dieser Auswahlcode bleibt wirksam, bis er durch einen anderen ersetzt wird.

Im folgenden werden Beispiele für die Anwendung der Kontrollisten gebracht (Adressen in Klammern sind im Unterprogramm relativ).

Beispiel 1

Speicherplatz	Befehl	Adresse	
0500	000Z	2430	Anfangsadresse der Kontrolliste
0610	000B	0500	Aufruffolge
0611	000R	(0442)	
0612	000U	(0445)	
0613	u.s.w.		
2430	000Z	0126	Kontrolliste
2431	000Z	4200	
2432	000Z	1205	

Dieses Beispiel veranlaßt das Unterprogramm 90 dezimale Worte ($126_{64}=90_{10}$) zu lesen, zu binärisieren und in aufeinanderfolgenden Speicherzellen beginnend in 4200 bei q=12 mit P=5 zu speichern; dann erfolgt Rückkehr nach 0613 des Hauptprogrammes.

Beispiel 2

Speicherplatz	Befehl	Adresse	
0501	000Z	2433	Anfangsadresse der Kontrolliste
0729	000B	0501	Aufruffolge
0730	000R	(0442)	
0731	000U	(0445)	
0732	u.s.w.		
2433	000Z	0000	Kontrolliste
2434	800Z	0000	
2435	800Z	0109	

Dieses Beispiel veranlaßt das Unterprogramm, solange Dezimalzahlen zu lesen, bis ein Schlüsselwort "w" für das Blockende auftritt. Die Daten werden binärisiert und in aufeinanderfolgenden Zellen, beginnend mit der beim vorigen Aufruf festgelegten Adresse, gespeichert bei q=-01 mit P=9; dann erfolgt Rückkehr nach 0732 des Hauptprogrammes.

Beispiel 3

Speicherplatz	Befehl	Adresse	
0502	000Z	2436	Anfangsadresse der Kontrolliste
1925	000B	0502	Aufruffolge
1926	000R	(0442)	
1927	000U	(0445)	
1928	u.s.w.		
2436	000Z	0001	Kontrolliste
2437	000Z	1340	
2438	000P	0000	

Dieses Beispiel veranlaßt das Unterprogramm, ein dezimales Wort zu lesen, zu binärisieren und im Akkumulator bei dem "q" und "P" zu speichern, die beim vorigen Aufruf gesetzt wurden; dann erfolgt Rückkehr nach 1928. Das Wort wird nicht in 1340 gespeichert. ("L" muß angegeben werden und kann eine beliebige Adresse sein).

Beispiel 4

Speicherplatz	Befehl	Adresse	
0503	800Z	0000	
3618	000B	0503	Aufruffolge
3619	000R	(0442)	
3620	000U	(0445)	
3621	u.s.w.		

Dieses Beispiel bewirkt, daß das Unterprogramm die gesamte Kontrolliste des vorherigen Durchlaufes benutzt.

Beispiel 5

Speicherplatz	Befehl	Adresse	
0504	000P	2439	Anfangsadresse der Kontrolliste
1231	000B	0504	Aufruffolge
1232	000R	(0442)	
1233	000U	(0413)	
1234	u.s.w.		
2439	800Z	0000	Kontrolliste
2440	000Z	1500	
2441	000Z	2700	
2442	000Z	0400	

Das Beispiel veranlaßt das Unterprogramm, sich selbst vorzubereiten: die Einheit entsprechend dem Code 04 auszuwählen und soviele Daten zu lesen, wie für den vorigen Durchlauf festgesetzt waren, die Daten zu binärisieren und sie in aufeinanderfolgenden Zellen, beginnend in 1500 bei q=27 mit P=0 zu speichern; dann würde Rückkehr nach 1234 erfolgen. Bei dieser Art Eingang wird überhaupt nichts gelesen (das erste Befehlsbit des Befehles, der die Adresse der Kontrolliste enthält, ist eine "1").

AUFRUFFOLGE

Aufruffolge 1 sollte benutzt werden, wenn die Eingabeeinheit nicht gewechselt wird; Aufruffolge 2, wenn sie gewechselt wird.

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe den Befehl, der die Anfangsadresse oder Kontrolliste enthält.
✓	R	L_o+442	Rückkehradresse setzen.
+1	U	L_o+445	Eingang in das Unterprogramm.
+2	u.s.w.		Hauptprogramm.

Die Eingabeeinheit wird nicht geändert (gewöhnlich ist das die Standardschreibmaschine).

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Anfangsadresse der Kontrolliste.
✓	R	L_o+442	Rückkehradresse setzen.
+1	U	L_o+413	Eingang in das Unterprogramm.
+2	u.s.w.		Hauptprogramm.

DATENEINGABE 2	J2-10.1

Als Eingabeeinheit wird die dem Auswahlcode in der Kontrolliste entsprechende genommen.

BEDIENUNG

1. Zeitbedarf

   Das Programm liest und binärisiert 40 fünfziffrige Zahlen pro Minute.

2. Programmeingabe

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten

   Das Unterprogramm wählt die Standardschreibmaschine zur Eingabe aus, wenn nicht für eine andere Auswahlmöglichkeit durch Aufruffolge 2 gesorgt wurde. Ausgabeeinheiten sind nicht erforderlich.

4. Überlauf

   Wird beim Eingang nicht beachtet und beim Ausgang zurückgesetzt.

ANMERKUNG

Aus Genauigkeitsgründen sollte das kleinstmögliche Q benutzt werden.

DATENEINGABE 3	J2-10.2

VERWENDUNGSZWECK

Das Programm J2-10.2 binärisiert, oder liest und binärisiert eine siebenstellige Dezimalzahl mit Vorzeichen.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Vor dem Sprung in das Unterprogramm muß die Aufruffolge die Rückkehradresse in das Unterprogramm speichern. Ebenso muß in manchen Fällen die zu binärisierende ganze Zahl im Akkumulator stehen. Bei der Rückkehr in das Hauptprogramm steht die binärisierte Zahl@30 im Akkumulator.

SPEICHERBEDARF — 1 Spur, 9 Sektoren; (man lasse das Programm in Sektor 00 einer beliebigen Spur beginnen).

benutzte Zwischenspeicher — keine

EINGABE

Die Eingabevariable ist die Dezimalzahl N, die durch das Unterprogramm eingelesen wird, oder vom Hauptprogramm geliefert wird.
Bei Eingabe durch das Unterprogramm darf N aus sieben Dezimalstellen plus Vorzeichen bestehen. Bei positiven Zahlen darf das Vorzeichen fehlen. Bei negativen Zahlen steht das Vorzeichen vor den sieben Dezimalstellen.
Wird die Zahl vom Hauptprogramm geliefert, so soll sie@30 oder@31 im Akkumulator stehen, je nachdem, welche Aufruffolge benutzt wird.
Das Vorzeichen der Zahl wird angegeben durch
Bitposition 2, wenn die Zahl@30 steht,
Bitposition 3, wenn die Zahl@31 steht.
Eine "1" in der entsprechenden Bitposition zeigt eine negative Zahl an, eine "0" eine positive Zahl.

AUFRUFFOLGEN

Aufruffolge 1 wird benutzt, wenn N vom Unterprogramm eingelesen wird.
Wird N vom Hauptprogramm geliefert, so kommt in Frage:
Aufruffolge 2, bei einer Zahl N@31
Aufruffolge 3, bei einer Zahl N@30

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
α	R	L_o + 44	Rückkehradresse setzen
α+1	U	L_o	Eingang in das Unterprogramm
α+2	u.s.w.		Hauptprogramm

Der Befehl in α speichert die Rückkehradresse in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. Der Befehl in α+2 ist der Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird. N wird vom Unterprogramm eingelesen.

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	I	[ ]	Lies N in den Akkumulator
α	R	L_o + 44	Rückkehradresse setzen
+1	U	L_o + 2	Eingang in das Unterprogramm
+2	u.s.w.		Rückkehr in das Hauptprogramm

Der Befehl in α-1 liest N in den Akkumulator. Der Befehl in α speichert die Rückkehradresse (α+2) in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o+2 des Unterprogrammes. Zelle α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird. Man beachte, daß eine "1" (negativ) oder eine "0" (positiv) in Bitposition 3 des Akkumulators das Vorzeichen von N bestimmt, und daß N bei q=31 steht.

Aufruffolge 3

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bring N in den Akkumulator
α	R	L_o + 44	Rückkehradresse setzen
+1	U	L_o + 58	Eingang in das Unterprogramm
+2	u.s.w.		Rückkehr in das Hauptprogramm

Der Befehl in α-1 bringt N aus dem Speicher in den Akkumulator. Der Befehl in α setzt die Rückkehradresse (α+2) in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung in das Unterprogramm. In α+2 steht der Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird. Man beachte, daß eine "1" (negativ) oder eine "0" (positiv) in Bitposition 2 des Akkumulators das Vorzeichen von N bestimmt, und daß N im Akkumulator @30 steht.

AUSGABE

Bei der Rückkehr in das Hauptprogramm steht die binärisierte Zahl N im Akkumulator bei q=30.

BEDIENUNG

1. Zeitbedarf

   Das Programm erfordert 1.1 Sekunden zum Lesen und Binärisieren einer fünfstelligen Zahl.

2. Programmeingabe

Der Streifen enthält das Programm in zwei Fassungen:
a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten

Bei Aufruffolge 1 wählt das Unterprogramm die Standardschreibmaschine aus. Bei Aufruffolge 2 oder 3 ist die Auswahl eine Funktion des Hauptprogrammes. Ausgabeeinheiten sind nicht erforderlich.

4. Überlauf

Überlauf wird vom Programm weder beachtet noch verändert.

DATENEINGABE 4	J2-10.3

VERWENDUNGSZWECK

Das Programm J2-10.3 übernimmt Daten in den Rechner zur Benutzung durch ein Hauptprogramm. Es liest eine vorgegebene Anzahl von dezimalen Datenworten, binärisiert sie, richtet sie nach einem angegebenen dezimalen Punkt aus und speichert sie in aufeinanderfolgenden Zellen bei vorgegebenem "q". Während eines Programmlaufes können beliebig viele Zahlen umgewandelt und gespeichert werden. Alle Informationen bezüglich der Umwandlung muß das Hauptprogramm liefern.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Die Aufruffolge muß die Anfangsadresse der Umwandlungskontrolliste sowie die Rückkehradresse liefern. Bei der Rückkehr in das Hauptprogramm sind alle Daten binärisiert und gespeichert.

SPEICHERBEDARF 3 Spuren, 52 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die zu binärisierenden Dezimalzahlen können aus algebraischen Vorzeichen und sieben Ziffern bestehen. Eine negative Zahl setzt sich zusammen aus ihrem Vorzeichen und sieben Ziffern; bei positiven Zahlen dürfen Vorzeichen und die linksseitigen Nullen fehlen.
Die Anzahl der einzulesenden Dezimalzahlen N ist nur durch den verfügbaren Speicherraum beschränkt. Für die Anzahl der Dezimalstellen P muß gelten: $0 \leq P \leq 11$. Für Z, den Wert von "q" muß gelten: $-36 \leq Z \leq +47$.
Die Kontrolliste für die Umwandlung, die in aufeinanderfolgenden Zellen steht, sieht folgendermaßen aus:

1. In einem einzigen Wort:
   a) Q als zweiziffrige Dezimalzahl im Spuradressteil.
   b) P als zweiziffrige Dezimalzahl im Sektoradressteil.
2. N in Spur-/Sektorschreibweise; d.h., mod 64.
3. Die Anfangsadresse für die Daten in dezimaler Spur-/Sektorschreibweise.

Q ist negativ, wenn der Befehl, in dem es enthalten ist, negativ ist. Alle anderen Befehle der Liste müssen positiv sein.
Zusätzlich muß ein Befehl mit der Anfangsadresse der Kontrolliste vorhanden sein. Dieser Befehl muß im Akkumulator stehen, wenn der Sprung in das Unterprogramm erfolgt.
Soll die Auswahl der Eingabeeinheit geändert werden, so ist eine spezielle Aufruffolge zu benutzen (das Unterprogramm wählt gewöhnlich die Standardschreibmaschine aus). Dazu ist ein vierter Befehl in der Kontrolliste notwendig. Dieser Befehl enthält als dezimale Spuradresse den erwünschten Auswahlcode. Der Auswahlcode bleibt solange wirksam, bis er durch einen anderen ersetzt wird.

AUFRUFFOLGE

Soll die Standardschreibmaschine als Eingabeeinheit benutzt werden, so ist Aufruffolge 1 zu benutzen. Aufruffolge 2 ist für variable Auswahlmöglichkeit gedacht.

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe die Anfangsadresse der Kontrolliste.
α	R	L_o+351	Rückkehradresse setzen.
+1	U	L_o+308	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α-1 bringt den Befehl mit der Anfangsadresse der Kontrolliste in den Akkumulator. Der Befehl in α speichert die Rückkehradresse (α+2) in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o+308 des Unterprogrammes, wobei L_o die Anfangsadresse des Unterprogrammes ist. α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe die Anfangsadresse der Kontrolliste.
α	R	L_o+351	Rückkehradresse setzen.
+1	U	L_o+300	Sprung in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Die Aufruffolge 2 ist identisch mit Aufruffolge 1, bis auf den Befehl in α+1. Dieser Befehl bewirkt einen Sprung nach L_o+300, sodaß die gewünschte Auswahl getroffen wird (in Abhängigkeit vom Auswahlcode der Kontrolliste).

BEDIENUNG

1. Zeitbedarf

   Das Programm liest und binärisiert 40 fünfziffrige Zahlen pro Minute.

2. Programmeingabe

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten

   Das Unterprogramm wählt die Standardschreibmaschine zur Eingabe aus, wenn nicht für eine andere Auswahlmöglichkeit durch Aufruffolge 2 gesorgt wurde. Ausgabeeinheiten sind nicht erforderlich.

4. Überlauf

   Wird beim Eingang nicht beachtet, vor der Rückkehr in das Hauptprogramm jedoch zurückgesetzt.

BEISPIELE

Im folgenden werden Beispiele für Umwandlungskontrollisten aufgeführt.

Beispiel 1

Speicherplatz	Befehl	Adresse	
0500	Z	2420	Angabe der Anfangsadresse der Kontrolliste.
0532	B	0500	} Aufruffolge
0533	R	L_o+0351	
0534	U	L_o+0308	
0535	u.s.w.		
2420	Z	1304	Q und P
2421	Z	0146	Anzahl der zu lesenden Worte.
2422	Z	1500	Anfangsadresse für die Daten.

Dieses Beispiel veranlaßt das Unterprogramm 110 Dezimalzahlen ($146_{64} = 110_{10}$) einzulesen, zu binärisieren, für P=4 einzurichten und in aufeinanderfolgenden Zellen in 1500 beginnend bei "q"=13 zu speichern, dann erfolgt Rückkehr nach 0535 des Hauptprogrammes.

Beispiel 2

Speicherplatz	Befehl	Adresse	
0501	Z	2423	Angabe der Anfangsadresse der Kontrolliste.
0636	B	0501	} Aufruffolge
0637	R	L_o+0351	
0638	U	L_o+0300	
0639	u.s.w.		
2423	800Z	0108	Q und P
2424	Z	0019	Anzahl der zu lesenden Worte.
2425	Z	1820	Anfangadresse für die Daten.
2426	Z	0800	Auswahlcode für die Eingabeeinheit.

Dieses Beispiel veranlaßt das Unterprogramm 19 Dezimalzahlen zu lesen, sie zu binärisieren, für P=8 einzurichten und in aufeinanderfolgenden Zellen beginnend in 1820 bei "q"=-01 zu speichern. Dann erfolgt Rückkehr nach 0639 des Hauptprogrammes. Die Eingabe erfolgt über die dem Code 08 entsprechende Einheit.

BEMERKUNG

Das kleinstmögliche "q" sollte zur Aufrechterhaltung der Genauigkeit benutzt werden. Das Programm hält nicht an, wenn eine Zahl beim beabsichtigten "q" nicht gespeichert werden kann, sondern arbeitet mit einem bedeutungslosen Wert weiter.

DATENEINGABE 5	J2-10.4

VERWENDUNGSZWECK

Das Programm J2-10.4 übernimmt Daten in den Rechner zur Benutzung durch ein Hauptprogramm. Es liest eine unbestimmte Anzahl von dezimalen Worten, binärisiert sie, richtet sie nach dem angegebenen dezimalen Punkt ein und speichert sie in aufeinanderfolgenden Zellen bei einem gegebenen "q". Während eines Programmlaufes können beliebig viele Dezimalzahlen umgewandelt und gespeichert werden. Alle Daten, die die Umwandlung steuern, müssen vom Hauptprogramm geliefert werden.
Dateneingabe 5 arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Die Aufruffolge muß dem Unterprogramm die Kontrolliste für die Umwandlung, sowie die Rückkehradresse liefern. Bei der Rückkehr in das Hauptprogramm sind alle Daten umgewandelt und gespeichert.

SPEICHERBEDARF 3 Spuren, 48 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Dateneingabe 5 kann Dezimalzahlen bestehend aus algebraischen Vorzeichen und sieben Ziffern, binärisieren. Eine negative Zahl muß bestehen aus ihrem Vorzeichen und sieben Ziffern; Vorzeichen und linksseitige Nullen dürfen bei positiven Zahlen fehlen.
Für die Anzahl der Dezimalstellen P muß gelten: $0 \leq P \leq 11$. Für Q, den Wert von "q" muß gelten: $-36 \leq Q \leq +47$.
Die Kontrolliste für die Umwandlung, die vor dem Eingang in das Unterprogramm in aufeinanderfolgenden Zellen gespeichert sein muß, setzt sich folgendermaßen zusammen:

1. In einem einzigen Wort:
   a) Q als zwei Dezimalziffern im Spuradressteil.
   b) P als zwei Dezimalziffern im Sektoradressteil.
2. Anfangsadresse der Daten in dezimaler Spur-/Sektorschreibweise.

Steht Q in einem negativen Befehl, so ist es negativ. Alle anderen Befehle der Liste müssen positiv sein.
Zusätzlich muß noch ein Befehl mit der Anfangsadresse der Kontrolliste vorhanden sein. Dieser Befehl muß im Akkumulator stehen, wenn der Sprung in das Unterprogramm erfolgt.
Die Eingabe der Dezimalzahlen wird beendet durch das spezielle Schlüsselwort w'.
Wenn die Eingabeeinheit geändert werden soll, kann eine spezielle Aufruffolge benutzt werden (das Unterprogramm wählt gewöhnlich die Standardschreibmaschine aus). Dieses erfordert einen zusätzlichen Befehl in der Kontrolliste, der den beiden anderen folgt. Dieser Befehl muß als dezimale Spuradresse den Auswahlcode für die gewünschte Einheit enthalten. Dieser Auswahlcode bleibt wirksam, bis er durch einen anderen Auswahlcode ersetzt wird.

AUFRUFFOLGE

Die Aufruffolge 1 sollte benutzt werden, wenn die Standardschreibmaschine als Eingabeeinheit in Frage kommt; im anderen Falle benutze man Aufruffolge 2, um eine spezielle Einheit auszuwählen.

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Anfangsadresse der Kontrolliste.
α	R	L_o+347	Rückkehradresse setzen.
+1	U	L_o+308	Sprung in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α-1 bringt den Befehl mit der Anfangsadresse der Kontrolliste in den Akkumulator. Der Befehl in α speichert die Rückkehradresse (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o+308 des Unterprogrammes, wobei L_o die Anfangsadresse des Unterprogrammes ist. α+2 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird.

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe Anfangsadresse der Kontrolliste.
α	R	L_o+347	Rückkehradresse setzen.
+1	U	L_o+300	Eingang in das Unterprogramm.
+2	u.s.w.		Rückkehr in das Hauptprogramm.

Aufruffolge 2 unterscheidet sich von Aufruffolge 1 nur durch den Befehl in α+1. Dieser Befehl bewirkt einen Sprung nach L_o+300 des Unterprogrammes, sodaß in Abhängigkeit vom Auswahlcode der Kontrolliste die entsprechende Einheit benutzt wird.

BEDIENUNG

1. Zeitbedarf

   Das Programm liest und binärisiert 40 fünfziffrige Zahlen pro Minute.

2. Programmeingabe

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten

   Das Unterprogramm wählt die Standardschreibmaschine für die Eingabe aus. Diese Auswahl kann durch eine andere Aufruffolge geändert werden (siehe Aufruffolge). Eine Ausgabeeinheit ist nicht erforderlich.

4. Überlauf

   Wird beim Eingang nicht beachtet und ist beim Ausgang zurückgesetzt.

BEISPIELE

Die folgenden Bespiele illustrieren die Anwendung der Kontrolliste für die Umwandlungen.

Beispiel 1

Speicherplatz	Befehl	Adresse	
1700	Z	2900	Angabe der Anfangsadresse der Kontrolliste.
2000	B	1700	Aufruffolge
2001	R	L_o^o+0347	
2002	U	L_o^o+0308	
2003	u.s.w.		
2900	Z	0100	Q und P
2901	Z	1500	Anfangsadresse der Daten.

Dieses Beispiel veranlaßt das Unterprogramm, eine unbestimmte Anzahl von Daten zu lesen, sie zu binärisieren, sie für ein P von Null einzurichten und sie in aufeinanderfolgenden Zellen, beginnend in 1500 bei q=1 zu speichern; dann erfolgt Rückkehr nach 2003 des Hauptprogrammes.

