Programmieren in der IBM/360 ASSEMBLER-Sprache

IBM

ASS-PU

Text 1

Programmierte Unterweisung

IBM Schule für Datenverarbeitung

Programmieren in der IBM /360 Assembler-Sprache (ASS-PU)

Eine Programmierte Unterweisung

Text 1

Aufgaben

Programmiert von:

Gerhard K. Bulling
Helmut Hertweck
Heinz Laubner
Rosa Maria Schartz

Diese Lehrgangsunterlagen wurden ausschließlich für Unterrichtszwecke geschaffen; sie sind in erster Linie auf klare Darstellung des Lehrstoffes ausgerichtet. Eine Gewährleistung für die Verwendung bei der praktischen Arbeit kann nicht übernommen werden; dafür stehen jeweils die neuesten Veröffentlichungen der IBM System- und Programmliteratur zur Verfügung.

© Copyright by IBM Deutschland GmbH 1972
Änderung des Textes bleibt vorbehalten

IBM Form R12-1731-1 (Text 1 bis Text 4 und Arbeitsmittel)
IBM Form R12-1732-1 (Arbeitsmittel)

Inhalt der ASS-PU

Vorwort zur ASS-PU

1. Die Programmierte Unterweisung ASS-PU

2. Einführung in die Methode der Programmierten Unterweisung

4 Textbände:

Text 1 : Aufgaben

Text 2 : Datenfluß, Verschlüsselung der Daten
Adressierung des Hauptspeichers
Maschineninstruktions-Formate

Text 3 : Assembleranweisungen

Text 4 : Maschineninstruktionen

Arbeitsmittel

(Ein ausführliches Inhaltsverzeichnis zu den einzelnen Textbänden findet sich zu Beginn der Arbeitsmittel).

V o r w o r t

Die Programmierte Unterweisung ASS-PU

1. Unterlagen

Diese Programmierte Unterweisung (PU) befaßt sich mit der Programmierung des IBM Systems /360 in der Assembler-Sprache, einer symbolischen Programmiersprache.

Voraussetzung für die Durcharbeitung der ASS-PU ist der Lernstoff der CG -PU. Besitzen Sie dieses Wissen nicht, ist es sinnlos, die ASS-PU zu beginnen.

Die ASS-PU besteht aus 4 Textbänden und den Arbeitsmitteln.

Die Textbände und die Arbeitsmittel sind zu einem Paket zusammengefaßt und unter der Form-Nummer IBM Form R 12-1731 zu beziehen. Darüberhinaus sind die Arbeitsmittel auch getrennt unter der Form-Nummer IBM Form R 12-1732 erhältlich.

In den T e x t b ä n d e n wird der Lernstoff - nach pädagogischen Gesichtspunkten geordnet - dargeboten. Diese Textbände enthalten den Stoff in folgender Gliederung:

Text 1 : Aufgaben

Text 2 : Datenfluß, Verschlüsselung der Daten, Adressierung des Hauptspeichers, Instruktionsformate

Text 3 : Assembleranweisungen

Text 4 : Maschineninstruktionen

Zu Text 2 :

In diesem Band werden Ihnen Kenntnisse über die Speicherung und die Adressierung der Daten und Instruktionen im Hauptspeicher vermittelt. - Diese Kenntnisse über die Konzeption des Systems /360 müssen beim Programmieren in der Assembler-Sprache berücksichtigt werden.

Zu Text 3 :

Dieser Band umfaßt die Regeln der Assembler-Sprache und verschiedene Assemblerinstruktionen, mit deren Hilfe die Umwandlung des Programms gesteuert und die Definitionen der Daten, Felder und Bereiche im Hauptspeicher vorgenommen werden.

Zu Text 4 :

Dieser Band behandelt Instruktionen, die der Verarbeitung der Daten in der Zentraleinheit dienen. Sie lernen z.B. die Instruktionen für das Übertragen, Vergleichen, Addieren, Multiplizieren usw. von Daten im Hauptspeicher kennen.

Zu Text 1 :

Der Band 1 enthält 15 Aufgaben. Anhand dieser Aufgaben überprüfen Sie selbst, ob Sie die Kenntnisse, die Sie sich aus den Bänden 2 - 4 angeeignet haben, auch beim Schreiben eines Programms anwenden können.

Die Arbeitsmittel enthalten verschiedene Arbeitsunterlagen (z.B. Aufgaben, Merkblätter, Übungen). Diese sind in Verbindung mit den Textbänden 1 - 4 nach jeweiliger Aufforderung durch das Programm zu verwenden. - Die Arbeitsmittel verbleiben nach dem Unterricht bei Ihnen.

2. Lernziel

Nach dem Durcharbeiten des Lernstoffs sollten Sie grundlegende Kenntnisse über die Assembler-Programmiersprache erworben haben. Insbesondere sollten Sie die behandelten Assembleranweisungen und Maschineninstruktionen beherrschen und über Maschineninstruktionsformate, Verschlüsselung der Daten und Adressierung des Hauptspeichers orientiert sein.
Schließlich sollten Sie aufgrund der erarbeiteten Aufgaben und Übungen in der Lage sein, ähnliche Aufgaben aus dem Bereich der Datenverarbeitung selbständig in der Assembler-Sprache zu programmieren.

3. Lernzeit

Zur Erarbeitung des Lernstoffs in dieser Programmierten Unterweisung benötigen Sie ca. 50 bis 120 Stunden. Die benötigte Lernzeit ist abhängig von Ihren Vorkenntnissen und Ihrem individuellen Lerntempo.

4. Abschlußfragen

Nachdem Sie die PU durchgearbeitet haben, können Sie Ihre gesamten Kenntnisse überprüfen, indem Sie die Abschlußfragen zunächst selbständig lösen und dann mit der vorgegebenen Lösung vergleichen.

Die Abschlußfragen und deren Lösung finden Sie in den Arbeitsmitteln.

5. Ablauf des Unterrichts

Für die Durcharbeitung der ASS-PU gibt es zwei verschiedene Lernwege:

1. Erarbeiten des Lernstoffs von den Aufgaben her (Lernweg 1).

2. Erarbeiten des Lernstoffs ohne Lösen von Aufgaben (Lernweg 2).

Lernweg 1

Der Aufgabenband (Text 1)

Immer wenn zur Lösung einer Aufgabe entsprechende Kenntnisse aus den Bänden Text 2, Text 3 oder Text 4 erforderlich sind, werden Sie von Text 1 in die entsprechenden Bände verwiesen. Sie erarbeiten sich dann in dem entsprechenden Band den angegebenen Lernstoff und kehren danach wieder in den Aufgabenband zurück.

Der Lernstoff der Bände Text 2, Text 3 und Text 4 wird also nicht in der bestehenden Reihenfolge erarbeitet, sondern jeweils durch die Erfordernisse des Aufgabenbandes Text 1 ausgewählt.

Lernweg 2

Bei der Erarbeitung des Lernstoffs ohne Aufgaben beginnen Sie mit Text 2 und arbeiten diesen von Anfang bis Ende durch. Danach folgen entsprechend Text 3 und Text 4.

Hierbei erfolgt also keine ständige Absicherung des angeeigneten Wissens durch selbständiges Lösen von Aufgaben. (Ausnahme: Pflichtaufgaben 5, 13, 14 und 15).

Den Lernweg 2 dürfen Sie nur dann gehen, wenn Sie bereits selbständig Programme für die IBM 1400-er- bzw. 7000-er-Systeme in SPS (nur bei 1401) bzw. AUTOCODER geschrieben haben.

In Ihren Arbeitsmitteln finden Sie eine "Übersicht über Aufbau und Inhalt dieser Programmierten Unterweisung". Anhand dieser Übersicht können Sie jederzeit feststellen, an welchem Punkt der Durcharbeitung Sie sich befinden und wie es weitergeht. Diese Übersicht gilt für beide Lernwege.

Einführung in die Methode der Programmierten Unterweisung.

Die "Programmierte Unterweisung" (PU) ist ein neuartiges Lehr- und Lernverfahren. - Der Lernstoff wird in kleinen Einheiten, die Lernelemente (LE) genannt werden, dargeboten. Jedes Lernelement umfaßt nur wenige Sätze.

Die meisten Lernelemente enthalten im Text eine Lücke. Diese soll vom Lernenden ausgefüllt werden. Andere Lernelemente enden mit einer Frage oder einer Aufgabe, die vom Lernenden beantwortet bzw. gelöst werden soll. - Dies erfordert, daß der Lernende ständig mitdenkt, da nur so die richtige Antwort bzw. Lösung gefunden werden kann. Die Programmierte Unterweisung bedingt somit eine ständig aktive Mitarbeit.

Die Informationen für die Antwort können entnommen werden:

a) dem betreffenden Lernelement,

b) einem vorausgegangenen Lernelement,

c) einer Abbildung, einem Merkblatt, oder einer Aufgabe,

d) einer Kombination der o. a. Möglichkeiten.

Die Lernelemente stehen links auf der Seite in der breiten Textspalte (Lerntextspalte). Die schmalere rechte Spalte ist für die Antworten vorgesehen (Antwortspalte). Die Antworten sind jeweils erst nach dem Lernelement in dieser Spalte gedruckt.

Mit einer Lochkarte soll der Lernende die vorgegebene Antwort solange verdecken, bis er seine eigene Antwort erarbeitet hat. Anschließend vergleicht er seine Antwort mit der seines Programmes. Dadurch prüft der Lernende nach jedem Lern- und Arbeitsschritt selbst, ob er den Lernstoff verstanden hat. Sie, als der Lernende, kontrollieren Ihren eigenen Lernerfolg.

Ein wesentlicher Vorteil der PU ist es, daß der Lernende das Lerntempo seinen persönlichen Gegebenheiten anpassen kann.

Die PU führt nur dann zum Erfolg, wenn Sie folgende Hinweise beachten:

1. Lesen Sie die einzelnen Lernelemente sorgfältig durch und e r a r b e i t e n Sie sich die jeweiligen Antworten.

2. Beginnen Sie mit dem nächsten Lernelement erst dann, wenn Sie das vorangegangene restlos verstanden haben.

3. Das Überspringen einzelner Lernelemente ist nicht sinnvoll, denn dadurch wird der Lernfluß und das allmähliche Erarbeiten des Lernstoffs gestört.

4. Wiederholungen sind bei aufmerksamem Studium nicht erforderlich, solange sie vom Programm nicht ausdrücklich verlangt werden. Ist ein Lerngebiet besonders wichtig, so wird es auch entsprechend intensiv geübt.

5. Versuchen Sie nicht, auswendig zu lernen, sondern zu verstehen.

6. Schreiben Sie Ihre Antworten bitte n i c h t in die entsprechenden Lernelemente. Dadurch kann das Lernprogramm evtl. auch noch von anderen Lernenden benutzt werden. - Ist eine schriftliche Antwort überhaupt erforderlich, so schreiben Sie diese bitte auf ein Extrablatt.

7. Für den Fall, daß Ihre Antwort nicht mit der vorgegebenen übereinstimmt, ergeben sich folgende Möglichkeiten:

   a) Ihre Antwort ist ähnlich der vorgegebenen und damit sinngemäß richtig. - Gehen Sie zum nächsten Lernelement über.

   b) Sie stellen aufgrund der vorgegebenen Antwort einen Fehler fest. - Lesen Sie das entsprechende Lernelement bzw. die vorangegangenen Lernelemente nochmals gründlich durch.

8. Die Arbeitsmittel verwenden Sie bitte nach Aufforderung durch das Programm in Verbindung mit dem Lerntext.

Auf den folgenden Seiten können Sie sich am Beispiel einiger Lernelemente mit der Methode der PU vertraut machen.

I Die Lernelemente bestehen meistens aus nur wenigen Sätzen. Die meisten Lernelemente enthalten im Text eine _____, die(zumindest gedanklich) von Ihnen auszufüllen ist.

Lücke

Es ist ein besonderer Vorzug der Programmierten Unterweisung, daß Sie Ihre Antworten sofort selbst überprüfen können. Es ist natürlich notwendig, daß die vorgegebene Antwort solange abgedeckt bleibt, bis Sie diese gefunden haben. Entnehmen Sie zu diesem Zweck Ihren Arbeitsmitteln die Abdeckkarte.

II Verdecken Sie mit dieser Karte die Antwortspalte so, daß Sie die Antwort zu diesem _____ nicht lesen können.

Führen Sie nun die Karte weiter bis zum nächsten Lernelement. Dann wird die Programmantwort sichtbar, und Sie können diese mit der Ihren vergleichen.

Lernelement

III Einige Lernelemente erfordern mehrere Antworten. - Decken Sie die erste Antwort auf, sobald Sie die erste _____ ausgefüllt haben; nach dem Ausfüllen der zweiten Lücke decken Sie die zweite _____ auf usw.

Lücke

Antwort

IV Besteht die Antwort aus mehreren Worten, etwa einem Artikel und einem Substantiv, so wird durch "(WW)" darauf aufmerksam gemacht.

Beispiel:
Nach einem Lernelement folgt in der Antwortspalte
__(WW)__.

die Antwort

V Einige Antworten bestehen aus einer Zahl oder Ziffer; das wird durch ein "(Z)" gekennzeichnet.

Beispiel:
Ein Tag hat __(Z)__ Stunden.

24

VI Andere Antworten erfordern mehrere Zahlen; das wird durch "(ZZ)" angezeigt.

Beispiel:
Zwischen 20 und 30 liegen die ungeraden Zahlen
__ZZ__.

21, 23, 25, 27, 29

VII Manche Antworten bestehen aus einer Zahl (Ziffer) und einem Wort bzw. aus einem Wort und einer Zahl (Ziffer). - Dies wird durch "(ZW)" bzw. "(WZ)" angezeigt.

Beispiel:
Ein Auto braucht auf 100 km 8 Liter Benzin. Auf 500 km braucht es dann __(ZW)__ Benzin.

40 Liter

VIII In manchen Lernelementen werden mehrere Antworten vorgegeben, aus denen die richtige Antwort auszuwählen ist. Die verschiedenen Antworten stehen in Klammern vor der Lücke und sind durch ein Komma getrennt.

Beispiel:
Dieses Lernelement (bietet, bietet nicht) _____ die Möglichkeit, eine aus mehreren vorgegebenen Antworten auszuwählen.

Hier wird die Art der Antwort nicht weiter gekennzeichnet.

bietet

IX Es ist auch möglich, daß auf eine Frage im Lernelement mehrere richtige Antworten gegeben werden können. - Die einzelnen Antworten sind hierbei durch Schrägstriche voneinander abgegrenzt.

Beispiel:
Eine Lücke mit einem (Z) in einem Lernelement bedeutet, daß eine _____ zu ergänzen ist.

Zahl/Ziffer

Beginnen Sie nun bitte mit der Erarbeitung des Lernstoffs in dieser Programmierten Unterweisung. - Wir wünschen Ihnen viel Erfolg!

Programmieren in der IBM /360 Assembler-Sprache (ASS-PU)

Eine Programmierte Unterweisung

Text 1

Aufgaben

Inhalt Text 1

		Seite
Einleitung zu Text 1		
Aufgabe	1	13
Aufgabe	2	45
Aufgabe	3	69
Aufgabe	4	97
Aufgabe	5	119
Aufgabe	6	153
Aufgabe	7	211
Aufgabe	8	255
Aufgabe	9	293
Aufgabe	10	307
Aufgabe	11	327
Aufgabe	12	337
Aufgabe	13	421
Aufgabe	14	443
Aufgabe	15	467

Einleitung zu Text 1

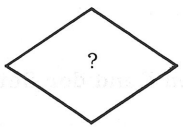
Die vorliegenden Aufgaben spannen den Bogen von der Einübung programmiertechnischen Rüstzeugs bis zur Vermittlung grundlegender Kenntnisse im Programmieren mit der Assembler-Sprache.

Der Fortgang in den Aufgaben ist ein Fortgang vom Einfachen zum Komplexen, wenn sich das Problem ausweitet, oder ein Variieren, sofern sich das Problem nicht grundsätzlich, sondern nur unter einigen Gesichtspunkten verschiebt.

So bilden die Aufgaben 1 bis 4, die dem übenden Umgang mit grundlegenden Operationen und gewissen Programmierhilfsmitteln dienen, eine sich ausweitende Folge von Aufgaben. Die folgenden Aufgaben variieren in der Behandlung der verschiedenen elementaren Rechenoperationen wie der Addition, Subtraktion, Multiplikation und Division. Die Aufgaben 13 bis 15 zeigen am Beispiel der Verarbeitung vieler gleichartiger Felder, wie beim Programmieren die Kenntnis von Verfahrensweisen unbedingt erforderlich ist.

Im Aufgabenband sind die einzelnen Lernelemente nicht mehr (wie in den Bänden 2 - 4) numeriert, und die Antworten stehen nicht rechts neben, sondern unterhalb der Lernelemente (unterhalb einer Reihe von Sternen (* * * * *)).

Außerdem werden im Aufgabenband zusätzlich folgende Symbole verwendet:

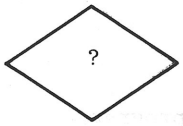
a) 

b) 

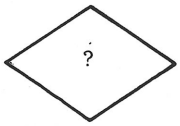
c) 

d) 

e) 

Zu a) 

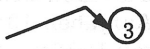
Dieses Zeichen wird verwendet, wenn Sie sich für eine von mehreren Möglichkeiten im Lernfortgang entscheiden sollen.



A Erstellen des Programmablaufplans ohne Hilfe

B Erstellen des Programmablaufplans mit Hilfe

* * * * *

A  , Seite 112

Entscheiden Sie sich für die Möglichkeit A, so dürfen Sie bis zum Punkt (3) der Seite 112 springen. Im anderen Fall (Möglichkeit B) geht es normal weiter.

Zu b) 

An dem vorstehenden Beispiel (zu a) haben Sie sicher erkannt, daß dieses Zeichen "springe nach ....." bedeutet.

Zu c) 

Die Stelle, zu der gesprungen wird, ist am linken Rand der Seite mit einem Kreis (◯ mit Nummer) versehen.

Sie erinnern sich:
Im Programmablaufplan wird dieser Kreis u. a. als Schrittkennzeichnung verwendet. In unserem Lernprogramm hat er eine analoge Funktion zur Kennzeichnung bestimmter Lernelemente oder Texte.

Zu d) 

Dieses Zeichen bedeutet ebenfalls "springe nach .....". Hierbei handelt es sich jedoch nicht um einen "Vorwärtssprung" (), sondern um einen "Rücksprung".

Beispiel:
Falls Sie die Aufgabe selbständig gelöst haben und mit der gezeigten Lösung nicht zurechtkommen  , Seite 210.

Zu e) 

Dieses Sprungzeichen wird verwendet, wenn innerhalb der Bände 1 - 4 zu springen ist.

Beispiel:

 Band 2, Seite 5 (Kapitel I)

 Band 4, Seite 134 (CP)



Dieses Zeichen wird als Aussteigepunkt verwendet. Sie finden es nur in den Bänden 2 - 4. Es zeigt an, daß hier ein Lernabschnitt, zu dem aus dem Aufgabenband verwiesen wurde, zu Ende ist. Dieser Aussteigepunkt ist mit einem Sprungzeichen (e) gekoppelt, das anzeigt, wo Sie fortfahren müssen.

Hinweis für die Zeiteinteilung (Lernweg 1):

Die einführenden Aufgaben (etwa Aufg. 1-4) sind recht einfach. Die folgenden Aufgaben werden allmählich umfangreicher, schwieriger und damit auch zeitaufwendiger. Das gilt ganz besonders für die Aufgaben 11 bis 15. - Als Faustregel können Sie sich merken, daß Sie für die Aufgaben 11 bis 15 (und den dazugehörenden Lernstoff aus den Bänden 2 bis 4) etwa die gleiche Zeit benötigen wie für die Aufgaben 1 bis 10 (mit Lernstoff aus den Bänden 2 bis 4).

Wurden bestimmte Programmierarbeiten des öfteren ausgeführt, so sind sie in nachfolgenden Aufgaben manchmal vorgegeben. Das bringt den Vorteil, daß die dadurch erübrigte Zeit für das wirklich Neue und damit Schwierigere einer Aufgabe weit lohnender zur Verfügung steht.

In der Einführung zu dieser Programmierten Unterweisung wurde bereits erwähnt, daß die gezeigten Programme im Hinblick auf die Ein- und Ausgabe der Daten nicht vollständig sind. Die Eintragungen, die den Programmen noch hinzuzufügen sind, werden Sie in einem Folgelehrgang kennenlernen.

Befolgen Sie bitte beim Durcharbeiten dieses Bandes alle Verweise zu den anderen Bänden. Zu einem vollen Erfolg kommen Sie nur, wenn Sie den vorgeschriebenen Weg genauestens einhalten.

Bevor wir uns die erste Aufgabe vornehmen, müssen wir einige grundlegende Dinge zum Programmieren kennenlernen bzw. unser bisheriges Wissen darüber erweitern.

Im einzelnen handelt es sich um folgende Themen:

1. Der Weg von einem bestimmten Problem bis zur Verarbeitung der Daten, insbesondere das symbolische Programm und seine Umwandlung in das Maschinenprogramm.

2. Felder und Bereiche im Hauptspeicher.

3. Das Programmschema - ein notwendiges Hilfsmittel zum Schreiben eines Programms.

Arbeiten Sie bitte hierzu die nachstehend aufgeführten Kapitel in der angegebenen Reihenfolge durch. - Fahren Sie erst danach in diesem Band fort.

Text 3, Seite 5 (Kapitel I)

Text 2, Seite 5 (Kapitel I)

Text 3, Seite 47 (Kapitel II)

Sie haben im Kapitel I von Text 3 (Assembleranweisungen) gelernt, wie man vom Programmablaufplan über das symbolische Programm zum Maschinenprogramm kommt.

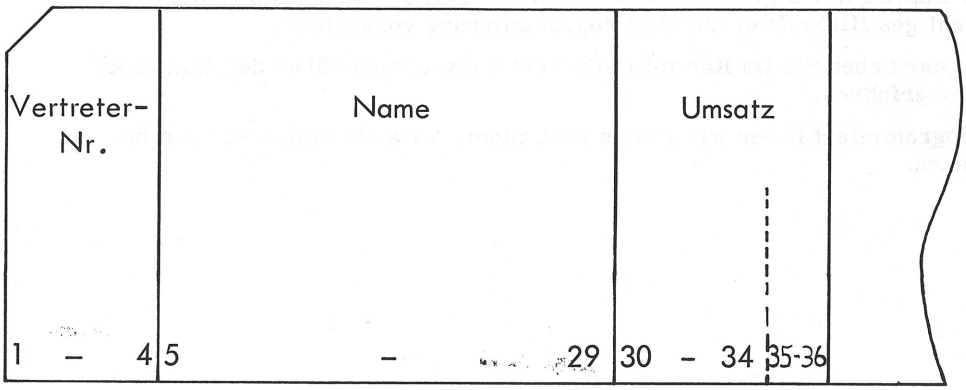
Im Kapitel II des gleichen Bandes wurde Ihnen das Programmschema als wichtiges Hilfsmittel für die Programmierung vorgestellt.

Ferner haben Sie im Kapitel I von Text 2 etwas mehr über den Hauptspeicher erfahren.

Programmiert haben wir jedoch noch nicht. - Damit wollen wir nun beginnen.

Aufgabe 1

Eingabe: Umsatzkarten



Ausgabe: Umsatzliste

Vertreter-Nr.	Name	Umsatz
0007	LISTENREICH DAGOBERT	24786,15
0018	GENUEGSAM HEINRICH	09873,82
0023	VOLLDAMPF VIKTOR	21007,73
0084	FUCHS FRANZ-JOSEF	33590,29
0116	HINTERWEILER THEODOR	08911,64
0712	SCHWERLICH OTTOKAR	05613,37
0814	GESCHWIND BRUNO	11314,48
1215	GUETLICH ARMIN	07865,20

Druckstellen	1 - 4	7 - 31	34 - 40

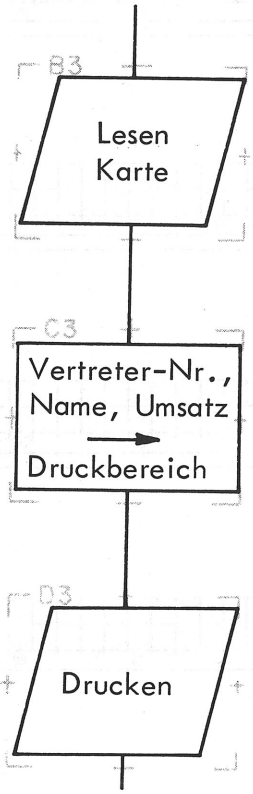
Abb. 1

Aufgabe 1

Sehen Sie bitte die Aufgabe 1 (Abb. 1) an. -

Aus Umsatzkarten ist eine Umsatzliste zu erstellen. - Hierbei handelt es sich um eine Arbeit im Einzelgang.

Der folgende Programmablaufplan (Ausschnitt) zeigt die Verarbeitung einer Karte.



Nur diese Schritte werden wir codieren. -

IBM System / 360
Speicherbelegung

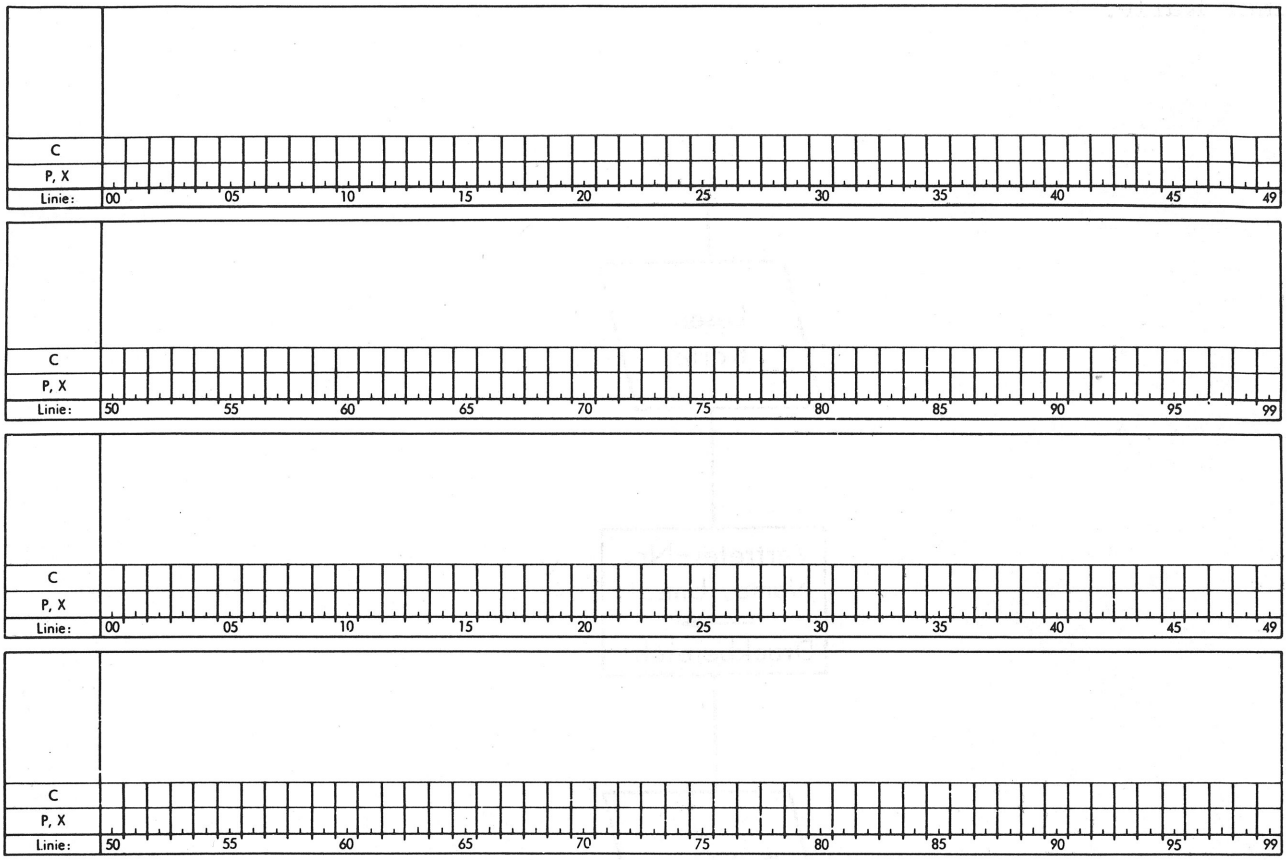


Abb. 2

Bevor wir jedoch mit der Codierung beginnen, müssen wir noch verschiedene Vorarbeiten erledigen.

Wie Sie wissen, müssen wir den Eingabe- und den Ausgabebereich selbst festlegen. Hierbei bedienen wir uns als Hilfsmittel einer Speicherbelegung.

Abb. 2 zeigt eine solche Speicherbelegung.

In der Speicherbelegung sind die Hauptspeicherstellen (Bytes) als Kästchen auf Linien aneinandergereiht. Insgesamt können auf einem (Speicherbelegungs-) Blatt 200 Hauptspeicherstellen (2 x 00-99 = 2 x 100) dargestellt werden.

Die Zahlen 00-49 der ersten und dritten Linie und die Zahlen 50-99 der zweiten und vierten Linie ergeben zusammen mit der Liniennummer die Hauptspeicheradressen. Beispiel: Wenn wir die Hauptspeicherstellen 700-799 darstellen wollen, so müssen wir in der ersten und zweiten Linie die Liniennummer 7 (Linie:7) eintragen. Selbstverständlich könnten hierfür auch die dritte und vierte Linie genommen werden.

Die Bedeutung der Eintragungen C, P und X für die Kästchen (Speicherstellen) erfahren Sie später.

IBM System / 360
Speicherbelegung

Aufgabe 1

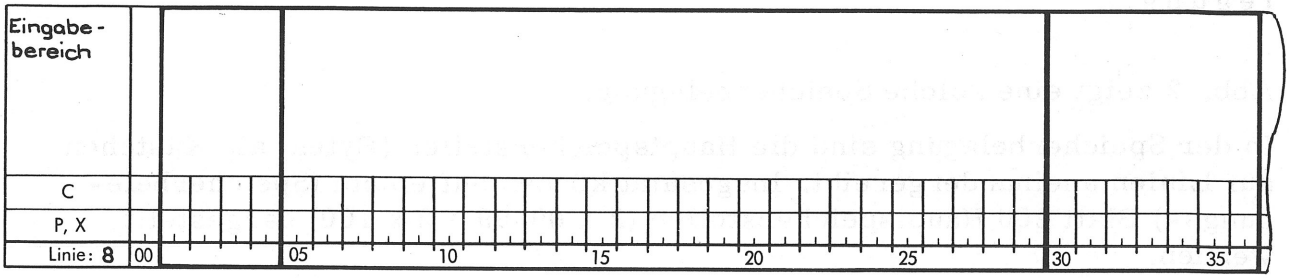


Abb. 3

Entnehmen Sie Ihren Arbeitsmitteln die Aufgabe 1. -

Damit alle für die Aufgabe benötigten Daten stellengerecht eingelesen werden können, muß der Eingabebereich ___(Z)___ Bytes (Speicherstellen) groß sein.

* * * * *

36

Sehen Sie sich bitte Abb. 3 an. -

Der Eingabebereich ist mit seinen einzelnen Feldern in eine Speicherbelegung (im Ausschnitt) eingetragen.

Der Eingabebereich umfaßt die Speicherstellen ___(Z)___ bis ___(Z)___.

* * * * *

801

836

Entnehmen Sie bitte Ihren Arbeitsmitteln eine Speicherbelegung. -

Zeichnen Sie den Eingabebereich entsprechend Abb. 3.

Nun wenden wir uns dem Ausgabebereich zu.

Dieser muß ___(Z)___ Bytes groß sein (s. Aufgabe 1).

* * * * *

40

Zeichnen Sie im Anschluß an den Eingabebereich den Ausgabebereich mit Feldeinteilung.

* * * * *

Vergleichen Sie Ihre Speicherbelegung mit der der Abb. 4 (auf der Rückseite dieses Blattes).

IBM System/360 Speicherbelegung

Aufgabe 1

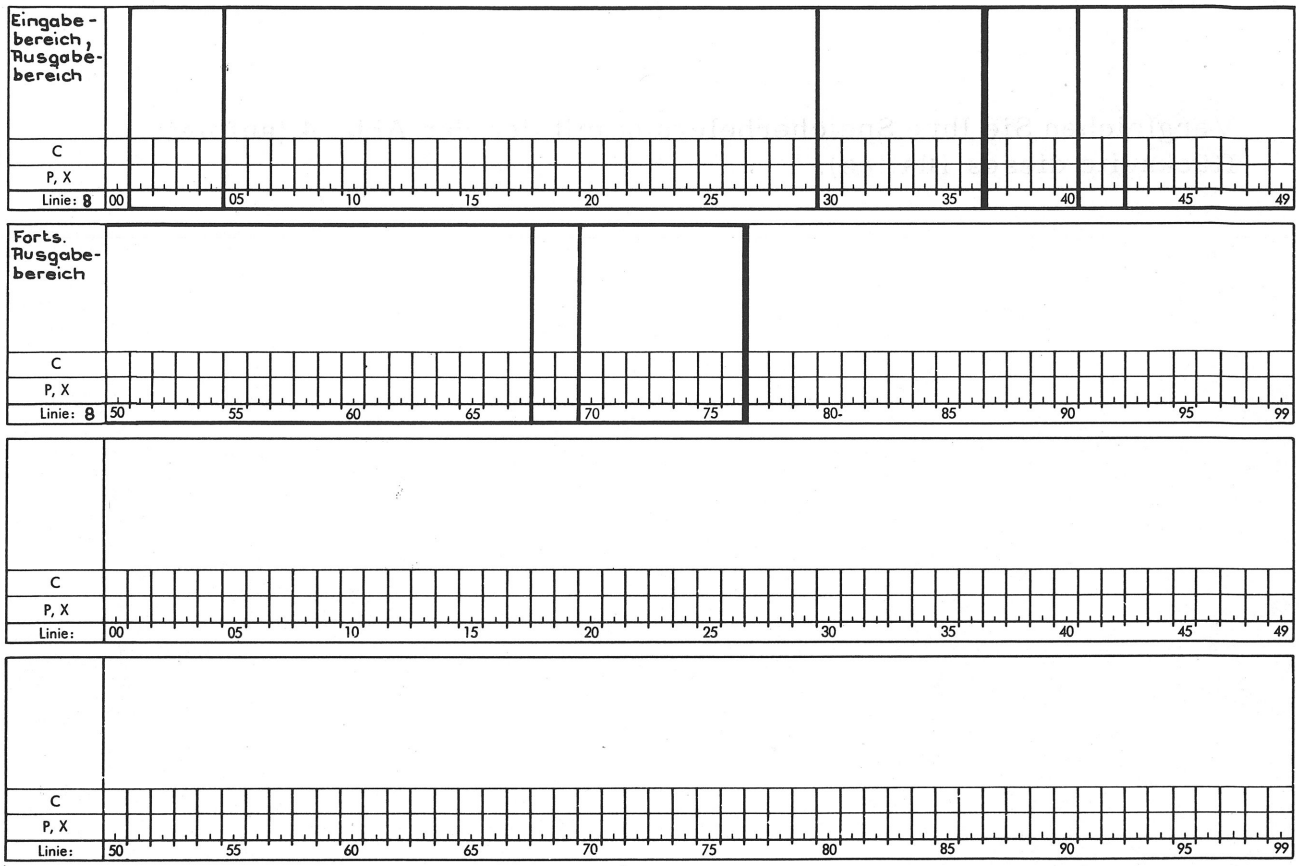


Abb. 4

Überprüfen Sie bitte Anfang, Ende und Einteilung des Ausgabebereiches.

Verbessern Sie Ihre Speicherbelegung soweit erforderlich.

Nun müssen wir den Bereichen und Feldern symbolische Namen zuordnen.

Die Bereiche nennen wir	EBER (Eingabebereich)
und	ABER (Ausgabebereich)
Die einzelnen Felder bezeichnen wir mit:	NR
	NAME
	UMSATZ

Damit wir zwischen "NR" im Eingabe- und "NR" im Ausgabebereich unterscheiden können, stellen wir den Namen im Eingabebereich ein E und den Namen im Ausgabebereich ein A voran.

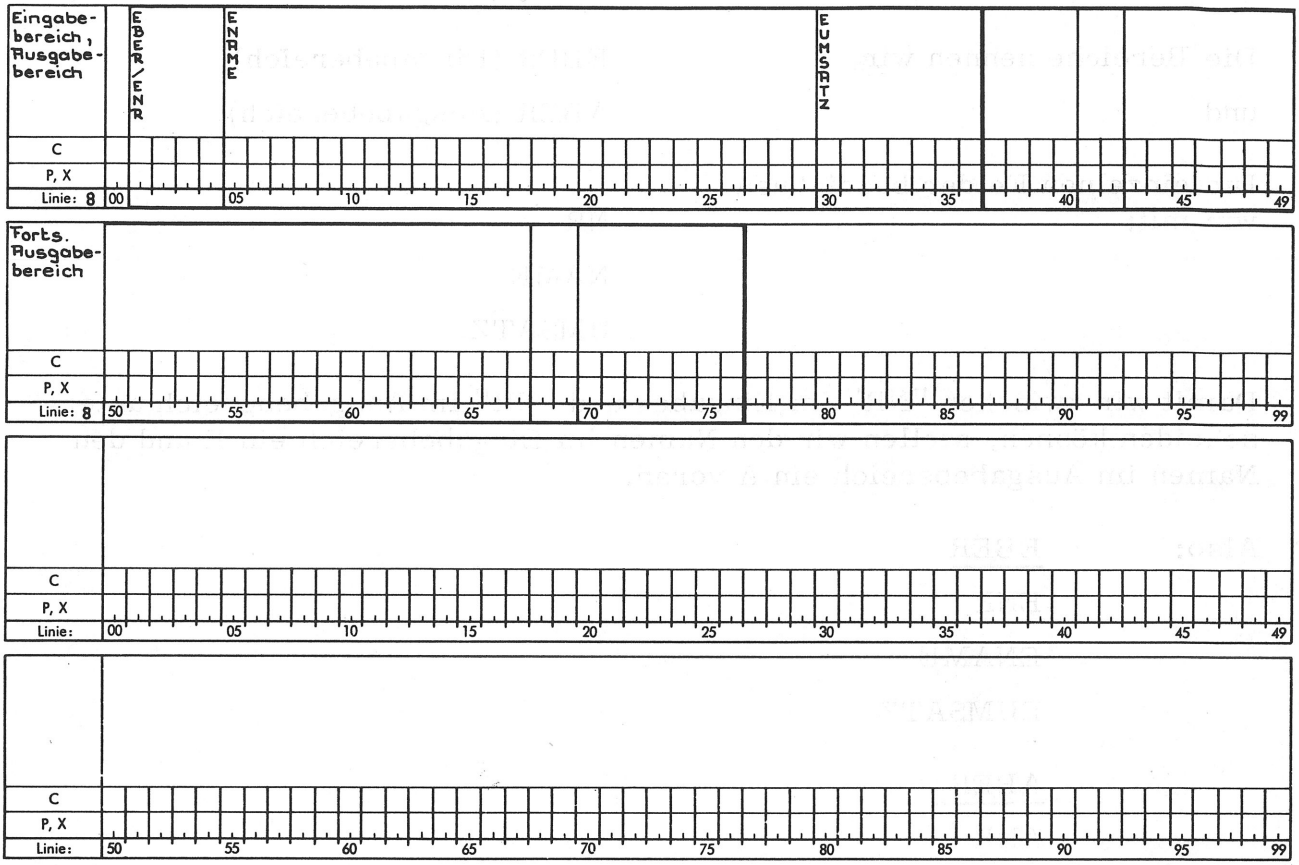
Also:

EBER
ENR
ENAME
EUMSATZ

ABER
ANR
ANAME
AUMSATZ

IBM System / 360 Speicherbelegung

Aufgabe 1


Eingabe-bereich, Ausgabe-bereich
EINGABE/AUSGABE
EINGABE
EINGABE
EIN

In Abb. 5 sind die Namen für den Eingabebereich eingetragen. -

Die Namen bezeichnen jeweils die linken Stellen des Bereiches bzw. der Felder.

Mit "ENAME" können wir somit in unserem Beispiel die Hauptspeicheradresse 805 symbolisch "ansprechen".

Womit können wir auf die Adresse 801 bezugnehmen?

* * * * *

Mit EBER und ENR

Anmerkung:

Die gleiche Adresse kann durch verschiedene symbolische Namen angesprochen werden.

Vervollständigen Sie nun Ihre Speicherbelegung mit allen Namen (Eingabebereich und Ausgabebereich).

* * * * *

Vergleichen Sie nun Ihre Speicherbelegung mit der der Abb. 6 (auf der Rückseite!).

Aufgabe 1

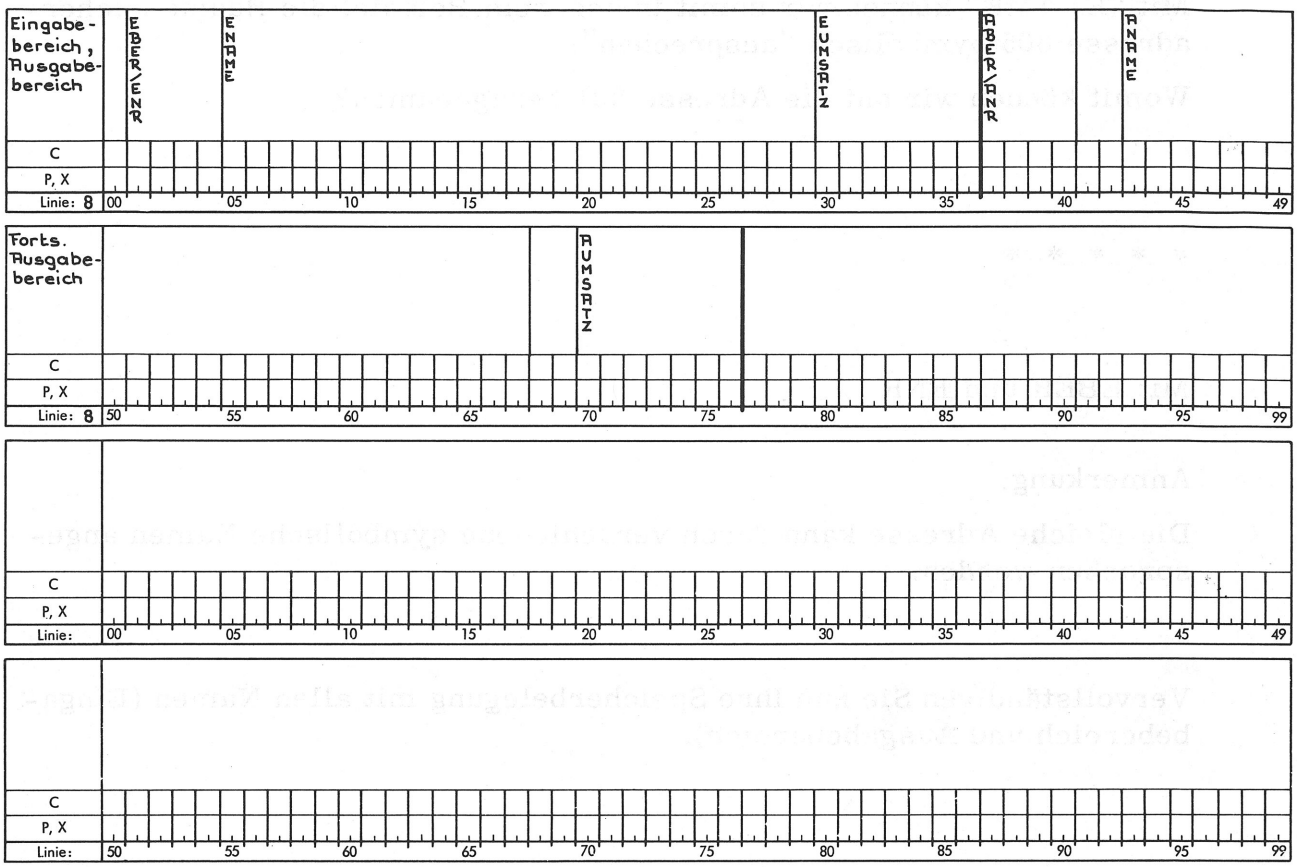


Abb. 6

Mit der Speicherbelegung haben wir uns ein wichtiges Hilfsmittel für die Codierung geschaffen.

Den Inhalt der Speicherbelegung müssen wir jedoch erst noch unserem DVS mitteilen. - Wir müssen ihm also sagen, wo der Eingabebereich beginnt, wie er heißt, wie lang er ist, welche Felder er enthält und wie lang diese sind. Das gleiche gilt auch für den Ausgabebereich.

Wir wissen das alles zwar schon, aber dem DVS müssen wir dies durch entsprechende Anweisungen erst noch zum Ausdruck bringen.

Wie man dies macht, erfahren Sie im Band 3.

→ Band 3, Seite 83 (START)

Seite 103 (DS)

Sie wissen nun, wie die START-Anweisung codiert wird und wie Eingabe- und Ausgabebereiche zu definieren sind.

Entnehmen Sie Ihren Arbeitsmitteln ein Programmschema.

Schreiben Sie nun die Definitionen zur Aufgabe 1. Benutzen Sie hierbei Ihre Speicherbelegung zur Aufgabe 1.

Verwenden Sie in der Kopfzeile "A01" für Problem (73 - 75).

* * * * *

Vergleichen Sie bitte Ihre Definitionen mit denen der Abb. 7 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 1	Lochhinweise		Problem A Ø 1 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite Ø 1 (76–77) von ____

Anweisung				Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)		(78–80)
*. ERSTELLEN UMSATZLISTE				Ø1Ø
	START	8Ø1		Ø2Ø
*. DEFINITIONEN				Ø3Ø
EBER	DS	ØCL36	EINGABE-BEREICH	Ø4Ø
ENR	DS	CL4		Ø5Ø
ENAME	DS	CL25		Ø6Ø
EUMSATZ	DS	CL7		Ø7Ø
ABER	DS	ØCL4Ø	AUSGABE-BEREICH	Ø8Ø
ANR	DS	CL4		Ø9Ø
	DS	CL2		1ØØ
ANAME	DS	CL25		11Ø
	DS	CL2		12Ø
AUMSATZ	DS	CL7		13Ø
				14Ø
				15Ø
				16Ø
				17Ø
				18Ø
				19Ø
				2ØØ

Abb. 7

Zu den Definitionen:

Ihre Eintragungen müssen mit denen der Abb. 7 übereinstimmen, soweit es sich nicht um die hinzugefügten Bemerkungen in den Zeilen 010, 030, 040 und 080 handelt.

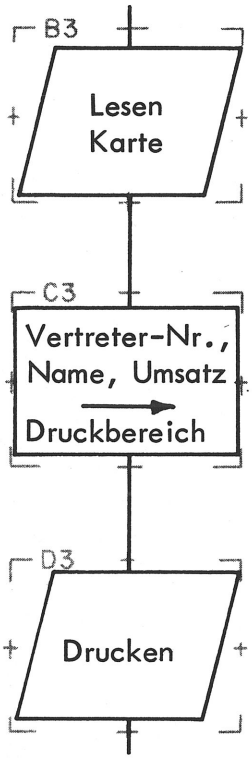
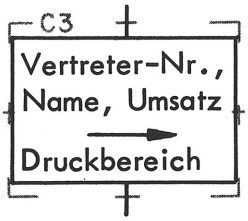

B3
Lesen
Karte
C3
Vertreter-Nr.,
Name, Umsatz
Druckbereich
D3
Drucken

Abb. 8

Nun werden wir die drei Schritte des Programmablaufplans codieren.

Aus Abb. 8 greifen wir uns den Verarbeitungsblock heraus.



Der Block bedeutet: "_____"

* * * * *

Übertrage Vertreter-Nr., Name und Umsatz nach Druckbereich.

Wir benötigen hierfür also die Operation "Übertragen Zeichen".
Diese Operation benötigen wir ____(Z)____ mal.

* * * * *

3 (je einmal für: Vertreter-Nr., Name, Umsatz.)

"Übertragen Zeichen" hat den symbolischen Operationsschlüssel MVC (engl.: MoVe Characters = Übertragen Zeichen).

In unserem Beispiel haben wir u.a. die Vertreter-Nr. aus dem Eingabebereich in den Ausgabebereich zu übertragen.

Die beiden Felder haben die symbolischen Namen _____ und _____.

* * * * *

ENR (und) ANR

"Übertrage Vertreter-Nr. nach Druckbereich" sieht codiert wie folgt aus:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35
	M.V.C.	A.N.R.,E.N.R.

Der Operationsschlüssel "MVC" gibt an, _____ zu tun ist (ganze Zeichen zu übertragen).

Der erste Operand (ANR) gibt an, _____ zu übertragen ist.

Der zweite Operand (ENR) gibt an, _____ die Daten kommen.

* * * * *

was
wohin
woher

Wieviele Zeichen übertragen werden, wird durch den ersten Operanden bestimmt.

Der erste Operand ist ANR. ANR ist definiert (vgl. Definitionen) als ein ____(Z)____ Speicherstellen (Bytes) großes Feld.

* * * * *

4

Nehmen Sie bitte Ihre Speicherbelegung zur Hand. -

Durch die Instruktion

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35
	M,V,C,	A,N,R,,E,N,R,

wird der Inhalt der Speicherstellen ____(Z)____ bis ____(Z)____
in die Speicherstellen ____(Z)____ bis ____(Z)____ übertragen.

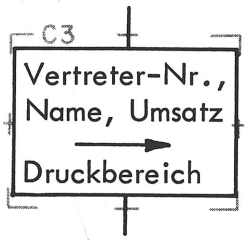
* * * * *

801 (bis) 804

837 (bis) 840

Entnehmen Sie Ihren Arbeitsmitteln ein Programmschema. -

Codieren Sie jetzt den Verarbeitungsblock



* * * * *

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	MVC	ANR,ENR
	MVC	ANAME,ENAME
	MVC	AUMSATZ,EUMSATZ

Die Reihenfolge dieser Instruktionen ist nicht zwingend.

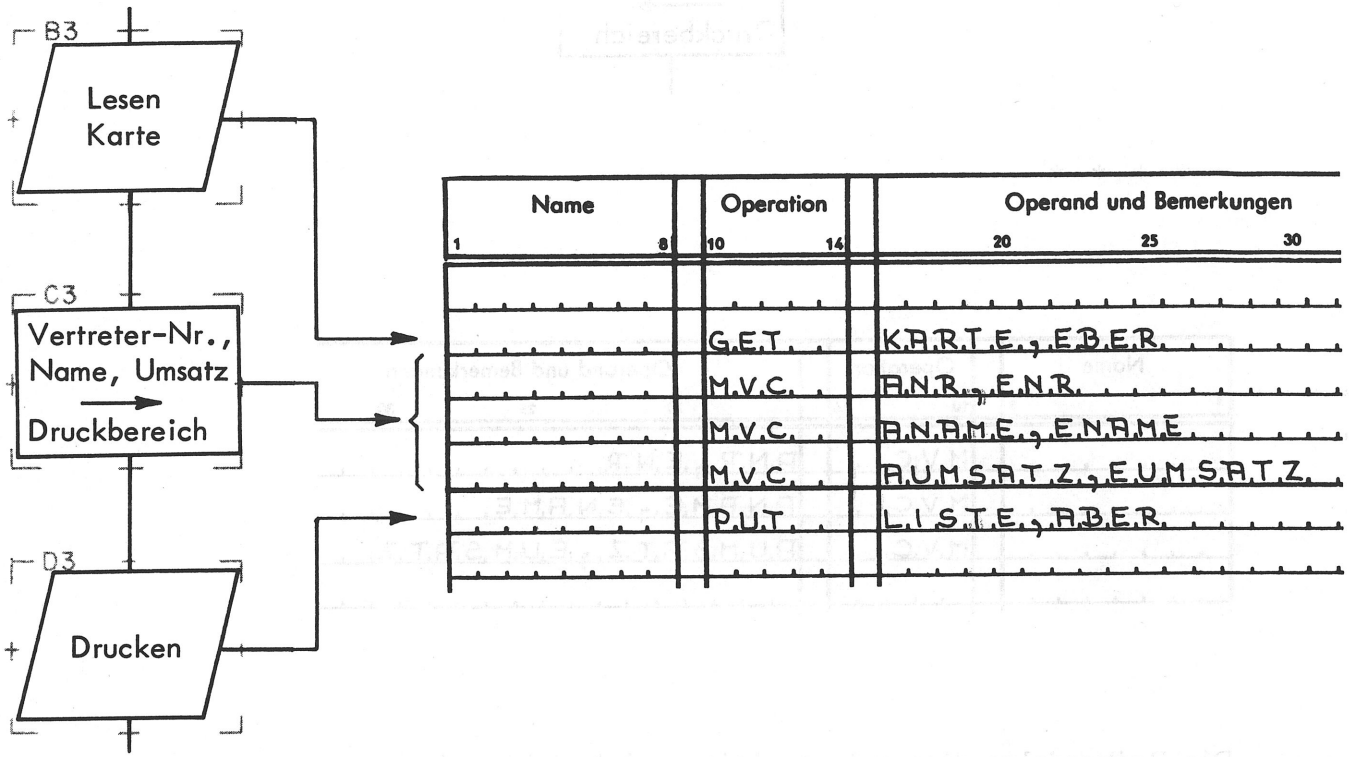

B3
Lesen
Karte
C3
Vertreter-Nr.,
Name, Umsatz
Druckbereich
D3
Drucken
Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
G.E.T.
K.A.R.T.E.,E.B.E.R.
M.V.C.
A.N.R.,E.N.R.
M.V.C.
A.N.A.M.E.,E.N.A.M.E.
M.V.C.
A.U.M.S.A.T.Z.,E.U.M.S.A.T.Z.
P.U.T.
L.I.S.T.E.,A.B.E.R.

Abb. 9

Jetzt müssen wir noch unsere beiden Ein-/Ausgabe-Blöcke codieren. -

Die Abb. 9 zeigt die hierfür erforderlichen Eintragungen.

In diesem Lehrgang befassen wir uns mit der Verarbeitung in der Zentraleinheit. Alle die Ein-/Ausgabe betreffenden Operationen werden in den Folgelehrgängen behandelt.

Die Instruktionen für Lesen Karte, Drucken und Stanzen sind jedoch zur Abrundung der Beispiele in diesen Lehrgang aufgenommen. Die genaue Wirkungsweise der Operationen wird aber erst im nächsten Lehrgang besprochen.

Der Operationsschlüssel GET (engl.) bedeutet einlesen.

In unserem "Beispiel"-System haben wir als Eingabeeinheit nur den Kartenleser, d.h. die Eingabe von Daten erfolgt immer durch Lochkarten.

Der erste Operand (KARTE) der GET-Instruktion in Abb. 9 gibt an, von welchem Datenbestand einzulesen ist.

Die Gesamtheit der einzugebenden Lochkarten bildet einen Datenbestand. Dieser Datenbestand hat in unserem Beispiel den Namen KARTE.

Der zweite Operand (EBER) gibt den Bereich an, in den einzulesen ist.

PUT (engl.) wird verwendet, wenn Daten auszugeben sind (z.B. als Liste oder in Lochkarten).

Der erste und zweite Operand geben (wie bei GET) den Namen des Datenbestandes und den Bereich an.

Bei jedem GET wird in unserem Beispiel also eine Lochkarte in den Eingabebereich eingelesen, und bei jedem PUT wird eine Zeile aus dem Ausgabebereich gedruckt.

Wir fassen zusammen:

Bei GET und PUT wird als erster Operand immer der Name des Datenbestandes angegeben. Der zweite Operand bezeichnet den Bereich, in den die Daten eingelesen oder aus dem die Daten ausgegeben werden.

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 1	Lochhinweise	Problem A01 (73–75)
		Zeichen	
Programmierer	Datum	Lochung	Seite 01 (76–77) von 1

Anweisung				Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)		(78–80)
* ERSTELLEN UMSATZLISTE				010
	START	B01		020
* DEFINITIONEN				030
EBER	DS	0CL36	EINGABE-BEREICH	040
ENR	DS	CL4		050
ENAME	DS	CL25		060
EUMSATZ	DS	CL7		070
ABER	DS	0CL40	AUSGABE-BEREICH	080
ANR	DS	CL4		090
	DS	CL2		100
ANAME	DS	CL25		110
	DS	CL2		120
AUMSATZ	DS	CL7		130
* PROGRAMMABLAUF				140
	GET	KARTE,EBER		150
	MVC	ANR,ENR		160
	MVC	ANAME,ENAME		170
	MVC	AUMSATZ,EUMSATZ		180
	PUT	LISTE,ABER		190
				200

Abb. 10

Die Abb. 10 zeigt die gesamte Codierung unseres Beispiels. -

Die Überschrift (Bemerkung) "PROGRAMMABLAUF" ist zur besseren Übersicht eingefügt.

Bitte vervollständigen Sie jetzt Ihr Programmschema entsprechend der Abb. 10.

IBM System / 360 Speicherbelegung

Aufgabe 1

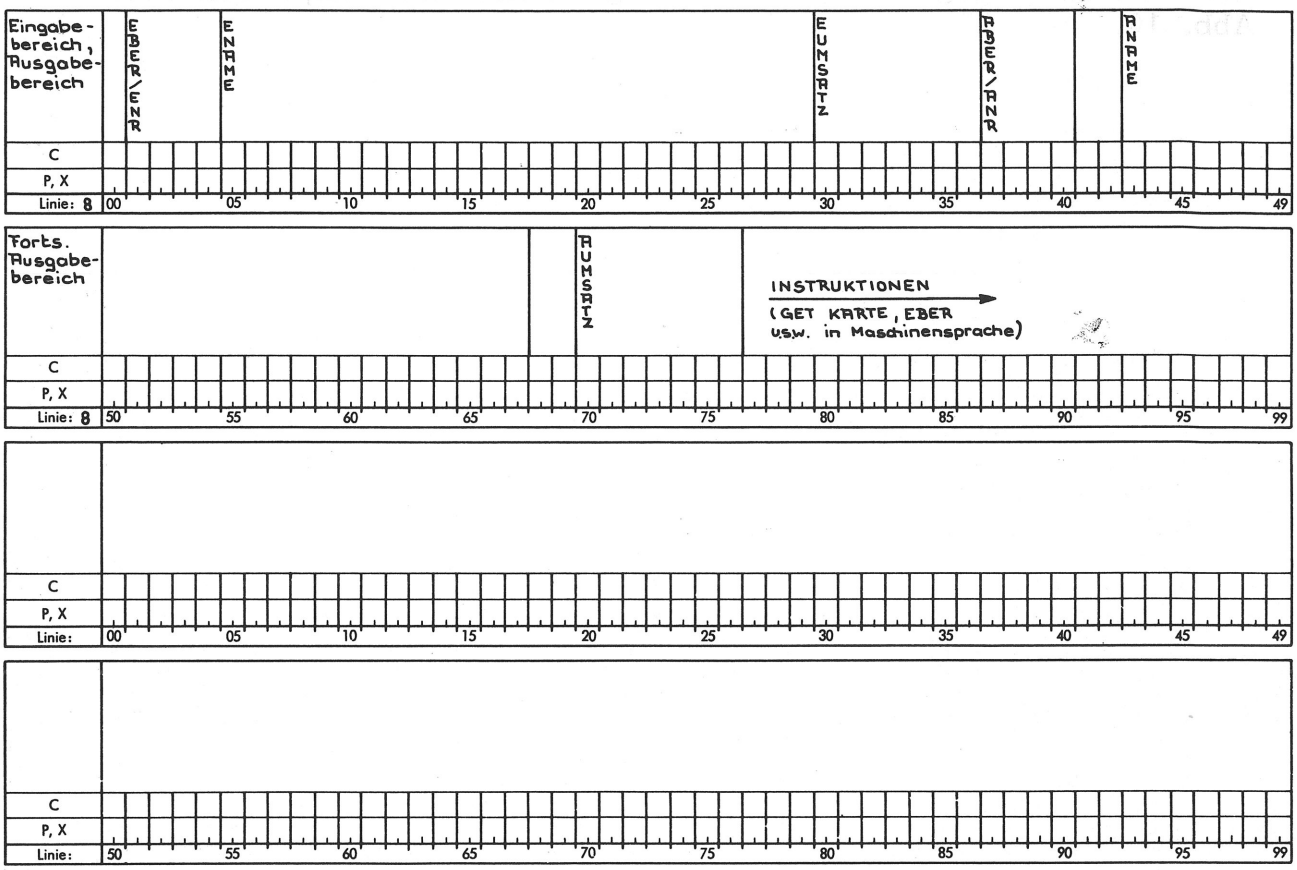


Abb. 11

Aufgrund der vorliegenden Codierung werden im Maschinenprogramm die Instruktionen unmittelbar im Anschluß an den Ausgabebereich gespeichert.

Ergänzen Sie bitte Ihre Speicherbelegung entsprechend Abb. 11.

Die Aufgabe 1 ist damit abgeschlossen. -

Legen Sie nun Aufgabenstellung, Speicherbelegung und Programmschema in einen Ordner ab.

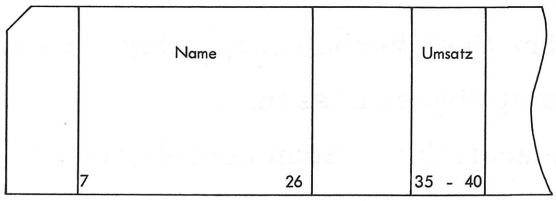
Bevor wir eine neue Aufgabe beginnen, schauen wir uns an, wie Zeichen im Hauptspeicher der /360 dargestellt werden. - Daran anschließend folgt eine zusammenfassende Beschreibung der Instruktion MVC mit Übungen.

Band 2, Seite 33 (Kapitel II)

Band 4, Seite 9 (MVC bis ⚠)

Aufgabe 2

Eingabe: Umsatzkarten



Ausgabe: Umsatzliste

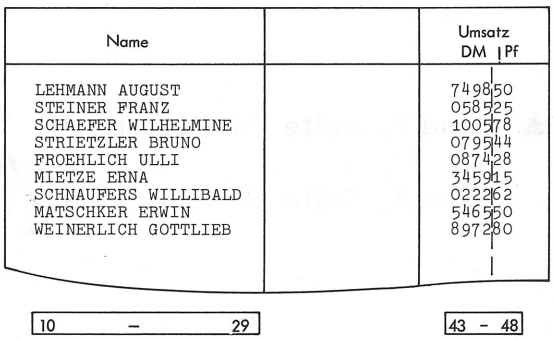


Abb. 1

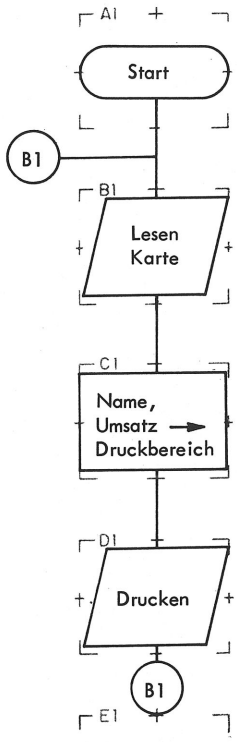


Abb. 2

Aufgabe 2

Bitte sehen Sie sich die Aufgabe 2 (Abb. 1) an. -

Die Aufgabenstellung ist die gleiche wie in Aufgabe 1.

In Aufgabe 1 sind wir zu einer Lösung gelangt, bei der lediglich eine Karte verarbeitet wird. Jetzt lösen wir die Aufgabe so, daß sämtliche Karten verarbeitet werden. Dazu programmieren wir eine Schleife.

Die Abb. 2 zeigt den entsprechenden Programmablaufplan. - Für jede Karte unseres Kartenpaketes laufen also folgende Schritte der Reihe nach ab:
B1, C1, D1.

Entnehmen Sie Ihren Arbeitsmitteln ein Entwurfsblatt für Programmablaufpläne.

Zeichnen Sie den Programmablaufplan zur Aufgabe 2 ab. Benutzen Sie dabei Ihre Zeichenschablone.

Entnehmen Sie den Arbeitsmitteln

a) die Aufgabe 2

b) eine Speicherbelegung.

Erstellen Sie bitte die Speicherbelegung unter Berücksichtigung folgender Angaben:

a) Der Eingabebereich soll mit der Speicherstelle 1001, der Ausgabebereich mit 1101 beginnen.

b) Verwenden Sie bitte die symbolischen Namen

EBER, ENAME, EUMSATZ,
ABER, ANAME, AUMSATZ.

* * * * *

Vergleichen Sie Ihre Speicherbelegung mit der Abb. 3 (Rückseite!).

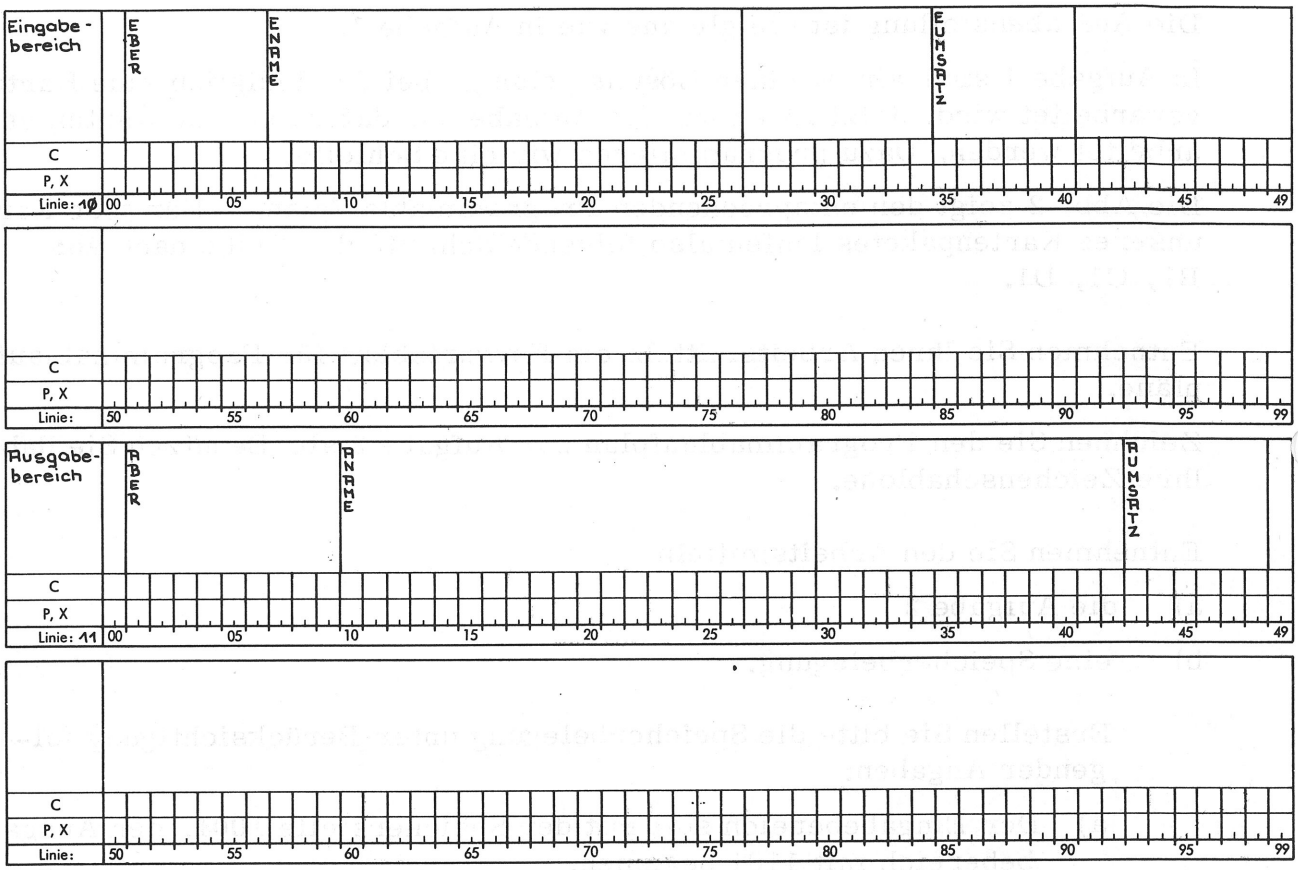

Eingabe-bereich
EBER
ENAME
EUMSATZ
C
P, X
Linie: 10
00 05 10 15 20 25 30 35 40 45 49
C
P, X
Linie:
50 55 60 65 70 75 80 85 90 95 99
Ausgabe-bereich
ABER
ANAME
AUMSATZ
C
P, X
Linie: 11
00 05 10 15 20 25 30 35 40 45 49
C
P, X
Linie:
50 55 60 65 70 75 80 85 90 95 99

Abb. 3

Verbessern Sie Ihre Speicherbelegung, wenn etwas falsch ist (mit Bleistift arbeiten!).

Nehmen Sie bitte ein Programmschema zur Hand. -

Schreiben Sie nun die Definitionen.
Reihenfolge der Eintragungen:

a) Problem A02

b) Überschrift "Erstellen Umsatzliste" (Bemerkung)

c) START-Anweisung

d) Überschrift "Definitionen" (Bemerkung)

e) DS-Anweisungen.

Hinweis: Beachten Sie bitte, daß der Ausgabebereich nicht direkt an den Eingabebereich anschließt.

* * * * *

Überprüfen Sie Ihre Codierung anhand der Abb. 4 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 2	Lochhinweise		Problem A Ø 2 (73 75)
		Zeichen		
Programmierer	Datum	Lochung		Seite Ø 1 1 (76 77) von ____

Anweisung				Karte Nr.
Name (1 8)	Operation (10 14)	Operand und Bemerkungen (20 25 30 35 40 45 50 55 60 65 70)		(78 80)
*.ERSTELL	LEN,UMSATZLISTE			Ø1Ø
	START	1ØØØ1		Ø2Ø
*.DEFINITIONEN				Ø3Ø
EBER	DS	ØCL4Ø		Ø4Ø
	DS	CL6		Ø5Ø
ENAME	DS	CL2Ø		Ø6Ø
	DS	CL8		Ø7Ø
EUMSATZ	DS	CL6		Ø8Ø
	DS	CL6Ø		Ø9Ø
ABER	DS	ØCL48		1ØØ
	DS	CL9		11Ø
ANAME	DS	CL2Ø		12Ø
	DS	CL13		13Ø
AUMSATZ	DS	CL6		14Ø
				15Ø
				16Ø
				17Ø
				18Ø
				19Ø
				2ØØ

Abb. 4

Zu den Definitionen:

Haben Sie darauf geachtet, daß zwischen dem EBER und dem ABER 60 Bytes frei bleiben müssen?

Falls Sie statt der DS-Anweisung in Zeile 090 noch einmal START mit Operand 1101 verwendet haben, so ist dies falsch. START darf nur einmal, am Anfang des Programms, gegeben werden.

Vergessen Sie bitte nicht, im Programmschema immer zwischen 0 (Ziffer Null) und O (Buchstabe O) durch Durchstreichen der Null(en) (Ø) zu unterscheiden.

Codieren Sie (auf Ihrem Programmschema) im Anschluß an die letzte Definitionszeile die drei Verarbeitungsschritte (Blöcke) des Programmablaufplans.

Beginnen Sie mit der Überschrift "Programmablauf".

Die Verzweigung zurück zu Schritt B1 lassen Sie bitte außer acht.

* * * * *

Vergleichen Sie jetzt Ihre Eintragungen mit denen der Abb. 5 ab Zeile 150 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 2	Lochhinweise		Problem A Ø 2 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite Ø 1 (76–77) von 1

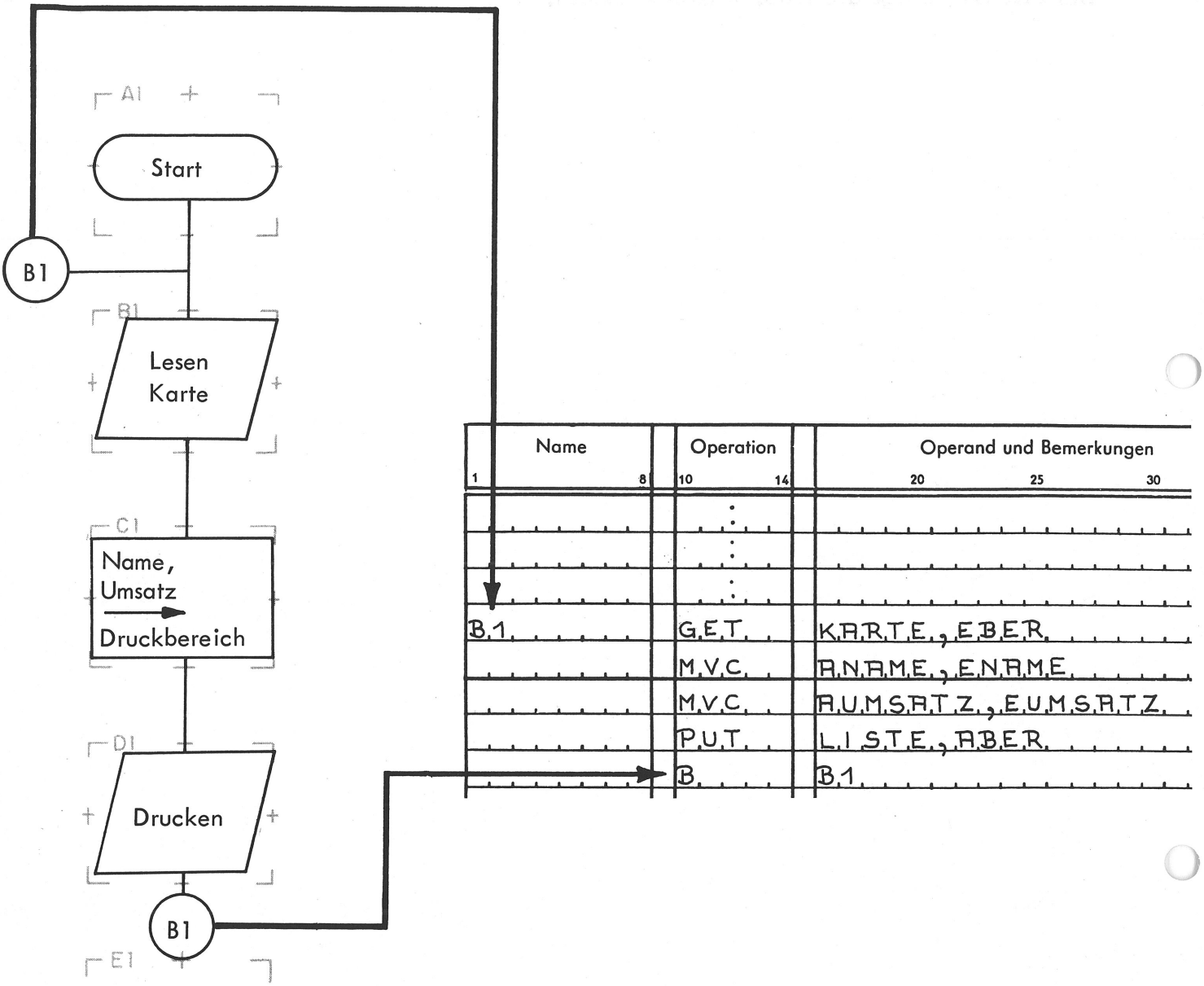
Anweisung			Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
*. ERSTELLEN UMSATZLISTE			Ø1Ø
	START	1ØØ1	Ø2Ø
*. DEFINITIONEN			Ø3Ø
EBER	DS	ØCL4Ø	Ø4Ø
	DS	CL6	Ø5Ø
ENAME	DS	CL2Ø	Ø6Ø
	DS	CL8	Ø7Ø
EUMSATZ	DS	CL6	Ø8Ø
	DS	CL6Ø	Ø9Ø
ABER	DS	ØCL48	1ØØ
	DS	CL9	11Ø
ANAME	DS	CL2Ø	12Ø
	DS	CL13	13Ø
AUMSATZ	DS	CL6	14Ø
*. PROGRAMMABLAUF			15Ø
B1	GET	KARTE,EBER	16Ø
	MVC	ANAME,ENAME	17Ø
	MVC	AUMSATZ,EUMSATZ	18Ø
	PUT	LISTE,ABER	19Ø
			2ØØ

Abb. 5

Nun müssen wir noch dafür sorgen, daß das Programm nach dem Drucken mit dem Einlesen der nächsten Karte fortfährt.

Die Codierung des Anschlußpunktes, also der Verzweigung (zurück) zu Schritt B1, zeigt die Abb. 6 (Rückseite!).





Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
B.1.	G.E.T.	K.A.R.T.E.,E.B.E.R.
	M.V.C.	A.N.A.M.E.,E.N.A.M.E.
	M.V.C.	A.U.M.S.A.T.Z.,E.U.M.S.A.T.Z.
	P.U.T.	L.I.S.T.E.,A.B.E.R.
	B.	B.1.

Abb. 6

B ist der symbolische Operationsschlüssel für eine u n b e d i n g t e Verzweigung (engl.: Branch = Verzweigen).

Der Operand der Instruktion ist der symbolische Name der Instruktion, zu der verzweigt wird (in unserem Beispiel ist dies die GET-Instruktion).

Wenn wir ein Feld im Hauptspeicher "ansprechen", so tun wir dies durch Angabe seines symbolischen Namens.

Nicht nur Feldern, sondern auch Instruktionen können wir einen Namen geben. - Auf den Namen der Instruktion können wir dann in anderen Instruktionen Bezug nehmen.

In unserem Beispiel (Abb. 6) hat die GET-Instruktion den Namen B1.
Auf diesen Namen nehmen wir in der _____-Instruktion Bezug.

* * * * *

B

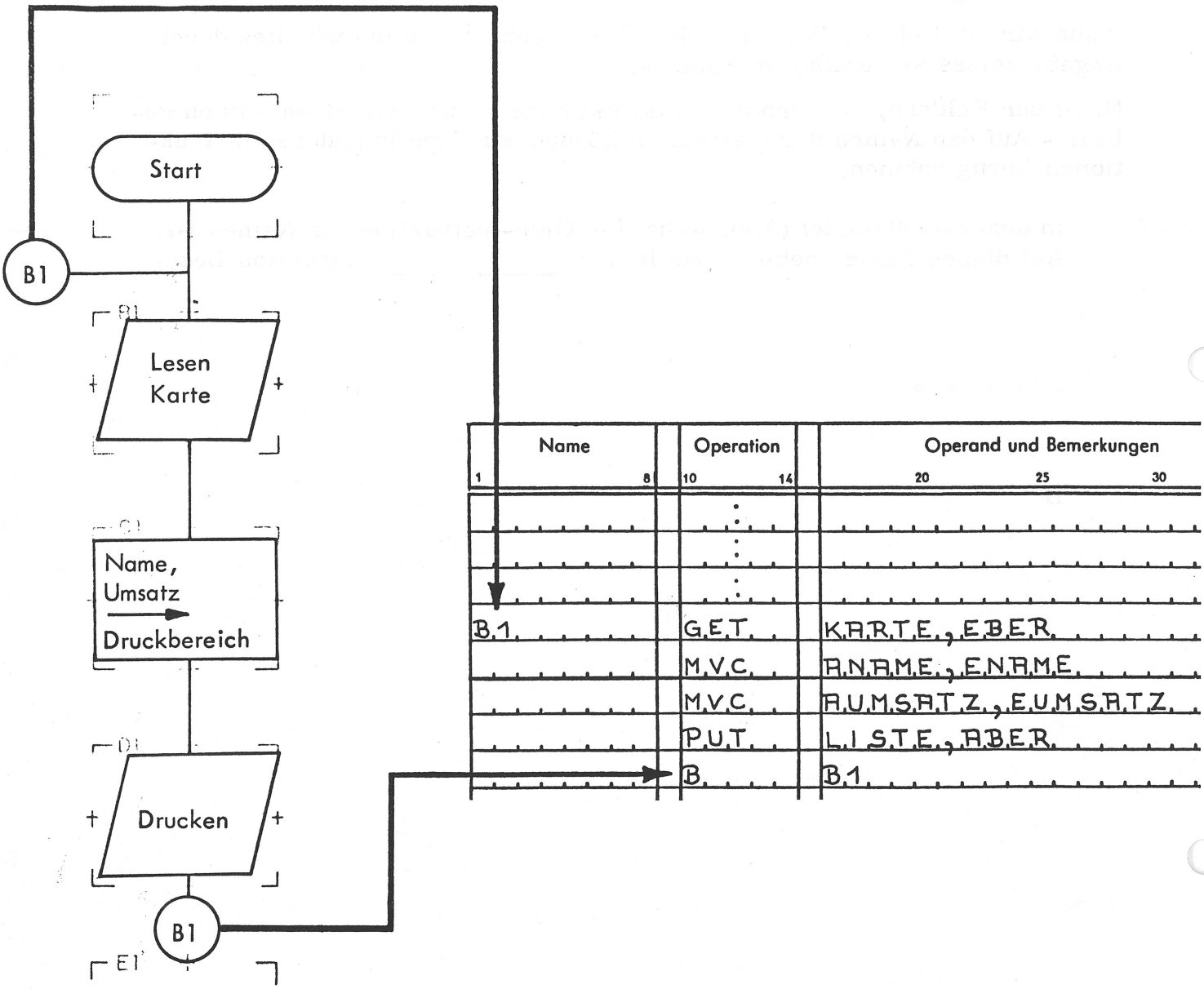

Start
B1
Lesen
Karte
Name,
Umsatz
Druckbereich
Drucken
B1
E1
Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
B.1
G.E.T.
K.A.R.T.E.,E.B.E.R.
M.V.C.
A.N.A.M.E.,E.N.A.M.E.
M.V.C.
A.U.M.S.A.T.Z.,E.U.M.S.A.T.Z.
P.U.T.
L.I.S.T.E.,A.B.E.R.
B.
B.1

Abb. 6

Die 5 Instruktionen unseres Programms werden immer wieder (bei der Verarbeitung jeder einzelnen Karte) durchlaufen. - Gäbe es die Verzweigungsinstruktion B nicht, müßten diese 5 Instruktionen immer wieder geschrieben werden (bei 100 000 zu verarbeitenden Umsatzkarten also 100 000 mal). So etwas wäre jedoch völlig unsinnig.

Wir machen uns nochmals klar, was die 5 Instruktionen bewirken.

```
B.1.....|GET..|KARTE,EBER.....
```

Durch diese Instruktion werden die Daten einer Lochkarte in den (WW).

Die Daten der Lochkarte ersetzen den bisherigen Inhalt des Eingabebereichs.

* * * * *

Eingabebereich eingelesen

Anmerkung:

Nach dem Lesen wird die Karte abgelegt (in ein Ablagefach des Kartenlesers).

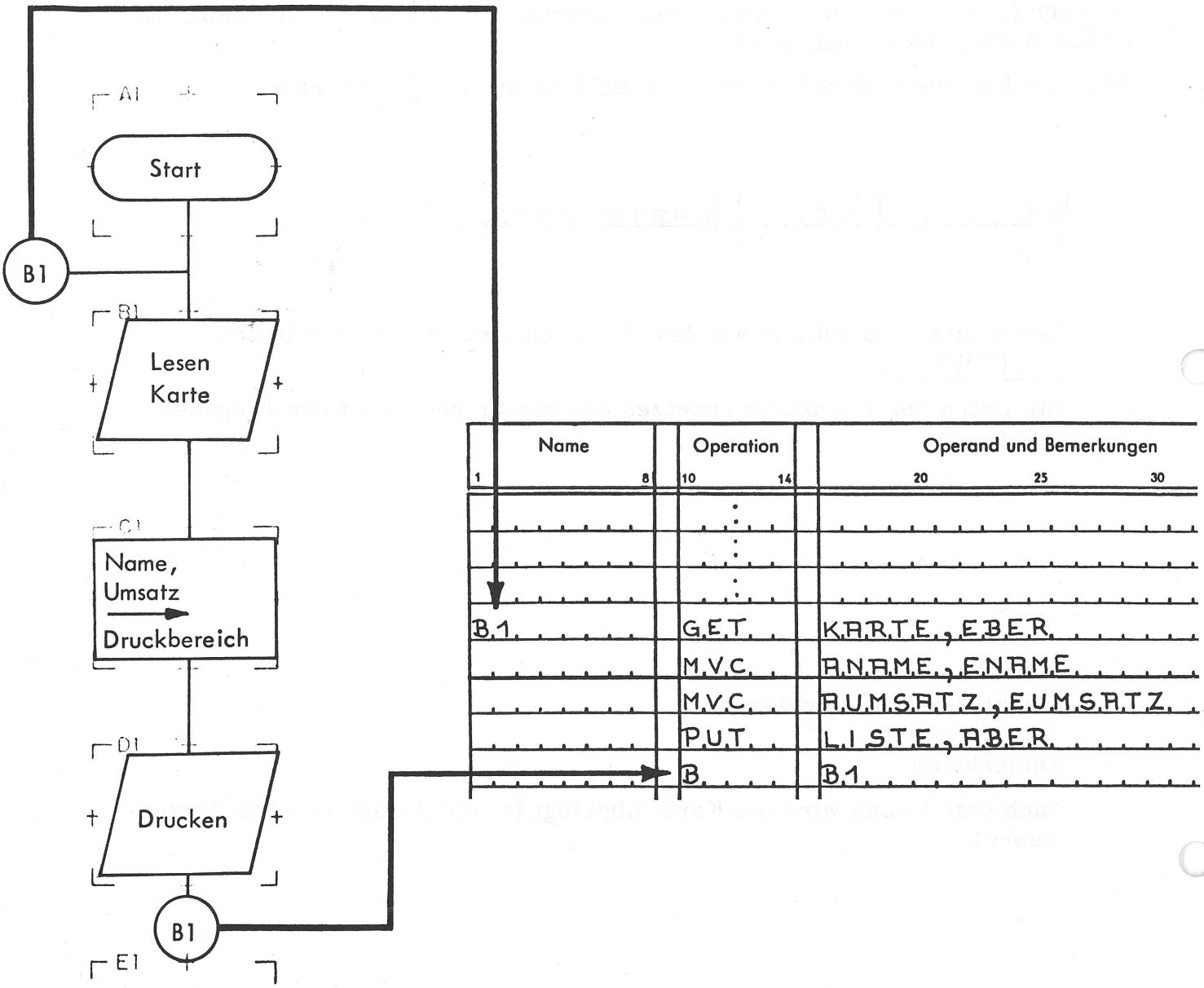
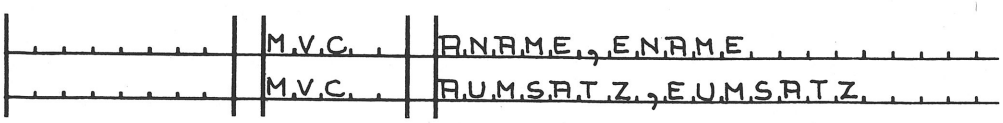

A1
Start
B1
B1
Lesen
Karte
C1
Name,
Umsatz
Druckbereich
D1
Drucken
B1
E1
Name
Operation
Operand und Bemerkungen
1
8
10
14
20
25
30
B.1
G.E.T.
K.A.R.T.E.,E.B.E.R.
M.V.C.
A.N.A.M.E.,E.N.A.M.E.
M.V.C.
A.U.M.S.A.T.Z.,E.U.M.S.A.T.Z.
P.U.T.
L.I.S.T.E.,A.B.E.R.
B.
B.1

Abb. 6



Die erste MVC-Instruktion überträgt den Inhalt des Feldes _____ in die Speicherstellen des Feldes _____.

Die übertragenen Daten ersetzen den Inhalt des Feldes _____.

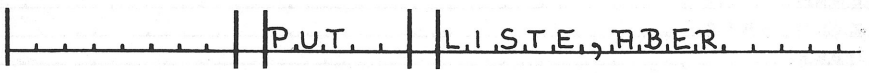
Die zweite MVC-Instruktion wirkt analog.

* * * * *

ENAME

ANAME

ANAME



Mit dieser Instruktion wird der Inhalt des _____ gedruckt.

* * * * *

Ausgabebereiches

Anmerkung:

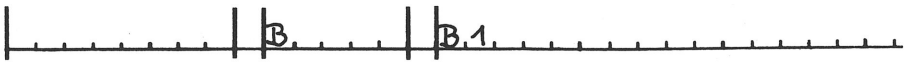
PUT schließt einen einzeiligen Zeilentransport ein. Dadurch wird Zeile für Zeile untereinander geschrieben.

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 2	Lochhinweise		Problem A02 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 011 (76–77) von 1

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.ERSTELL	EN.UMS	ARTZLIST E	010
	START	1000 1	020
*.DEFINIT	IONEN		030
EBER	DS	0CL40	040
	DS	CL6	050
ENAME	DS	CL20	060
	DS	CL8	070
EUMSATZ	DS	CL6	080
	DS	CL60	090
ABER	DS	0CL48	100
	DS	CL9	110
ANAME	DS	CL20	120
	DS	CL13	130
AUMSATZ	DS	CL6	140
*.PROGRAMM	MABLAUF		150
B1	GET	KARTE,EBER	160
	MVC	ANAME,ENAME	170
	MVC	AUMSATZ,EUMSATZ	180
	PUT	LIST E,ABER	190
	B	B1	200
	END		210

Abb. 7



Diese Instruktion bewirkt eine (unbedingte) Verzweigung zu der Instruktion mit dem Namen _____.

Diese Instruktion wird dann als nächste durchgeführt.

* * * * *

B1

Abb. 7 zeigt die gesamte Codierung der Aufgabe 2. -

Die erste Anweisung ist die _____-Anweisung.

Die letzte, für Sie noch neue Anweisung ist die _____-Anweisung.

* * * * *

START

END

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 2	Lochhinweise		Problem A Ø 2 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite Ø 1 (76–77) von 1

Anweisung			Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
*. ERSTELLEN UMSATZLISTE			Ø1Ø
	START	1ØØ1	Ø2Ø
*. DEFINITIONEN			Ø3Ø
EBER	DS	ØCL4Ø	Ø4Ø
	DS	CL6	Ø5Ø
ENAME	DS	CL2Ø	Ø6Ø
	DS	CL8	Ø7Ø
EUMSATZ	DS	CL6	Ø8Ø
	DS	CL6Ø	Ø9Ø
ABER	DS	ØCL48	1ØØ
	DS	CL9	11Ø
ANAME	DS	CL2Ø	12Ø
	DS	CL13	13Ø
AUMSATZ	DS	CL6	14Ø
*. PROGRAMMABLAUF			15Ø
B1	GET	KARTE,EBER	16Ø
	MVC	ANAME,ENAME	17Ø
	MVC	AUMSATZ,EUMSATZ	18Ø
	PUT	LISTE,ABER	19Ø
	B	B1	2ØØ
	END		21Ø

Abb. 7

Die END-Anweisung ist immer die letzte Eintragung im Programmschema. Sie besagt, daß das Programm zu Ende ist.

Durch die START- und END-Anweisungen wird dem DVS bei der Umwandlung des symbolischen Programms in das Maschinenprogramm angezeigt, daß alle Instruktionen, die nach START und vor _____ stehen, zum gleichen Programm gehören.

* * * * *

END

Vervollständigen Sie bitte nun Ihr Programmschema entsprechend Abb. 7. (Vergessen Sie bitte nicht die Numerierung - 210 - bei END).

Die Aufgabe 2 ist damit gelöst. -

Sie können auf der Speicherbelegung noch andeuten, daß die Instruktionen im Anschluß an den Ausgabebereich gespeichert werden (vgl. Aufgabe 1).

Legen Sie jetzt Aufgabe, Programmablaufplan, Speicherbelegung und Programmschema ab. -

Allgemeine Hinweise zur Codierung

Beim Codieren müssen wir darauf achten, daß die Operanden auch so geschrieben werden, wie sie definiert wurden.

Wir dürfen also im Operandenfeld keine Namen verwenden, die gar nicht oder anders _____ sind.

Wir merken uns:

Alle symbolischen Namen im Operandenfeld müssen die gleiche Schreibweise wie im Namenfeld (Stellen 1-8) haben.

* * * * *

definiert

Name 1 8	Operation 10 14	Operand und Bemerkungen 20 25 30
EUMSATZ	DS	CL6
	:	
AUMSATZ	DS	CL6
	:	
	:	
	MVC	AUMSATZ,EUMS

Die Umwandlung der obigen MVC-Instruktion würde zu (einem, keinem) _____ Fehler führen, da das Umwandlungsprogramm (weiß, nicht weiß) _____, daß mit EUMS das definierte Feld EUMSATZ gemeint ist.

* * * * *

einem
nicht weiß

	Name (1–8)	Operation (10–14)	Operand und (20–2)
	* ERSTELLEN UMSATZLISTE		
		START	1ØØ1
	* DEFINITIONEN		
①	EBER	DS	ØCL4Ø
		DS	CL6
③	ENAME	DS	CL2Ø
		DS	CL8
⑤	EUMSATZ	DS	CL6
		DS	CL6Ø
⑥	ABER	DS	ØCL48
		DS	CL9
②	ANAME	DS	CL2Ø
		DS	CL13
④	AUMSATZ	DS	CL6

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
⑦	B1	GET	KARTE, EBER ①
		MVC	ANAME ②, ENAME ③
		MVC	AUMSATZ ④, EUMSATZ ⑤
		PUT	LISTE, ABER ⑥
		B	B1 ⑦
		END	

Abb. 8

Sehen Sie sich bitte Abb. 8 an. -

Die zusammenwirkenden Symbole sind durch gleiche Ziffern (1-7) gekennzeichnet.

Welche symbolischen Namen stehen im Widerspruch zu den vorstehenden Ausführungen?

* * * * *

KARTE und LISTE

Anmerkung:

In einem vollständigen Programm mit kompletter Eingabe und Ausgabe sind auch die Symbole KARTE und LISTE definiert. Wie dies gemacht wird, erfahren Sie im nächsten Lehrgang.

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 2	Lochhinweise	Problem A02 (73–75)
		Zeichen	
Programmierer	Datum	Lochung	Seite 01 (76–77) von 1

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.ERSTELL	ENDUMSATZLISTE		010
	START	1000	020
*.DEFINITIONEN			030
EBER	DS	0CL40	040
	DS	CL6	050
ENAME	DS	CL20	060
	DS	CL8	070
EUMSATZ	DS	CL6	080
	DS	CL60	090
ABER	DS	0CL48	100
	DS	CL9	110
ANAME	DS	CL20	120
	DS	CL13	130
AUMSATZ	DS	CL6	140
*.PROGRAMM	MABLAUF		150
B1	GET	KARTE,EBER	160
	MVC	ANAME,ENAME	170
	MVC	AUMSATZ,EUMSATZ	180
	PUT	LISTE,ABER	190
	B	B1	200
	END		210

Abb. 9

Maschinen- und Assembler-Instruktionsanweisungen

Es gibt Anweisungen für Assemblerinstruktionen und Maschineninstruktionen.

Betrachten Sie bitte Abb. 9. -

Die Assemblerinstruktionsanweisungen sind gekennzeichnet.

Die Anweisungen START, DS und END dienen zur Steuerung der Umwandlung.

Sie geben an:

a) Anfang und Ende des symbolischen Programms

b) Längen und Adressen der Felder und Bereiche.

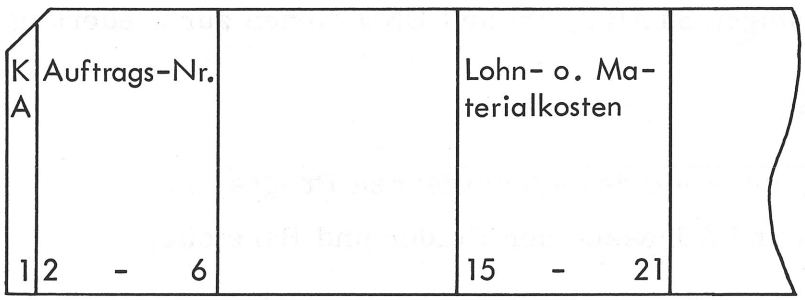
Im Maschinenprogramm sind diese Anweisungen nicht mehr vorhanden.

Bei der Umwandlung werden die Anweisungen für Maschineninstruktionen in Maschineninstruktionen übersetzt. Diese Instruktionen bilden das Maschinenprogramm.

Maschineninstruktionsanweisungen sind in unserem Beispiel der Abb. 9 die Instruktionen GET, MVC, PUT und B.

Aufgabe 3

Eingabe: Lohn- und Materialkarten



Kartenarten:

"1" in Spalte 1 — Lohnkarte

"2" in Spalte 1 — Materialkarte

Ausgabe: Liste der Lohn- und Materialkosten

Auftrags-Nr.	Lohnkosten	Materialkosten
00513	0094350	
00513		0205103
00513		0075881
00513	0123639	
01024	0109875	
01024		0038256
01024		0096127
01024	0088586	
01024		0134019
03152	0112570	
03152		0082663
03152		0009284
03152	0097423	
03604	0124714	
5 - 9	13 - 19	22 - 28

Abb. 1

Sehen Sie sich bitte die Aufgabe 3 (Abb. 1) an. -

Aus Lohn- und Materialkarten ist eine Liste der Lohn- und Materialkosten zu erstellen.

Die Lohn- und Materialkarten haben (die gleiche, nicht die gleiche) _____ Karteneinteilung.

* * * * *

die gleiche

Eine Unterscheidung zwischen Lohn- und Materialkarten ist aufgrund der _____ möglich.

* * * * *

Kartenart

Die Lohnkarten haben die Kartenart ____(Z)____, die Materialkarten die Kartenart ____(Z)____.

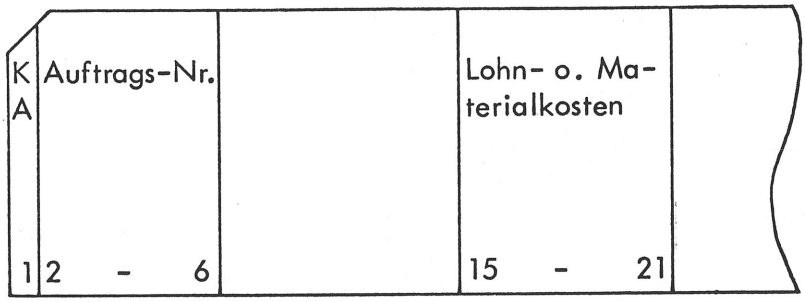
* * * * *

1

2

Aufgabe 3

Eingabe: Lohn- und Materialkarten



Kartenarten:

"1" in Spalte 1 — Lohnkarte

"2" in Spalte 1 — Materialkarte

Ausgabe: Liste der Lohn- und Materialkosten

Auftrags-Nr.	Lohnkosten	Materialkosten
00513	0094350	
00513		0205103
00513		0075881
00513	0123639	
01024	0109875	
01024		0038256
01024		0096127
01024	0088586	
01024		0134019
03152	0112570	
03152		0082663
03152		0009284
03152	0097423	
03604	0124714	
5 - 9	13 - 19	22 - 28

Abb. 1

In der Liste haben wir getrennte Spalten für Lohn- und Materialkosten.

Bei einer Karte mit der KA (Kartenart) "1" ist der Betrag aus den Spalten ____(Z)____ bis ____(Z)____ in den Druckstellen ____(Z)____ bis ____(Z)____ zu drucken.

Bei einer Karte mit der KA "2" ist der Betrag in den Druckstellen ____(Z)____ bis ____(Z)____ zu drucken.

* * * * *

15 (bis) 21

13 (bis) 19

22 (bis) 28

Es ist deshalb bei jeder Karte zu prüfen, ob es sich um eine Lohnkarte (KA 1) oder eine Materialkarte (KA 2) handelt.

Dies erfordert einen _____ mit einer _____.

* * * * *

Vergleich

Konstanten

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 3 Nr.: Seite:

Programmierer: Datum:

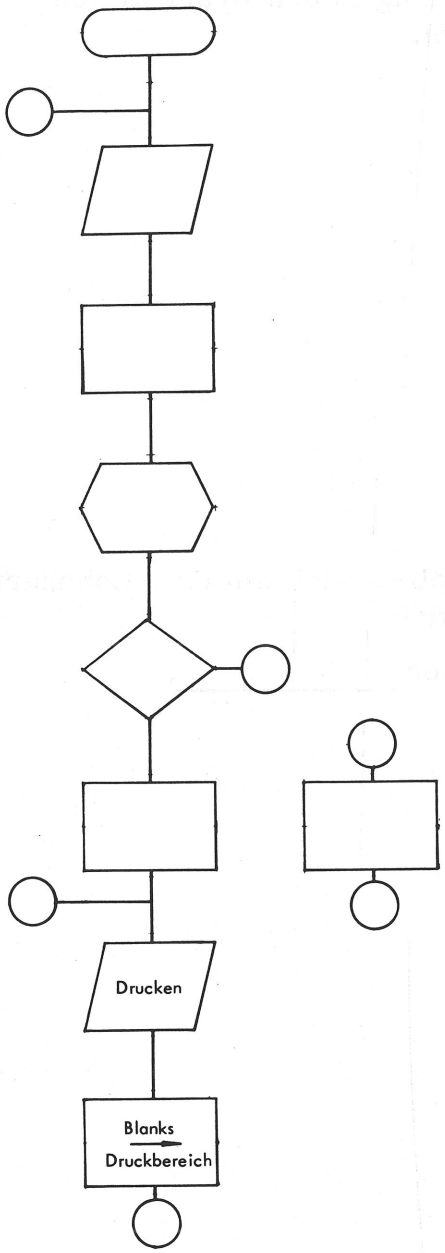


Abb. 2

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) die Aufgabe 3

b) ein Entwurfsblatt für Programmablaufpläne.

Versuchen Sie nun, den Programmablaufplan zu erstellen. Die Abb. 2 gibt Ihnen hierbei Hilfestellung.

* * * * *

Vergleichen Sie Ihren Programmablaufplan mit dem der Abb. 3 (Rückseite!).

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 3 Nr.: _____ Seite: _____

Programmierer: _____ Datum: _____

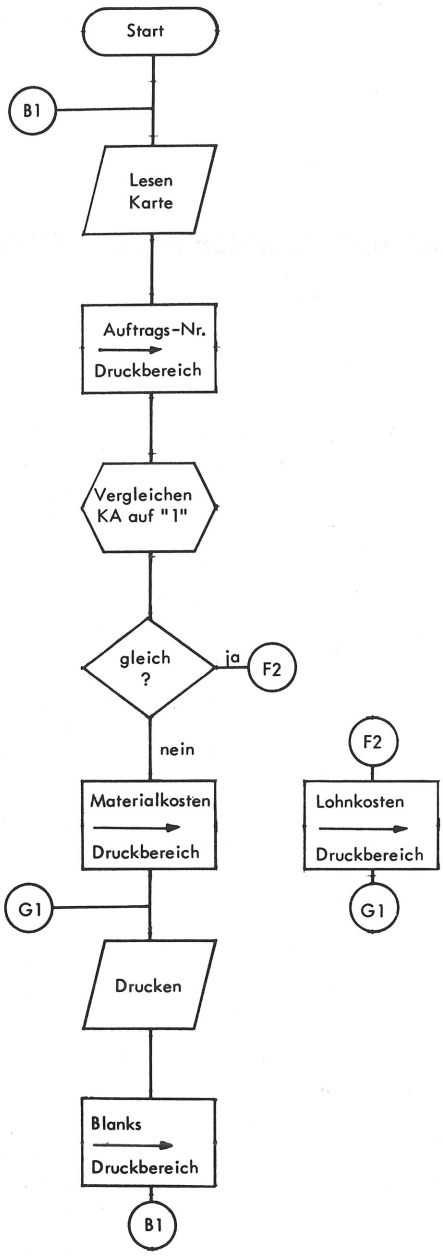


Abb. 3

Zum Programmablaufplan:

Nach dem Drucken werden Blanks in den Druckbereich übertragen. Versuchen Sie zu begründen, warum dies erforderlich ist.

* * * * *

Sinngemäß:

Damit keine Daten einer vorangehenden Karte mit angeschrieben werden. Es muß verhindert werden, daß gedruckt werden:

a) bei einer Lohnkarte die Materialkosten einer vorhergehenden Materialkarte,

b) bei einer Materialkarte die Lohnkosten einer vorhergehenden Lohnkarte.

Beachten Sie bitte: Es wird immer der ganze Inhalt des Ausgabebereiches gedruckt.

Würden nach dem Drucken keine Blanks in den Druckbereich übertragen, so ergäbe sich folgendes Listenbild (vgl. Aufgabe 3):

00513	0094350	
00513	0094350	0205103
00513	0094350	0075881
00513	0123639	0075881

Stimmt Ihr Programmablaufplan mit dem der Abb. 3 überein (1), Seite 77

Der Programmablaufplan in Abb. 3 zeigt eine mögliche Lösung.

Ihr Programmablaufplan ist selbstverständlich auch dann richtig, wenn Sie im Schritt D1 auf Kartenart 2 prüfen. (Die Eintragungen in den Blökken F1 und F2 müßten dann vertauscht sein.)

Das Übertragen der Auftrags-Nr. in den Druckbereich könnten wir auch unmittelbar vor dem Drucken vornehmen.

Ferner wäre möglich, nach dem Schritt F2 nochmals die gleichen Schritte wie G1 und H1 anzuschließen. Allerdings hätten wir dann zwei Schritte mehr zu programmieren.

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 3 Nr.: Seite:

Programmierer: Datum:

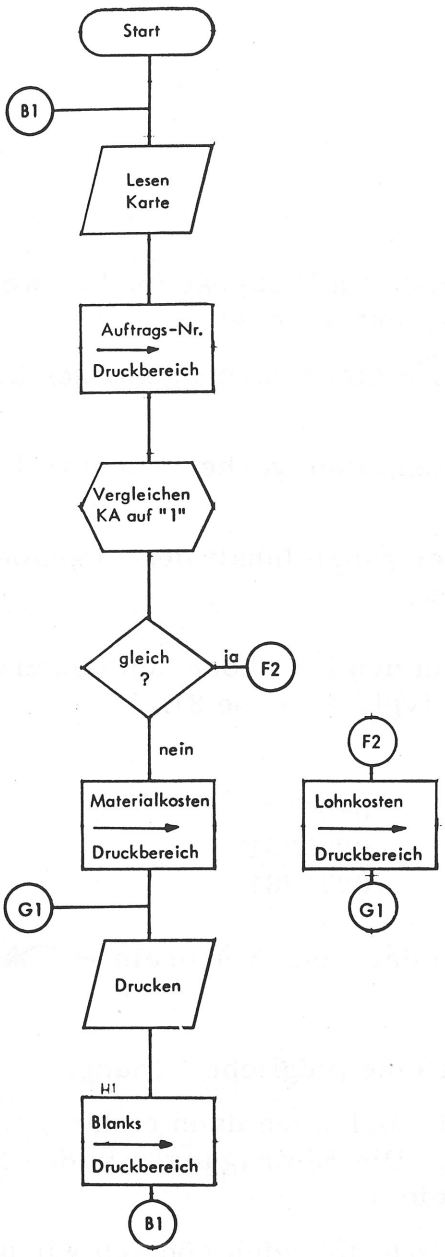


Abb. 3

(1)

Als Grundlage für die Codierung benützen wir den in Abb. 3 gezeigten Programmablaufplan, unabhängig davon, wie Ihr Entwurf aussieht.

Entnehmen Sie bitte den Arbeitsmitteln den Programmablaufplan zur Aufgabe 3.

Zeichnen Sie nun den Eingabe- und den Ausgabebereich in eine Speicherbelegung ein. Beachten Sie bitte folgende Angaben:

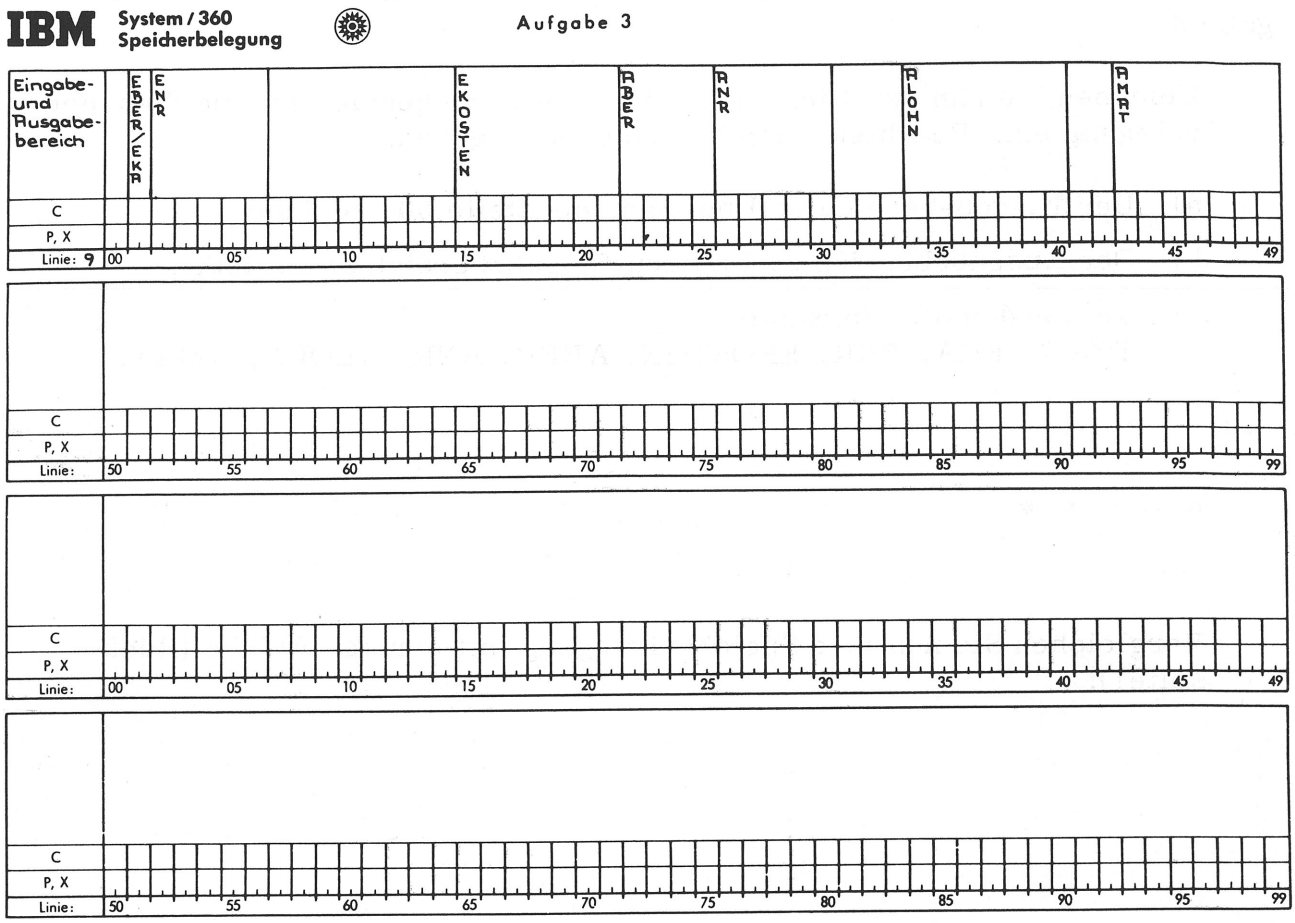
a) Der Eingabebereich soll bei Speicherstelle 901 beginnen.

b) Der Ausgabebereich soll an den Eingabebereich anschließen.

c) Verwenden Sie die Namen
EBER, EKA, ENR, EKOSTEN, ABER, ANR, ALOHN, AMAT.

* * * * *

Vergleichen Sie nun Ihre Speicherbelegung mit der der Abb. 4 (Rückseite!).



IBM System / 360 Speicherbelegung

Aufgabe 3

Eingabe- und Ausgabe-bereich		EBER/EKA	ENR		EKOSTEN	ABER	ANR		ALOHN		AMAT
C											
P, X											
Linie: 9	00	05	10	15	20	25	30	35	40	45	49

C											
P, X											
Linie:	50	55	60	65	70	75	80	85	90	95	99

C											
P, X											
Linie:	00	05	10	15	20	25	30	35	40	45	49

C											
P, X											
Linie:	50	55	60	65	70	75	80	85	90	95	99

Abb. 4

Verbessern Sie Ihre Speicherbelegung soweit erforderlich.

Wir wissen noch nicht, wie man eine Konstante definiert und wie ein Vergleich mit anschließender Abfrage programmiert wird.

Bevor wir mit der Codierung beginnen, lernen wir dies jetzt kennen.

Band 3, Seite 187 (DC-C)

Band 4, Seite 79 (CLC)

Seite 111 (Verzweigungsoperationen)

Nehmen Sie nun wieder die Aufgabe 3 und den dazugehörigen Programmablaufplan zur Hand.

Sie wissen nun, wie man Konstanten im Hauptspeicher definiert.

Erinnern Sie sich nun wieder an die Aufgabe.
Wir müssen die Kartenart feststellen, da Lohn- und Materialkosten in verschiedene Druckbereiche zu übertragen sind (Block D1 und E1). Außerdem muß durch Übertragen von Blanks in den Druckbereich verhindert werden, daß bei einer Lohnkarte die Materialkosten einer eventl. vorangegangenen Materialkarte oder bei einer Materialkarte die Lohnkosten einer eventl. vorangegangenen Lohnkarte mitgedruckt werden (Block H1).

Wir benötigen also für unsere Aufgabe 2 Konstanten:

a) eine für ___(WW)___ und

b) eine für ___(WW)___.

* * * * *

a) den Vergleich

b) die Übertragung der Blanks in den Druckbereich

Wir befassen uns nun mit dem Blanken im Druckbereich. Sehen Sie sich dabei Ihre Speicherbelegung an.

Nach dem Drucken einer Materialkarte müssen Blanks in den Ausgabebereich unbedingt nach _____ und nach dem Drucken einer Lohnkarte unbedingt nach _____ übertragen werden.

* * * * *

AMAT

ALOHN

Blanken wir nach dem Drucken stets nur eines der beiden Felder AMAT oder ALOHN, so müßten wir unseren Programmablaufplan abändern. Warum?

* * * * *

Sinngemäß:

Weil wir vor dem Übertragen der Blanks nochmals den Vergleich auf KA "1" mit anschließender Entscheidung einbauen müßten. (Sonst wüßten wir ja nicht, in w e l c h e s Feld die Blanks zu übertragen wären.)

Einfacher ist es, in beide Felder mit einer Instruktion nach dem Drucken Blanks zu übertragen. Zusammen mit den beiden Bytes zwischen den Feldern ALOHN und AMAT würde das eine 16 Bytes große Konstante erfordern.

Was stünde nach Ausführung der nachstehenden Instruktion im Feld AMAT?

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–25–30)
	MVC	ALOHN,KBLANKS
	:	
ALOHN	DS	CL7
	:	
KBLANKS	DC	CL16' '

* * * * *

Das, was v o r h e r auch schon enthalten war.
(Die Länge des ersten Operanden beendet die MVC-Instruktion!)

Geht man derart vor, so erfordert das Blanken der beiden Felder ALOHN und AMAT 2 Instruktionen. Eine Instruktion genügt, wenn man den gesamten Ausgabebereich (ABER) blankt. Wir bilden uns deshalb eine _____(Z)_____ Bytes große Konstante.

* * * * *

28

IBM System / 360 Speicherbelegung

Aufgabe 3

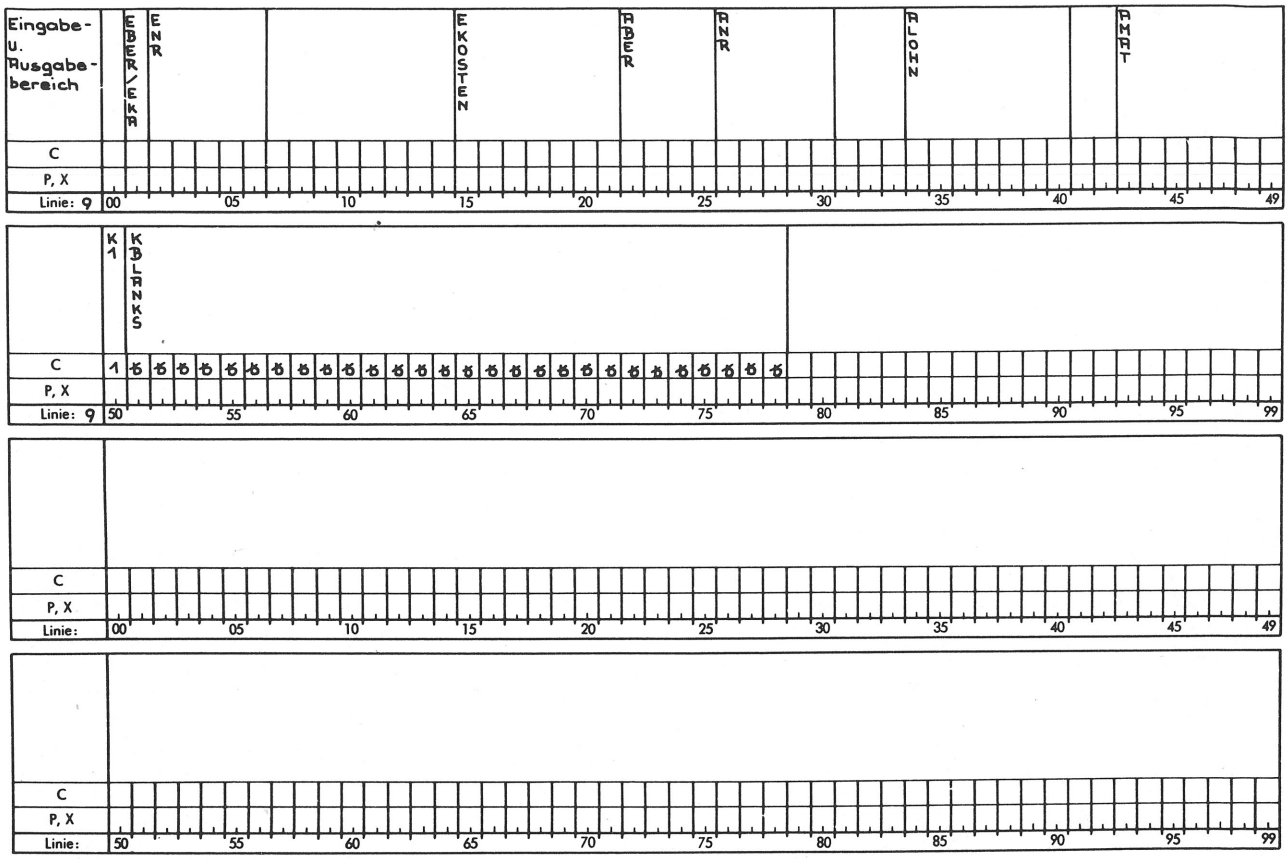


Abb. 5

Um die Kartenart festzustellen, braucht man zum Vergleich auf KA "1" die Konstante "1". Diese ist ____(Z)____ Byte(s) groß.
Wir nennen Sie K1.

* * * * *

1

Zeichnen Sie nun die beiden Konstanten in Ihre Speicherbelegung so ein, wie in Abb. 5 gezeigt.

Da es sich um Zeichenkonstanten handelt, wird ihr Inhalt in die mit C (engl.: Character = Zeichen) bezeichneten Kästchen eingetragen.

Schreiben Sie nun die Definitionen.

Beachten Sie bitte:
Benutzen Sie hierbei ein Programmschema. Verwenden Sie als Überschriften "Liste der Lohn- und Materialkosten" und "Definitionen".
Tragen Sie in Problem "A03" ein.

* * * * *

Vergleichen Sie bitte Ihre Definitionen mit denen der Abb. 6 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm Aufgabe 3		Lochhinweise		Problem A03 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 01 (76 77) von 2

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.L.I.S.T.E.	D.E.R. .L.O.H	N.-.U.N.D..M.A.T.E.R.I.A.L.K.O.S.T.E.N.	010
	S.T.A.R.T	9.0.1.	020
*.D.E.F.I.N.I	T.I.O.N.E.N		030
E.B.E.R.	D.S.	0.C.L.2.1.	040
E.K.A.	D.S.	C.L.1.	050
E.N.R.	D.S.	C.L.5.	060
	D.S.	C.L.8.	070
E.K.O.S.T.E.N.	D.S.	C.L.7.	080
A.B.E.R.	D.S.	0.C.L.2.8.	090
	D.S.	C.L.4.	100
A.N.R.	D.S.	C.L.5.	110
	D.S.	C.L.3.	120
A.L.O.H.N.	D.S.	C.L.7.	130
	D.S.	C.L.2.	140
A.M.A.T.	D.S.	C.L.7.	150
K.1.	D.C.	C.'1.'	160
K.B.L.A.N.K.S.	D.C.	C.L.2.8.'. '	170
			180
			190
			200

Abb. 6

Verbessern Sie Ihre Definitionen soweit erforderlich.

Versuchen Sie nun, den Programmablauf selbständig zu codieren. Nehmen Sie hierfür ein neues Programmschema.

Verwenden Sie als Überschrift "Programmablauf".

Codieren Sie zuerst die linke Folge, danach den Zweig F2.

* * * * *

Vergleichen Sie Ihre Codierung mit der der Abb. 7 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 3		Lochhinweise		Problem A03 (73–75)
			Zeichen		
Programmierer		Datum	Lochung		Seite 012 (76–77) von 2

Anweisung				Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)		(78–80)
*. PROGRAMMABLAUF.				010
B1.	GET.	KARTE,FEHLER.		020
	MVC.	ANR.,ENR.		030
	CLC.	EKA.,K1.		040
	BE.	F2.		050
	MVC.	AMAT.,EKOSTEN.		060
G1.	PUT.	LISTE,ABER.		070
	MVC.	ABER.,KBLANKS.		080
	B.	B1.		090
F2.	MVC.	ALOHN.,EKOSTEN.		100
	B.	G1.		110
	END.			120
				130
				140
				150
				160
				170
				180
				190
				200

Abb. 7

Stimmt Ihre Lösung mit der der Abb. 7 überein (Aufgabe 4). → (1), Seite 97

Zur Codierung des Programmablaufs:

Nehmen Sie bitte Ihren Programmablaufplan zur Hand. -

Der Schritt E1 stellt eine _____ Verzweigung dar.

Wenn der Vergleich auf KA 1 "gleich" ergab, wird nach Schritt _____ verzweigt.

* * * * *

bedingte

F2

Bei nicht "gleich" wird nicht nach F2 verzweigt, und das Programm wird mit dem _____ Schritt fortgesetzt.

Der darauffolgende Schritt ist Schritt _____.

* * * * *

darauffolgenden

F1

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 3		Lochhinweise		Problem A03 (73–75)
			Zeichen		
Programmierer		Datum	Lochung		Seite 012 (76–77) von 2

Anweisung				Karte Nr.
Name (1–8)		Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
*.PROGRAMM		MABLAUF		010
B1		GET	KARTE,EBER	020
		MVC	ANR,ENR	030
		CLC	EKA,K1	040
		BE	F2	050
		MVC	AMAT,EKOSTEN	060
G1		PUT	LISTE,ABER	070
		MVC	ABER,KBLANKS	080
		B	B1	090
F2		MVC	ALOHN,EKOSTEN	100
		B	G1	110
		END		120
				130
				140
				150
				160
				170
				180
				190
				200

Abb. 7

Sehen Sie sich die Abb. 7 an. -

Beim Codieren müssen im Anschluß an die BE-Instruktion die Instruktionen geschrieben werden, die als nächste ablaufen sollen, (wenn, wenn nicht) _____ verzweigt wird.

Zum Schluß codieren wir dann den Schritt F2 und die Verzweigung nach G1.

* * * * *

wenn nicht

A

Programm Aufgabe 3 | Zeichen

Programmierer | Datum | Lochung

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)	Anweisung	Karte Nr. (78–80)
*.PROGRAMM	MABLAUF			010
B1	GET	KARTE,EBER	1	020
	MVC	ANR,ENR	2	030
	CLC	EKA,K1	3	040
	BE	F2	4	050
	MVC	AMAT,EKOSTEN	5	060
G1	PUT	LISTE,ABER	6	070
	MVC	ABER,KBLANKS	7	080
	B	B1	8	090
F2	MVC	ALOHN,EKOSTEN	-	100
	B	G1	-	110
	END			120

B

Programm Aufgabe 3 | Zeichen

Programmierer | Datum | Lochung

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)	Anweisung	Karte Nr. (78–80)
*.PROGRAMM	MABLAUF			010
B1	GET	KARTE,EBER	1	020
	MVC	ANR,ENR	2	030
	CLC	EKA,K1	3	040
	BE	F2	4	050
	MVC	AMAT,EKOSTEN	-	060
G1	PUT	LISTE,ABER	7	070
	MVC	ABER,KBLANKS	8	080
	B	B1	9	090
F2	MVC	ALOHN,EKOSTEN	5	100
	B	G1	6	110
	END			120

Abb. 8

Verfolgen Sie anhand Ihres Programmablaufplans und der Abb. 8 A den Programmablauf bei einer Karte der KA "2" (Materialkarte).

(Die Ziffern 1-8 geben an, wie die Instruktionen nacheinander ablaufen).

Bei einer Karte der KA "2" werden nicht die Instruktionen in Zeile ____(Z)____ und ____(Z)____ durchlaufen.

* * * * *

100

110

Die Abb. 8 B zeigt den Programmablauf bei einer Karte der Kartenart KA "1".

Die Instruktion in Zeile ____(Z)____ wird hierbei nicht durchlaufen.

* * * * *

060

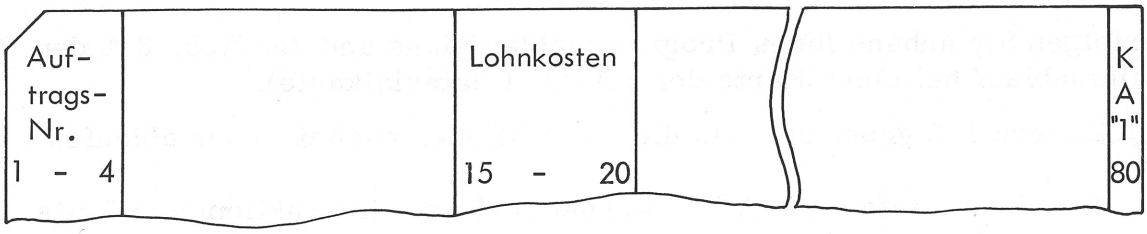
Verbessern Sie nun Ihre Codierung soweit notwendig.

Die Aufgabe 3 ist damit abgeschlossen.

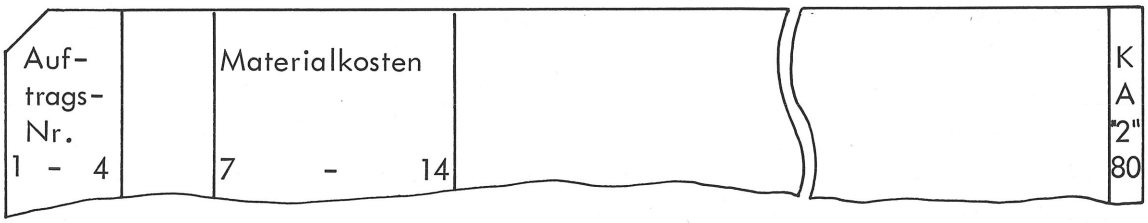
Aufgabe 4

Eingabe: Lohn- und Materialkarten

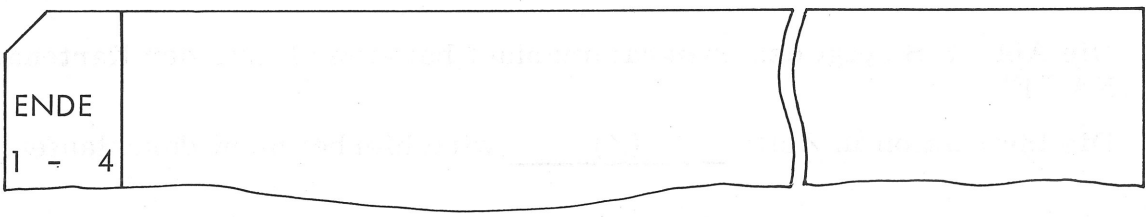
Lohnkarte



Materialkarte



"ENDE"-Karte



Ausgabe: Liste der Lohn- und Materialkosten

Auftrags-Nr.	Lohn- u. Material-kosten	Kostenart L = Lohn M = Material
0407	04050000	M
0407	00038000	M
0407	053345	L
0655	012438	L
0778	00525000	M
0778	155385	L
0824	11050000	M
0824	00895000	M
0824	534449	L
0824	01538000	M
1045	03740000	M
1045	086475	L
1125	06480000	M
1125	159453	L
10-13	17 - 24	27

Abb. 1

①

Aufgabe 4

Sehen Sie sich bitte die Aufgabe 4 (Abb. 1) genau an. -
Die Aufgabe 4 unterscheidet sich von der Aufgabe 3 dadurch, daß

a) Lohn- und Materialkosten in unterschiedlichen Spalten enthalten sind

b) Lohn- und Materialkosten ein gemeinsames Druckfeld haben und durch eine Kostenart "L" oder "M" zu kennzeichnen sind

c) das Ende des Kartenpakets durch eine "ENDE"-Karte angezeigt ist

Um die "ENDE"-Karte zu erkennen, müssen wir einen Vergleich der Lochkartenspalten 1-4 mit der Konstanten _____ durchführen.

* * * * *

ENDE

Wie in der Aufgabe 3 sind auch Lohn- und Materialkosten auseinanderzuhalten. Dies erfordert einen Vergleich auf _____.

* * * * *

Kartenart

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) die Aufgabe 4

b) ein Entwurfsblatt für Programmablaufpläne.

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 4 Nr.: _____ Seite: _____

Programmierer: _____ Datum: _____

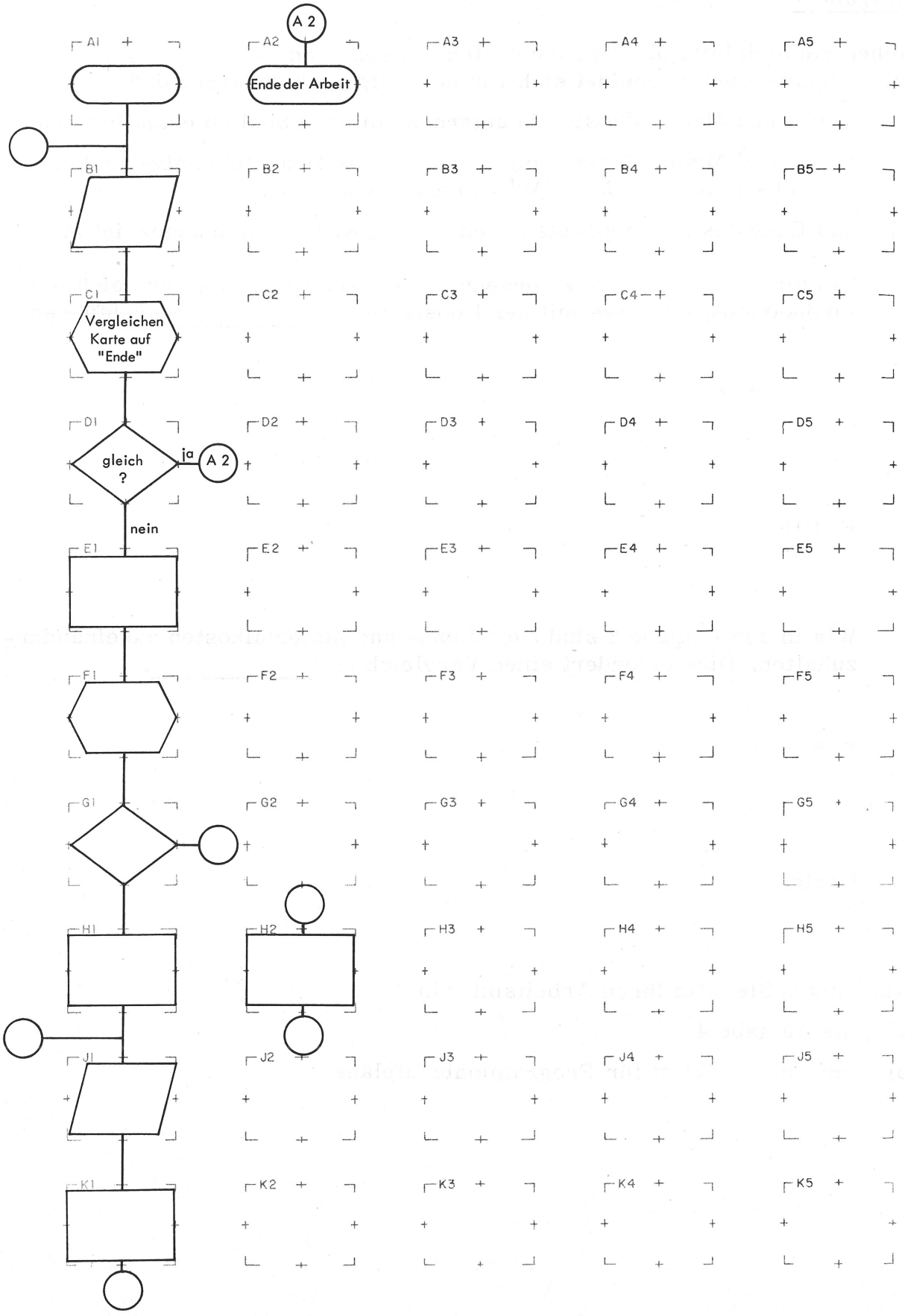


Abb. 2

Erstellen Sie nun den Programmablaufplan zu Aufgabe 4.

Die Abb. 2 gibt Ihnen dabei Hilfestellung.

* * * * *

Vergleichen Sie Ihren Programmablaufplan mit dem der Abb. 3 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 4 Nr.: Seite:

Programmierer: Datum:

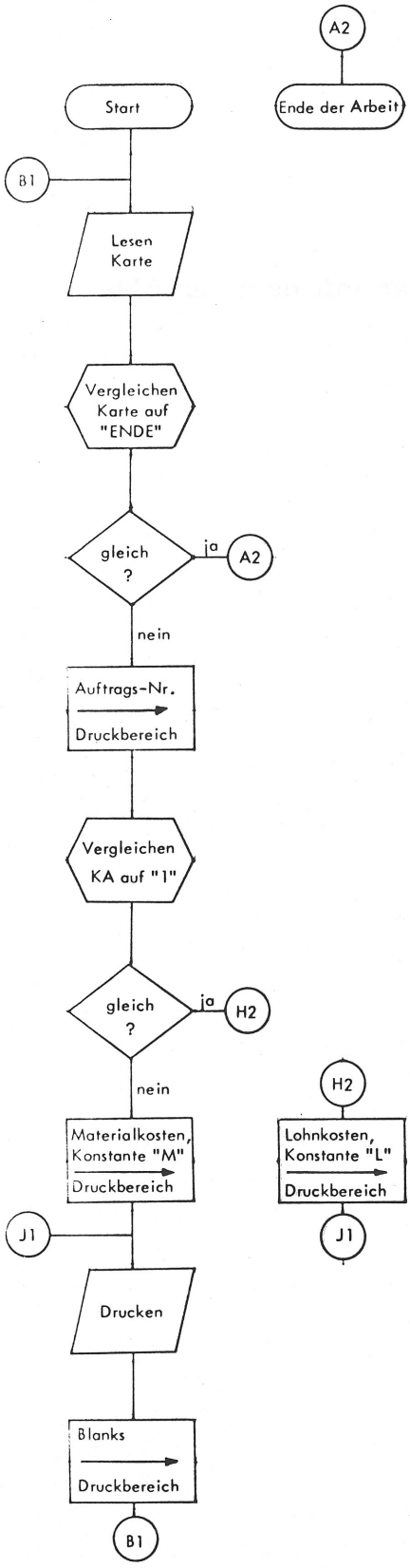


Abb. 3

Zum Programmablaufplan:

Man wird sofort nach dem Lesen einer Karte prüfen, ob nicht die "ENDE"-Karte vorliegt. Ihr Auftreten führt sofort zum Ende der Arbeit, während sonst die Verarbeitung der gelesenen Karte (Lohn- oder Materialkarte) beginnen kann (mit Block E1).

Falls Ihr Programmablaufplan Fehler enthält, brauchen Sie ihn nicht zu verbessern. Entnehmen Sie Ihren Arbeitsmitteln den Programmablaufplan zur Aufgabe 4.

IBM System / 360 Speicherbelegung

Aufgabe 4

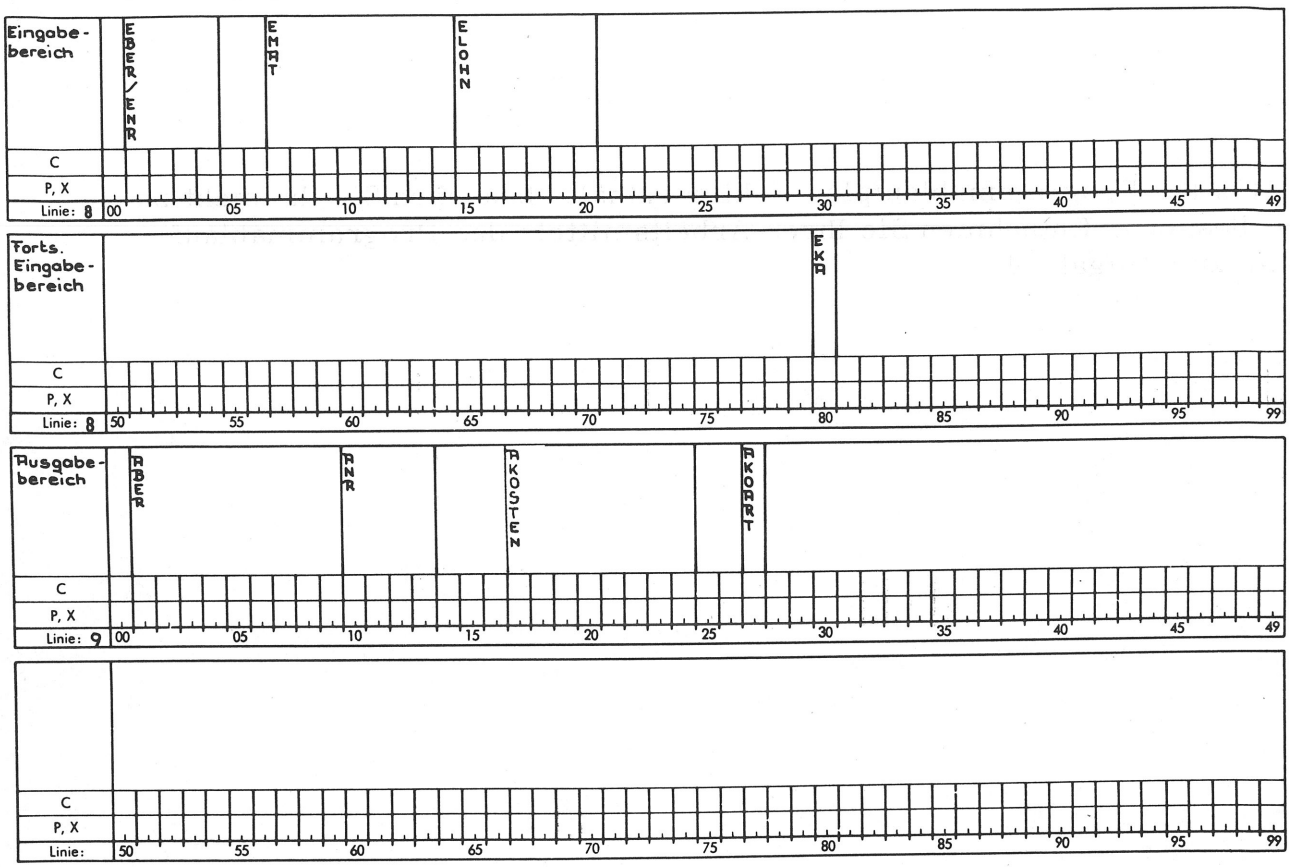


Abb. 4

Sehen Sie sich die Speicherbelegung in Abb. 4 an.

Die Felder EMAT und ELOHN sind (unterschiedlich, gleich) _____ groß.

Das Feld EMAT umfaßt ____(Z)____ Bytes, das Feld ELOHN ____(Z)____ Bytes.

* * * * *

unterschiedlich

8

6

Das Druckfeld AKOSTEN ist ____(Z)____ Bytes groß.

* * * * *

8

In Abhängigkeit von der Kartenart ist in das Feld AKOSTEN mal das 6 Bytes große Feld ELOHN (bei KA 1) und mal das 8 Bytes große Feld EMAT (bei KA 2) zu übertragen.

Betrachten Sie bitte die Lohnkosten in der dritten Ausgabezeile der Liste. -

Würde die folgende Instruktion zu diesem Listenbild führen? (Begründen Sie Ihre Antwort.)

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	MVC	AKOSTEN,ELOHN

* * * * *

Nein

Sinngemäß:

Der Betrag würde um 2 Bytes nach links versetzt, und außerdem würde noch der Inhalt von 2 weiteren Bytes (rechts von ELOHN) mit angeschrieben.

Das Listenbild würde also wie folgt aussehen:

```
(040500|00)      1. Zeile
(000380|00)      2. Zeile
 053345|XX       3. Zeile
```

Um eine wie im Listenbild gezeigte Ausgabe des Lohns zu erhalten, ist ELOHN nicht nach AKOSTEN, sondern um 2 Stellen weiter nach rechts zu übertragen. Außerdem muß man die 2 linken Bytes von AKOSTEN blanken. (Das Blanken des gesamten Druckbereichs ist nicht nötig.)

Wir können diese Forderungen erfüllen, indem wir neue Felder im Ausgabebereich definieren. Eine Lösung hierzu folgt.

Wesentlich einfacher gelangen wir durch die Möglichkeit zum Ziel, die es uns erlaubt, auch Teile eines Feldes anzusprechen.

Wie dies gemacht wird, lernen wir jetzt kennen.

Band 3, Seite 155 (Adreßrechnung und explizite Längenangabe)

Band 4, Seite 23 (MVC, 2. Teil)

Sie sind jetzt in der Lage, die Aufgabe 4 selbständig zu lösen.

Nehmen Sie bitte die Aufgabe und den Programmablaufplan zur Hand.

Vergegenwärtigen Sie sich die Aufgabe, indem Sie anhand des Programmablaufplans die einzelnen Verarbeitungsschritte noch einmal durchgehen.

Entnehmen Sie nun die Speicherbelegung zur Aufgabe 4.

Die erforderlichen Konstanten sind in die Speicherbelegung noch nicht eingetragen. Wir werden dies jetzt tun.

Stellen Sie fest:

a) wieviele Konstanten wir benötigen,

b) wozu wir diese Konstanten benötigen.

* * * * *

a) 5

b) zum Vergleich auf "ENDE"-Karte
zum Vergleich auf KA "1"
für Kostenart "L"
für Kostenart "M"
zum Blanken

Das Blanken der 2 linken Bytes von AKOSTEN kann auf verschiedene Weise geschehen. - Wir könnten den ganzen Druckbereich oder das Feld AKOSTEN oder auch nur die 2 linken Bytes von AKOSTEN blanken. Wir tun das letztere und übertragen hierzu eine 2 Bytes große Blankkonstante.

Tragen Sie bitte die Konstanten im Anschluß an den Ausgabebereich in folgender Reihenfolge und mit folgenden Namen in die Speicherbelegung ein:

KENDE, K1, KL, KM, KBLANKS

* * * * *

Vergleichen Sie Ihre Speicherbelegung mit der der Abb. 5 (Rückseite!).

IBM System / 360 Speicherbelegung

Aufgabe 4

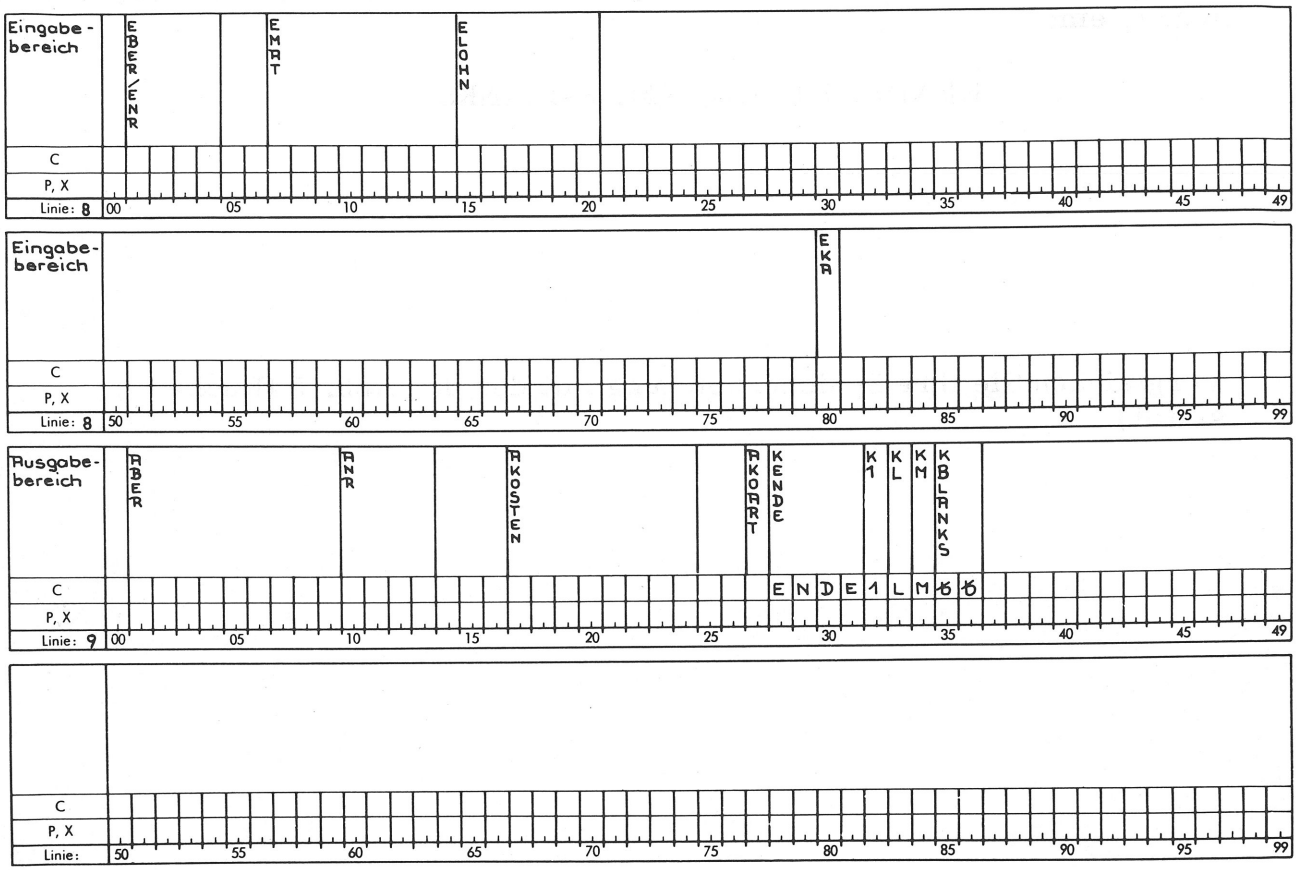


Abb. 5

Verbessern Sie bitte Ihre Speicherbelegung soweit erforderlich.

Codieren Sie nun die Aufgabe 4.

Schreiben Sie zuerst die Definitionen und codieren Sie dann den Programmablauf.

Den Schritt A2 codieren Sie bitte wie folgt:

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
A.2.	Ende der	Arbeit

Welche Instruktion hierfür einzutragen wäre, interessiert uns in diesem Lehrgang nicht.