Beispiel 2

Speicherplatz	Befehl	Adresse	
1701	Z	2902	Angabe der Anfangsadresse der Kontrolliste.
3000	B	1701	Aufruffolge
3001	R	L_o^o+0347	
3002	U	L_o^o+0300	
3003	u.s.w.		
2902	800Z	0209	Q und P
2903	Z	1200	Anfangsadresse der Daten.
2904	Z	0400	Auswahlcode für die Eingabeeinheit.

Dieses Beispiel veranlaßt das Unterprogramm, eine unbestimmte Anzahl von Datenworten über die dem Auswahlcode 04 entsprechende Einheit zu lesen, zu binärisieren, für P=9 einzurichten und in aufeinanderfolgenden Speicherzellen, beginnend in 1200 bei q=-02, zu speichern; dann erfolgt Rückkehr in das Hauptprogramm nach 3003.

BEMERKUNG

Das kleinstmögliche "q" sollte gewählt werden, um die Genauigkeit der Daten aufrecht zu erhalten. Das Programm hält nicht an, wenn eine Zahl beim angegebenen "q" nicht gespeichert werden kann, sondern arbeitet mit einem bedeutungslosen Wert weiter.

DATENAUSGABE 1	J3-10.0

VERWENDUNGSZWECK

Das Programm J3-10.0 verwandelt Dualzahlen, die im Speicher stehen, in Dezimalzahlen, einschließlich Vorzeichen und Kommastelle. Das "q" der Dualzahl kann zwischen Null und 30 liegen. Die Anzahl der Dezimalstellen ist begrenzt auf

$$\frac{30-\text{"q"}}{3}$$

Maximal werden 10 Stellen ausgegeben. Linksseitige Nullen werden unterdrückt. Die "Datenausgabe" arbeitet wie ein Unterprogramm und wird durch eine Aufruffolge des Hauptprogrammes angesprochen. Vor dem Sprung ins Unterprogramm muß die auszugebende Dualzahl im Akkumulator stehen. Das erforderliche "q" der Dualzahl muß in der dem Sprungbefehl folgenden Speicherzelle stehen. Bei der Rückkehr ins Hauptprogramm ist das dezimale Äquivalent der Dualzahl bereits ausgegeben. Das Hauptprogramm kann so geschrieben werden, daß die "Datenausgabe 1" wiederholt angesprochen wird. Auf diese Weise kann eine ganze Liste von Dezimalzahlen ausgegeben werden.

SPEICHERBEDARF 4 Spuren

BENUTZTE ZWISCHENSPEICHER keine

EINGABE

Die Eingabevariable für "Datenausgabe 1" sind:

1. Die Dualzahl N, deren dezimaler Wert gebildet werden soll,
2. Der Wert Q, der das "q" der Dualzahl darstellt.

Beim Sprung ins Unterprogramm muß N im Akkumulator stehen und Q in der dem Sprungbefehl folgenden Zelle.

N darf eine beliebige Dualzahl sein, und Q muß eine positive Zahl sein, die nie größer als 30 ist. Q kann eingegeben werden als Sektorteil eines Z-Befehles oder als hexadezimale Konstante @ 29.

Die Anzahl (F) der Stellen hinter dem Komma wird durch den Wert von Q entsprechend der Formel

$$F = \frac{30-q}{3}$$

bestimmt. (Der Wert von $F = \frac{30-q}{3}$ wird auf eine ganze Zahl abgerundet)

Duales Komma (q)	Stellen hinter dem Komma	Stellen vor dem Komma
30	0	*
28 - 29	0	9
25 - 27	1	8
22 - 24	2	7
19 - 21	3	6
16 - 18	4	5
13 - 15	5	4
10 - 12	6	3
7 - 9	7	2
4 - 66	8	1
3 - 1	9	1
0	9	0

*Für q = 30 werden in Abhängigkeit von der Größe der Zahl 9 oder 10 Stellen geschrieben.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
α	R	$L_o + 4$	Speichere die Adresse von Q im Unterprogramm
$\alpha + 1$	U	L_o	Eingang ins Unterprogramm
$\alpha + 2$	XZ	00qq	Q @ 29 (als dezimale Sektoradresse)
$\alpha + 3$	usw.		Rückkehr ins Hauptprogramm

In der gezeigten Aufruffolge bringt der Befehl auf Speicherplatz α die Adresse (α+2) des Q enthaltenden Z-Befehles in das Operandenadressfeld des Befehles in Zelle L_o+4. Dabei ist L_o die erste Zelle des Unterprogrammes. Der Befehl in α+1 bewirkt einen Sprung nach L_o. Die Zelle α+2 enthält den Wert von Q @ 29. Die Zelle α+3 enthält den ersten Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes ausgeführt wird.

Anmerkung

Vor der Aufruffolge muß N in den Akkumulator gebracht werden.

AUSGABE

Jedesmal, wenn ein Sprung aus dem Hauptprogramm nach "Datenausgabe 1" erfolgt, wird die im Akkumulator enthaltene Zahl N entsprechend der Größe von Q umgeformt und als Dezimalzahl ausgegeben. Um eine möglichst genaue Ausgabe zu erhalten, soll der kleinstmögliche Wert von Q benutzt werden. Wird dies beachtet, so beträgt der Fehler $\pm$ 1 auf der letzten Dezimalstelle.

Der Rechner wählt die Schreibmaschine zur Ausgabe aus. Maximal werden 10 Stellen ausgegeben. Linksseitige Nullen werden unterdrückt. Jeder gedruckten Zahl folgt ein Tabulatorsprung.

BENUTZUNGSANLEITUNG

1. Rechenzeit

   ca. 30 Zahlen pro Minute über Flexowriter

2. Genauigkeit

   Die Genauigkeit der ausgegebenen Zahl hängt unmittelbar vom Q-Wer

3. Eingabe des Programmes

Der Streifen enthält das Programm in zwei Formen:

a) willkürlich verlegbar, hexadezimal, einzugeben mittels J1-10.0

b) willkürlich verlegbar, dezimal, im Kodierungsblattformat

4. Erforderliche Eingabe/Ausgabe-Einheiten

Eine Eingabeeinheit wird nicht benutzt. Das Unterprogramm wählt die Schreibmaschine zur Ausgabe an.

5. Überlauf

Beim Sprung in das Unterprogramm wird ein erfolgter Überlauf nicht beachtet. Überlauf ist automatisch ausgeschaltet (zurückgesetzt), wenn die Rückkehr in das Hauptprogramm erfolgt.

6. Fehlerstops

Speicherplatz	Bedeutung und Beseitigen der Ursache
L_o + 361	Q ist negativ oder größer als 30. Nach Drücken der Start-Taste wird das Unterprogramm verlassen, ohne daß Ausgabe erfolgt.

BEMERKUNGEN

Möchte man eine andere Ausgabeeinheit als die Standard-Schreibmaschine benutzen, so ändere man die folgenden Druckbefehle des Unterprogrammes.

Speicherplatz	Befehl
L_o + 35	XP[ ]
+ 37	XP[ ]
+ 54	XP[ ]
+ 101	XP[ ]
+ 119	XP[ ]
+ 136	XP[ ]
+ 209	XP[ ]
+ 211	XP[ ]
+ 235	80XP[ ]
+ 238	80XP[ ]
+ 241	80XP[ ]
+ 244	80XP[ ]
+ 247	80XP[ ]
+ 250	80XP[ ]
+ 253	80XP[ ]
+ 256	80XP[ ]
+ 259	80XP[ ]

Gewöhnlich werden Brüche nicht gerundet. Es ist jedoch möglich,

einen Additionsbefehl in L_o + 128 zu diesem Zweck einzufügen. Die Wirkung des hinzuaddierten Wertes hängt vom "q" ab, bei dem die Zahl gedruckt werden soll. Runden kann auch (ohne Addition) erfolgen, indem man den Wert der Konstanten in L_o + 352 verkleinert. Auch hier hängt die Wirkung vom "q" der zu druckenden Zahl ab. Ganzzahlige Ausgabe wird gerundet, wenn das "q" 28 oder 29 beträgt, und ist exakt bei q = 30.

DATENAUSGABE 2 | J3-10.1

VERWENDUNGSZWECK

Das Programm J3-10.1 druckt den Inhalt aufeinanderfolgender Zellen in dezimaler Schreibweise. Eine Zahl umfaßt 8 Ziffern plus Vorzeichen und dezimalen Punkt. Die zu druckenden Zahlen müssen ganze Zahlen bei q = 30 sein. Bei positiven Zahlen werden alle linksseitigen Nullen und das Pluszeichen durch Leertasten ersetzt. Der Punkt wird nur gedruckt, wenn Stellen hinter dem Komma angegeben sind. Worte, die das gleiche Format haben, können in einer Gruppe zusammengefaßt werden. Aufeinanderfolgende Zahlen bilden einen Block. Gleichartige Worte bilden eine Gruppe. Es sind beliebig viele Blöcke und innerhalb der Blöcke beliebig viele Gruppen möglich.
"Datenausgabe 2" arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Die Aufruffolge muß liefern:

1. die Gesamtzahl der auszugebenden Worte,
2. die Anzahl von aufeinanderfolgenden Worten in jeder Gruppe
3. die Feldbreite einer jeden Zahl
4. die Zahl der Kommastellen eines jeden Wortes.

Bei der Rückkehr in das Hauptprogramm sind die angegebenen Ziffern ausgedruckt.

SPEICHERBEDARF

5 Spuren, 48 Sektoren, beginnend mit Sektor 00 einer beliebigen Spur. Keine Zwischenspeicher.

EINGABE

Die Eingabevariablen für "Datenausgabe 2" bestehen aus zwei Sorten von Schlüsselworten:

N_t Blockangaben und N_i Gruppenangaben.

Diese Ausgabeschlüsselworte müssen von der Aufruffolge in der folgenden Reihenfolge geliefert werden:

N_t Dieses Schlüsselwort muß negativ sein. Es enthält die Gesamtzahl der aufeinanderfolgenden auszugebenden Worte bei q = 15 und die Zelle des ersten Wortes bei q = 29.

N_i Dieses Schlüsselwort muß positiv sein. Es enthält folgende Information:

(N_i) Die Anzahl aufeinanderfolgender Worte, die gleiches P und D erfordern. N_i steht im Operationsteil des Wortes.

(W_i) Die Feldbreite einer jeden Zahl oder Gruppe. W_i steht im Spurteil des Wortes.

(D_i) Die Anzahl von Kommastellen in jedem Gruppenwort. D_i steht im Sektorteil des Wortes.

Die zu druckenden Zahlen müssen @ 30 gespeichert und kleiner als $\pm$ $1x10^7$ sein. Die Grenzen für N_t, N_i, W_i und D_i sind:

$8 \geq D_i \geq 0$

$128 \geq W_i \geq (2 + D_i)$ wobei genügend Raum für Punkt und Vorzeichen gelassen wird.

$15 \geq (N_i \text{ und } N_t) \geq 1$

$\sum N_i = N_t$ für jeden Block

Für Zahlen ohne Kommastellen braucht kein dezimaler Punkt einkalkuliert zu werden. Werden die erforderlichen Schranken für W_i und D_i nicht erfüllt, oder umfaßt die angegebene Feldbreite nicht die zu druckende Zahl, so wird die Zahl falsch ausgegeben. Liegen die Werte N_t und N_i zwischen 10 und 15, so können sie durch die hexadezimale Zeichen F, G, J, K, Q und W angegeben werden.

AUFRUFFOLGE

Zwei Aufruffolgen sind vorgesehen. Bei Aufruffolge 1 wählt das Unterprogramm die Schreibmaschine als Ausgabeeinheit aus; Aufruffolge 2 wird nur benutzt, wenn eine andere Ausgabeeinheit in Frage kommt oder wenn die Ausgabeeinheit gewechselt werden soll.

Speicherplatz	Befehl	Adresse	
α	R	L_o + 22	Adresse des ersten Schlüsselwortes ins Unterprogramm speichern
+ 1	U	L_o	Eingang ins Unterprogra
+ 2	80X(N_t)	ttss	N_t @ 15 und Anfangsadresse der Ausgabegrößen
+ 3	(N_1)	W_1D_1	Formatangaben
+ 4	(N_2)	W_2D_2	Formatangaben
.			
.			
.			
+ n	N_i	W_iD_i	
+ n + 1			Beginne mit dem nächsten Block oder kehre ins Hau programm zurück.

L_o ist die Anfangsadresse des Unterprogrammes. Zelle α + n + 1 enthält entweder das N_t Schlüsselwort für den nächsten Block oder den nächsten Befehl des Hauptprogrammes.

Speicherplatz	Befehl	Adresse	
- 1	B	( )	Bring Auswahlschlüsselwo
α	R	L_o + 421	Rückkehradresse setzen
+ 1	U	L_o + 532	Eingang ins Unterprogra
+ 2	usw.		Hauptprogramm

Der Befehl in α-1 bringt das Auswahlschlüsselwort in den Akku (jeder Befehl, der dies bewirkt, kann benutzt werden). L_o ist die Anfangsadresse des Unterprogrammes. Diese Aufruffolge bereitet das Unterprogramm nur vor; daraus resultiert keinerlei Ausgabe.

AUSGABE

Das Programm druckt das dezimale Äquivalent eines jeden Wortes entsprechend den Formatangaben aus. Linksseitige Nullen werden unterdrückt. Das Vorzeichen steht hinter der Zahl; Leertaste bei positiven Zahlen, Minuszeichen bei negativen. Der dezimale Punkt wird nur geschrieben, wenn Kommastellen angegeben sind. Alle beabsichtigten Leertasten zwischen Zahlen folgen hinter den Zahlen. Es werden keine Tabulatorsprünge und Wagenrückläufe ausgegeben.

BEDIENUNG

1. Zeitbedarf

In einem einfachen Beispiel, unter Benutzung der Standardschreibmaschine, waren für 100 Worte mit W_i = 9, 155 Sek. erforderlich, einschließlich der Ausgänge, Wagenrückläufe und Adressenmodifikationen nach jeweils 10 Worten.

Beispiel

Es seien 10 Zahlen in 3600 bis 3605 und 3630 bis 3633 gespeichert. Die Zahlen sollen folgendermaßen gedruckt werden:

1	2	3	4	5	6	7
XX.XXX±	XX.XXX±	XX.XXX±	XX.XXX±	XXXX.XX±	XXXX.XX±	XXXXX±...

8	9	10
XXXXX±	XXXXX±	XXXXX±...

1. Zahl: 2 Ziffern vor dem Komma, 3 Ziffern nach dem Komma
2. - 4. Zahl: 2 Ziffern vor dem Komma, 3 Ziffern nach dem Komma, 2 Leertasten
5. u.6. Zahl: 4 Ziffern vor dem Komma, 2 Ziffern nach dem Komma, 2 Leertasten
7.-10. Zahl: 5 Ziffern vor dem Komma, 0 Ziffern nach dem Komma, 3 Leertasten.

Das Unterprogramm beginne in 1600.
Die Aufruffolge möge in 2243 beginnen und nicht folgendermaßen aus:

Speicherplatz	Befehl	Adresse	Anmerkungen
2243	R	1622	
2244	U	1600	
2245	80X6	3600	N_t = 6; erste Zahl in 3600
2246	X1	0703	N_i = 1; (I=2)+(D=3)+2 = 7
2247	X3	0903	N_2 = 3; (I=2)+(D=3)+(S=2)+2 = 9
2248	X2	1002	N_3 = 2; (I=4)+(D=2)+(S=2)+2 =10
2249	80X4	3630	N_t = 4; erste Zahl in 3630
2250	X4	0900	N_1 = 4; (I=5)+(S=3)+1 = 9

Die Spalte Anmerkungen zeigt jeden Wert für N und die Berechnung der Werte. Man beachte, daß die ΣN_i in jedem Block dem Wert N_t entspricht. Außerdem beachte man, daß, nachdem (I), Anzahl von Ziffern der ganzen Zahl, (D), Anzahl der Dezimalstellen und (S), Leertasten, aufaddiert sind, noch die Konstante 2 für Punkt und Vorzeichen hinzuaddiert wird.

In Zelle 2250 ist diese Konstante nur 1, weil kein dezimaler Punkt erforderlich ist.

DATENAUSGABE 3 | J3-10.2

VERWENDUNGSZWECK

Das Programm J3-10.2 druckt den Inhalt aufeinanderfolgender Zellen in dezimaler Schreibweise. Jede Zahl kann bestehen aus: 7 Ziffern, Vorzeichen, dezimalem Punkt und linksseitigen Nullen. Die zu druckenden Größen müssen ganze Zahlen bei q = 30 sein. Linksseitige Nullen und das Pluszeichen werden durch Leertasten ersetzt. Der Punkt wird nur geschrieben, wenn Kommastellen angegeben sind. Größen, die im gleichen Format ausgeschrieben werden sollen, können in Gruppen zusammengefaßt werden. Es sind beliebig viele Blöcke aufeinanderfolgender Zellen und Gruppen gleichartiger Worte innerhalb der Blöcke möglich.

Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes aufgerufen wird. Die Aufruffolge muß liefern:

1. die Gesamtzahl der auszugebenden Worte,
2. die Anzahl aufeinanderfolgenden Worte in jeder Gruppe,
3. die Feldbreite einer jeden Zahl und
4. die Anzahl der Kommastellen eines jeden Wertes.

Bei der Rückkehr ins Hauptprogramm sind die angegebenen Werte ausgedruckt.

SPEICHERBEDARF

5 Spuren, 13 Sektoren,
5 Spuren und 7 Sektoren, wenn der Eingang für die Auswahl der Ausgabeeinheit nicht benutzt wird.
Keine Zwischenspeicher.

EINGABE

Die Eingabevariablen bestehen aus zwei Arten von Schlüsselworten: N_t Blockangaben und N_i Gruppenangaben. Diese Schlüsselworte zur Kontrolle der Ausgabe müssen von der Aufruffolge geliefert werden:

N_t Dieses Schlüsselwort muß negativ sein. Es enthält die Gesamtzahl der aufeinanderfolgenden auszugebenden Worte bei q = 15 und die Anfangsadresse der Daten bei q = 29.

N_i Dieses Schlüsselwort muß positiv sein. Es enthält folgende Information:

(N_i) Die Gesamtzahl aufeinanderfolgender Worte der Gruppe mit gleichem P und D. N_i steht im Befehlsteil des Wortes.

(W_i) Die Feldbreite einer jeden Zahl der Gruppe. W_i steht im Spurteil des Wortes.

(D_i) Die Anzahl der Kommastellen eines jeden Wortes der Gruppe. D_i steht im Sektorteil des Wortes.

Die zu druckenden Zahlen müssen im Speicher bei q = 30 stehen und kleiner als $\pm$ 1 x 10^7 sein. Die Schranken für N_t, N_i, W_i und D_i sind:

$7 \geq D_i \geq 0$

$128 \geq W_i \geq (2 + D_i)$ was genügend Raum für Punkt und Vorzeichen freiläßt

$15 \geq (N_i \text{ und } N_t) \geq 1$

$\Sigma N_i = N_t$ für jeden Block

Speicherplatz	Befehl	Adresse	
α	R	L_o + 22	Adresse des ersten Schlüsselwortes in das Unterprogramm setzen.
+ 1	U	L_o	Eingang ins Unterprogramm (L_o + 7 ist als Eingang äquivalent zu L_o).
+ 2	80X(N_t)	ttss	N_t @ 15 und Anfangsadresse der Daten
+ 3	(N_1)	W_1D_1	Formatangaben
+ 4	(N_2)	W_2D_2	Formatangaben
.			
.			
.			
+ n	N_i	W_iD_i	
+ n + 1			Beginn des nächsten Blockes oder Rückkehr ins Hauptprogramm (Rückkehrbefehl muß positiv sein).

L_o ist die Anfangsadresse des Unterprogrammes. Zelle α + n + 1 enthält entweder das N_t-Schlüsselwort für den nächsten Block oder den nächsten Befehl des Hauptprogrammes.

Speicherplatz	Befehl	Adresse	
- 1	B	( )	Bringe Auswahlcode
α	R	L_o + 512	Rückkehradresse setzen
+ 1	U	L_o + 220	Eingang ins Unterprogramm
+ 2	usw.		Hauptprogramm

Der Befehl in α-1 bringt den Auswahlcode für die Ausgabeeinheit in den Akku (jeder Befehl, der dies tut, ist zulässig). L_o ist die Anfangsadresse des Unterprogrammes. Diese Aufruffolge bereitet das Unterprogramm nur vor; Ausgabe resultiert daraus nicht.

AUSGABE

Das Programm schreibt das dezimale Äquivalent eines jeden für die Ausgabe definierten Wortes aus. Linksseitige Nullen werden unter-

drückt. Das Vorzeichen folgt der Zahl, Leertaste für positives und Minuszeichen für negatives Vorzeichen. Der Punkt wird nur geschrieben, wenn Kommastellen angegeben sind. Alle vorgesehenen Leertasten stehen hinter den Zahlen. Tabulatorsprünge und Wagenrückläufe werden nicht ausgegeben.

BEDIENUNG

1. Zeitbedarf

In einem einfachen Beispiel, unter Benutzung der Standardschreibmaschine, waren für 100 Worte mit $W_i = 9$ 155 Sekunden erforderlich, einschließlich der Ausgänge, Wagenrückläufe und Adressenmodifikationen nach jeweils 10 Worten.

Beispiel

Es seien 10 Zahlen in 3600 bis 3605 und 3630 bis 3633 gespeichert. Die Zahlen sollen folgendermaßen gedruckt werden:

```
1       2          3         4         5          6          7
XX.XXX±XX.XXX±.. XX.XXX±  XX.XXX±  XXXX.XX±  XXXX.XX±  XXXXX±...
8         9          10
XXXXX±   XXXXX±   XXXXX±...
```

1. Zahl:	2 Ziffern vor dem Komma, 3 Ziffern nach dem Komma,
2. - 4. Zahl:	2 Ziffern vor dem Komma, 3 Ziffern nach dem Komma, 2 Leertasten
5. und 6. Zahl:	4 Ziffern vor dem Komma, 2 Ziffern nach dem Komma, 2 Leertasten
7. und 10. Zahl:	5 Ziffern vor dem Komma, 0 Ziffern nach dem Komma, 3 Leertasten.

Das Unterprogramm beginne in 1600. Die Aufruffolge möge in 2243 beginnen und sieht folgendermaßen aus:

Speicherplatz	Befehl	Adresse	Anmerkungen
2243	R	1622	
2244	U	1600	
2245	80X6	3600	$N_t = 6$; erste Zahl in 3600
2246	X1	0703	$N_i = 1$; (I=2)+(D=3)+2 = 7
2247	X3	0903	$N_2 = 3$; (I=2)+(D=3)+(S=2)+2 = 9
2248	X2	1002	$N_3 = 2$; (I=4)+(D=2)+(S=2)+2 = 10

Speicherplatz	Befehl	Adresse	Anmerkungen
2249	80X4	3630	N_t = 4; erste Zahl in 3630
2250	X4	0900	N_1 = 4; (I=5)+(S=3)+1 = 9

Die Spalte Anmerkungen zeigt jeden Wert für N und die Berechnung der Werte. Man beachte, daß die $\sum N_i$ in jedem Block dem Wert N_t entspricht. Außerdem beachte man, daß, nachdem

(I) Anzahl von Ziffern der ganzen Zahl,
(D) Anzahl der Dezimalstellen und
(S) Leertasten

aufaddiert sind, noch die Konstante 2 für Punkt und Vorzeichen hinzuaddiert wird. In Zelle 2250 ist diese Konstante nur 1, weil kein dezimaler Punkt erforderlich ist.

BEMERKUNG

Dieses Programm arbeitet zufriedenstellend sowohl mit der mod 18 als auch mit der mod 9 Maschine.

DATENAUSGABE 4	J3-10.3

VERWENDUNGSZWECK

Das Programm druckt den Inhalt des Akku aus. Bis zu neun Dezimalstellen plus Vorzeichen, dezimaler Punkt und Tabulator werden ausgegeben. Linksseitige Nullen werden unterdrückt, und die Ausgabe wird richtig gedruckt. Die auszugebenden Daten dürfen bei einem beliebigen, von 10 vorgegebenen q's stehen.

Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Die Aufruffolge muß liefern: 1. die Ziffernzahl C, die vor dem dezimalen Punkt stehen kann, und 2. die Ziffernzahl D, die nach dem dezimalen Punkt gedruckt werden soll (q ist durch C angegeben).

SPEICHERBEDARF

3 Spuren, keine Zwischenspeicher

EINGABE

Die Eingabe besteht aus dem zu druckenden Datenwort und dem Schlüsselwort für C und D.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
$\alpha - 1$	B	( )	(Beliebige Befehlsfolge, die Daten liefert). Bringe Datenwort in den Akku.
α	R	Lo + 12	
$\alpha + 1$	U	Lo	Eingang in das Unterprogramm
$\alpha + 2$	Z	ODOC	D @ 23 und C @ 29
$\alpha + 3$	etc.	Hauptprogramm	

SCHLÜSSELWORT

D darf eine beliebige Zahl zwischen 0 und 9 sein, vorausgesetzt, daß $C+D \leq 9$ ist.

C darf eine beliebige Zahl zwischen 0 und 9 sein, vorausgesetzt, daß $C+D \leq 9$. Das q der zu druckenden Größe wird durch C geliefert, wie die folgende Tabelle zeigt.

C	q
0	0
1	4
2	7
3	10
4	14

C	q
5	17
6	20
7	24
8	27
9	30

AUSGABE

Die Größe im Akku wird mit C Stellen vor dem Komma (linksseitige Nullen werden unterdrückt) und D Stellen hinter dem Komma (wohlaufgerundet), einem Minuszeichen bzw. einer Leertaste und einem Tabulatorsprung gedruckt.

BEDIENUNG

1. Zeitbedarf — Variable, abhängig von C und D. Ausgabegeschwindigkeit ca. 6 Zeichen/
2. Genauigkeit — Werden neun Ziffern gedruckt, so kann der neunten Stelle ein Fehler von 1 anhaften.
3. Programmeingabe — Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hex., zulässig für Programmeingabe 1 und
   b) dezimal in Kodierungsblattformat
4. Eingabe/Ausgabe-Einheiten — Die Ausgabe erfolgt über die 121-er Schreibmaschine. Eingabeeinheiten sind nicht erforderlich.

FEHLERSTOPS

Lo + 120 — C+D ist größer als 9. Nach "Start" wird "fld" gedrückt, und es erfolgt Rückkehr ins Hauptprogramm.

Lo + 120 — Die auszugebende Zahl kann nicht mit C Dezimalziffern gedruckt werden. Nach "Start" wird "big" ausgedruckt, und es erfolgt Rückkehr ins Hauptprogramm.

BEISPIEL

Es soll eine Zahl des Formates XXXXX.XX± gedruckt werden.

```
1345        B       (     )  zu druckende Zahl @ 17
1346        R       2312     Lo + 12
1347        U       2300     Lo
1348       XZ       0205
1349      usw.
```

Vorausgesetzt ist, daß Datenausgabe 4 in 2300 beginnt. Die Aufruffolge beginnt in 1345.

DATENAUSGABE 5	J3-10.4

VERWENDUNGSZWECK

Das Programm J3-10.4 gibt eine neunstellige Dezimalzahl plus Vorzeichen und Punkt aus. Bei einer negativen Zahl wird das Minuszeichen geschrieben; anstelle des Pluszeichens erscheint eine Leertaste.

Das Programm arbeitet wie ein Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Die Aufruffolge muß in Zelle α+ 2 ein Schlüsselwort angeben, das die Anzahl der Ziffern des ganzzahligen Teiles angibt.

SPEICHERBEDARF

120 Sektoren
keine Zwischenspeicher

EINGABE

Die Eingabe besteht aus der auszudruckenden Zahl und der Kennziffer C, die die Ziffernzahl des ganzzahligen Teiles angibt.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	Bemerkung
- 1	B	L(N)	Bringe Datenwort in den Akku.
α	R	L_o + 12	Adresse des Schlüsselwortes ins Unterprogramm setzen
+ 1	U	L_o	Eingang ins Unterprogramm
+ 2	Z	000C	Kennbuchstabe C, der "q" entspricht (siehe Tabelle)
+ 3	usw.		

α- 1 braucht keinen Bringe-Befehl zu enthalten. Jeder Befehl, der das Argument in dem Akku läßt, ist zulässig.

C ist eine Kennziffer zwischen 0 und 9.

AUSGABE

Die Ausgabe besteht aus:

Neun Dezimalziffern, Vorzeichen bzw. Leertaste, wenn sie positiv ist, und dem dezimalen Punkt. Linksseitige Nullen werden durch Leertasten ersetzt.

SCHLÜSSELWORT

C	q	Output
0	0	.XXXXXXXXX
1	4	X.XXXXXXXX
2	7	XX.XXXXXXX
3	10	XXX.XXXXXX
4	14	XXXX.XXXXX
5	17	XXXXX.XXXX
6	20	XXXXXX.XXX
7	24	XXXXXXX.XX
8	27	XXXXXXXX.X
9	30	XXXXXXXXX.

Anmerkung

Jede Dualzahl wird als Bruch skaliert und umgewandelt. Die Kennziffer wird vom Unterprogramm als Anschlagzähler verwandt. Nach jeder Zahl wird ein Tabulatorsprung ausgeführt. Dann erfolgt Rückkehr nach $\alpha + 3$ des Hauptprogrammes.

BEDIENUNG

1. Zeitbedarf — 3.0 Sekunden pro Zahl
2. Genauigkeit — der maximale Fehler ist eine 1 auf der neunten Ziffernstelle
3. Programmeingabe — Das Programm liegt in zwei Fassungen vor:
   a) willkürlich verlegbar hex. und
   b) dez. in Koordinierungsblattformat.

   Zulässiges Eingabeprogramm ist Programmeingabe 1
4. Eingabe/Ausgabe-Einheiten — Das Programm wählt die Lochstreifenschreibmaschine für die Ausgabe aus.

DRUCKBEFEHLE

Lo + 0048	xp0200
Lo + 0053	xp0200
Lo + 0111	xp0200
Lo + 0117	xp0200
Lo + 0132	80xp0200

FEHLERSTOPS

Speicherplatz	Bedeutung
L_o + 46	Das Argument ist $\geq 10^c$. Der Rechner hält an. Nach "Start" Ausgang ohne Drucken.

DATENAUSGABE	J3-10.5

VERWENDUNGSZWECK

Das Programm J3-10.5 druckt in dezimaler Schreibweise eine oder mehrere Gruppen von Zahlen, von denen jede eine oder mehrere Zahlen enthalten kann. Die Zahlen innerhalb einer Gruppe müssen in aufeinanderfolgenden Zellen beim gleichen q gespeichert sein. Linksseitige Nullen werden durch Leertastenersetzt; Leertaste oder Minuszeichen folgt hinter der letzten Dezimalstelle.

SPEICHERBEDARF

224 Zellen (3 1/2 Spuren)

Als Zwischenspeicher werden die Sektoren 42 und 43 von Spur 63 benutzt.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse
α	R	Lo + 3
α +1	U	Lo
α +2	Z	Loc_1
α +3	Z	$N_1 q_1$
α +4	Z	Loc_2
α +5	Z	$N_2 q_2$
.	.	.
.	.	.
.	.	.
$[\alpha+2i]$	Z	Loc_i
$[(\alpha+2i)+1]$	Z	$N_i q_i$
$[(\alpha+2i)+2]$ usw.		

Dabei ist Loc_i die Adresse der ersten Zahl in der i-ten Gruppe; N_i die Gesamtzahl von Daten der i-ten Gruppe; q_i das binäre q (Zwischenstellen) aller Daten der i-ten Gruppe.

N_i und q_i belegen jeweils 2 dezimale Ziffern, (die Spur- und Sektorziffern entsprechend)

Der Befehl in $(\alpha+2i+2)$ darf kein Z-Befehl sein.

AUSGABE

Jede ausgegebene Zahl besteht aus 8 - 10 Dezimalziffern, einem Punkt, einer Leertaste bzw. einem Minuszeichen und einem Tabulatorsprung.

Zur Definition einer jeden Datengruppe sind zwei Z-Befehle erforderlich. Nach Ausgabe der letzten Datengruppe erfolgt Ausgang zu dem Befehl, der der Aufruffolge folgt; dieser Befehl darf kein Z-Befehl sein.

q	Ausgabe-Format
0	.XXXXXXXX±
0 - 4	X.XXXXXXX±
4 - 7	XX.XXXXXX±
7 - 10	XXX.XXXXX±
10 - 14	XXXX.XXXX±
14 - 17	XXXXX.XXX±
17 - 20	XXXXXX.XX±
20 - 24	XXXXXXX.X±
24 - 27	XXXXXXXX.±
27 - 30	XXXXXXXXX.±
30 - 31	XXXXXXXXXX.±

BEISPIEL

Speicherplatz	Befehl	Adresse	Bemerkungen	
α	xR	Lo + 3		
α + 1	xU	Lo		
α + 2	xZ	2100	Loc_1 = 2100	
α + 3	xZ	0407	N_1 = 4; q_1 = 7	(1)
α + 4	xZ	2110	Loc_2 = 2110	
α + 5	xZ	1115	N_2 = 11; q_2 = 15	(2)
α + 6	xB	β		(3)

Die obige Aufruffolge veranlaßt folgendes:

1. Druckt den Inhalt der Zellen 2100, 01, 02 und 03 als XX.XXXXXX± für die Zahlen, deren absoluter Wert kleiner als 100.00000 oder als XXX.XXXXX± für die Zahlen, die über diesem Bereich liegen (siehe q = 7 in der Tabelle)
2. Druckt den Inhalt von 2100 bis 2120 als XXXXX.XXX±
3. Ausgang nach α+6, wo der Nicht-Z-Befehl die Aufruffolge abschließt.

FEHLERSTOPS

Speicherplatz	Bedeutung und Behebung
Lo + 207	N < 1. Nach Start erfolgt Ausgang ohne Drucken.

NÄHERE ANGABEN ZUM PROGRAMM

1. Genauigkeit	Die Ausgabe kann einen Fehler von 1 auf der achten gedruckten Stelle aufweisen.
2. Zeitbedarf	ca. 25 Worte/min.
3. Programmeingabe	Das Programm liegt in zwei Fassungen vor:

willkürlich verlegbar hex.,
und dez. in Kodierungsblattformat.

Beide Streifen sind mit Programmeingabe 1 einlesbar

HEXADEZIMALE AUSGABE 2	J4-10.1

VERWENDUNGSZWECK

Das Programm J4-10.1 stanzt den Inhalt eines vorgegebenen Speicherbereiches auf Streifen und zwar in willkürlich verlegbarer hexadezimaler Form. Für jeden Block von 64 Worten (oder weniger) wird ein hexadezimales Füllschlüsselwort und eine Prüfsumme mitgestanzt. Die Ausgabe der Inhalte erfolgt in steigender Reihenfolge der Speicheradressen.
Der so erzeugte Programmstreifen kann mittels J1-10.0 eingelesen werden.

SPEICHERBEDARF 6 Spuren

Benutzte Zwischenspeicher Tabelle der hexadezimalen Bereiche (siehe "Eingabe")

EINGABE

J4-10.1 subtrahiert einen vorgegebenen Modifikator von allen Befehlsworten (mit Ausnahme von solchen, die ein I oder P im Befehlsteil haben). Daher müssen alle Befehle (oder hexadezimale Worte, die wie Befehle aussehen), die nicht modifiziert werden sollen, in der Tabelle der hexadezimalen Bereiche gekennzeichnet werden. Der Bereich, der diese Tabelle enthalten soll, und die Bestimmung der in dieser Tabelle enthaltenen Worte, wird im Anschluß an die Programmangaben "tble" und "hex" dem Rechner geliefert.
Nach dem Sprung zum Programmanfang stellt der Rechner folgende Fragen:

Ausdruck	Antwort
Tble	Man gebe die Grenzen für die Tabelle mit den hexadezimalen Bereichen in dezimaler Spur-/Sektorschreibweise ein. Werden keine Grenzen eingegeben, so wählt der Rechner Spur 63 aus. Wenn ein Bereich festgelegt wird, so gebe man Anfangs- und Endadresse in einem Wort an,

z.B. 0300 0463' würde die Spuren 3 und 4 für eine Tabelle festlegen. Die Tabelle lässt $(\frac{N}{2}-1)$ Eingänge zu - wobei N die Anzahl für den Tabellenbereich festgelegten Speicherplätze darstellt. Werden z.B. zwei Spuren festgelegt, so können 63 Eingänge gemacht werden.

Hex

Dieses ist ein Mehrfacheingang, d. h. die gesamte Tabelle der hexadezimalen Bereiche wird als eine Einheit aufgefasst. Man gebe in dezimaler Spur-/Sektorschreibweise die Grenzen für alle Bereiche, die mit absoluten Adressen ausgestanzt werden sollen, ohne Rücksicht auf das Erscheinen der Worte an. Man gebe die Anfangs- und Endadresse einer jeden Gruppe zusammenhängender Speicherzellen als ein Wort an, z.B. 0300 0301' 0402 0432' 05330533' u.s.w.
Jedes Wort, das im Befehlsteil P oder I enthält, wird mit einer absoluten Adresse ausgegeben. Ebenso wird jedes Wort, das "1" Bit außerhalb von Vorzeichen, Befehlsteil und Adressteil hat, mit absoluten Adressen ausgegeben. Diese brauchen nicht durch "Hex" festgelegt zu werden.

Pnch

Man gebe - in dezimaler Spur-/Sektorschreibweise - die Grenzen des Speicherbereiches an, deren Inhalt ausgegeben werden soll. Man gebe Anfangs- und Endadresse in einem Wort an. z.B. 0300 0663'

Modi

Man gebe - in dezimaler Spur-/Sektorschreibweise - den Wert an, der von jedem Wort abgezogen werden muß, damit es als relative Adresse er-

scheint. z.B. 0300'
Die für die Ausgabe festgelegten Speicherzellen müssen sich in Serie in steigender Folge befinden. Es dürfen beliebig viele Zellen vorgegeben werden.

AUFRUFFOLGE

Das Programm wird aufgerufen durch Sprung zu seiner Anfangsadresse.

AUSGABE

Das Programm J4-10.1 stanzt den Inhalt eines vorgegebenen Speicherbereiches auf Streifen in willkürlich verlegbarer hexadezimaler Form. Die Daten werden in Blöcken von je 64 Worten ausgegeben (der letzte Block darf weniger enthalten). Jedem Block geht ein hexadezimales Füllschlüsselwort voraus, und jedem Block folgt eine berechnete Prüfsumme. Ein Wagenrücklauf wird nach höchstens 10 Worten gestanzt. Zwischen jeden ausgegebenen Wortblock werden Zwischenräume gestanzt. Die Ausgabe erfolgt in der aufsteigenden Reihenfolge der Speicherzellen. Der Inhalt der ausgegebenen Speicherzellen wird vom Programm nicht verändert.
Wenn die gewünschte Ausgabe beendet ist, hält der Rechner an. Nach START erfolgt ein Sprung zum Anfang des Programmes.

BEDIENUNG

1. PROGRAMMEINGABE

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar dezimal in Kodierungsblattformat.

2. ERFORDERLICHE EINGABE-/AUSGABEEINHEITEN

   Die Standardschreibmaschine wird vom Programm für Eingabe und Ausgabe ausgewählt.

HEXADEZIMALE AUSGABE 3	J4-10.2

VERWENDUNGSZWECK

Das Programm J4-10.2 stanzt den Inhalt eines vorgegebenen Speicherbereiches auf Streifen, und zwar in dem für J5-10.0 zulässigen Format. Die gewünschte Information wird in hexadezimaler Schreibweise gestanzt, und zwar in Gruppen von 64 Worten. Zwischen den Einzelgrößen einer Gruppe werden Zwischenräume gestanzt. Jede Gruppe enthält eine ganze Spur. Zu jeder Gruppe gehört ein hexadezimales Eingabeschlüsselwort, sowie eine Prüfsumme. Es können beliebig viele Spuren in einem Zuge ausgestanzt werden, jedoch müssen sie sich in aufsteigender Reihenfolge befinden.

SPEICHERBEDARF	4 Spuren oder 4 Spuren und 17 Sektoren, je nachdem welche Ausgabeeinheit ausgewählt werden soll. (siehe unter Programmeingabe).
Benutzte Zwischenspeicher	keine

EINGABE

Nach dem Sprung zur Anfangsadresse des Programmes wählt der Rechner die Schreibmaschine zur Eingabe aus und verlangt folgende Eingaben:

1. L_i Adresse der ersten auszustanzenden Spur als 2-ziffrige Spuradresse.
2. L_f Die Adresse der letzten auszustanzenden Spur als 2-ziffrige Spuradresse

Wenn der Rechner anhält und die Eingabe der Ausgabegrenzen verlangt, und bevor die Eingabe der Endadresse erfolgt, können negative Worte eingegeben werden, die unmittelbar auf Streifen gestanzt werden. Diese Worte werden vom Programm Hexadezimale Eingabe 3 als Sprunganweisungen aufgefaßt, und müssen also den Erfordernissen dieses Programmes entsprechen.

AUFRUFFOLGE

Das Programm wird aufgerufen durch Sprung zu seiner Anfangsadresse. Die Programme mit Bootstrap enden mit einem automatischen Sprung zur Anfangsadresse; die anderen Fassungen des Programmes enden mit einem Stop und Transfer. Siehe Programmeingabe.

AUSGABE

Das Programm stanzt den Inhalt von aufeinanderfolgenden Spuren in hexadezimaler Schreibweise auf Streifen. Die Ausgabe schreitet von der vorgegebenen Anfangsadresse zur vorgegebenen Endadresse fort. Dabei werden nur ganze, keine Teilspuren ausgestanzt.

Die hexadezimalen Worte befinden sich auf dem Streifen in 64-er Gruppen. Jede Gruppe umfaßt eine getrennte Spur, und die Worte innerhalb einer Gruppe haben ein bestimmtes durch Leertasten getrenntes Schreibmuster. Die Sektorreihenfolge innerhalb jeder Gruppe ist wie folgt:

56, 58, 60, 63, 01, 03, 05, 62, 00, 02, 04, 06, 08, 10, 12,
07, 09, 11, 13, 15, 17, 19, 14, 16, 18, 20, 22, 24, 26, 21,
23, 25, 27, 29, 31, 33, 28, 30, 32, 34, 36, 38, 40, 35, 37,
39, 41, 43, 45, 47, 42, 44, 46, 48, 50, 52, 54, 49, 51, 53,
55, 57, 59, 61.

Jeder Gruppe geht ein hexadezimales Eingabeschlüsselwort voraus, und jeder Gruppe folgt eine Prüfsumme.