* * * * *

Überprüfen Sie Ihre Codierung Zeile für Zeile anhand von Abb. 6 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 4	Lochhinweise	Problem A04 (73–75)
		Zeichen	
Programmierer	Datum	Lochung	Seite 01 (76–77) von 2

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
* LISTE DER LOHN- UND MATERIALKOSTEN			010
* DEFINITIONEN			020
	START	801	030
EBER	DS	0CL80	040
ENR	DS	CL4	050
	DS	CL2	060
EMAT	DS	CL8	070
ELOHN	DS	CL6	080
	DS	CL59	090
EKA	DS	CL1	100
	DS	CL20	110
ABER	DS	0CL27	120
	DS	CL9	130
ANR	DS	CL4	140
	DS	CL3	150
AKOSTEN	DS	CL8	160
	DS	CL2	170
AKOART	DS	CL1	180
KENDE	DC	C'ENDE' oder CL4'ENDE'	190
K1	DC	C'1' oder CL1'1'	200

IBM System / 360 Assembler
Programmschema

Programm		Lochhinweise	Problem A04 (73–75)
		Zeichen	
Programmierer	Datum	Lochung	Seite 02 (76–77) von 2

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
KL	DC	C'L' oder CL1'L'	010
KM	DC	C'M' oder CL1'M'	020
KBLANKS	DC	CL2' ' oder C' '	030
* PROGRAMMABLAUF			040
B1	GET	KARTE,EBER	050
	CLC	ENR,KENDE oder EBER(4),KENDE	060
	BE	A2	070
	MVC	ANR,ENR	080
	CLC	EKA,K1	090
	BE	H2	100
	MVC	AKOSTEN,EMAT	110
	MVC	AKOART,KM	120
J1	PUT	LISTE,ABER	130
	MVC	AKOSTEN(2),KBLANKS	140
	B	B1	150
H2	MVC	AKOSTEN+2(6),ELOHN	160
	MVC	AKOART,KL	170
	B	J1	180
A2	Ende der Arbeit		190
	END		200

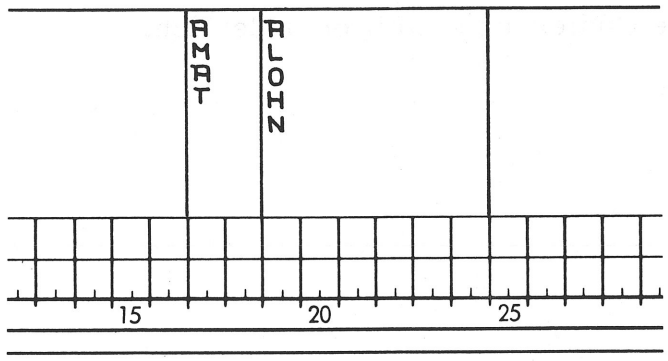
Abb. 6

Zur Codierung

Wenn Sie den Schritt A2 vor dem Schritt H2 codiert haben, ist dies natürlich auch richtig.

Verbessern Sie Ihre Codierung soweit erforderlich.

Ausgabe-
bereich



ABER	DS	ØCL27
	DS	CL9
ANR	DS	CL4
	DS	CL3
AMAT	DS	ØCL8
	DS	CL2
ALOHN	DS	CL6
	MVC	AMAT, EMAT
	:	:
	:	:
	MVC	ALOHN, ELOHN

Abb. 7

Bei der Übertragung von ELOHN in den Druckbereich wären wir auch ohne Adreßrechnung und explizite Längenangabe ausgekommen, wenn wir dem Druckfeld 2 Namen gegeben hätten und wie in Abb. 7 verfahren wären.

Sehen Sie sich die Lösung in Abb. 7 genau an. -

Wie wichtig jedoch Adreßrechnung und explizite Längenangabe sind, werden Sie bei den weiteren Aufgaben selbst feststellen.

Die Aufgabe 4 ist damit abgeschlossen. Legen Sie sie bitte ab. -

Ab der nächsten Aufgabe befassen wir uns mit Beispielen zu arithmetischen Operationen.

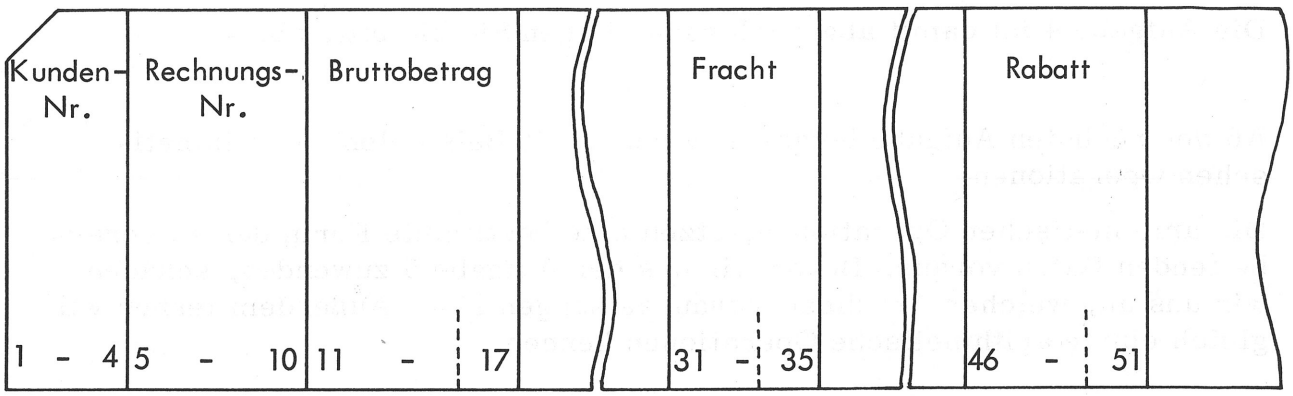
Die arithmetischen Operationen setzen eine bestimmte Form der zu verarbeitenden Daten voraus. Bevor wir uns der Aufgabe 5 zuwenden, schauen wir uns an, welcher Art diese Voraussetzungen sind. Außerdem lernen wir gleich einige arithmetische Operationen kennen.

→ Band 2, Seite 99 (Kapitel III)

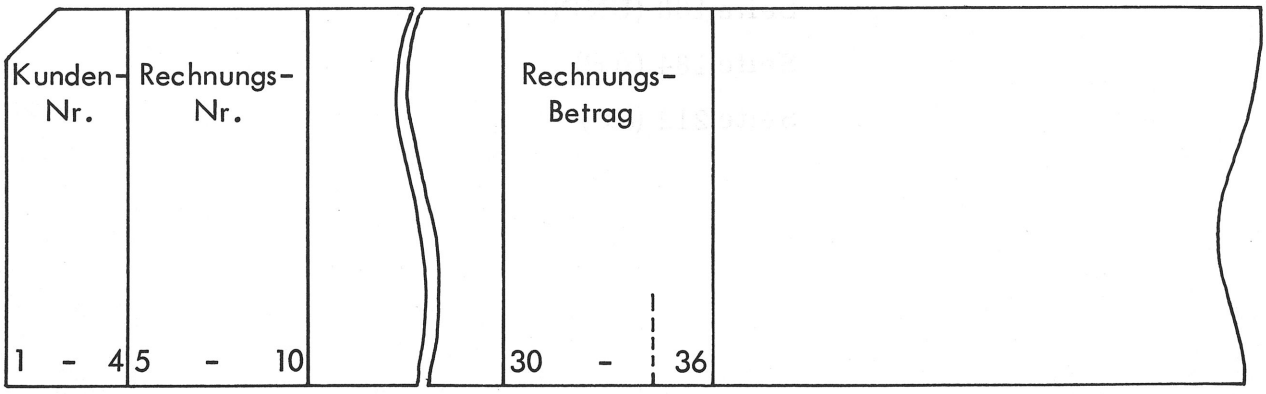
Band 4, Seite 146 (PACK)
Seite 168 (UNPK)
Seite 184 (AP)
Seite 212 (SP)

Aufgabe 5

Eingabe: Rechnungssummen-Karte



Ausgabe: Offene-Posten-Karte



Verarbeitung:

Je Kundennummer gibt es nur eine Rechnungssummen-Karte.

Rechnungsbetrag = Bruttobetrag - Rabatt + Fracht

Den Schluß des Kartenpakets bildet die "ENDE"-Karte (Spalten 1 - 4 enthalten ENDE).

Abb. 1

Aufgabe 5

Sehen Sie sich bitte die Aufgabe 5 (Abb. 1) genau an. -

Entnehmen Sie Ihren Arbeitsmitteln die Aufgabe 5.

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 5 Nr.: _____ Seite: _____

Programmierer: _____ Datum: _____

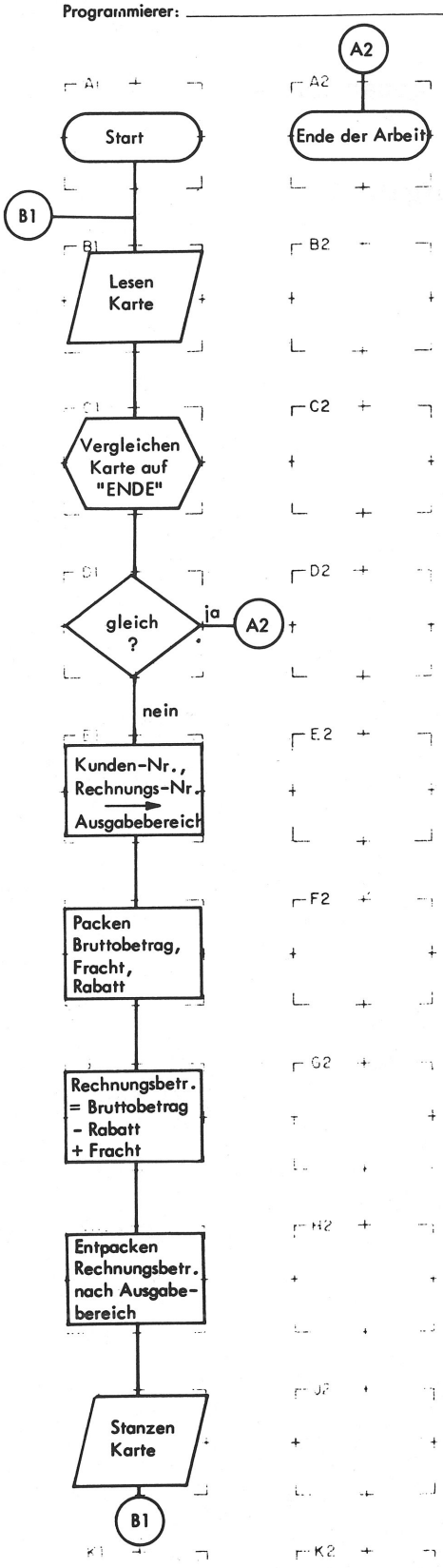


Abb. 2

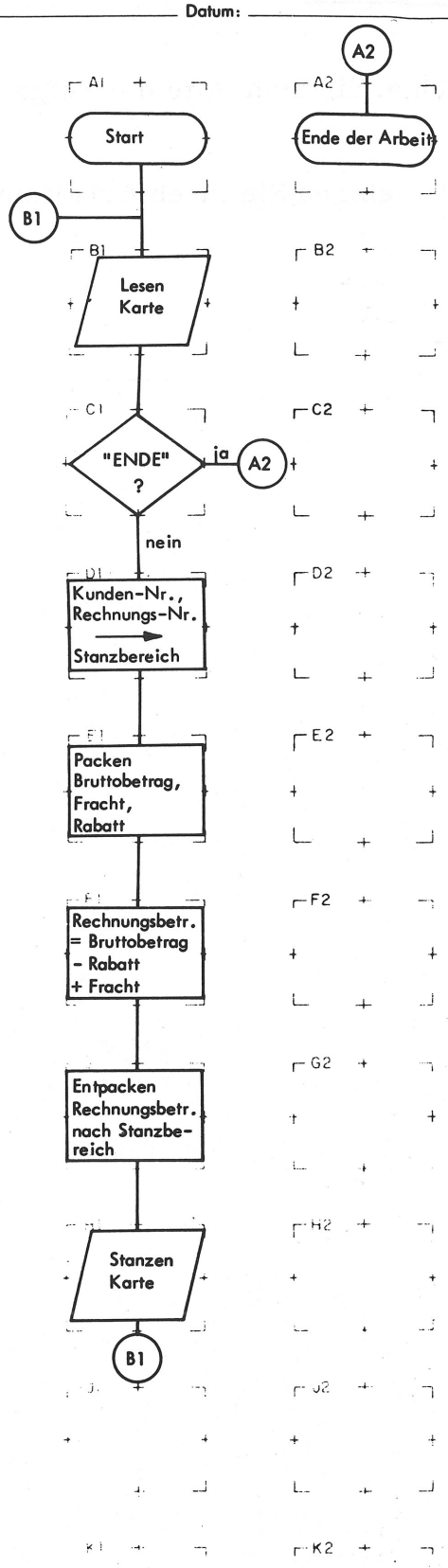


Abb. 3

Die Abb. 2 zeigt den Programmablaufplan (einen möglichen) zu Aufgabe 5. -

Verfolgen Sie den Programmablauf Schritt für Schritt. -

Anstelle einer Programm-Modifikation mit anschließender Entscheidung werden wir künftig nur noch den Entscheidungsblock verwenden. - In den Entscheidungsblock tragen wir dann ein, auf was zu prüfen ist.
In Abb. 3 ist die Prüfung auf "ENDE"-Karte zu einem Block zusammengefaßt.

An der Codierung ändert sich dadurch natürlich nichts.

Entnehmen Sie bitte Ihren Arbeitsmitteln den in Abb. 3 gezeigten Programmablaufplan zu Aufgabe 5.

Bevor wir die Speicherbelegung erstellen, sind noch einige Vorüberlegungen nötig.

Den Rechnungsbetrag bilden wir in einem besonderen Rechenfeld.

Das Rechenfeld muß ____(Z)____ Bytes groß sein (vgl. Ausgabekarte).

* * * * *

4

Zum Packen der Fracht und des Rabatts verwenden wir ebenfalls besondere Felder.

Das Feld zum Packen der Fracht muß ____(Z)____ Bytes umfassen.

Zum Packen des Rabatts benötigen wir ein ____(Z)____ Bytes großes Feld.

* * * * *

3

4

Zeichnen Sie nun die Speicherbelegung. Beachten Sie bitte folgende Angaben:

a) Eingabebereich ab 801,
Ausgabebereich ab 901.

b) Verwenden Sie folgende symbolische Namen für den Eingabe- und den Ausgabebereich:
EBER, EKDNR, ERNR, EBRUTTO, EFRACHT, ERABATT,
ABER, AKDNR, ARNR, ARBETRAG.

c) Unmittelbar im Anschluß an den Ausgabebereich sind die Konstante "ENDE" und die Rechenfelder zu speichern.

Die symbolischen Namen hierfür sind:
KENDE, RFRBETR, PFRACHT, PRABATT
(RF = Rechenfeld, P = gepacktes Feld).

* * * * *

Vergleichen Sie Ihre Speicherbelegung mit der der Abb. 4 (Rückseite!).

IBM System / 360
Speicherbelegung

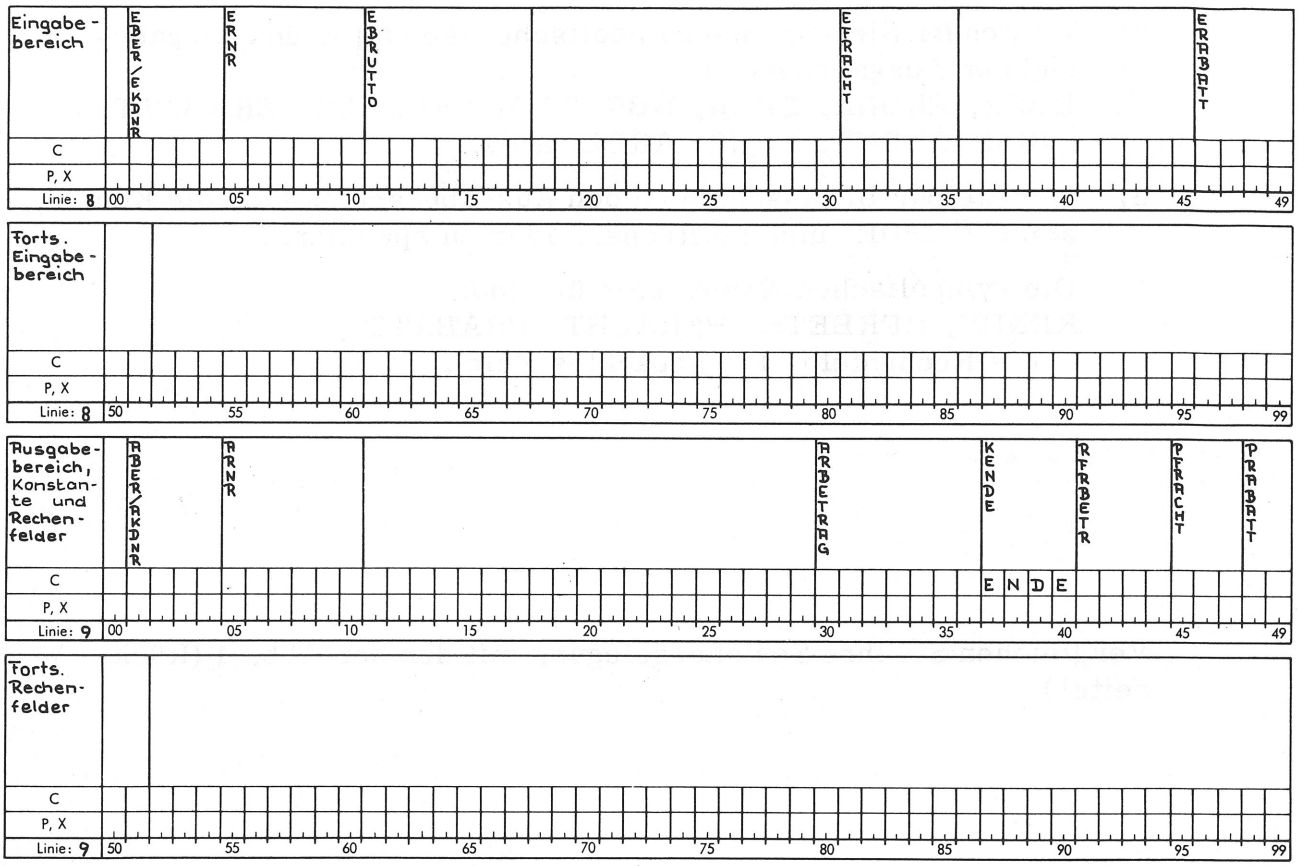


Abb. 4

Verbessern Sie Ihre Speicherbelegung soweit erforderlich.

Schreiben Sie jetzt die Definitionen. Verwenden Sie als Überschriften "Erstellen Offene-Posten-Karten" und "Definitionen".

* * * * *

Überprüfen Sie bitte Ihre Definitionen anhand der der Abb. 5 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm Aufgabe 5		Lochhinweise		Problem A Ø 5 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite Ø 1 (76–77) von 2

Anweisung			Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
*. ERSTELLEN OFFENE-POSTEN-KARTEN			Ø1Ø
*. DEFINITIONEN			Ø2Ø
	START	8Ø1	Ø3Ø
EBER	DS	ØCL51	Ø4Ø
EKDNR	DS	CL4	Ø5Ø
ERNR	DS	CL6	Ø6Ø
EBRUTTO	DS	CL7	Ø7Ø
	DS	CL13	Ø8Ø
EFRACHT	DS	CL5	Ø9Ø
	DS	CL1Ø	1ØØ
ERABATT	DS	CL6	11Ø
	DS	CL49	12Ø
ABER	DS	ØCL36	13Ø
AKDNR	DS	CL4	14Ø
ARNR	DS	CL6	15Ø
	DS	CL19	16Ø
ARBETRAG	DS	CL7	17Ø
KENDE	DC	C'ENDE'	18Ø
RFRBETR	DS	CL4	19Ø
PFRACHT	DS	CL3	2ØØ
PRABATT	DS	CL4	21Ø

Abb. 5

Verbessern Sie Ihre Definitionen soweit erforderlich.

Ein Hinweis für Lernende, die vom Lernweg 2 zum erstenmal in den Aufgabenband verzweigt sind:

In diesem Lehrgang befassen wir uns mit der Verarbeitung der Daten in der Zentraleinheit. Alle die Ein-/Ausgabe betreffenden Operationen werden in den Folgelehrgängen behandelt. Die Instruktionen für Lesen Karte,Drucken und Stanzen sind jedoch zur Abrundung der Beispiele in diesen Lehrgang aufgenommen. Die genaue Wirkungsweise dieser Instruktionen wird erst im nächsten Lehrgang besprochen.
Wir codieren für:

Lesen Karte -
Drucken -

	GET	KARTE,EBER
	PUT	LISTE,ABER

Der Operationsschlüssel GET bedeutet "Einlesen". PUT wird verwendet, wenn Daten auszugeben sind.

Der 1. Operand bezeichnet in symbolischer Form den Datenbestand (KARTE bzw. LISTE). Der 2. Operand gibt den Bereich (EBER bzw. ABER) an, in den die Daten eingelesen oder aus dem die Daten ausgegeben werden.

Codieren Sie bitte nun noch den Programmablauf.

Verwenden Sie als Überschrift "Programmablauf". "Stanzen Karte" ist wie folgt zu programmieren:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	PUT	STANZ,ABER

Anmerkung:

Die auszugebenden Lochkarten haben als Datenbestand den Namen STANZ.

* * * * *

Vergleichen Sie bitte Ihre Codierung mit der Abb. 6 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 5		Lochhinweise		Problem A05 (73–75)
			Zeichen		
Programmierer		Datum	Lochung		Seite 012 (76–77) von 2

Anweisung				Karte Nr.
Name (1–8)		Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
*.PROGRAMMABLAUF.				010
B1		GET	KARTE,EBER	020
		CLC	EKDNR,KENDE oder EBER(4),KENDE	030
		BE	B2	040
		MVC	AKDNR,EKDNR	050
		MVC	ARNR,ERNR	060
		PACK	RFRBETR,EBRUTTO	070
		PACK	PFRBCHT,EFRBCHT	080
		PACK	PRABART,ERABART	090
		SP	RFRBETR,PRABART	100
		AP	RFRBETR,PFRBCHT	110
		UNPK	ARBETRAG,RFRBETR	120
		PUT	STANZ,ABER	130
		B	B1	140
B2		Ende der	Arbeit	150
		END		160
				170
				180
				190
				200

Abb. 6

Zur Codierung des Programmablaufs:

Nehmen Sie bitte Ihre Speicherbelegung zur Hand.

Die Rechnungs-Nummer steht im Eingabebereich und im Ausgabebereich jeweils direkt im Anschluß an die _____.

Zur Übertragung der beiden Nummern vom Eingabe- in den Ausgabebereich würde also ___(WW)___ genügen.

* * * * *

Kunden-Nummer

eine Instruktion

Wie müßte die Instruktion lauten, wenn wir die Speicherbelegung nicht ändern würden?

* * * * *

a)

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	M.V.C.	A.K.D.N.R.(.1.Ø.),E.K.D.N.R.

oder :

b)

	M.V.C.	A.B.E.R.(.1.Ø.),E.B.E.R.

Die Namen der Felder, die im Programm nicht benutzt werden, können bei den Definitionen wegfallen.

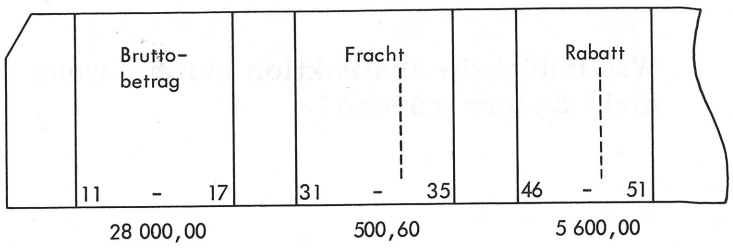
Eine andere Lösung wäre, im EBER und im ABER je ein 10 Bytes großes Feld zu bilden.

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
	:	
E.N.R.	D.S.	C.L.1.Ø.
	:	
A.N.R.	D.S.	C.L.1.Ø.
	:	
	:	
	M.V.C.	A.N.R.,E.N.R.

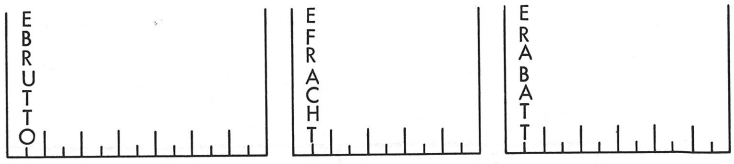
Verbessern Sie nun Ihre Codierung, sofern sie Fehler enthält.

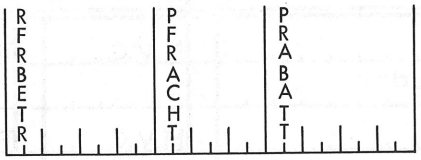
Übung zu Aufgabe 5

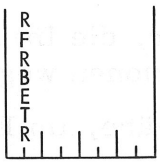
Tragen Sie bitte den jeweiligen Inhalt der Felder (mit Hexadezimalziffern) nach Ausführung der entsprechenden Instruktion(en) ein. Die Ausgangszahlenwerte stehen unter der Lochkarte.

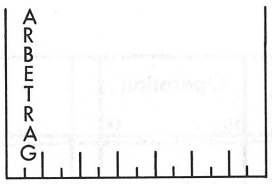



```
GET.. | KARTE,EBER.....
PACK. | RFRBETR,EBRUTTO..
PACK. | PFRACHT,EFRACHT..
PACK. | PRABATT,ERABATT..

SP... | RFRBETR,PRABATT..
AP... | RFRBETR,PFRACHT..

UNPK. | ARBETRAG,RFRBETR
```










```
PUT... | STANZ,ABER.....
```

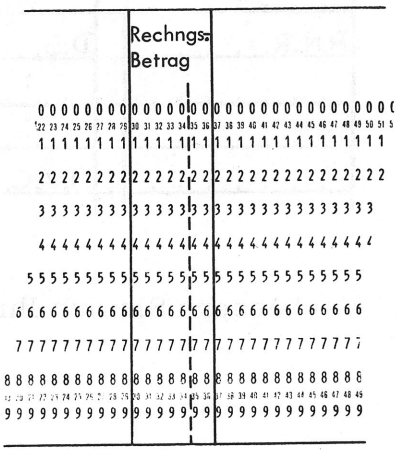


Abb. 7

Sehen Sie sich bitte die Abb. 7 an.

Wir werden jetzt an einem Zahlenbeispiel den Weg der Rechendaten von der Eingabe bis zur Ausgabe verfolgen.

Entnehmen Sie bitte Ihren Arbeitsmitteln die Übung 101.

Machen Sie sich klar, was zu tun ist.

Lösen Sie jetzt die Übung.

* * * * *

Überprüfen Sie bitte Ihre Lösung anhand der der Abb. 8 (Rückseite!).

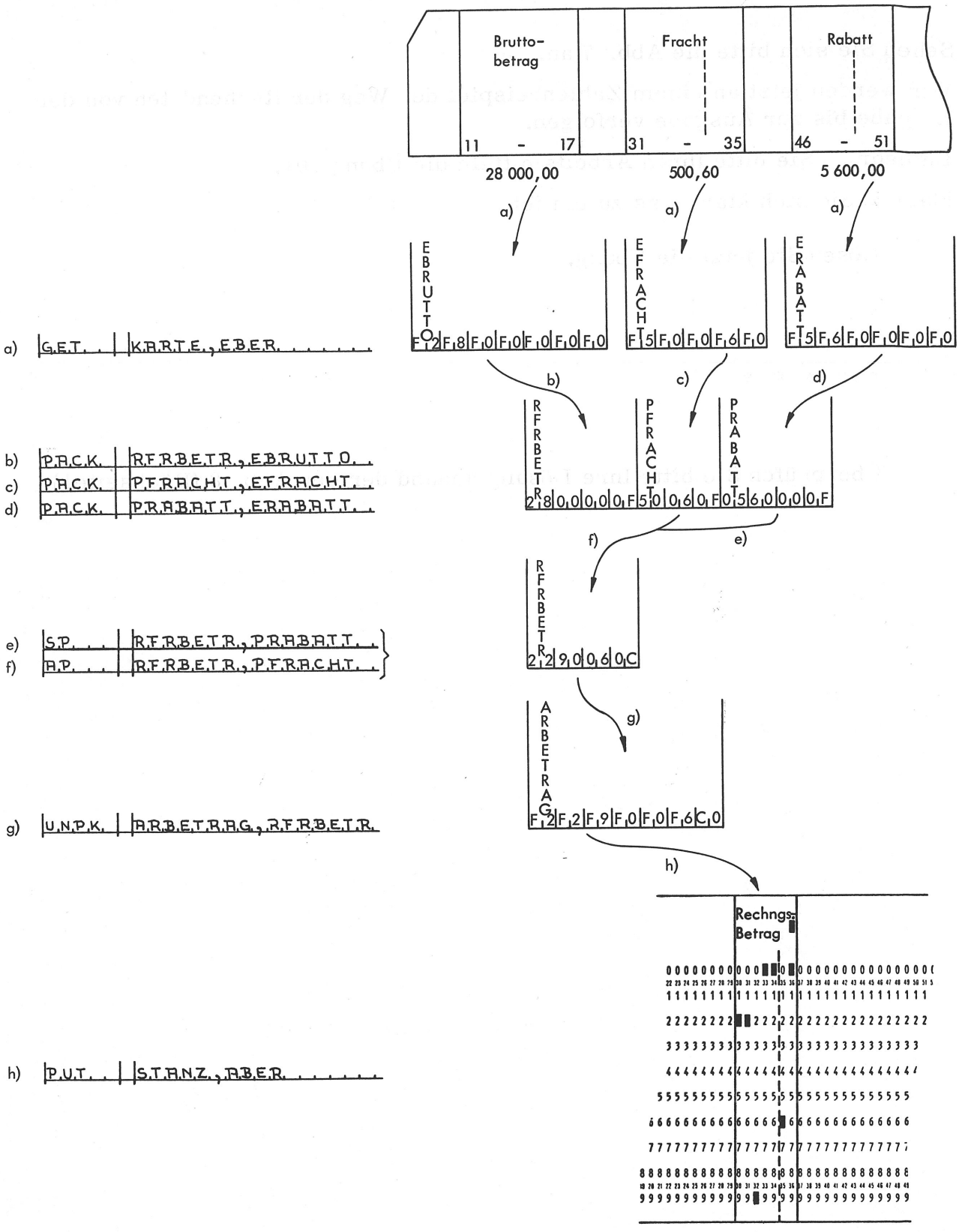

Brutto-
betrag
Fracht
Rabatt
11 - 17
31 - 35
46 - 51
28 000,00
500,60
5 600,00
a)
a)
a)
EBRUTTO
EFRACHT
ERABATT
F12F8F0F0F0F0F0F0
F5F0F0F6F0
F5F6F0F0F0F0
b)
c)
d)
RFRBETR
PFRACHT
PRABATT
2800000F
50060F
05600F
f)
e)
RFRBETR
2290600C
g)
ARBETRAG
F12F9F0F0F6C0
h)
Rechngs-
Betrag
a) GET.. KARTE,EBER.....
b) PACK. RFRBETR,EBRUTTO..
c) PACK. PFRACHT,EFRACHT..
d) PACK. PRABATT,ERABATT..
e) SP... RFRBETR,PRABATT..
f) AP... RFRBETR,PFRACHT..
g) UNPK. ARBETRAG,RFRBETR
h) PUT.. STANZ,ABER.....

Abb. 8

Zur Übung:

Haben Sie darauf geachtet, daß der Rechnungsbetrag mit Vorzeichen (12er-Lochung) in die Lochkarte gestanzt wird?

Der Rechnungsbetrag ist immer positiv. Es wird also in jede Karte eine _____(Z)_____er-Lochung als Vorzeichen gestanzt.

* * * * *

12

Wir werden jetzt untersuchen, was zu tun ist, wenn das Stanzen des Vorzeichens unterbleiben soll.

Wie müßte in Abb. 8 der Betrag in ARBETRAG stehen, wenn er ohne Zonenlochung in die Karte gestanzt werden sollte?

* * * * *

F 2	F 2	F 9	F 0	F 0	F 6	F 0

Wir müssen also in das linke Halbbyte des rechten Bytes ein hexadezimales _____ bringen.

Das rechte Halbbyte (darf, darf nicht) _____ verändert werden.

* * * * *

F

darf nicht

Läßt sich dies mit der Operation MVC erreichen? (Begründen Sie Ihre Antwort).

* * * * *

Nein

MVC verändert immer den Inhalt eines ganzen Bytes.

Es gibt nun Instruktionen, die nicht ganze Bytes, sondern Halbbytes ansprechen und unser Problem somit lösen.

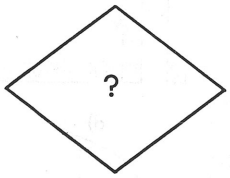
Diese Instruktionen lernen Sie jetzt kennen.

→ Band 4, Seite 30 (MVZ)

Seite 42 (MVN)

Die Aufgabe 5 ist nun so zu lösen, daß das Stanzen der 12er-Lochung unterbleibt. - Dies kann unter Verwendung der Instruktionen MVZ und MVN erreicht werden.

Wir müssen uns nun darüber klar werden, an welcher Stelle des Programmablaufs die MVN- oder MVZ-Instruktion einzufügen ist und wie diese lauten muß.

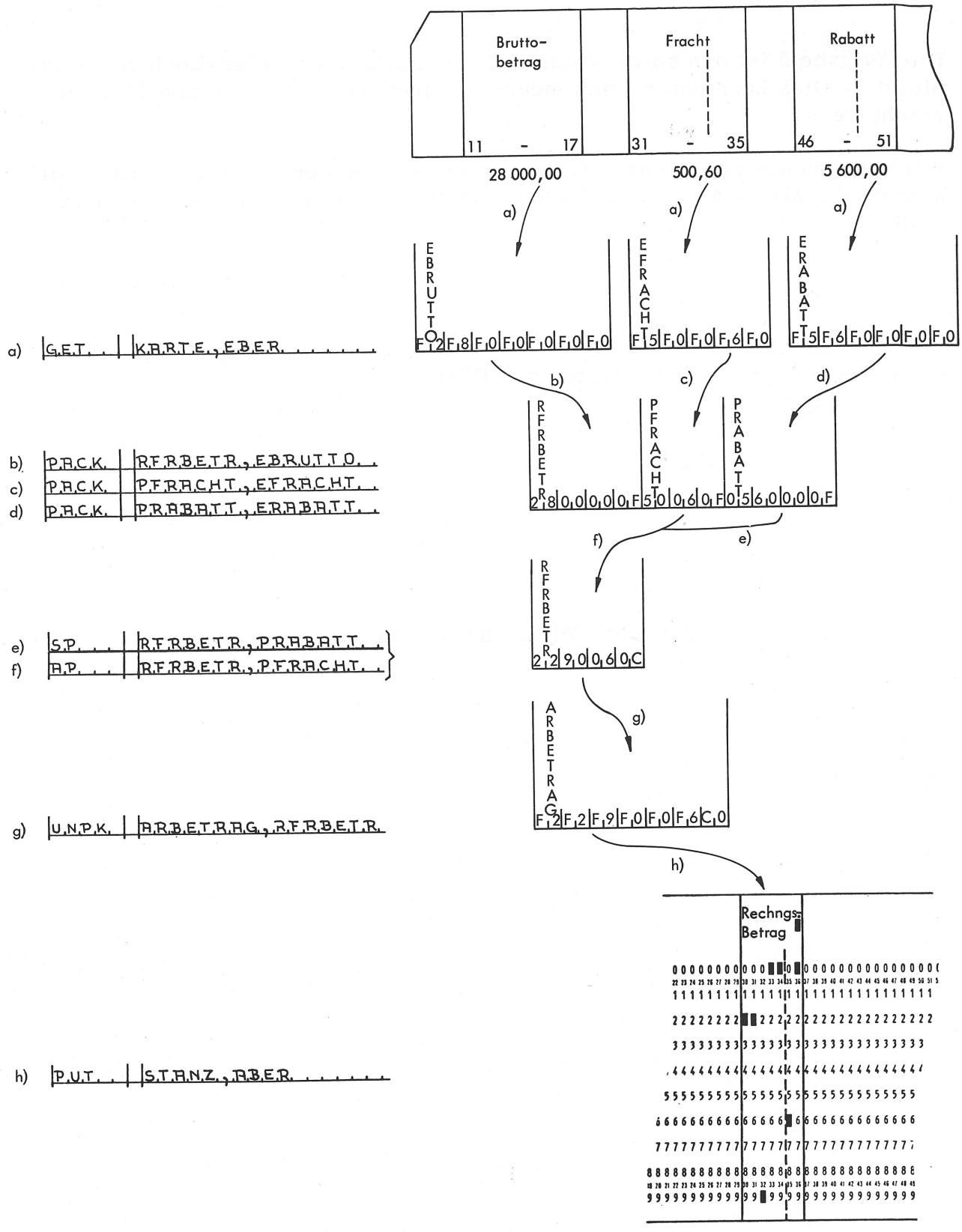


A Lösen ohne Hilfe

B Lösen mit Hilfe

* * * * *

A → (1), Seite 145



a) G.E.T. . | K.A.R.T.E., E.B.E.R. . . . . .

b) P.A.C.K. | R.F.R.B.E.T.R., E.B.R.U.T.T.O. .

c) P.A.C.K. | P.F.R.A.C.H.T., E.F.R.A.C.H.T. .

d) P.A.C.K. | P.R.A.B.A.T.T., E.R.A.B.A.T.T. .

e) S.P. . . | R.F.R.B.E.T.R., P.R.A.B.A.T.T. .

f) A.P. . . | R.F.R.B.E.T.R., P.F.R.A.C.H.T. .

g) U.N.P.K. | A.R.B.E.T.R.A.G., R.F.R.B.E.T.R.

h) P.U.T. . | S.T.A.N.Z., A.B.E.R. . . . . .

Abb. 8

1. Lösungshinweis

Betrachten Sie bitte Abb. 8.

In die rechte Spalte des Feldes "Rechnungsbetrag" soll nur die Ziffernlochung kommen.

Deshalb muß vor dem Stanzen in das linke Halbbyte des rechten Bytes von ARBETRAG ein hexadezimales ____(Z)____ gebracht werden.

* * * * *

F

Wir müssen also das hexadezimale ____(Z)____ durch ein hexadezimales F ersetzen.

* * * * *

C

Hierzu benötigen wir die Operation _____.

* * * * *

MVZ

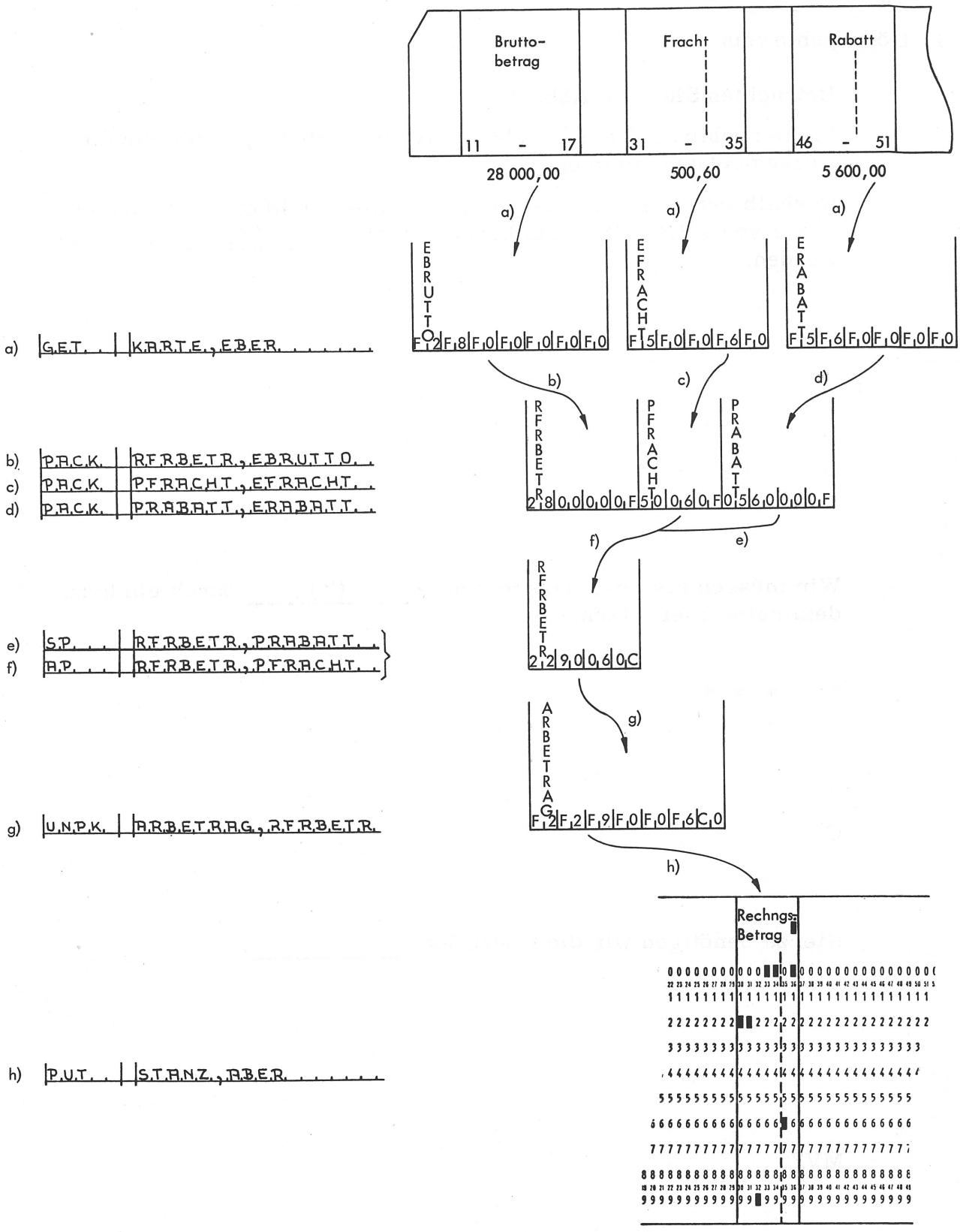

Brutto-
betrag
Fracht
Rabatt
11 - 17
31 - 35
46 - 51
28 000,00
500,60
5 600,00
a)
EBRUTTO
EFRACHT
ERABATT
b)
c)
d)
RFRBETR
PFRACHT
PRABATT
e)
f)
g)
ARBETRAG
h)
Rechngs-
Betrag
a) GET. KARTE, EBER.
b) PACK. RFRBETR, EBRUTTO.
c) PACK. PFRACHT, EFRACHT.
d) PACK. PRABATT, ERABATT.
e) SP. RFRBETR, PRABATT.
f) AP. RFRBETR, PFRACHT.
g) UNPK. ARBETRAG, RFRBETR.
h) PUT. STANZ, ABER.

Abb. 8

2. Lösungshinweis

Die UNPK-Instruktion soll zu folgendem Inhalt in ARBETRAG führen:

F 2	F 2	F 9	F 0	F 0	F 6	F 0

Dies bedingt, daß RFRBETR vor Ausführung der UNPK-Instruktion den Inhalt _____ hat.

* * * * *

2 2	9 0	0 6	0 F

Wir müssen also in das _____ Halbbyte des rechten Bytes von RFRBETR ein hexadezimales ____(Z)____ bringen.

Hierzu benötigen wir die Operation _____.

* * * * *

rechte

F

MVN

Ein hexadezimales F steht in Abb. 8 jeweils im rechten Halbbyte des rechten Bytes der Felder _____ und _____.

* * * * *

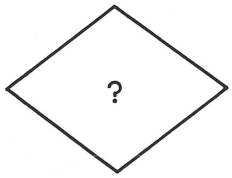
PFRACHT

PRABATT

Aufgrund dieser Lösungshinweise für die Verwendung von MVZ und MVN sollten Sie erkannt haben, wie das kleine Problem zu lösen ist.

(1) Nehmen Sie nun wieder die Übung 101 zur Hand.

Das Stanzen der 12er-Lochung ist zu verhindern.



A Lösung mit MVN

B Lösung mit MVZ

* * * * *

A → (3), Seite 149

(2) Die MVZ-Instruktion ist (vor, nach) _____ der _____ Instruktion einzufügen.

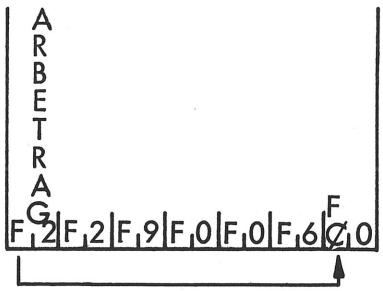
* * * * *

nach UNPK oder: vor PUT

Die Instruktion muß lauten: _____

* * * * *

Operation	Operand und Bemerkungen
10 14	20 25 30 35
MVZ	ARBETRAG+6(1),ARBETRAG



Anmerkung:

Der zweite Operand muß nicht ARBETRAG sein. Wir könnten genausogut jedes andere linke Halbbyte, das ein hexadezimales F enthält, übertragen, z. B. auch von den Feldern (s. Übungsblatt)

EBRUTTO (EBRUTTO, EBRUTTO+1.... EBRUTTO+6),
EFRACHT (EFRACHT, EFRACHT+1.... EFRACHT+4),
ERABATT (ERABATT, ERABATT+1.... ERABATT+5).

Es wäre auch möglich, eine entsprechende Konstante zu definieren und von dieser ein hexadezimales F zu übertragen.

Sollten Sie jedoch versucht haben, mit MVZ ein rechtes Halbbyte zu übertragen, so ist dies falsch. Mit MVZ werden immer nur linke Halbbytes übertragen.

Wenn Sie die Lösung mit MVN überspringen wollen → (4), Seite 151.

③ Die MVN-Instruktion ist (vor, nach) _____ der _____-Instruktion einzufügen.

* * * * *

nach AP oder: vor UNPK

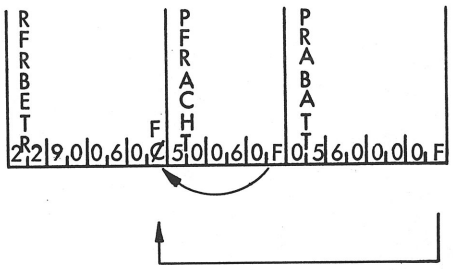
Die Instruktion muß lauten: _____

* * * * *

Operation	Operand und Bemerkungen
10 14	20 25 30 35
M,V,N, ,	R,F,R,B,E,T,R,+,3,(,1,),,P,F,R,A,C,H,T,+,2,

oder:

M,V,N, ,	R,F,R,B,E,T,R,+,3,(,1,),,P,R,A,B,A,T,T,+,3,



Jede andere Lösung ist falsch. (Es sei denn, Sie wissen schon mehr, als bisher zu lernen war.)

Wenn Sie die Lösung mit MVZ auch interessiert ② , Seite 147

④

Das Einfügen einer der beiden Instruktionen (MVZ oder MVN) in unsere Codierung der Aufgabe 5 ersparen wir uns.

Die Aufgabe 5 ist damit abgeschlossen.

Zum Vergleich von Dezimalzahlen (z.B. Rechenergebnissen) mit verschiedenen Vorzeichenverschlüsselungen genügt der logische Vergleich CLC nicht. Dieser Vergleich führt nicht in jedem Fall zu dem Ergebnis, daß eine p o s i t i v e Zahl größer ist als eine n e g a t i v e.

Darum gibt es neben dem logischen einen arithmetischen Vergleich.

Außerdem brauchen wir für den Vergleich mit (gepackten) Rechenergebnissen Konstanten, die in gepackter Form gespeichert sind.

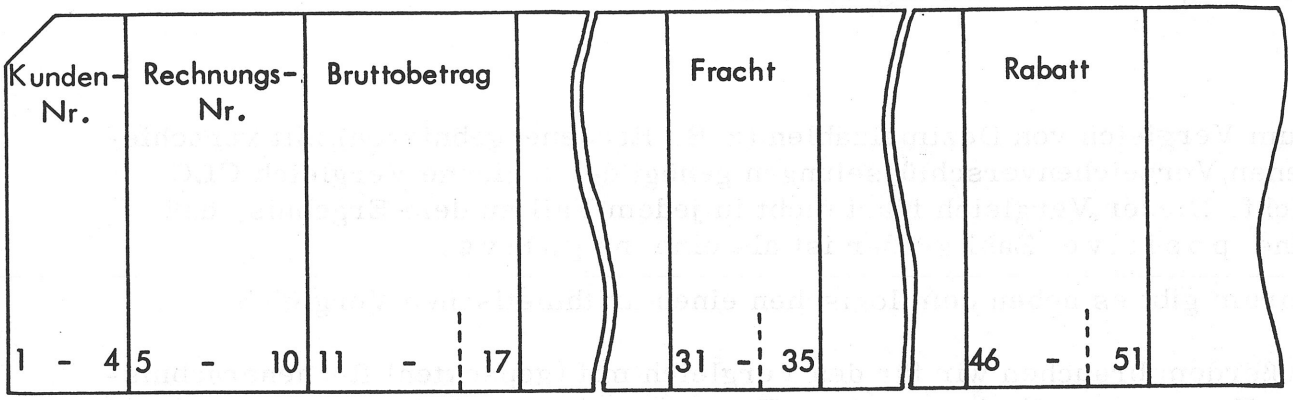
Wir lernen jetzt gepackte Konstanten und den arithmetischen Vergleich kennen.

→ Band 3, Seite 209 (DC-P)

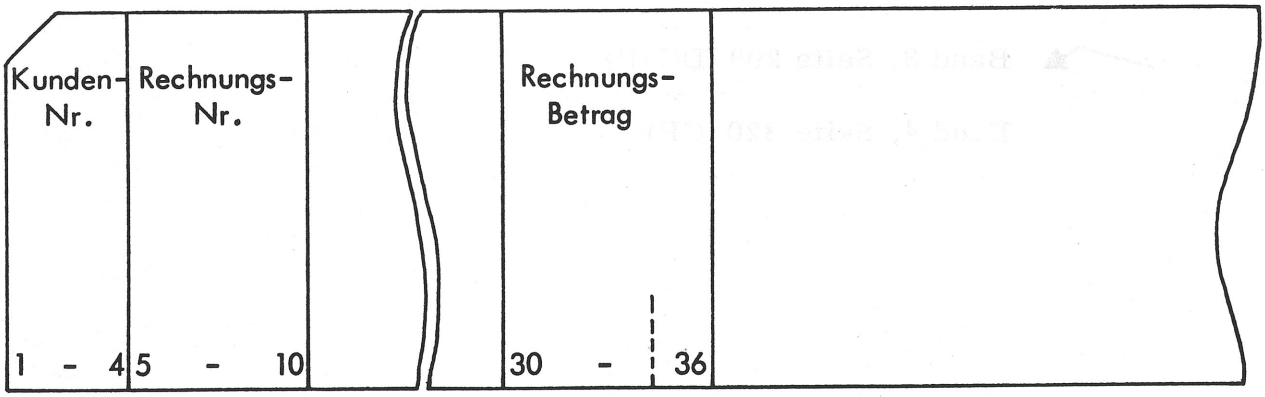
Band 4, Seite 320 (CP)

Aufgabe 6

Eingabe: Rechnungssummen-Karte



Ausgabe: Offene-Posten-Karte



Verarbeitung:

Je Kundennummer gibt es nur eine Rechnungssummenkarte.

Rechnungsbetrag = Bruttobetrag - Rabatt + Fracht.

Wenn der Nettobetrag (Bruttobetrag - Rabatt) größer als DM 1000,00 ist, wird keine Fracht berechnet. In diesem Fall ist der Nettobetrag gleich dem Rechnungsbetrag.

Der Rechnungsbetrag soll ohne 12er-Vorzeichen gestanzt werden.

Den Schluß des Kartenpakets bildet die "ENDE"-Karte (Spalten 1-4 enthalten ENDE).

Abb. 1

Aufgabe 6

Die Aufgabe 6 (Abb. 1) stellt eine Erweiterung der Aufgabe 5 dar. Die Fracht wird in dieser Aufgabe nur unter bestimmten Bedingungen in Rechnung gestellt.

Sehen Sie sich nun die Aufgabe genau an. -

Geben Sie bitte an, bei welchen der folgenden Nettobeträge die Fracht berechnet wird.

a) 784,15

b) 1227,30

c) 1000,01

d) 999,99

e) 1000,00

* * * * *

a) 784,15

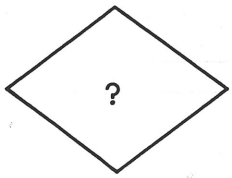
d) 999,99

e) 1000,00

Entnehmen Sie Ihren Arbeitsmitteln

a) die Aufgabe 6

b) ein Entwurfsblatt für Programmablaufpläne.



A Erstellen des Programmablaufplans ohne Hilfe

B Erstellen des Programmablaufplans mit Hilfe

* * * * *

A → (1), Seite 159

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 6 Nr.: Seite:

Programmierer: Datum:

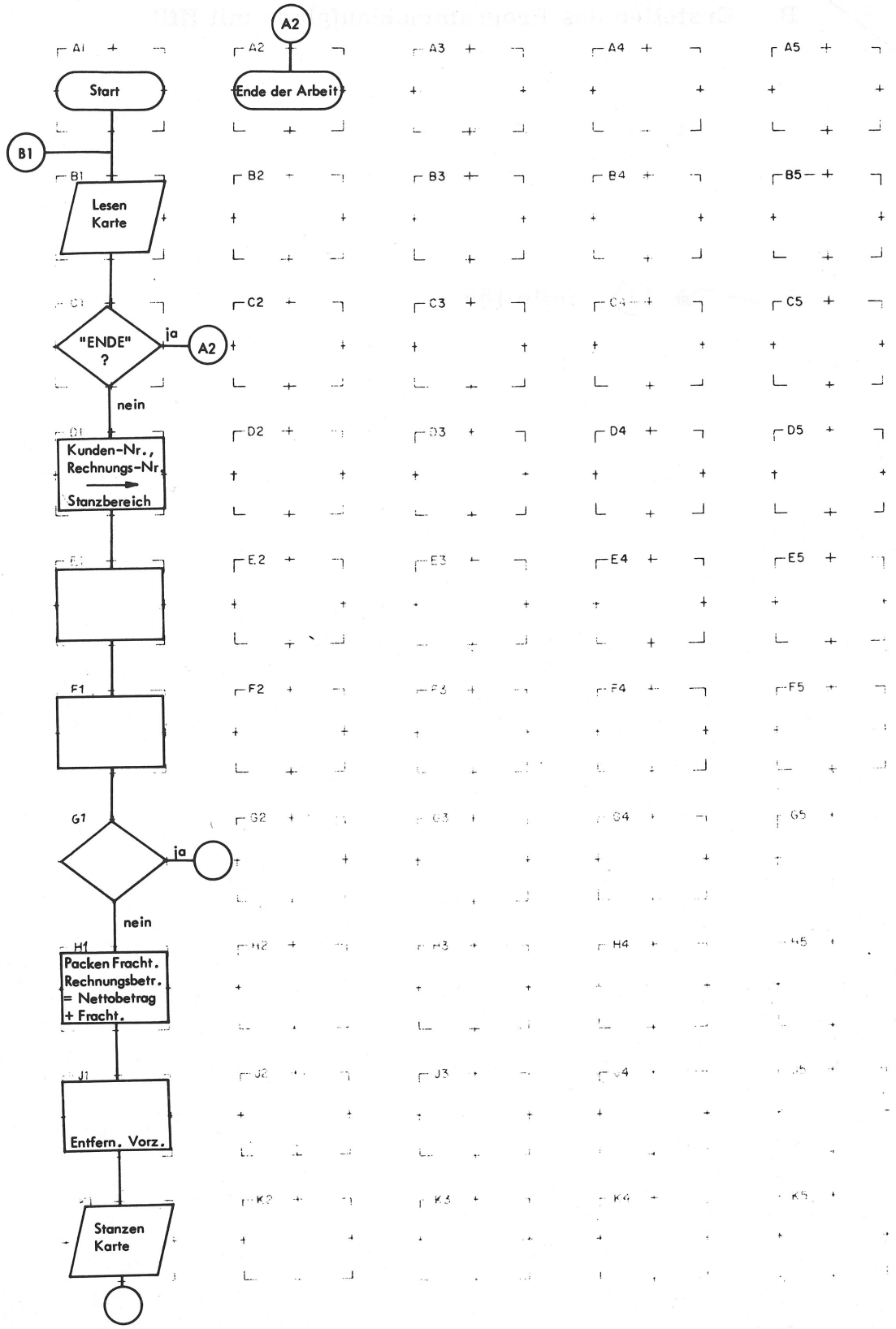


Abb. 2

Die Abb. 2 zeigt alle Blöcke des Programmablaufplans, teils mit, teils ohne Eintragungen.

Außer den fehlenden Eintragungen fehlt noch eine Schrittkennzeichnung. (Es ist noch der Schritt zu kennzeichnen, mit dem es weitergehen soll, wenn die Bedingung in Block G1 erfüllt ist).

Zeichnen Sie bitte jetzt den Programmablaufplan ab und ergänzen Sie die fehlenden Angaben.

* * * * *

→ (2), Seite 161

(1) Erstellen Sie den Programmablaufplan (vgl. Programmablaufplan zur Aufgabe 5).

Beachten Sie bitte:

Fassen Sie einen Vergleich mit anschließender Abfrage in einem Entscheidungsblock zusammen.

* * * * *

→ (2), Seite 161

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 6 Nr.: Seite:

Programmierer: Datum:

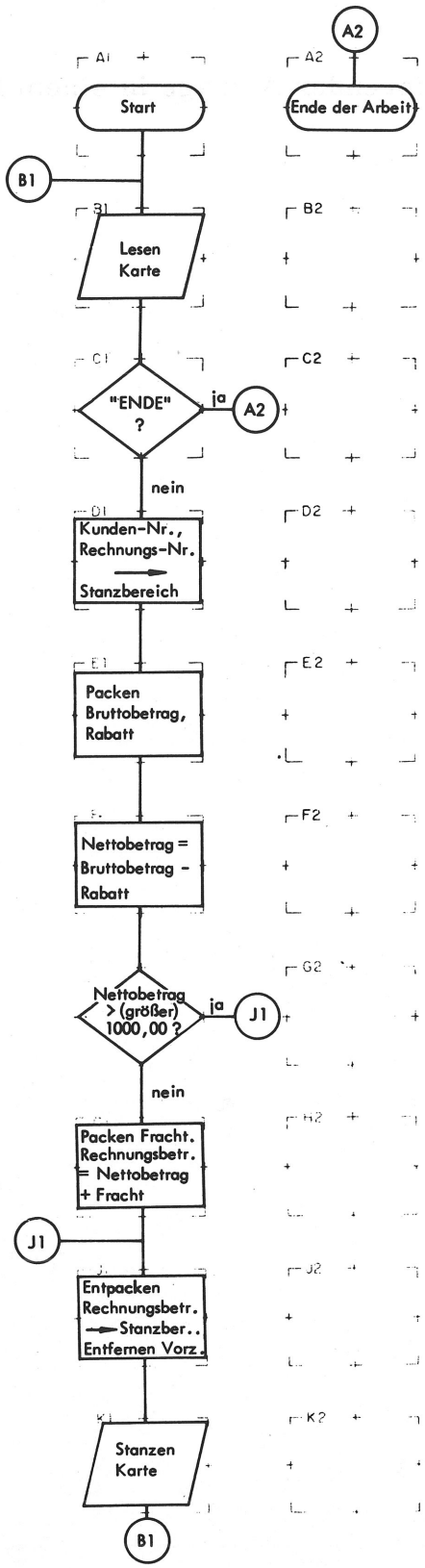


Abb. 3

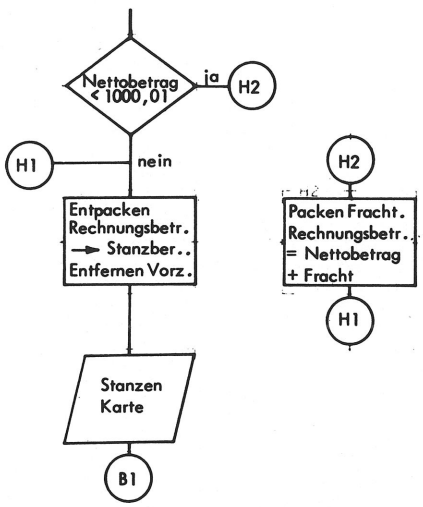


Abb. 4

(2)

Vergleichen Sie nun Ihren Programmablaufplan mit dem der Abb. 3. -

Zum Programmablaufplan:

Die Fracht könnte auch vor der Errechnung des Nettobetrages gepackt werden. Sinnvoller ist es jedoch, die Fracht erst dann zu packen, wenn feststeht, daß sie auch zu berechnen ist. - (Ein Nettobetrag größer DM 1 000,00 ist gleich dem Rechnungsbetrag).

Falls Sie die Prüfung des Nettobetrages auf < (kleiner) 1 000,01 vorgenommen haben (nur möglich, wenn Sie gesprungen sind), muß der weitere Programmablauf wie in Abb. 4 aussehen.

Allgemein gilt:

Es besteht keine Vorschrift darüber, wieviele Eintragungen ein Verarbeitungsblock enthalten darf oder soll. Ihr Programmablaufplan ist auch dann richtig, wenn Sie z. B. zwei aufeinanderfolgende Blöcke haben, in der vorgegebenen Lösung die gleichen Eintragungen aber in einem Verarbeitungsblock zusammengefaßt sind. - Das gleiche gilt auch für den umgekehrten Fall.

Eine einheitliche Codierung ist jedoch nur dann möglich, wenn wir uns an den vorgegebenen Programmablaufplan halten. -

Entnehmen Sie deshalb Ihren Arbeitsmitteln den Programmablaufplan zur Aufgabe 6.

Für den Vergleich des Nettobetrags mit 1 000,00 brauchen wir eine (Zeichenkonstante, gepackte Konstante) _____.

* * * * *

gepackte Konstante

Aufgabe 6

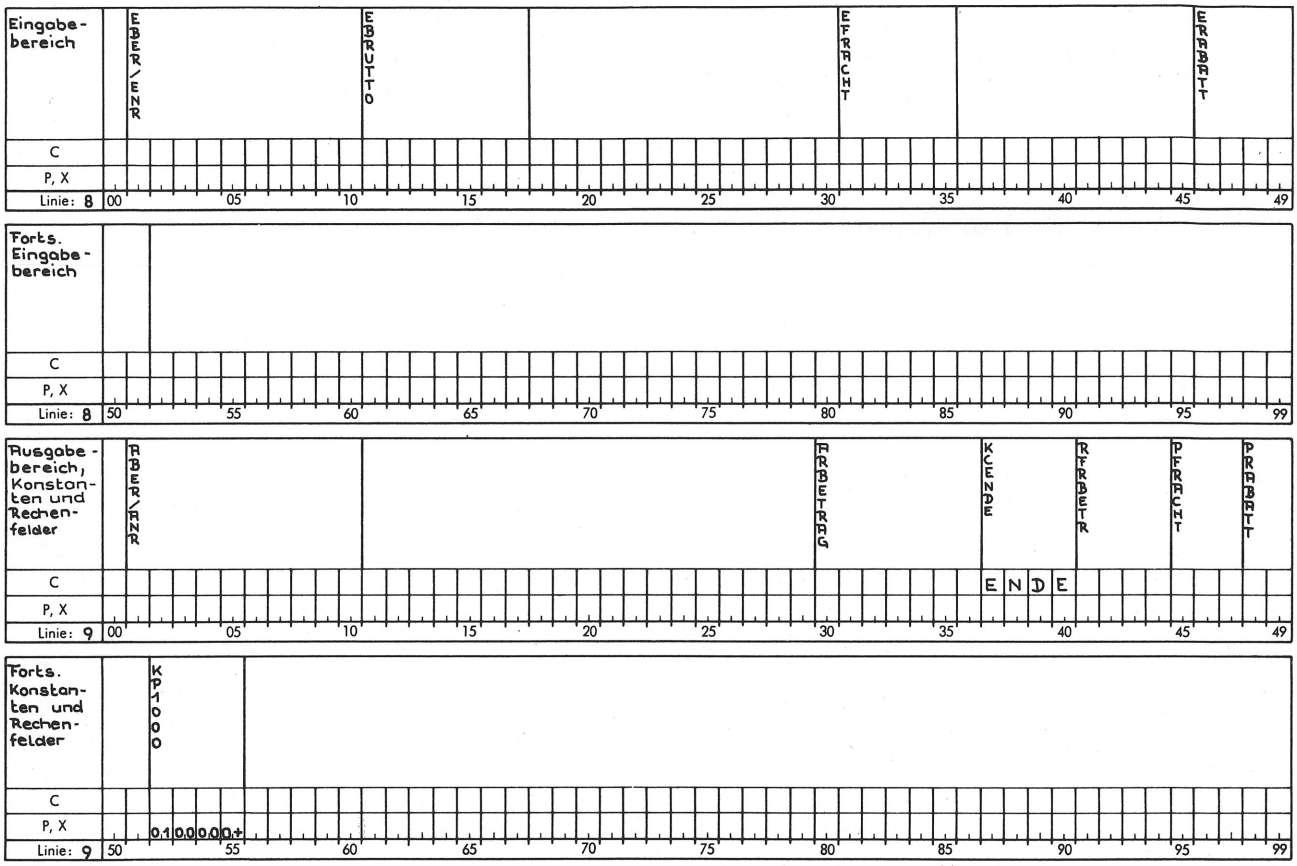


Abb. 5

Der Wert 1 000,00 belegt als gepackte Konstante ____(Z)____ Bytes.

Geben Sie an,

a) welchen Inhalt die Bytes erhalten,

b) wie die dazugehörige DC-Anweisung lauten muß.

* * * * *

4

a) |0 1|0 0|0 0|0 +|

b) . . | |D.C. . . . | |P.'1ØØØØØ'. .

Die Abb. 5 zeigt die Speicherbelegung. Sie stimmt bis auf zwei Änderungen mit der Speicherbelegung zur Aufgabe 5 überein. Die Kunden-Nr. und die Rechnungs-Nr. sind im Eingabe- und Ausgabebereich jeweils in einem Feld zusammengefaßt. Außerdem wurde die Konstante KP1000 hinzugefügt.

Künftig wollen wir bei den symbolischen Namen für die Konstanten den Konstantentyp mit zum Ausdruck bringen.

Deshalb vereinbaren wir, die Zeichenkonstanten KC... und die gepackten Konstanten KP... zu nennen (s. KCENDE und KP1000).

Entnehmen Sie Ihren Arbeitsmitteln

a) die Speicherbelegung zur Aufgabe 6

b) die dazugehörenden Definitionen.

Beachten Sie bitte:

Entgegen der Darstellung in der Speicherbelegung ist der Ausgabebereich so definiert, daß er direkt an den Eingabebereich anschließt.

Zur besseren Übersicht wird man öfters die Bereiche getrennt zeichnen, sie aber beim Definieren direkt aneinanderlegen.

Codieren Sie nun den Programmablauf.

* * * * *

Vergleichen Sie Ihre Codierung mit der der Abb. 6 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm Aufgabe 6		Lochhinweise	Problem A06 (73–75)
		Zeichen	
Programmierer	Datum	Lochung	Seite 012 (76 77) von 2

Anweisung: Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.PROGRAMM	MABLAUF		010
B1	GET	KARTE,EBER	020
	CLC	EBER(4),KCENDE oder ENR(4),KCENDE	030
	BE	A2	040
	MVC	ANR,ENR	050
	PACK	RFRBETR,EBRUTTO	060
	PACK	PRABATT,ERABATT	070
	SP	RFRBETR,PRABATT	080
	CP	RFRBETR,KP1000	090
	BH	J1	100
	PACK	PFRACHT,EFRACHT	110
	AP	RFRBETR,PFRACHT	120
J1	UNPK	ARBETRAG,RFRBETR	130
	MVZ	ARBETRAG+6(1),ARBETRAG	140
	PUT	STRANZ,ABER	150
	B	B1	160
A2	Ende der Arbeit		170
	END		180
			190
			200

Abb. 6

Zur Codierung des Programmablaufs:

Falls Sie zu dem Vergleich von RFRBETR mit der Konstanten KP 1000 die Operation CLC benutzt haben, ist dies natürlich _____.

* * * * *

(auch) richtig

Denn RFRBETR und KP 1000 haben dieselbe Vorzeichenverschlüsselung (hexadezimal C) und dieselbe Länge (4 Bytes).
Vgl. hierzu . . . Band 3, Seite 213, LE 6
und Band 4, Seite 212, Vorzeichen bei der Operation SP.

Verbessern Sie nun Ihre Codierung soweit erforderlich.

Die Aufgabe 6 ist abgeschlossen.

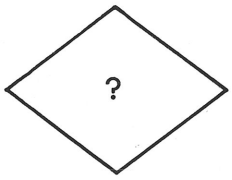
Wir kennen bisher zwei Arten von Konstanten, die Zeichenkonstanten (C) und die gepackten Konstanten (P). Wir werden jetzt eine weitere Konstantenart kennenlernen.

→ Band 3, Seite 223 (DC-X)

Wir wenden uns jetzt der Multiplikation zu.

Auf den nächsten Seiten werden die Terminologie und die allgemeinen Regeln der Multiplikation (Schulwissen) behandelt. -

Im einzelnen geht es um folgende Begriffe und Regeln: Faktor, Multiplikand, Multiplikator, Produkt, Stellenzahl und Anzahl der Dezimalstellen des Produkts, Runden, Vorzeichen.



A Ich bin sicher, daß ich diese Dinge beherrsche.

B Ich möchte mein Wissen hierzu wieder auffrischen.

* * * * *

A → (1), Seite 183

Zur Terminologie der Multiplikation

9	x	3	=	27
Faktor	x	Faktor	=	Produkt

Das Ergebnis einer Multiplikation von zwei _____ ist das _____.

* * * * *

Faktoren

Produkt

8	x	4	=	32
Multiplikand	x	Multiplikator	=	Produkt

Die beiden Faktoren werden auch _____ und _____ genannt.

* * * * *

Multiplikand

Multiplikator

Der Faktor mit der geringeren Stellenanzahl sollte als Multiplikator dienen, damit die Multiplikation möglichst schnell abläuft.

Nachfolgend sind für fünf Multiplikationen je zwei Faktoren (A und B) dargestellt. Wählen Sie bitte jeweils den Multiplikator aus.

	A	B	Multiplikator
a)	X X X	X X X X	_____
b)	X X X X	X X X	_____
c)	X X	X X X X X	_____
d)	X	X X X	_____
e)	X X X X	X X X X	_____

* * * * *

a) A

b) B

c) A

d) A

e) A oder B

Die Stellenzahl des Produktes

Das Produkt umfaßt maximal soviel Stellen wie die Summe der Stellen beider Faktoren.

Wenn der Multiplikator vier Stellen und der Multiplikand sechs Stellen umfaßt, so hat das Produkt maximal ____(Z)____ Stellen.

* * * * *

10

Bestimmen Sie für die drei folgenden Beispiele die maximale Stellenzahl des Produktes:

	Multiplikand	Multiplikator	Produkt
a)	X X X	X X	(Z)
b)	X X X X X	X X X X X	(Z)
c)	X X X X X X X	X X	(Z)

* * * * *

a) 5

b) 10

c) 9

Die Anzahl der Dezimalstellen im Produkt

Häufig haben die Faktoren eine oder mehrere Dezimalstellen (Stellen hinter dem Komma, wie z.B. bei der Zahl "33,85").

Es gilt: Die Anzahl der Dezimalstellen des Produktes ist gleich der Summe der Dezimalstellen der Faktoren.

Bei der Multiplikation der Faktoren XX, XX und X,XXX ergeben sich für das Produkt ____(Z)____ Dezimalstellen.

* * * * *

5

Wenn wir den Lohnsatz je Std. von DM 3,41 mit einer Stundenzahl von 38,75 multiplizieren, so erhalten wir unter Vernachlässigung des Kommas "1321375".

Das Komma muß zwischen den Ziffern ____(Z)____ und ____(Z)____ des Produktes eingesetzt werden.

Unter Berücksichtigung des Kommas ergibt sich somit "____(Z)____".

* * * * *

2 (und) 1

132,1375

Das Runden von Dezimalzahlen

Das Produkt von DM 132,1375 ist nicht gerundet. - Durch Runden (auf DM und Pf) ergibt sich ein Betrag von DM _____(Z)_____.

* * * * *

132,14

Gerundet wird durch Addition einer "5".

Soll das gerundete Produkt n Stellen umfassen, so wird die "5" in die (n + 1)-te Stelle addiert. - Dabei ist die linke Stelle als die erste, die rechte Stelle als die letzte Stelle zu betrachten.

Wollen wir also aus dem ungerundeten Produkt "132,1375" das gerundete Produkt "132,14" erhalten, so müssen wir in der _____ Stelle eine _____(Z)_____ addieren.

Das gerundete Produkt reicht dann bis zur _____ Stelle.

* * * * *

6-ten

5

5-ten

Das Vorzeichen des Produktes

Das Vorzeichen des Produktes ist abhängig von den Vorzeichen der Faktoren.

Haben die Faktoren gleiche Vorzeichen (entweder "plus" oder "minus"), so ist das Produkt positiv. Sind hingegen die Vorzeichen ungleich ("plus" und "minus"), so ist das Produkt negativ.

Bestimmen Sie bitte das Vorzeichen des Produktes (+ oder -), wenn folgende Vorzeichen der Faktoren gegeben sind:

	Faktor	Faktor	Vorz. des Produktes
a)	+	+	_____
b)	+	-	_____
c)	-	+	_____
d)	-	-	_____

* * * * *

a) +

b) -

c) -

d) +

Sie sind nun wieder mit den allgemeinen Grundlagen der Multiplikation vertraut.

(1) Nun lernen wir die Programmierung der Multiplikation im System /360 kennen. Hierzu benötigen wir 2 neue Instruktionen.

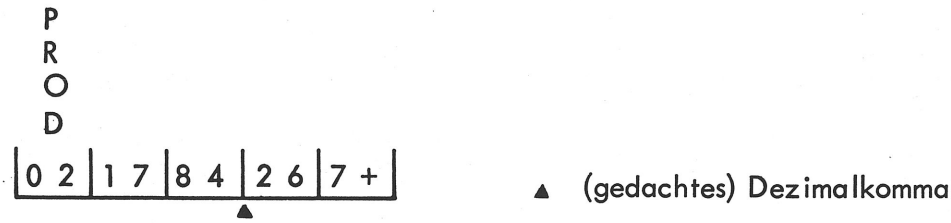
→ Band 4, Seite 222 (ZAP)

Seite 252 (MP)

Nachdem Sie nun wissen, mit welcher Instruktion man zwei Zahlen miteinander multipliziert, unter welchen Voraussetzungen diese Multiplikation abläuft und zu welchen Resultaten sie führt, üben wir nun das Runden der Produkte.