Alle negativen Sprungschlüsselworte werden ausgestanzt, sobald sie eingegeben werden.

Im Anschluß an die Ausgabe der Endspur (L_f) erfolgt ein Sprung zur Anfangsadresse des Programmes, sodaß neue Ausgabegrenzen eingegeben werden können.

HEXADEZIMALE AUSGABE 3	J4-10.2

BEDIENUNG

1. ZEITBEDARF

   ca. 26 Sekunden sind pro Spur erforderlich, wenn die Ausgabe über den 151-er Stanzer erfolgt.

2. PROGRAMMEINGABE

   Der Programmstreifen enthält das Programm in vier Fassungen:

   a) willkürlich verlegbar hexadezimal mit Bootstrap auf Spur 00.
   b) willkürlich verlegbar hexadezimal mit Bootstrap auf Spur 63.
   c) willkürlich verlegbar hexadezimal zulässig für Programmeingabe 1 (J1-10.0)
   d) willkürlich verlegbar dezimal in Kodierungsblattformat.

   Das Eingabeverfahren variiert in Abhängigkeit von der jeweiligen Fassung und der benutzten Eingabeeinheit:

   Willkürlich verlegbar hexadezimal mit Bootstrap

   Willkürlich verlegbare hexadezimale Programme mit eigenem Bootstrap gestatten es dem Bediener, das Programm entweder über die 121-er Schreibmaschine oder über den 141-er Leser einzugeben, da für jede Eingabeeinheit ein besonderer Bootstrap vorgesehen ist. Sie erscheinen auf dem Streifen in der folgenden Reihenfolge:

   1. Spur-00-Bootstrap für Eingabe des Programmes über den 141-er Leser.
   2. Spur-00-Bootstrap für Eingabe des Programmes über die 121-er Schreibmaschine.
   3. Programm J4-10.2.
   4. Spur-63-Bootstrap für Eingabe des Programmes über den 141-er Leser.
   5. Spur-63-Bootstrap für Eingabe des Programmes über die 121-er Schreibmaschine.
   6. Programm J4-10.2

Man wähle einen Bootstrap entsprechend der gewünschten Spur und Eingabeeinheit aus, und gebe ihn entsprechend der Bootstrapprozedur ein(beschrieben in Programmeingabe 1).

Der erste im Bootstrap gespeicherte Befehl ist der Modifikator, der die Anfangsadresse des zu speichernden Programmes angibt. Auf dem Streifen befinden sich folgende (auf Wunsch veränderbare) Modifikatoren:

Spur-00-Bootstrap hat den Modifikator 0000.
Spur-63-Bootstrap hat den Modifikator 6000.

Nach Eingabe des Bootstrap werden zunächst 4 Spuren des Programmes automatisch eingelesen, dann hält der Rechner. Soll das Programm unter Benutzung der 121-er Schreibmaschine als Ausgabeeinheit ausgeführt werden, so ist jetzt ein Sprung "von Hand" zur Anfangsadresse notwendig.(Siehe Prozedur zum "Springen von Hand") Möchte der Bediener den 151-er Stanzer benutzen, so sollte er START drücken anstelle des oben erwähnten Springens von Hand, wenn der Rechner während des Programmeinlesens anhält. Es werden dann 14 Befehle gespeichert, die die vorher gespeicherten Druckbefehle ändern, sodaß der 151-er Stanzer als Ausgebeeinheit benutzt werden kann. Nach Speicherung dieser 14 Befehle hält der Reohner an. Damit ein Sprung zur Anfangsadresse des Programmes erfolgt, ist nur START zu drücken.

Willkürlich verlegbar, zulässig für Programmeingabe 1

Diese Fassung wird unter Benutzung der in Programmeingabe 1 beschriebenen Prozedur eingegeben. Das Einleseprogramm liest die vier Spuren des Programmes und hält dann. Soll die Schreibmaschine als Ausgabeeinheit benutzt werden, so drücke man "Eingabe von Hand" und die START-Taste. Die Schreibmaschine wird jetzt für die Eingabe benutzt. Man gebe einen Stop und Transfer zur Anfangsadresse des Programmes, drücke den Hebel START COMP und dann die START-Taste.

Möchte der Bediener die Ausgabeeinheit beliebig auswählen können, so verfahre man wie folgt, wenn der Stop des Eingabeprogrammes erfolgt:

1. Drücke den Hebel EINGABE VON HAND
2. Drücke die START-Taste. Die Schreibmaschine wird für Eingabe ausgewählt.
3. Schreibe das Schlüsselwort der Programmeingabe 1 für die Eingabeeinheit, über die das Programm eingelesen wird; z.B. S000 0000 für den 141-er Leser.
4. Drücke den Hebel START COMP.
5. Löse den Hebel EINGABE VON HAND.
6. Drücke die START-Taste.

Der Rest des Streifens (17 Befehle) wird eingelesen, und der Rechner hält an. (Am Ende dieser Folge steht ein Sprung und Transfer zur Anfangsadresse des Stanz-Programmes. Wird jetzt die START-Taste gedrückt, so beginnt die Ausführung des Programmes und die Schreibmaschine wird für Eingabe und Ausgabe ausgewählt.) Um die Ausgabeeinheit festzulegen, springe man von Hand nach L_o + 0400, wobei L_o die Anfangsadresse des benutzten Eingabeprogrammes ist. Das Programm wählt die Schreibmaschine für Eingabe aus. Man gebe das Auswahlschlüsselwort für die gewünschte Ausgabeeinheit als zweiziffrige Spuradresse ein. Das Programm modifiziert die Druckbefehle, die die gewünschte Einheit auswählen und beginnt zu arbeiten (d.h., verlangt Anfangs- und Endspurnummer) ohne anzuhalten.

3. DAS "SPRINGEN VON HAND"

a) Wahlschalter auf EINGABE VON HAND stellen.
b) Schreibe einen unbedingten Sprungbefehl zur gewünschten Zelle in hexadezimale Schreibweise; z.B. für Zelle 6000, schreibt man 000 U3J00.
c) Drücke FÜLLEN/LÖSCHEN
d) Wahlschalter auf Einzeloperation stellen.
e) Drücke AUSFÜHREN.

f) Wahlschalter auf NORMAL stellen.

g) Drücke die START-Taste.

4. ERFORDERLICHE EINGABE-/AUSGABEEINHEITEN

Für die Eingabe wird die Standardschreibmaschine ausgewählt. Die Ausgabeeinheit wird vom Eingabeprogramm entsprechend den Anweisungen des Bedieners ausgewählt.

HEXADEZIMALE AUSGABE 4	J4-10.3

VERWENDUNGSZWECK

Das Programm J4-10.3 stanzt Speicherblöcke in hexadezimaler Form aus. Die erhaltenen Streifen können mittels Programmeingabe 1 oder 2, (J1-10.0 oder J1-10.1) eingelesen werden. Die Ausgabeeinheit kann vom Benutzer nach Wunsch festgelegt werden.

SPEICHERBEDARF

3 Spuren

Benutzte Zwischenspeicher: keine

EINGABE

Die Eingabe besteht aus dem dezimalen Auswahlcode, der Anfangs- und Endadresse des zu stanzenden Blocks, und einer Adresse für das absolute "Stop und Sprung" Schlüsselwort (.000XXXX). Diese Daten müssen in dezimaler Spur-/Sektorschreibweise angegeben werden.

AUSGABE

Die Ausgabe besteht aus Blöcken von bis 64 Worten in hexadezimaler Form. Jedem Block geht ein Füllschlüsselwort voraus (VNNCHHHH; siehe Programmeingabe 1 oder 2), und jedem Block folgt eine Prüfsumme, in der das Füllschlüsselwort und jedes Wort des gestanzten Blocks enthalten sind.
Nach jeweil acht Worten wird ein Wagenrücklauf gestanzt. Bei Worten, die gleich Null sind, werden alle acht Nullen unterdrückt. Drei linksseitige Nullen (wie in den meisten Befehlsworten) werden unterdrückt. Alle anderen Worte werden in Form von acht hexadezimalen Zeichen gestanzt.

ZEITBEDARF

Stanzer der Lochstreifenschreibmaschine	ca. 68 Sek./Spur
Schnellstanzer	ca. 34 Sek./Spur

BEDIENUNG

Programmeingabe: Der Streifen enthält das Programm in zwei Fassungen:

a) willkürlich verlegbar hexadezimal, zulässig für Programmeingabe 1 (J1-10.0) und
b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

Bedienung des Programms:

1. Springe zur Anfangsadresse des Programms.
2. Schreibe den dezimalen Auswahlcode: 02 für den Stanzer der Lochstreifenschreibmaschine oder 06 für den Schnellstanzer.
3. Schreibe die Anfangsadresse des zu stanzenden Blocks in dezimaler Spur-/Sektorschreibweise.

4. Schreibe die Endadresse des auszustanzenden Blocks in dezimaler Spur-/Sektorschreibweise. (Die Endadresse muß größer gleich der Anfangsadresse sein.)

5. Schreibe die Adresse, die im "Stop und Sprung"-Schlüsselwort stehen soll, in dezimaler Spur-/Sektorschreibweise. Dies bewirkt, daß nach dem vollständigen Block .000TTSS gestanzt wird. Ist nach diesem Block kein "Stop und Sprung" erwünscht, so drücke man nur "Rechnen Start" (Schreibmaschine) ohne vorher einzugeben. (Ist .0000000 erwünscht, so ist vor "Rechnen Start" mindestens eine Null zu schreiben.)

6. Wenn das Stanzen beendet ist, beginnt das Programm wieder mit Schritt 3 und verlangt eine neue Anfangsadresse.

Erforderliche Eingabe-/Ausgabeeinheiten:

Das Programm wählt die Lochstreifenschreibmaschine für die Eingabe aus. Die Ausgabeeinheit ist durch den dezimalen Auswahlcode festzulegen: 02 für den Stanzer der Schreibmaschine; 06 für den Schnellstanzer.

ANMERKUNG

Das Programm setzt Überlauf zurück. Überlauf ist nach Stanzen eines Blocks zurückgesetzt.

ALPHANUMERISCHE AUSGABE 1	J4-11.0

VERWENDUNGSZWECK

Das Program J4-11.0 übersetzt Schlüsselworte und bewirkt ihre Ausgabe als alphabetischer und /oder numerischer Text. Der alphanumerische Text muß vom Hauptprogramm in 4-Bitart gespeichert sein. Er besteht aus einer Gruppe von Worten, die sich aus speziellen Codes entsprechend den Zeichen und Funktionen der Schreibmaschine zusammensetzen. Die Menge der für J4-11.0 zu speichernden Daten ist nur durch den verfügbaren Speicherraum begrenzt. Die Informationen müssen in aufsteigender Folge in aufeinanderfolgenden Zellen gespeichert werden.
J4-11.0 arbeitet wie ein Unterprogramm. Es wird durch eine Aufruffolge im Hauptprogramm aufgerufen. Vor dem Sprung ins Unterprogramm muß die Anfangsadresse des auszugebenden Datenblocks im Unterprogramm gespeichert werden. Die auf den Sprungbefehl folgenden Speicherplätze müssen die Schlüsselworte enthalten. Bei der Rückkehr ins Hauptprogramm ist die Ausgabe bereits erfolgt.

SPEICHERBEDARF 56 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die Anfangsadresse der Schlüsselwortliste muß vor dem Sprung ins Unterprogramm gespeichert werden.
Der Datenblock muß aus aufeinanderfolgenden Worten bestehen, die unmittelbar hinter dem Sprungbefehl der Aufruffolge stehen. Jedes Wort besteht aus vier Schlüsselsymbolen, welche den alphabetischen Zeichen sowie den Funktionen der Schreibmaschine entsprechen. Das letzte Wort des Blocks wird durch das spezielle Schlüsselsymbol 7Q gekennzeichnet. Da jedes Wort vier Schlüsselsymbole (acht Symbole) enthalten muß, sind zwei bis sechs Nullen nötig, um das letzte Wort zu füllen. Die Anzahl der Worte eines Blocks ist nur durch den vorhandenen Speicherraum begrenzt. Im Hauptprogramm muß vor jedem Block von 63 oder weniger Schlüsselworten ein Schlüsselwort zur hexadezimalen Eingabe stehen, wie es für Programmeingabe 1 erforderlich ist.

Liste der alphanumerischen Schlüsselsymbole für die Zeichen und Funktionen der Schreibmaschine.

Alphanumerisches Zeichen	Schlüssel-symbol	Alphanumerisches Zeichen	Schlüssel-symbol
)0	04	Bb	0F
L1	0J	Cc	6F
*2	14	Dd	2F
"3	1J	Ee	4F
Δ4	24	Ff	54
%5	2J	Gg	5J
$6	34	Hh	62
π7	3J	Ii	22

Alphanumerisches Zeichen	Schlüssel-symbol	Alphanumerisches Zeichen	Schlüssel-symbol
Σ8	44	Jj	64
(9	4J	Kk	6J
Leertaste	06	Ll	0J
_-	0A	Mm	3F
=+	16	Nn	32
:;	1A	Oo	46
?/	26	Pp	42
].	2A	Qq	74
[,	36	Rr	1F
Tab	30	Ss	7F
Unten	08	Tt	5F
Oben	10	Uu	52
Farbe	18	Vv	3A
WR	20	Ww	7J
Rücktaste	28	Xx	4A
'	40	Yy	12
Schlußsymbol	7Q	Zz	02
Aa	72		

Das folgende Beispiel zeigt, wie man eine Datengruppe für "Alphanumerische Ausgabe 1" zusammenstellt.

Programmeingabe-Schlüsselwort	Speicherplatz	Befehl	Adresse
	α	R	L_o
	+1	U	L_o
,0000004	+2	2042	0J72
	+3	3232	2232
	+4	5J06	4272
	+5	127F	7Q00
	+6	u.s.w.	

Dieses Beispiel bewirkt einen Wagenrücklauf, druckt PLANNING PAYS und kehrt ins Hauptprogramm nach α+6 zurück. 7Q im letzten Wort ist das Endschlüsselsymbol, die Schlußnullen vervollständigen lediglich das Wort.

AUFRUFFOLGE

Es gibt zwei Aufruffolgen für "Alphanumerische Ausgabe 1". Die Aufruffolge 1 wird benutzt, wenn die Standardschreibmaschine als Ausgabeeinheit in Frage kommt. Aufruffolge 2 wird benutzt, wenn eine andere Ausgabeeinheit in Frage kommt.

Aufruffolge 1

Speicherplatz	Befehl	Adresse	
α	R	L_o	Anfangsadresse der Daten nach Speicherzelle L_o bringen.
+1	U	L_o	Eingang ins Unterprogramm.
+2	(erstes Schlüsselwort)		
.			Schlüsselworte den alphanumerischen Text darstellend.
.			
.			
+n	(letztes Schlüsselwort)		
+n+1	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α speichert die Anfangsadresse des Datenblocks im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung ins Unterprogramm. Die Speicherzellen α+2 bis α+n enthalten die n Schlüsselworte für alphanumerischen Text.

ALPHANUMERISCHE AUSGABE 1	J4-11.0

Die Speicherzelle α+n+1 enthält den Befehl des Hauptprogramms, der nach Durchlaufen des Unterprogramms als erster ausgeführt wird.

Aufruffolge 2

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Auswahlcode in den Akkumulator bringen.
α	R	L_o	Anfangsadresse der Schlüsselworte im Unterprogramm speichern.
+1	U	L_o + 50	Eingang ins Unterprogramm.
+2	(erstes Schlüsselwort)		
.			Schlüsselworte, den alphanumerischen
.			Text darstellend.
.			
+n	(letztes Schlüsselwort)		
+n+1	u.s.w.		Rückkehr ins Hauptprogramm.

Diese Aufruffolge unterscheidet sich von der ersten durch zwei Befehle. Der Befehl in α-1 bringt den Auswahlcode in den Akkumulator. Dieser Auswahlcode kann im Operandenspurteil eines Z-Befehls stehen; z.B. xZttoo, wobei tt ein beliebiger Auswahlcode ist. Der Befehl in α+1 ruft das Unterprogramm auf durch einen Sprung nach L_o + 50.

AUSGABE

Bei der Rückkehr aus dem Unterprogramm ist der alphanumerische Text bereits ausgegeben.

BEDIENUNG

1. Zeitbedarf: ca. 660 Zeichen pro Minute.
2. Eingabe des Programms:
   Der Streifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.
3. Erforderliche Eingabe-/Ausgabeeinheiten:
   Eine Eingabeeinheit ist nicht erforderlich. Soll eine andere Ausgabeeinheit als die Schreibmaschine benutzt werden, so ändere man die Spuradresse in den Zellen L_o+06, L_o+14, L_o+22 und L_o+30 des Unterprogramms entsprechend (siehe LGP-21 Programmierungshandbuch, die die Codes der entsprechenden Einhalten enthalten.
4. Überlauf:
   Wird weder vom Programm beachtet noch während des Ablaufs verursacht.

ALPHANUMERISCHE AUSGABE 2	J4-11.1

VERWENDUNGSZWECK

Das Programm J4-11.1 gibt alphabetische und/oder numerische Informationen aus, die ohne Binärisierung gespeichert wurden. Die alphanumerischen Daten müssen als fünf-Zeichen lange Worte in 6-Bitart gespeichert worden sein. Ein Datenblock für J4-11.1 darf nicht mehr als 63 Zellen umfassen. Die Daten müssen in aufeinanderfolgenden Zellen gespeichert sein.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogrammes angesprochen wird. Die Anfangsadresse des Datenblockes muß dem Unterprogramm vor dem Eingang geliefert werden. Bei der Rückkehr in das Hauptprogramm sind die angegebenen Informationen bereits ausgegeben.

SPEICHERBEDARF 24 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Vor dem Sprung in das Unterprogramm muß die Aufruffolge in die erste Zelle des Unterprogrammes die Adresse des Z-Befehles einsetzen, der die Anfangsadresse des auszugebenden alphanumerischen Datenblockes enthält. Dadurch ist es möglich, den Datenblock an einer beliebigen Stelle zu speichern, anstatt ihn wie üblich der Aufruffolge folgen zu lassen. Der Z-Befehl darf nicht fehlen.
Die Tabelle der Daten besteht aus aufeinanderfolgenden Worten von je 5 Zeichen bzw. Schreibmaschinenfunktionen (das letzte Wort darf weniger enthalten). Die Anzahl von Worten eines Blockes darf nicht größer als 63 sein. Die Daten müssen in aufsteigender Folge in aufeinanderfolgenden Zellen gespeichert sein. Den Daten muß bei der Speicherung das Konstantenschlüsselwort von J1-10.0 vorausgehen, damit sie nicht binärisiert werden. (Das Eingabeschlüsselwort setzt auch automatisch ein Blockendekennzeichen "ins letzte Wort".

Beispiel

Programmeingabe-schlüsselwort	Speicherplatz	Befehl	Adresse
	α	R	L_o
	α+1	U	L_o
	α+2	Z	A_o
-	α+3	u.s.w.	
A0000003	A_o	PLANN	
	A_o+1	ING P	
	A_o+2	AYS	

Dieses Beispiel bewirkt den Ausdruck von PLANNING PAYS und Rückkehr nach α+3 des Hauptprogrammes. Das Blockendekennzeichen wird vom Programmeingabeschlüsselwort gespeichert. Die alphanumerischen Daten können an einer beliebigen Stelle im Speicher stehen.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
α	R	L_o	Adresse des Z-Befehles in L_o einsetzen.
+1	U	L_o	Eingang in das Unterprogramm.
+2	Z	A_o	Anfangsadresse des Datenblockes.
+3	u.s.w.		Rückkehr in das Hauptprogramm.

Der Befehl in α speichert die Adresse (α+2) des Befehles, der die Anfangsadresse des Datenblockes enthält, in das Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogrammes. Zelle α+2 enthält den Z-Befehl mit A_o, der Anfangsadresse des Datenblockes. Zelle α+3 enthält den Befehl des Hauptprogrammes, der nach Durchlaufen des Unterprogrammes als erster ausgeführt wird. Man beachte bitte, daß der Z-Befehl immer in der Aufruffolge stehen muß, selbst wenn der Datenblock der Aufruffolge unmittelbar folgt.

AUSGABE

Bei der Rückkehr in das Hauptprogramm ist der gewünschte Datenblock bereits ausgegeben.

BEDIENUNG

1. Zeitbedarf

   ca. 190 Zeichen pro Minute.

2. Programmeingabe

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar hexadezimal, zulässig für Programmeingabe 1, und
   b) willkürlich verlegbar dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten

   Eingabeeinheiten sind nicht erforderlich. Die Standardschreibmaschine wird für die Ausgabe ausgewählt. Soll eine andere Ausgabeeinheit benutzt werden, so ist die Spuradresse in den Zellen L_o+03, L_o+05, L_o+07, L_o+09 und L_o+011 des Unterprogrammes

   durch das entsprechende Auswahlkennzeichen zu ersetzen (siehe LGP-21 Programmierungshandbuch, wo die Auswahlkennzeichen und die zugehörigen Einheiten gelistet sind).

4. Überlauf

   Wird vom Programm weder getestet noch verändert.

BEMERKUNGEN

Das Programm sollte mit Sektor 00 beginnen, damit die Operationen optimal sind.

HEXADEZIMALE EINGABE 3	J5-10.0

VERWENDUNGSZWECK

Das Programm J5-10.0 liest und speichert hexadezimale Streifen, die vom Programm Hexadezimale Ausgabe 3 (J4-10.2) hergestellt wurde Es können nur volle Spuren gespeichert werden, und zwar nur soviele, wie Speicherraum verfügbar ist. Charakteristisch für das Programm ist seine hohe Einlesegeschwindigkeit.

SPEICHERBEDARF 3 Spuren

Benutzte Zwischenspeicher keine

EINGABE

Nach Sprung zur Anfangsadresse des Unterprogramms wählt der Rechner die Schreibmaschine aus und verlangt ingabe der Kennziffern für die vom Unterprogramm zu benutzende Eingabeeinheit (0000 für den Schnelleser oder 0200 für die Lochstreifenschreibmaschine). Schreibe das entsprechende Wort in hexadezimaler Schreibweise und drücke "Rechnen Start".
Das Programm übernimmt Streifen in speziellem Format in hexadezimaler Schreibweise; die hexadezimalen Worte müssen in 64er Gruppen angeordnet sein, wobei jede Gruppe einer Spur entspricht. Jeder Gruppe muß ein vier-Zeichen langes Schlüsselwort für hexadezimale Eingabe der Form tt00 vorausgehen, wobei tt die Spurnummer darstellt, und jeder Gruppe muß eine Prüfsumme, zusammengesetzt aus allen Worten der Gruppe, folgen.
Die Worte werden in einem bestimmten, durch Leertasten getrennten Muster gespeichert, sodaß sie auf dem Streifen in dieser Reihenfolge und Anordnung stehen müssen, damit sie in den richtigen Sektor gespeichert werden. Das Eingabemuster der Sektoren sieht folgendermaßen aus:

56, 58, 60, 63, 01, 03, 05, 62, 00, 02, 04, 06, 08, 10, 12, 07, 09,
11, 13, 15, 17, 19, 14, 16, 18, 20, 22, 24, 26, 21, 23, 25, 27, 29,
31, 33, 28, 30, 32, 34, 36, 38, 40, 35, 37, 39, 41, 43, 45, 47, 42,
44, 46, 48, 50, 52, 54, 49, 51, 53, 55, 57, 59, 61.

Das Unterprogramm ist in der Lage, ein "Stop und Sprung"-Schlüsselwort der Form 8000ttss zu übernehmen, wobei ttss eine Spur-/Sektoradresse in hexadezimaler Schreibweise ist. Das Schlüsselwort muß unbedingt auf eine Prüfsumme folgen, sonst wird es als hexadezimales Wort gespeichert und stört die Eingabe. Wird dieses Schlüsselwort nach einer Prüfsumme gelesen, so hält der Rechner an. Bei "Start" erfolgt ein Sprung zu der im Schlüsselwort angegebenen Adresse.
Während der Speicherung einer Gruppe von hexadezimalen Worten bildet "Hexadezimale Eingabe 3" eine Prüfsumme. Nach jeder vollen Spur wird diese Prüfsumme mit der auf dem Streifen befindlichen verglichen. Bei Gleichheit wird weiter eingelesen. Bei Ungleichheit wird ein Wagenrücklauf ausgeführt, ein "e" gedruckt und angehalten (siehe Fehlerstops).

AUFRUFFOLGE

Das Programm wird aufgerufen durch Sprung zu seiner Anfangsadresse. Wurde das Programm gerade erst eingelesen, so genügt es, "Start" zu drücken.

AUSGABE

Das einzulesende Programm wird auf den angegebenen Spuren in der richtigen Reihenfolge gespeichert.

BEDIENUNG

1. Zeitbedarf: ca. 13 Sek. zur Eingabe einer Spur.

2. Programmeingabe:
   Der Streifen enthält das Programm in vier Fassungen:
   a) Willkürlich verlegbar hexadezimal mit Bootstrap auf Spur 00.
   b) Willkürlich verlegbar hexadezimal mit Bootstrap auf Spur 63.
   c) Willkürlich verlegbar hexadezimal zulässig für Programmeingabe 1 (J1-10.0).
   d) Willkürlich verlegbar dezimal in Kodierungsblattformat.

   Das Eingabeverfahren variiert in Abhängigkeit von der benutzten Fassung.

   Willkürlich verlegbar hexadezimal mit Bootstrap

   Willkürlich verlegbare hexadezimale Programme mit eigenem Bootstrap gestatten es dem Bediener, das Programm entweder über die Lochstreifenschreibmaschine oder über den Schnelleser einzugeben, da für jede Eingabeeinheit ein besonderer Bootstrap vorgesehen ist. Sie erscheinen auf dem Streifen in der folgenden Reihenfolge:

   1. Spur-00-Bootstrap für Eingabe des Programms über den Schnelleser.
   2. Spur-00-Bootstrap für Eingabe des Programms über die Lochstreifenschreibmaschine.
   3. Programm J5-10.0.
   4. Spur-63-Bootstrap für Eingabe des Programms über den Schnelleser.
   5. Spur-63-Bootstrap für Eingabe des Programms über die Lochstreifenschreibmaschine.
   6. Programm J5-10.0.

   Man wähle einen Bootstrap entsprechend der gewünschten Spur und Eingabeeinheit aus, und gebe ihn entsprechend der Bootstrapprozedur ein (beschrieben in Programmeingabe 1).
   Der erste im Bootstrap gespeicherte Befehl ist der Modifikator, der die Anfangsadresse des zu speichernden Programmes angibt. Auf dem Streifen befinden sich folgende (auf Wunsch veränderbare) Modifikatoren:

   Spur-00-Bootstraps haben den Modifikator 0000.
   Spur-63-Bootstraps haben den Modifikator 6100.

   Willkürlich verlegbar hexadezimal, zulässig für Programmeingabe 1

   Diese Fassung wird nach dem in Programmeingabe 1 beschriebenen Verfahren eingelesen.

3. Erforderliche Eingabe-/Ausgabeeinheiten:
   Die Eingabeeinheit wird durch einen geeigneten Auswahlcode zu Beginn der Operation des Unterprogramms ausgewählt (siehe Eingabe). Eine Ausgabeeinheit ist nicht erforderlich.
4. Fehlerstops:

Ausdruck	Bedeutung und Behebung
e	Die Prüfsummen stimmen nicht überein. Lege den Streifen zurück bis vor die letzte eingelesene Gruppe und beginne neu.

BEMERKUNG

Das Programm sollte in einer geradzahligen Spur beginnen, damit das Einlesen optimal erfolgt.

ZUSATZ

Der Streifen "Hexadezimale Eingabe 3" in willkürlich verlegbarer hexadezimaler Fassung mit Spur-00-oder Spur-63-Bootstrap, kann so geändert werden, daß er zu Anfang eines Programmstreifens stehen kann. Nach den folgenden Änderungen wählt das Programm für die Eingabe des ihm folgenden Programms die gleiche Eingabeeinheit aus, die zu seiner eigenen Eingabe benutzt worden war.

```
'c01gj'8'u02g4'j's01k0'10'c0068'30'u0034'64'u0068'w4'b00kj'w8'
u02g8'70'u0074'78'u007j'80'u0084'88'c0118'8j'b0100'
```

```
1q0'a0000'1q4'a0000'1q8'a0000'1qj'a0000'1w0'a0000'1w4'a0000'
1w8'a0000'200'800i0000'204'18'2f4'kj'2f8'q4'2fj''c00w4''
```

Die Änderung geschieht also durch vier zusätzliche Befehle in der ersten Zeile und durch Änderung des letzten Wortes in der letzten Zeile. Der Rest des Programms, einschließlich Bootstrap, bleibt unverändert.

RICHTUNG ODER WINKEL EINGABE UND AUSGABE 1	J9-11.0

VERWENDUNGSZWECK

Das Programm J9-11.0 liest oder druckt einen positiven oder negativen Winkel, oder eine Richtung, bestehend aus der Angabe des Quadranten und einem positiven Winkel. Alle Winkel müssen kleiner als 512^{o} sein, und Richtungswinkel dürfen nicht größer als 90^{o} sein. Der eingegebene Winkel oder Scheitelkreis der Richtung wird binärisiert und in den Akkumulator gebracht.
Das Ausdrucken eines Winkels erfolgt in dezimaler Schreibweise in Grad und Minuten, jeweils gefolgt von einem Strich, und in Sekunden, bis zu einer Zehntelsekunde genau. Bei einer Richtung wird der Winkel genauso ausgedruckt, nur daß ein Zeichen N (Norden) oder S (Süden) vorausgeht und ein Zeichen E (Osten) oder W (Westen) folgt.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogramms aufgerufen wird. Vor dem Sprung ins Unterprogramm muß die Aufruffolge die Rückkehradresse speichern. Im Falle der Ausgabefunktion muß der Winkel oder die Richtung im Akkumulator stehen, bevor der Eingang ins Unterprogramm erfolgt. Bei der Rückkehr ins Hauptprogramm ist der Winkel bzw. die Richtung gelesen und in den Akkumulator gebracht, falls die Eingabefunktion aufgerufen war, oder der Winkel bzw. die Richtung sind ausgedruckt, falls die Ausgabefunktion aufgerufen war.

SPEICHERBEDARF	2 Spuren, 61 Sektoren
Benutzte Zwischenspeicher	die Sektoren 2, 5, 15, 17, 48, 51, 52, 56 und 60 der Spur 63

EINGABE

Die Eingabe für das Programm besteht aus einer der beiden folgenden Möglichkeiten:

1. Einem positiven Winkel in dezimaler Schreibweise in Grad, Minuten und Sekunden.
2. Einer Richtung, bestehend aus den Zeichen S oder N, einem positiven Winkel in dezimaler Schreibweise in Grad, Minuten, Sekunden und den Zeichen E oder W.

Jeder Wert muß als ein Wort angegeben werden; z.B.: 1422038' zur Angabe eines Winkels von $142^{o}20'38''$; S204010W zur Angabe der Richtung $SW20^{o}40'10''$; oder N000342E zur Angabe der Richtung NE3'42''. Die Winkelangabe kann maximal siebenziffrig sein; linksseitige Nullen dürfen fehlen. Alle Winkel müssen kleiner als 512^{o} sein. Jede Richtungsangabe muß aus acht Zeichen bestehen; bei verschwindenden Größen müssen die Nullen ausgeschrieben werden; algebraische Vorzeichen werden nicht angegeben. Die Winkelangaben bei Richtungen dürfen nicht größer als 90^{o} sein.

AUFRUFFOLGE

Im Unterprogramm sind drei Aufruffolgen vorgesehen: 1. für Eingabe, 2. für Winkelausgabe, 3. für Richtungsausgabe. In jedem Fall ist L_o die Anfangsadresse des Unterprogramms.

1. Eingabe:

Zum Einlesen eines Winkels oder einer Richtung bestimmt.

Speicherplatz	Befehl	Adresse	
α	R	$L_o + 47$	Rückkehradresse im Unterprogramm speichern.
+1	U	L_o	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr ins Hauptprogramm.

2. Winkelausgabe:

Zum Drucken eines Winkels bestimmt. Der Winkel muß sich im Akkumulator bei q=9 befinden, bevor der Sprung ins Unterprogramm erfolgt.

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe den Winkel in den Akkumulator.
α	R	L_o+249	Speichere Rückkehradresse im Unterprogramm.
+1	U	L_o+100	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr ins Hauptprogramm.

3. Richtungsausgabe:

Zum Drucken einer Richtung bestimmt. Der Scheitelkreis der Richtung muß sich im Akkumulator bei q=9 befinden, wenn der Sprung ins Unterprogramm erfolgt.

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Scheitelkreis in den Akkumulator bringen.
α	R	L_o+219	Rückkehradresse ins Unterprogramm speichern.
+1	U	L_o+200	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr ins Hauptprogramm.

In den Aufruffolgen 2 und 3 braucht der Befehl in α-1 kein "Bringe" Befehl zu sein. Jede Befehlsfolge, die die auszugebende Größe @9 in den Akkumulator bringt, ist zulässig. In allen Aufruffolgen speichert der Befehl in α die Rückkehradresse (α+2) ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung zur entsprechenden Stelle des Unterprogramms. In α+2 steht der Befehl des Hauptprogramms, der nach Durchlaufen des Unterprogramms als erster ausgeführt wird.

AUSGABE

Das Programm binärisiert jeden eingegebenen Winkel oder den Scheitelkreis jeder Richtungsangabe und setzt sie bei q=9 in den Akkumulator.
Jeder Winkelausdruck erfolgt in dezimaler Form in Grad, Minuten, Sekunden und Zehntelsekunden, wobei den Graden und Minuten ein Trennungsstrich folgt; z.B. würde ein Winkel von 80° 37'01'' ausgedruckt als 80-37-01.0.
Bei Ausgabe einer Richtung wird der Winkel gleicherweise geschrieben, jedoch geht einer der Buchstaben N oder S der Winkelangabe voraus, und es folgt ihr einer der Buchstaben E oder W; d.h. eine Richtung von NW 42°1'28'' würde gedruckt als N 42-01-28.0W.

RICHTUNG ODER WINKEL EINGABE UND AUSGABE 1	J9-11.0

BEDIENUNG

1. Genauigkeit: max. Fehler: ± 1 Sekunde.
2. Zeitbedarf: ca. 1,5 Sekunden zum Lesen oder Schreiben einer Größe.
3. Programmeingabe:
   Der Steifen enthält das Programm auf zwei Arten:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
4. Erforderliche Eingabe-/Ausgabeeinheiten:
   Das Unterprogramm wählt die Lochstreifenschreibmaschine sowohl für die Eingabe als auch für die Ausgabe aus. Die Auswahl kann geändert werden durch Änderung folgender Speicherinhalte:

Eingabe (80XIttOO)	Ausgabe (80XPttOO)
L_o + 1	L_o+105
	L_o+113
Ausgabe (XPttOO)	L_o+116
	Lo+123
L_o+121	L_o+126
L_o+131	L_o+133
L_o+140	L_o+136
L_o+210	L_o+142
L_o+216, L_o+218	

5. Überlauf:
   Wird beim Eingang nicht getestet und beim Ausgang jedoch zurückgesetzt.

BEMERKUNG

Nach jedem ausgegebenen Wert wird ein Tabulatorsprung ausgeführt. Um das Ausdrucken des Punktes und der letzten Dezimalstelle zu unterdrücken, ändere man L_o+137 von E0252 auf U0247.

TRACE 1	K1-10.0

VERWENDUNGSZWECK

Das Programm K1-10.0 erleichtert das Austesten eines im Speicher stehenden Festkommaprogramms durch eine eingehende Niederschrift. Beliebige oder alle Adressen, Befehle und/oder die Ergebnisse ihrer Ausführung können ausgedruckt werden. Adressen werden in dezimaler Spur-/Sektorschreibweise geschrieben; Rechenergebnisse in hexadezimaler Form.

SPEICHERBEDARF	3 Spuren, 51 Sektoren
Benutzte Zwischenspeicher	keine

EINGABE

Zum Austesten richtet das Unterprogramm simulierte Zähler und Befehlsregister, einen Akkumulator und eine PST im Speicher ein.
Nach Sprung zur Anfangsadresse des Unterprogramms wählt der Rechner die Schreibmaschine aus, führt einen Wagenrücklauf aus und verlangt die Adresse des Hauptprogramms, ab der geprüft werden soll, in dezimaler Spur-/Sektorschreibweise.
Wenn die Adresse, ab der geprüft werden soll, eingegeben ist, oder wenn PST eingeschaltet wurde, wählt der Rechner die Schreibmaschine aus, führt einen Wagenrücklauf aus und verlangt die Eingabe eines Schlüsselwortes, das die Form des Ausdrucks steuert. Eine Liste von Schlüsselzeichen und deren Funktionen folgt. Die dort erwähnten Register, Akkumulator, Zähl- und Befehlsregister, sind die vom Unterprogramm simulierten Register.

Schlüsselzeichen	Funktion
T	Schaltet PST ein. Diese "innere Programmsprungtaste" hat während der Prüfung des zu prüfenden Programmes die gleiche Funktion, wie die wirkliche PST während normaler Operation des zu prüfenden Programms.
C	Drucke den Inhalt des Zählregisters in dezimaler Spur-/Sektorschreibweise nach der Ausführung eines jeden Befehls des zu prüfenden Programms aus.
R	Drucke nach Ausführen eines jeden Befehls des zu prüfenden Programms den Inhalt des Befehlsregisters in Befehlsformat aus. Dem Befehl geht eine Leertaste oder ein Minuszeichen voraus.
A	Drucke nach der Ausführung eines jeden Befehls des zu prüfenden Programms den Inhalt des Akkumulators in hexadezimaler Schreibweise aus.

Anmerkung: Eine geeignete Kombination der obigen Schlüsselzeichen kann eingegeben werden und bleibt wirksam, bis sie durch eine neue ersetzt wird. Werden nur einige der obigen Schlüsselzeichen eingegeben, so sind alle nicht zu dieser Zeit eingegebenen unwirksam. Die Reihenfolge der Schlüsselzeichen ist beliebig.

Schlüsselzeichen	Funktion
X	Rückkehr zur Anfangsadresse des Unterprogramms zur Eingabe einer neuen Adresse für das Prüfen.
0	Prüfe von der Stelle aus weiter ohne Ausdrucken, an der angehalten wurde.
(keine)	"Starten" ohne Eingabe von T, C, R oder A bewirkt weiterprüfen ab der Stelle, wo angehalten wurde; dabei bleiben die Schlüsselworte von vorher wirksam.

Anmerkung: Der Akkumulator wird nie von "Trace 1" neu gesetzt. Für dieses "Haushalten" muß das zu prüfende Programm Sorge tragen.

AUFRUFFOLGE

Programmaufruf durch Sprung zur Anfangsadresse.

AUSGABE

Das Programm druckt den Inhalt der simulierten Register und/oder des Akkumulators je nach Schlüsselwortangabe aus, nachdem jeder Befehl des zu prüfenden Programms ausgeführt wurde.
Die Befehle des zu prüfenden Programms werden in der Tat genauso ausgeführt wie beim normalen Programmablauf. Jede auftretende Ausgabe wird in die gleiche Zeile geschrieben wie der Inhalt der simulierten Register. Wenn das Unterprogramm prüft ohne zu drucken, so wird jede auftretende Ausgabe genau da ausgeführt, wo auch der P-Befehl des zu prüfenden Programms steht.

Wird PST während des Prüfens eingeschaltet, so führt der Rechner einen Wagenrücklauf aus und druckt die Inhalte der simulierten Register entsprechend der angegebenen Schlüsselzeichen aus.
Dann hält der Rechner an und verlangt die Eingabe eines neuen Schlüsselwortes. Fortlaufendes Drücken von "Start" bewirkt die Ausführung des jeweils nächsten Befehls des zu prüfenden Programms und zwar so, als befände sich der Rechner in "Einzeloperation". Diese Betriebsart wird solange fortgesetzt, bis PST ausgeschaltet wird.
Man beachte bitte, daß das Prüfen in dieser Betriebsart ohne Auszudrucken zu Verwechslungen führen kann, wenn ein Eingabebefehl des zu prüfenden Programms angetroffen wird. Da PST jetzt eingeschaltet ist, führt der Rechner einen Wagenrücklauf aus, druckt die Adresse des letzten geprüften Befehls und verlangt Eingabe eines neuen Schlüsselwortes. Drücken von "Start" gestattet es, fortzufahren. Verlangt aber der jetzt folgende Befehl wieder Eingabe, so hält der Rechner wieder auf einem Eingabebefehl - verlangt jedoch diesmal Eingabe vom zu prüfenden Programm her. Diese Art von Eingabe kann daran erkannt werden, daß kein Adressenausdruck erfolgte, bevor der Rechner anhielt.

BEDIENUNG

1. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für J1-10.0 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

2. Erforderliche Eingabe-/Ausgabeeinheiten:

   Das Unterprogramm wählt die Lochstreifenschreibmaschine für Eingabe und Ausgabe aus.

3. Programmsprungtasten:

Taste	Funktion
PST	AUS - keine Wirkung.
	EIN - Sprung zu einem Eingabebefehl des Unterprogramms (neue Schlüsselworte).

4. Fehlerstops:

Das Unterprogramm führt als Stops alle Stopbefehle des zu prüfenden Programms aus. Die einzigen Stopbefehle des Unterprogramms sind programmiert vor der Eingabe der Schlüsselworte und vor der Eingabe einer neuen Prüfadresse.

BEMERKUNG

Verlangt das zu prüfende Programm während des Prüfens Eingabe, so geht das wenigstbedeutende Bit des eingegebenen Wortes verloren.

SPEICHERAUSDRUCKEN 1	K2-10.0

VERWENDUNGSZWECK

Das Programm K2-10.0 gibt den Inhalt eines vorgegebenen Speicherbereiches aus. Es gibt verschiedene Arten von Ausgabefassungen:
a) in Kodierungsblattformat,
b) willkürlich verlegbar dezimal, zulässig für J1-10.0
c) beide: a und b.
Die Worte können als Befehle oder als hexadezimale Konstanten ausgegeben werden, und zwar je nach Aussehen oder entsprechend den Angaben des Benutzers. Werden hexadezimale Worte ausgestanzt, so geht ihnen das Eingabeschlüsselwort für hexadezimale Konstanten voraus. Die Ausgabe erfolgt Zelle nach Zelle in aufsteigender Folge.