Man wird geneigt sein, vom handschriftlichen Rechnen her das Runden als höchst einfach abzutun. Daß beim Programmieren hingegen das Runden nicht so einfach ist, werden wir nun sehen.

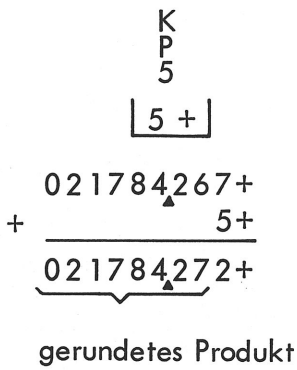
Bei einer Multiplikation haben wir folgendes Produkt erhalten:



Das Produkt ist zu runden und mit zwei Dezimalstellen auszugeben. Wir runden, indem wir eine Konstante ____(Z)____ addieren.

* * * * *

5



Codieren Sie bitte hierzu:

a) die Definition der Konstanten KP5

b) die erforderliche Instruktion.

* * * * *

Zur Kontrolle Ihrer Eintragungen blättern Sie bitte um.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35 40
KP5	DC	P'+5' oder P'5' oder X'5C'
	:	
	:	
	AP	PROD,,KP5

Das nachstehende Produkt ist zu runden.

P
R
O
D

0 2	1 7	8 4	2 6	7 +

▲

Das gerundete Produkt ist mit einer Dezimalstelle auszugeben.
Zum Runden benötigen wir eine Konstante mit folgendem Inhalt:
_____.

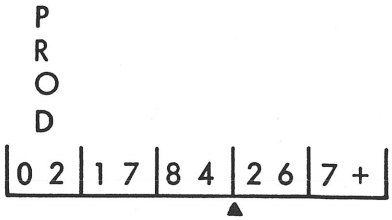
* * * * *

0 5	0 +

	0 2	1 7	8 4	2 6	7 +
			▲		
+				0 5	0 +
	0 2	1 7	8 4	3 1	7 +
			▲		

Beschreiben Sie bitte, was die nachstehende Instruktion bewirken würde.

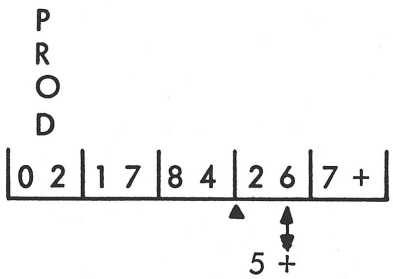
Name (1–8)	Operation (10–14)	Operand und Bemerk (20–25)
KP5	DC	P'+5'
	:	
	:	
	BP	PROD(4),KP5



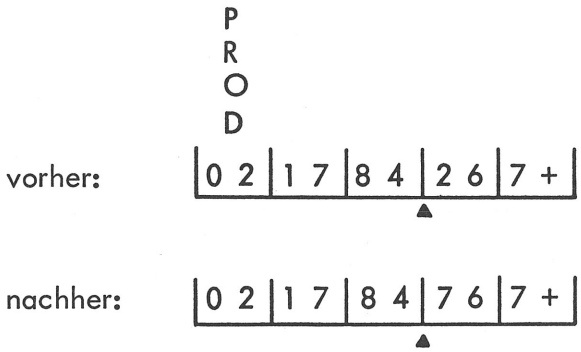
* * * * *

Sinngemäß:

Diese Instruktion würde zu einem Gültigkeitsfehler führen. Der erste Operand PROD(4) hat in seinem rechten Halbbyte kein Vorzeichen.



Wie müßte die Codierung der Konstanten und der Instruktion lauten, damit die Rundung folgendes Ergebnis liefert:



* * * * *

Name	Operation	Operand und Bemerkung
1 8	10 14	20 25
K.P.5.Ø.Ø.	D.C.	P.'+5.Ø.Ø.'
	:	
	:	
	R.P.	P.R.O.D.,K.P.5.Ø.Ø.

Runden von Produkten mit wechselnden Vorzeichen

Wir sind bisher davon ausgegangen, daß das zu rundende Produkt immer ein positiver Wert ist.

Das nachstehende Produkt ist zu runden und mit 2 Dezimalstellen auszugeben.

Das Produktfeld hat folgenden Inhalt:

P
R
O
D

0 2	1 7	8 4	2 6	7 -
		▲		

Als Ergebnis ist dann (Z) auszugeben.

* * * * *

021784,27-

Welches Ergebnis erhalten wir aufgrund dieser Instruktion?

Name	Operation	Operand
1 8	10 14	20
K.P.5.	D.C. . .	P.'+5.' . . . .
. . . . .	. . : . .	. . . . .
. . . . .	R.P. . .	P.R.O.D.,K.P.5.

P
R
O
D

0 2	1 7	8 4	2 6	7 -

* * * * *

P
R
O
D

0 2	1 7	8 4	2 6	2 -

Diese Rundung führt also nicht zu dem gewünschten Ergebnis.

Damit diese Instruktion ein richtiges Ergebnis bringt, muß die zu addierende Konstante den Inhalt _____ haben.

* * * * *

K
P
5

5 -

Anmerkung:

Zum gleichen Ergebnis kämen wir, wenn wir die Konstante 5+ subtrahieren würden.

Wir halten fest:

Ein positiver Wert wird gerundet, indem eine _____ Konstante addiert wird.

Ein negativer Wert wird gerundet, indem eine _____ Konstante addiert wird.

* * * * *

positive

negative

Bei Produkten mit w e c h s e l n d e n Vorzeichen (mal positiv, mal negativ) müssen wir darauf achten, daß die Konstante mit dem Vorzeichen addiert wird, das das jeweilige Produkt hat.

Es gibt verschiedene Möglichkeiten, dies zu programmieren.

Wir könnten im Anschluß an die Multiplikation das Vorzeichen des Produktes prüfen und anschließend

a) bei positivem Produkt eine positive Konstante und bei negativem Produkt eine negative Konstante addieren,

oder

b) jeweils eine positive Konstante addieren (bei positivem Produkt) bzw. subtrahieren (bei negativem Produkt).

Es gibt jedoch eine einfachere Methode, bei der wir ohne Prüfung des Produktvorzeichens und mit nur einer Konstanten auskommen.

Wir verfahren hierbei wie folgt:

Vor der Rundung übertragen wir das Vorzeichen des _____ in die Vorzeichenstelle der _____.
Dadurch wird bei der anschließenden Addition

a) bei einem positiven Produkt eine _____ Konstante addiert,

b) bei einem negativen Produkt eine _____ Konstante addiert.

* * * * *

Produktes

Konstanten

positive

negative

Zur Übertragung des Vorzeichens benötigen wir die Operation _____.

* * * * *

MVN

P
R
O
D

0 X	X X	X X	X X	X V

Das obige Produktfeld hat wechselnde Vorzeichen.

Schreiben Sie auf ein Programmschema die Instruktionen und Konstanten, die erforderlich sind, wenn die Rundungs- "5"

a) in die letzte (rechte) Stelle zu addieren ist,

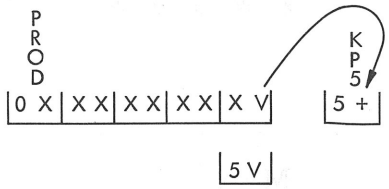
b) in die viertletzte Stelle zu addieren ist.

* * * * *

Überprüfen Sie Ihre Codierung anhand der Abb. 1 (Rückseite!).

a)

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,N,	K,P,5,,P,R,O,D,+,4,
	A,P,	P,R,O,D,,K,P,5,
	:	
K,P,5,	D,C,	P,',5,'



b)

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	M,V,N,	K,P,5,Ø,Ø,Ø,+,2,(,1,),,P,R,O,D,+,4,
	A,P,	P,R,O,D,,K,P,5,Ø,Ø,Ø,
	:	
K,P,5,Ø,Ø,Ø,	D,C,	P,',5,Ø,Ø,Ø,'

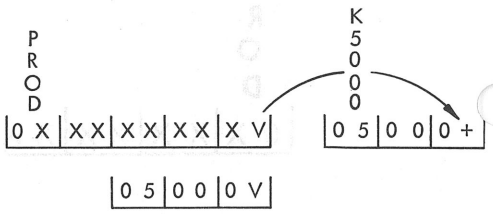


Abb. 1

Bei der weiteren Verarbeitung eines gerundeten Produktes werden die Dezimalstellen ab der Rundungsstelle vernachlässigt.

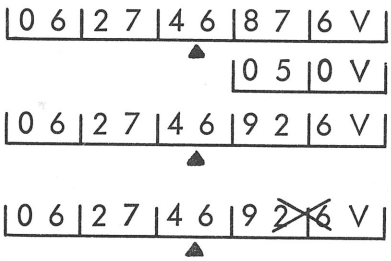
Beispiel:

Produkt vor der Rundung

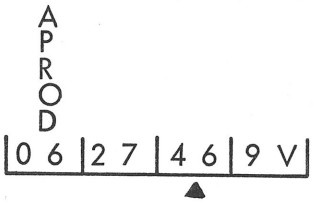
Rundungskonstante

Gerundetes Produkt

Zu vernachlässigende Dezimalstellen ✕

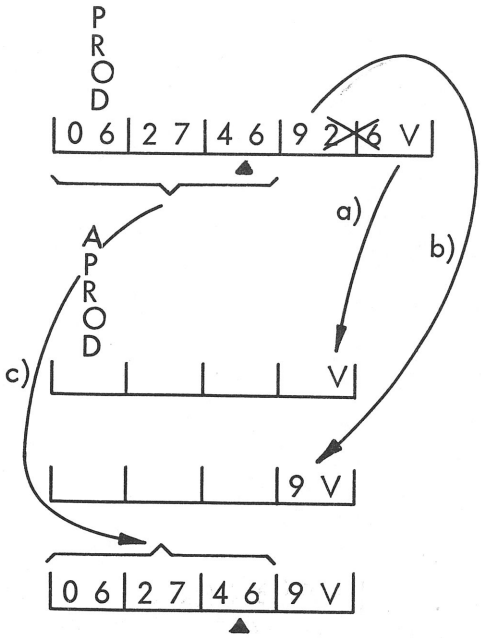


Wir untersuchen, was zu tun ist, wenn das gerundete Produkt wie folgt nach APROD gebracht werden soll:



Folgende Lösung wäre denkbar:

	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
a)	MVN	APROD+3(1),PROD+4
b)	MVZ	APROD+3(1),PROD+3
c)	MVC	APROD(3),PROD



Die 3 Instruktionen könnten auch in jeder anderen Folge stehen.

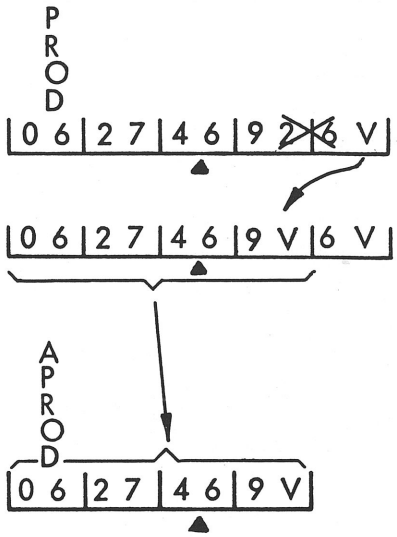
Es gibt eine einfachere Lösung, bei der nur 2 Instruktionen erforderlich sind. - Versuchen Sie diese Lösung zu finden.

* * * * *

Hier die 2 möglichen Lösungen:

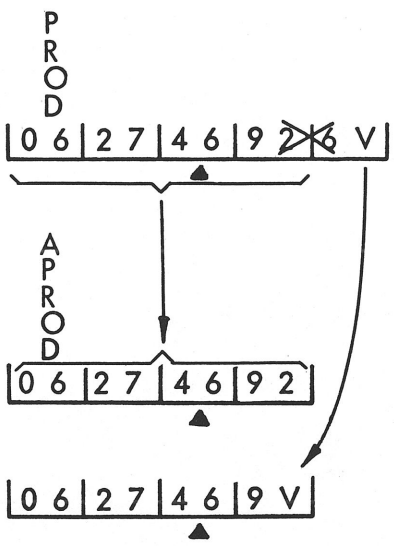
Lösung A

Operation	Operand und Bemerkungen
10 14	20 25 30
M.V.N. .	P.R.O.D.+3.(.1.).,P.R.O.D.+4.
. . . . .	. . . . .
M.V.C. .	A.P.R.O.D.,P.R.O.D. . . . .
oder: Z.A.P. .	A.P.R.O.D.,P.R.O.D.(.4.) . . .

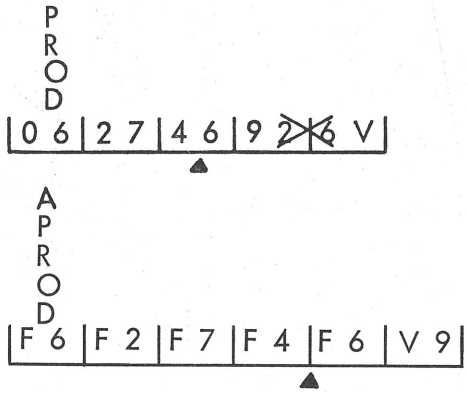


Lösung B

Operation	Operand und Bemerkungen
10 14	20 25 30
M.V.C. .	A.P.R.O.D.,P.R.O.D. . . . .
M.V.N. .	A.P.R.O.D.+3.(.1.).,P.R.O.D.+4.



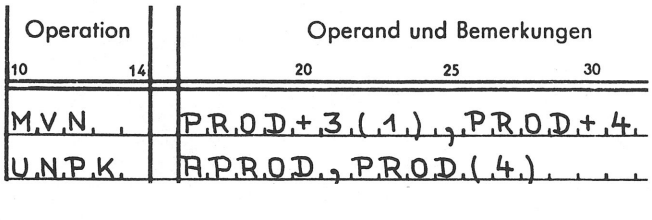
Ähnlich verfahren wir, wenn das gerundete Produkt ungepackt (entpackt) nach APROD zu bringen ist.

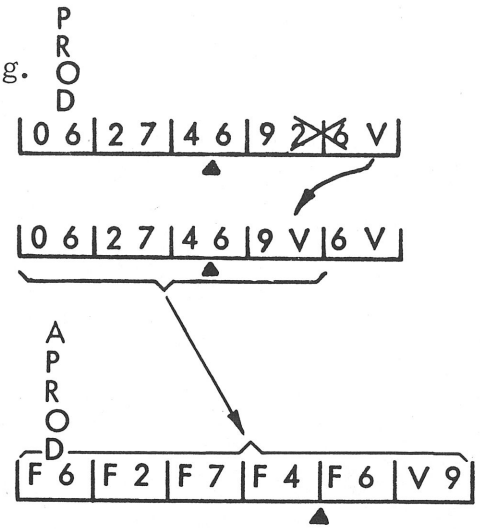


Schreiben Sie bitte nur die Instruktionen, die wir in diesem Fall benötigen.

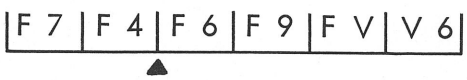
* * * * *

Hierbei gibt es nur eine sinnvolle Lösung.



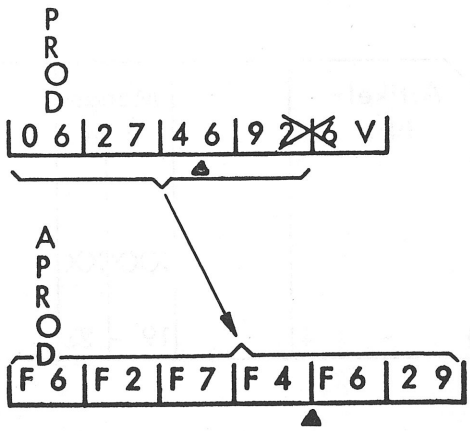


Falls Sie die explizite Längenangabe im 2. Operanden der UNPK-Instruktion weggelassen haben, erhalten Sie folgendes unschöne Ergebnis in APROD:



Nicht zum Ziel kommen wir, wenn wir gleich mit UNPK beginnen.

Operation	Operand und Bemerkung
10 14	20 25
U.N.P.K.	A.P.R.O.D.,.P.R.O.D.(.4.).

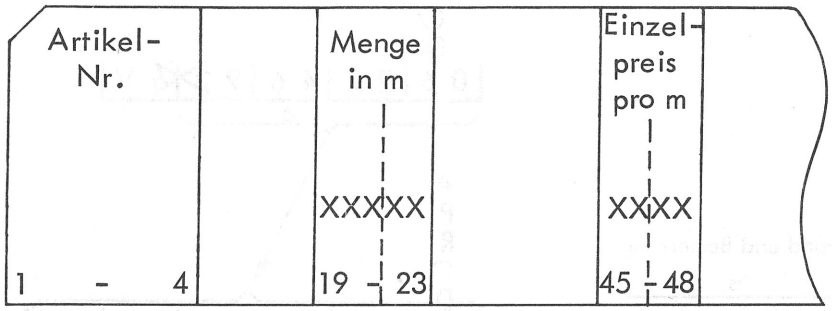


Die UNPK-Instruktion wird zwar vom DVS durchgeführt, jedoch können wir weder mit MVN, noch mit MVZ den Inhalt eines rechten Halbbytes in ein linkes Halbbyte bringen. Dies wäre aber im Anschluß an die UNPK-Instruktion nötig.

Nach diesen Vorüberlegungen wenden wir uns jetzt der nächsten Aufgabe zu.

Aufgabe 7

Eingabe: Artikelkarten



Ausgabe: Liste

Artikel-Nr.	Menge in m	Einzelpreis pro m	Betrag
0041	00450	1275	00057138
0062	00825	0630	00051198
0080	01400	0595	00083130
0085	18000	6015	10827000
0096	00375	2950	00110163
0104	08980	0680	00610464
0110	23000	0275	00632500
0118	04400	3385	01489400
0121	01650	0420	00069300
0134	00760	0573	00043555
13-16	19-23	26-29	32 - 38

Verarbeitung: Betrag = Menge x Einzelpreis.

Der Betrag ist zu runden.

Abb. 1

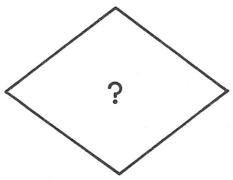
Aufgabe 7

Bitte sehen Sie sich die Aufgabe 7 (Abb. 1) genau an. -

Entnehmen Sie Ihren Arbeitsmitteln

a) die Aufgabe 7

b) ein Entwurfsblatt für Programmablaufpläne.



A Erstellen des Programmablaufplans ohne Hilfe

B Erstellen des Programmablaufplans mit Hilfe

* * * * *

A → (1), Seite 215

Nachstehende Aufstellung zeigt, was wir in unserem Programmablaufplan brauchen. - Die Reihenfolge der Punkte ist jedoch durcheinandergeraten.

Ordnen Sie bitte die Punkte zu einer logischen Reihenfolge.

Drucken

Runden Betrag

Prüfen auf Ende

Lesen Karte

Entpacken Betrag nach Druckbereich

Verzweigen zum Lesen der nächsten Karte

Packen Menge und Einzelpreis

Ende der Arbeit

Übertragen Artikel-Nr., Menge und Einzelpreis nach Druckbereich

Errechnen Betrag

* * * * *

a) Lesen Karte

b) Prüfen auf Ende

c) Übertragen Artikel-Nr., Menge und Einzelpreis nach Druckbereich *)

d) Packen Menge und Einzelpreis

e) Errechnen Betrag

f) Runden Betrag

g) Entpacken Betrag nach Druckbereich

h) Drucken

i) Verzweigen zum Lesen der nächsten Karte

k) Ende der Arbeit

*) Diese Übertragungen könnten genausogut erst vor dem Drucken erfolgen.

Jeder der Punkte a - k entspricht einem Block im Programmablaufplan.

(1) Erstellen Sie bitte den Programmablaufplan.

* * * * *

Überprüfen Sie bitte Ihren Programmablaufplan anhand der Abb. 2 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 7 Nr.: _____ Seite: _____

Programmierer: _____ Datum: _____

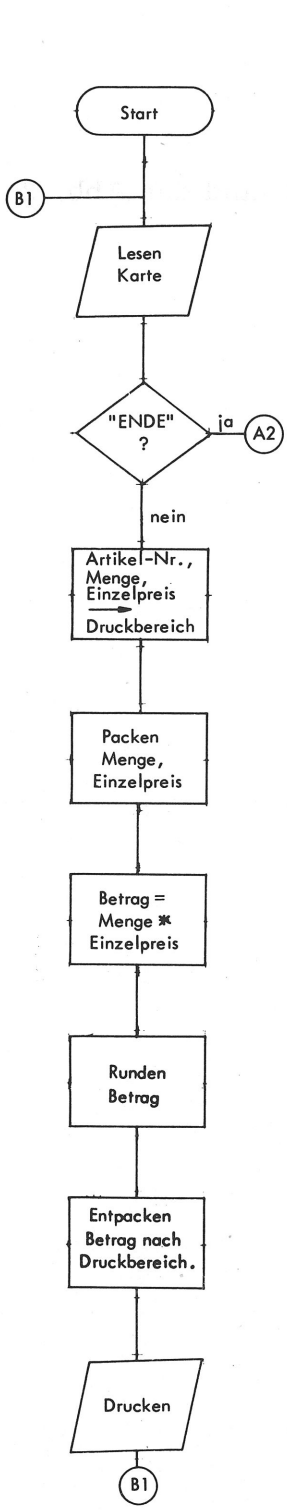


Abb. 2

Entnehmen Sie Ihren Arbeitsmitteln den Programmablaufplan zur Aufgabe 7.

Bevor wir die Speicherbelegung zeichnen, stellen wir einige Vorüberlegungen an.

Betrachten Sie bitte die Aufgabenstellung. -

Für die gepackte Menge benötigen wir ___(Z)___ Bytes.

Der gepackte Einzelpreis erfordert ___(Z)___ Bytes.

* * * * *

3

3

Das Rechenfeld für das Produkt muß also ___(Z)___ Bytes groß sein.

* * * * *

6

Das Produkt wird mit ___(Z)___ Dezimalstellen errechnet.

* * * * *

4

IBM System / 360 Speicherbelegung

Aufgabe 7

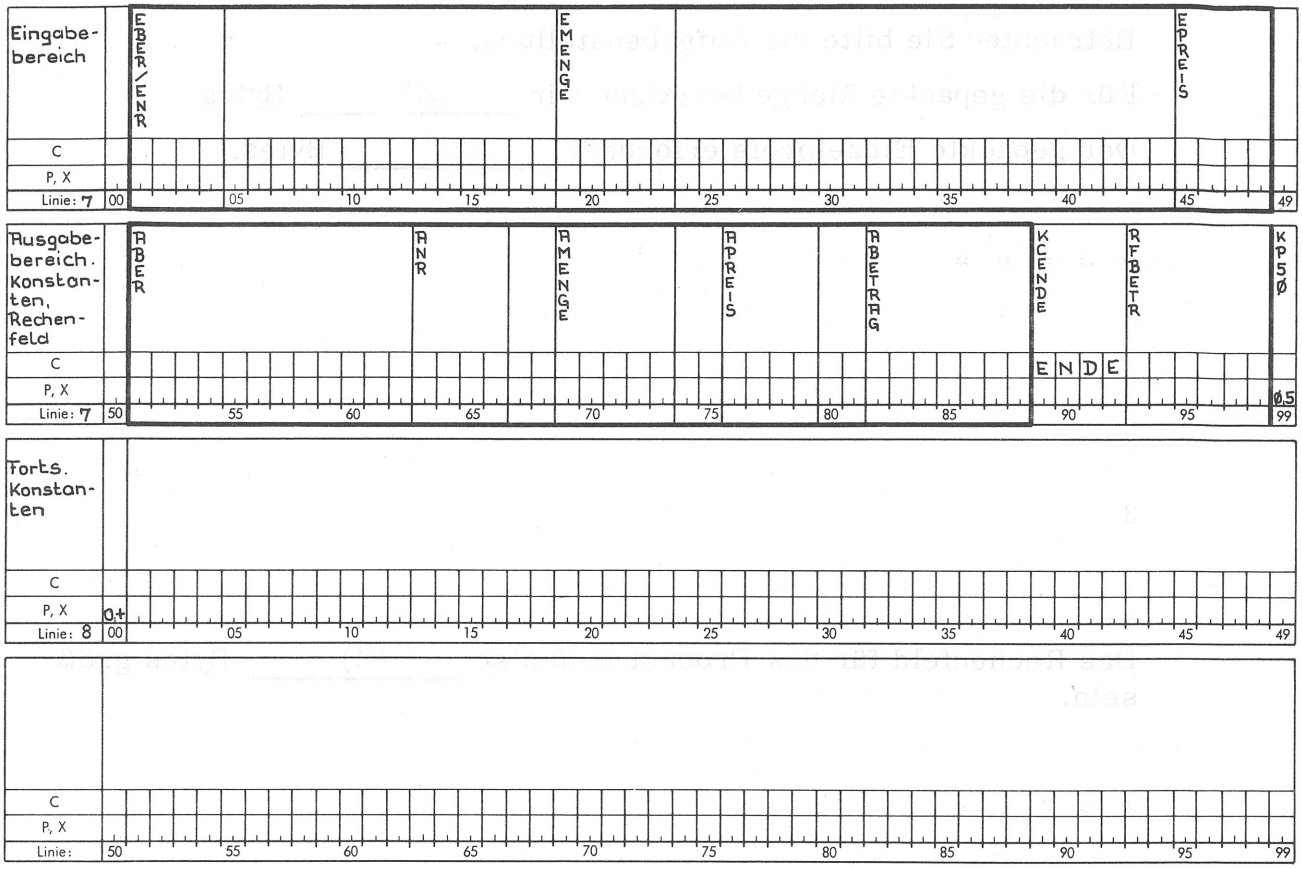


Abb. 3

Der Betrag ist zu runden und mit ____(Z)____ Dezimalstellen auszugeben.

Die Rundungskonstante muß folgenden Inhalt haben: _____.

* * * * *

2

0	5	0	+

Nun zur Speicherbelegung:

Zum Packen der Menge benutzen wir kein Extrafeld, da wir sie gleich in das Rechenfeld als Multiplikand einbringen. Den Preis packen wir im Eingabebereich (Packen ins gleiche Feld).

Abb. 3 zeigt die entsprechende Speicherbelegung.

Vergleichen Sie die Speicherbelegung mit der Aufgabenstellung.

Entnehmen Sie Ihren Arbeitsmitteln die Speicherbelegung zur Aufgabe 7.

IBM System / 360 Assembler
Programmschema

Programm Aufgabe 7		Lochhinweise		Problem A07 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 01 (76–77) von ____

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
* AUFGABE 7			010
* DEFINITIONEN			020
	START	701	030
EBER	DS	0CL48	040
ENR	DS	CL4	050
	DS	CL14	060
EMENGE	DS	CL5	070
	DS	CL21	080
EPREIS	DS	CL4	090
	DS	CL2	100
ABER	DS	0CL38	110
	DS	CL12	120
ANR	DS	CL4	130
	DS	CL2	140
AMENGE	DS	CL5	150
	DS	CL2	160
APREIS	DS	CL4	170
	DS	CL2	180
ABETRAG	DS	CL7	190
KCENDE	DC	C'ENDE'	200
RFBETR	DS	CL6	210
KP50	DC	P'+50'	220

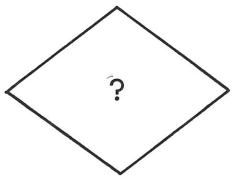
Abb. 4

Abb. 4 zeigt die Definitionen.

Vergleichen Sie bitte die Definitionen mit Ihrer Speicherbelegung. (Der Ausgabebereich ist so definiert, daß er unmittelbar an den Eingabebereich anschließt.)

Entnehmen Sie bitte Ihren Arbeitsmitteln die Definitionen zur Aufgabe 7.

Wir codieren nun den Programmablauf.



A Codieren des Programmablaufs ohne Hilfe

B Codieren des Programmablaufs mit Hilfe

* * * * *

A → (3), Seite 237

(2) Betrachten Sie bitte den Block E1 des Programmablaufplans.

Wie lautet die Instruktion zum Packen der Menge und welchen Inhalt hat RFBETR danach?

* * * * *

EMENGE

F X	F X	F X	F X	F X

RFBETR

0 0	0 0	0 0	X X	X X	X F

Operation	Operand und Bemerkungen
10 14	20 25 3
PACK	RFBETR,EMENGE

Der Preis wird ins gleiche Feld (EPREIS) gepackt. Welche Instruktion ist hierfür erforderlich und welchen Inhalt hat das Feld nach Ausführung der Instruktion?

* * * * *

EPREIS

F X	F X	F X	F X

0 0	0 X	X X	X F

Operation	Operand und Bemerkungen
10 14	20 25 30
PACK	EPREIS,EPREIS

oder:

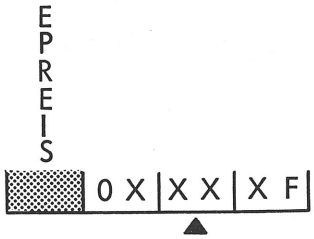
PACK EPREIS+1(3),EPREIS

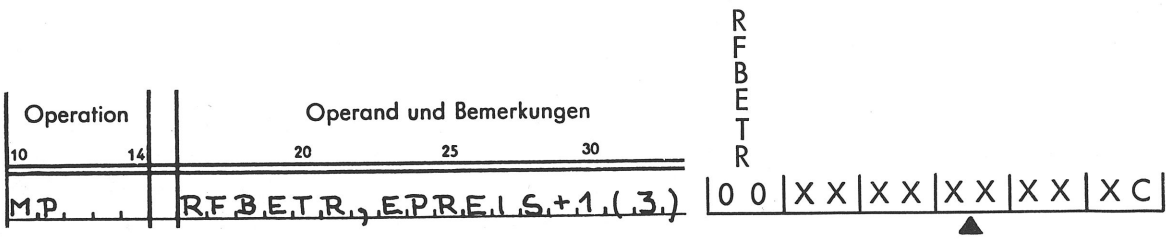
F X	0 X	X X	X F

Für die weitere Verarbeitung werden wir auf jeden Fall nur die 3 rechten Bytes von EPREIS verwenden.

Wie lautet die Codierung für den Block F1 und welches Ergebnis erhalten wir?

* * * * *





Das Vorzeichen des Produkts (muß, muß nicht)_____ auf die Rundungskonstante übertragen werden.

* * * * *

muß nicht (das Produkt ist immer positiv)

Codieren Sie jetzt die Rundung.

* * * * *

KP50

| 0 5 | 0 C |

Operation		Operand und Beme
10 14		20 25
A.P. . .		R.F.B.E.T.R.,.K.P.5.Ø.

RFBETR

| 0 0 | X X | X X | X X | X X | X C |

Wir nehmen uns nun den Block H1 vor. Bei der Ausgabe unseres Produktes sind zwei Dezimalstellen zu vernachlässigen.

RFBETR

| 0 0 | X X | X X | X X | X X | X C |

zu vernachlässigen

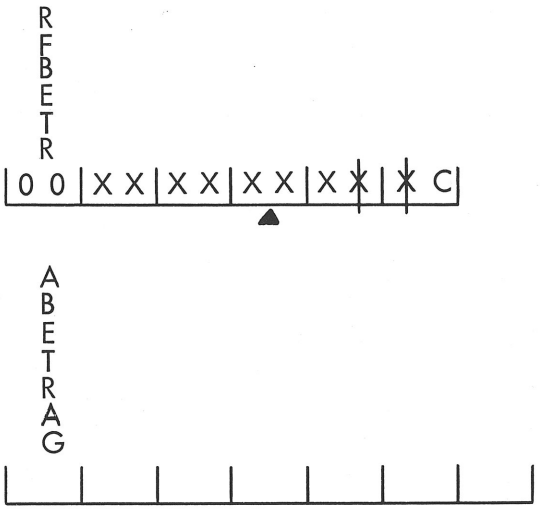
Das Feld ABETRAG ist (Z) Bytes groß. Das genügt auch, denn das linke Byte von RFBETR enthält immer (ZZ).

* * * * *

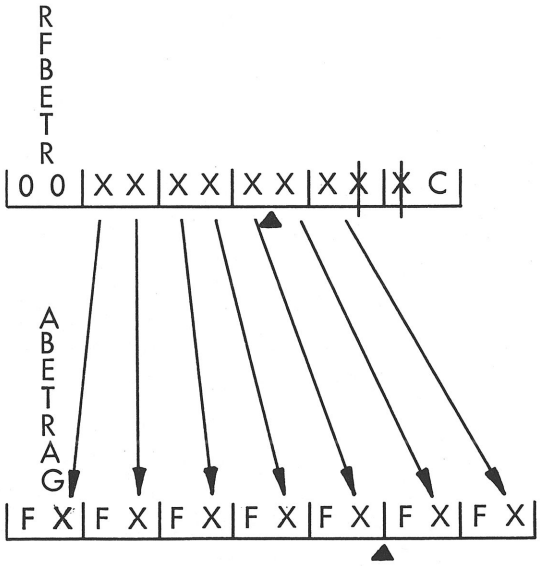
7

| 0 | 0 |

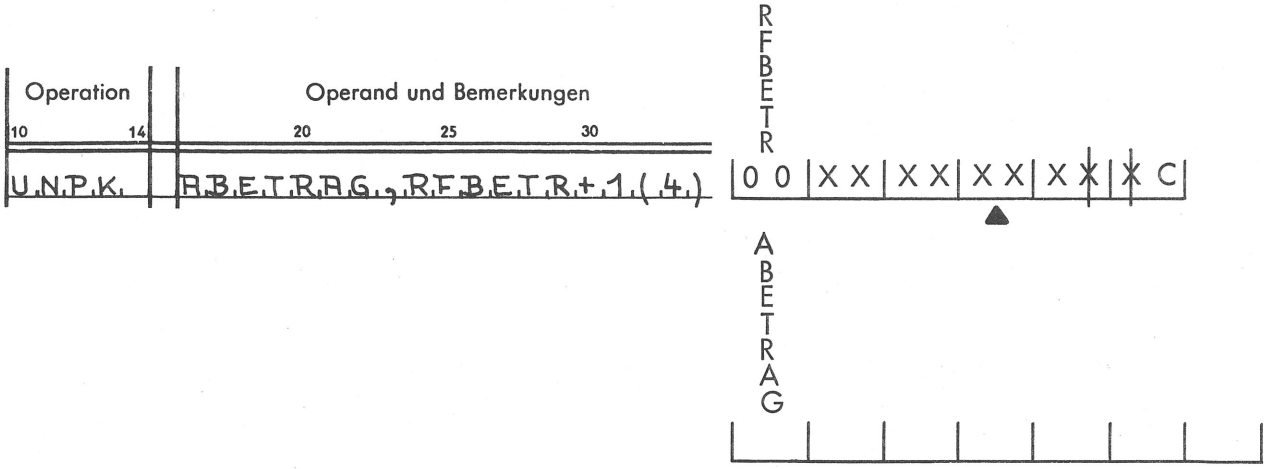
Zeichnen Sie bitte auf, wie der Betrag vor dem Drucken in ABETRAG stehen muß.



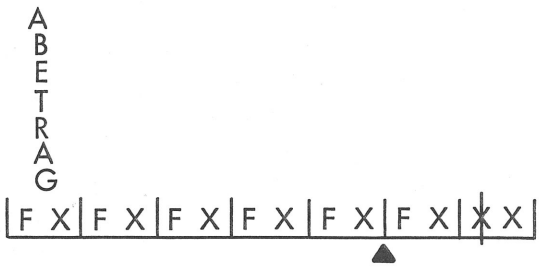
* * * * *



Welcher Inhalt ergibt sich in ABETRAG nach Ausführung folgender Instruktion:



* * * * *



Hierbei wird die vorletzte zu vernachlässigende Dezimalstelle mit entpackt.

Beispiel:

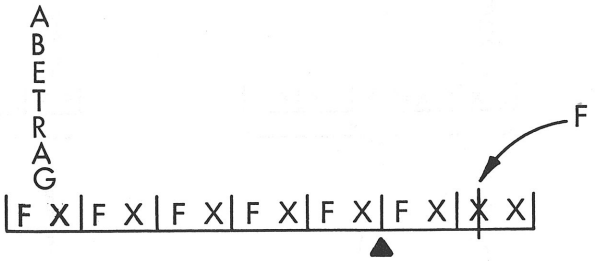
Das rechte Byte von ABETRAG enthält nach dem Entpacken | 0 | 9 | . Beim Drucken würde statt einer 9 gar nichts gedruckt, da diesem Inhalt kein druckbares Zeichen entspricht.

Wir müssen also vor dem Drucken in das linke Halbbyte des rechten Bytes von ABETRAG ein hexadezimales ___(Z)___ bringen.

* * * * *

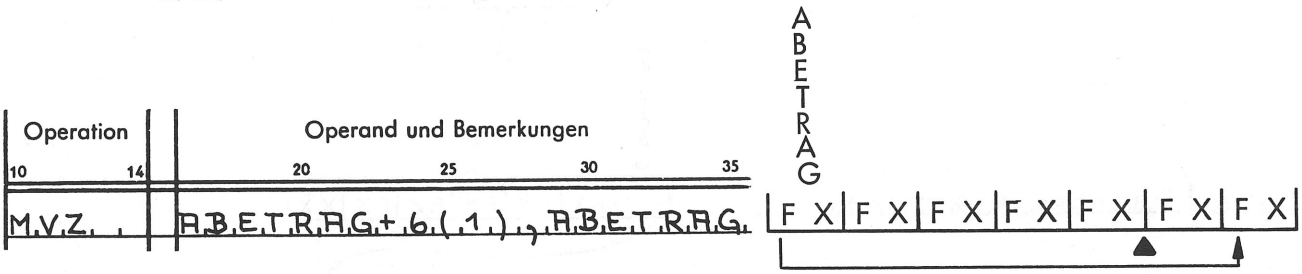
F

Codieren Sie nun die hierfür erforderliche Instruktion.



* * * * *

z. B.



Das gleiche erreichen wir, wenn wir v o r dem Entpacken ein hexadezimales F in das r e c h t e Halbbyte des 5. Bytes von RFBETR übertragen.

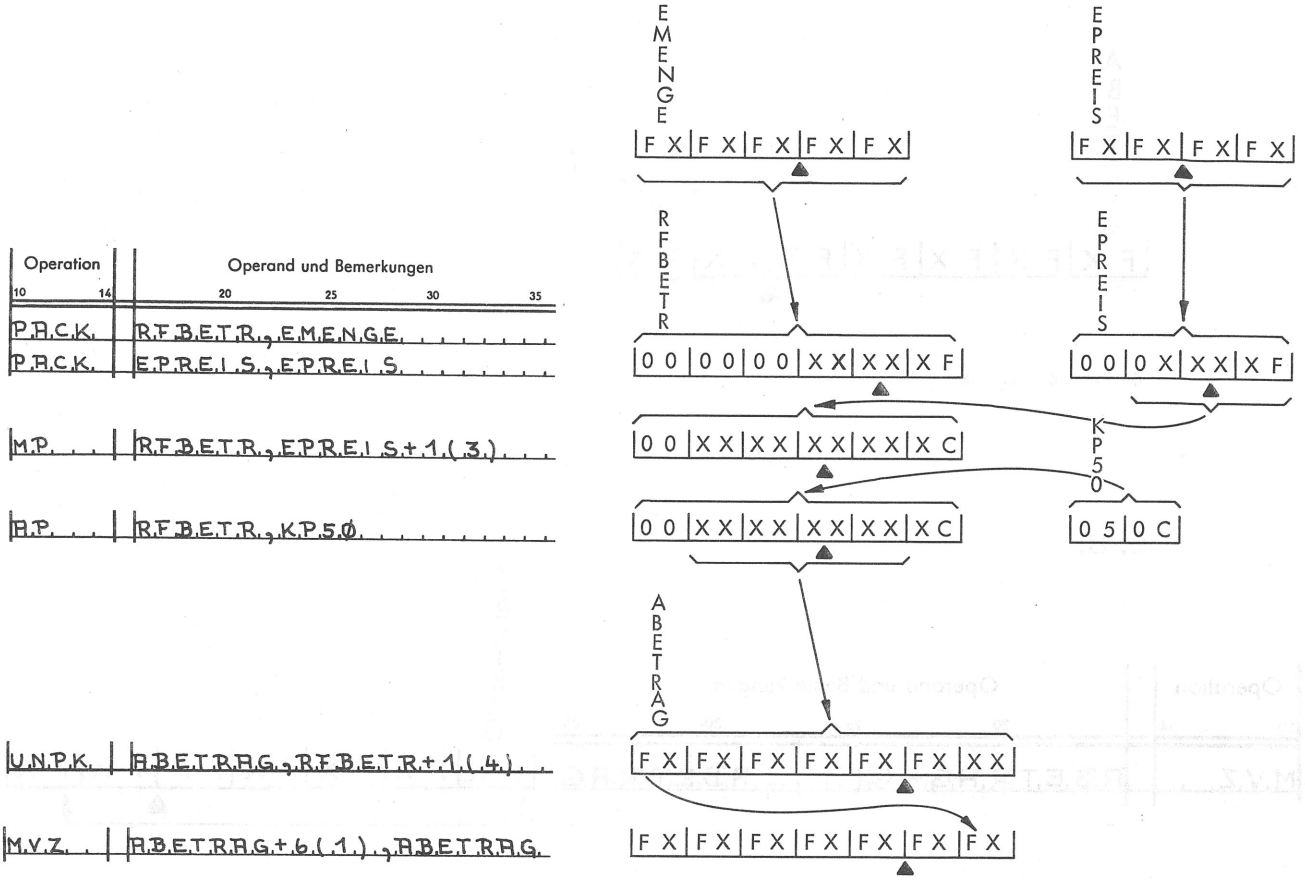

Operation
Operand und Bemerkungen
10
14
20
25
30
35
P.A.C.K. R.F.B.E.T.R.,E.M.E.N.G.E.
P.A.C.K. E.P.R.E.I.S.,E.P.R.E.I.S.
M.P. R.F.B.E.T.R.,E.P.R.E.I.S.+1.(3.)
A.P. R.F.B.E.T.R.,K.P.5Ø.
U.N.P.K. A.B.E.T.R.A.G.,R.F.B.E.T.R.+1.(4.)
M.V.Z. A.B.E.T.R.A.G.+6.(1.),A.B.E.T.R.A.G.
EMENGE
EPREIS
RFBETR
EPREIS
KP50
ABETRAG
F X F X F X F X F X
F X F X F X F X
0 0 0 0 0 0 X X X X F
0 0 X X X X X X X C
0 0 X X X X X X X C
0 5 0 C
F X F X F X F X F X F X X
F X F X F X F X F X F X F X

Abb. 5

Die Abb. 5 zeigt in einem Überblick die im Zusammenhang mit der Multiplikation erforderlichen Instruktionen und die davon betroffenen Felder.

Gehen Sie bitte nochmals Instruktion für Instruktion durch.

Zeichnen Sie bitte nun den rechten Teil der Abb. 5 - also die Felder - ab.

→ (4) , Seite 237

③

Hinweis:

Sie erleichtern sich die Codierung, wenn Sie den jeweiligen Inhalt der im Zusammenhang mit der Multiplikation wichtigen Felder aufzeichnen.

④ Codieren Sie bitte nun den Programmablauf.

* * * * *

Vergleichen Sie Ihre Codierung mit der der Abb. 6 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 7	Lochhinweise	Problem A07 (73–75)
		Zeichen	
Programmierer	Datum	Lochung	Seite 02 (76 77) von 2

	Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
	*.PROGRAM	MABLAUF.		010
	B1	GET	KARTE,EBER	020
		CLC	EBER(4),KCENDE oder ENR,KCENDE	030
		BE	A2	040
		MVC	ANR,ENR	050
		MVC	AMENGE,EMENGE	060
		MVC	APREIS,EPREIS	070
a)		PACK	RFBETR,EMENGE	080
b)		PACK	EPREIS,EPREIS	090
c)		MP	RFBETR,EPREIS+1(3)	100
d)		AP	RFBETR,KP50	110
e)		UNPK	ABETRAG,RFBETR+1(4)	120
f)		MVZ	ABETRAG+6(1),ABETRAG	130
		PUT	LIST,ABER	140
		B	B1	150
	A2	Ende der Arbeit		160
		END		170
				180
				190
				200

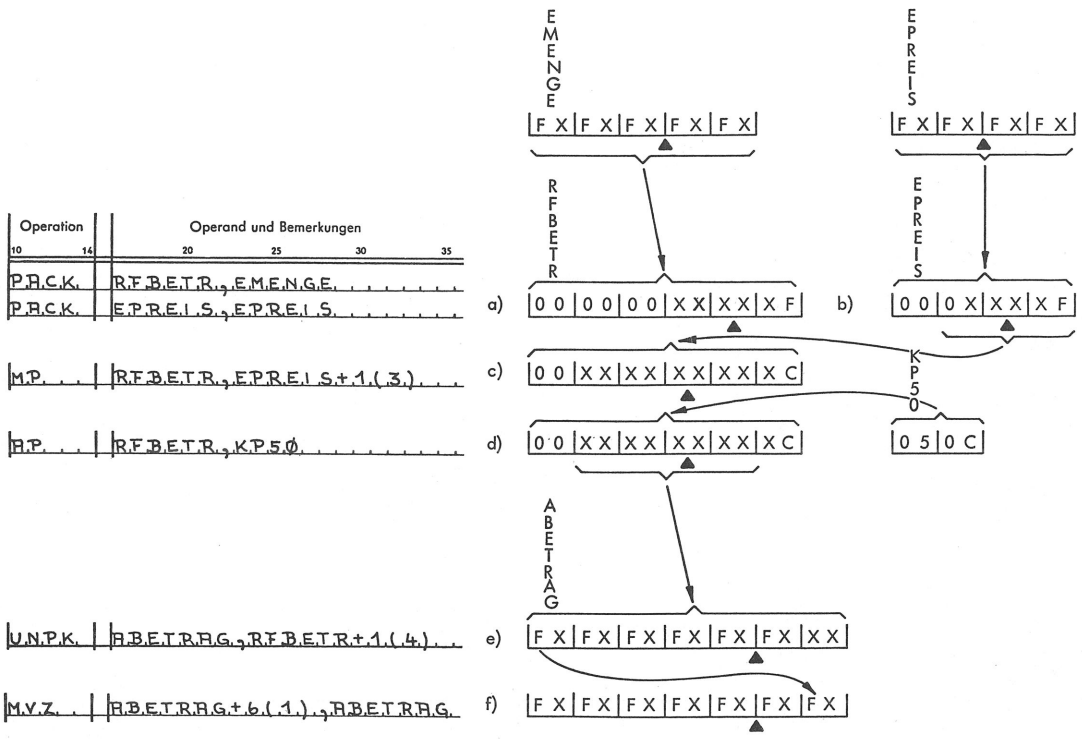


Abb. 6

Verbessern Sie Ihre Codierung soweit erforderlich.

Falls Sie die Hinweise zur Lösung übersprungen haben und mit der gezeigten Codierung nicht zurechtkommen (2) , Seite 223.

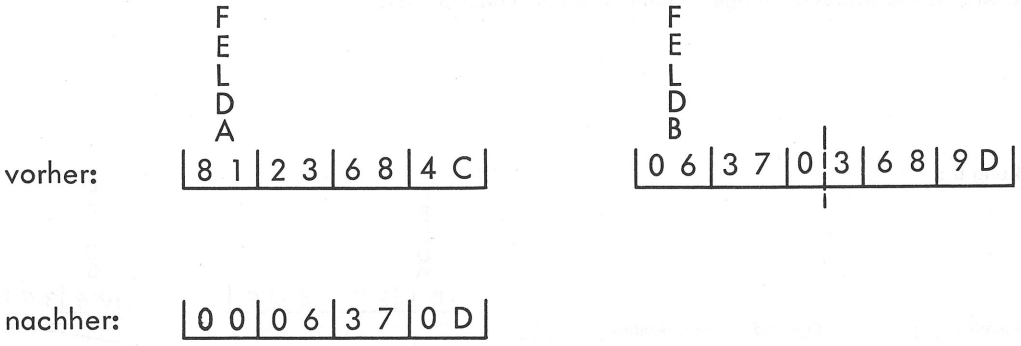
Die Aufgabe 7 ist damit abgeschlossen.

Wir lernen jetzt eine Instruktion kennen, die es ermöglicht, den Inhalt eines Feldes um ein Halbbyte zu verschieben. Welche Bedeutung dieser Instruktion zukommt, werden Sie in den folgenden Übungen und Aufgaben erfahren.

Band 4, Seite 300 (MVO)

Bevor wir mit der nächsten Aufgabe beginnen, werden wir in einigen Übungen die Operation MVO in dem Zusammenhang kennenlernen, in dem sie am häufigsten ihre Anwendung findet.

Gegeben sind:



Der Inhalt von Feld B ist unter Vernachlässigung der rechten 4 Ziffern, jedoch mit Vorzeichen in das Feld A zu bringen.

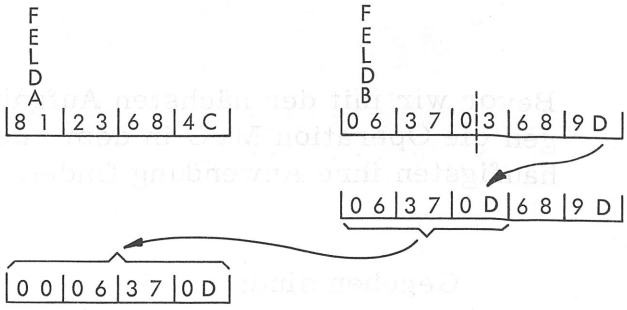
Welche Instruktionen sind hierfür erforderlich? (Zeichnen Sie sich den jeweiligen Inhalt von Feld A auf.)

* * * * *

Vergleichen Sie bitte Ihre Lösung mit denen der Abb. 7 (Rückseite!).

Lösung A :

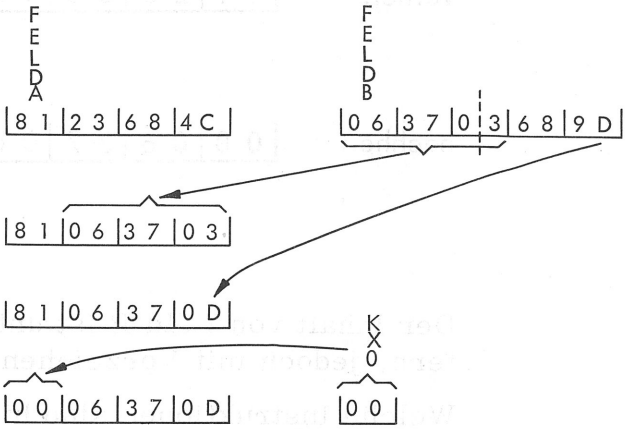
Operation	Operand und Bemerkungen
10 14	20 25 30
M.V.N.	F.E.L.D.B.+2.(.1.).,F.E.L.D.B.+4.
Z.A.P.	F.E.L.D.A.,F.E.L.D.B.(.3.). . . . .



Bei diesem Vorgehen wird der ursprüngliche Inhalt von Feld A verändert.
Die umgekehrte Instruktionsfolge ist nicht möglich (Vorzeichen!).

Lösung B :

Operation	Operand und Bemerkungen
10 14	20 25 30
M.V.C.	F.E.L.D.A.+1.(.3.).,F.E.L.D.B. . .
M.V.N.	F.E.L.D.A.+3.(.1.).,F.E.L.D.B.+4.
M.V.C.	F.E.L.D.A.(.1.).,K.X.Ø. . . . .



Die Konstante KX0 könnte genausogut zuerst übertragen werden.

Lösung C :

	Operation	Operand und Bemerkungen
	10 14	20 25 30
	S.P.	F.E.L.D.A.,F.E.L.D.A. . . . .
oder:	M.V.O.	F.E.L.D.A.,K.X.Ø. . . . .
oder:	Z.A.P.	F.E.L.D.A.,K.P.Ø. . . . .
	M.V.C.	F.E.L.D.A.+1.(.3.).,F.E.L.D.B. .
	M.V.N.	F.E.L.D.A.+3.(.1.).,F.E.L.D.B.+4.

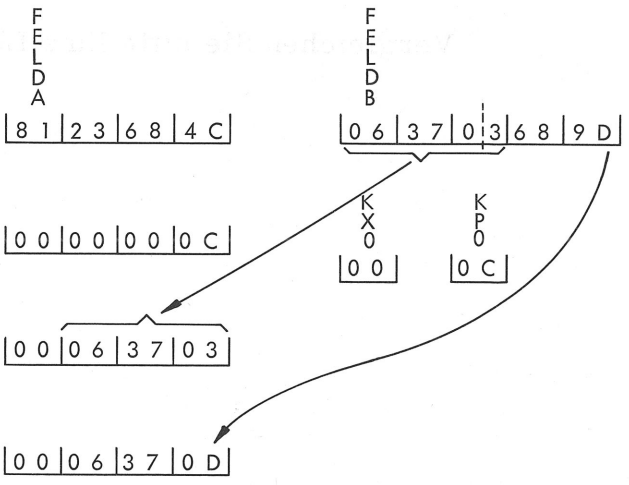


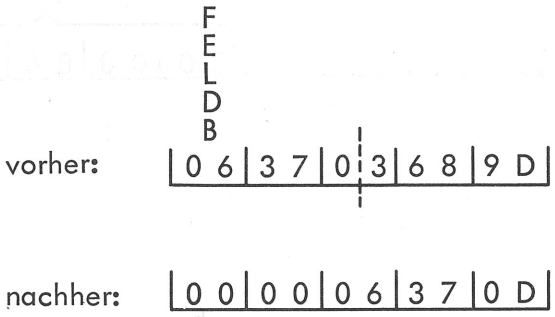
Abb. 7

Machen Sie sich bitte mit allen aufgezeigten Lösungen vertraut.

Die Lösung A erfordert nur zwei, alle anderen Lösungen drei Instruktionen.

Zu beachten ist dabei, daß bei der Lösung A der ursprüngliche Inhalt des Feldes B verändert wird. Wenn dieser Inhalt zur weiteren Verarbeitung nochmals benötigt wird, muß man eine der anderen Lösungen nehmen.

Wir wandeln die vorherige Übung ab und nehmen eine Stellenversetzung im gleichen Feld vor.

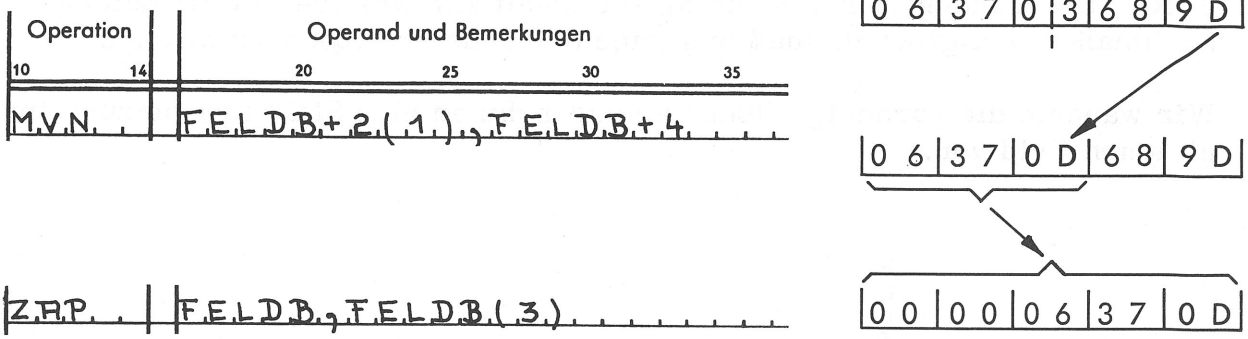


Welche Instruktionen benötigen wir hierfür?

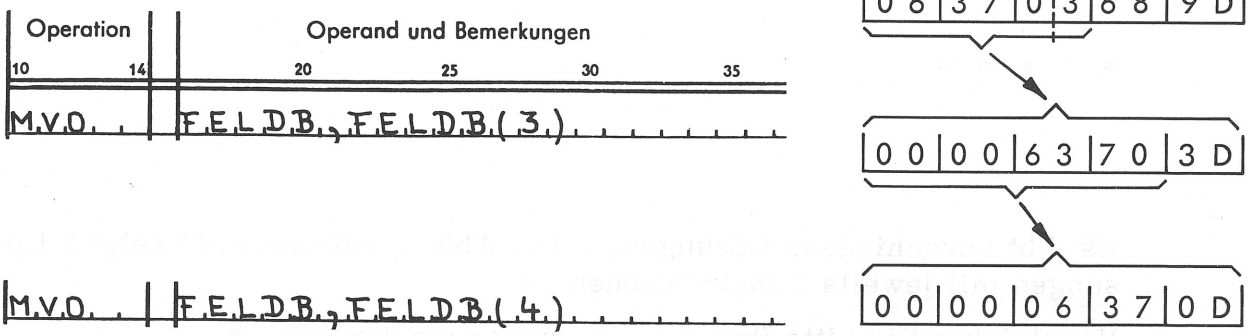
* * * * *

Es gibt verschiedene Lösungen. - Die Abb. 8 (Rückseite!) zeigt 2 Lösungen mit jeweils 2 Instruktionen. -
Vergleichen Sie bitte Ihre Lösung mit denen der Abb. 8.

Lösung A:



Lösung B:



Die zweite MVO-Instruktion könnte auch lauten:

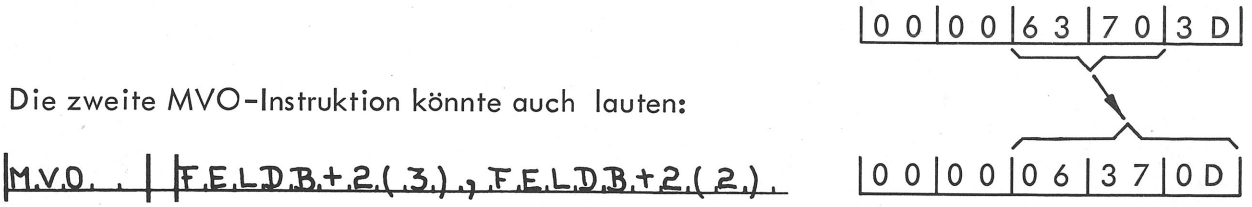
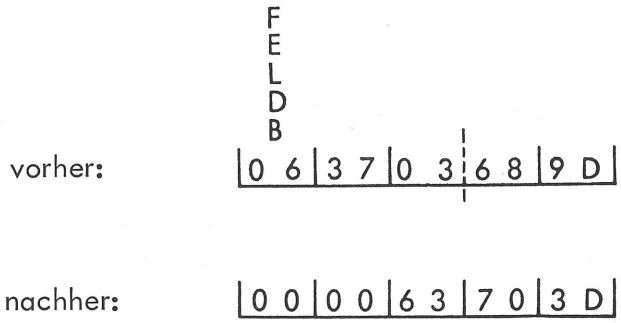


Abb. 8

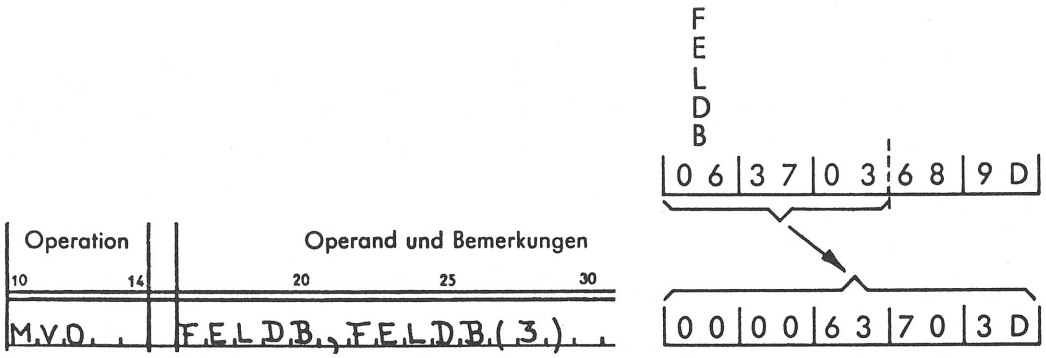
Sie müssen beide in Abb. 8 gezeigten Lösungen verstehen.

Der Inhalt von FELDB ist, wie nachstehend gezeigt, zu versetzen.

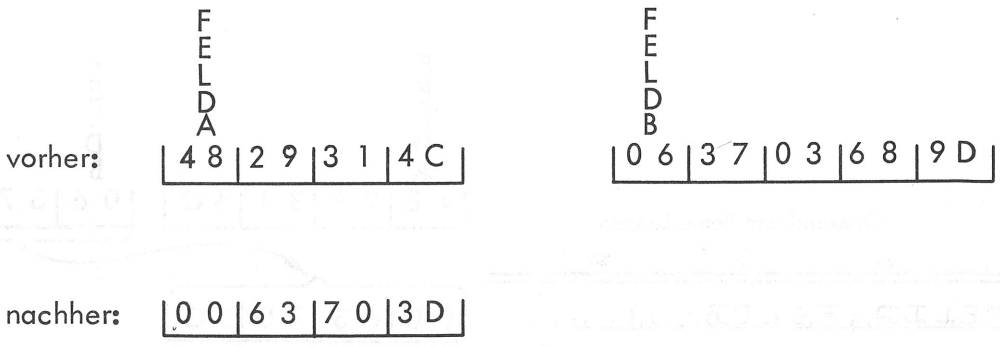


Wie lautet (lauten) die hierfür erforderliche(n) Instruktion(en)?

* * * * *



Gegeben sind:



Welche Instruktionen benötigen wir, wenn der Inhalt von FELDB unter Vernachlässigung der rechten drei Ziffern nach FELDA gelangen soll?

* * * * *

Vergleichen Sie bitte Ihre Lösung mit der der Abb. 9 (Rückseite!).

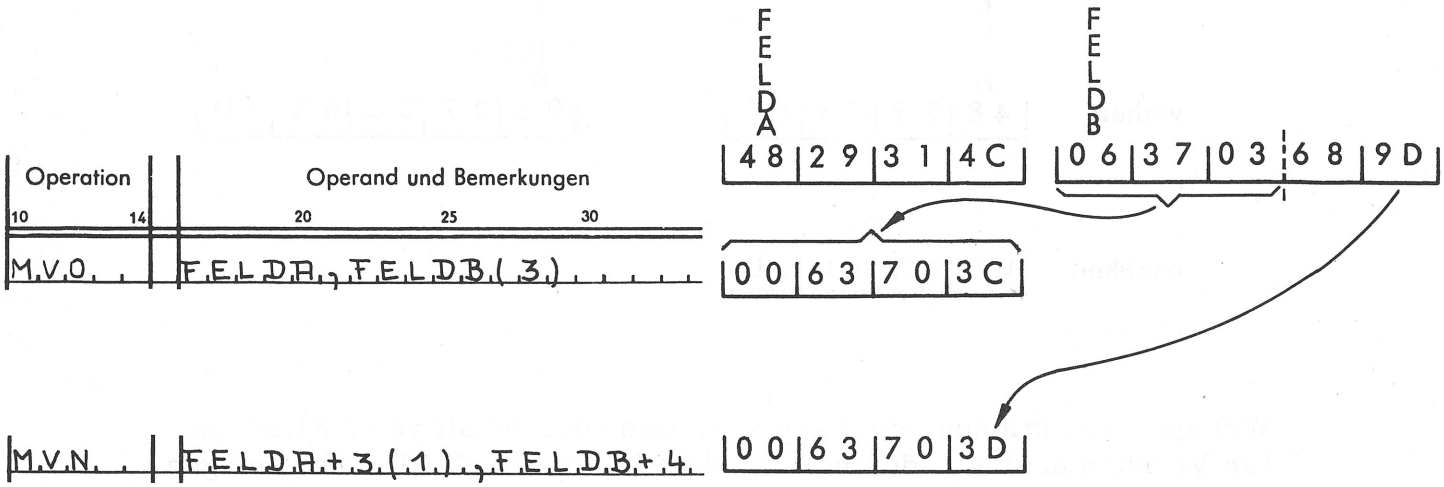

FELDA
FELDB
Operation
Operand und Bemerkungen
10
14
20
25
30
M.V.O. .
F.E.L.D.A.,F.E.L.D.B.(3.) . . . . .
M.V.N. .
F.E.L.D.A.+3.(1.),F.E.L.D.B.+4.
4 8 2 9 3 1 4 C
0 6 3 7 0 3 6 8 9 D
0 0 6 3 7 0 3 C
0 0 6 3 7 0 3 D

Abb. 9

Anmerkung zur Abb. 9:

Die umgekehrte Instruktionsfolge (zuerst MVN, dann MVO) ist natürlich genauso richtig.

A

FELDA vorher:

0 2	3 4	6 7	2 D

FELDA nachher:

0 0	0 2	3 4	6 D

B

FELDA vorher:

9 2	7 4	8 D

FELDB

0 8	2 7	6 3	1 7	8 4	2 C

nachher:

0 8	2 7	6 C

C

FELDB vorher:

0 2	3 4	6 7	2 1	6 C

FELDB nachher:

0 0	0 0	0 0	2 3	4 C

D

FELDA vorher:

8 7	4 3	2 D

FELDB

0 2	3 4	6 7	2 1	6 C

FELDA nachher:

2 3	4 6	7 C

E

FELDB vorher:

0 0	2 7	6 8	4 3	1 9	6 C

FELDB nachher:

0 0	0 2	7 6	8 4	3 1	9 C

F

FELDA vorher:

7 6	4 2	0 0	1 D

FELDB

0 0	6 4	9 3	2 1	8 C

FELDA nachher:

0 0	0 0	6 4	9 C

Abb. 10

Wir ziehen aus den vorausgegangenen Übungen das Fazit:

Es gibt eine einfache Regel, nach der wir ohne lange Überlegungen sofort sagen können, ob für eine Stellenversetzung die Operation MVO erforderlich ist oder nicht:

Ist um eine u n g e r a d e Anzahl Stellen zu versetzen (d. h. sind von den rechten Stellen 1, 3, 5, 7 ... usw. Stellen zu vernachlässigen), brauchen wir immer die Operation MVO.

Entscheiden Sie bitte nach dieser Regel, in welchen Fällen der Abb. 10 die Operation MVO unbedingt erforderlich ist.

Notieren Sie sich nur die entsprechenden Buchstaben (A-F) und schreiben Sie bitte nur in diesen Fällen die erforderlichen Instruktionen daneben (auf ein Extrablatt).

* * * * *

Überprüfen Sie Ihre Antworten anhand der Abb. 11 (Rückseite!).

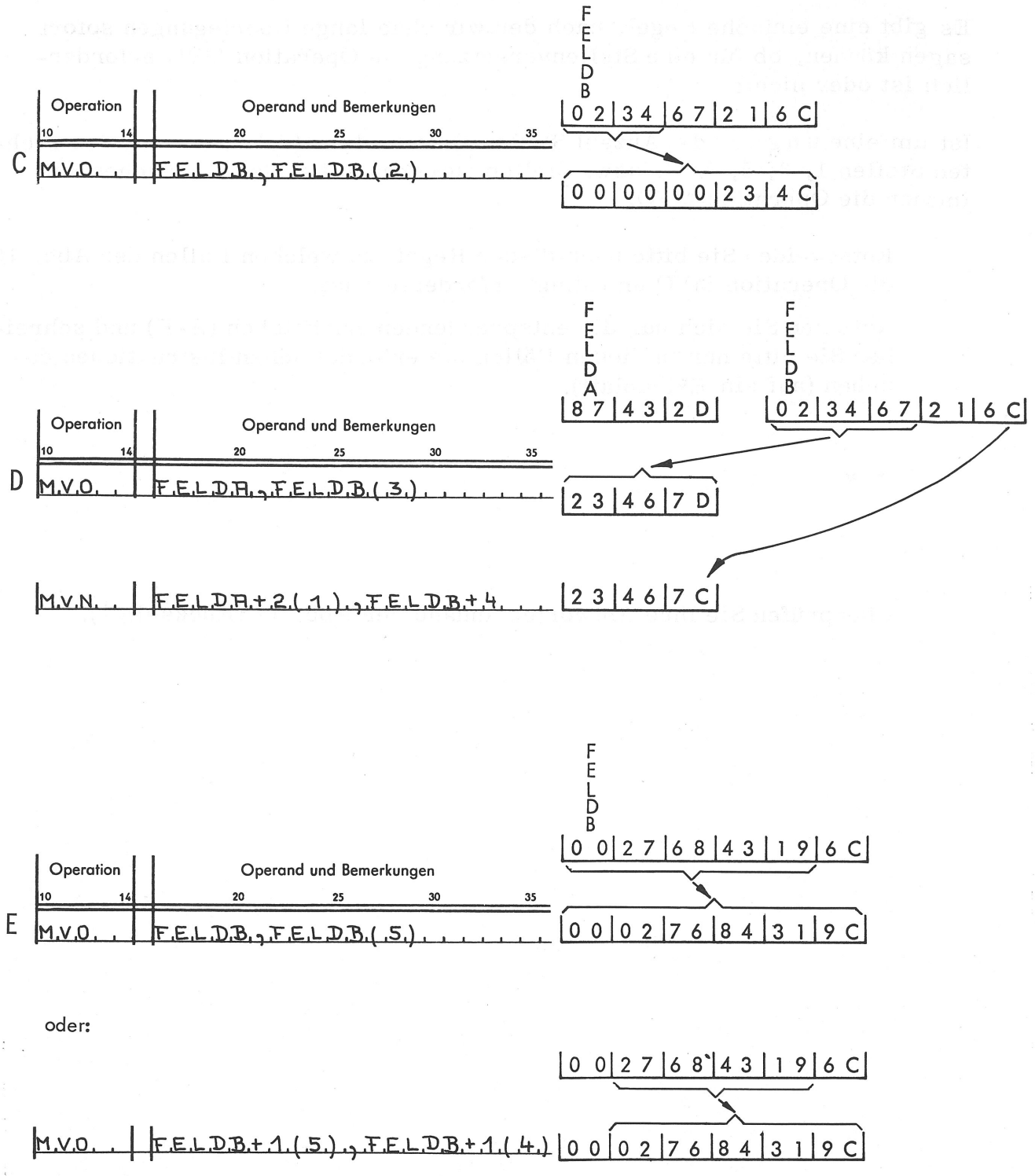

C
Operation
Operand und Bemerkungen
10
14
20
25
30
35
M.V.O. .
F.E.L.D.B.,F.E.L.D.B.(.2.) . . . . . .
FELDB
0 2 3 4 6 7 2 1 6 C
0 0 0 0 0 0 2 3 4 C
D
Operation
Operand und Bemerkungen
M.V.O. .
F.E.L.D.A.,F.E.L.D.B.(.3.) . . . . . .
FELDA
8 7 4 3 2 D
FELDB
0 2 3 4 6 7 2 1 6 C
2 3 4 6 7 D
M.V.N. .
F.E.L.D.A.+2(.1.),F.E.L.D.B.+4. . .
2 3 4 6 7 C
E
Operation
Operand und Bemerkungen
M.V.O. .
F.E.L.D.B.,F.E.L.D.B.(.5.) . . . . . .
FELDB
0 0 2 7 6 8 4 3 1 9 6 C
0 0 0 2 7 6 8 4 3 1 9 C
oder:
0 0 2 7 6 8 4 3 1 9 6 C
M.V.O. .
F.E.L.D.B.+1.(.5.),F.E.L.D.B.+1.(.4.)
0 0 0 2 7 6 8 4 3 1 9 C

Abb. 11

Wir wenden uns jetzt der Aufgabe 8 zu.

Bitte blättern Sie um.

Aufgabe 8

Eingabe: Artikelkarten

Artikel-Nr.	Menge in kg	Einzelpreis pro kg	
	XXX | XXX	X | XX	
3 - 6	7 - 12	13 - 15	

Ausgabe: Liste

Artikel-Nr.	Menge in kg	Einzelpreis pro kg	Betrag
0014	413|120	1|37	0565|97
0081	025|050	3|50	0087|68
0105	081|610	2|63	0214|63
0123	050|000	0|70	0035|00
0128	121|350	0|50	0060|68
0150	512|200	2|84	1454|65
0183	060|100	4|80	0288|48
0224	035|216	3|93	0138|40
0251	002|134	5|20	0011|10
0306	235|000	6|60	1551|00
8 - 11	14 - 19	22-24	27 - 32

Verarbeitung: Betrag = Menge x Einzelpreis

Der Betrag ist zu runden.

Abb. 1

Aufgabe 8

Sehen Sie sich bitte die Aufgabe 8 (Abb. 1) an. -

Diese Aufgabe unterscheidet sich von der Aufgabe 7 nur in der Codierung der im Zusammenhang mit der Multiplikation wichtigen Instruktionen.

Vorgegeben sind deshalb Programmablaufplan, Speicherbelegung und Definitionen.

Bitte entnehmen Sie Ihren Arbeitsmitteln zur Aufgabe 8

a) die Aufgabe

b) den Programmablaufplan

c) die Speicherbelegung

d) die Definitionen.

Wir müssen nun die Speicherbelegung und die Definitionen noch um das Rechenfeld und die benötigte Rundungskonstante erweitern.

Betrachten Sie bitte die Aufgabenstellung.

Für die gepackte Menge benötigen wir ____(Z)____ Bytes. Für den gepackten Preis benötigen wir ____(Z)____ Bytes. Das Produktfeld muß also ____(Z)____ Bytes groß gewählt werden.

* * * * *

4

2

6

Das Produkt (der Betrag) wird mit ____(Z)____ Dezimalstellen errechnet.

Der Betrag ist zu runden und mit zwei Dezimalstellen auszugeben.

Die Rundungskonstante muß lauten: _____.

* * * * *

5 (3+2)

5 0	0 +

Ergänzen Sie nun entsprechend die Speicherbelegung und die Definitionen. Benutzen Sie die symbolischen Namen RFBETR und KP500.

* * * * *

Vergleichen Sie bitte Ihre hinzugefügten Eintragungen mit Abb. 2 (Rückseite!).

IBM System / 360 Speicherbelegung

Aufgabe 8

Eingabe-, Ausgabe-bereich, Konstante		EBER	ENR	EMENGE	EPREIS	ABER	ANR	AMENGE	APREIS	ABETRAG	KCENDE
C											E N
P, X											
Linie: 7	00	05	10	15	20	25	30	35	40	45	49

Konst. u. Rechenf. (Forts.)		RFBETR	KP500
C	D E		
P, X			500+
Linie: 7	50	55	60 65 70 75 80 85 90 95 99

C	
P, X	
Linie:	00 05 10 15 20 25 30 35 40 45 49

C	
P, X	
Linie:	50 55 60 65 70 75 80 85 90 95 99

IBM System / 360 Assembler Programmschema

Programm Aufgabe 8		Lochhinweise	Problem A Ø 8 (73 75)
		Zeichen	
Programmierer	Datum	Lochung	Seite Ø 1 (76 77) von 2

Anweisung: Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
* AUFGABE	8		Ø1Ø
* DEFINITIONEN			Ø2Ø
	START	7Ø1	Ø3Ø
EBER	DS	ØCL15	Ø4Ø
	DS	CL2	Ø5Ø
ENR	DS	CL4	Ø6Ø
EMENGE	DS	CL6	Ø7Ø
EPREIS	DS	CL3	Ø8Ø
ABER	DS	ØCL32	Ø9Ø
	DS	CL7	1ØØ
ANR	DS	CL4	11Ø
	DS	CL2	12Ø
AMENGE	DS	CL6	13Ø
	DS	CL2	14Ø
APREIS	DS	CL3	15Ø
	DS	CL2	16Ø
ABETRAG	DS	CL6	17Ø
KCENDE	DC	C'ENDE'	18Ø
RFBETR	DS	CL6	19Ø
KP5ØØ	DC	P'+5ØØ'	2ØØ

Abb. 2

Codieren Sie nun den Programmablauf.

Beachten Sie bitte:
Zeichnen Sie sich den jeweiligen Inhalt der im Zusammenhang mit der Multiplikation wichtigen Felder auf.

* * * * *

Überprüfen Sie bitte Ihre Codierung anhand der Abb. 4 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 8	Lochhinweise		Problem	A08
		Zeichen			
Programmierer	Datum	Lochung		Seite	02 von 2

Name	Operation	Operand und Bemerkungen	Karte Nr.
*.PROGRAMMABLAUF			010
B1	GET	KARTE,EBER	020
	CLC	EBER(4),KCENDE	030
	BE	A2	040
	MVC	ANR,ENR	050
	MVC	AMENGE,EMENGE	060
	MVC	APREIS,EPREIS	070
	PACK	RFBETR,EMENGE	080
	PACK	EPREIS,EPREIS oder EPREIS+1(2),EPREIS	090
	MP	RFBETR,EPREIS+1(2)	100
	AP	RFBETR,KP5ØØ	110
	MVO	RFBETR,RFBETR(4) oder RFBETR+1(5),RFBETR+1(3)	120
	UNPK	ABETRAG,RFBETR	130
	MVZ	ABETRAG+5(1),ABETRAG	140
	PUT	LISTE,ABER	150
	B	B1	160
A2	Ende der	Arbeit	170
	END		180
			190
			200

Abb. 3

Stimmt Ihre Codierung mit der vorgegebenen überein → (1),
Seite 263.

Sehen Sie sich die Abb. 4 (Rückseite!) an. -

Operation	Operand und Bemerkungen
10 14	20 25 30 35
PACK	RFBETR,EMENGE
PACK	EPREIS,EPREIS
MP	RFBETR,EPREIS+1(2)
AP	RFBETR,KP500
MVO	RFBETR,RFBETR(4)
UNPK	ABETRAG,RFBETR
MVZ	ABETRAG+5(1),ABETRAG

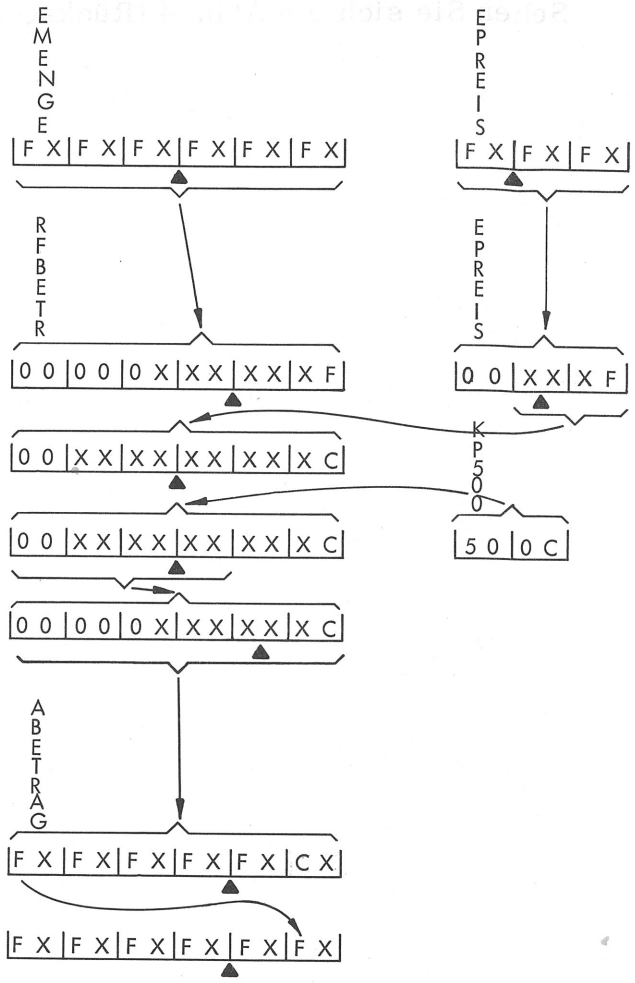


Abb. 4

Die Abb. 5 zeigt die für diese Aufgabe wesentlichen Instruktionen und deren Auswirkungen.

Gehen Sie diese bitte Schritt für Schritt durch.

Zeichnen Sie die Abb. 4 ab, sofern Sie es für nötig halten.

Verbessern Sie nun Ihre Codierung.

①

Zum Schluß der Aufgabe sei noch auf einen allgemeinen Sachverhalt hingewiesen. - Nehmen Sie dazu bitte nochmals Ihren Programmablaufplan zur Hand.

Vergleichen Sie den Block H1 mit den dafür codierten Instruktionen.

Hinter der Eintragung "Entpacken Betrag nach Druckbereich" verbirgt sich außer dem Entpacken noch eine Stellenversetzung (mit MVO) und ein Ersetzen der Vorzeichenverschlüsselung "C" durch "F" (mit MVZ).

Um beim Erstellen des Programmablaufplans nicht schon alle Einzelheiten überlegen zu müssen, bleiben wir auch künftig bei der "groben" Angabe wie im Block H1. Erst bei der Codierung werden wir uns um die maschinengerechte Durchführung der Angaben kümmern, und dabei ergibt sich von selbst, ob eine Auffächerung einer "groben" Angabe in mehrere "feinere" Angaben nötig ist.

Die Aufgabe 8 ist damit abgeschlossen.

Entnehmen Sie bitte Ihren Arbeitsmitteln das Merkblatt 101.

Lesen Sie das Merkblatt aufmerksam durch.

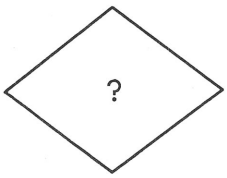
Wir wenden uns jetzt der Division zu.

Band 4, Seite 278 (DP)

Sie kennen nun den Ablauf der Division im System /360 und die zugehörige Instruktion. Auch haben Sie sich mit der von Ihrem Schulwissen her bekannten Terminologie wieder vertraut gemacht.

Nicht berührt und deswegen nachzutragen sind Überlegungen zu folgender Frage: Wie erhält man eine gewünschte Genauigkeit des Quotienten, d.h. wie erreicht man, daß der Quotient eine ganz bestimmte Anzahl von Dezimalstellen (Stellen nach dem Komma) aufweist?

Zur Beantwortung dieser Frage wird kurz auf die Division zweier Zahlen mit Dezimalstellen eingegangen.



A Ich bin sicher, daß ich diese Dinge noch beherrsche.

B Ich möchte mein Wissen hierzu wieder auffrischen.

* * * * *

A → (1), Seite 281

Beseitigung der Dezimalstellen des Divisors durch Erweitern

Hat der Divisor Dezimalstellen, so multipliziert man vor der Division Dividend und Divisor mit einer solchen Zehnerpotenz (10, 100, 1 000 usw.), daß der Divisor eine ganze Zahl wird. Man nennt diesen Rechenvorgang "Erweitern".

Man erweitert, indem man beim Dividenden und beim Divisor das Komma jeweils um so viele Stellen nach rechts verschiebt, wie der Divisor Dezimalstellen aufweist. Hat der Dividend vor dem Erweitern weniger Dezimalstellen als der Divisor, so müssen ihm Nullen angehängt werden.

Führen Sie die Erweiterung in den folgenden Aufgaben durch, so daß der Divisor eine ganze Zahl wird.

a) 96,563 : 1,7 ; _____(Z)_____ : _____(Z)_____

b) 71 : 0,31 ; _____(Z)_____ : _____(Z)_____

c) 35,4 : 2,345 ; _____(Z)_____ : _____(Z)_____

* * * * *

a) 965,63 : 17

b) 7100 : 31

c) 35400 : 2345

Die Divisionen könnten nun ausgeführt werden. Beim Überschreiten des Kommas im Dividenden wird auch im Quotienten das Komma gesetzt.

Beachten Sie:

Die Stellung des Kommas im Quotienten muß vom Programmierer bestimmt und festgehalten werden. (Für das System /360 sind beim Rechnen mit gepackten Zahlen überhaupt keine Kommas vorhanden!)

Genauigkeit des Quotienten

Berechnen wir z.B. 8,2 : 3 handschriftlich, so werden wir gewöhnlich so lange dividieren, bis der sich ergebende Rest kleiner als der Divisor ist:

```
8,2 : 3 = 2,7
 6
 22
 21
  1 ... Rest, kleiner als Divisor 3
```

Wünschen wir eine größere Genauigkeit, d.h. weitere Dezimalstellen des Quotienten, so werden wir dem Dividenden Nullen "anhängen".

Im obigen Beispiel ergeben sich folgende Divisionen beim Anhängen

einer Null:

```
8,20 : 3 = 2,73
 6
 22
 21
  10
   9
   1
```

zweier Nullen:

```
8,200 : 3 = 2,733
 6
 22
 21
  10
   9
   10
    9
    1
```

Anmerkung:

Der vereinfachten Darstellung wegen werden üblicherweise die Nullen dem Dividenden nicht angehängt, sondern gleich den jeweiligen Resten hinzugefügt.

Aus den aufgeführten Beispielen sehen Sie, daß allgemein gilt:

Die Anzahl der Dezimalstellen von Dividend und Quotient stimmen überein.

Diese Feststellung ist insofern für die Division im System /360 wichtig, als unter der Beachtung der Längenregel dem Quotienten genau dieselbe Stellenzahl wie dem Dividenden zur Verfügung steht und demzufolge die Division genau so wie in den gezeigten Beispielen verläuft. - Der Quotient wird mit derselben Anzahl von Dezimalstellen errechnet, wie der Dividend beim Einbringen ins Rechenfeld aufweist.

Damit also bei der Division 8,2 : 3 der Quotient mit 4 Dezimalstellen errechnet wird, muß man den Dividenden in das Rechenfeld für die Division einbringen in der Form ___(Z)___.

* * * * *

8▲2000 (▲ (gedachtes) Komma)

Daher müssen wir dem Dividenden ___(Z)___ Nullen anhängen.

* * * * *

3

Das Runden

Will man als Ergebnis einer Division einen gerundeten Quotienten mit n Dezimalstellen erhalten, so muß man bei der Division einen Quotienten mit (n + 1) Dezimalstellen errechnen. In der letzten Stelle wird dann die "Rundungs-5" addiert.

Wenn man "2" durch "3" dividiert und der gerundete Quotient drei Dezimalstellen umfassen soll, sieht die Rechnung so aus:

2,0000 : 3 = 0,6666	Division
5	Runden
0,6671	
0,667	Endergebnis

Dividieren Sie bitte "10" durch "6"; der gerundete Quotient soll zwei Dezimalstellen haben. - Es ergibt sich _____(Z)_____.

* * * * *

1,67

Wir fassen zusammen:

Hat der Divisor Dezimalstellen, so wird man zunächst so erweitern, daß der Divisor eine ganze Zahl wird.

Dann muß man sich darüber klar werden, wie genau man den Quotienten errechnen will. Danach richtet sich, ob man den Dividenden in der vorgegebenen Stellenzahl übernehmen kann oder ob ihm, bei weiteren gewünschten Dezimalstellen des Quotienten, Nullen anzuhängen sind. Sollten gar weniger Dezimalstellen beim Ergebnis genügen, kann man entsprechende letzte Stellen des Dividenden entfallen lassen.

Soll der Quotient gerundet werden, muß der Dividend eine Dezimalstelle mehr enthalten als der gerundete Quotient.

Es folgen nun Übungen.

Gegeben ist:

96,563 : 1,7

Zunächst erweitern wir Dividend und Divisor, so daß der Divisor keine Dezimalstellen enthält:

(Z) : (Z)

* * * * *

965,63

17

Der gerundete Quotient soll drei Dezimalstellen haben. Wir müssen daher den Quotienten auf vier Dezimalstellen berechnen (XX,XXXR). Folglich muß der Dividend (Z) Dezimalstellen haben.

Es ergibt sich also:

(Z) : (Z) = XX,XXXR

* * * * *

4

965,6300

17

In den folgenden Übungen sind die Stellen für den Dividenden, den Divisor und für den gerundeten Quotienten gegeben.

Stellen Sie bitte den Dividenden fest, wie er für die Durchführung der Division benötigt wird. - Das erste Beispiel ist ausgeführt.

	Dividend	Divisor	gerundeter Quotient	bereitzustellender Dividend
a)	XX,X	XX,XX	XX,X	XXX0,00
b)	XXX	XXX	X,XX	_____
c)	X,X	X,XXX	X,XXX	_____
d)	XXXX	XX,X	XXXX,X	_____
e)	X,XXX	XX	X,XXX	_____

* * * * *

b) XXX,000

c) XX00,0000

d) XXXX0,00

e) X,XXX0

(1) Wir üben nun das richtige Einbringen des Dividenden in das Rechenfeld.

Gegeben sind:

Dividend: XXXX▲XXX Divisor: XXX▲XX

Der Quotient ist mit 3 Dezimalstellen zu errechnen (keine Rundung!).

Vor der Division müssen also die beiden Zahlen die folgende Form haben:

Dividend: _____ Divisor: _____

* * * * *

(Dividend) XXXXXX▲X00 (Divisor) XXXXX

Wir müssen dem Dividenden also ____(Z)____ Nullen anhängen.

* * * * *

2

Es liegt nahe, hierfür eine Multiplikation mit 100 zu programmieren. Dies werden wir jedoch nicht tun. (Die Multiplikation dauert relativ lange.)

Anstelle zu multiplizieren, werden wir unseren Dividenden um ____(Z)____ Stellen nach _____ versetzen.

* * * * *

2

links

Wir müssen dann noch dafür sorgen, daß der Dividend rechts die entsprechenden Nullen hat.

Erweiterter Dividend	Divisor
XXXXXX▲X00	XXXXX

Der gepackte Dividend benötigt ____(Z)____ Bytes. Der gepackte Divisor benötigt ____(Z)____ Bytes.

Das Rechenfeld für die Division muß also ____(Z)____ Bytes groß sein.

* * * * *

5 |X X|X X|X X|X 0|0 C|

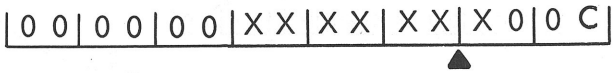
3 |X X|X X|X C|

8

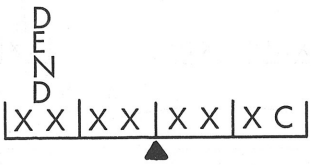
Vor der Division ist der _____ rechtsbündig ins Rechenfeld zu bringen.
Das Rechenfeld muß vor der Division also folgenden Inhalt haben:
_____.

* * * * *

Dividend



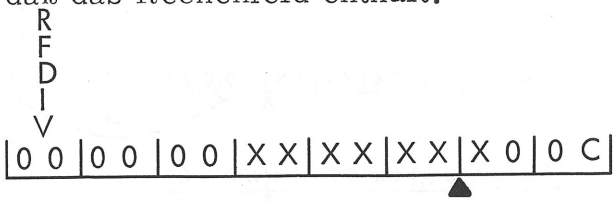
Betrachten Sie bitte die folgenden Felder:





(noch nicht erweitert)

Der Dividend (DEND) ist so in das Rechenfeld (RFDIV) zu bringen, daß das Rechenfeld enthält:



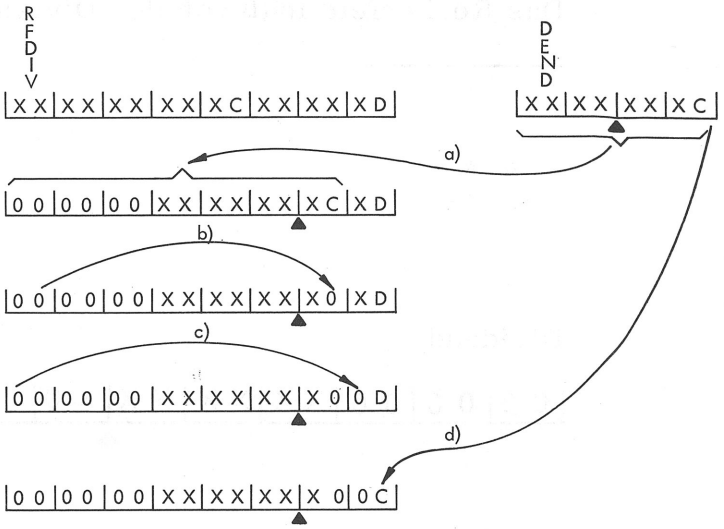
Schreiben Sie bitte auf ein Programmschema die hierfür erforderlichen Instruktionen. Zeichnen Sie sich dabei den jeweiligen Inhalt von RFDIV (und evtl. nötige Konstanten) auf ein Extrablatt.

* * * * *

Es gibt verschiedene Lösungen.
Vergleichen Sie bitte Ihre Lösung mit denen der Abb. 1 (Rückseite!).

Lösung A:

	Operation	Operand und Bemerkungen
	10 14	20 25 30
a)	ZAP	RFDIV(7),DEND
b)	MVN	RFDIV+6(1),RFDIV
c)	MVZ	RFDIV+7(1),RFDIV
d)	MVN	RFDIV+7(1),DEND+3



Lösung B:

	Operation	Operand und Bemerkungen
	10 14	20 25 30
a)	MVO	RFDIV,KXNULL
b)	ZAP	RFDIV(7),DEND
c)	MVN	RFDIV+6(1),KXNULL
d)	MVN	RFDIV+7(1),DEND+3

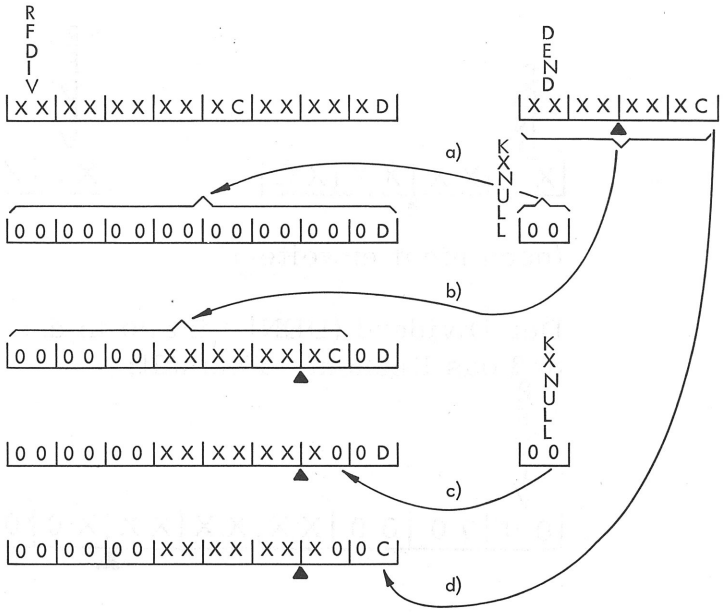


Abb. 1

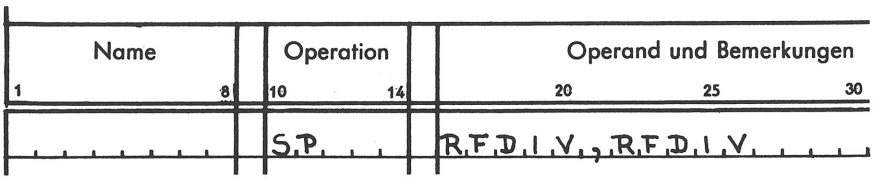
Arbeiten Sie bitte jede Lösung Instruktion für Instruktion durch.

Die Übertragung des Vorzeichens ist nur dann erforderlich, wenn wechselnde Vorzeichen vorkommen.

Betrachten Sie bitte die Lösung B (Abb. 1).

Zur Löschung des Rechenfeldes könnte (bei a)) statt MVO auch ZAP verwendet werden. Allerdings benötigen wir dann eine gepackte Konstante Null.

Was würde folgende Instruktion anstelle der MVO-Instruktion bewirken?

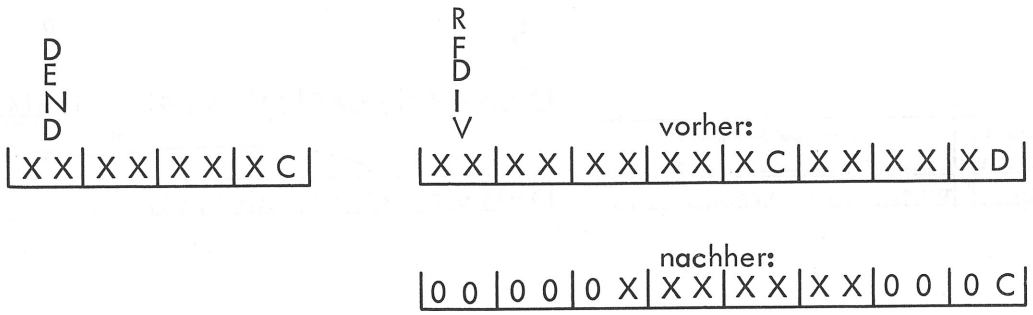


* * * * *

Wir bekämen einen Gültigkeitsfehler aufgrund des Vorzeichens im 5. Byte.
(Ein Vorzeichen darf nur im rechten Byte vorkommen. Alle anderen Bytes müssen je zwei der Ziffern 0-9 enthalten.)

Zeichnen Sie die Abb. 1 ab, wenn Sie es für erforderlich halten.

Wir wandeln das vorangegangene Beispiel ab. Dem Dividenden sind 3 Nullen anzuhängen.

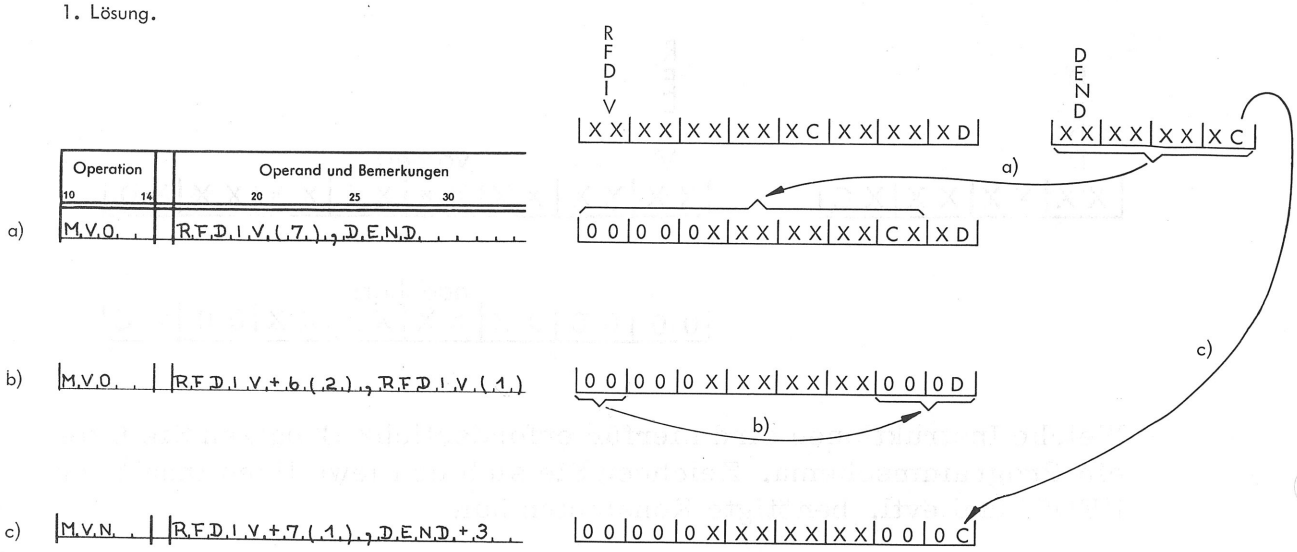


Welche Instruktionen sind hierfür erforderlich? Benutzen Sie bitte ein Programmschema. Zeichnen Sie sich den jeweiligen Inhalt von RFDIV und evtl. benötigte Konstanten auf.

* * * * *

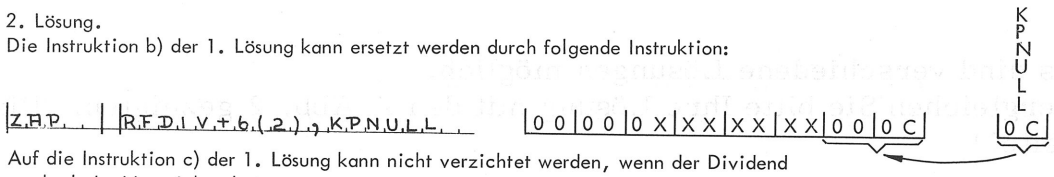
Es sind verschiedene Lösungen möglich.
Vergleichen Sie bitte Ihre Lösung mit den in Abb. 2 gezeigten. (Rückseite!)

1. Lösung.



Die MVN-Instruktion könnte auch ganz am Anfang stehen.

2. Lösung.
Die Instruktion b) der 1. Lösung kann ersetzt werden durch folgende Instruktion:



Auf die Instruktion c) der 1. Lösung kann nicht verzichtet werden, wenn der Dividend wechselndes Vorzeichen hat.

Abb. 2

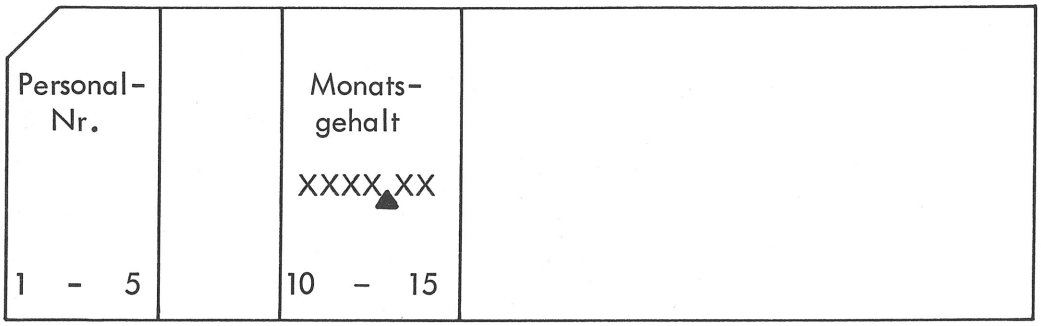
Falls Sie mehr Instruktionen verwendet haben, kann dies auch richtig sein.

Die gezeigten Lösungen müssen Ihnen klar sein.

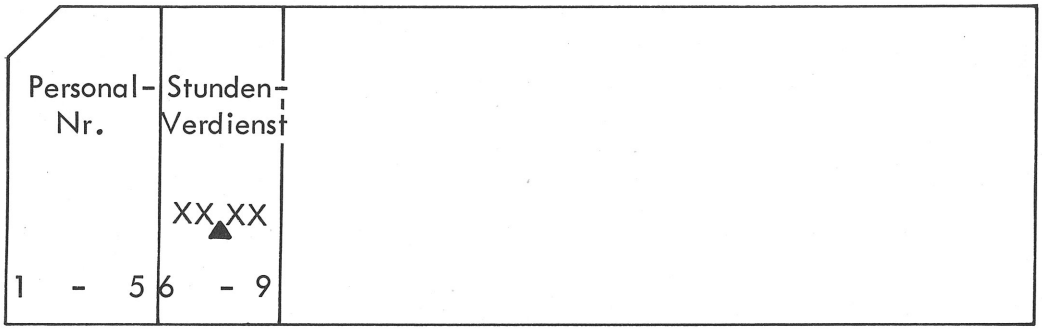
Wir wenden uns jetzt der Aufgabe 9 zu.

Aufgabe 9:

Eingabe: Gehaltskarte



Ausgabe: Statistikkarte



Verarbeitung:

Die monatliche Arbeitszeit beträgt 170,50 Stunden.

Stundenverdienst = Monatsgehalt : 170,50.

Der Stundenverdienst ist zu runden.

Der Stundenverdienst ist ohne Vorzeichen zu stanzen.

Abb. 1

Aufgabe 9

Machen Sie sich bitte mit der Aufgabe 9 (Abb. 1) vertraut. -

Entnehmen Sie Ihren Arbeitsmitteln

a) die Aufgabe 9

b) ein Entwurfsblatt für Programmablaufpläne.

Erstellen Sie bitte den Programmablaufplan.

* * * * *

Vergleichen Sie Ihren Programmablaufplan mit dem der Abb. 2 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 9 Nr.: Seite:

Programmierer: Datum:

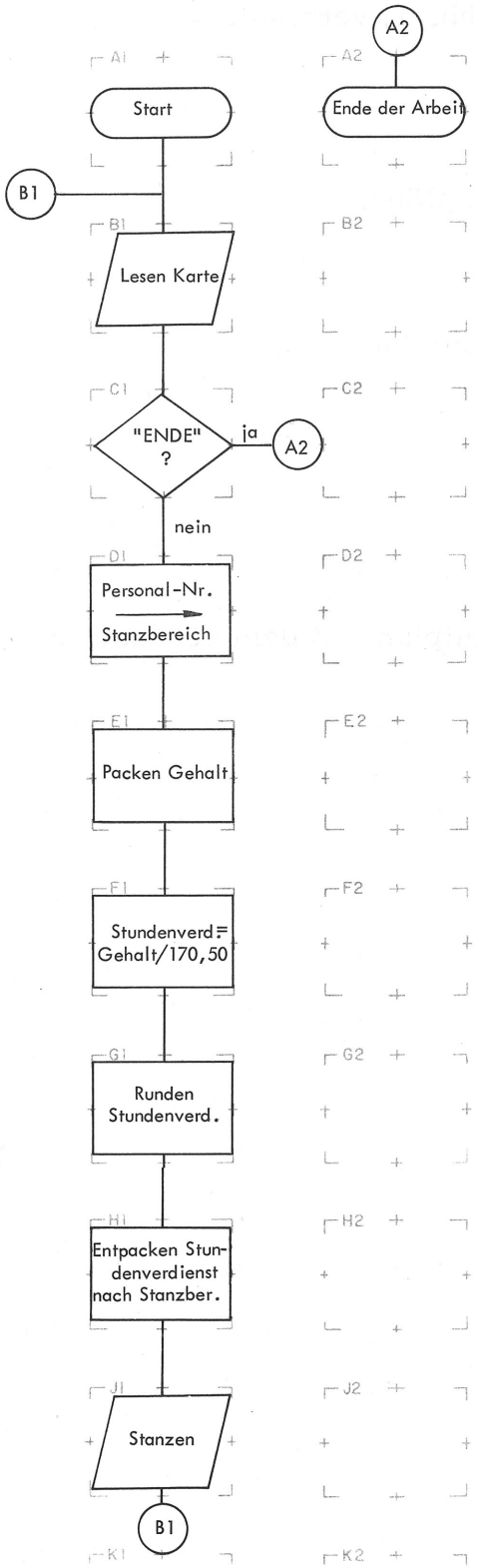


Abb. 2

Zum Programmablaufplan:

Vielleicht haben Sie detailliertere Eintragungen in den Blöcken E1 und H1 vorgenommen bzw. weitere Blöcke hinzugefügt.
In Abb. 2 enthalten die Blöcke E1 und H1 jeweils nur "grobe" Angaben.

Entnehmen Sie bitte Ihren Arbeitsmitteln den Programmablaufplan zur Aufgabe 9.

Ermitteln der Länge des Rechenfeldes.

1. Bestimmen der Länge des Dividenden.
   Der Divisor und der Dividend werden mit der gleichen Zehnerpotenz multipliziert (aber nur im Geiste!), so daß der Divisor keine Dezimalstellen mehr enthält.

   Bei der Division erhält der Quotient so viele Dezimalstellen, wie der Dividend hat. Soll der Quotient weitere Dezimalstellen erhalten, so müssen dem Dividenden entsprechend viele Nullen angehängt werden.

2. Bestimmen der Länge des Rechenfeldes.
   Länge gepackter Dividend + Länge gepackter Divisor = Länge Rechenfeld. Länge jeweils in Bytes.

   Anmerkung: Bei der Länge des Dividenden sind die evtl. angehängten Nullen mitzuzählen.

Abb. 3

Bevor wir die Speicherbelegung zeichnen, machen wir uns klar, wie groß unser Rechenfeld sein muß.

Lesen Sie bitte den Text in Abb. 3 aufmerksam durch.

Der gepackte Dividend belegt ____(Z)____ Bytes. Der gepackte Divisor benötigt ____(Z)____ Bytes. Das Rechenfeld muß daher ____(Z)____ Bytes groß sein.

* * * * *

5 |X X|X X|X X|0 0|0 C|

3 |1 7|0 5|0 C|

8

Zeichnen Sie nun die Speicherbelegung. Beachten Sie bitte folgende Angaben:

a) Eingabebereich ab 801, daran anschließend Ausgabebereich, Rechenfeld und Konstanten.

b) Verwenden Sie bitte die nachstehenden symbolischen Namen:
EBER, ENR, EGEHALT,
ABER, ANR, ASVD,
RFSVD, KCENDE, KPSTD (Konstante 170,50).

* * * * *

Überprüfen Sie bitte Ihre Speicherbelegung anhand der Abb. 4 (Rückseite!).

Aufgabe 9:

IBM System / 360
Speicherbelegung

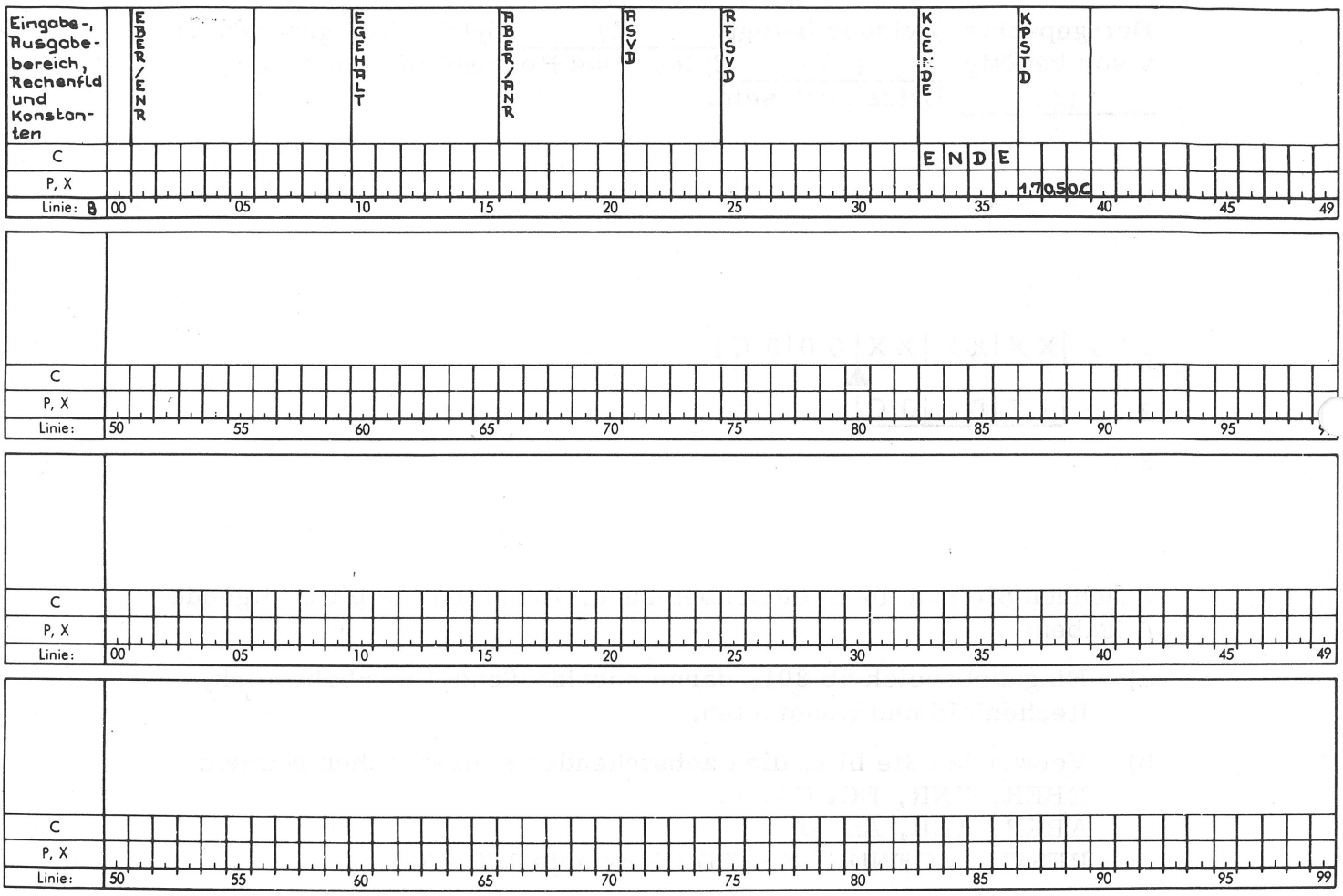


Abb. 4

Verbessern Sie bitte Ihre Speicherbelegung soweit notwendig.

Schreiben Sie nun die Definitionen. Verwenden Sie die Überschriften "Errechnen durchschnittlicher Stundenverdienst" und "Definitionen".

* * * * *

Vergleichen Sie bitte Ihre Definitionen mit denen der Abb. 5 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 9	Lochhinweise		Problem A09 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 01 (76–77) von 2

Anweisung				Karte Nr.
Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)		(78–80)
*, ,E,R,R,E,C,H,N	E,N, ,D,U,R	C,H,S,C,H,N,I ,T,T,L,I ,C,H,E,R, ,S,T,U,N,D,E,N,V,E,R,D,I ,E,N,S,T,		0,1,0
*, ,D,E,F,I,N,I,T	I ,O,N,E,N			0,2,0
	S,T,A,R,T	8,0,1,		0,3,0
E,B,E,R,	D,S,	0,C,L,1,5,		0,4,0
E,N,R,	D,S,	C,L,5,		0,5,0
	D,S,	C,L,4,		0,6,0
E,G,E,H,A,L,T,	D,S,	C,L,6,		0,7,0
A,B,E,R,	D,S,	0,C,L,9,		0,8,0
A,N,R,	D,S,	C,L,5,		0,9,0
A,S,V,D,	D,S,	C,L,4,		1,0,0
R,F,S,V,D,	D,S,	C,L,8,		1,1,0
K,C,E,N,D,E,	D,C,	C,',E,N,D,E,',		1,2,0
K,P,S,T,D,	D,C,	P,',1,7,0,5,0,'		1,3,0
				1,4,0
				1,5,0
				1,6,0
				1,7,0
				1,8,0
				1,9,0
				2,0,0

Abb. 5

Verbessern Sie bitte Ihre Definitionen falls nötig.

Entnehmen Sie bitte Ihren Arbeitsmitteln das Merkblatt 102 (2 Seiten).

Lesen Sie das Merkblatt aufmerksam durch.

Codieren Sie nun den Programmablauf.

Beachten Sie bitte:

a) Verwenden Sie bei der Codierung Merkblatt 102 als Hilfsmittel.

b) Zeichnen Sie sich den jeweiligen Inhalt der wichtigen Felder auf.

c) Tragen Sie weitere erforderliche Konstanten in die Speicherbelegung ein.

d) Erweitern Sie die Definitionen entsprechend.

* * * * *

Überprüfen Sie Ihre Eintragungen anhand der Abb. 7 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 9	Lochhinweise		Problem A09 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 02 (76–77) von 2

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Anweisung	Karte Nr. (78–80)
* PROGRAMMABLAUF			BLOCK IM PROGRAMMABLAUFPLAN:	010
B1	GET	KARTE,EBER	B1	020
	CLC	EBER(4),KCENDE	C1	030
	BE	A2	C1	040
	MVC	ANR,ENR	D1	050
	PACK	EGEHALT,EGEHALT	E1	060
	MVO	RFSVD(7),EGEHALT	E1	070
	ZAP	RFSVD+6(2),KP0	E1	080
	DP	RFSVD,KPSTD	F1	090
	AP	RFSVD(5),KP5	G1	100
	MVO	RFSVD(5),RFSVD(4)	G1	110
	UNPK	ASVD,RFSVD+2(3)	H1	120
	MVZ	ASVD+3(1),ASVD	H1	130
	PUT	STANZ,ABER	J1	140
	B	B1		150
A2	Ende der Arbeit		A2	160
	END			170
				180
				190
				200

Eingabe-, Ausgabe-bereich, Rechen-feld u. Konstante	EBER/ENR	EGEHALT	ABER/ANR	ASVD	RFSVD	KCENDE	KPSTD	KP0	KP5
C						ENDE			
P, X							17050C	0C	5C

Linie: 8 (00–49)

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Anweisung	Karte Nr. (78–80)
* ERRECHNEN DURCHSCHNITTLICHER STUNDENVERDIENST				010
* DEFINITIONEN				020
	START	801		030
EBER	DS	0CL15		040
ENR	DS	CL5		050
	DS	CL4		060
EGEHALT	DS	CL6		070
ABER	DS	0CL9		080
ANR	DS	CL5		090
ASVD	DS	CL4		100
RFSVD	DS	CL8		110
KCENDE	DC	C'ENDE'		120
KPSTD	DC	P'17050'		130
KP0	DC	P'0'		140
KP5	DC	P'5'		150

Abb. 6

Zur Codierung:

Die Codierung in Abb. 6 zeigt eine mögliche Lösung.

Auf den folgenden Seiten sind die im Zusammenhang mit der Division wesentlichen Instruktionen und Felder aufgeführt. Außerdem werden weitere Lösungsmöglichkeiten angedeutet.

Operation	Operand und Bemerkungen
10 14	20 25 30
P,A,C,K,	E,G,E,H,A,L,T,,E,G,E,H,A,L,T, . . .
M,V,O, .	R,F,S,V,D,(,7,),,E,G,E,H,A,L,T, . . .
Z,A,P, .	R,F,S,V,D,+,6,(,2,),,K,P,Ø, ...

Es genügt nicht, nur auf das Vorzeichen eine "0" zu übertragen, da das Rechenfeld nach der 1. Division ja rechts den Rest enthält.

D,P, . .	R,F,S,V,D,,K,P,S,T,D, ...
A,P, . .	R,F,S,V,D,(,5,),,K,P,5, ...
M,V,O, .	R,F,S,V,D,(,5,),,R,F,S,V,D,(,4,) . .
U,N,P,K,	A,S,V,D,,R,F,S,V,D,+,2,(,3,), ...
M,V,Z, .	A,S,V,D,+,3,(,1,),,A,S,V,D, ...

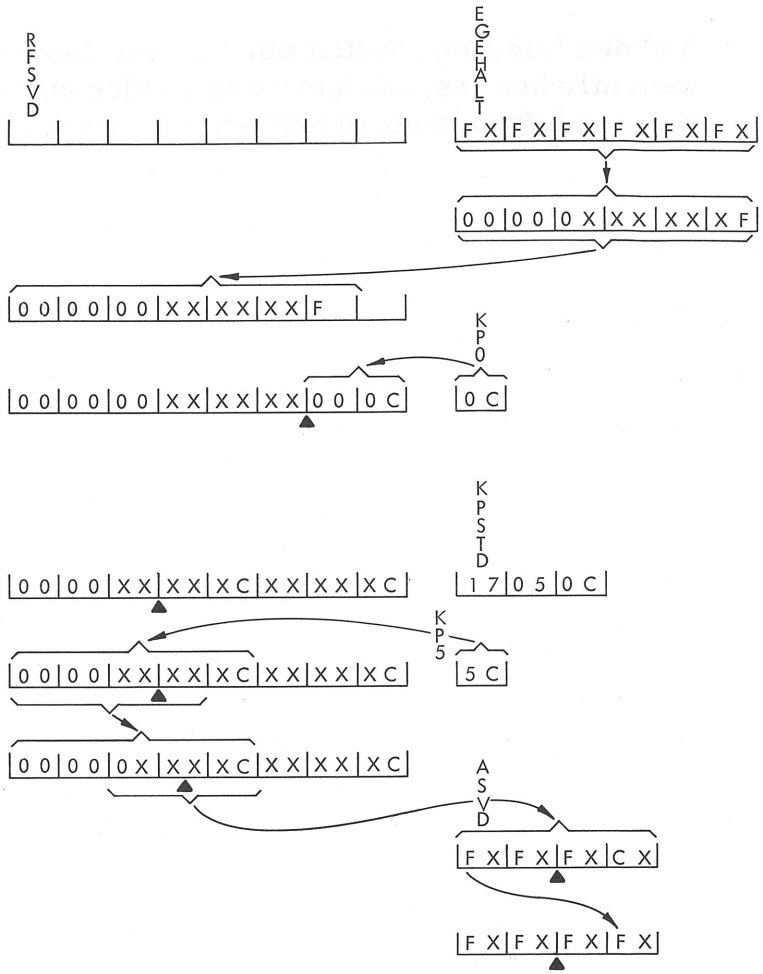


Abb. 7

Überlegen Sie bei jeder Instruktion, warum sie so lautet und was sie bewirkt.

Andere mögliche Lösungen:

Statt ZAP eine weitere MVO-Instruktion und eine MVN-Instruktion:

```
MVO.  | RFSVD+6(2),RFSVD(1)...
MVN.  | RFSVD+7(1),EGEHALT+5..
```

Die beiden Instruktionen sind auch in der umgekehrten Reihenfolge richtig.

Die MVN-Instruktion wird überflüssig, wenn das Rechenfeld RFSVD wie folgt definiert wird:

```
RFSVD...  | DC...  | PL8'0'.
oder
RFSVD...  | DC...  | XL8'C'.
```

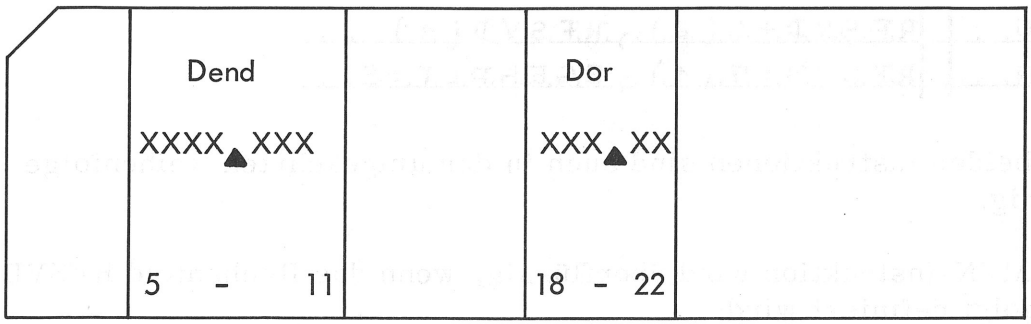
Das Einbringen des Vorzeichens in RFSVD ist nämlich nur für die e r s t e Division von Bedeutung.

Verbessern Sie nun Ihre Codierung soweit erforderlich.

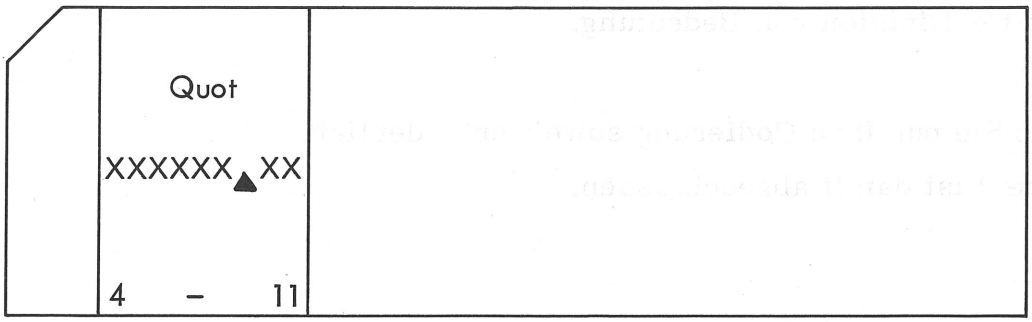
Die Aufgabe 9 ist damit abgeschlossen.

Aufgabe 10

Eingabe:



Ausgabe:



Verarbeitung:

Sowohl der Dividend (Dend) als auch der Divisor (Dor) können positiv und negativ sein.

Der Divisor kann auch null sein.

Der Quotient ist zu runden.

Abb. 1

Aufgabe 10

Sehen Sie sich bitte die Aufgabe 10 (Abb. 1) genau an. -

Sie wissen, daß bei der Division ein Divisor _____ nicht erlaubt ist.

* * * * *

null

Wir müssen also vor der Division den Divisor prüfen.

Die Verarbeitung der betreffenden Karte (muß unterbleiben, muß nicht unterbleiben) _____, wenn der Divisor gleich "0" ist.

* * * * *

muß unterbleiben

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) die Aufgabe 10

b) den Programmablaufplan zur Aufgabe 10.

Machen Sie sich mit dem Programmablaufplan vertraut.

Wir bestimmen jetzt die Größe des Rechenfeldes. (Verwenden Sie hierzu Merkblatt 102.)

Nach dem Entfernen der Dezimalstellen im Divisor sieht unser Dividend wie folgt aus: _____.

* * * * *

XXXXXX▲X

Da der Quotient mit ____(Z)____ Dezimalstellen zu errechnen ist, müssen wir dem Dividenden ____(Z)____ Nullen anhängen.

* * * * *

3 (wegen Runden)

2 (XXXXXX▲X00 : XXXXX = XXXXXX▲XXX)

Das Rechenfeld muß ____(Z)____ Bytes groß gewählt werden.

* * * * *

8

Dend 5 Bytes											Dor 3 Bytes			
X	X	X	X	X	X	X	0	0	V	+	X	X	X	V
				▲										

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) die Speicherbelegung zur Aufgabe 10

b) die dazugehörenden Definitionen.

Vergleichen Sie die Definitionen mit der Speicherbelegung.

Eventuell noch erforderliche Konstanten tragen wir später nach.

Codieren Sie nun den Programmablauf.

Beachten Sie bitte:
Verwenden Sie dabei wieder Merkblatt 102.
Zeichnen Sie den jeweiligen Inhalt der wichtigen Felder auf.

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 10 Nr.: Seite:

Programmierer: Datum:

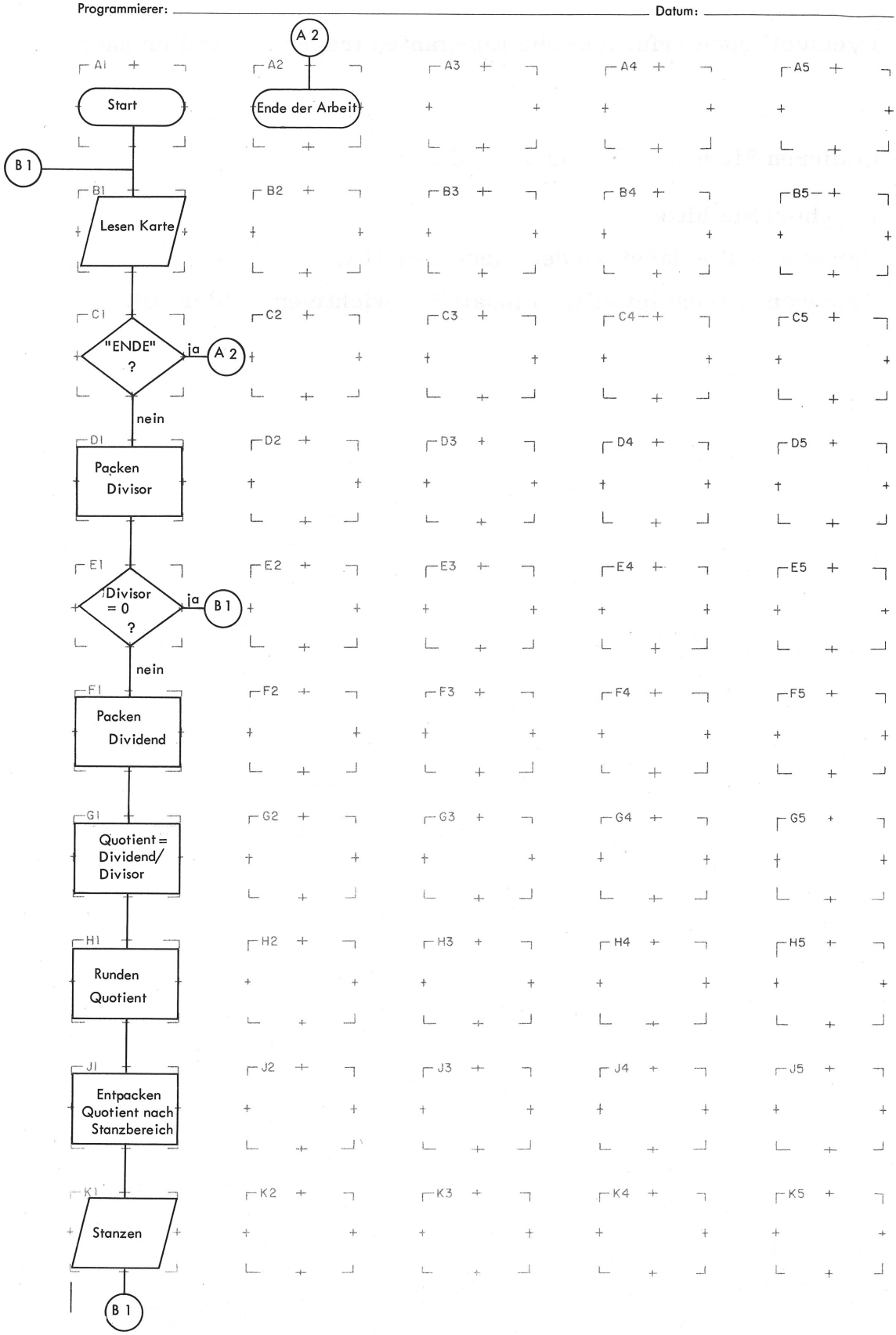


Abb. 2

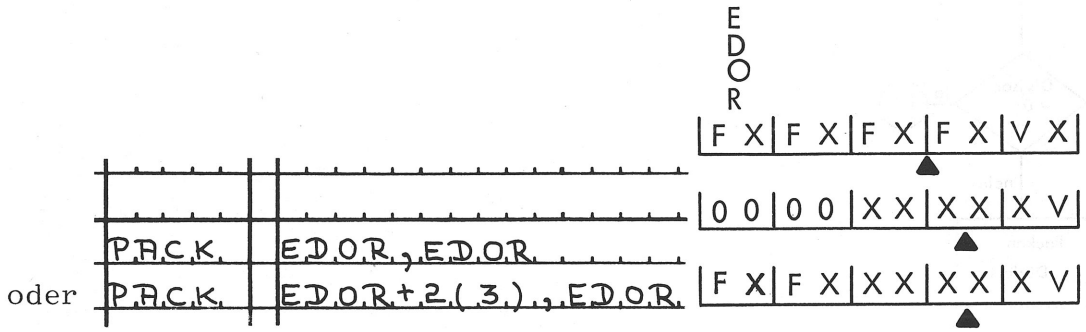
Wir gehen jetzt eine mögliche Codierung Schritt für Schritt durch. Im Anschluß daran erhalten Sie Hinweise auf weitere Lösungen.

Die Codierung der Schritte B1 und C1 dürfte keine Schwierigkeiten bereitet haben.

Die Instruktionen müssen lauten:

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–25–30)
B.1.	G.E.T.	K.A.R.T.E.,E.B.E.R.
	C.L.C.	E.B.E.R.(.4.),K.C.E.N.D.E.
	B.E.	A.2.

Der Block D1 ist wie folgt zu codieren:



Für die Prüfung auf Divisor null (Block E1) haben Sie wahrscheinlich einen gepackten Vergleich mit anschließender Abfrage programmiert.

Die Instruktionen hierfür müßten lauten:

	C.P.	E.D.O.R.,K.P.Ø. bzw. E.D.O.R.+2.(3.),K.P.Ø.
	B.E.	B.1.

EDOR+2(3) muß es heißen, wenn das Empfangsfeld in der PACK-Instruktion EDOR+2(3) war.

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 10 Nr.: Seite:

Programmierer: Datum:

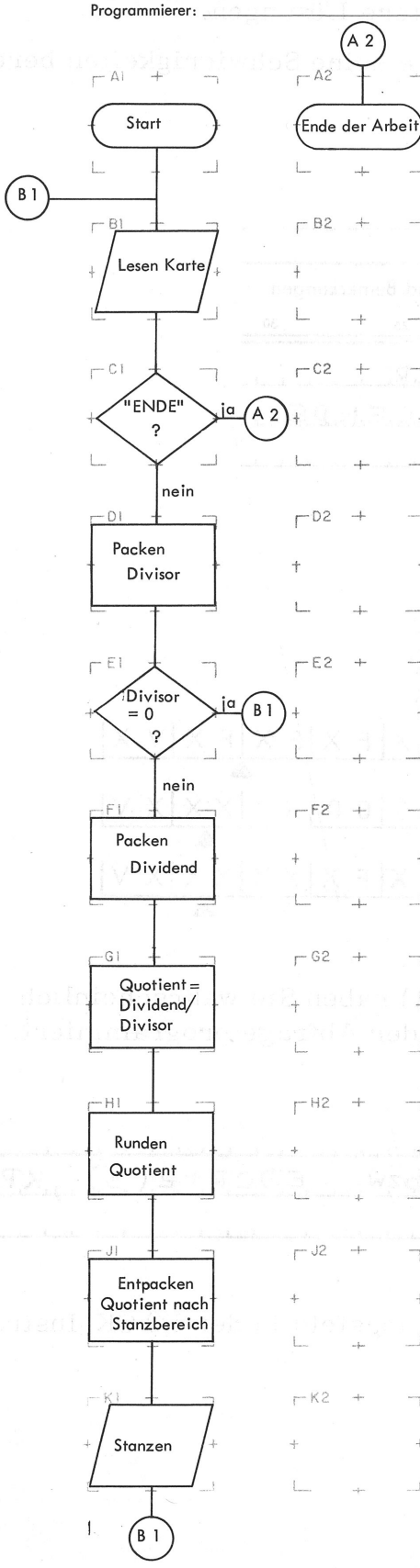


Abb. 2

Die Konstante "KP0" braucht in jedem Fall nur (Z) Byte(s) groß zu sein.

* * * * *

1 (Ist bei CP der 2. Operand kürzer als der erste, so wird er automatisch - bis zur Länge des 1. Operanden - mit Nullen erweitert verarbeitet.)

Ein logischer Vergleich ist in diesem speziellen Fall auch möglich, jedoch sehr umständlich (Vorzeichen des Divisors!).

Es gibt aber eine weitere Möglichkeit, festzustellen, ob der Divisor gleich null ist. Hierzu benötigen wir nicht einmal eine Konstante.

Sie wissen, daß die arithmetischen Operationen AP, SP und ZAP den Bedingungsschlüssel neu setzen (selbstverständlich auch CP). Wir können also im Anschluß an diese Operationen eine Abfrage auf positiv, negativ oder null durchführen.

Welche beiden Instruktionen sind zur Prüfung auf Divisor null zu schreiben?

* * * * *

ZAP	EDOR,EDOR bzw. EDOR+2(3),EDOR+2(3)
BZ	B1

Wir löschen und addieren den Divisor in die gleichen Stellen. Dabei wird der Bedingungsschlüssel neu gesetzt.

Aufgrund der anschließenden BZ-Instruktion (Branch on Zero, Verzweigen bei Null) wird dann nach B1 verzweigt, wenn der Divisor null ist.

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 10 Nr.: Seite:

Programmierer: Datum:

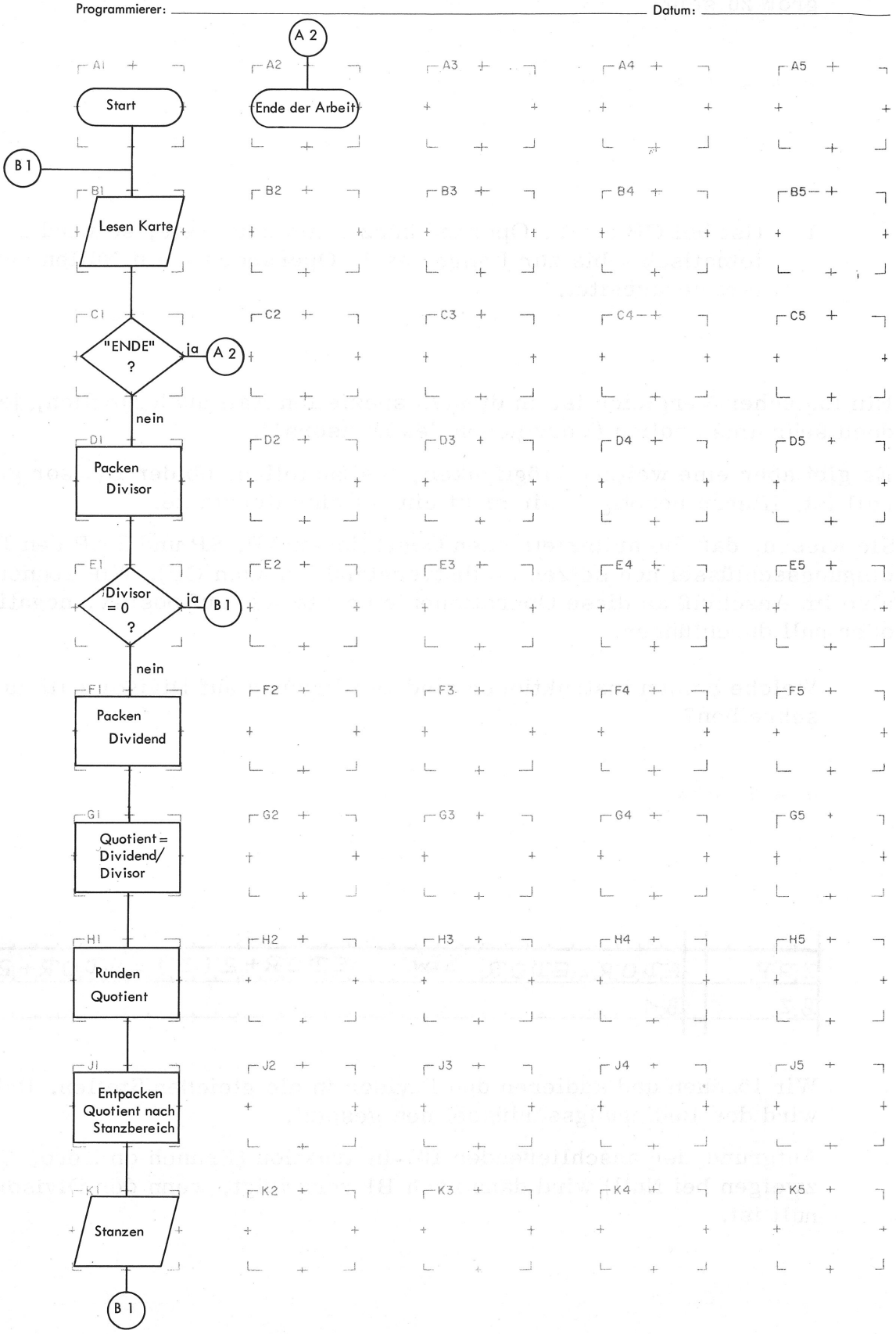
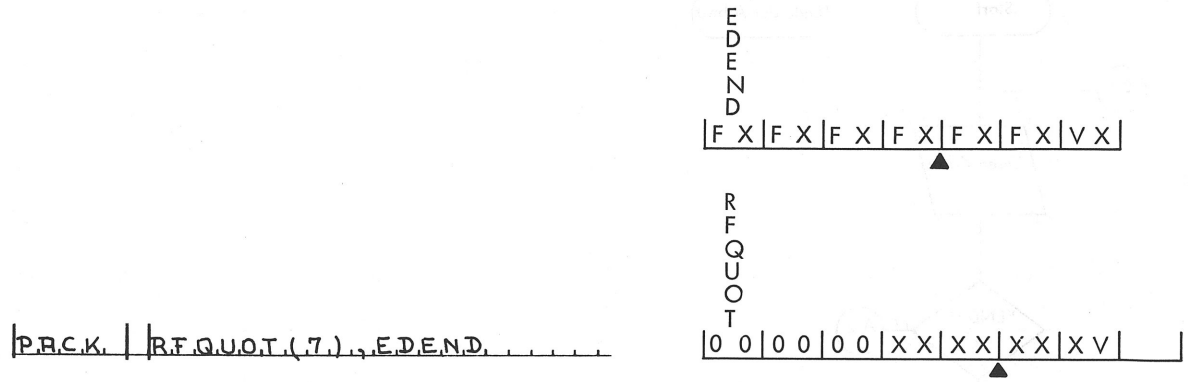


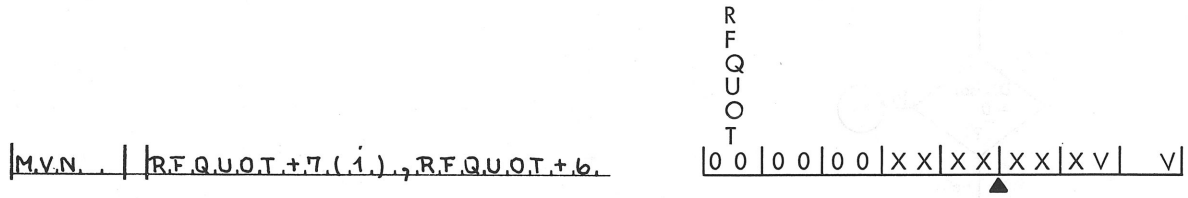
Abb. 2

Nun codieren wir den Block F1.

Den Dividenden packen wir so ins Rechenfeld, daß wir anschließend noch 2 Nullen anhängen können.



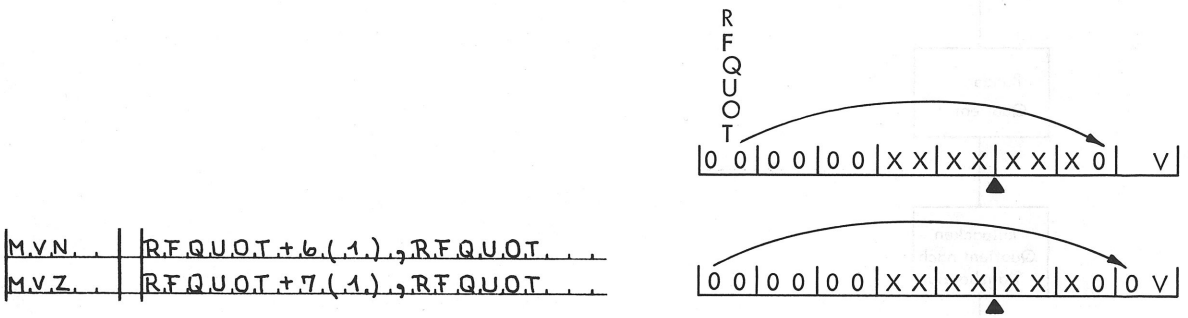
Bevor wir die beiden Nullen anhängen, müssen wir jedoch zuerst das Vorzeichen in das rechte Byte bringen.



Nun können wir die beiden Nullen einbringen.

Hierzu ist allerdings die Instruktion MVO unbrauchbar, da diese die letzte Ziffer des Dividenden zerstören würde. Wir benötigen also eine MVN- und eine MVZ-Instruktion.

Die Codierung sieht wie folgt aus:



IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 10 Nr.: Seite:

Programmierer: Datum:

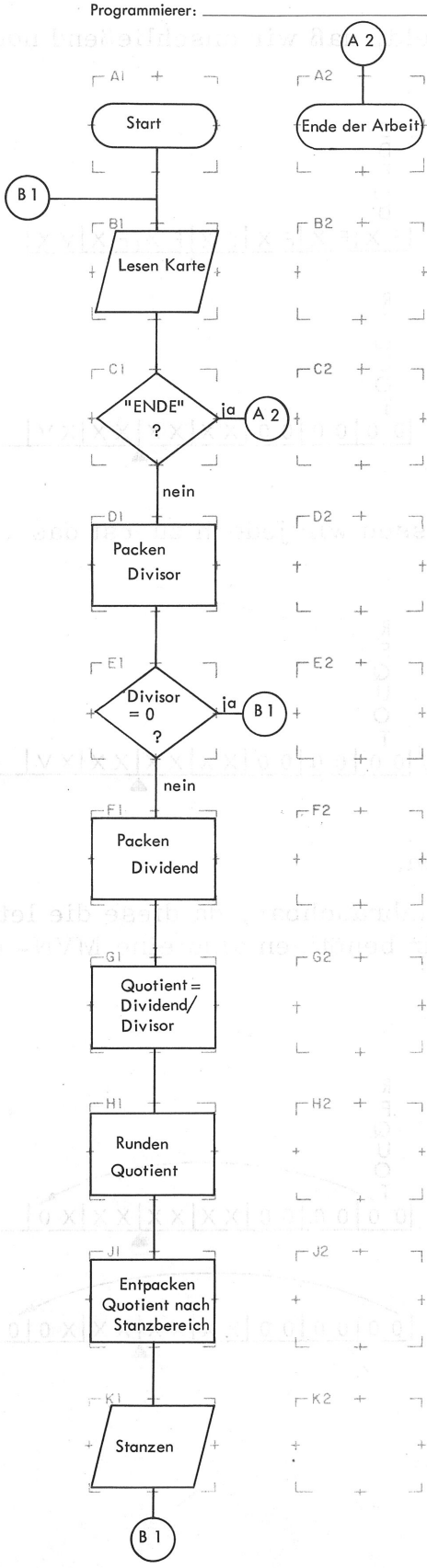
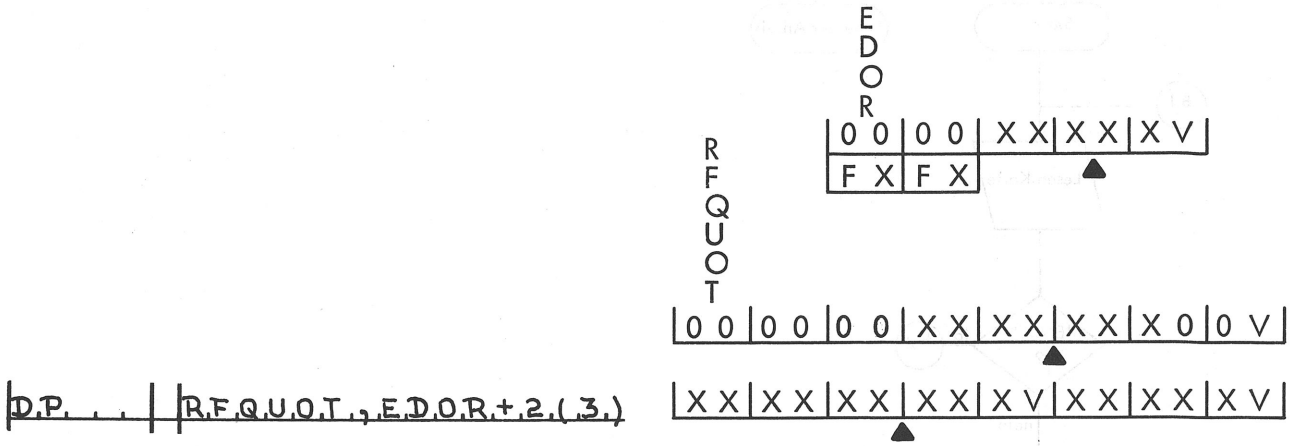


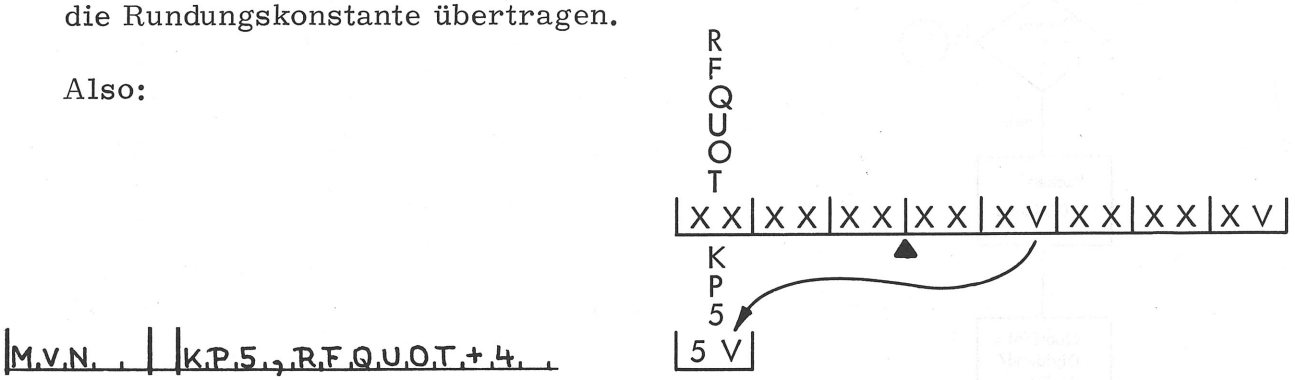
Abb. 2

Die Codierung für den Block G1 lautet:



Wir codieren jetzt den Block H1.
Vor der Rundung müssen wir das jeweilige Vorzeichen des Quotienten auf die Rundungskonstante übertragen.