SPEICHERBEDARF 9 Spuren

Benutzte Zwischenspeicher keine

EINGABE

Nach Sprung zur Anfangsadresse von K2-10.0 verlangt der Rechner folgende Informationen:

Frage	Antwort
lins	Anzahl der Zeilen (lines) pro Seite als zweiziffrige Dezimalzahl.
locs	Anzahl der auf einer Seite zu druckenden Speicheradressen als zwei Dezimalziffern.
wrds	Die Anzahl der in eine Zeile zu druckenden Worte (wird nur wirksam, wenn PS-4 ein ist) als zwei Dezimalziffern.
skip	Die Anzahl von Leerzeilen zwischen zwei Seiten als zweiziffrige Dezimalzahl.
Anmerkung:	Die obigen Antworten müssen aus zwei Ziffern bestehen: Null muß eingegeben werden als "00". Die Leerzeilen zwischen den Seiten werden ausgegeben, wenn entweder die volle Zahl von Speicheradressen gedruckt wurde, oder wenn die Zahl der ausgegebenen Zeilen plus "skip" gleich "lins" ist.

Frage	Antwort
md a	Die Grenzen des Speicherbereichs, in dem der Modifikator wirksam werden soll (siehe unten), in einem Wort in dezimaler Spur-/Sektorschreibweise (d.h. TTSSTTSS).
modi	Der Wert, der vom Adressteil der Befehle im Modifikationsbereich subtrahiert werden soll, in dezimaler Spur-/Sektorschreibweise.
hx a	Die Grenzen des Bereichs, der in Form von hexadezimalen Worten ausgegeben wird, ohne Rücksicht auf den Inhalt, als ein Wort pro Bereich in dezimaler Spur-/Sektorschreib-

Frage	Antwort
	weise (d,h. TTSSTTSS). Die Beendigung der Eingabe wird durch zweimaliges Drücken von "Rechnen Start" nach der letzten Adresse gekennzeichnet.
pnch	Die Grenzen des Ausgabebereichs als ein Wort in dezimaler Spur-/Sektorschreibweise (d.h. TTSSTTSS).

Anmerkung: Bei allen Adressenangaben muß die zweite Adresse höher sein als die erste.

Ist PS-32 beim Eingang ins Programm eingeschaltet, so läßt der Rechner die ersten vier Fragen aus und benutzt die Werte vom vorherigen Durchlauf bzw. die im Programm schon gesetzten. Nach Einlesen des Programms sind folgende Werte gesetzt: lins=51, locs=32, wrds=08 und skip=08.
Nach der Antwort zu "pnch" befindet sich kein Programmstop, so daß der Schreibmaschinenhebel "Stanzen ein" gedrückt sein sollte, wenn "Start" gedrückt wird.

AUFRUFFOLGE

Das Programm wird aufgerufen durch Sprung zur Anfangsadresse.

AUSGABE

Die Ausgabe erfolgt in einer von zwei Fassungen, je nachdem ob PS-4 ein- oder ausgeschaltet ist.

a) Jede Zeile beginnt mit einer Adresse (Absolute minus Modifikator), gefolgt von einer Leertaste und einem Befehl pro Zeile. Eingabeschlüsselworte für hexadezimale Worte werden je eines zu einer Zeile ausgedruckt und abgezählt nach der Anzahl von Zeilen pro Seite.

b) Jede Zeile beginnt mit der Adresse (Absolute minus Modifikator) des ersten Wortes dieser Zeile, gefolgt von einer Leertaste und soviel Worten, wie für "wrds" angegeben ist. Eingabeschlüsselworte für hexadezimale Worte werden nicht nach der Anzahl von Worten pro Zeile abgezählt.

Bei beiden Formaten werden acht Zeichen ausgegeben: hexadezimale Worte enthalten linksseitige Nullen, wenn anwendbar; in Befehlsworten werden linksseitige Nullen durch Leertasten ersetzt.
Allen Befehlen außerhalb des Modifikationsbereichs wird bei der Ausgabe ein "X" vorangesetzt. P- und I-Befehle werden nie modifiziert und immer mit einem "X" versehen. Hexadezimale Worte werden in Gruppen von höchstens 63 Worten ausgegeben; jeder Gruppe geht ein Schlüsselwort für hexadezimale Eingabe voraus (siehe J1-10.0 bezüglich des Schlüsselwortes).
Jedes Wort, das ein "1" Bit außerhalb der Vorzeichenstelle des Operationsteils und des Operandenadressfeldes enthält, wird als hexadezimales Wort ausgegeben. Sieht ein Wort wie ein Befehl aus, ist aber in Wirklichkeit ein hexadezimales Wort, so muß dies beim Eingang von "hx a" angegeben werden. Werden die Befehle des Modifikationsbereichs nicht als hexadezimale Worte gekennzeichnet, bzw. sind es keine P oder I-Befehle, so wird der Modifikator von den Operandenadressen abgezogen, und sie werden nicht durch ein "X" gekennzeichnet.
Ausgegeben wird der Inhalt aufeinanderfolgender Zellen in aufsteigender Reihenfolge. Die Programmsprungtasten können zur Lenkung der Ausgabeart und des Formats verwandt werden. Ist der gesamte Bereich ausgegeben, so hält der Rechner an; Drücken von "Start" bewirkt Sprung zur Anfangsadresse des Programms.

SPEICHERAUSDRUCKEN 1	K2-10.0

BEDIENUNG

1. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

2. Erforderliche Eingabe-/Ausgabeeinheiten:

   Das Unterprogramm wählt die Lochstreifenschreibmaschine zur Ein- und Ausgabe aus.

3. Programmsprungtasten:

Taste	Funktion
PS-4	EIN - druckt eine Adresse (minus Modifikator) und eine vorgegebene Zahl von Worten pro Zeile. Schlüsselworte für hexadezimale Eingabe werden nicht gezählt.
	AUS - druckt eine Adresse und ihren Inhalt pro Zeile. Ein Schlüsselwort für hexadezimale Eingabe wird je als eine Zeile gezählt.
PS-8	EIN - springe zur Anfangsadresse des Unterprogramms und verlange neue Parameter, wenn das nächste Wort ausgegeben ist.
	AUS - keine Wirkung.
PS-16	EIN - halte nach Ausgabe eines jeden Wortes. Fortsetzung nach "Start" drücken.
	AUS - keine Wirkung.
PS-32	EIN - der Rechner benutzt das Seitenformat vom vorherigen Durchlauf, oder wenn das Programm gerade erst eingelesen wurde, das dem Programmoriginal eigentümliche Seitenformat. (d.h. die ersten vier Fragen und Antworten unterbleiben).
	AUS - Rechner verlangt Seitenformatangaben.

BEMERKUNG

Das optimale Arbeiten ist unabhängig von der Anfangssektornummer. Wird nur ein Speicherausdruck benötigt, so benutze man das Programm Speicherausdrucken 2, K2-10.1.

ZUSATZ

Unterdrücken des Adressenausdrucks:
Bei der Erzeugung eines dezimalen Streifens wird eine Adresse zu Beginn jeder Zeile ausgegeben. Dieser Streifen kann mittels Pro-

grammeingabe 1 über die Schreibmaschine, aber nicht über den Schnelleser eingelesen werden.
Durch Änderung eines Befehls in K2-10.0 wird die Ausgabe der Speicheradressen unterdrückt; es entsteht so ein Streifen, der über den Schnelleser eingegeben werden kann.

Ersetze	in Zelle	durch
$B[L_o+0242]$	L_o+0234	$U[L_o+0241]$

L_o ist die Anfangsadresse von K2-10.0.

SPEICHERAUSDRUCKEN 2	K2-10.1

VERWENDUNGSZWECK

Das Programm K2-10.1 druckt den Inhalt aufeinanderfolgender Zellen aus. Die Worte werden in hexadezimalem oder in Befehlsformat gedruckt, je nach Inhalt bzw. in Abhängigkeit vom Zustand von PST. Ausgegeben wird der Inhalt aufeinanderfolgender Zellen in aufsteigender Reihenfolge.

SPEICHERBEDARF	2 Spuren
Benutzte Zwischenspeicher	keine

EINGABE

Nach Sprung zur Anfangsadresse verlangt der Rechner folgende Ausgabeparameter:

1. L_o - Anfangsadresse des auszugebenden Bereichs in dezimaler Spur-/Sektorschreibweise.
2. L_f - Endadresse des auszugebenden Bereichs in dezimaler Spur-/Sektorschreibweise.

Die Endadresse eines Bereichs muß jeweils höher sein als die Anfangsadresse. Ist L_f zahlenmäßig kleiner als L_o, so schreibt der Rechner L_o, einen Wagenrücklauf und verlangt neue Parameter.
Nach Ausgabe des vorgegebenen Speicherbereichs verlangt der Rechner neue Ausgabeparameter.

AUFRUFFOLGE

Das Programm wird aufgerufen durch einen Sprung zur Anfangsadresse.

AUSGABE

Das Programm druckt pro Zeile eine Adresse und acht Worte, bis der Inhalt der Endadresse ausgedruckt ist. Hexadezimale Worte werden ausgedruckt, wenn "1" Bits außerhalb von Vorzeichen, Operationsteil und Operandenadressteil sind, oder wenn PST eingeschaltet ist. Befehlsworte werden gedruckt, wenn obige Bedingungen nicht zutreffen. Negativen Befehlen geht ein Minuszeichen voraus. Ausgegeben wird der Inhalt aufeinanderfolgender Zellen in aufsteigender Folge.

BEDIENUNG

1. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

2. Erforderliche Eingabe-/Ausgabeeinheiten:

   Das Unterprogramm wählt die Lochstreifenschreibmaschine zur Ein- und Ausgabe aus.

3. Programmsprungtasten:

Taste	Funktion
PST	EIN - drucke in hexadezimalem Wortformat.
	AUS - drucke in dem durch den Speicherinhalt definierten Format.

BEMERKUNG

Dieses Programm ist als einfaches Hilfsmittel zur Speicheruntersuchung gedacht. Wird eine für Programmeingabe 1 zulässige Ausgabe gewünscht, so benutze man K2-10.0.

SUCHPROGRAMM 1	K3-10.0

VERWENDUNGSZWECK

Das Programm K3-10.0 sucht einen vorgegebenen Speicherbereich nach einem bestimmten Argument ab, das entweder eine dezimale Adresse oder eine hexadezimale Konstante sein kann.

SPEICHERBEDARF 1 Spur, 37 Sektoren

BEDIENUNG

Drücke "Eingabe von Hand" auf der Lochstreifenschreibmaschine und springe nach L_o.

EINGABE

Das Programm schreibt folgende Buchstaben aus und verlangt Antworten darauf über die Tastatur der Schreibmaschine:

Buchstaben	Eingabe
i	Anfangsadresse des abzusuchenden Bereichs.
f	Endadresse des abzusuchenden Bereichs.
a	Argument. Ist PST gelöst, so gibt man eine dezimale Adresse ein. Ist PST gedrückt, so gebe man eine hexadezimale Konstante ein.

AUSGABE

PST gelöst:

Ist der Adressteil einer Zelle des abzusuchenden Bereichs identisch mit der eingegebenen Adresse, so erfolgt ein Wagenrücklauf, und die Adresse der Zelle wird dezimal ausgeschrieben, gefolgt von einer Leertaste und dem in ihr enthaltenen Befehl. Negativen Befehlen geht ein Minuszeichen voraus.

PST gedrückt:

Ein Wagenrücklauf wird ausgeführt und die Adresse der Zelle ausgedruckt.

PROGRAMMSTOPS

Wenn die Suche beendet ist, hält der Rechner in L_o + 27 an. Bei "Start" erfolgt Rückkehr nach L_o.

PROGRAMMTESTPROGRAMM	K9-10.0

VERWENDUNGSZWECK

Das Hilfsprogramm K9-10.0 gestattet es dem Programmierer, sein Programm auszuprüfen und zu verbessern, indem es folgende Funktionen ausführt:

1. Den Inhalt vorgegebener Zellen ändern.
2. Die Adressen und den Inhalt vorgegebener Zellen ausdrucken.
3. Einen vorgegebenen Speicherbereich nach einer bestimmten Adresse absuchen.
4. Beschreibe einen Speicherblock mit einer bestimmten Bit-Konfiguration.
5. Springe zu einer vorgegebenen Adresse.
6. Füge in das zu prüfende Programm zeitlich fixierte Rückkehrbefehle auf sich selbst ein, zur Benutzung während des anschließenden Testens.
7. Füge in das zu prüfende Programm einen zeitlich fixierten Stop ein, zur Benutzung beim anschließenden Testen.
8. Mache den vorher gesetzten, zeitlich fixierten Stop im zu prüfenden Programm wieder rückgängig.

SPEICHERBEDARF 7 Spuren, 32 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Informationen werden dem Unterprogramm durch 13 spezielle Schlüsselworte geliefert. Einige davon oder alle können verwandt werden. Die Funktion der Schlüsselworte wird auf den folgenden Seiten erläutert. Es wird vorausgesetzt, daß die Auswahl der gewünschten Eingabeeinheit geschieht, bevor Eingabe ins Unterprogramm versucht wird. Der Rechner hält auf einem Lesebefehl, bis das richtige Schlüsselwort eingegeben wird. Da "A" und "H" Schlüsselwort Doppelfunktionen haben, so wird jedes als zwei verschiedene Schlüsselworte betrachtet.

Anmerkung: Bei allen Eingabeschlüsselworten sind Anfangs- und Endadresse einschließlich gemeint. Eine Endadresse, die kleiner oder gleich einer Anfangsadresse ist, hat nur Wirkung auf eine Zelle. Alle Adressen müssen in dezimaler Spur-/Sektorschreibweise angegeben werden. Der Rechner kehrt nach Ausführung einer jeden Funktion zu einem Lesebefehl zurück. Das zu prüfende Programm muß gespeichert sein, wenn das Testverfahren beginnt.

BUCHSTABENFREIES SCHLÜSSELWORT Beispiel: 1021'B1150'

Dieses Schlüsselwort hat keinen Kennbuchstaben vor dem eingegebenen Befehlswort, das den Inhalt einer Zelle ersetzen soll. Stattdessen wird die gewünschte absolute Adresse benutzt, gefolgt von dem Befehlswort, das in die angegebene Zelle ohne Rücksicht auf deren vorherigen Inhalt gespeichert werden soll.

Die Operandenadresse des neuen Befehls wird bei der Eingabe nicht modifiziert. Nach jeder erfolgreichen Eingabe wird ein Wagenrücklauf ausgeführt.
Das obige Beispiel würde in Zelle 1021 den Befehl B1150 speichern.

H (Inhalt ändern) Beispiel: 563'H'35JWK'

Speichere in eine Zelle ein hexadezimales Wort. Es wird die absolute Adresse benutzt, der das "H" und das hexadezimale Wort folgt. Nach erfolgreicher Eingabe wird ein Wagenrücklauf ausgeführt.
Das obige Beispiel setzt in Zelle 0563 das hexadezimale Wort 00035JWK.

H (Druckkontrolle) Beispiel: H3200'3210'

Drucke in hexadezimalem Format den Inhalt aller Zellen zwischen Anfangs- und Endadresse. Eine Adresse und acht Worte werden pro Zeile gedruckt.
Das obige Beispiel würde den Inhalt der Zellen 3200 bis 3210 ausdrucken.

A (Inhalt ändern) Beispiel: 1111'A'TOTAL'

Speichere in eine Zelle ein alphanumerisches Wort, wobei jeder Buchstabe zwei Dualstellen nach links geschiftet wurde.
Das Beispiel würde den Inhalt von 1111 mit dem alphanumerischen Wort TOTAL füllen.

A (Druckkontrolle) Beispiel: A4200'4216'

Drucke in alphanumerischer Form den Inhalt aller Zellen zwischen Anfangs- und Endadresse. Eine Adresse und acht Worte werden pro Zeile gedruckt.
Das Beispiel würde den Inhalt der Zellen zwischen 4200-4216 ausdrucken.

P Beispiel: P2000'3203'

Drucke in Programmfassung den Inhalt aller Zellen zwischen Anfangs- und Endadresse. D.h., der Ausdruck erscheint in Befehlsformat, wenn alle Bits außerhalb von Vorzeichen, Operationsteil und Adressteil "0" sind; in hexadezimaler Schreibweise, wenn sie es nicht alle sind. Eine Adresse und ihr Inhalt werden pro Zeile gedruckt.
Ist der Ausdruck in Befehlsformat und ist PS-4 ausgeschaltet, so kann der Inhalt der Zelle, auf die sich die Operandenadresse bezieht, auch ausgedruckt werden, und zwar in Abhängigkeit von der Beschaffenheit des darin enthaltenen Befehls:

Befehl	Format
R, Y	Befehlswort
B, D, N, M, E, A, S	hexadezimales Wort
Z, I, P, U, T, H, C	kein Ausdruck

Das Beispiel würde den Inhalt der Zellen von 2000-3203 drucken.

I Beispiel: I1000'2100'

Drucke in Befehlsformat den Inhalt der Zellen zwischen Anfangs- und Endadresse. Eine Adresse und acht Worte werden pro Zeile gedruckt.
Das Beispiel würde den Inhalt der Zellen von 1000 bis 2100 ausdrucken.

S — Beispiel: S1645'3706'2204'

Suche den Speicher zwischen Anfangs- und Endadresse nach einer bestimmten Adresse ab. Die Adressen bedeuten der Reihenfolge nach: Anfangsadresse, Endadresse, zu suchende Adresse. Die gefundene Adresse und die alphabetische Bezeichnung des Befehls wird ausgedruckt. Die Suche wird fortgesetzt, bis der ganze Bereich abgesucht ist. Wird die Adresse mehrmals gefunden, so wird für jeden Ausdruck eine neue Zeile genommen. Nach beendeter Suche wird ein Wagenrücklauf ausgeführt.
Das Beispiel würde die Zellen 1645 bis 3706 nach der Adresse 2204 absuchen.

C — Beispiel: C2101'2110'F3F3F3F2'

Fülle den Bereich zwischen Anfangs- und Endadresse mit einer bestimmten Bit-Konfiguration. Nach Beendigung wird ein Wagenrücklauf und ein Sprung zur Anfangsadresse des Unterprogramms ausgeführt.
Die Bit-Konfiguration ist als hexadezimales Wort anzugeben.
Das Beispiel würde die Zellen von 2101 bis 2110 füllen mit der Bit-Konfiguration [10100011101000111010001110100 01].
Anmerkung: Da ein Speicherwort 31 Bits lang ist, würde das letzte Bit verloren gehen.

T — Beispiel: T1000'

Halte und springe nach "Start" zur angegebenen Adresse. Die Sprungadresse steht vor dem Sprung im Akkumulator.
Das Beispiel würde einen Sprung nach 1000 bewirken.

R — Beispiel: R3663'

Setze einen Rückkehrbefehl (U) in die angegebene Zelle des Unterprogramms. Wird der Sprung während der Operation angetroffen, so wird der ursprüngliche Inhalt wieder in die Zelle gespeichert, und es erfolgt ein Sprung zur Anfangsadresse des Unterprogramms. Wird der ursprüngliche Inhalt nicht wieder in die Zelle gespeichert, weil der Sprung nicht angetroffen wurde, so bewirkt ein Sprung auf diese Zelle das Wiedereinsetzen des ursprünglichen Inhalts in sie. Setze keinen Rückkehrbefehl, wenn ein vorhergehender nicht rückgängig gemacht wurde.
Das Beispiel würde einen Rückkehrbefehl in Zelle 3663 speichern.

SS — Beispiel: SS1230'

Setze einen Stop in die angegebene Zelle. Jeder Befehl des zu prüfenden Programms kann durch einen Stop ersetzt werden; es darf jedoch immer nur einer gesetzt sein, sonst erfolgt der Fehlerausdruck e. Der ursprüngliche Inhalt wird wieder gespeichert, wenn ein Sprung zu einem Stopbefehl des Unterprogramms erfolgt; in der Zelle hinter diesem Stopbefehl steht der ursprüngliche Inhalt der Zelle des zu prüfenden Programms.

Gleichzeitig wird ein Sprung zur gewünschten Adresse+1 in der nächstfolgenden Zelle des Unterprogramms gespeichert und ein Sprung zu obiger Adresse+2 in die dritte folgende Zelle. Nach "Start" arbeitet das Programm weiter.
Das Beispiel würde einen Stop in Zelle 1230 setzen.

RS — Beispiel: RS0000'

Mache den Stop rückgängig; ersetze den Zelleninhalt durch den ursprünglichen Inhalt. Ein Fehlerausdruck e erfolgt, wenn vorher kein Stop gesetzt worden war. Die Adresse ist bedeutungslos und darf daher aus vier willkürlichen Zeichen bestehen.
Das Beispiel würde einen Stop in jeder beliebigen Zelle rückgängig machen.

Bei allen Bereichen muß die Endadresse größer sein als die Anfangsadresse.

AUFRUFFOLGE

Das Programm wird aufgerufen durch einen Sprung zur Anfangsadresse.

AUSGABE

Die ausgabe hängt vom gewählten Schlüsselwort ab. Als Formate kommen, wieder in Abhängigkeit vom Schlüsselwort, in Frage: Befehlsformat, hexadezimales Format, alphanumerisches Format.

BEDIENUNG

1. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

2. Erforderliche Eingabe-/Ausgabeeinheiten:

   Das Unterprogramm wählt die Schreibmaschine für Eingabe und Ausgabe aus.