Also:

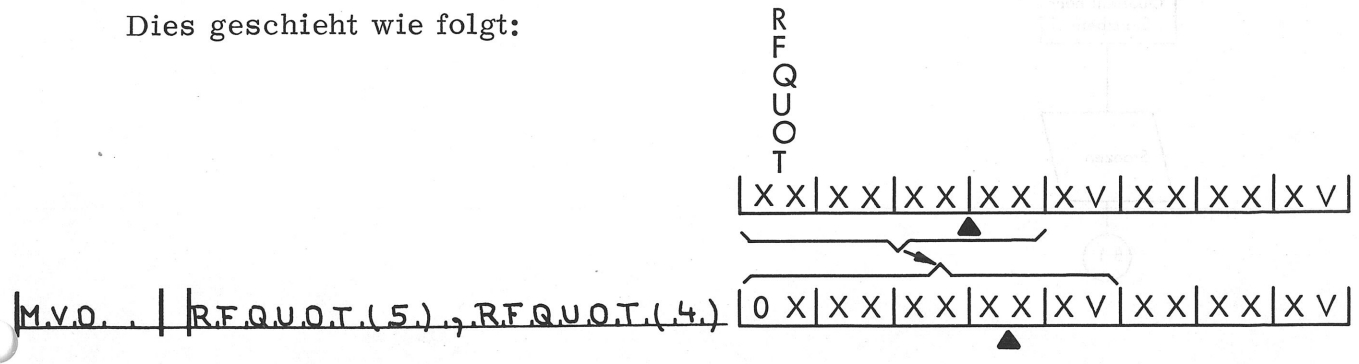


Die Instruktion für das Runden lautet:

R.P. . . | |R.F.Q.U.O.T.(5.),K.P.5.

Nun müssen wir zur Vorbereitung der Ausgabe unseren Quotienten um eine Stelle nach rechts versetzen.

Dies geschieht wie folgt:



IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 10 Nr.: Seite:

Programmierer: Datum:

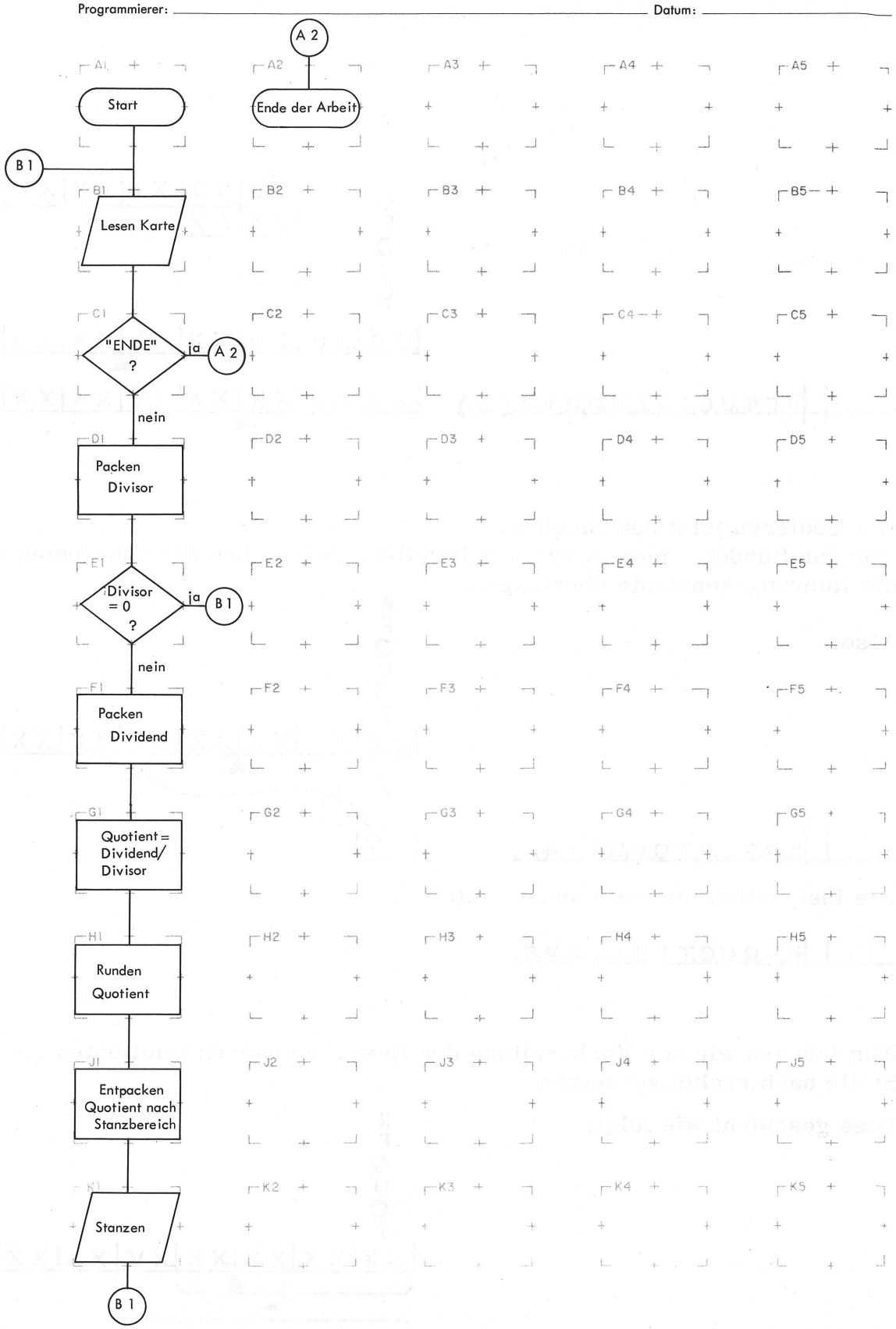
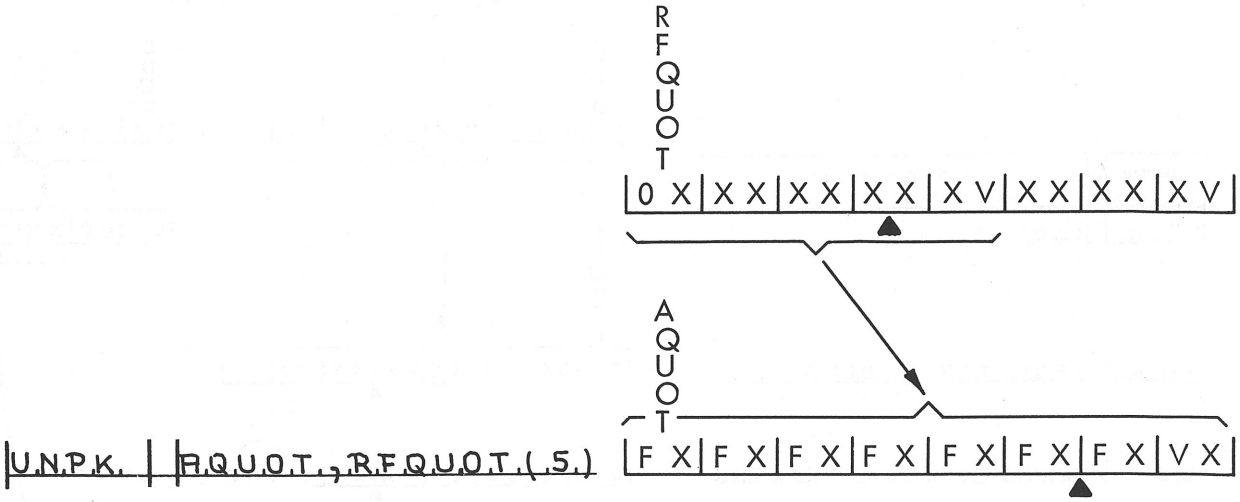


Abb. 2

Das Entpacken des Quotienten (Block J1) bekommen wir durch:



Die restlichen Blöcke sind wie folgt zu codieren:

	P.U.T.	S.T.A.N.Z., A.B.E.R.
	B.	B.1
A.2.	Ende der Arbeit	
	E.N.D.	

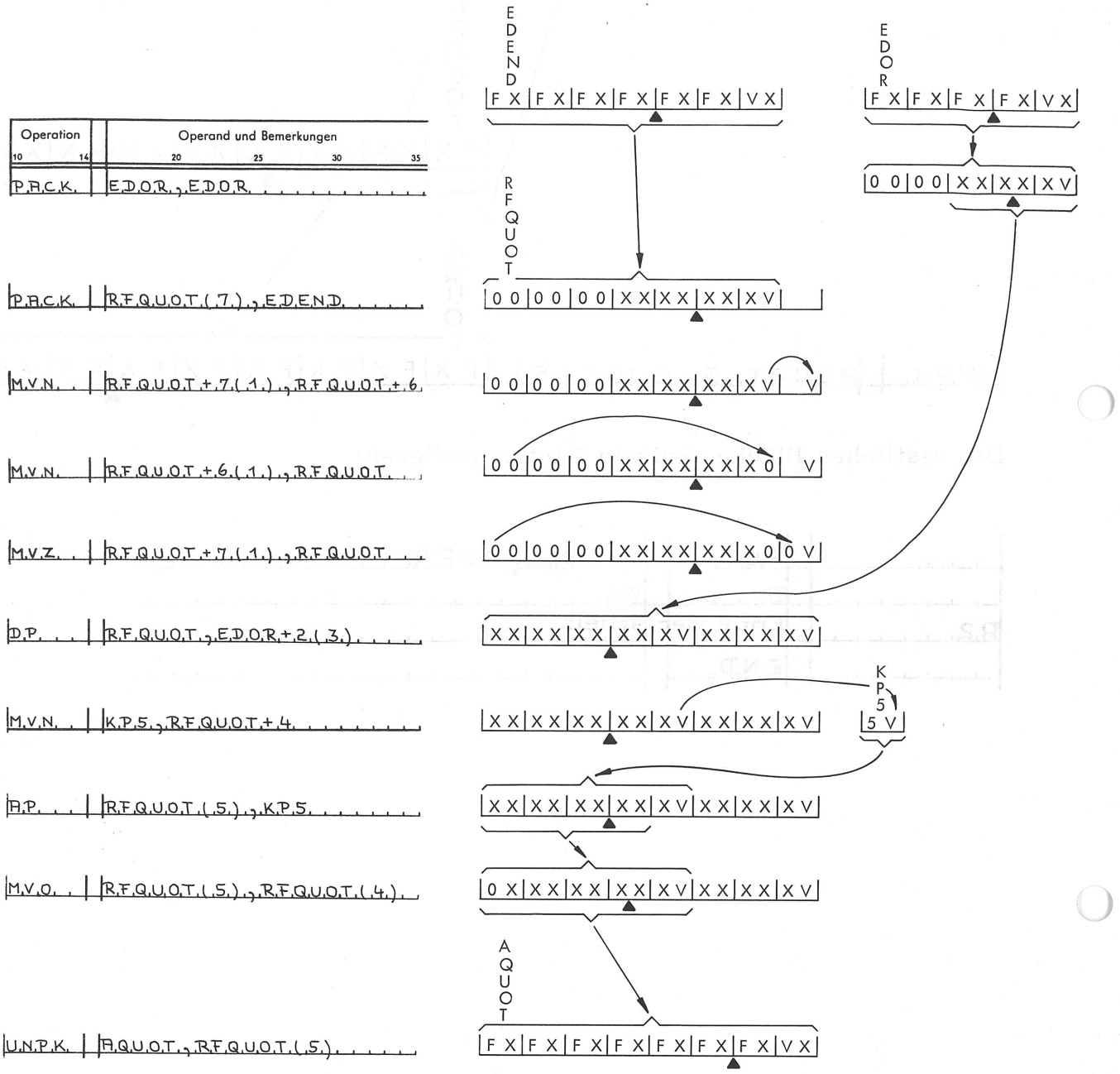

Operation
Operand und Bemerkungen
10 14 20 25 30 35
P.A.C.K. E.D.O.R.,E.D.O.R.
P.A.C.K. R.F.Q.U.O.T.(.7.),E.D.E.N.D.
M.V.N. R.F.Q.U.O.T.+7.(.1.),R.F.Q.U.O.T.+6.
M.V.N. R.F.Q.U.O.T.+6.(.1.),R.F.Q.U.O.T.
M.V.Z. R.F.Q.U.O.T.+7.(.1.),R.F.Q.U.O.T.
D.P. R.F.Q.U.O.T.,E.D.O.R+2.(.3.)
M.V.N. K.P.5.,R.F.Q.U.O.T.+4.
A.P. R.F.Q.U.O.T.(.5.),K.P.5.
M.V.O. R.F.Q.U.O.T.(.5.),R.F.Q.U.O.T.(.4.)
U.N.P.K. A.Q.U.O.T.,R.F.Q.U.O.T.(.5.)
EDEND
EDOR
RFQUOT
AQUOT
KP5

Abb. 3

Die Abb. 3 zeigt die wesentlichen Instruktionen und Felder nochmals im Zusammenhang. (Die Prüfung auf Divisor null ist der besseren Übersicht wegen ausgelassen.)

Hinweise für weitere Lösungen:

a) Wir können auch den Dividenden ins gleiche Feld packen und ihn dann z. B. mit ZAP nach RFQUOT bringen.

b) Es ist möglich, RFQUOT vor dem Einbringen des Dividenden zu löschen. Dann würde die MVZ-Instruktion in Abb. 3 hinfällig.

c) Die dem Dividenden anzuhängenden Nullen können wir auch einer Konstanten "Null" entnehmen (`⌊0 0⌋`).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 10		Lochhinweise		Problem A10 (73–75)
			Zeichen		
Programmierer		Datum	Lochung		Seite 012 (76–77) von 2

Name (1–8)	Operation (10–14)	Anweisung: Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.PROGRAM	MABLAUF		010
B1	GET	KARTE,EBER	020
	CLC	EBER(4),KCENDE	030
	BE	A2	040
	PACK	EDOR,EDOR	050
	ZAP	EDOR,EDOR	060
	BZ	B1	070
	PACK	RFQUOT(7),EDEND	080
	MVN	RFQUOT+7(1),RFQUOT+6	090
	MVN	RFQUOT+6(1),RFQUOT	100
	MVZ	RFQUOT+7(1),RFQUOT	110
	DP	RFQUOT,EDOR+2(3)	120
	MVN	KP5,RFQUOT+4	130
	AP	RFQUOT(5),KP5	140
	MVO	RFQUOT(5),RFQUOT(4)	150
	UNPK	AQUOT,RFQUOT(5)	160
	PUT	STANZ,ABER	170
	B	B1	180
A2	Ende der Arbeit		190
	END		200
			210

Abb. 4

Die Abb. 4 zeigt die Codierung des gesamten Programmablaufs.

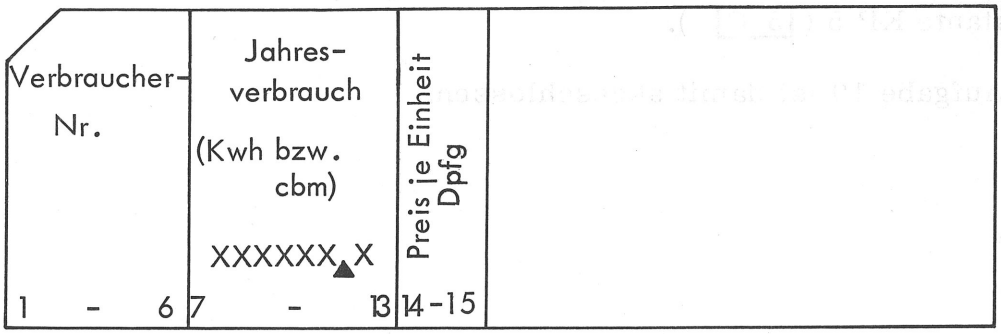
Die Codierung finden Sie auch in Ihren Arbeitsmitteln.

Vervollständigen Sie nun die Speicherbelegung und die Definitionen um die Konstante KP 5 (5 C).

Die Aufgabe 10 ist damit abgeschlossen.

Aufgabe 11

Eingabe : Verbrauchskarte



(Kwh = Kilowattstunden ; cbm = Kubikmeter)

Ausgabe: Verbrauchsliste

Verbraucher-Nr.	Jahresverbrauch	Preis je Einheit	Jahresbetrag	Durchschnittlicher Monatsbetrag
001010	0031205	011	00034326	0002860
001010	0002800	085	0002380	000198
001030	0004201	030	0001260	000105
001030	0000905	085	0000769	000064
002087	0046304	008	0003704	000308
002087	0003602	085	0003061	000255
153002	0856400	007	0059948	004995
153002	2208000	085	1876800	156400
160923	1103750	007	0077262	006438
160923	0508125	085	0431906	035992
9 - 14	17 - 23	26-28	31 - 38	41 - 47

Verarbeitung:

Jahresbetrag = Jahresverbrauch · Preis je Einheit .

Der Jahresbetrag ist zu runden.

Monatsbetrag = Jahresbetrag : 12.

Der Monatsbetrag ist aus dem n i c h t gerundeten Jahresbetrag zu errechnen.

Der Monatsbetrag ist zu runden.

Alle Werte sind positiv.

Abb. 1

Aufgabe 11

Sehen Sie sich die Aufgabe 11 (Abb. 1) genau an. -

Entnehmen Sie bitte Ihren Arbeitsmitteln die Aufgabe 11.

Diese Aufgabe enthält eine Multiplikation und eine Division.

Sowohl das Produkt, als auch der Quotient sind zu _____

Es ist jedoch darauf zu achten, daß der Quotient (durchschnittlicher Monatsbetrag) aus dem _____ Produkt zu errechnen ist.

* * * * *

runden

ungerundeten

Das Rechenfeld für die Multiplikation erfordert ____(Z)____ Bytes.

* * * * *

6 (|X X|X X|X X|X +| · |0 X|X+| = |0 0|X X|X X|X X|X X|X +|)

Das Rechenfeld für die Division muß ____(Z)____ Bytes groß sein.

* * * * *

7 (|X X|X X|X X|X X|X +| : |0 1|2 +| = |0 X|X X|X X|X X|X +|0 X|X +|)

Betrachten Sie bitte den Preis in der Karte und in der Liste. -

In der Liste wird der Preis als DM-Betrag geschrieben. - Deshalb ist den Daten aus der Karte generell eine ____(Z)____ vorangestellt. Wie erreichen wir dies?

* * * * *

0

z. B. durch Übertragen einer Konstanten "Null"
oder durch Definition einer Konstanten "Null" in der Stelle 26 des Druckbereiches.

Wir entscheiden uns für die letztere Möglichkeit. Sie hat den Vorteil, daß wir uns die Übertragungsoperation sparen. Der Druckbereich braucht nach dem Drucken nicht geblankt zu werden, so daß die "Null" erhalten bleibt.

Lösen Sie nun die gesamte Aufgabe.

Gehen Sie dabei wie folgt vor:

a) Erstellen Programmablaufplan

b) Erstellen Speicherbelegung

   Verwenden Sie die folgenden symbolischen Namen:
   EBER, ENR, EJVERBR, EPREIS,
   ABER, ANR, AJVERBR, APREIS, AJBETR, AMBETR,
   KCENDE, RFJBETR, RFMBETR, KP5, KP12.

c) Codieren Definitionen und Programmablauf.

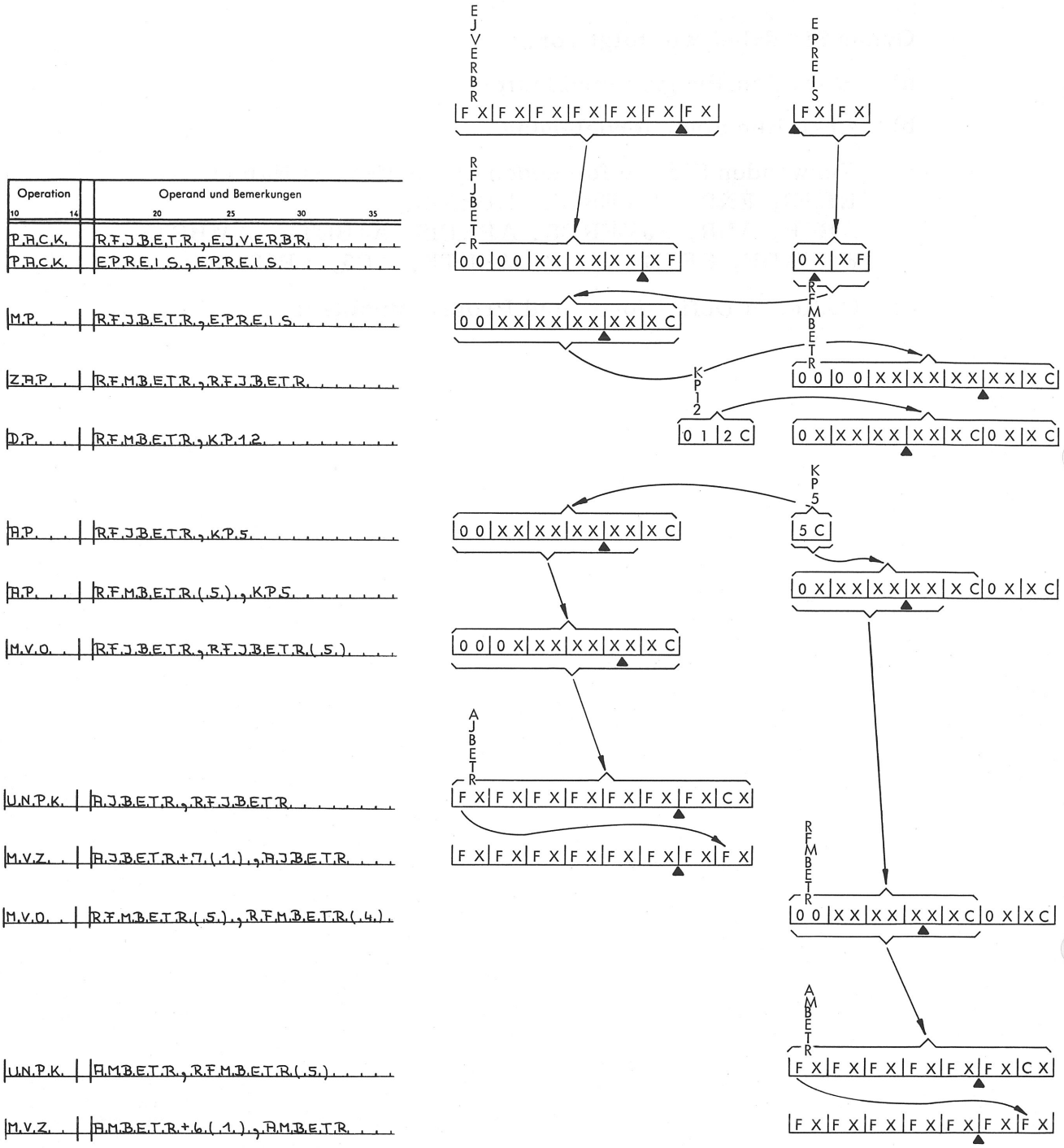

Operation
Operand und Bemerkungen
10 14 20 25 30 35
P.B.C.K. R.F.J.B.E.T.R.,E.J.V.E.R.B.R.
P.B.C.K. E.P.R.E.I.S.,E.P.R.E.I.S.
M.P. R.F.J.B.E.T.R.,E.P.R.E.I.S.
Z.A.P. R.F.M.B.E.T.R.,R.F.J.B.E.T.R.
D.P. R.F.M.B.E.T.R.,K.P.1.2.
A.P. R.F.J.B.E.T.R.,K.P.5.
A.P. R.F.M.B.E.T.R.(.5.),K.P.5.
M.V.O. R.F.J.B.E.T.R.,R.F.J.B.E.T.R.(.5.).
U.N.P.K. A.J.B.E.T.R.,R.F.J.B.E.T.R.
M.V.Z. A.J.B.E.T.R.+7.(.1.),A.J.B.E.T.R.
M.V.O. R.F.M.B.E.T.R.(.5.),R.F.M.B.E.T.R.(.4.).
U.N.P.K. A.M.B.E.T.R.,R.F.M.B.E.T.R.(.5.).
M.V.Z. A.M.B.E.T.R.+6.(.1.),A.M.B.E.T.R.
EJVERBR
EPREIS
RFJBETR
RFMBETR
KP12
KP5
AJBETR
AMBETR

Abb. 2

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) den Programmablaufplan zur Aufgabe 11

b) die Speicherbelegung

c) die gesamte Codierung.

Vergleichen Sie Ihre Lösung zur Aufgabe 11 mit der vorgegebenen.

Zum Programmablaufplan:

Wesentliche Unterschiede im Programmablaufplan ergeben sich, wenn man nach der Errechnung des Jahresbetrags nicht sofort den Monatsbetrag errechnet, sondern zunächst den Jahresbetrag rundet und in den Druckbereich bringt. In diesem Fall ist darauf zu achten, daß der Jahresbetrag vor dem Runden ins Rechenfeld für die Division gebracht wird.

Zur Codierung:

Die Abb. 2 zeigt die im Zusammenhang mit der Multiplikation und Division wesentlichen Instruktionen und Felder.

Gehen Sie bitte die einzelnen Instruktionen nacheinander durch und vergleichen Sie diese mit Ihrer Codierung.

Die Aufgabe 11 ist abgeschlossen.

Es ist nicht üblich, daß Beträge wie in der Aufgabe 11 geschrieben werden.

Die nachstehende Tabelle zeigt, wie Beträge normalerweise geschrieben werden:

Betrag in der Lochkarte:		Betrag auf der Liste:
54037122	→	540.371,22
00145736	→	1.457,36
00010021	→	100,21
00000050	→	0,50
00000001	→	0,01

Für die Aufbereitung der Daten in der oben gezeigten Form gibt es die Instruktion "Aufbereiten zum Drucken". - Diese lernen wir jetzt kennen.

Band 4, Seite 361 (ED)

Aufgabe 12

Entnehmen Sie bitte Ihren Arbeitsmitteln die Aufgabe 12 (2 Seiten).

Sehen Sie sich die Aufgabe 12 genau an. -

Je Teilenummer können mehrere Zugangskarten und mehrere Abgangskarten vorhanden sein, aber nur eine Bestandskarte.

Es gibt Bestände ohne Zugänge und Abgänge. Bei einem Bestand ohne Zugänge und Abgänge ist der alte Bestand gleich dem neuen Bestand (vgl. Teilenummer 00006 in der Liste).

Möglich sind auch Zugänge und Abgänge ohne Bestände.

In der Liste ist ein Beispiel für Zugänge und Abgänge ohne Bestände (hier handelt es sich um neu hinzugekommene Teilenummern).

Dies ist der Fall bei der Teilenummer ____(Z)____.

* * * * *

00007

Der neue Bestand wird je Teilenummer aus dem Bestand (alt) plus den Zugängen, abzüglich den Abgängen, errechnet.

Damit diese Rechnung für jede Teilenummer getrennt erfolgt, müssen wir eine _____ durchführen.

* * * * *

Gruppenkontrolle (Teilenummern-Vergleich)

Betrachten Sie bitte die Spalte "Nummer" in der Bestandsliste.

Die Nummer wird jeweils nur bei ___(WW)___ einer Gruppe gedruckt.

* * * * *

der ersten Karte

In der Lochkarte befinden sich Bestand, Zugang und Abgang in den _____ Spalten.

Auf der Liste werden sie jedoch in _____ Druckstellen angeschrieben.

* * * * *

gleichen

getrennten / verschiedenen

Auf der Liste sind der alte Bestand mit "_____" und der neue Bestand mit "_____" zu kennzeichnen.

* * * * *

ALT

NEU

Was benötigen wir, damit das Datum und die Mengen (Bestand, Zugang, Abgang) jeweils mit den entsprechenden Punkten geschrieben werden?

* * * * *

Sinngemäß: Entsprechende Schablonen

Nun zur Ausgabekarte:

Das Datum, das in die Spalten 7-10 gestanzt werden soll, kommt aus ____(WW)____.

* * * * *

der Vorlaufkarte

Welcher Bestand wird in die Spalten 11-15 gestanzt?

* * * * *

Der neue Bestand

Wir werden jetzt den Programmablaufplan erstellen. - Wir gehen dabei schrittweise vor.

Entnehmen Sie bitte Ihren Arbeitsmitteln ein Entwurfsblatt für Programmablaufpläne.

Die erste Karte unseres Kartenpakets ist die _____

* * * * *

Vorlaufkarte

Was muß nach dem Lesen dieser Karte erfolgen?

* * * * *

Wir müssen das Datum in den Stanzbereich übertragen.

Zeichnen Sie bitte die entsprechenden Schritte. - Beginnen Sie mit Block A1. (Vergessen Sie bitte nicht Start.)

* * * * *

Vergleichen Sie bitte Ihre Schritte mit denen der Abb. 1 (Rückseite!).

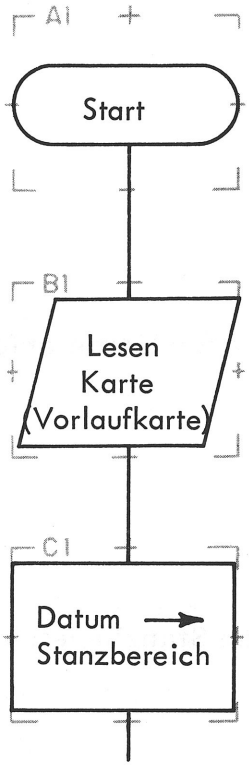

A1
Start
B1
Lesen
Karte
(Vorlaufkarte)
C1
Datum
Stanzbereich

Abb. 1

Nun müssen wir die nächste Karte lesen. - Diese Karte kann eine Bestandskarte oder eine Zugangskarte oder auch eine Abgangskarte sein.

Unabhängig davon, ob die erste Karte einer Gruppe eine Bestandskarte oder Zugangs- oder Abgangskarte ist, sind die Nummer und das Datum der Karte zu drucken und die Nummer in den Stanzbereich zu übertragen (vgl. Aufgabenstellung).

Außerdem können wir gleich die Menge (Bestand, Zugang oder Abgang) packen.

Die weitere Verarbeitung hängt davon ab, welche Kartenart vorliegt.

Blättern Sie bitte um.

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: _____ Seite: _____

Programmierer: _____ Datum: _____

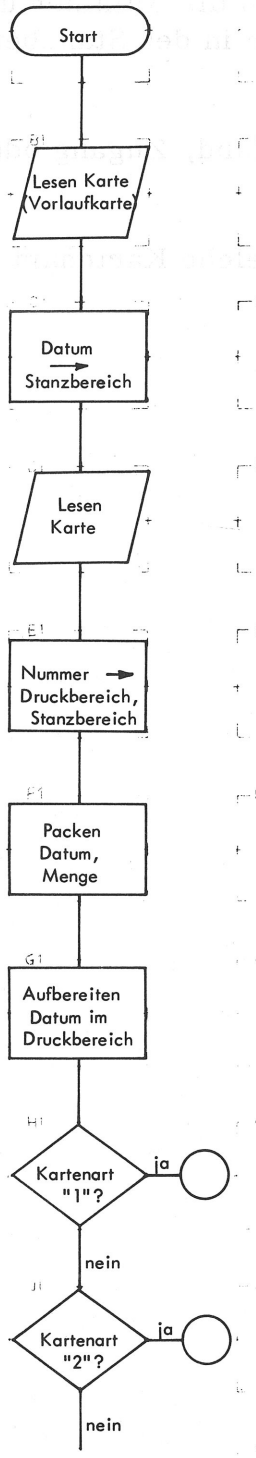


Abb. 2

Der Programmablaufplan in Abb. 2 ist um die entsprechenden Schritte erweitert.

Sehen Sie sich bitte die einzelnen Blöcke genau an.

Die Eintragung "Aufbereiten Datum im Druckbereich" (Block G1) verwenden wir im Gegensatz zu "Datum ⟶ Druckbereich" dann, wenn das Datum in eine Schablone einzugeben und nicht einfach zu übertragen ist.

Ergänzen Sie bitte Ihren bisherigen Programmablaufplan entsprechend Abb. 2.

Wir überlegen nun, was bei den verschiedenen Kartenarten zu tun ist.

① Bei einer Bestandskarte (KA"1") müssen wir:

a) Den Bestand in ein _____ addieren.

b) Den Bestand im Druckbereich in der Spalte "_____" aufbereiten.

c) Die Bezeichnung "_____" in den Druckbereich bringen.

d) _____

* * * * *

a) Den Bestand in ein Rechenfeld addieren.

b) Den Bestand im Druckbereich in der Spalte "Bestand" aufbereiten.

c) Die Bezeichnung "ALT" in den Druckbereich bringen.

d) Drucken

② Bei einer Zugangskarte müssen wir:

a) _____

b) _____

c) _____

* * * * *

a) Den Zugang in das Rechenfeld addieren.

b) Den Zugang im Druckbereich in der Spalte "Zugang" aufbereiten.

c) Drucken

③ Bei einer Abgangskarte müssen wir:

a) _____

b) _____

c) _____

* * * * *

a) Den Abgang in das Rechenfeld subtrahieren.

b) Den Abgang im Druckbereich in der Spalte "Abgang" aufbereiten.

c) Drucken

Betrachten Sie bitte Ihren Programmablaufplan.

Wenn wir im Block J1 als Antwort "nein" erhalten, haben wir es mit einer _____-Karte zu tun.

* * * * *

Abgangs (-Karte)

Erweitern Sie bitte Ihren Programmablaufplan um die Verarbeitung der Abgangskarte (vgl. ③ , oben).
(Fahren Sie bitte nach dem Block K1 mit A2 fort.)

* * * * *

Vergleichen Sie bitte die hinzugekommenen Schritte mit denen der Abb. 3 (Rückseite!).

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

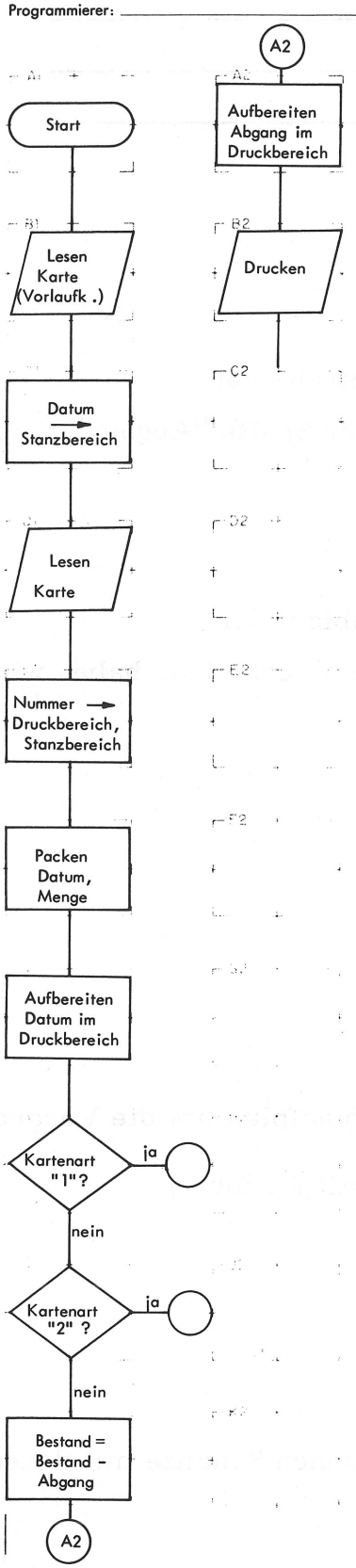


Abb. 3

Verbessern Sie Ihren Programmablaufplan soweit erforderlich.

Bevor wir uns überlegen, was nach dem Drucken erfolgen muß, nehmen wir uns die Verarbeitung der Bestandskarte und der Zugangskarte vor.

Zeichnen Sie bitte die für die Bestandskarte erforderlichen Verarbeitungsschritte (vgl. (1) , Seite 349). -

Beachten Sie bitte:

Beginnen Sie dabei mit Block A4. Ergänzen Sie die entsprechende bedingte Verzweigung.

* * * * *

Überprüfen Sie bitte die hinzugefügten Schritte anhand der Abb. 4 (Rückseite!).

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

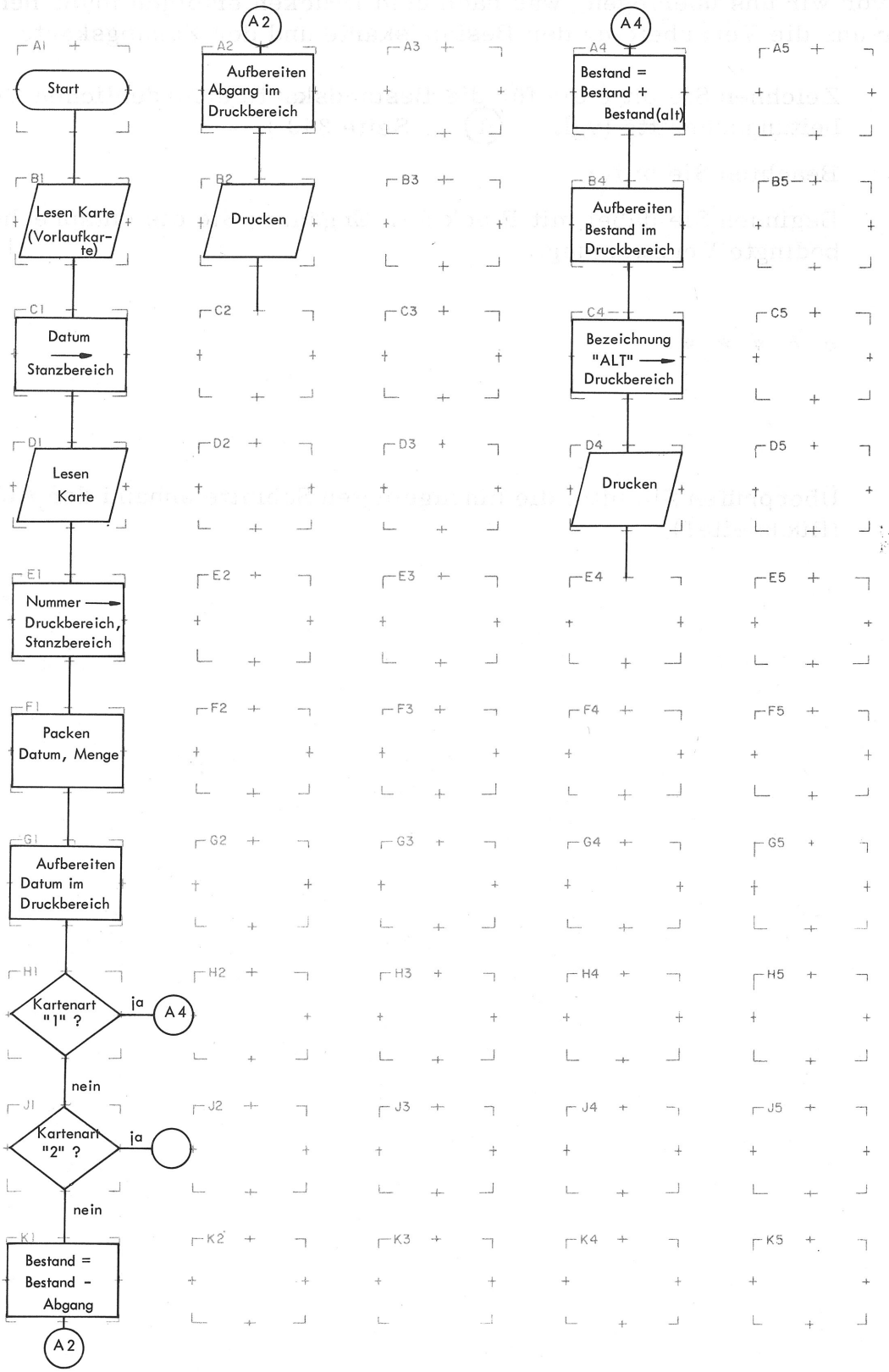


Abb. 4

Verbessern Sie Ihren Programmablaufplan soweit erforderlich.

Nun nehmen wir uns die Verarbeitung der Zugangskarte vor.

Zeichnen Sie die entsprechenden Schritte ab Block F4 (vgl. Seite 349). (2)

Vervollständigen Sie dabei die bedingte Verzweigung in J1.

* * * * *

Vergleichen Sie bitte die hinzugefügten Schritte mit denen der Abb. 5 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

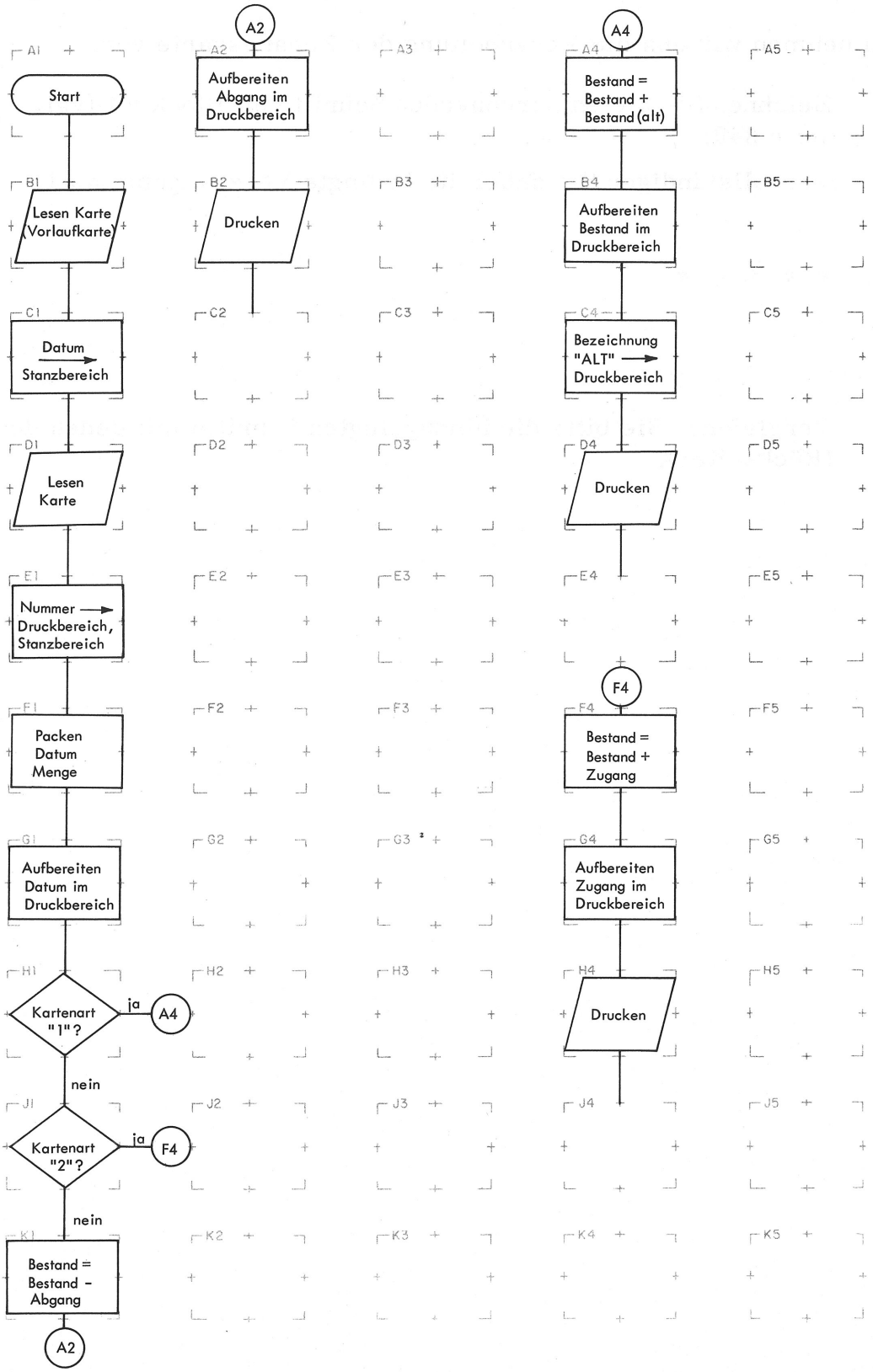


Abb. 5

Verbessern Sie Ihren Programmablaufplan soweit erforderlich.

Wir wiederholen und überprüfen unseren bisherigen Programmablaufplan.

Legen Sie bitte die Aufgabenstellung und den Programmablaufplan nebeneinander.

Wir beginnen im Programmablaufplan mit Schritt D1 und gehen davon aus, daß als erste Karte (nach der Vorlaufkarte) eine Bestandskarte gelesen wird.

Die Nummer dieser Karte wird in den Druckbereich und in den Stanzbereich übertragen (E1).

Anschließend werden das Datum und die Menge gepackt.

Danach wird das Datum zum Drucken aufbereitet (G1).

Die Frage nach Kartenart "1" ergibt "ja" und es wird zu Schritt A4 verzweigt.

Der (alte) Bestand wird zum bisherigen Bestand (bisheriger Bestand = 0) addiert und anschließend zum Drucken aufbereitet.

Der Bestand wird mit der Bezeichnung "ALT" versehen.

Danach wird gedruckt.

Soweit ist alles in Ordnung.

Nun nehmen wir an, daß als erste Karte nach der Vorlaufkarte eine Zugangskarte gelesen wurde.

Zunächst wird auch hier die Nummer in den Druckbereich und den Stanzbereich übertragen, das Datum und die Menge werden gepackt, und das Datum wird zum Drucken aufbereitet.

Die anschließende Frage nach Kartenart "1" ergibt "nein". Bei der zweiten Abfrage erhalten wir dann "ja", und es wird zu Schritt F4 verzweigt.

Der Zugang wird addiert und zum Drucken aufbereitet.

Danach wird gedruckt.

Soweit ist auch hier alles richtig.

Verfolgen Sie nun den Programmablauf unter der Voraussetzung, daß als erste Karte nach der Vorlaufkarte eine Abgangskarte gelesen wird.

Sie stellen fest, daß soweit auch hier alles in Ordnung ist.

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

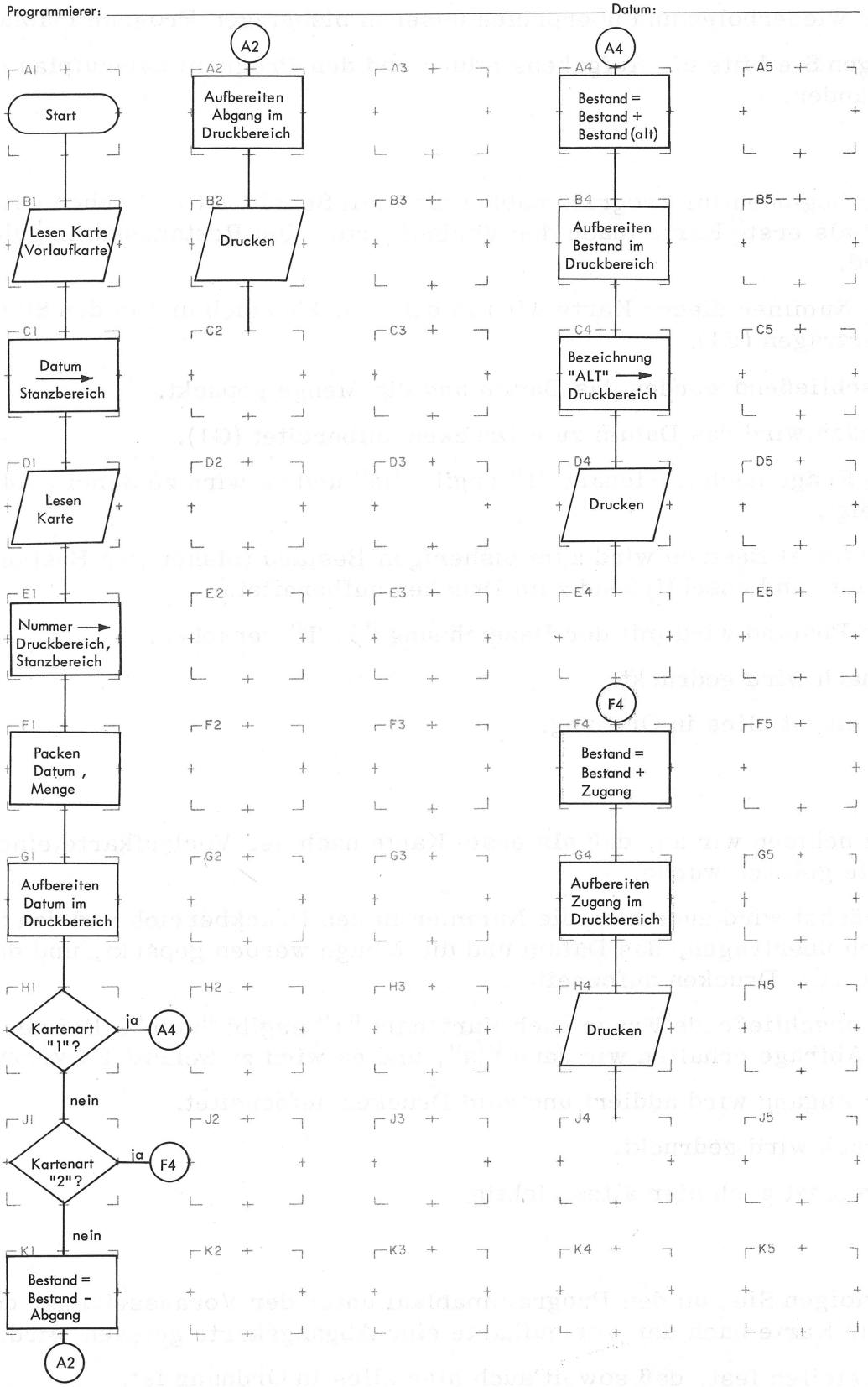


Abb. 5

Betrachten Sie bitte die ersten beiden Zeilen in der Bestandsliste. - Was würde in die zweite Zeile gedruckt, wenn wir nach dem Drucken der ersten Zeile (alter Bestand) im Programmablaufplan zum Block D1 verzweigen und die Zugangskarte wie dort gezeigt verarbeiten würden?

* * * * *

Sinngemäß:
Außer Datum und Zugang würden zusätzlich noch Nummer und Bestand (einschl. der Bezeichnung "ALT") gedruckt werden.

Wir müssen also darauf achten, daß von der vorangehenden Druckzeile nichts mitgedruckt wird. - Dies gilt für alle drei Kartenarten.

Wir müssen daher nach jedem Drucken in die entsprechenden Stellen _____ übertragen.

* * * * *

Blanks

In unserem Programmablaufplan haben wir drei Blöcke mit der Eintragung "Drucken". - An jeden dieser Blöcke wäre jetzt ein Block mit "Blanks ⟶ Druckbereich" anzufügen.

Wie können wir den Programmablaufplan vereinfachen und trotzdem das Gleiche erreichen?

* * * * *

Sinngemäß:
Indem wir gleich im Anschluß an die Blöcke C4 und G4 nach Schritt B2 verzweigen und an diesen den Block mit "Blanks ⟶ Druckbereich" anschließen.

Ändern Sie bitte Ihren Programmablaufplan entsprechend.

* * * * *

Vergleichen Sie Ihren Programmablaufplan mit dem der Abb. 6 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

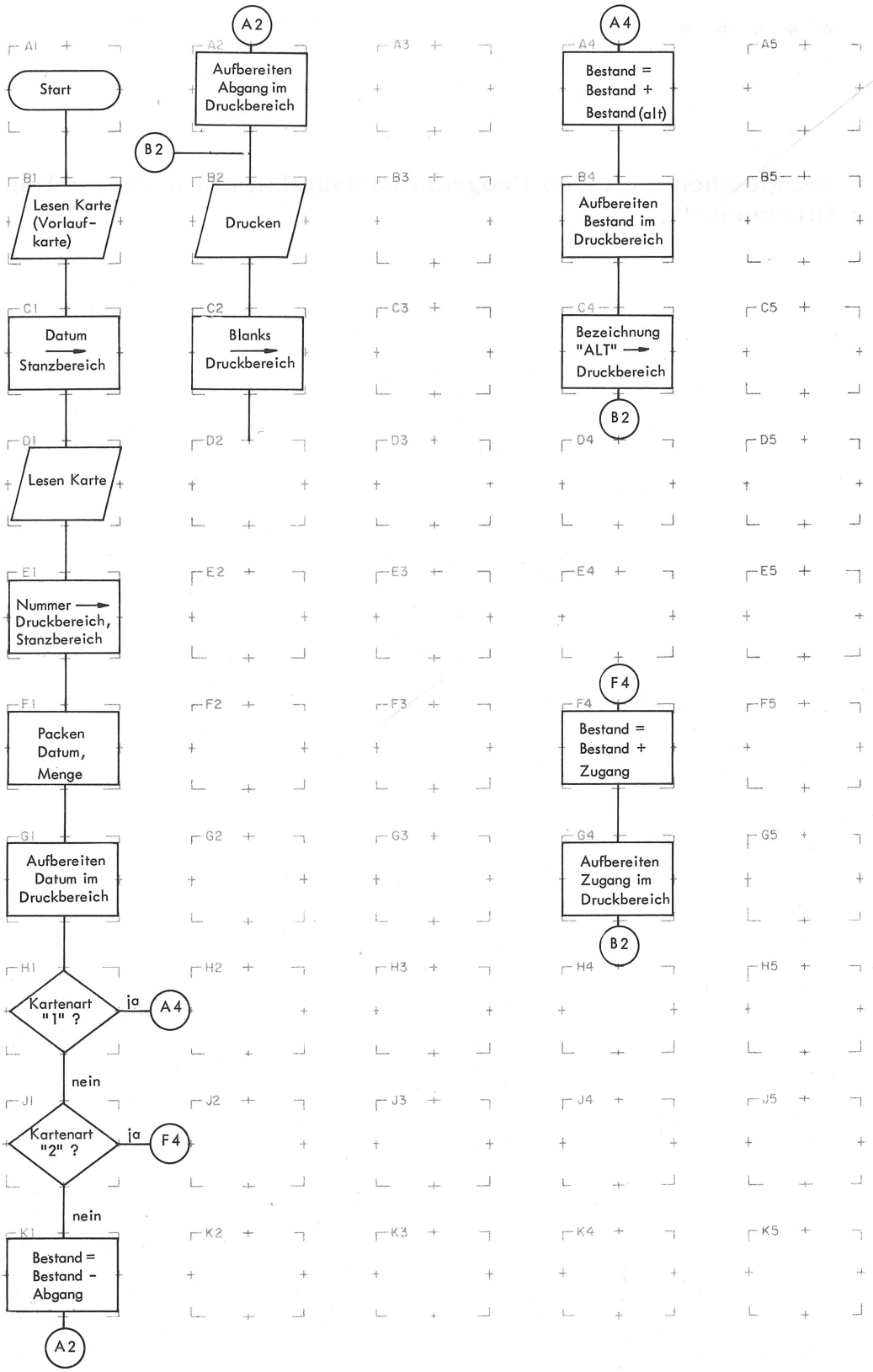


Abb. 6

Damit haben wir ab Schritt B2 wieder gemeinsame Verarbeitungsschritte, unabhängig davon, welche Kartenart vorliegt.

Nun müssen wir die nächste Karte einlesen und prüfen, ob sie zur (WW) gehört.

* * * * *

gleichen Gruppe (die gleiche Nummer hat wie die vorangehende Karte)

Zeichnen Sie bitte die hierfür erforderlichen Schritte in Ihren Programmablaufplan ein. (Fragen Sie nach Gruppenwechsel.)

* * * * *

Vergleichen Sie bitte die ergänzten Schritte mit denen in Abb. 7 (Rückseite!).

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

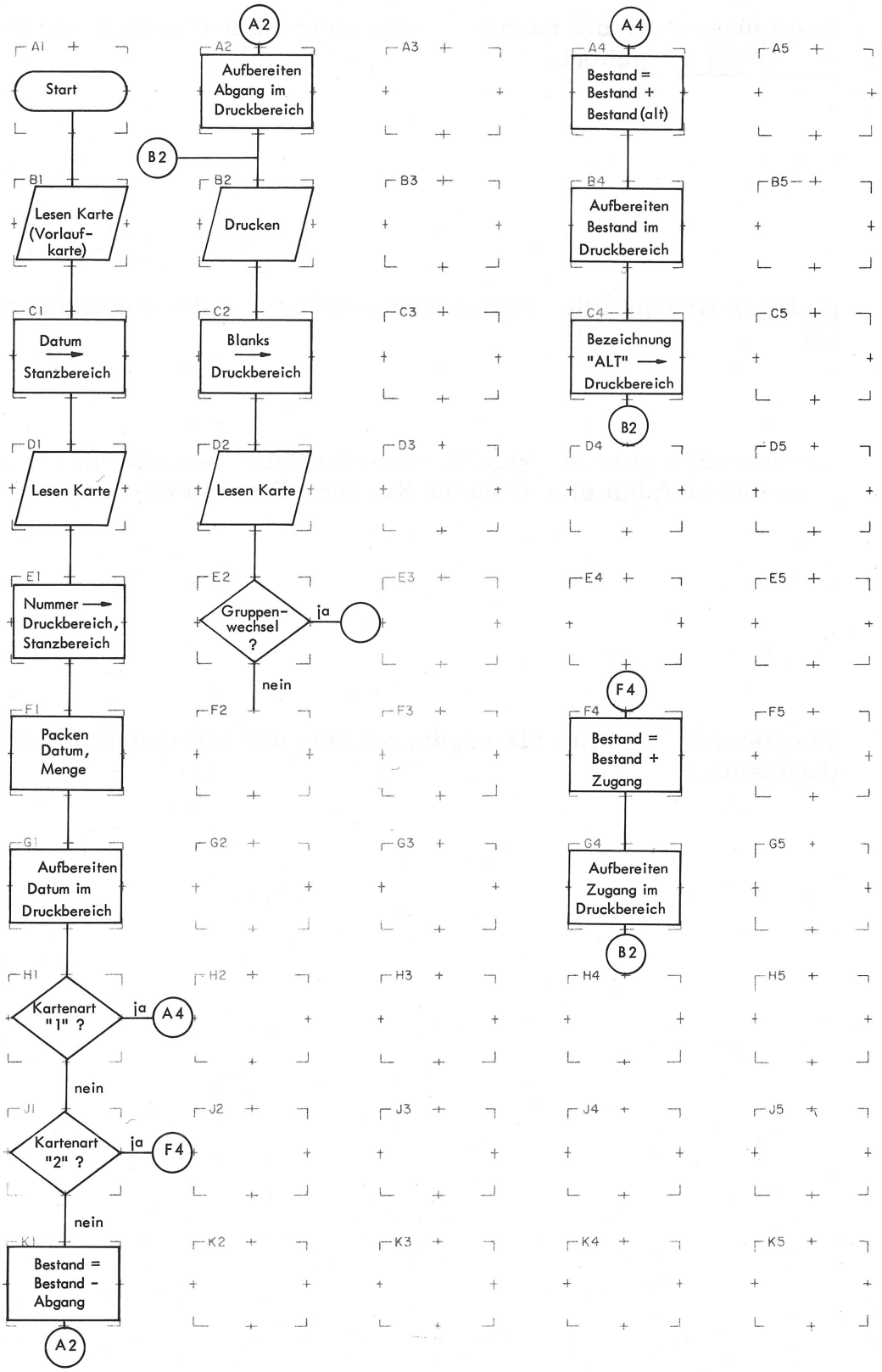


Abb. 7

Verbessern Sie Ihren Programmablaufplan soweit erforderlich.

Wir untersuchen zunächst, was zu tun ist, wenn kein Gruppenwechsel erfolgt.

Die gelesene Karte ist entweder eine _____ oder _____.

* * * * *

Zugangskarte

Abgangskarte

(Je Teilenummer gibt es nur eine Bestandskarte!)

Für beide Kartenarten gilt, daß das _____ zum Drucken aufbereitet werden muß. Die weitere Verarbeitung hängt dann von der Art der Karte ab.

* * * * *

Datum

Die Schritte hierfür sind schon in unserem bisherigen Programmablaufplan enthalten.

Wenn wir Gruppenwechsel "nein" erhalten, können wir zu Schritt _____ verzweigen.

* * * * *

F1

Erweitern Sie Ihren Programmablaufplan entsprechend.

* * * * *

Vergleichen Sie bitte die Ergänzungen mit dem Programmablaufplan der Abb. 8 (Rückseite!).

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

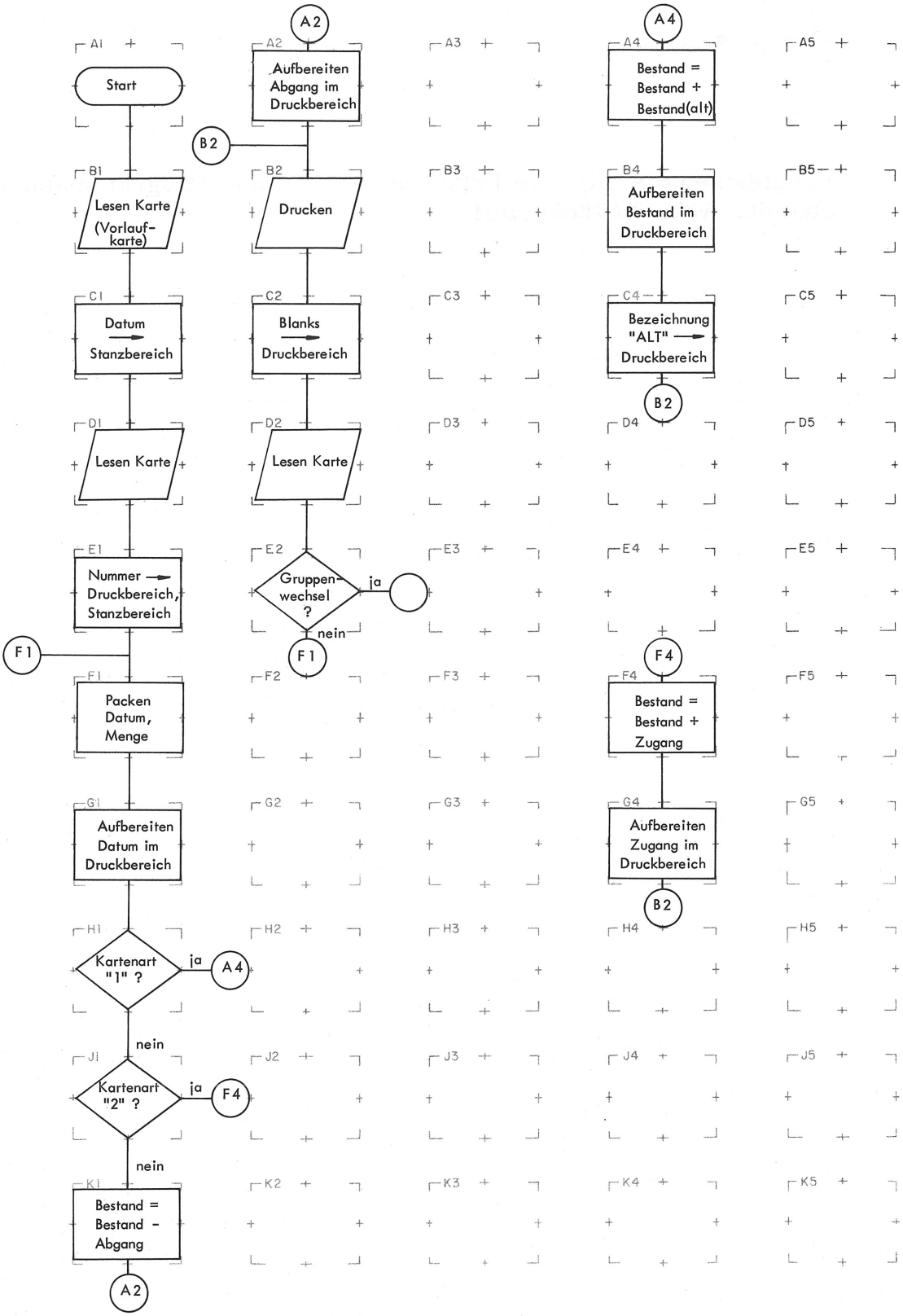


Abb. 8

Solange kein Gruppenwechsel erfolgt, wird immer nach F1 verzweigt, und die gleichen Schritte werden immer wieder ausgeführt.

Gemäß unserer Aufgabenstellung muß bei einem Gruppenwechsel zweierlei erfolgen:

a) (WW)

b) (WW)

* * * * *

Sinngemäß:

a) Der neue Bestand muß zusammen mit der Bezeichnung "NEU" gedruckt werden.

b) Außerdem muß eine Bestandskarte gestanzt werden.

Zeichnen Sie bitte die entsprechenden Schritte hierfür. - Beginnen Sie bei Block A5.

Ergänzen Sie bitte dabei die Abfrage nach Gruppenwechsel.

* * * * *

Überprüfen Sie die hinzugefügten Schritte anhand der Abb. 9 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

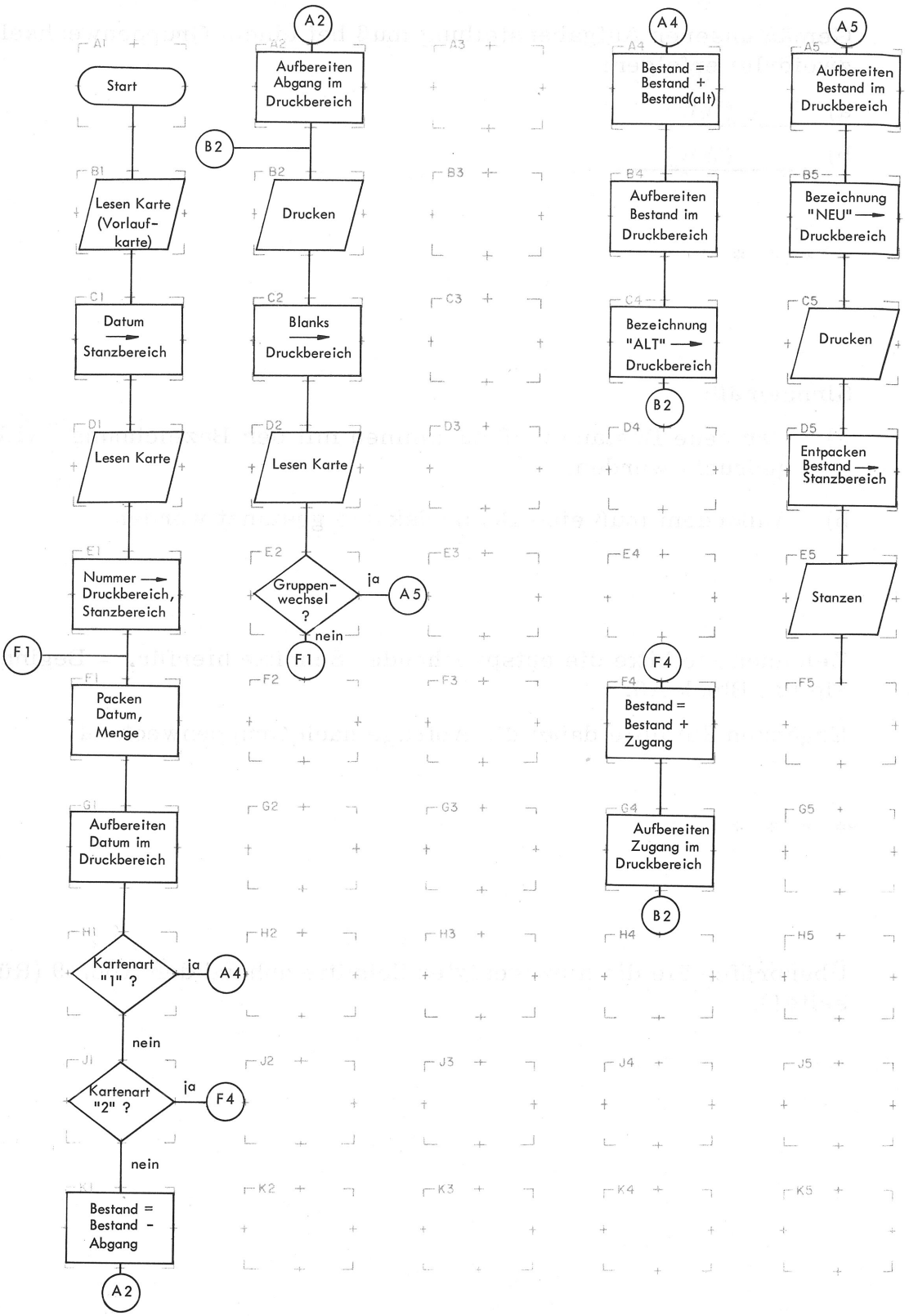


Abb. 9

Falls Sie im Anschluß an das Drucken Blanks in den Druckbereich gebracht haben, so ist das richtig. Aus Gründen, die Sie später sehen werden, wird der entsprechende Block jedoch nicht an dieser Stelle eingefügt.

Vielleicht haben Sie auch daran gedacht, vor dem Stanzen die KA"1" in den Stanzbereich zu übertragen. Dies ist hier weggelassen, weil wir bei der Codierung die "1" direkt in den Stanzbereich definieren werden.

Wir müssen jetzt die Karte verarbeiten, die den _____ bewirkt hat.

* * * * *

Gruppenwechsel

Bei der ersten Karte jeder Gruppe müssen wir wieder die Nummer in den Druckbereich und den Stanzbereich übertragen.

In unserem Programmablaufplan ist dies schon vorhanden.

Gemeint ist der Block _____.

* * * * *

E1

Wenn wir jedoch sofort nach Schritt E1 verzweigen, so wird die Menge der vorliegenden Karte zu dem Bestand der vorangehenden Gruppe addiert (bei einer Bestands- oder Zugangskarte) bzw. subtrahiert (bei einer Abgangskarte).

Das Rechenfeld Bestand muß also vorher _____ werden.

* * * * *

gelöscht

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

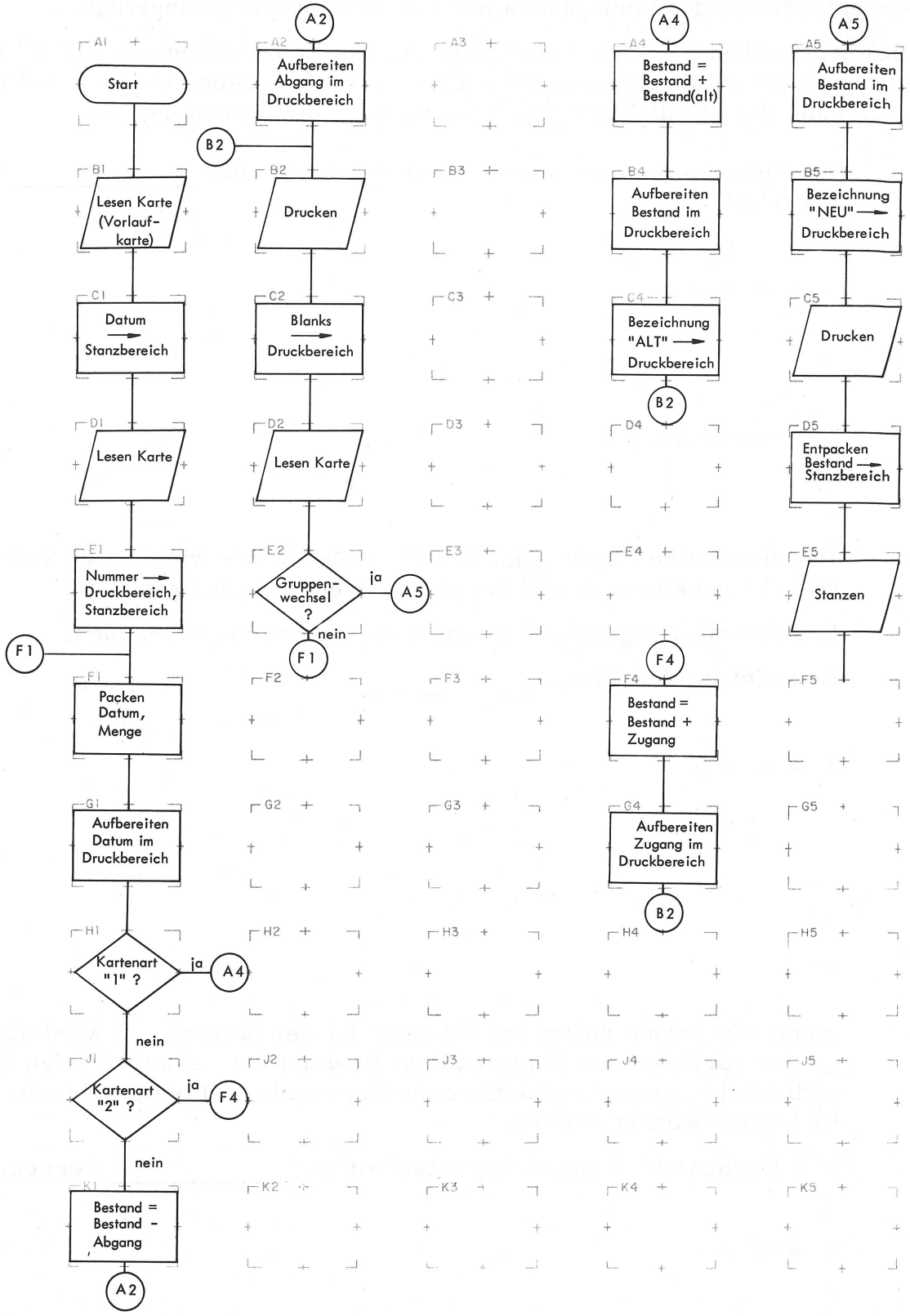


Abb. 9

Außerdem sorgen wir jetzt dafür, daß beim nächsten Drucken nicht der neue Bestand der vorangehenden Gruppe gedruckt wird.

Dies wäre der Fall, wenn die erste Karte der neuen Gruppe eine _____ oder _____ wäre.

* * * * *

Zugangskarte

Abgangskarte

Wir können also erst dann nach Schritt E1 verzweigen, wenn wir das Rechenfeld gelöscht und Blanks in den Druckbereich übertragen haben.

Erweitern Sie nun Ihren Programmablaufplan entsprechend.

* * * * *

Vergleichen Sie bitte Ihren ergänzten Programmablaufplan mit dem der Abb. 10 (Rückseite!).

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

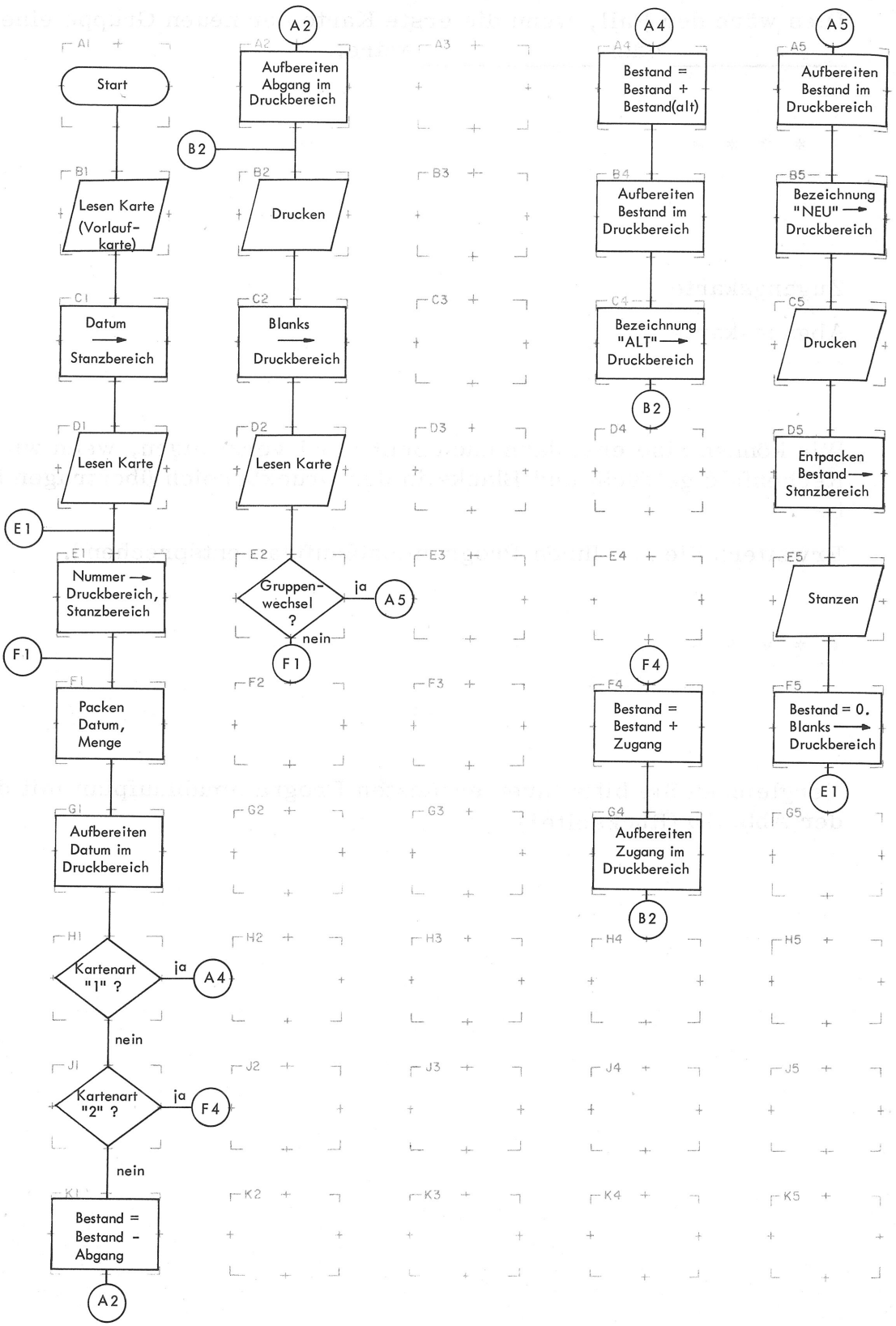


Abb. 10

Verbessern Sie Ihren Programmablaufplan soweit erforderlich.

Jetzt fehlt uns nur noch das Ende der Arbeit.

Was muß erfolgen, wenn wir im Block D2 die "ENDE"-Karte gelesen haben?

a) ____(WW)____

b) ____(WW)____

* * * * *

Sinngemäß:

a) Es muß noch der neue Bestand der letzten Gruppe gedruckt werden.

b) Außerdem ist noch die letzte Bestandskarte zu stanzen.

Wir schieben deshalb vor die Frage "Gruppenwechsel" die Frage nach der "ENDE"-Karte ein.

Wenn die "ENDE"-Karte gelesen wurde, benötigen wir die gleichen Schritte wie bei einem _____.

* * * * *

Gruppenwechsel

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

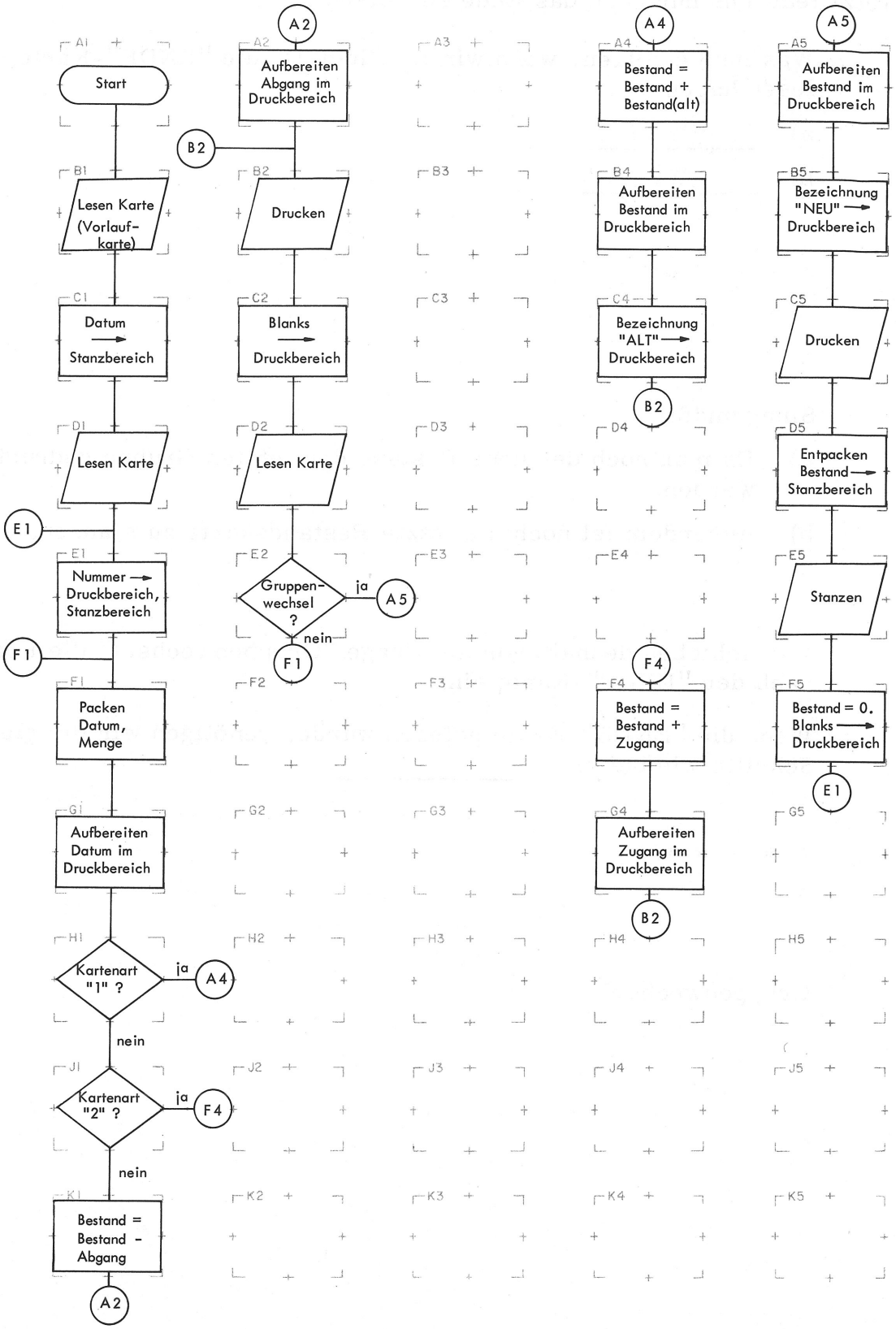


Abb. 10

Nach dem Stanzen darf es jedoch nicht mit den Schritten F5, E1 usw. weitergehen, sondern es muß zum "Ende der Arbeit" verzweigt werden.

Um aber nicht die gleichen Schritte wie nach einem Gruppenwechsel nochmals programmieren zu müssen, fragen wir nach dem Stanzen erneut, ob die "ENDE"-Karte vorliegt. Bei einer "Ja"-Antwort verzweigen wir dann zum "Ende der Arbeit".

Verbessern und erweitern Sie bitte Ihren Programmablaufplan um die beiden "ENDE"-Abfragen und das Ende der Arbeit.

Verwenden Sie für "Ende der Arbeit" den Block J5.

* * * * *

Ihr Programmablaufplan muß jetzt dem der Abb. 11 entsprechen (Rückseite!).

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

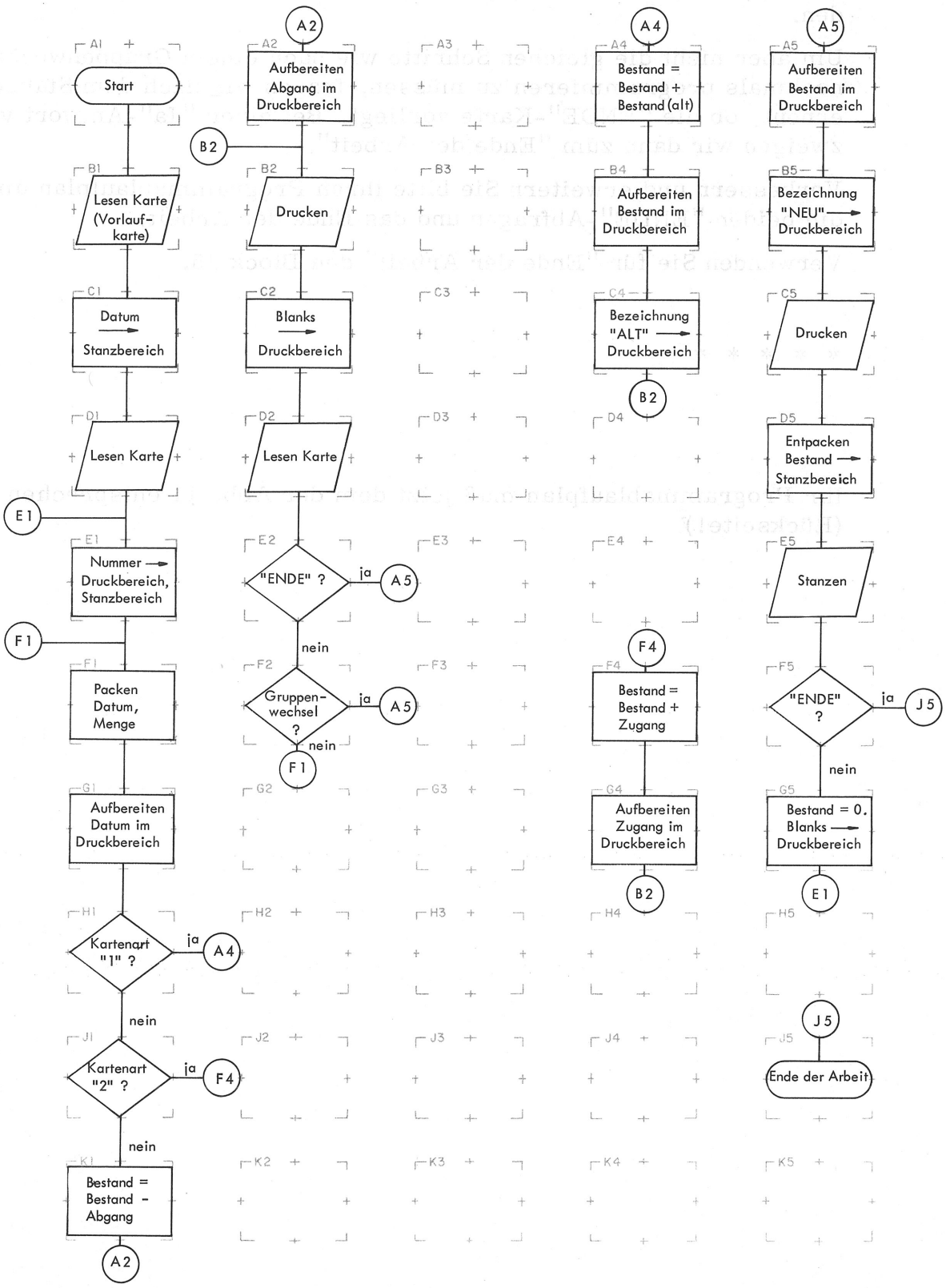


Abb. 11

Verbessern Sie Ihren Programmablaufplan soweit erforderlich.

Vielleicht haben Sie darüber nachgedacht, was geschehen würde, wenn wir nach dem Lesen der Karte (Schritt D2) nicht nach "ENDE" fragen würden.

Der weitere Ablauf wäre der:
Die Daten der Spalten 2-6 in der "ENDE"-Karte (NDEbb) würden mit der Nummer der vorangehenden Gruppe verglichen werden.

Dies würde zu einem _____ führen.

* * * * *

Gruppenwechsel

Anschließend würde zu Schritt A5 verzweigt. Das entspricht genau dem, was wir haben wollen. - Die Frage auf "ENDE" wäre somit an dieser Stelle überflüssig, jedoch nicht nach dem Stanzen (Schritt E5).

Wir sind bisher davon ausgegangen, daß zu Beginn einer Arbeit der Druckbereich und der Stanzbereich Blanks enthalten.

Aus verschiedenen Gründen, auf die in diesem Lehrgang nicht eingegangen werden kann, können jedoch alle mit DS reservierten Felder und Bereiche einen anderen Inhalt als Blanks haben.

Somit kann sich ergeben, daß z. B. beim Drucken außer den gewünschten Daten, die wir in den Druckbereich eingebracht haben, noch andere mitgedruckt werden. - Dies können wir verhindern, indem wir vor dem ersten Drucken den Druckbereich blanken. Das gleiche gilt analog für den Stanzbereich. - Das Blanken dieser Bereiche kann unterbleiben, wenn vor dem ersten Drucken bzw. Stanzen alle Stellen des Bereiches mit Ausgabedaten aufgefüllt werden.

Betrachten Sie bitte die Ausgabekarte und die Bestandsliste von Aufgabe 12.

Wählen Sie von den unten aufgeführten Antworten die richtige aus.

Wir müssen vor dem ersten Drucken und dem ersten Stanzen:

A Den Druckbereich und den Stanzbereich blanken

B Nur den Stanzbereich blanken

C Nur den Druckbereich blanken

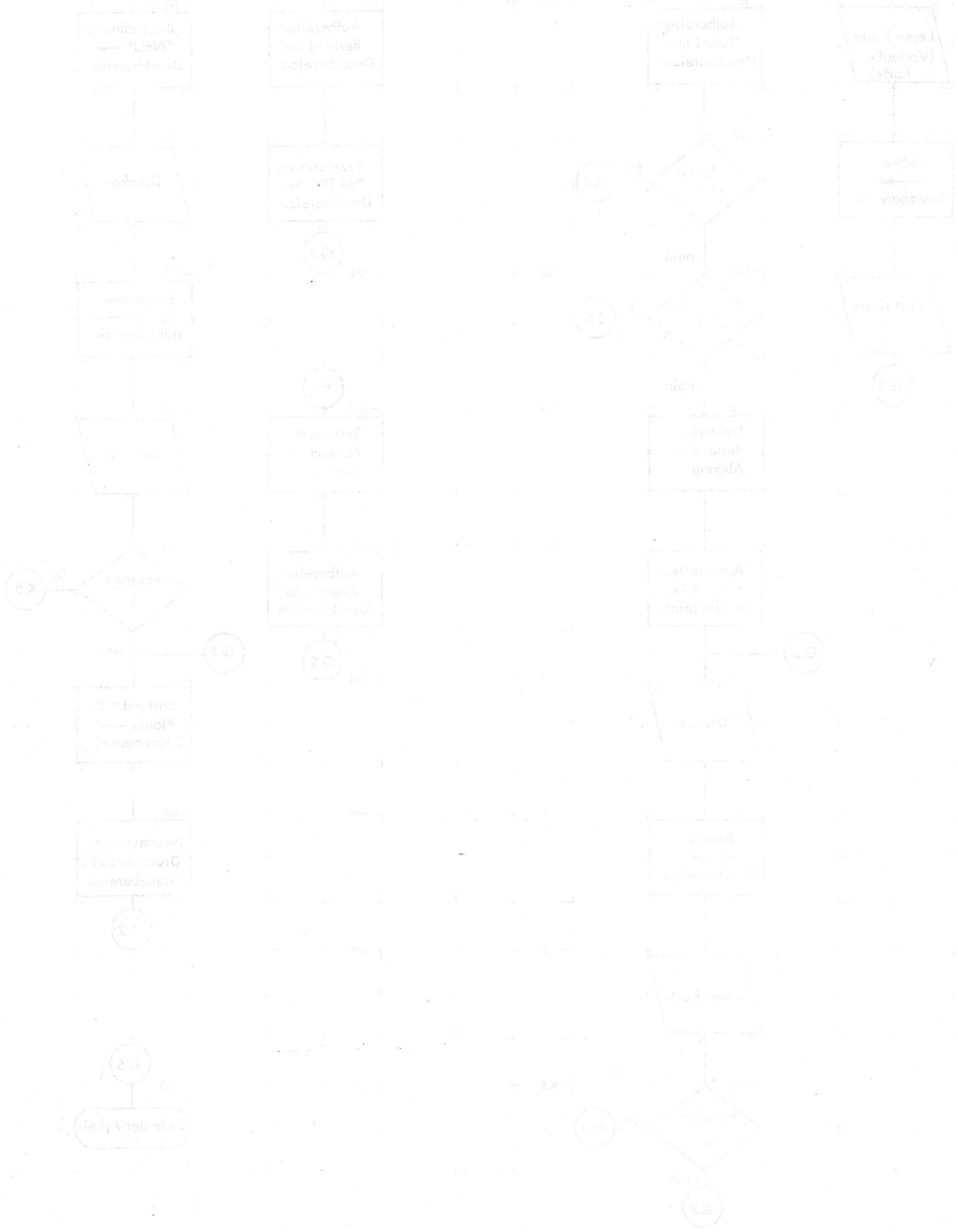
* * * * *

Richtig ist C

Der Stanzbereich muß nicht geblankt werden, da ja vor dem ersten Stanzen alle Felder durch das Programm einen entsprechenden Inhalt bekommen und die Felder lückenlos aufeinanderfolgen.

Aus den vorstehend genannten Gründen ist es auch günstiger, Rechenfelder mit DC (und Inhalt +0) und nicht mit DS zu definieren.

Unser Programmablaufplan ist so aufgebaut, daß die Menge der ersten Karte (der KA 1, 2 oder 3) zu dem bisherigen Inhalt des Rechenfeldes Bestand addiert bzw. von dem bisherigen Inhalt subtrahiert wird. - Das Rechenfeld muß also zu Beginn gelöscht sein (Nullen enthalten).



IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

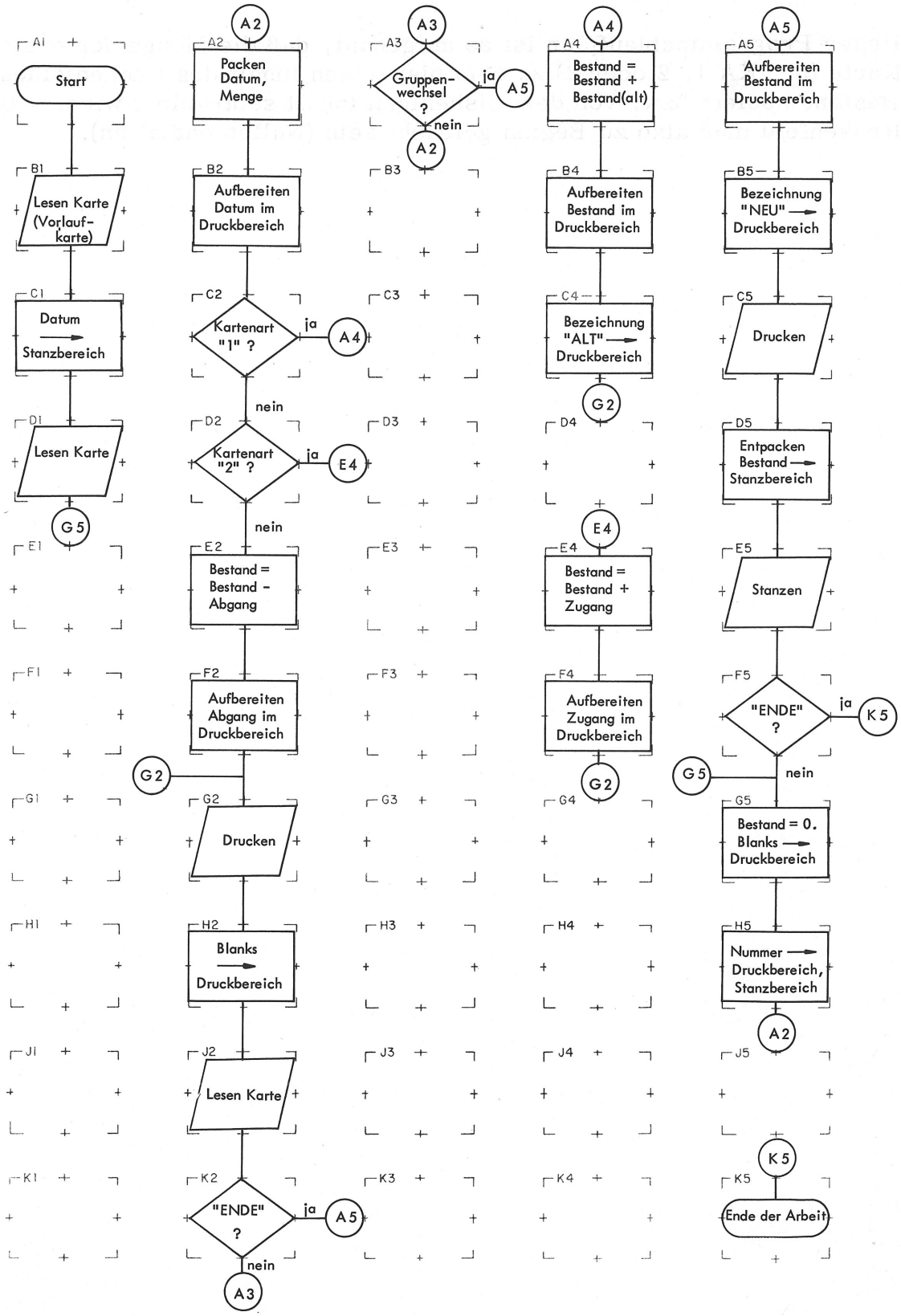


Abb. 12

Sehen Sie sich bitte die Abb. 12 genau an. -

Der Programmablaufplan in Abb. 12 enthält die gleichen Funktionen wie unser bisheriger Programmablaufplan.

Der gezeigte Programmablaufplan ist jedoch in sinnvollere Folgen eingeteilt.

Die mit Schritt A1 beginnende Folge kann man als Eröffnungsfolge bezeichnen.

Die Folgen, beginnend mit A2, A4 und E4, zeigen die Verarbeitung der verschiedenen Kartenarten.

Die Folge ab A5 zeigt den Programmablauf bei Gruppenwechsel.

Nach der Eröffnungsfolge wird nach Schritt _____ verzweigt. Dadurch ist sichergestellt, daß unser Rechenfeld zu Beginn gelöscht ist und der Druckbereich _____ enthält.

* * * * *

G5

Blanks

IBM **Entwurfsblatt für Programmablaufpläne**

Programm: Aufgabe 12 Nr.: Seite:

Programmierer: Datum:

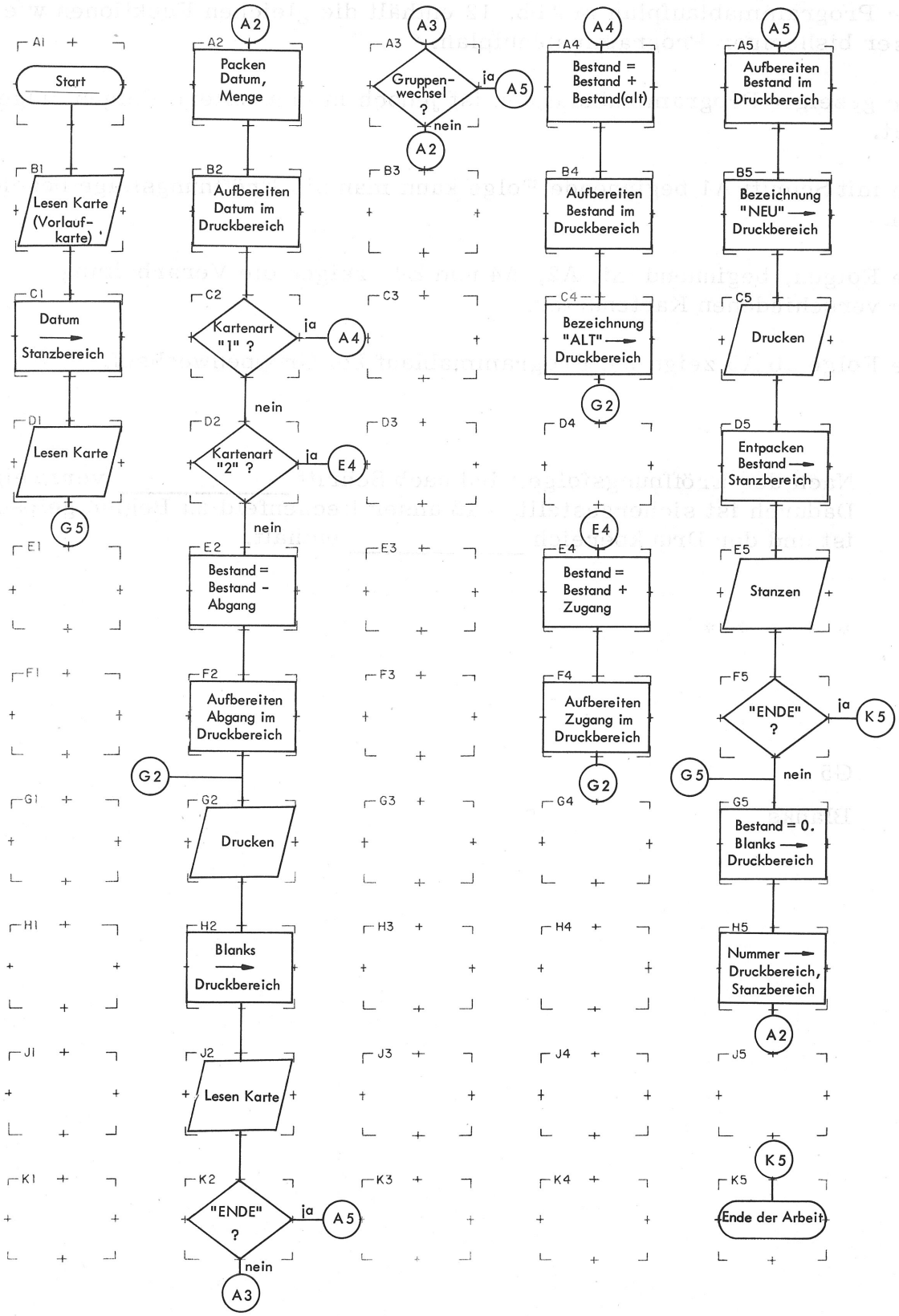


Abb. 12

Bevor wir in unserer Aufgabe 12 fortfahren, werden wir anhand von "Beispiel"-Daten den in Abb. 12 gezeigten Programmablaufplan auf seine Richtigkeit überprüfen. - "Wir spielen also ein bißchen DVS".

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) den in Abb. 12 gezeigten Programmablaufplan

b) die Übung 102 (2 Seiten).

Erstellen Sie die Bestandsliste aufgrund der gegebenen Datenkarten und des vorliegenden Programmablaufplans.

* * * * *

→ (4), Seite 389

④

Vergleichen Sie bitte Ihre Bestandsliste mit der der Aufgabe 12. -

Die beiden Listen müssen übereinstimmen.

Bevor wir zur Speicherbelegung übergehen, stellen wir noch einige Vorüberlegungen an.

Das Rechenfeld zur Errechnung des neuen Bestandes muß ____(Z)____ Bytes groß sein.

* * * * *

3

Wir benötigen ____(Z)____ Schablonen.

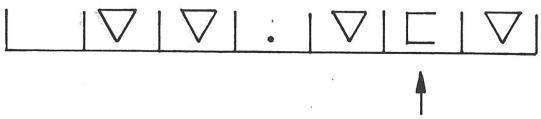
* * * * *

2 (eine für das Datum und eine gemeinsame für Zugang, Abgang und Bestand).

Wie muß die Schablone zur Aufnahme der Mengen (Zugang, Abgang und Bestand) aussehen, wenn sich folgende Druckbilder ergeben sollen:

01400+	soll gedruckt werden:	1.400
00400+	soll gedruckt werden:	400
00000+	soll gedruckt werden:	0

* * * * *



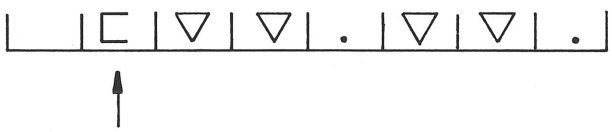
Damit beim nächsten Byte die Füllzeichen-einsteuerung unterbleibt (wichtig, wenn der Bestand 00000+ ist).

Betrachten Sie bitte die Spalte "Datum" in der Bestandsliste.

Wie muß die Schablone zur Aufnahme des Datums aussehen?

Beachten Sie bitte, daß z. B. das Datum "0606" gepackt | 0 0 | 6 0 | 6 + | ergibt.

* * * * *



Dieses Zeichen dient an dieser Stelle

a) zur Aufnahme der zusätzlichen Null, die wir durch das Pakken bekommen. (Diese wird anschließend durch das Füllzeichen ersetzt.)

b) zur Ausschaltung der Füllzeicheneinsteuerung ab dem nächsten Byte.

Nun überlegen wir uns, welche Konstanten wir benötigen.

Nehmen Sie bitte hierzu die Aufgabe und den Programmablaufplan zur Hand.

Zur Prüfung der Kartenart sind eine Konstante "____(Z)____" und eine Konstante "____(Z)____" erforderlich.

Zum Vergleich auf "ENDE" benötigen wir eine Konstante "_____".

Zur Kennzeichnung des alten und des neuen Bestandes benötigen wir die Konstanten "_____" und "_____".

* * * * *

1

2

ENDE

ALT

NEU

Die zu stanzenden Bestandskarten sind in der Spalte 1 mit der Kartenart "1" zu kennzeichnen.
Wie verfahren wir hier am zweckmäßigsten?

* * * * *

Sinngemäß:

Indem wir eine Konstante "1" in der entsprechenden Stelle des Stanzbereiches definieren.

Zum Blanken des Druckbereichs gehen wie wie folgt vor:

Wir definieren in die linke Stelle des Druckbereiches ein Blank. Dieses übertragen wir dann in die anschließenden Speicherstellen.

Wir wenden uns jetzt der Speicherbelegung und den Definitionen zu. -

Entnehmen Sie hierzu Ihren Arbeitsmitteln

a) die Speicherbelegung zur Aufgabe 12

b) die dazugehörenden Definitionen (2 Seiten).

Es sind noch das Rechenfeld und die Konstanten (einschließlich Schablonen) nachzutragen.

Vervollständigen Sie Speicherbelegung und Definitionen in der hier angegebenen Reihenfolge:

RFBEST, SCHDATUM (Schablone für das Datum),

SCHMENGE (Schablone für die Mengen),

KC1, KC2, KCENDE, KCALT, KCNEU.

* * * * *

Überprüfen Sie bitte Ihre Eintragungen anhand der Abb. 13 (Rückseite!).

IBM System / 360 Speicherbelegung

Aufgabe 12

		00	05	10	15	20	25	30	35	40	45	49
Eingabe-bereich		EBEREK	ENR	EDATUM	EMENGE							
	C											
	P, X											
Linie: 7												

		50	55	60	65	70	75	80	85	90	95	99
Stanz-bereich		SBER	SNR	SDATUM	SBESTZNG							
	C	1										
	P, X											
Linie: 7												

		00	05	10	15	20	25	30	35	40	45	49
Druck-bereich		DBER	DNR	DDATUM	DNUGANG	DABGANG	DBESTZNG					
	C	6										
	P, X											
Linie: 8												

		50	55	60	65	70	75	80	85	90	95	99
Forts. Druckbe-reich, Rechenfld. und Konstanten		DBEZ	RFBEST	SCHDATUM	SCHMENGE	KC1	KC2	KCENDE	KCALT	KCNEU		
	C			[▽▽.▽▽.	▽▽.▽[▽	1	2	ENDE	ALT	NEU		
	P, X		00000+									
Linie: 8												

IBM System / 360 Assembler Programmschema

Programm: Aufgabe 12

Programmierer | Datum

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
*,B,E,S,T,A,N,D	D,S	R,E,C,H,N,U,N,G,
*,D,E,F,I,N,I,T,I,O,N,E,N		
	S,T,A,R,T	7,Ø,1
E,B,E,R,	D,S,	Ø,C,L,1,5,
E,K,A,	D,S,	C,L,1,
E,N,R,	D,S,	C,L,5,
E,D,A,T,U,M,	D,S,	C,L,4,
E,M,E,N,G,E,	D,S,	C,L,5,
	D,S,	C,L,3,5,
S,B,E,R,	D,S,	Ø,C,L,1,5,
	D,C,	C,',1,',
S,N,R,	D,S,	C,L,5,
S,D,A,T,U,M,	D,S,	C,L,4,
S,B,E,S,T,A,N,D	D,S,	C,L,5,
	D,S,	C,L,3,5,
D,B,E,R,	D,S,	Ø,C,L,5,7,
	D,C,	C,',',
	D,S,	C,L,1,3,
D,N,R,	D,S,	C,L,5,
	D,S,	C,L,2,

IBM System / 360 Assembler Programmschema

Programm: Aufgabe 12

Programmierer | Datum

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
D,D,A,T,U,M,	D,S,	C,L,6,
	D,S,	C,L,4,
D,Z,U,G,A,N,G,	D,S,	C,L,6,
	D,S,	C,L,2,
D,A,B,G,A,N,G,	D,S,	C,L,6,
	D,S,	C,L,2,
D,B,E,S,T,A,N,D	D,S,	C,L,6,
	D,S,	C,L,1,
D,B,E,Z,	D,S,	C,L,3,
R,F,B,E,S,T,	D,C,	P,L,3,',Ø,',
S,C,H,D,A,T,U,M	D,C,	C,L,8,', [▽▽.▽.▽.',
S,C,H,M,E,N,G,E	D,C,	C,L,7,', ▽▽.▽[▽,',
K,C,1,	D,C,	C,',1,',
K,C,2,	D,C,	C,',2,',
K,C,E,N,D,E,	D,C,	C,',E,N,D,E,',
K,C,A,L,T,	D,C,	C,',A,L,T,',
K,C,N,E,U,	D,C,	C,',N,E,U,',

Abb. 13

Machen Sie sich mit der gesamten Speicherbelegung und den dazugehörenden Definitionen in Abb. 13 vertraut. Beachten Sie dabei die Konstanten "1" und "6" im Stanz- und Druckbereich.

In Abb. 13 ist RFBEST als Konstante mit dem Inhalt null definiert. Wir könnten RFBEST hier auch mit "DS CL3" definieren, da das Feld vor dem ersten Rechnen gelöscht wird (vgl. Block G5). Wir werden das Löschen des Rechenfeldes jedoch so codieren, daß wir uns auf die in Abb. 13 gezeigte Definition beziehen.

Verbessern Sie bitte Speicherbelegung und Definitionen soweit erforderlich.

Wir kommen nun zur Codierung des Programmablaufs.

Wir gehen dabei so vor, daß wir mit der Folge A1 beginnen und danach die Folgen A2, A3, A4, E4, A5, K5 codieren. Selbstverständlich könnten wir nach der Codierung des Schrittes D1 mit dem Schritt G5 fortfahren (ohne Verzweigung).

Günstiger ist es jedoch, in der o. a. systematischen Weise vorzugehen. Wir haben dabei einen besseren Überblick über das, was schon codiert ist und was noch zu codieren ist.

Codieren Sie bitte zunächst nur die linke Folge.

Verwenden Sie als Überschrift "Programmablauf".

Beginnen Sie bitte auf einem neuen Programmschema.

* * * * *

Vergleichen Sie bitte Ihre Codierung mit der Abb. 14 (Rückseite!).

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 12	Lochhinweise		Problem A12 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 013 (76–77) von ____

Anweisung				Karte Nr.
Name (1–8)		Operation (10–14)	Operand und Bemerkungen (20–70)	(78–80)
*. PROGRAMM		MABLAUF.		010
		GET.	KARTE,EBER	020
		MVC.	SDATUM,EDATUM	030
		GET.	KARTE,EBER	040
		B.	G5.	050
				060
				070
				080
				090
				100
				110
				120
				130
				140
				150
				160
				170
				180
				190
				200

Abb. 14

Verbessern Sie Ihre Codierung soweit erforderlich.

Bevor wir in der Codierung fortfahren, stellen wir noch einige Vorüberlegungen an.

Nehmen Sie bitte Ihre Speicherbelegung zur Hand.

Das Feld DDATUM ist ____(Z)____ Bytes groß. Die Schablone SCHDATUM umfaßt ____(Z)____ Bytes. Die Instruktion zur Übertragung der Schablone nach DDATUM muß also lauten: _____

* * * * *

6

8

Operation	Operand und Bemerkungen
10 14	20 25 30 35
MVC,	DDATUM-2(8),SCHDATUM

Nun überlegen wir, wie die anschließende "ED-Instruktion" aussehen muß.

Wenn wir z. B. das Datum "0606" in EDATUM packen, erhalten wir |0 0|0 0|6 0|6 +| .

Von dem gepackten Datum sind in die Schablone die ____(Z)____ (rechten, linken) _____ Bytes einzugeben.

* * * * *

3

rechten

Die erforderliche Instruktion hierfür ist: _____

* * * * *

Operation	Operand und Bemerkungen
10 14	20 25 30 35
ED	DDATUM-2(7),EDATUM+1

Bei einer expliziten Länge "8" erhielten wir als Ergebnis 06.06 (und nicht 06.06.), da durch das positive Vorzeichen die Füllzeicheneinsteuerung ab dem nächsten Byte (dem ".") wieder wirksam ist.

Sollten Sie als zweiten Operanden nur "EDATUM" angegeben haben, so ist das falsch. In diesem Fall würde versucht, in die Schablone 7 Ziffern einzugeben. Die Schablone kann aber nur 5 Ziffern aufnehmen.

Betrachten Sie bitte den Block H2 des Programmablaufplans. -

Wir haben in die linke Stelle des Druckbereiches ein "Blank" definiert. Zum Blanken des Druckbereiches übertragen wir dieses Blank in alle anschließenden Druckstellen.

Wie lautet die entsprechende Instruktion?

* * * * *

Operation	Operand und Bemerkungen
10 14	20 25 30
MVC	DBER+1(56),DBER

Zunächst wird das Blank von DBER nach DBER + 1 übertragen, anschließend wird von DBER + 1 nach DBER + 2 übertragen usw.

Was würde geschehen, wenn anstelle der expliziten Länge 56 mit der impliziten Länge 57 von DBER gearbeitet würde? (Nehmen Sie bitte Ihre Speicherbelegung zu Hilfe.)

* * * * *

Sinngemäß:

Es würde ein Blank zuviel übertragen. Dieses käme in das linke Byte von RFBEST. Die Folge davon wäre, daß wir z. B. bei einer folgenden "AP"- oder "SP"-Instruktion aufgrund dieses Blanks in RFBEST ein falsches Rechenergebnis bekämen. (Blank = hexadezimal 40.)

Erweitern Sie nun Ihre Codierung durch die Folgen A2 - A3.

* * * * *

Überprüfen Sie bitte Ihre Codierung anhand der Abb. 15 (Rückseite!).

Problem A12 (73–75)

Seite 013 (76–77) von 5

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
A2	PACK	EDATUM,EDATUM
	PACK	EMENGE,EMENGE
	MVC	DDATUM-2(8),SCHDATUM
	ED	DDATUM-2(7),EDATUM+1
	CLC	EKA,KC1 oder KC1,EKA
	BE	A4
	CLC	EKA,KC2
	BE	E4
	SP	RFBEST,EMENGE+2(3)
	MVC	DABGANG-1(7),SCHMENGE
	ED	DABGANG-1(7),EMENGE+2
G2	PUT	LISTE,DBER
	MVC	DBER+1(56),DBER
	GET	KARTE,EBER
	CLC	EBER(4),KCENDE

Problem A12 (73–75)

Seite 014 (76–77) von 5

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
	BE	A5
	CLC	ENR,SNR
	BNE	A5
	B	A2

Abb. 15

Die Codierung der Abfrage zu Block A3 könnte auch lauten:

Operation	Operand und Bemerkung
10 14	20 25
B,N,E, . .	A,5, ...
B,E, . .	A,2, ...

oder:

B,E, . .	A,2, ...
B,N,E, . .	A,5, ...

oder:

B,E, . .	A,2, ...
B, ...	A,5, ...

Verbessern Sie Ihre Codierung soweit erforderlich.

Codieren Sie nun den restlichen Programmablauf bis zum Schluß.

Beachten Sie bitte:
Die Instruktion zum Stanzen lautet:

P,U,T, . .	S,T,A,N,Z,,S,B,E,R, . .

Das Rechenfeld RFBEST löschen Sie bitte durch Subtraktion von sich selbst.

* * * * *

Vergleichen Sie bitte Ihre Codierung Instruktion für Instruktion mit der in Abb. 16 gezeigten (Rückseite!).

Problem A12 (73–75)

Seite 014 (76–77) von 5

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
A4	A.P.	R.F.B.E.S.T.,E.M.E.N.G.E.+2.(3.)
	M.V.C.	D.B.E.S.T.A.N.D.-1.(7.),S.C.H.M.E.N.G.E.
	E.D.	D.B.E.S.T.A.N.D.-1.(7.),E.M.E.N.G.E.+2.
	M.V.C.	D.B.E.Z.,K.C.A.L.T.
	B.	G.2.
E4	A.P.	R.F.B.E.S.T.,E.M.E.N.G.E.+2.(3.)
	M.V.C.	D.Z.U.G.A.N.G.-1.(7.),S.C.H.M.E.N.G.E.
	E.D.	D.Z.U.G.A.N.G.-1.(7.),E.M.E.N.G.E.+2.
	B.	G.2.
A5	M.V.C.	D.B.E.S.T.A.N.D.-1.(7.),S.C.H.M.E.N.G.E.
	E.D.	D.B.E.S.T.A.N.D.-1.(7.),R.F.B.E.S.T.
	M.V.C.	D.B.E.Z.,K.C.N.E.U.
	P.U.T.	L.I.S.T.E.,D.B.E.R.
	U.N.P.K.	S.B.E.S.T.A.N.D.,R.F.B.E.S.T.
	P.U.T.	S.T.A.N.Z.,S.B.E.R.
	C.L.C.	E.B.E.R.(4.),K.C.E.N.D.E.

Problem A12 (73–75)

Seite 015 (76–77) von 5

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
	B.E.	K.5.
G.5	S.P.	R.F.B.E.S.T.,R.F.B.E.S.T.
	M.V.C.	D.B.E.R.+1.(5.6.),D.B.E.R.
	M.V.C.	D.N.R.,E.N.R.
	M.V.C.	S.N.R.,E.N.R.
	B.	A.2.
K.5	Ende der Arbeit	
	E.N.D.	

Abb. 16

Verbessern Sie bitte Ihre Codierung soweit erforderlich.

Die Aufgabe 12 ist damit gelöst. Wir werden jedoch noch einige Überlegungen zu dieser Aufgabe anstellen, nachdem wir unser Wissen erweitert haben.

Wir lernen nun je eine weitere Übertragungs- und Vergleichsoperation kennen.



Band 4, Seite 50 (MVI)

Seite 98 (CLI)

Nehmen Sie bitte den Programmablaufplan zu Aufgabe 12 zur Hand und betrachten Sie die Blöcke C2 und D2.

In unserer Codierung haben wir die Prüfung auf Kartenart "1" oder "2" durch einen Vergleich (CLC) mit einer Konstanten "1" bzw. "2" vorgenommen.

Wie können wir dies nun anders programmieren?
Schreiben Sie bitte die Instruktionen zu den Blöcken C2 und D2.

* * * * *

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
	C.L.1	E.K.A.,C.'1'
	B.E.	A.4
	C.L.1	E.K.A.,C.'2'
	B.E.	E.4

Die Konstanten KC1 und KC2 (s. Codierung zu Aufgabe 12) würden bei dieser Lösung nicht mehr benötigt.

Wir erweitern unser Wissen zur Konzeption des Systems /360.

Zu diesem Zweck wenden wir uns jetzt den Kapiteln "Dualdaten" und "Adressierung des Hauptspeichers" zu.

Band 2, Seite 139 (Kapitel IV)
Seite 185 (Kapitel V)
Band 3, Seite 275 (USING)
Band 4, Seite 432 (BALR)

Sie wissen nun, wie Hauptspeicheradressen in Maschineninstruktionen dargestellt werden und was wir entsprechend dieser Darstellung bei der Programmierung beachten müssen:

Alle bisher gelösten Aufgaben müßten durch USING- und BALR-Anweisungen erweitert werden. Der Einfachheit halber beschränken wir uns jedoch auf die letzte gelöste Aufgabe.

Nehmen Sie bitte Ihre Codierung der Aufgabe 12 zur Hand. -

Wie ist diese Codierung abzuändern, wenn für das Programm als Basisregister das Register 8 zu verwenden und der augenblickliche Stand des Zuordnungszählers als Basisadresse anzunehmen ist?

Benutzen Sie bitte hierzu ein Programmschema.

* * * * *

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkunge (20 – 25)
*, BESTAND	SRECHN	UNG, ...
*, DEFINIT	IONEN	. . . . .
. . . . .	START	7Ø,1 . . . . .
. . . . .	BALR	8,,Ø . . . . .
. . . . .	USING	*,,8 . . . . .
. . . . .	. : .	. . . . .
. . . . .	. . .	. . . . .

Falsch ist, wenn Sie BALR und USING zwischen Definitionen und Programmablauf eingeschoben haben.

Warum?

* * * * *

Sinngemäß:

Weil die Basisadresse dann größer wäre, als die Adressen der davorliegenden Felder (von EBER bis KCNEU). - Dies darf jedoch nicht sein, da es ja keine negativen relativen Adressen gibt.

IBM System / 360 Assembler
Programmschema

Programm		Lochhinweise		Problem (73 75)
		Zeichen		
Programmierer	Datum	Lochung		Seite (76 77) von ____

Anweisung				Karte Nr.
Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (16 – 71)		(78 – 80)
*.BESTAND	SREICHN	U.N.G.		010
*.DEFINITIONEN				020
	START	7Ø1		030
	BALR	8,Ø		040
	USING	*,8		050
	B	B1		060
EBER	DS	ØCL15		070
	:			080
	:			090
	:			100
	:			110
KCNEU	DC	C'NEU'		120
*.PROGRAMMABLAUF				130
B1	GET	KARTE,EBER		140
	MVC	SDATUM,EDATUM		150
	:			160
	:			170
				180
				190
				200

Abb. 17

Einen Reinfall erleben wir jedoch auch, wenn wir n u r BALR und USING nach START einfügen.

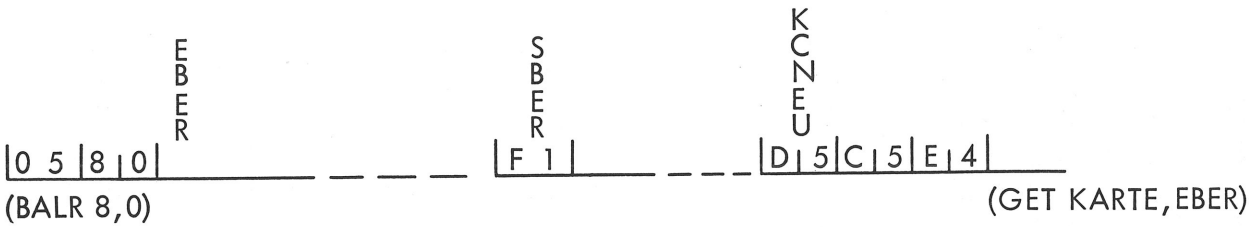
Überlegen Sie bitte, was bei der Ausführung des Programms geschehen würde.

* * * * *

Sinngemäß:

Nach Ausführung der BALR-Instruktion würde der Programmablauf unterbrochen, da keine weiteren Instruktionen folgen.

Das Programm würde wie folgt in aufeinanderfolgende Speicherplätze geladen (vereinfachte Darstellung).



Zwischen der ersten Instruktion des Programms (BALR) und der zweiten (GET) liegen die Bereiche etc. - Wenn nach BALR als nächste Instruktion die Instruktion GET auszuführen ist, so muß die GET-Instruktion auch im Anschluß an BALR gespeichert sein bzw. nach BALR muß zu der GET-Instruktion verzweigt werden.

Wir müssen also nach der USING-Anweisung noch eine Verzweigung einfügen und der GET-Instruktion einen Namen geben.

Die richtige Codierung ersehen Sie aus Abb. 17.

Ergänzen Sie bitte Ihre Codierung der Aufgabe 12 entsprechend Abb. 17. Fügen Sie die drei hinzugekommenen Zeilen am Fuße von Seite 01 hinzu und numerieren Sie diese mit 031, 032 und 033.

Hätten wir bei der Aufgabe 12 zuerst den Programmablauf codiert und dann die Definitionen geschrieben, wäre die Verzweigungsinstruktion nicht erforderlich gewesen. In diesem Falle wären ja alle Instruktionen vor den Bereichen etc. gespeichert worden.

Vielleicht haben Sie bemerkt, daß aufgrund der hinzugefügten Instruktionen der EBER nun nicht mehr bei der in START angegebenen Adresse (701) beginnt. Die Instruktionen BALR und B belegen zusammen 6 Bytes (2 und 4 Bytes). Daher, so schließen wir, muß der EBER bei der Adresse 707 beginnen.

Das wäre nun richtig, wenn nicht folgendes passieren würde: Ist die angegebene Startadresse nicht durch 8 teilbar, so wird bei der Umwandlung der Zuordnungszähler automatisch auf die nächste Doppelwortbegrenzung gesetzt.

In unserem Fall (Aufgabe 12) heißt dies, daß als Startadresse automatisch die Adresse 704 genommen wird, auch wenn wir 701 angegeben haben. Der EBER beginnt somit in Wirklichkeit bei der Adresse 710 (704 + 6).

Der besseren Überschaubarkeit wegen werden wir jedoch auch künftig unsere Speicherbelegungen wie bisher zeichnen.

Wir lernen nun drei weitere Assembler-Instruktionsanweisungen kennen.

→ Band 3, Seite 299 (DROP)

Seite 91 (END)

Seite 95 (ORG)

Nehmen Sie bitte wieder Ihre Speicherbelegung und Ihre Codierung zur Aufgabe 12 zur Hand.

Wir haben bei den Definitionen durch zwei Anweisungen DS CL35 (vor SBER und DBER) dafür gesorgt, daß der Stanzbereich und der Druckbereich so gespeichert werden, wie es die Speicherbelegung zeigt.

Die beiden DS-Anweisungen könnten wir nun durch entsprechende ORG-Anweisungen ersetzen.

Wie müßten die beiden Eintragungen lauten, wenn in diesen jeweils auf die Adresse von EBER Bezug genommen würde?

* * * * *

ORG. .	EBER+50.	(vor SBER)
und	. . . . .	
ORG. .	EBER+100.	(vor DBER)

Die Verwendung von ORG anstelle von DS hat den Vorteil, daß uns hierbei weitaus seltener ein Fehler unterläuft (z. B. durch falsches Abzählen oder falsches Rechnen).

Nun müssen wir noch unsere END-Anweisung vervollständigen.

Die nachzutragende Verzweigungsadresse nennen wir ANFANG.

"ANFANG" ist außerdem als Name der _____-Instruktion hinzuzufügen.

* * * * *

BALR (-Instruktion) (diese ist ja die erste auszuführende Instruktion unseres Programms)

Ergänzen Sie bitte Ihre Codierung entsprechend.

Damit können Sie die Aufgabe 12 endgültig ablegen.

Aufgabe 13

Eingabe: Verbrauchskarten

Verbraucher-Nr.	Verbrauch 1	Verbrauch 2	Verbrauch 3	Verbrauch 4	Verbrauch 5	Verbrauch 6	Verbrauch 7	Verbrauch 8	Verbrauch 9	Verbrauch 10	Verbrauch 11	Verbrauch 12	
1 - 5	6 - 11	12 - 17	18 - 23	24 - 29	30 - 35	36 - 41	42 - 47	48 - 53	54 - 59	60 - 65	66 - 71	72 - 77	

Verbrauch 1 - 12 = Monatsverbrauch Januar - Dezember

Ausgabe: a) Verbrauchsliste

Verbraucher-Nr.	Jahresverbrauch	Durchschnittsverbrauch je Monat
12	8.372	697
340	5.519	459
766	7.212	601
933	984	82
967	6.416	534
2340	322.480	26.873
83460	8.563.071	713.589
84001	10.343.953	861.996

Druckstellen: 9 - 13 | 16 - 25 | 28 - 34

b) Statistikkarte

Verbraucher-Nr.	Jahresverbrauch	Durchschnittsverbrauch	
1 - 5	6 - 13	14 - 19	

Zur Verarbeitung:

Der Jahresverbrauch (Summe von Verbrauch 1-12) wird max. 8-stellig, der Durchschnittsverbrauch je Monat (Jahresverbrauch : 12) max. 6-stellig.
Der Durchschnittsverbrauch ist nicht zu runden. Der Jahresverbrauch und der Durchschnittsverbrauch sind ohne positive Vorzeichen in die Lochkarte zu stanzen.

Abb. 1

Aufgabe 13

Sehen Sie sich bitte die Aufgabe 13 (Abb. 1) genau an. -

Entnehmen Sie Ihren Arbeitsmitteln

a) die Aufgabe 13

b) den Programmablaufplan zur Aufgabe 13.

Machen Sie sich bitte, bevor Sie umblättern, mit dem im Programmablaufplan gezeigten Lösungsweg vertraut.

IBM Entwurfsblatt für Programmablaufpläne

Programm: Aufgabe 13 Nr.: Seite:

Programmierer: Datum:

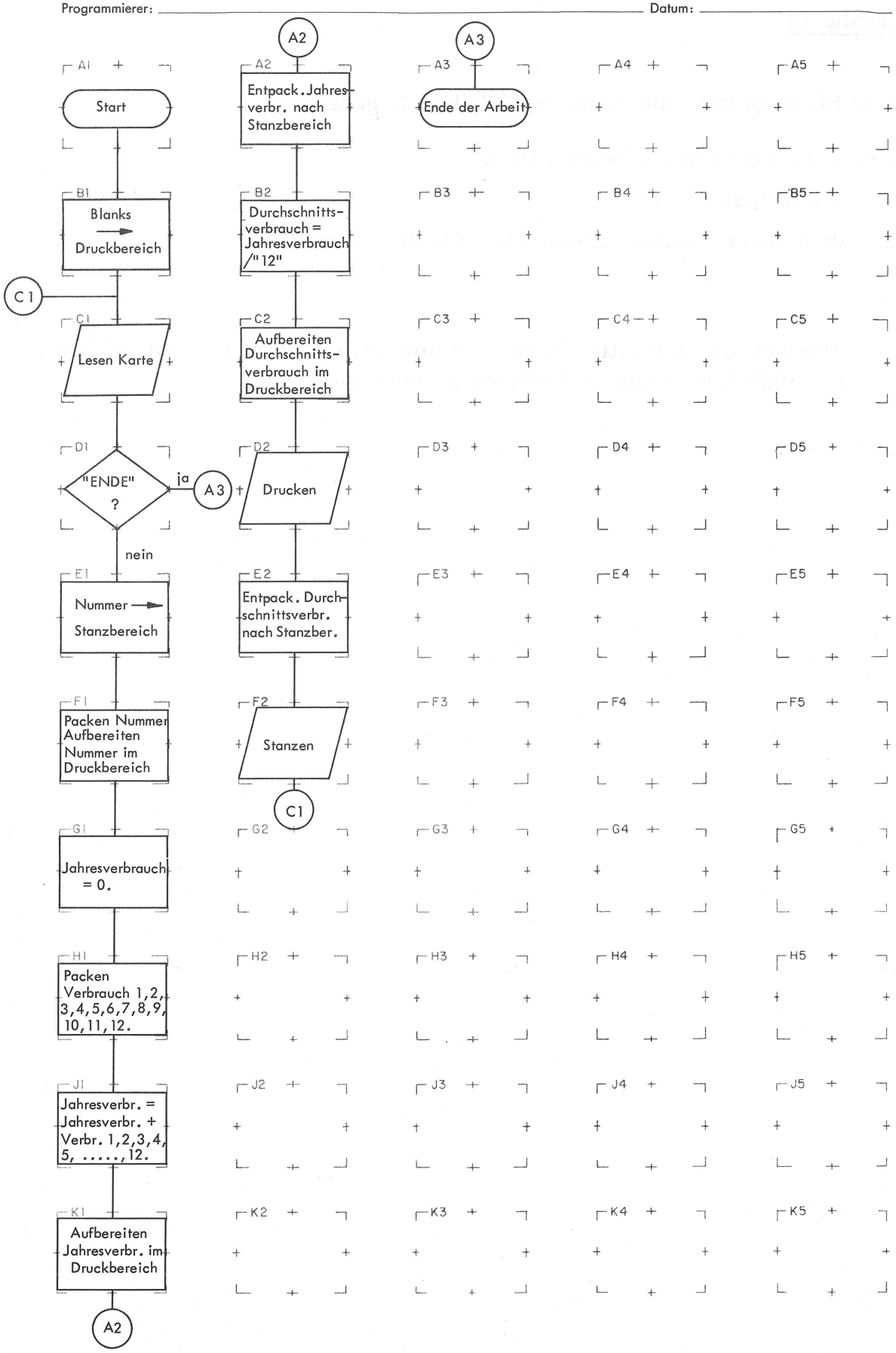


Abb. 2

Zum Programmablaufplan:

Zu Beginn blanken wir den Druckbereich, damit wir sicher sind, daß die nicht durch das Programm aufgefüllten Druckstellen Blanks enthalten (vgl. z. B. Stellen vor Verbraucher-Nr.).

Wesentlich umfangreicher würde der Programmablaufplan, wenn wir die Verarbeitung der 12 Verbrauchsfelder einzeln aufzeigen würden (also: Packen Verbrauch 1, Jahresverbrauch = Jahresverbrauch + Verbrauch 1; Packen Verbrauch 2, ...). - In Abb. 2 ist diese Verarbeitung in zwei Blöcken zusammengefaßt.

Da die Nummer ohne "Führende Nullen" zu drucken ist, müssen wir sie zum Drucken entsprechend aufbereiten.

Wir überlegen nun, welche Konstanten, Rechenfelder und Schablonen erforderlich sind.

Zunächst zu den Konstanten.

Zur Prüfung auf "ENDE"-Karte (Block D1) benötigen wir die Konstante _____.

* * * * *

ENDE

Das Feld Jahresverbrauch löschen wir diesmal nicht durch Subtraktion von sich selbst. Wir lösen dies durch Einbringen einer Konstanten "____(Z)_____" (Block G1).

* * * * *

0 ($\lfloor 0+ \rfloor$)

Zur Errechnung des monatlichen Durchschnittsverbrauchs ist eine Konstante "____(Z)____" erforderlich.

* * * * *

12 (0 1 2 +)

Nun bestimmen wir die Rechenfelder.

Da der Jahresverbrauch max. ____(Z)____-stellig wird, muß das Rechenfeld hierfür ____(Z)____ Bytes groß sein.

* * * * *

8

5

(1) Der errechnete Jahresverbrauch ist unser Dividend bei der Errechnung des durchschnittlichen Monatsverbrauchs.

Da der Dividend (Jahresverbrauch) ____(Z)____ Bytes und der Divisor ("12") ____(Z)____ Bytes umfassen, muß das Rechenfeld zur Errechnung des Monatsverbrauchs ____(Z)____ Bytes groß sein.

* * * * *

5

2

7

Jetzt müssen wir uns noch über die benötigten Schablonen klarwerden.

Betrachten Sie bitte die Ausgabeliste. -

Erforderlich sind ____(Z)____ verschiedene Schablonen.

* * * * *

3 (je eine für Nummer, Jahresverbrauch und Monatsverbrauch)

Wie muß die Schablone zur Aufnahme der Nummer aussehen?

* * * * *

	▽	▽	▽	▽	▽

Das Rechenfeld "Jahresverbrauch" ist 5 Bytes groß (| 0 X | X X | X X | X X | X + |)

In der Schablone muß Platz zur Aufnahme von ____(Z)____ Ziffern sein.

Wie muß also die Schablone aussehen?

* * * * *

9 (da bei ED nur ganze Bytes verarbeitet werden, muß von der Schablone auch die linke "0" aufgenommen werden)

	▽	▽	▽	.	▽	▽	▽	.	▽	▽	▽

Nun fehlt uns noch die Schablone für den durchschnittlichen Monatsverbrauch.

Laut Aufgabenstellung wird der Monatsverbrauch max. ____(Z)____-stellig.

Das Rechenfeld "Monatsverbrauch" ist ____(Z)____ Bytes groß.

Nach der Division hat das Rechenfeld folgenden Inhalt:

_____.

* * * * *

6

7 (vgl. (1) , Seite 425)

0 0	0 X	X X	X X	X +	0 X	X +

Monatsverbrauch Rest

Das linke Byte des Rechenfeldes enthält immer "0 0".

Diese beiden Nullen können wir bei der Festlegung der Schablone außer Betracht lassen. - Dies gilt jedoch nicht für die anschließende Null im zweiten Byte.

Folglich muß die Schablone Platz für die Aufnahme von ____(Z)____ Ziffern bieten.

Wie muß also die Schablone aussehen?

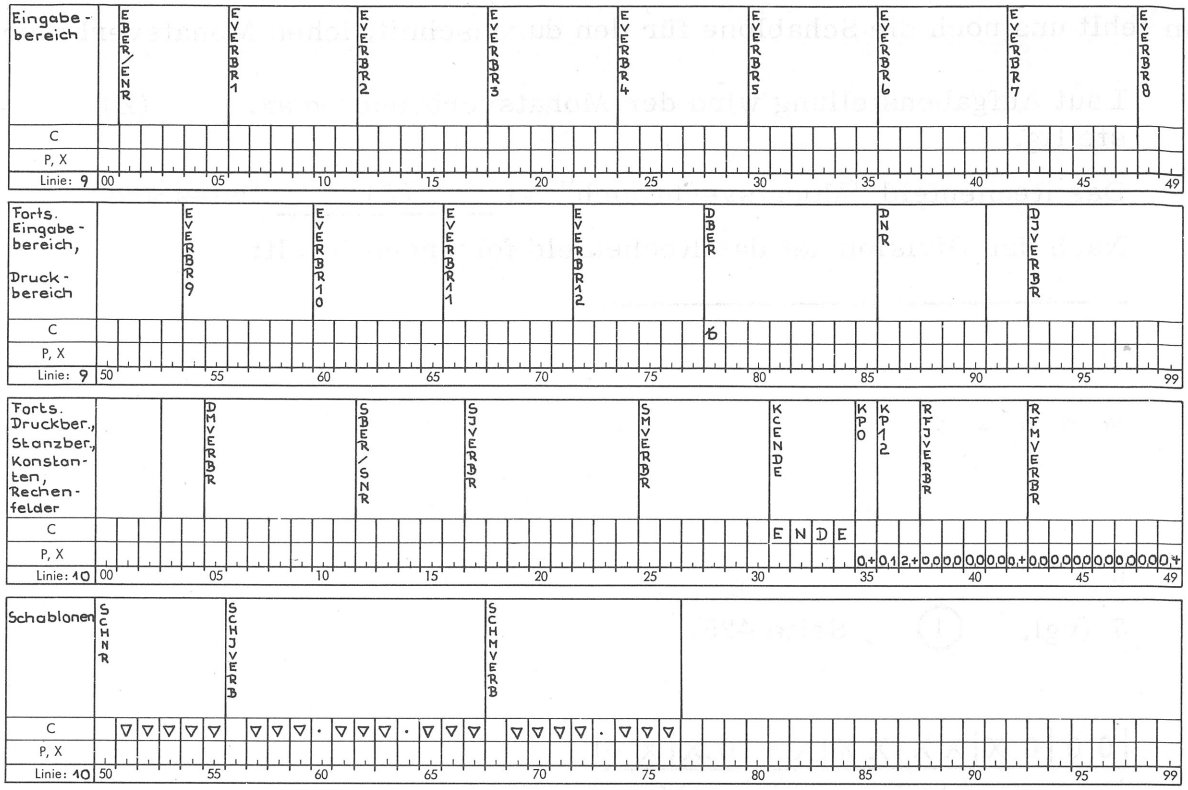
* * * * *

7

_	▽	▽	▽	▽	.	▽	▽	▽

IBM System / 360 Speicherbelegung

Aufgabe 13



IBM System / 360 Assembler Programmschema

Programm: Aufgabe 13

Programmierer | Datum

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–30)
*.V.E.R.B.R.A.U.	C.H.S.S.T.A.T.	I.S.T.I.K.
	S.T.A.R.T.	8,9,5.
A.N.F.A.N.G.	B.A.L.R.	1,Ø,,Ø.
	U.S.I.N.G	*,,1,Ø.
	B.	B.1.
*.D.E.F.I.N.I.T	I.O.N.E.N	
E.B.E.R.	D.S.	Ø.C.L.7.7.
E.N.R.	D.S.	C.L.5.
E.V.E.R.B.R.1.	D.S.	C.L.6.
E.V.E.R.B.R.2.	D.S.	C.L.6.
E.V.E.R.B.R.3.	D.S.	C.L.6.
E.V.E.R.B.R.4.	D.S.	C.L.6.
E.V.E.R.B.R.5.	D.S.	C.L.6.
E.V.E.R.B.R.6.	D.S.	C.L.6.
E.V.E.R.B.R.7.	D.S.	C.L.6.
E.V.E.R.B.R.8.	D.S.	C.L.6.
E.V.E.R.B.R.9.	D.S.	C.L.6.
E.V.E.R.B.R.1.Ø.	D.S.	C.L.6.
E.V.E.R.B.R.1.1.	D.S.	C.L.6.
E.V.E.R.B.R.1.2.	D.S.	C.L.6.

IBM System / 360 Assembler Programmschema

Programm: Aufgabe 13

Programmierer | Datum

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
D.B.E.R.	D.S.	Ø.C.L.3,4,
	D.C.	C.' '
	D.S.	C.L.7.
D.N.R.	D.S.	C.L.5.
	D.S.	C.L.2.
D.J.V.E.R.B.R.	D.S.	C.L.1.Ø.
	D.S.	C.L.2.
D.M.V.E.R.B.R.	D.S.	C.L.7.
S.B.E.R.	D.S.	Ø.C.L.1.9.
S.N.R.	D.S.	C.L.5.
S.J.V.E.R.B.R.	D.S.	C.L.8.
S.M.V.E.R.B.R.	D.S.	C.L.6.
K.C.E.N.D.E.	D.C.	C.'E.N.D.E.'
K.P.Ø.	D.C.	P.'Ø.'
K.P.1.2.	D.C.	P.'1.2.'
R.F.J.V.E.R.B.R	D.C.	P.L.5.'Ø.'
R.F.M.V.E.R.B.R	D.C.	P.L.7.'Ø.'
S.C.H.N.R.	D.C.	C.'. ∇.∇.∇.∇.∇.'
S.C.H.J.V.E.R.B	D.C.	C.'. ∇.∇.∇. . ∇.∇.∇. . ∇.∇.∇.'
S.C.H.M.V.E.R.B	D.C.	C.'. ∇.∇.∇.∇. . ∇.∇.∇.'

Abb. 3

Die Abb. 3 zeigt die Speicherbelegung und die dazugehörenden Definitionen.

Vergleichen Sie bitte Speicherbelegung und Definitionen.

Nehmen Sie bitte den Programmablaufplan zur Aufgabe 13 zur Hand.

Codieren Sie nun den Programmablauf. - Beginnen Sie bitte mit der Überschrift "Programmablauf".

Blättern Sie bitte um.

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) die Speicherbelegung zur Aufgabe 13

b) die Codierung (5 Seiten).

Überprüfen Sie Ihre Codierung des Programmablaufs.

Falls Sie zum Löschen des Rechenfeldes MVO anstelle von ZAP verwendet haben, ist dies natürlich auch richtig, vorausgesetzt, daß Sie als zweiten Operanden ein Byte mit dem Inhalt |0 0| benutzt haben.

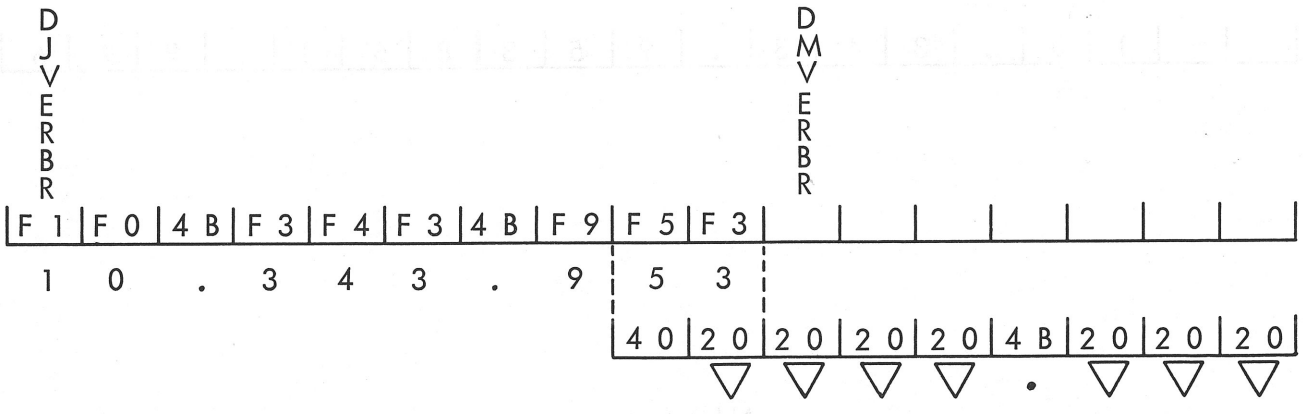
Betrachten Sie bitte Ihre Speicherbelegung. -

Was würde geschehen, wenn das Feld "DMVERBR" unmittelbar an das Feld "DJVERBR" anschließen würde?

* * * * *

Bei der vorliegenden Codierung würden beim Einbringen der Schablone "SCHMVERB" die beiden letzten Ziffern im Feld "DJVERBR" zerstört.

Beispiel (s. letzte Zeile Verbrauchsliste):



	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30 – 35 – 40)
a)	MVC	DMVERBR-2(9),SCHMVERB
b)	ED	DMVERBR-2(9),RFMVERBR+1
c)	MVC	DJVERBR-2(12),SCHJVERB
d)	ED	DJVERBR-2(12),RFJVERBR

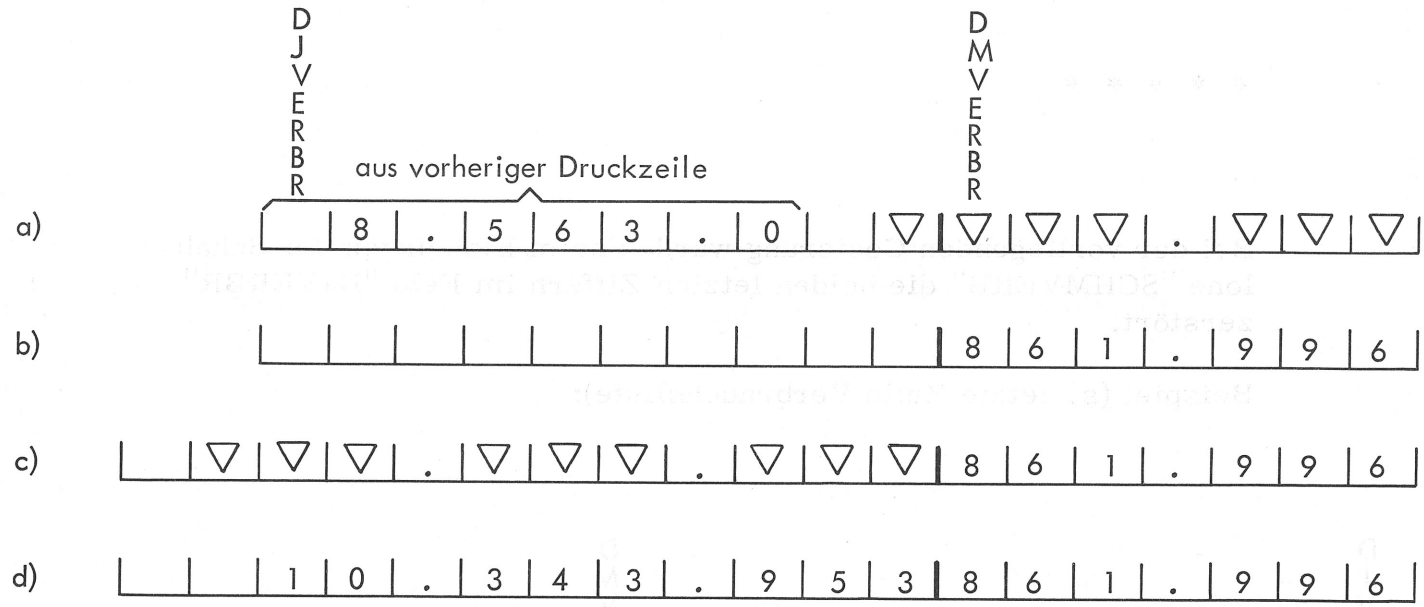


Abb. 4

Wie könnten wir dies verhindern, ohne die betroffenen Felder im Druckbereich auseinanderzurücken?