3. Programmsprungtasten:

Taste	Funktion
PS-4	AUS - Unter Benutzung des 'P'-Schlüssels kann auch der Inhalt der Adresse eines Befehlswortes ausgegeben werden.
	EIN - Keine Wirkung.
PST	AUS - Keine Wirkung.
	EIN - Beendet Druck und Suchoperationen und bewirkt jeweils einen Sprung zur Anfangsadresse des Unterprogramms.

4. Fehlerausdruck:

e	Ein Fehler wurde gefunden. In jedem Falle bewirkt "Start", daß der Rechner ein neues Schlüsselwort verlangt.

BEMERKUNGEN

Das gespeicherte Unterprogramm schützt sich selbst. D.h., keine Änderung wird in einer seiner Zellen vorgenommen. Sollte man dies versuchen, so wird e ausgeschrieben. Die Parameter des geschützten Bereichs können geändert werden durch Sprung nach L_o+0660—wobei L_o die Anfangsadresse des Unterprogramms ist—und durch Eingabe der neuen gewünschten Anfangs- und Endadressen. Z.B.: 1000'1732' würden als neuen geschützten Bereich die Zellen von 1000 bis 1732 definieren.

SORTIEREN 1	L1-10.0

VERWENDUNGSZWECK

Das Programm L1-10.0 sortiert eine Liste von Zahlen innerhalb des Speichers der Größe nach (kleinster Wert - größter Wert), entsprechend einer vorgegebenen Maske. Die sortierten Zahlen stehen auf dem gleichen Speicherbereich wie die unsortierten, so daß kein zusätzlicher Speicherraum erforderlich ist. Durch die Maske wird festgelegt, welche Wortteile verglichen werden.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogramms aufgerufen wird. Beim Sprung ins Unterprogramm muß die Anfangsadresse der zu sortierenden Zahlen im Akkumulator stehen, und die Adresse der Maske muß in das Unterprogramm gespeichert worden sein. Die beiden auf den Sprung folgenden Zellen müssen die Maske und die Endadresse der Liste enthalten. Bei der Rückkehr ins Hauptprogramm stehen die Zahlen der Größe nach (kleinster Wert - größter Wert) sortiert auf dem ursprünglichen Speicherbereich der Liste.

SPEICHERBEDARF	1 Spur, 32 Sektoren
Benutzte Zwischenspeicher	die Sektoren 54, 55, 62 und 63 der Spur 63

EINGABE

Reihenfolge der Eingabevariablen:

1. L_i, Anfangsadresse der zu sortierenden Zahlen in dezimaler Spur-/Sektorschreibweise.
2. Willkürliche Maske, die den Sortiervorgang kontrolliert, in hexadezimaler Schreibweise.
3. L_f, Endadresse der zu sortierenden Zahlen in dezimaler Spur-/Sektorschreibweise.

Die Maske darf keine "1" in der Vorzeichenstelle haben, da das Unterprogramm das Vorzeichen der zu sortierenden Zahlen nicht beachtet. L_i und L_f sind einschließlich gemeint.
Beim Sprung ins Unterprogramm muß L_i @ 29 im Akkumulator stehen. Die auf den Sprungbefehl folgenden Zellen müssen die Maske und L_f enthalten. Die zu sortierenden Zahlen müssen in aufeinanderfolgenden Zellen stehen.

AUFRUFFOLGE

Speicherzelle		Befehl	Adresse	
	-1	B	[ ]	Bringe L_i @ 29.
	α	R	L_o + 3	Adresse der Maske ins Unterprogramm setzen.
	+1	U	L_o	Sprung ins Unterprogramm.
,0000001	+2	[	$_o$]	Maske.
	+3	XZ	L_f	Angabe von L_f.
	+4	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt L_i in den Akkumulator @29. (Jeder Befehl, der dieses bewirkt, kann benutzt werden.) Der Befehl in α speichert die Adresse der Maske (α+2) im Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Zelle α+2 enthält die hexadezimale Maske. Der Befehl in α+3 enthält L_f. Zelle α+4 enthält den Befehl des Hauptprogramms, der nach Durchlaufen des Unterprogramms als erster ausgeführt wird.

BEDIENUNG

1. Zeitbedarf:

   Durchschnittswerte: 32 Worte - 114 sec.
   64 Worte - 6 min.
   128 Worte - 27 min.

2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten:

   Weder Eingabe- noch Ausgabeeinheiten sind erforderlich.

4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEISPIEL

In den Zellen 1500 bis 1731 seien Zahlen gespeichert, die nach ihrem Inhalt auf den Bitpositionen 7 bis 11 und 24-29 zu sortieren sind. "Sortieren 1" sei gespeichert auf 700 bis 832, und das Hauptprogramm belege die Zellen 900 bis 1215. Die Aufruffolge müßte so aussehen:

Speicherplatz		Befehl	Adresse	
	1011	XB	1211	Bringe L_i @29.
	1012	XR	0703	Adresse der Maske setzen.
	1013	XU	0700	Sprung ins Unterprogramm.
,0000001'	1014	01W0	00WJ	Maske.
	1015	XZ	1731	L_f.
	1016	u.s.w.		Rückkehr ins Hauptprogramm.
	.			
	.			
	1211	XZ	1500	L_i.

Der Befehl in 1011 bringt 1500 @ 29 in den Akkumulator; der Befehl in 1012 speichert die Adresse der Maske nach 0703; und der Befehl in 1013 bewirkt einen Sprung zur Anfangsadresse von "Sortieren 1". Zelle 1014 enthält die Sortiermaske. Man beachte, daß nur die Bitpositionen verglichen werden, denen "1"-er Bits in der Maske entsprechen. Der Befehl in 1015 legt die Endadresse der Liste fest. Zelle 1016 enthält den Befehl des Hauptprogrammes, zu dem die Rückkehr erfolgt.

SORTIEREN 2	L1-10.1

VERWENDUNGSZWECK

Das Programm L1-10.1 sortiert in einem zusammenhängenden Speicherbereich Einwort- oder Mehrwort- "Intervalle" der Größe nach (kleinster Wert - größter Wert) entsprechend einem im jeweiligen "Intervall" liegenden, vorgegebenen "Sortier-Schlüsselwort". Jedes Intervall kann 1 bis 64 Worte lang sein. Zum Zwischenspeichern der Intervallinhalte während des Sortiervorgangs wird ein zusätzlicher Speicherbereich benötigt. Das Sortieren geschieht in Abhängigkeit von einer willkürlichen Maske.
"Sortieren 2" arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogramms aufgerufen wird. Beim Sprung ins Unterprogramm muß ein Schlüsselwort, das die Anzahl der zu verarbeitenden Intervalle und die Anfangsadresse des ersten Intervalls angibt, im Akkumulator stehen. Außerdem muß die Aufruffolge das "Sortier-Schlüsselwort" und die Maske liefern. Bei der Rückkehr ins Hauptprogramm stehen die Intervalle sortiert im ursprünglichen Speicherbereich.

SPEICHERBEDARF	4 Spuren
Benutzte Zwischenspeicher	2 Spuren (L_o+0400 bis L_o+0563, wobei L_o die Anfangsadresse des Unterprogramms ist.

EINGABE

Die Eingabe besteht aus drei von der Aufruffolge zu liefernden hexadezimalen Schlüsselworten und den zu sortierenden Inhalten der Intervalle. Die Schlüsselworte sind:

1. R, Anzahl der Intervalle bei q=15; und L_i Anfangsadresse des ersten Intervalls bei q=29.
2. N, Anzahl der Worte in jedem Intervall bei q=15; und K, die Lage des Sortier-Schlüsselwortes innerhalb eines Intervalls bei q=29. (K=3 bei 29 besagt, daß das dritte Wort des Intervalls das Sortier-Schlüsselwort ist.
3. Maske, die die Bitpositionen angibt, an denen beim Sortieren verglichen werden soll.

Die Anzahl der in einem einzigen Arbeitsgang sortierbaren Intervalle ist nur durch den verfügbaren Speicherraum begrenzt. Für N, die Anzahl der Worte pro Intervall, gilt: $1 \leq N \leq 64$. Die Maske sollte keine "1" in der Vorzeichenstelle haben, da das Unterprogramm das Vorzeichen der Worte beim Sortieren nicht beachtet.
Die zu sortierenden Intervalle müssen in aufeinanderfolgenden Zellen stehen.

AUFRUFFOLGE

Speicherplatz		Befehl	Adresse	
	-1	B	[ ]	Bringe Wort mit R und L_i.
	α	R	L_o + 20	Adresse der Kontrolldaten speichern.
	+1	U	L_o	Eingang ins Unterprogramm.
,0000002	+2	[N	K^o]	N@15; K@29.
	+3	[	]	Maske
	+4	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt R@15 und L_i@29 in den Akkumulator. (Jeder Befehl, der dies bewirkt, ist zulässig.) Der Befehl in α speichert die Adresse (α+2) des Befehls, der N und K enthält, ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Zelle α+2 enthält N@15 und K@29. Zelle α+3 enthält die Maske. Nach α+4 erfolgt die Rückkehr aus dem Unterprogramm.

BEDIENUNG

1. Zeitbedarf:

   Durchschnittlich 3 Sek. pro Wort bei kleinen Beispielen.

2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten:

   Weder Eingabe- noch Ausgabeeinheiten sind erforderlich.

4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEISPIELE

Das Unterprogramm beginne in 300; das Hauptprogramm belege die Spuren 10 bis 15.

Beispiel 1

Es sind 37 Worte, gespeichert in 4900 bis 4936, als 37 Einwortintervalle zu sortieren, unter alleiniger Berücksichtigung der Bits 15 bis 30.

Speicherplatz		Befehl	Adresse	
	1010	XB	1559	Bringe R und L_i.
	1011	XR	0320	Speichere N-K-Wortadresse.
	1012	XU	0300	Springe nach "Sortieren 2".
,0000002'	1013	0001	0004	N=1@15; K=1@29.
	1014	0001	wwwQ	Maske für Bits 15 bis 30.
	1015	u.s.w.		Rückkehr ins Hauptprogramm.
	.			
	.			
,0000001'	1559	0025	3100	r=37@15; L_i=4900.

Man beachte, daß das Unterprogramm beim Sortieren nur die Wortteile vergleicht, die "1" Bits in der Maske entsprechen.

Beispiel 2

Es sind 32 Worte, gespeichert in 3200 bis 3231 als acht vier-Wort Intervalle entsprechend der Größe des zweiten Wortes eines jeden Intervalls der Größe nach zu sortieren (kleinster Wert - größter Wert; hier aber die Intervalle nach der Größe jeweils des zweiten Wortes aller Intervalle). Dabei soll nur an den Bitpositionen 4 bis 11 und 16 bis 20 verglichen werden.

Speicherplatz		Befehl	Adresse	
	1401	XB	1563	Bringe R und L_i.
	1402	XR	0320	Speichere N-K-Wortadresse.
	1403	XU	0300	Sprung nach "Sortieren 2".
,0000002'	1404	0004	0008	N=4@15; K=2@29.
	1405	0ww0	W800	Maske für Bits 4-11 und 16-20.
	1406	u.s.w.		Rückkehr ins Hauptprogramm.
	.			
	.			
,0000001'	1563	0008	2000	R=8@15; L_i=3200.

Es folgen Probeausdrucke, die zeigen, was die beiden Aufruffolgen bewirken würden.

PROBEAUSDRUCK

Beispiel 1

Speicherplatz	unsortiert	sortiert
4900	0wj00004'	kf000002'
4901	q0000010'	0wj00004'
4902	00jj0018'	0w000006'
4903	0kqw0026'	0fg00008'
4904	00qj001j'	0k00000f'
4905	7wqj002f'	0w00000j'
4906	0fg00008'	0q00000q'
4907	0w000030'	q0000010'
4908	0k00000f'	0k000012'
4909	0q00003j'	00000014'
4910	0g000032'	0j000016'
4911	0w00000j'	00jj0018'
4912	0q00003w'	0f00001f'
4913	0f00001f'	00qj001j'
4914	qj00003j'	0000001q'
4915	00000014'	qk000020'
4916	w000012j'	0k000022'
4917	0w000006'	0j000024'
4918	0q00000q'	0k000028'
4919	qk000020'	7wqj002f'
4920	3j000064'	0w000030'
4921	fg000190'	0g000032'
4922	kf000002'	0q000034'

Speicherplatz	unsortiert	sortiert
4923	0k000012'	00000036'
4924	00000036'	0q000038'
4925	0k000022'	0g00003f'
4926	0j000016'	0q00003j'
4927	0q000038'	qj00003j'
4928	0w000258'	0q00003q'
4929	0f00003w'	0f00003q'
4930	0j000024'	3j000064'
4931	0000001w'	0f0000j8'
4932	0g00003f'	w000012j'
4933	0f0000j8'	fg000190'
4934	0q000034'	0w0001w2'
4935	0k000028'	0w000258'
4936	0w0001w2'	0kqw0026'

Beispiel 2

Speicherplatz	unsortiert	sortiert
3200	080403j0'	0f0803j0'
3201	02q05000'	0080f800'
3202	00001000'	000f0800'
3203	00g00005'	01606800'
3204	0404003g'	010j03j0'
3205	03g0q800'	01607000'
3206	00048000'	0fg09800'
3207	02q0w800'	00j080j0'
3208	030j03g0'	040403g0'
3209	02q0w800'	0160q000'
3210	03g00000'	00j0q000'
3211	0010j0w0'	000gq000'
3212	040403g0'	080403j0'
3213	0160q000'	02q05000'
3214	00j0q000'	00001000'
3215	000gq000'	00g00004'
3216	040403j0'	030j03g0'
3217	03g0f800'	02q0w800'
3218	0ww0w800'	03g00000'
3219	0q801000'	0010j0w0'
3220	090w03j0'	040403j0'
3221	03g0f800'	03g0f800'
3222	016j8000'	0ww0w800'
3223	03g0f800'	0q801000'
3224	010j03j0'	090w03j0'
3225	01607000'	03g0f800'
3226	0fg09800'	016j8000'
3227	00j080j0'	03g0f800'
3228	0f0803j0'	0404003f'
3229	0080f800'	03g0q800'
3230	000f0800'	00048000'
3231	01606800'	02q0w800'

SORTIEREN 3	L1-10.2

VERWENDUNGSZWECK

Das Programm L1-10.2 sortiert eine Liste von Zahlen innerhalb des Speichers der Größe nach (kleinster Wert - größter Wert), entsprechend einer vorgegebenen Maske. Obwohl die sortierten Zahlen auf ihren ursprünglichen Speicherbereich gelegt werden, sind zusätzliche Zellen erforderlich. Durch die Maske wird festgelegt, welche Wortteile verglichen werden.
Das Programm arbeitet als Unterprogramm, das durch eine Aufruffolge des Hauptprogramms aufgerufen wird. Beim Sprung ins Unterprogramm muß die Anzahl der Worte in der Liste im Akkumulator stehen. Die Zellen, die dem Sprungbefehl folgen, müssen die Anfangsadresse der Liste und die Maske enthalten. Bei der Rückkehr ins Hauptprogramm stehen die Zahlen der Größe nach (kleinster Wert - größter Wert) sortiert auf dem ursprünglichen Speicherbereich der Liste.

SPEICHERBEDARF

3 Spuren

Benutzte Zwischenspeicher: keine

EINGABE

Im folgenden sind die Eingabevariablen für "Sortieren 3" in der Reihenfolge ihres Auftretens in der Aufruffolge gelistet:

1. N, Anzahl der zu sortierenden Werte in dezimaler Spur-/Sektorschreibweise.
2. So, Anfangsadresse der Worte in dezimaler Spur-/Sektorschreibweise.
3. Die willkürliche Maske, die den Sortiervorgang leitet, in hexadezimaler Schreibweise.

Die zu sortierenden Zahlen müssen positiv sein, und in aufeinanderfolgenden Zellen liegen.

AUFRUFFOLGE

Speicherplatz		Befehl	Adresse	
	-1	B	[ ]	Bringe N@29.
α		XR	L_o + 63	Speichere die Adresse des ersten Listenwortes im Unterprogramm.
	+1	XU	L_o	Sprung ins Unterprogramm.
	+2	XZ	S_o^o	Anfangsadresse der Liste.
,0000001'	+3	[ ]		Maske
	+4	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt N in den Akkumulator. (Jeder Befehl, der dies bewirkt, ist zulässig.) Der Befehl in α speichert die Adresse (α+2) des ersten Listenwortes ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms.

Zelle +2 enthält die Adresse S_o des ersten Listenwortes und Zelle +3 enthält das hexadezimale Wort, das die Maske darstellt. Nach +4 erfolgt Rückkehr aus dem Unterprogramm.

BEDIENUNG

1. Zeitbedarf:

   Durchschnittswerte: 32 Worte - 1,95 Min.
   64 Worte - 5,50 Min.
   128 Worte -12,87 Min.

2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten:

   Weder Eingabe- noch Ausgabeeinheiten sind erforderlich.

4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEISPIEL

Der Inhalt der Zellen 1514 bis 1645 ist der Größe nach zu sortieren unter alleiniger Berücksichtigung der Bitpositionen 3 bis 9 und 18 bis 23. "Sortieren 3" belege 0500 bis 0763, und das Hauptprogramm belege 3600 bis 4228.

Speicherplatz		Befehl	Adresse	
	3715	B	4221	Bringe N (96 Worte)
	3716	XR	0563	Listenadresse setzen.
	3717	XU	0500	Eingang ins Unterprogramm.
	3718	XZ	1514	Anfangsadresse der Liste.
,0000001	3719	1WJO	3W00	Maske
	3720	u.s.w.		
	.			
	.			
	4221	XZ	0132	N=1 Spur, 32 Sektoren.

Man beachte bitte, daß beim Sortieren nur die Bitpositionen verglichen werden, denen "1" Bits in der Maske entsprechen.

SORTIEREN 4	L1-10.3

VERWENDUNGSZWECK

Das Programm L1-10.3 sortiert "Einwort-" oder "Mehrwortintervalle". In einer Liste von Einwortintervallen wird jedes Wort als Vergleichswort betrachtet. In einer Liste von Mehrwortintervallen kann ein beliebiges Wort an der gleichen Stelle aller Intervalle als Vergleichswort festgelegt werden.
Der Sortiervorgang kann nach Belieben nach einem der drei folgenden Verfahren durchgeführt werden:

Verfahren I: Sortiere die Vergleichsworte der Größe nach (kleinster Wert - größter Wert) und stelle eine aufeinanderfolgende Liste der Adressen auf, in denen diese Vergleichsworte zu finden sind; d.h. die erste Zelle der neuen Liste enthält die Adresse des Schlüsselwortes mit dem kleinsten numerischen Wert aus der Originalliste.

Verfahren II: Sortiere die Vergleichsworte der Größe nach (kleinster Wert - größter Wert) und stelle nur eine aufeinanderfolgende Liste der sortierten Schlüsselworte her.

Verfahren III: Sortiere die Vergleichsworte der Größe nach (kleinster Wert - größter Wert) und stelle eine neue Liste der sortierten Intervalle her. Bei diesem Verfahren hat die neue sortierte Liste die gleiche Länge wie die Originalliste.

Anmerkung: Sind zwei Vergleichsworte gleich groß, so wird das erste davon angetroffene so behandelt, als habe es den kleineren Wert.

Das Programm wird durch eine Aufruffolge des Hauptprogramms angesprochen. Beim Sprung nach "Sortieren 4" muß die Anfangsadresse der Parameterliste im Akkumulator stehen. Bei der Rückkehr ins Hauptprogramm ist die Liste entsprechend den Angaben der Parameterliste sortiert und gespeichert.

SPEICHERBEDARF 3 Spuren

Benutzte Zwischenspeicher keine

PROGRAMMVARIABLEN

Die sieben Programmparameter müssen in folgender Reihenfolge angegeben werden:

1. bbbb, die Anfangsadresse der unsortierten Liste.
2. aaaa, die Anfangsadresse des Bereichs, auf den die sortierte Liste zu speichern ist.
3. nnnn, die Anzahl der "Intervalle" der Liste, in dezimaler Spur-/Sektorschreibweise.

4. wwww, die Anzahl der Worte pro "Intervall", in dezimaler Spur-/Sektorschreibweise.

5. kkkk, gibt an, das wievielte Wort eines jeden "Intervalls" als Vergleichswort zu nehmen ist.

6. t, die Nummer des Sortierverfahrens: 1 für Verfahren I, 2 für Verfahren II, oder 3 für Verfahren III.

7. mmmm mmmm, die Maske, in hexadezimaler Schreibweise.

Anmerkung: Die Maske darf keine "1" in der Vorzeichenstelle haben, da das Programm das Vorzeichen der Schlüsselworte nicht beachtet.

AUFRUFFOLGE

Die Anfangsadresse der Parameterliste wird mit β bezeichnet; β muß im Akkumulator stehen, wenn der Sprung ins Unterprogramm erfolgt.

Speicherplatz	Befehl	Adresse	
α	B	L [β]	Bringe die Anfangsadresse der Parameterliste in den Akkumulator.
+1	R	L_o	Rückkehradresse speichern.
+2	U	$L_o + 1$	Eingang ins Unterprogramm.
+3	u.s.w.		Rückkehr ins Hauptprogramm.

Die Liste für die Unterprogrammparameter ist:

Speicherplatz	Befehl
β	XZ bbbb
+1	XZ aaaa
+2	XZ nnnn
+3	XZ wwww
+4	XZ kkkk
+5	XZ 000t
+6	mmmmmmmm

Dabei ist: bbbb die Anfangsadresse der unsortierten Liste; aaaa die Anfangsadresse des Bereichs für die sortierte Liste; nnnn die Anzahl der "Intervalle" pro Liste; wwww die Anzahl der Worte pro "Intervall"; kkkk die Nummer des Wortes in allen Intervallen, das als Vergleichswort zu nehmen ist; t die Nummer des Sortierverfahrens; mmmmmmmm die Vergleichsmaske.

BEDIENUNG

1. Zeitbedarf:

   64 Einwortausdrücke in 27 Min. 15 Sek.
   32 Zweiwortausdrücke in 7 Min. 25 Sek.

2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten:

   Weder Eingabe- noch Ausgabeeinheiten sind erforderlich.

4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEISPIEL

Das Beispiel demonstriert die drei möglichen Sortierverfahren:

Beispielaufruffolge für ein ab 4100 gespeichertes "Sortieren 4".

```
3013        B 3129
3014        R 4100
3015        U 4201
3016        (Rückkehr nach hier)

3129       XZ 3236
3236       XZ 1000
3237       XZ 2020
3238       XZ 0005
3239       XZ 0003
3240       XZ 0002
3241       XZ 000t     t = 1, 2 oder 3
3242      000000WQ
```

Die obige Aufruffolge legt eine Liste von fünf 3-Wortintervallen fest, die ab 1000 gespeichert sind. Die Liste ist bezüglich des zweiten Wortes eines jeden Intervalls zu sortieren unter Benutzung der Maske 000000WQ. Die sortierte Liste ist ab 2020 zu speichern.

Originalliste			Datenbeispiel	Verf. I Ergebnis	Verf. II Ergebnis	Verf. III Ergebnis
(Sp/Se)	(HEX)	(Sp/Se)		(Sp/Se)	(HEX)	(HEX)
1000	7658	2020		1010	614	3846
1001	136	2021		1004	428	614
1002	3429	2022		1001	136	4948
1003	1876	2023		1013	346	1876
1004	428	2024		1007	170	428
1005	4470	2025				4470
1006	18208	2026				7658
1007	170	2027				136
1008	7720	2028				3429
1009	3846	2029				5650
1010	614	2030				346
1011	4948	2031				6312
1012	5650	2032				18208
1013	346	2033				170
1014	6312	2034				7720

BINÄRISIEREN GANZER ZAHLEN 1	L3-10.0

VERWENDUNGSZWECK

Das Programm L3-10.0 binärisiert eine positive 8-stellige Dezimalzahl. Linksseitige Nullen müssen mit angegeben werden.
Der Aufruf des Unterprogramms erfolgt durch eine Aufruffolge des Hauptprogramms. Beim Sprung ins Unterprogramm muß die Rückkehradresse gespeichert sein und das Argument im Akkumulator stehen.
Bei der Rückkehr ins Hauptprogramm steht die binärisierte Zahl im Akkumulator.

SPEICHERBEDARF

58 Sektoren (beginnend in 00)

Benutzte Zwischenspeicher: keine

EINGABE

Die Eingabevariable, eine positive 8-stellige Dezimalzahl muß bei q=31 im Akkumulator stehen, wenn der Sprung ins Unterprogramm erfolgt. Linksseitige Nullen müssen mit angegeben werden.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-1	I	[ ]	Eingabe des Argumentes @31.
α	R	L_o + 42	Speichere Rückkehradresse ins Unterprogramm.
+1	U	L_o	Eingang ins Unterprogramm.
+2	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α-1 bringt das Argument @31 in den Akkumulator. Der Befehl in α speichert die Rückkehradresse (α+2) ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Nach α+2 des Hauptprogramms erfolgt Rückkehr.

AUSGABE

Bei der Rückkehr ins Hauptprogramm steht die binärisierte Zahl @30 im Akkumulator.

BEDIENUNG

1. Zeitbedarf: 530 Millisekunden im Durchschnitt.
2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
3. Erforderliche Eingabe-/Ausgabeeinheiten: Sind nicht erforderlich.
4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

UMSPEICHERN 1	L4-10.0

VERWENDUNGSZWECK

Das Programm L4-10.0 speichert den Inhalt eines vorgegebenen Speicherbereichs auf einen anderen vorgegebenen Speicherbereich um. Die umgespeicherten Worte werden in keiner Weise verändert. Der Wortblock ist innerhalb des Speichers willkürlich verlegbar.
Das Programm wird angesprochen durch eine Aufruffolge, die liefern muß: 1. die Anfangsadresse des Datenblocks, 2. die neue Anfangsadresse, 3. die Anzahl der umzuspeichernden Worte. Bei der Rückkehr ins Hauptprogramm ist der Datenblock umgespeichert.

SPEICHERBEDARF 56 Sektoren

Benutzte Zwischenspeicher keine

EINGABE

Die Eingabevariablen in folgender Reihenfolge:

1. M_i, die Anfangsadresse des umzuspeichernden Wortblocks in dezimaler Spur-/Sektorschreibweise.
2. M_t, die Anfangsadresse des Bereichs, auf den die Worte umzuspeichern sind, in dezimaler Spur-/Sektorschreibweise.
3. N, die Anzahl der umzuspeichernden Worte in dezimaler Spur-/Sektorschreibweise.

Diese Kontrolldaten müssen sich in den drei Zellen befinden, die auf den Sprungbefehl der Aufruffolge folgen.
Der Wortblock muß in aufeinanderfolgenden Zellen gespeichert sein und darf in jeder Richtung umgespeichert werden; d.h., M_t darf größer oder kleiner als M_i sein.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
α	R	L_o	Adresse der Anfangsadresse des Datenblocks ins Unterprogramm setzen.
+1	U	L_o	Sprung ins Unterprogramm.
+2	Z	M_i	Alte Anfangsadresse des Datenblocks.
+3	Z	M_t	Neue Anfangsadresse des Datenblocks.
+4	Z	NNNN	Anzahl der umzuspeichernden Worte.
+5	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α speichert die Adresse $(\alpha+2)$ ins Unterprogramm. Der Befehl in $\alpha+1$ bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Der Befehl in $\alpha+2$ enthält die Anfangsadresse des umzuspeichernden Wortblocks; der Befehl in $\alpha+3$ enthält die Anfangsadresse des neuen Bereichs, also des umgespeicherten Datenblocks; $\alpha+4$ enthält die Anzahl der umzuspeichernden Worte des Blocks.

BEDIENUNG

1. Zeitbedarf: ca. 18 Sek. pro Spur.
2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.
3. Erforderliche Eingabe-/Ausgabeeinheiten: Sind nicht erforderlich.
4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

BEISPIEL

Es sind 200 Worte eines Speicherbereichs, der in 700 beginnt, umzuspeichern auf einen Bereich der in 4200 beginnt. "Umspeichern 1" liege ab 300.

Speicherplatz	Befehl	Adresse	Anmerkung
0500	R	0300	L_o
0501	U	0300	L^o
0502	Z	0700	M_i^o
0503	Z	4200	M_i^i
0504	Z	0308	$(308_{64} = 200_{10})$
0505	u.s.w.		

STREIFENDOPPLER 1	L4-11.0

VERWENDUNGSZWECK

Das Programm L4-11.0 doppelt beliebige, für den LGP-21 kodierte Streifen. Der Streifen wird über den Schnelleser gelesen und über den Schnellstanzen wieder ausgestanzt. Der so erhaltene Streifen ist ein getreues Doppel des Originalstreifens, mit der Ausnahme, daß alle Löschcodes weggelassen wurden.

SPEICHERBEDARF 1 Spur

Benutzte Zwischenspeicher keine

EINGABE

"Streifendoppler 1" verarbeitet jeden Streifen, der für den LGP-21 kodiert wurde. Der Streifen muß über den Schnelleser eingelesen werden.

AUFRUFFOLGE

Eingang ins Programm erfolgt durch einen Sprung nach L_o+1, wobei L_o die Anfangsadresse des Programms ist. Auf dem Programmstreifen ist ein Sprung nach L_o+1 vorgesehen.

AUSGABE

Nach Sprung zur Anfangsadresse beginnt das Programm den zu doppelnden Streifen einzulesen und über den Schnellstanzer auszustanzen. Der ausgegebene Streifen ist eine genaue Kopie des gelesenen Streifens, mit der Ausnahme, daß Löschcodes weggelassen wurden. Das Lesen und Stanzen wird fortgesetzt, bis der gesamte Streifen gedoppelt ist.

BEDIENUNG

1. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

2. Erforderliche Eingabe-/Ausgabeeinheiten:

   Der Schnelleser wird für die Eingabe benötigt und der Schnellstanzer für die Ausgabe. Beide werden vom "Streifendoppler 1" ausgewählt.

3. Programmsprungtasten: Keine

4. Fehlerstops: Keine

TABELLENABSUCHEN 1	L7-10.0

VERWENDUNGSZWECK

Das Programm L7-10.0 sucht aus einer Tabelle von Werten die Adresse des Wertes, der mit dem vorgegebenen Vergleichswort übereinstimmt. Die Tabelle muß in aufsteigender Folge gespeichert sein und wenigstens zwei Zahlen enthalten. Alle Daten müssen in Festkommaform vorliegen. Das Programm sucht nach der "Halbierungsmethode". D.h., das Vergleichswort wird zuerst mit dem Wort in der Mitte der Tabelle verglichen. Bei Gleichheit ist die Suche beendet. Ist das Vergleichswort größer, so wird es mit dem Wort verglichen, das in der Mitte zwischen Mitte und Ende der Tabelle liegt. Ist das Kontrollwort kleiner als das mittlere gewesen, so wird es mit dem Wort verglichen, das in der Mitte zwischen Anfang und Mitte der Tabelle liegt. Die Suche wird in dieser Weise fortgesetzt, bis Gleichheit vorhanden ist.
Der Aufruf des Unterprogramms erfolgt durch eine Aufruffolge des Hauptprogramms. Beim Sprung ins Unterprogramm muß das Vergleichswort im Akkumulator stehen, und die Aufruffolge muß die Anfangsadresse der Tabelle und die Anzahl der darin enthaltenen Worte liefern. Beim Sprung ins Hauptprogramm steht die Adresse, bei der Gleichheit gefunden wurde, im Akkumulator.

SPEICHERBEDARF	3 Spuren
Benutzte Zwischenspeicher	die Sektoren 01, 27 und 44 der Spur 63.

EINGABE

Die Eingabevariable ist ein Vergleichswort, C. Dieses Vergleichswort kann jedes beliebige Wort des Speichers sein und muß im Akkumulator stehen, wenn der Sprung ins Unterprogramm erfolgt.
Alle Daten müssen in Festkommaform vorliegen. Die Tabelle muß aus mindestens zwei Worten bestehen. Die Worte müssen in aufeinanderfolgenden Zellen gespeichert sein und zwar in aufsteigender numerischer Reihenfolge.
Die Zellen hinter dem Sprungbefehl der Aufruffolge müssen enthalten:

1. NNNN, die Anzahl von Worten der Tabelle in dezimaler Spur-/Sektorschreibweise, d.h. modulo 64.
2. T_o, die Anfangsadresse der Tabelle in dezimaler Spur-/Sektorschreibweise.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
-1	B	[ ]	Bringe C in den Akkumulator.
α	R	L_o + 31	Adresse von NNNN ins Unterprogramm speichern.
+1	U	L_o	Eingang ins Unterprogramm.

Speicherplatz	Befehl	Adresse	
α +2	XZ	NNNN	Größe der Tabelle.
+3	XZ	T_o	Anfangsadresse der Tabelle.
+4	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl α-1 bringt C in den Akkumulator. (Jeder Befehl, der dies bewirkt, kann benutzt werden.) Der Befehl in α speichert die Adresse (α+2) ins Unterprogramm. Der Befehl in α+1 bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Zelle α+2 enthält NNNN, modulo 64; XZ0132 bedeutet z.B. 96 Tabellenworte. Der Befehl in α+3 enthält T_o. Nach α+4 erfolgt Rückkehr aus dem Unterprogramm.

AUSGABE

Bei der Rückkehr ins Hauptprogramm steht die Adresse des Wortes, bei dem Gleichheit mit dem Vergleichswort festgestellt wurde, @ 29 im Akkumulator.

BEDIENUNG

1. Zeitbedarf:

   Die Operationszeit hängt von der Länge der Tabelle und der Lage des gesuchten Wortes ab.

2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, decimal, in Kodierungsblattformat.

3. Erforderliche Eingabe-/Ausgabeeinheiten: sind nicht erforderlich.

4. Überlauf:

   Wird beim Eingang nicht beachtet und beim Ausgang nicht zurückgesetzt.

5. Programmstops:

Speicherzelle	Bedeutung	Behebung
L_o+0130	Kein Wort der Tabelle stimmt mit C überein, oder C liegt außerhalb der Folge.	Nicht fortfahren. Daten müssen verbessert werden.
L_o+0218	Entweder wurde C geändert oder die Inhalte der Zellen L_o+247 bis L_o+259 würden geändert, als das Unterprogramm geändert wurde, um mit gleichem C und gleichen Tabellenwerten fortzufahren. (Siehe unter "Bemerkungen".)	Nicht fortfahren.

BEMERKUNGEN

Sollen C und die Tabellen wiederholt benutzt werden, ohne daß etwas davon geändert wird, so sind die folgenden Programmände-

rungen vorzunehmen nach der ersten Suche, um die folgenden Suchen zu beschleunigen; dabei ist L_o die Anfangsadresse des Unterprogramms:

1. Ändere den Inhalt von L_o+0033 von N0019 auf U0054.
2. Ändere den Inhalt von L_o+0056 von E0214 auf U0149.

Die Inhalte der Zellen L_o+0007 und L_o+0247 bis L_o+0259 müssen für folgende Suchen aufbewahrt werden.

SCHLEIFENZÄHLER 1	L8-10.0

VERWENDUNGSZWECK

Das Programm L8-10.0 erhöht die Operandenadressen beliebiger vorgegebener Befehle, durchläuft eine vorgegebene Befehlsfolge so oft als angegeben, und setzt die Operandenadressen auf ihren ursprünglichen Wert zurück. Die angegebenen Adressen werden vor jedem Schleifendurchlauf um 1 erhöht; ebenso werden alle Befehle der vorgegebenen Folge bei jedem Schleifendurchlauf ausgeführt.
Der Aufruf erfolgt durch eine Aufruffolge des Hauptprogramms. Die Aufruffolge muß liefern: die Anfangsadresse der Befehlsfolge, die Anzahl der Schleifendurchläufe. Bei der Rückkehr ins Hauptprogramm ist die vorgesehene Anzahl von Durchläufen erfolgt, und alle Adressen, die erhöht worden waren, sind auf ihre ursprünglichen Werte zurückgesetzt.

SPEICHERBEDARF 1 Spur

Benutzte Zwischenspeicher keine

EINGABE

Die wiederholt auszuführende Gruppe von Befehlen muß im Hauptprogramm unmittelbar vor der Aufruffolge stehen. Alle Befehle können benutzt werden.
Die Operandenadressen aller negativen Befehle der Gruppe werden vor jedem Schleifendurchlauf um 1 erhöht; d.h., alle Befehle, denen 800 vorausgeht. Der Befehl 800A2000 würde beispielweise vor dem ersten Schleifendurchlauf geändert zu 800A2001. Die Adressen positiver Befehle werden nicht verändert.
Die Aufruffolge muß folgende Informationen in der angegebenen Reihenfolge liefern, wenn der Sprung ins Unterprogramm erfolgt:

1. S_o, die Anfangsadresse der Gruppe in dezimaler Spur-/Sektorschreibweise innerhalb eines I-Befehls.
2. X, die Anzahl von Schleifendurchläufen in dezimaler Spur-/Sektorschreibweise; d.h., modulo 64 innerhalb eines N-Befehls.

AUFRUFFOLGE

Speicherplatz	Befehl	Adresse	
α	R	L_o	Anfangsadresse der Befehlsgruppe ins Unterprogramm setzen.
+1	U	L_o	Eingang ins Unterprogramm.
+2	I	S_o	Anfangsadresse der zu durchlaufenden Gruppen.
+3	N	XXXX	Schleifenzahl.
+4	u.s.w.		Rückkehr ins Hauptprogramm.

Der Befehl in α speichert die Adresse $(\alpha+2)$ ins Unterprogramm. Der Befehl in $\alpha+1$ bewirkt einen Sprung nach L_o, der Anfangsadresse des Unterprogramms. Der Befehl in $\alpha+2$ enthält die Adresse S_o. Der Befehl in $\alpha+3$ enthält den Wert X. Nach $\alpha+4$ erfolgt Rückkehr aus dem Unterprogramm.

BEDIENUNG

1. Zeitbedarf:

   Die erforderliche Zeit ist für jede Schleife variabel und hängt ab vom Umfang und von der Anzahl der Durchläufe einer Schleife.

2. Programmeingabe:

   Der Streifen enthält das Programm in zwei Fassungen:
   a) willkürlich verlegbar, hexadezimal, zulässig für Programmeingabe 1 und
   b) willkürlich verlegbar, dezimal, in Kodierungsblattformat.

3. Überlauf:

   Wird vom Programm nicht getestet.

BEMERKUNG

Die Operandenadresse eines jeden negativen Befehls innerhalb einer Schleife wird erhöht.
Wird ein Ausgang aus dem Unterprogramm vorgenommen, bevor die festgelegte Schleifenzahl durchlaufen ist, so muß der Inhalt von Zelle L_o+2 wie folgt geändert werden:

alter Inhalt	neuer Inhalt
U (L_o + 18)	U (L_o + 3)

Es ist möglich, aus einer Schleife, die gerade ausgeführt wird, herauszuspringen (beispielsweise in ein anderes Unterprogramm), wenn nur das Unterprogramm, zu dem gesprungen wird, keine modifizierbaren (negativen) Befehle enthält.

BEISPIEL

Es ist der Wert L zu errechnen entsprechend der Formel:

$$L = \frac{(2x + y)\, z}{w}$$

Gegeben:

Zehn Werte für x in den Zellen 3100-3109 bei q = 10.
Zehn Werte für y in den Zellen 3200-3209 bei q = 20.
Zehn Werte für z in den Zellen 3300-3309 bei q = 15.
Zehn Werte für w in den Zellen 3400-3409 bei q = 10.

Zu speichern:

Zehn berechnete Werte von L in den Zellen 3500-3509 bei q = 23.

Verfügbare Konstanten:

2 @ 15 in Zelle 3000.
1 @ 5 in Zelle 3001.
1 @ 3 in Zelle 3002.

"Schleifenzähler 1" sei gespeichert in Spur 40.

Programmeingabe Schlüsselwort	Speicher Nr.	Befehl Oper. Adr.	Inhalt der Adresse	Bemerkungen
;0002000' /0000000'				
	2000	B 3000'	2 @ 15	
	01	800M 3100'	X @ 10	
	02	D 3001'	1 @ 5	
	03	800A 3200'	Y @ 20	
	04	D 3001'	1 @ 5	Problem
	05	800M 3300'	Z @ 15	
	06	800D 3400'	W @ 10	
	07	M 3002'	1 @ 3	
	08	800C 3500'	L @ 23	
	09	R 4000'	L_0	
	10	U 4000'	L_0	
	11	I 2000'	erster Bef.	Aufruf des
	12	N 0010'	Anzahl der Schleifen	Schleifen-zählers.
	13	Z 0000'	Stop	

RECHNER-KONTROLLIERTE DRUCKSCHLEIFE

Ein Ausgabeunterprogramm ist oft als Rechner-kontrollierte Druckschleife programmiert. Da es nötig sein würde, den Zähler zwischen Druckbefehlen während einer normalen Druckschleife zu speichern, würde höchstens einmal pro Trommelumdrehung gedruckt werden können, was einer Druckgeschwindigkeit von ca. 20 Zeichen pro Sekunde entspricht. Die Programmierungsmethode stellt bei der Ausgabe über die Lochstreifenschreibmaschine kein Problem dar. Wird dagegen der Schnellstanzer benutzt, so ist eine schnellere Druckschleife erwünscht. Die unten gezeigte Methode gestattet es, mit einer Geschwindigkeit von ca. 40 Zeichen pro Sekunde zu stanzen (2 Druckbefehle pro Trommelumdrehung).

Beispiel:

Die Anzahl der auszudruckenden Zeichen sei als Zähler bei q = 30 gespeichert. Das folgende Programm druckt ein Zeichen (Wagenrücklauf zum Beispiel) so oft, wie der Zähler angibt.

Speicherplatz	Befehl	Adresse	Bemerkungen
0000	C	0050	Formal wegspeichern
0001	S	( )	Anzahl
0002	U	0059	Nulltest für die Anzahl
0003	E	0053	43WW WWWQ
0004	XP	0200	
0005	A	0055	GJOO 0002
0006	T	0056	

Speicherplatz		Befehl	Adresse	Bemerkungen
	0007	U	0060	Ausgang
	....	.	....	
,000 0004	0049	43WW	WWWQ	
	0050	(	)	Formal weggespeichert
	0051	GJOO	0002	
	0052			Nicht benutzt
,000 0003	0053	43WW	WWWQ	
	0054			Nicht benutzt
	0055	GJOO	0002	
	0056	E	0049	
	0057	XP	0200	
	0058	A	0051	GJOO 0002
	0059	T	0003	
	0060	U	( )	Ausgang

Eurocomp GmbH · Elektronische Rechenanlagen · 495 Minden 2
Schillerstr. 72 · Tel. 0571/83421 · Telex 097867 (schofae minden)

Printed in Germany D 013/465/2-01