* * * * *

Sinngemäß:

In diesem Fall müßten wir zuerst das Feld "DMVERBR" und erst danach das Feld "DJVERBR" auffüllen (vgl. Abb. 4).

(2)

Betrachtungen zur Lösung der Aufgabe 13.

Im Programmablaufplan haben wir für das Packen und Addieren der 12 Verbrauchsfelder zwei Blöcke gezeichnet. - Bei der Codierung mußten wir dafür zusammen 24 Instruktionen schreiben. Die einzelnen Instruktionen unterscheiden sich nur geringfügig voneinander.

Die Operationen heißen immer PACK bzw. AP. Bei der AP-Instruktion ist der 1. Operand auch immer RFJVERBR. - Unterschiede ergeben sich lediglich durch die Operanden EVERBR1, EVERBR2, ... EVERBR12.

Stellen Sie sich vor, wir hätten statt der 12 Felder z.B. 50 Felder in gleicher Weise zu verarbeiten. Dies würde eine noch umfangreichere Schreibarbeit verlangen.

Wir versuchen nun, einen Weg zu finden, bei dem wir das gleiche Ziel mit weniger Aufwand erreichen können.

Vergleichen Sie anhand der Speicherbelegung die Adressen der 12 Verbrauchsfelder. -

Von einem Verbrauchsfeld zum nächsten erhöht sich die Adresse jeweils um ____(Z)____.

* * * * *

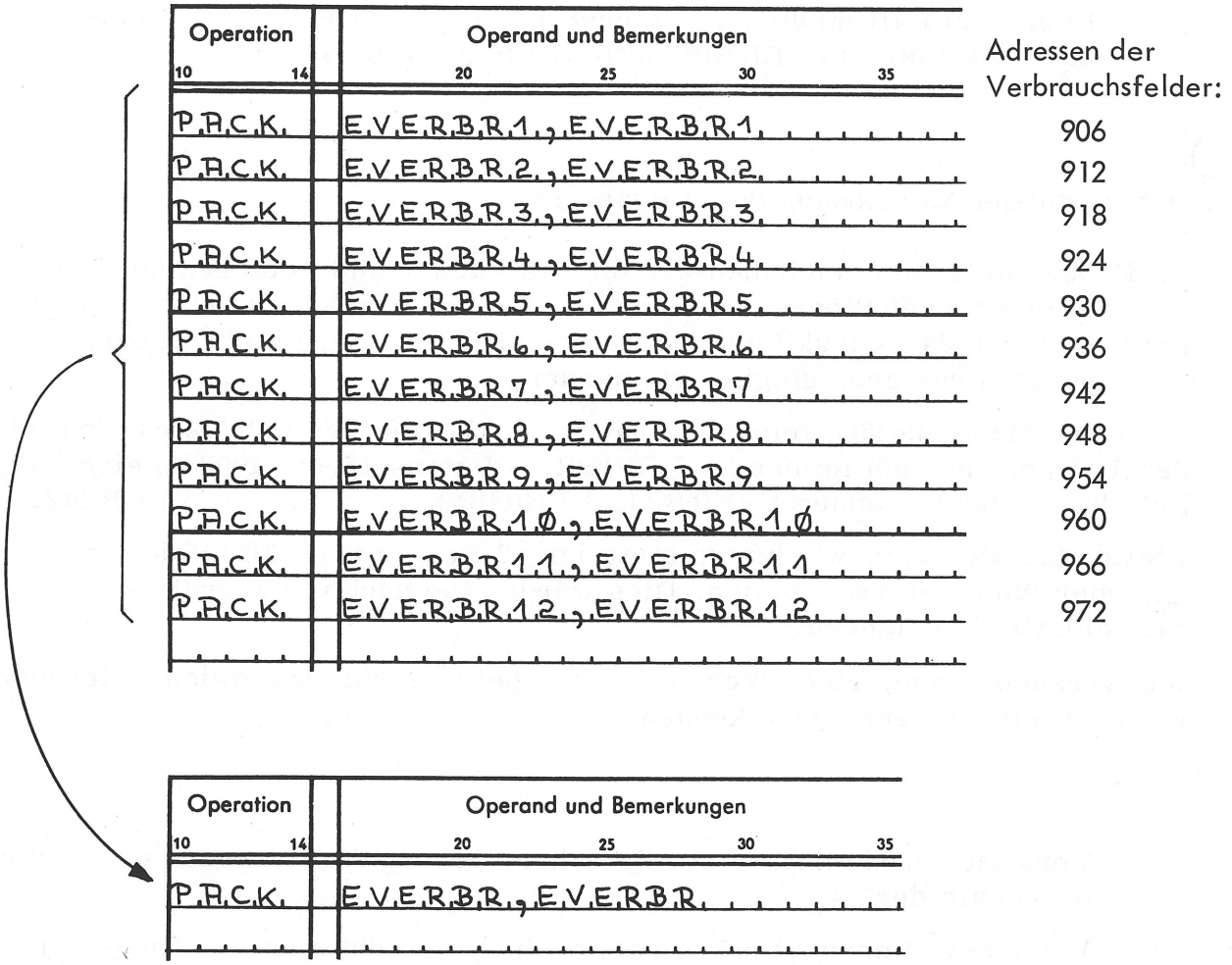

Operation
Operand und Bemerkungen
10
14
20
25
30
35
P.A.C.K. E.V.E.R.B.R.1.,E.V.E.R.B.R.1.
P.A.C.K. E.V.E.R.B.R.2.,E.V.E.R.B.R.2.
P.A.C.K. E.V.E.R.B.R.3.,E.V.E.R.B.R.3.
P.A.C.K. E.V.E.R.B.R.4.,E.V.E.R.B.R.4.
P.A.C.K. E.V.E.R.B.R.5.,E.V.E.R.B.R.5.
P.A.C.K. E.V.E.R.B.R.6.,E.V.E.R.B.R.6.
P.A.C.K. E.V.E.R.B.R.7.,E.V.E.R.B.R.7.
P.A.C.K. E.V.E.R.B.R.8.,E.V.E.R.B.R.8.
P.A.C.K. E.V.E.R.B.R.9.,E.V.E.R.B.R.9.
P.A.C.K. E.V.E.R.B.R.1.Ø.,E.V.E.R.B.R.1.Ø.
P.A.C.K. E.V.E.R.B.R.1.1.,E.V.E.R.B.R.1.1.
P.A.C.K. E.V.E.R.B.R.1.2.,E.V.E.R.B.R.1.2.
Adressen der Verbrauchsfelder:
906
912
918
924
930
936
942
948
954
960
966
972
Operation
Operand und Bemerkungen
10
14
20
25
30
35
P.A.C.K. E.V.E.R.B.R.,E.V.E.R.B.R.

Abb. 5

Wenn wir von der Adresse von EVERBR1 ausgehen, so erhalten wir durch jeweiliges Erhöhen um 6 die Adressen der übrigen Verbrauchsfelder (EVERBR2 bis EVERBR12).

Wir bräuchten somit nur eine PACK-Instruktion und nur eine AP-Instruktion zu schreiben, wenn es uns gelänge, diese eine Instruktion zwölfmal hintereinander, aber mit jeweils anderen Adressen, ablaufen zu lassen.

Hierzu müßten wir eine entsprechende Programmschleife programmieren.

Abb. 5 zeigt nochmals deutlich unser Problem.
Sehen Sie sich bitte Abb. 5 genau an. -

Anstelle der 12 PACK-Instruktionen soll unser Programm hinterher nur noch eine PACK-Instruktion enthalten.

Hieraus ergibt sich notwendigerweise zweierlei:

1. Diese eine Instruktion muß bei der Ausführung des Programms 12 mal ablaufen, da ja 12 Verbrauchsfelder zu verarbeiten sind.

2. Mit dieser Instruktion müssen nacheinander 12 verschiedene Verbrauchsfelder angesprochen werden. Damit dies erfolgen kann, muß die Adresse EVERBR entsprechend modifiziert (verändert) werden.

Für EVERBR müssen dann bei der Verarbeitung nacheinander folgende Adressen eingesetzt werden:

906	Bei der Verarbeitung des 1. Verbrauchsfeldes
912	Bei der Verarbeitung des 2. Verbrauchsfeldes
.	
.	
.	
972	Bei der Verarbeitung des 12. Verbrauchsfeldes

(3)

Bevor wir uns jedoch der Lösung dieses Problems zuwenden können, müssen wir uns noch einige Kenntnisse über Konstanten und Instruktionen für Dualdaten sowie über Formate der Instruktionen aneignen.

→ Band 3, Seite 237 (DC -F, H, A)

Band 4, Seite 459 (A, AH, AR, LR, L, LA)

Band 2, Seite 217 (Kapitel VI)

Band 4, Seite 503 (Anhang zu den Instruktionen)

Band 3, Seite 307 (Kapitel VI)

Sie haben nun das Wissen, das zur Lösung unseres im Anschluß an die Aufgabe 13 aufgeworfenen Problems erforderlich ist. - Wir werden uns die Lösung schrittweise erarbeiten, machen uns jedoch zunächst wieder mit der Problemstellung vertraut. - Wiederholen Sie bitte hierzu den Lernstoff von (2) , Seite 435 bis (3) , Seite 437 und fahren Sie danach an dieser Stelle fort.

Nun überlegen wir uns, auf welche Weise wir eine Adresse verändern können.

Zwei Möglichkeiten, eine Adresse zu verändern, sind naheliegend. Änderung der Adresse durch:

a) Verändern der Basisadresse

b) Verändern der relativen Adresse.

Beide Möglichkeiten kommen jedoch nicht in Frage.

Begründung:

zu a) Verändern der Basisadresse

Durch eine Änderung der Basisadresse ändern wir nicht nur die Adresse eines Feldes, sondern alle Adressen des Programms.

Würden wir nach der Verarbeitung des 1. Verbrauchsfeldes die Basisadresse um 6 erhöhen, so würde der Verbrauch 2 nicht mehr nach RFJVERBR, sondern nach RFJVERBR+6 addiert. RFJVERBR+6 ist aber die Adresse des zweiten Bytes von RFMVERBR (vgl. Speicherbelegung).

Hieraus ergibt sich für unsere Aufgabe, daß eine Veränderung des Basisregisters 10 katastrophale Auswirkungen hätte.

zu b) Verändern der relativen Adresse

Sie wissen, daß relative Adressen von 0 - 4095 möglich sind.

Eine relative Adresse könnten wir also nur dann erhöhen, wenn wir genau wüßten, daß der Wert 4095 nicht überschritten wird.

Dies entzieht sich jedoch meist unserer Kenntnis. (Eine Erklärung hierzu ist in diesem Lehrgang nicht möglich.)

Eine Änderung der relativen Adresse scheidet somit ebenfalls bei unseren Überlegungen aus.

Trotzdem müssen und können wir erreichen, daß für EVERBR zwölf verschiedene Adressen nacheinander gebildet werden, o h n e daß sich dadurch am übrigen Programm etwas ändert.

Die Lösung ist im Grunde sehr einfach. Wir werden für EVERBR - und zwar nur für EVERBR - ein anderes Register zur Adreßbildung verwenden.

Basisregister bleibt das Register 10. Dieses Register darf jedoch nicht zur Bildung der Adresse EVERBR verwendet werden.

Zur Bildung der Adressen von EVERBR (Adressen für Verbrauch 1, Verbrauch 2 etc.) benutzen wir das Register 11.

Wie dies durchzuführen ist, erfahren Sie bei der Aufgabe 14.

Aufgabe 14

Die Aufgabe 14 entspricht der Aufgabe 13. - Für die Verarbeitung der 12 Verbrauchsfelder ist eine Programmschleife zu programmieren.

Eingabe: Verbrauchskarten

Verbraucher-Nr.	Verbrauch 1	Verbrauch 2	Verbrauch 3	Verbrauch 4	Verbrauch 5	Verbrauch 6	Verbrauch 7	Verbrauch 8	Verbrauch 9	Verbrauch 10	Verbrauch 11	Verbrauch 12	
1 - 5	6 - 11	12 - 17	18 - 23	24 - 29	30 - 35	36 - 41	42 - 47	48 - 53	54 - 59	60 - 65	66 - 71	72 - 77	

Verbrauch 1 - 12 = Monatsverbrauch Januar - Dezember

Ausgabe: a) Verbrauchsliste

Verbraucher-Nr.	Jahresverbrauch	Durchschnittsverbrauch je Monat
12	8.372	697
340	5.519	459
766	7.212	601
933	984	82
967	6.416	534
2340	322.480	26.873
83460	8.563.071	713.589
84001	10.343.953	861.996

Druckstellen 9 - 13 | 16 - 25 | 28 - 34

b) Statistikkarte

Verbraucher-Nr.	Jahresverbrauch	Durchschnittsverbrauch	
1 - 5	6 - 13	14 - 19	

Zur Verarbeitung:

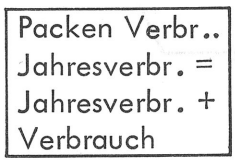
Der Jahresverbrauch (Summe von Verbrauch 1-12) wird max. 8-stellig, der Durchschnittsverbrauch je Monat (Jahresverbrauch : 12) max. 6-stellig.
Der Durchschnittsverbrauch ist nicht zu runden. Der Jahresverbrauch und der Durchschnittsverbrauch sind ohne positive Vorzeichen in die Lochkarte zu stanzen.

Abb. 1

Schauen Sie sich bitte die Aufgabe 14 (Abb. 1) an.

Entnehmen Sie bitte Ihren Arbeitsmitteln die Aufgabe 14.

Wir wenden uns jetzt dem Programmablaufplan zu.



Dieser Block zeigt die Verarbeitung eines Verbrauchsfeldes.

Damit alle Verbrauchsfelder verarbeitet werden, muß er bei jeder Karte ___(Z)___ mal durchlaufen werden.

Zur Steuerung dieses Ablaufes benötigen wir einen Zähler.

* * * * *

12

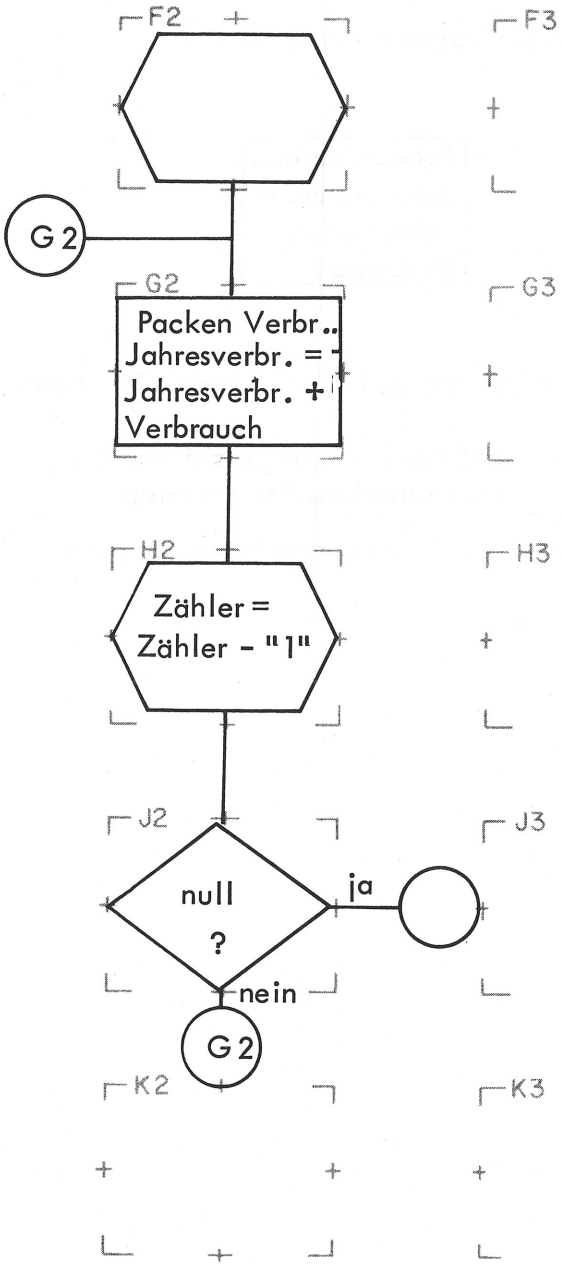

F2
F3
G 2
G2
G3
Packen Verbr..
Jahresverbr. =
Jahresverbr. +
Verbrauch
H2
H3
Zähler =
Zähler - "1"
J2
J3
null
?
ja
nein
G 2
K2
K3

Abb. 2

Betrachten Sie bitte Abb. 2, insbesondere die Eintragungen in den Blöcken H2 und J2.

Welche Eintragung muß der Block F2 erhalten?

* * * * *

Zähler = "12"

Damit haben wir die eine Hälfte des Problems gelöst.

Nun müssen wir noch dafür sorgen, daß nicht 12 mal das gleiche Feld verarbeitet wird, wie es bis jetzt der Fall wäre.

Zur Modifizierung der Adressen haben wir das Register 11 vorgesehen.

V o r der Verarbeitung des 1. Verbrauchsfeldes bringen wir dessen Adresse in das Register 11.

N a c h der Verarbeitung eines Verbrauchsfeldes müssen wir dann das Register 11 jeweils um ___(Z)_____ erhöhen.

* * * * *

6

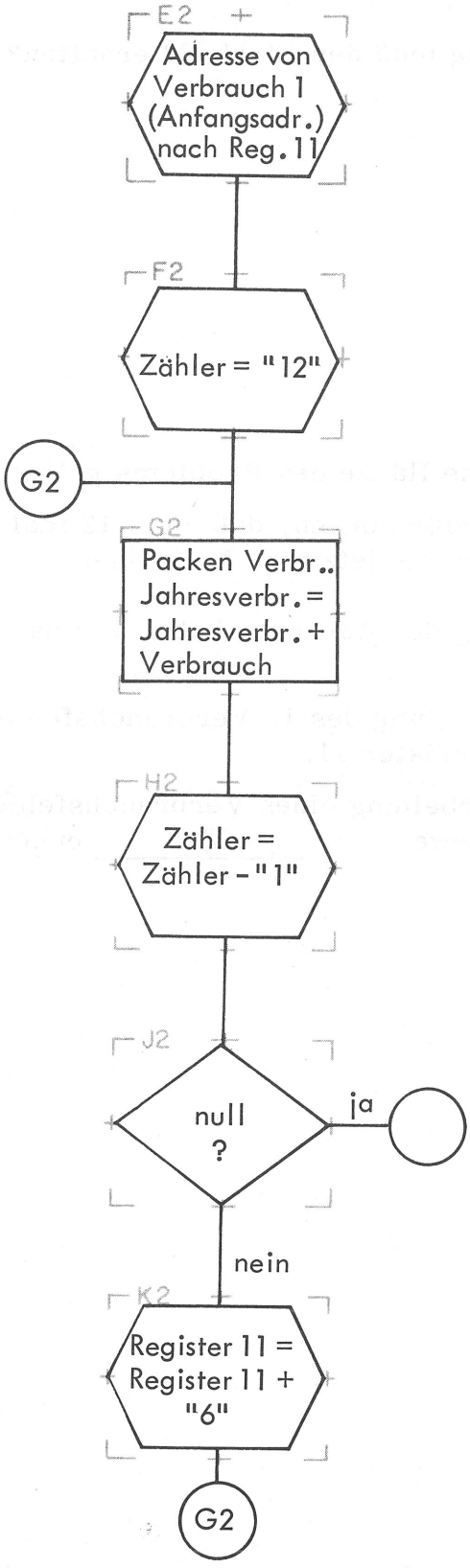

E2
Adresse von Verbrauch 1 (Anfangsadr.) nach Reg. 11
F2
Zähler = "12"
G2
G2
Packen Verbr.. Jahresverbr. = Jahresverbr. + Verbrauch
H2
Zähler = Zähler - "1"
J2
null ?
ja
nein
K2
Register 11 = Register 11 + "6"
G2

Abb. 3

In der Abb. 3 sind die hierfür erforderlichen Schritte gezeigt.

Sehen Sie sich die hinzugekommenen Schritte genau an.

Wir überprüfen die in Abb. 3 gezeigten Schritte. Hierzu benötigen wir noch:

a) die Speicherbelegung zur Aufgabe 13

b) die Übung 103 aus den Arbeitsmitteln.

Nehmen Sie bitte diese Unterlagen zur Hand.

Das Übungsblatt zeigt, was bei der Durchführung der einzelnen Schritte geschieht.
Durch den Schritt _____ wird die Anfangsadresse ____(Z)____ in das Register 11 gebracht.

* * * * *

E2 (vgl. Abb. 3)

906 (Adresse des 1. Verbrauchsfeldes; vgl. Speicherbelegung)

Anschließend wird in den Zähler der Wert ____(Z)____ gebracht.

* * * * *

12 (Anfangswert)

Danach wird das ____(Z)____ Verbrauchsfeld verarbeitet.

* * * * *

1.

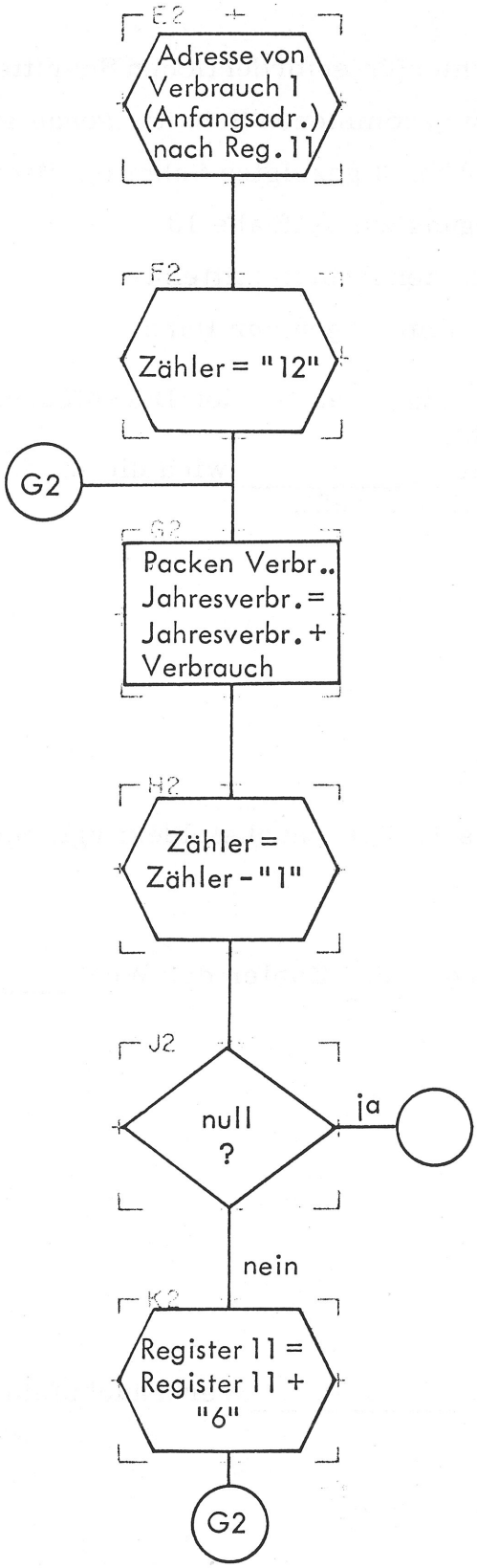

E2
Adresse von Verbrauch 1 (Anfangsadr.) nach Reg. 11
F2
Zähler = "12"
G2
G2
Packen Verbr.. Jahresverbr. = Jahresverbr. + Verbrauch
H2
Zähler = Zähler - "1"
J2
null ?
ja
nein
K2
Register 11 = Register 11 + "6"
G2

Abb. 3

Der Zähler wird durch Schritt _____ um "1" vermindert und hat danach den Wert ____(Z)____.

* * * * *

H2

11

Die folgende Frage auf "Zähler null" (Schritt J2) ergibt _____.

* * * * *

nein

Bevor zu Schritt _____ verzweigt wird, wird das Register 11 um 6 erhöht.
Danach enthält das Register 11 die Adresse ____(Z)____.
Dies ist die Adresse des ____(Z)____ Verbrauchsfeldes.

* * * * *

G2

912

2.

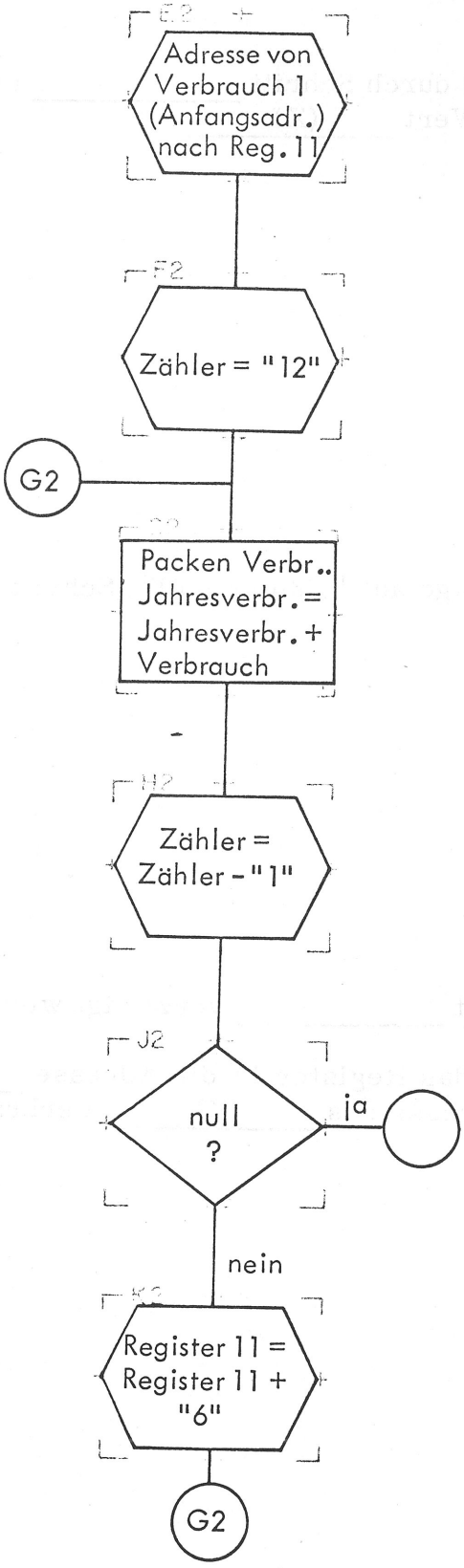

Adresse von Verbrauch 1 (Anfangsadr.) nach Reg. 11
Zähler = "12"
G2
Packen Verbr.. Jahresverbr. = Jahresverbr. + Verbrauch
Zähler = Zähler - "1"
null ?
ja
nein
Register 11 = Register 11 + "6"
G2

Abb. 3

Bei der Wiederholung des Schrittes G2 wird nun das (Z) Verbrauchsfeld verarbeitet.

* * * * *

2.

Verfolgen Sie nun den weiteren Ablauf und ergänzen Sie bitte das Übungsblatt (ab Schritt G2 (11)) in entsprechender Weise.

* * * * *

Vergleichen Sie bitte die hinzugefügten Eintragungen mit denen der Abb. 4 (Rückseite!).

Schritt ( )*)	Inhalt Register 11	entspricht Adresse von:	Inhalt des Zählers	Es wird verarbeitet:	Antwort auf Frage "Zähler Null"
E2	906	Verbrauch 1			
F2	"	"	12		
G2 (1)	"	"	"	Verbrauch 1	
H2 (1)	"	"	11		
J2 (1)	"	"	"		nein
K2 (1)	912	Verbrauch 2	"		
G2 (2)	"	"	"	Verbrauch 2	
H2 (2)	"	"	10		
J2 (2)	"	"	"		nein
K2 (2)	918	Verbrauch 3	"		
G2 (3)	"	"	"	Verbrauch 3	
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Verbessern Sie Ihre Eintragungen soweit notwendig.

Die noch fehlenden Schritte im Programmablaufplan entsprechen denen unseres Programmablaufplans zu Aufgabe 13.

Nehmen Sie nun bitte folgende Unterlagen zur Hand:

a) die Speicherbelegung zur Aufgabe 13

b) den Programmablaufplan zur Aufgabe 14 aus den Arbeitsmitteln.

Für die Aufgabe 14 muß diese Speicherbelegung erweitert werden. Um welche Felder zu erweitern ist, ersehen wir aus den Schritten H2 und K2 des Programmablaufplans. Bitte sehen Sie sich diese Schritte an. Die Speicherbelegung muß um einen Zähler und zwei Konstanten erweitert werden.

Der Zähler muß ____(Z)_____ Byte(s) groß sein.

* * * * *

2

Zur Verminderung des Zählers (Schritt H2) benötigen wir eine Konstante 1 in _____ Darstellungsform.

* * * * *

gepackter

Zur Erhöhung des Registers 11 (Schritt K2) ist eine Konstante 6 in _____ Darstellungsform erforderlich.

* * * * *

dualer

Für diese duale 6 können wir eine _____-Konstante oder eine _____-Konstante benutzen.

* * * * *

Halbwort

Wort

Wir entscheiden uns für die Halbwort-Konstante, da diese vollkommen ausreichend ist.

Entnehmen Sie bitte Ihren Arbeitsmitteln die Speicherbelegung zur Aufgabe 14.

Die hinzugekommenen 3 Felder schließen sich an die letzte Schablone an.

Da wir die einzelnen Verbrauchsfelder nicht mehr mit eigenen symbolischen Namen (EVERBR1, EVERBR2, ...), sondern mit Hilfe des Registers 11 ansprechen werden, benötigt nur das erste Verbrauchsfeld einen Namen (EVERBR).

Erstellen Sie nun die Definitionen (vgl. Definitionen zu Aufgabe 13).

Beachten Sie bitte, daß die 12 Verbrauchsfelder mit einer DS-Anweisung definiert werden.

* * * * *

Überprüfen Sie bitte Ihre Definitionen anhand der Abb. 5. (Rückseite!)

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 14	Lochhinweise		Problem A14 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 011 (76–77) von ____

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.V.E.R.B.R.A	U C.H.S.S.T.A	T.1.S.T.1.K.	010
	S.T.A.R.T	B.9.5.	020
A.N.F.A.N.G.	B.A.L.R.	1.0.,0.	030
	U.S.I.N.G	*.,1.0.	040
	B.	B.1.	050
E.B.E.R.	D.S.	0.C.L.7.7.	060
E.N.R.	D.S.	C.L.5.	070
E.V.E.R.B.R.	D.S.	1.2.C.L.6.	080
D.B.E.R.	D.S.	0.C.L.3.4.	090
	D.C.	C.' '	100
	D.S.	C.L.7.	110
D.N.R.	D.S.	C.L.5.	120
	D.S.	C.L.2.	130
D.J.V.E.R.B.R.	D.S.	C.L.1.0.	140
	D.S.	C.L.2.	150
D.M.V.E.R.B.R.	D.S.	C.L.7.	160
S.B.E.R.	D.S.	0.C.L.1.9.	170
S.N.R.	D.S.	C.L.5.	180
S.J.V.E.R.B.R.	D.S.	C.L.8.	190
S.M.V.E.R.B.R.	D.S.	C.L.6.	200

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 14	Lochhinweise		Problem A14 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 012 (76–77) von ____

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
K.C.E.N.D.E.	D.C.	C.' E.N.D.E.'	010
K.P.0.	D.C.	P.' 0.'	020
K.P.1.2.	D.C.	P.' 1.2.'	030
R.F.J.V.E.R.B.R	D.C.	P.L.5.' 0.'	040
R.F.M.V.E.R.B.R	D.C.	P.L.7.' 0.'	050
S.C.H.N.R.	D.C.	C.' ▽▽▽▽▽▽.'	060
S.C.H.J.V.E.R.B	D.C.	C.' ▽▽▽.▽▽▽.▽▽▽.'	070
S.C.H.M.V.E.R.B	D.C.	C.' ▽▽▽▽.▽▽▽.'	080
Z.A.E.H.L.E.R.	D.S.	C.L.2.	090
K.P.1.	D.C.	P.' 1.'	100
K.H.6.	D.C.	H.' 6.'	110
			120
			130
			140
			150
			160
			170
			180
			190
			200

Abb. 5

Zu den Definitionen:

Wenn Sie EVERBR mit DS . . | C L 7 2 definiert haben, so haben Sie nicht zwölf 6 Bytes große Verbrauchsfelder, sondern ein 72 Bytes großes Verbrauchsfeld definiert.

Wenn Sie den Zähler mit einer DC-Anweisung definiert haben, ist das natürlich auch in Ordnung.

Verbessern Sie Ihre Definitionen soweit erforderlich.

Bevor wir den gesamten Programmablauf codieren, nehmen wir uns die Blöcke E2 und G2 gesondert vor.

> Im Block E2 ist die Adresse von EVERBR in das Register 11 zu bringen.
>
> Die Instruktion hierfür muß lauten: _____.

* * * * *

Operation	Operand und Bemerkungen
10 14	20 25 30
L R . .	1 1 , , E V E R B R . . . . .

Das Register 11 enthält nun die richtige (effektive) Adresse des ersten Verbrauchsfeldes. Durch die Addition einer binären 6 (Block K2 des Programmablaufplans) steht die Adresse des nächsten Verbrauchsfeldes und durch Wiederholung der Addition schließlich die Adressen sämtlicher Verbrauchsfelder im Register 11 zur Verfügung.

Wir müssen nun dafür sorgen, daß die jeweiligen Inhalte des Registers 11 als Adressen der Operanden in der PACK-Instruktion (Block G2) verwendet werden. - Das können wir mit der expliziten Adressierung erreichen.

Wir müssen also in der PACK-Instruktion für beide Operanden explizit angeben:

a) die relative Adresse

b) die Länge des Feldes

c) das Basisregister.

Da das Register 11 immer die jeweils richtige effektive Adresse enthält, müssen wir als relative Adresse ____(Z)____ angeben.
Als Feldlänge muß ____(Z)____ angegeben werden.

* * * * *

0

6

Wie muß also die PACK-Instruktion lauten?

* * * * *

Operation	Operand und Bemerkungen
10 14	20 25 30 35
PACK	Ø(6,11),Ø(6,11)

Die Verbrauchsfelder liegen nun in der folgenden gepackten Form vor:

00	00	0X	XX	XX	X+

Die rechten 4 Bytes dieser Felder sind jetzt in das Feld für den Jahresverbrauch (RFJVERBR) zu addieren.

Erinnern Sie sich bitte, daß in einer Instruktion auch ein Operand symbolisch und der andere explizit angegeben werden kann und beantworten Sie folgende Frage:

Wie muß die der PACK-Instruktion folgende AP-Instruktion lauten, wenn nur die rechten 4 Bytes addiert werden sollen?

* * * * *

```
| |A.P. . . | |R.F.J.V.E.R.B.R.,.2.(.4.,.1.1.). .
```

Sie wissen nun alles, was zur Codierung des Programmablaufs erforderlich ist.

Codieren Sie bitte den Programmablauf.

* * * * *

Überprüfen Sie bitte Ihre Codierung anhand der Abb. 6. (Rückseite!)

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 14	Lochhinweise		Problem 14 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 03 (76–77) von ____

Name (1–8)	Operation (10–14)	Anweisung: Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
*.PROGRAMM	MABLAUF		010
B1	MVC	DBER+1(3,3),DBER	020
B2	GET	KARTE,EBER	030
	CLC	EBER(4),KCENDE	040
	BE	B4	050
	ZAP	RFJVERBR,KP0	060
	MVC	SNR,ENR	070
	PACK	ENR,ENR	080
	MVC	DNR-1(6),SCHNR	090
	ED	DNR-1(6),ENR+2	100
	LA	1,EVERBR	110
	MVC	ZAEHLER,KP12	120
G2	PACK	0(6,1),0(6,1)	130
	AP	RFJVERBR,2(4,1)	140
	SP	ZAEHLER,KP1	150
	BZ	B3	160
	AH	1,KH6	170
	B	G2	180
B3	MVC	DJVERBR-2(12),SCHJVERB	190
	ED	DJVERBR-2(12),RFJVERBR	200

IBM System / 360 Assembler
Programmschema

Programm	Aufgabe 14	Lochhinweise		Problem 14 (73–75)
		Zeichen		
Programmierer	Datum	Lochung		Seite 04 (76–77) von ____

Name (1–8)	Operation (10–14)	Anweisung: Operand und Bemerkungen (20–70)	Karte Nr. (78–80)
	UNPK	SJVERBR,RFJVERBR	010
	MVZ	SJVERBR+7(1),SJVERBR	020
	ZAP	RFMVERBR,RFJVERBR	030
	DP	RFMVERBR,KP12	040
	MVC	DMVERBR-2(9),SCHMVERB	050
	ED	DMVERBR-2(9),RFMVERBR+1	060
	PUT	LIST,DBER	070
	UNPK	SMVERBR,RFMVERBR(5)	080
	MVZ	SMVERBR+5(1),SMVERBR	090
	PUT	STANZ,SBER	100
	B	B2	110
B4	Ende der	Arbeit	120
	END	ANFANG	130

Abb. 6

Sollten Sie zur Erhöhung des Registers 11 die Operation AP anstatt AH verwendet haben, so ist dies ein schwerer Fehler.

Die Codierung des Programmablaufs finden Sie in Ihren Arbeitsmitteln.

Die Aufgabe 14 ist damit abgeschlossen.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35
	:	
	:	
	:	
	USING	*,10
	USING	EVERBR,11
	:	
	:	
	:	
	:	
G2	PACK	EVERBR,EVERBR
	AP	RFJVERBR,EVERBR+2(4)

a) PACK-Instruktion

OP	L1	L2	B1	D1	B2	D2
F2	5	5	11	0	11	0

b) AP-Instruktion

FA	4	3	10	141	11	2

Abb. 7

Aufgabe 15

Die Aufgabe 15 entspricht der Aufgabe 14. - Wir werden jedoch eine andere Lösung kennenlernen.

Nehmen Sie bitte Programmablaufplan und Codierung zur Aufgabe 14 zur Hand.

Die beiden für die Verarbeitung der 12 Verbrauchsfelder zu modifizierenden Instruktionen sind:

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30)
. . . . .	PACK.	Ø.(6,1.1),Ø.(6,1.1) . . .
. . . . .	AP. . .	RFJVERBR,2.(4,1.1) . . .

Die explizite Angabe der Operanden war erforderlich, damit die Adreßbildung hierbei nicht mit dem Basisregister 10, sondern mit dem Register 11 erfolgt.

Auch bei s y m b o l i s c h e r Angabe der Operanden kann erreicht werden, daß zur Adreßbildung das Register 11 herangezogen wird.

Hierzu bedarf es allerdings einer weiteren USING-Anweisung.

Wir codieren wie in Abb. 7 gezeigt.

Dadurch erhalten wir als Ergebnis der Umwandlung die in Abb. 7 gezeigten Maschineninstruktionen (beide Instruktionen haben SS-Format).

Sehen Sie sich die Abbildung genau an. -

Bei der Ausführung des Programms wird bei der PACK-Instruktion zur Bildung der Adressen das Register 11 benutzt.

Bei der AP-Instruktion wird die Adresse des ersten Operanden (RFJVERBR) mit Hilfe des Registers 10, die Adresse des zweiten Operanden mit Hilfe des Registers 11 gebildet.

Wir haben also das gleiche erreicht wie durch explizite Angabe der entsprechenden Operanden.

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35 4
*, V,E,R,B,R,A,U	C,H,S,S,T,A	T,1,S,T,1,K,
	S,T,A,R,T	8,9,5,
A,N,F,A,N,G,	B,A,L,R,	1,Ø,,Ø,
	U,S,I,N,G	*,,1,Ø,
	U,S,I,N,G	E,V,E,R,B,R,,1,1,
	B,	B,1,
E,B,E,R,	D,S,	Ø,C,L,7,7,
E,N,R,	D,S,	C,L,5,
E,V,E,R,B,R,	D,S,	1,2,C,L,6,
D,B,E,R,	D,S,	Ø,C,L,3,4,

Abb. 8

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30 35 4
*, V,E,R,B,R,A,U	C,H,S,S,T,A	T,1,S,T,1,K,
	S,T,A,R,T	8,Ø,Ø,
A,N,F,A,N,G,	B,A,L,R,	1,Ø,,Ø,
	U,S,I,N,G	*,,1,Ø,
	U,S,I,N,G	E,V,E,R,B,R,,1,1,
*, P,R,O,G,R,A,M	M,A,B,L,A,U	F,
B,1,	M,V,C,	D,B,E,R,+1,(3,3,),,D,B,E,R,
A,2,	G,E,T,	K,A,R,T,E,,E,B,E,R,
G,2,	P,A,C,K,	E,V,E,R,B,R,,E,V,E,R,B,R,
	A,P,	R,F,J,V,E,R,B,R,,E,V,E,R,B,R,+2,(4,)

*, D,E,F,I,N,I,T	I,O,N,E,N	
K,C,E,N,D,E,	D,C,	C,',E,N,D,E,'
D,B,E,R,	D,S,	Ø,C,L,3,4,
S,B,E,R,	D,S,	Ø,C,L,1,9,
E,B,E,R,	D,S,	Ø,C,L,7,7,
E,N,R,	D,S,	C,L,5,
E,V,E,R,B,R,	D,S,	1,2,C,L,6,
	E,N,D,	A,N,F,A,N,G,

Abb. 9

Betrachten Sie bitte Ihre Codierung der Aufgabe 14. -

Nach dem bisher Gesagten müßten wir die Codierung, wie in Abb. 8 gezeigt, abändern.

Dadurch würde das Register 11 jedoch nicht nur zur Adreßbildung für EVERBR, sondern auch für alle n a c h EVERBR stehenden Symbole (DBER etc.) verwendet werden. Denn durch USING EVERBR,11. wird dem Umwandlungsprogramm gesagt, daß ab EVERBR das Register 11 zu verwenden ist.

Wir können dies umgehen, indem wir unsere Codierung, wie in Abb. 9 gezeigt, umstellen.

Sehen Sie sich bitte die Abb. 9 genau an. (Der Einfachheit halber wählen wir als Startadresse 800.)

Der Eingabebereich liegt nun am Ende unseres Programms.

Für alle v o r EVERBR definierten Symbole wird bei der Umwandlung der Instruktionen das Register 10 als Basisregister eingesetzt.
Ab EVERBR wird dann das Register 11 benutzt.

Da nach EVERBR keine weiteren Symbole folgen (die als Basisregister auch Register 11 erhalten würden), wird nur für EVERBR das Register 11 eingesetzt.

> Damit unser Programm richtig läuft, ist mehr erforderlich als nur diese Umstellung.
>
> Wie lautet in der Codierung zur Aufgabe 14 die Instruktion, mit der wir die Adresse von EVERBR in das Register 11 bringen?

* * * * *

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–25–30)
	LA	11,EVERBR

Name 1 8		Operation 10 14		Operand und Bemerkungen 20 25 30 35
*.VERBRAU	C	HSSTA	T	ISTIK
		START		8ØØ
ANFANG		BALR		1Ø,Ø
		USING		*,1Ø
		USING		EVERBR,11
*.PROGRAMM		MABLAU	F	
B1		MVC		DBER+1(33),DBER
B2		GET		KARTE,EBER
:		:		:
:		:		:
:		:		:
:		:		:
G2		PACK		EVERBR,EVERBR
		AP		RFJVERBR,EVERBR+2(4)
:		:		
:		:		
:		:		

*.DEFINITI	O	NEN		
KCENDE		DC		C'ENDE'
:		:		:
:		:		:
:		:		:
DBER		DS		ØCL34
:		:		:
SBER		DS		ØCL19
:		:		
EBER		DS		ØCL77
ENR		DS		CL5
EVERBR		DS		12CL6
		END		ANFANG

Abb. 9

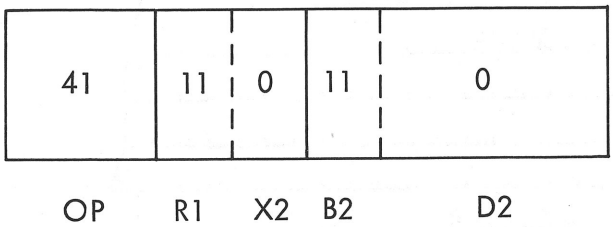
Diese Instruktion ist jedoch nun sinnlos geworden. Die Adresse von EVERBR kann erst dann gebildet werden, wenn das Register 11 die entsprechende Basisadresse schon enthält.

Das Ganze wird deutlich, wenn wir uns ansehen, wie die LA-Instruktion als Maschineninstruktion aussieht. - LA ist eine Instruktion im RX-Format.

```
LA  . . | 11,EVERBR .
```

ergibt umgewandelt:

41	11	0	11	0
OP	R1	X2	B2	D2



Bei der Ausführung dieser Instruktion würde die durch X2, B2 und D2 gebildete Adresse in das Register 11 kommen.

In dem vorliegenden Fall bliebe damit der bisherige Inhalt des Registers 11 (den wir nicht kennen!) erhalten.

Erläuterung:

X2 ist 0 und bedeutet keine Indizierung.

B2 ist Register 11.

D2 ist die relative Adresse 0.

Der Inhalt des Registers 11 wird also um 0 erhöht und ersetzt den bisherigen Inhalt des Registers 11.

Wir müssen also einen anderen Weg einschlagen, um die Adresse von EVERBR in das Register 11 zu bringen. -

Bei der Umwandlung wird eine Symboltabelle erstellt, die alle symbolischen Namen und deren effektive Adressen enthält. - Während der Umwandlung ist somit auch bekannt, welche effektive Adresse EVERBR hat.

Wir benötigen diese Adresse jedoch während der Ausführung des Programms als Basisadresse in Register 11.

Sie wissen, daß bei der Umwandlung die effektive Adresse von EVERBR als Adreßkonstante gespeichert werden kann. - Bei der Ausführung des Programms bringen wir diese Konstante in das Register 11, bevor wir EVERBR ansprechen.

	Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30 – 35)
	*.VERBRAU	CHSSTA	TISTIK
		START	8ØØ
	ANFANG	BALR	1Ø,Ø
		USING	*,1Ø
		USING	EVERBR,11
	*.PROGRAMM	MABLAU	T
	B1	MVC	DBER+1(3.3),DBER
	A2	GET	KARTE,EBER
	:	:	:
	:	:	:
	:	:	:
X		L	11,KAEVERBR
	G2	PACK	EVERBR,EVERBR
		AP	RFJVERBR,EVERBR+2(4)
	:	:	:
	:	:	:
		AH	11,KH6
	:	:	:
	:	:	:

	*.DEFINIT	IONEN	
	KCENDE	DC	C'ENDE'
	:	:	:
X	KAEVERBR	DC	A(EVERBR)
	:	:	:
	DBER	DS	ØCL34
	:	:	:
	:	:	:
	:	:	:
	SBER	DS	ØCL19
	:	:	:
	:	:	:
	:	:	:
	:	:	:
	EBER	DS	ØCL77
	ENR	DS	CL5
	EVERBR	DS	12CL6
		END	ANFANG

Abb. 10

Sie wissen, wie Adreßkonstanten definiert werden.

Wie muß die Anweisung lauten, wenn die Adresse von EVERBR als Konstante (Name: KAEVERBR) zu speichern ist?

* * * * *

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
KAEVERBR	DC	A(EVERBR)

Diese Definitionszeile müssen wir unserer bisherigen Codierung hinzufügen.

Ferner müssen wir die Konstante KAEVERBR in das Register 11 bringen, bevor das erste Verbrauchsfeld gepackt wird. - Dies geschieht mit der Instruktion: _____.

* * * * *

Name	Operation	Operand und Bemerkungen
1 8	10 14	20 25 30
	L	11,KAEVERBR

Die Abb. 10 zeigt im Ausschnitt, wie die Codierung nun aussieht.

	Name (1–8)		Operation (10–14)		Operand und Bemerkungen (20–35)
	*VERBRAUCHSSTATISTIK				
			START		800
	ANFANG		BALR		10,0
			USING		*,10
			USING		EVERBR,11
	*PROGRAMM		MABLAUF		
	B1		MVC		DBER+1(33),DBER
	A2		GET		KARTE,EBER
	:		:		:
	:		:		:
	:		:		:
X			L		11,KAEVERBR
	G2		PACK		EVERBR,EVERBR
			AP		RFJVERBR,EVERBR+2(4)
	:		:		:
	:		:		:
			AH		11,KH6
	:		:		:
	:		:		:

	*DEFINITIONEN				
	KCENDE		DC		C'ENDE'
	:		:		:
X	KAEVERBR		DC		A(EVERBR)
	:		:		:
	DBER		DS		0CL34
	:		:		:
	:		:		:
	:		:		:
	SBER		DS		0CL19
	:		:		:
	:		:		:
	:		:		:
	:		:		:
	EBER		DS		0CL77
	ENR		DS		CL5
	EVERBR		DS		12CL6
			END		ANFANG

Abb. 10

Wiederholung

Betrachten Sie bitte Abb. 10. -

In unserem Programm werden die Register 10 und 11 als Basisregister eingesetzt.

Dabei ist zu beachten, daß das Register 11 nur für EVERBR verwendet wird. Alle übrigen Hauptspeicheradressen werden mit Hilfe des Registers 10 gebildet.

Die Adresse der Adreßkonstanten KAEVERBR wird also mit Hilfe des Registers ____(Z)____ gebildet.

* * * * *

10

KAEVERBR enthält die effektive Adresse von _____. Diese wird als Basisadresse in das Register 11 geladen.

* * * * *

EVERBR

Welche der folgenden Aussagen ist richtig?

Während der Programmausführung

A werden die Inhalte (Basisadressen) der Register 10 und 11 verändert.

B wird der Inhalt des Registers 10 verändert.

C wird der Inhalt des Registers 11 verändert.

D werden die Inhalte der Register 10 und 11 nicht verändert.

* * * * *

C

Entnehmen Sie bitte Ihren Arbeitsmitteln

a) die Speicherbelegung zur Aufgabe 15

b) die Codierung zur Aufgabe 15 (4 Blätter).

Vergleichen Sie bitte diese Unterlagen mit den entsprechenden der Aufgabe 14.

Sie haben in dieser Aufgabe an einem einfachen Problem wie der Verarbeitung nacheinanderfolgender Felder gesehen, daß man zu einer Lösung gelangt, wenn man außer der zu Beginn eines Programms nötigen USING-Anweisung durch eine weitere USING-Anweisung ein zusätzliches Basisregister für die Bildung von zu modifizierenden Adressen bereitstellt.

Die Schwierigkeiten beim Gebrauch mehrerer USING-Anweisungen zumal für den Ungeübten liegen darin, zu entscheiden, welches der bereitgestellten Basisregister für die Adreßbildung eines jeweiligen Feldes herangezogen wird. Über diese Zuordnung von Basisregistern zu den definierten Feldern muß man auf jeden Fall Klarheit haben. Deshalb beschäftigen wir uns jetzt am Beispiel der Aufgabe 15 eingehender mit diesem Problem.

SYMBOLTABELLE

Symbol (Name)	Adresse
KCENDE	1200
KP0	1204
KP12	1205
KP1	1207
KAEVERBR	1208
KH6	1212
RFJVERBR	1214
RFMVERBR	1219
ZAEHLER	1226
SCHNR	1228
SCHJVERBR	1234
SCHMVERBR	1246
DBER	1255
DNR	1263
DJVERBR	1270
DMVERBR	1282
SBER	1289
SNR	1289
SJVERBR	1294
SMVERBR	1302
EBER	1308
ENR	1308
EVERBR	1313

Abb. 1

Symboltabelle - USING-Tabelle

Wir befassen uns jetzt etwas näher mit der Umwandlung des symbolischen Programms in das Maschinenprogramm. - Dabei soll uns nur interessieren, wie zu einem symbolischen Namen das entsprechende Basisregister ausgewählt und die relative Adresse bestimmt wird.

Bei der Umwandlung wird zunächst das ganze Programm eingelesen. Dabei werden alle symbolischen Namen (siehe Programmschema "Name") mit ihren effektiven Adressen in einer Symboltabelle gespeichert.

Nehmen Sie bitte Ihre Speicherbelegung zur Aufgabe 15 zur Hand.

Alle in der Speicherbelegung aufgeführten Namen werden - sofern sie definiert sind - in die Symboltabelle eingetragen.

In unserem Beispiel werden die in Abb. 1 gezeigten Symbole und Adressen in die Symboltabelle aufgenommen. (In die Symboltabelle werden natürlich auch die Namen der Instruktionen aufgenommen. Diese lassen wir hierbei außer Betracht.)

Wir gehen davon aus, daß die Speicherbelegung die tatsächlichen Speicherplätze der einzelnen Felder wiedergibt. - Vergleichen Sie bitte die Eintragungen in der Symboltabelle (Abb. 1) mit denen Ihrer Speicherbelegung.

Nach Erstellung der Symboltabelle werden die symbolischen Instruktionen in der Reihenfolge der Codierung in Maschineninstruktionen umgewandelt. - Hierbei wird für jede symbolische Hauptspeicheradresse in einer Instruktion eine relative Adresse errechnet. Diese wird zusammen mit der Basisregisternummer in die Maschineninstruktion eingesetzt.

Bevor jedoch die erste Instruktion, in der eine Hauptspeicheradresse angesprochen wird, umgewandelt werden kann, muß dem Umwandlungsprogramm mindestens ein Basisregister mit zugehöriger Basisadresse bekannt sein. Dafür müssen wir durch entsprechende USING-Anweisungen sorgen.

SYMBOLTABELLE

Symbol (Name)	Adresse
KCENDE	1200
KP0	1204
KP12	1205
KP1	1207
KAEVERBR	1208
KH6	1212
RFJVERBR	1214
RFMVERBR	1219
ZAEHLER	1226
SCHNR	1228
SCHJVERBR	1234
SCHMVERBR	1246
DBER	1255
DNR	1263
DJVERBR	1270
DMVERBR	1282
SBER	1289
SNR	1289
SJVERBR	1294
SMVERBR	1302
EBER	1308
ENR	1308
EVERBR	1313

Abb. 1

Nehmen Sie bitte die Codierung zur Aufgabe 15 zur Hand. -

Durch die Assemblerinstruktions-Anweisung

| U,S,I,N,G | *,,1,Ø,,,,,,

wird dem Umwandlungsprogramm mitgeteilt, daß das Register
____(Z)____ als _____ zu verwenden ist.

Durch den * wird weiterhin ausgesagt, daß der augenblickliche Stand des Zuordnungszählers (802) als _____ anzunehmen ist.

* * * * *

10

Basisregister

Basisadresse

Daraufhin wird in eine USING-Tabelle das Register 10 mit der Adresse 802 aufgenommen.

Die nächste USING-Anweisung (siehe Programmschema) bewirkt, daß in die USING-Tabelle ferner das Register ____(Z)____ mit der Adresse ____(Z)____ eingetragen wird.

* * * * *

11

1313

(Die Adresse von EVERBR holt sich das Umwandlungsprogramm aus der Symboltabelle.)

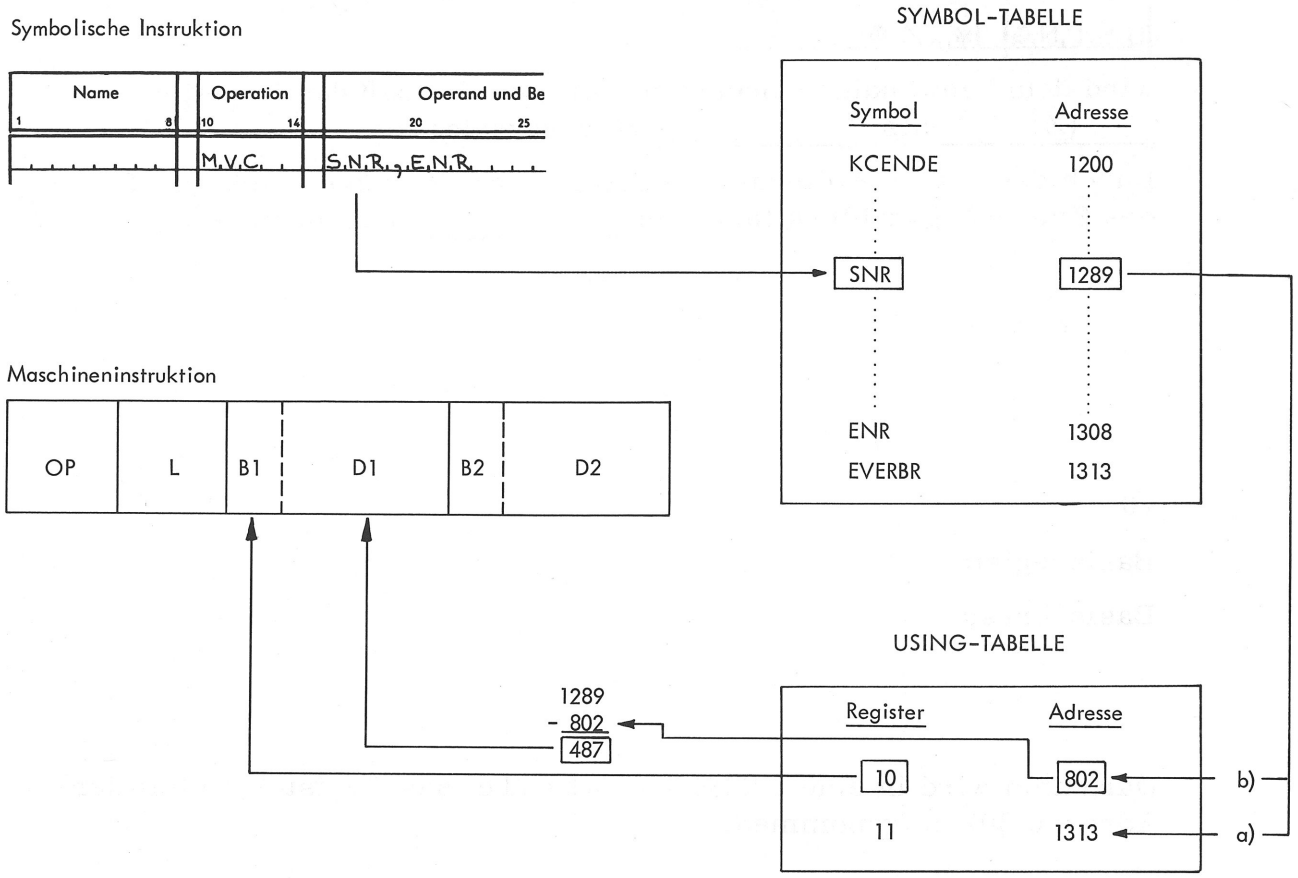

Symbolische Instruktion
Name
Operation
Operand und Be
1
8
10
14
20
25
M,V,C,
S,N,R,,E,N,R,
SYMBOL-TABELLE
Symbol
Adresse
KCENDE
1200
SNR
1289
ENR
1308
EVERBR
1313
Maschineninstruktion
OP
L
B1
D1
B2
D2
USING-TABELLE
1289
- 802
487
Register
Adresse
10
802
b)
11
1313
a)

Abb. 2

Die USING-Tabelle sieht danach wie folgt aus:

Register	Adresse
10	802
11	1313

Es werden nun, in der Codierfolge weitergehend, bei den folgenden Instruktionen die symbolischen Adressen in Maschinenadressen umgewandelt.

Bitte sehen Sie sich die Abb. 2 an. -

Anhand dieser Abbildung verfolgen wir, wie die Umwandlung vor sich geht. - Der besseren Übersicht wegen ist nur die Umwandlung des ersten Operanden (SNR) gezeigt.

Der Ablauf ist wie folgt:

1. Aufsuchen SNR in der Symboltabelle
2. Entnehmen der effektiven Adresse 1289
3. Vergleichen der in der USING-Tabelle angegebenen Adressen mit der Adresse 1289, bis sich gleich oder kleiner ergibt. - Dabei wird bei der höchsten Adresse in der USING-Tabelle begonnen.

   In unserem Beispiel erhalten wir beim zweiten Vergleich (b) "kleiner".
4. Errechnen der relativen Adresse. - Hierbei wird die bei dem Vergleich ermittelte (gleich große oder kleinere) Adresse als Basisadresse genommen.
5. Einsetzen der Registernummer und der relativen Adresse in die Maschineninstruktion.

   Geben Sie bitte an, welche Basisregister und welche relativen Adressen sich für ENR und EVERBR ergeben.

   ENR: Basisregister __(Z)__, relative Adresse __(Z)__.
   EVERBR: Basisregister __(Z)__, relative Adresse __(Z)__.

* * * * *

ENR: Basisregister 10, relative Adresse 506.

EVERBR: Basisregister 11, relative Adresse 0.

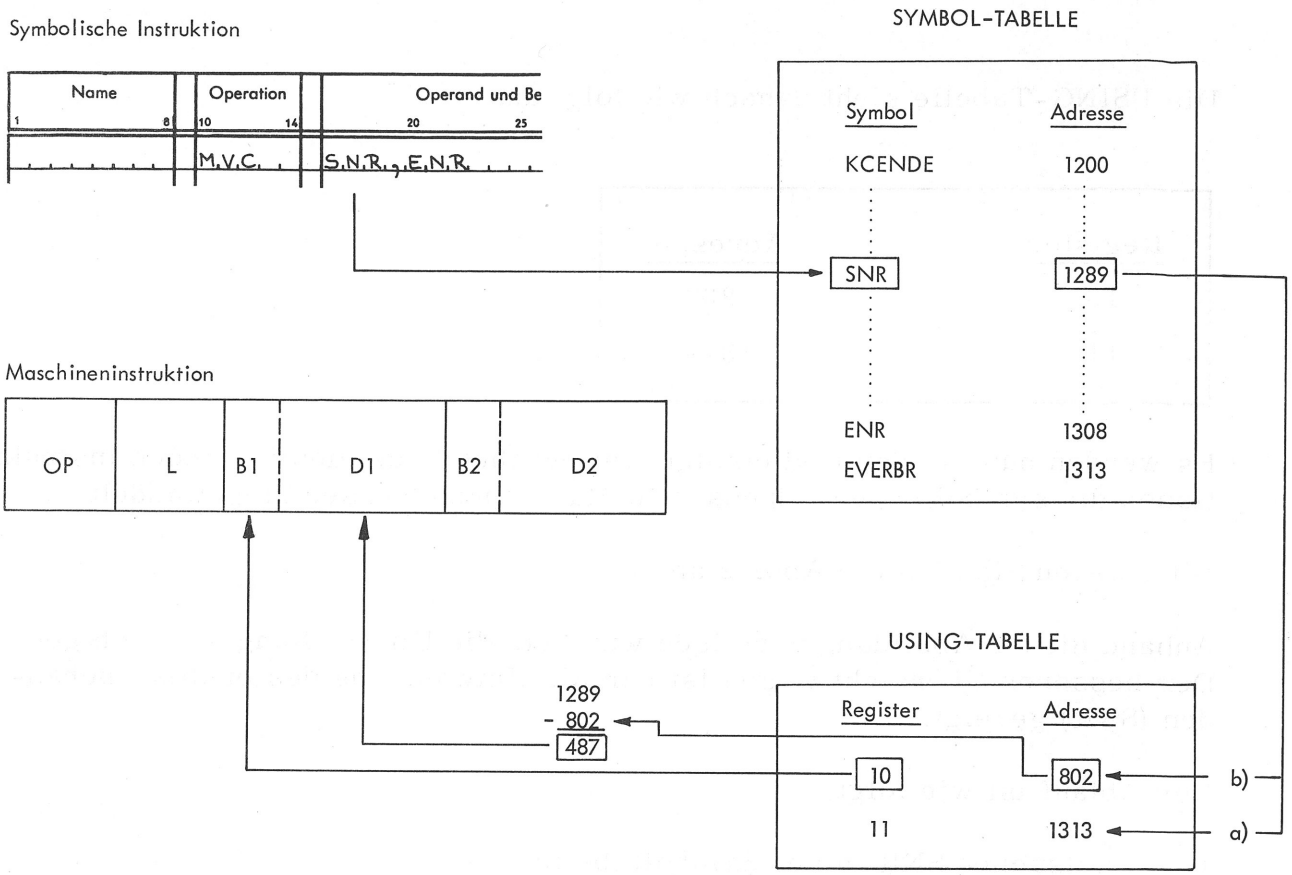


Abb. 2

Name	Operation	Operand und Bemerkungen
*. V.E.R.B.R.A.U.C.H.S.S.T.R.		T.I.S.T.I.K.
	S.T.A.R.T.	8.0.0.
A.N.F.A.N.G.	B.A.L.R.	1.0.,.0.
	U.S.I.N.G.	*.,.1.0.
*. P.R.O.G.R.A.M.M.A.B.L.A.U.F.		
B.1.	M.V.C.	D.B.E.R.+.1.(.3.3.).,.D.B.E.R.
A.2.	G.E.T.	K.A.R.T.E.,.E.B.E.R.
	L.	1.1.,.K.A.E.V.E.R.B.R.
G.2.	P.A.C.K.	E.V.E.R.B.R.,.E.V.E.R.B.R.
*. D.E.F.I.N.I.T.I.O.N.E.N		
K.C.E.N.D.E.	D.C.	C.'.E.N.D.E.'
K.A.E.V.E.R.B.R.	D.C.	A.(.E.V.E.R.B.R.)
E.B.E.R.	D.S.	0.C.L.7.7.
E.N.R.	D.S.	C.L.5.
	U.S.I.N.G.	E.V.E.R.B.R.,.1.1.
E.V.E.R.B.R.	D.S.	1.2.C.L.6.
	E.N.D.	A.N.F.A.N.G.

Abb. 3

Aus Abb. 2 ist auch klar ersichtlich, daß nur für EVERBR das Basisregister 11 benutzt wird, da die Adressen aller anderen Felder kleiner als 1313 (der höchsten Adresse in der USING-Tabelle) sind.

Betrachten Sie bitte Abb. 3. -

Hier ist unsere zweite USING-Anweisung unmittelbar vor der Definition von EVERBR eingefügt. An dieser Stelle aber ist diese USING-Anweisung für unser Programm u n w i r k s a m.

Bei der Umwandlung der PACK-Instruktion (Schritt G2) wird für EVERBR als Basisregister das Register 10 und als relative Adresse 511 (1313-802; siehe vorhergehende Ausführungen) in die Maschineninstruktion eingesetzt.

Versuchen Sie zu begründen, warum dies so ist. (Benutzen Sie dazu ein Extrablatt.)

* * * * *

Zum Zeitpunkt der Umwandlung der PACK-Instruktion enthält die USING-Tabelle nur eine Eintragung: Register 10, Adresse 802. Das Register 11 ist also noch gar nicht in die USING-Tabelle aufgenommen worden. Dies geschieht erst ganz zum Schluß, wenn gar keine Instruktionen mehr umzuwandeln sind. - Deshalb ist die USING-Anweisung unwirksam.

Bei der Ausführung des Programms würde das Register 11 zwar richtig geladen, aber da für EVERBR als Basisregister 10 verwendet würde, würde 12mal das gleiche (erste) Verbrauchsfeld gepackt.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 35)
*.VERBRAUCHSSTATISTIK		
	START	800
ANFANG	BALR	10,0
	USING	*,10
*.PROGRAMMABLAUF		
B1	MVC	DBER+1(33),DBER
A2	GET	KARTE,EBER
	:	:
	:	:
	:	:
	:	:
	LA	11,EVERBR
	USING	EVERBR,11
G2	PACK	EVERBR,EVERBR
	AP	RFJVERBR,EVERBR+2(4)
	:	:
	:	:
	:	:
	:	:
*.DEFINITIONEN		
KCENDE	DC	C'ENDE'
	:	:
	:	:
	:	:
EBER	DS	0CL77
ENR	DS	CL5
EVERBR	DS	12CL6
	END	ANFANG

SYMBOLTABELLE (Auszug)

Symbol	Adresse
KCENDE	1200
⋮	⋮
RFJVERBR	1214
⋮	⋮
EVERBR	1313

Abb. 4

Wir wiederholen noch einmal den sehr wichtigen Tatbestand:

Die USING-Tabelle enthält zu Beginn der Umwandlung der symbolischen Adressen in Nummern von Basisregistern und relativen Adressen nicht bereits sämtliche USING-Anweisungen, sondern sie wird bei dieser Umwandlung erst aufgebaut. In der Codierung nachfolgende USING-Anweisungen werden auch erst nachfolgend in die Tabelle aufgenommen. - Für alle in der Codierfolge vor einer USING-Anweisung stehenden Instruktionen wird also das durch diese USING-Anweisung bereitgestellte Basisregister auf keinen Fall benutzt.

Wir müssen also darauf achten, daß die Register zum richtigen Zeitpunkt in die USING-Tabelle aufgenommen werden, d.h. v o r Bezugnahme auf das Symbol, für das USING gelten soll.

Die Abb. 4 zeigt hierzu ein interessantes Beispiel.

Sehen Sie sich bitte die Abb. 4 genau an. -

In den folgenden Lernelementen untersuchen wir, wie sich die USING-Anweisungen auf die LA-, PACK- und AP-Instruktion auswirken.

Welche Eintragungen enthält die USING-Tabelle zum Zeitpunkt der Umwandlung der LA-Instruktion?

* * * * *

Register 10, Adresse 802

Bei der Umwandlung der LA-Instruktion wird somit als Basisregister das Register ____(Z)____ eingesetzt. Die relative Adresse ist ____(Z)____.

* * * * *

10

511

Name (1–8)	Operation (10–14)	Operand und Bemerkungen (20–35)
*. VERBRAU	CHSSTA	TISTIK
	START	8ØØ
ANFANG	BALR	1Ø,Ø
	USING	*,1Ø
*. PROGRAM	MABLAU	F
B1	MVC	DBER+1(33),DBER
A2	GET	KARTE,EBER
⋮	⋮	⋮
	LA	11,EVERBR
	USING	EVERBR,11
G2	PACK	EVERBR,EVERBR
	AP	RFJVERBR,EVERBR+2(4)
⋮	⋮	⋮
*. DEFINIT	IONEN	
KCENDE	DC	C'ENDE'
⋮	⋮	⋮
EBER	DS	ØCL77
ENR	DS	CL5
EVERBR	DS	12CL6
	END	ANFANG

SYMBOLTABELLE (Auszug)

Symbol	Adresse
KCENDE	1200
⋮	⋮
RFJVERBR	1214
⋮	⋮
EVERBR	1313

Die der LA-Instruktion folgende USING-Anweisung bewirkt, daß das Register ___(Z)___ mit der Adresse ___(Z)___ in die ___ aufgenommen wird.

* * * * *

11

1313

USING-Tabelle

Die USING-Tabelle enthält also nun folgende Eintragungen:

Register	Adresse
10	802
11	1313

Bei der Umwandlung der PACK-Instruktion erhalten wir für EVERBR als Basisregister das Register ___(Z)___ und als relative Adresse ___(Z)___

* * * * *

11

0

Aus der Symboltabelle wird die Adresse von EVERBR (1313) entnommen. Diese Adresse wird anschließend gegen die höchste Adresse in der USING-Tabelle verglichen. Dabei erhalten wir sofort gleich. Die gefundene Adresse wird als Basisadresse zur Errechnung der relativen Adresse verwendet (1313 - 1313 = 0) und als Basisregister wird das dazugehörende Register (11) genommen.

Name (1 – 8)	Operation (10 – 14)	Operand und Bemerkungen (20 – 25 – 30 – 35)
*. VERBRAUCHSSTATISTIK		
	START	800
ANFANG	BALR	10,0
	USING	*,10
*. PROGRAMMABLAUF		
B1	MVC	DBER+1(33),DBER
A2	GET	KARTE,EBER
:	:	:
:	:	:
:	:	:
:	:	:
	LA	11,EVERBR
	USING	EVERBR,11
G2	PACK	EVERBR,EVERBR
	AP	RFJVERBR,EVERBR+2(4)
:	:	:
:	:	:
:	:	:
:	:	:
*. DEFINITIONEN		
KCENDE	DC	C'ENDE'
:	:	:
:	:	:
:	:	:
EBER	DS	0CL77
ENR	DS	CL5
EVERBR	DS	12CL6
	END	ANFANG

SYMBOLTABELLE (Auszug)

Symbol	Adresse
KCENDE	1200
⋮	⋮
RFJVERBR	1214
⋮	⋮
EVERBR	1313

Geben Sie bitte an, welche Basisregister und relativen Adressen sich bei der Umwandlung der AP-Instruktion für RFJVERBR und EVERBR+2 ergeben.

RFJVERBR: Basisregister ____(Z)____, relative Adresse ____(Z)____.
EVERBR+2: Basisregister ____(Z)____, relative Adresse ____(Z)____.

* * * * *

RFJVERBR: Basisregister 10, relative Adresse 412.

EVERBR+2: Basisregister 11, relative Adresse 2.

Erinnern Sie sich noch an die Aufgabe 15. - Wir erreichen auf diese Weise das Gleiche wie in Aufgabe 15 unter Verwendung der Adreßkonstanten.

Vergleichen Sie bitte die beiden Lösungen.

Ihre Arbeitsmittel enthalten ein Merkblatt über die Symbol- und USING-Tabelle (Merkblatt 103). - Darin finden Sie nochmals das bereits gezeigte Beispiel.

Zusammenfassender Überblick über die Lösungen der Aufgaben 13 bis 15

Den Aufgaben 13 bis 15 liegt diesselbe Aufgabenstellung zugrunde, so daß Sie sich eigentlich nur in den dargebotenen Lösungen unterscheiden. Eine kurze Zusammenfassung und Gegenüberstellung dieser Lösungen ist nicht nur der Wiederholung und der vergleichenden Betrachtungen wegen nötig, sondern überwiegend der Wichtigkeit, diese zu kennen.

In der Aufgabe 13 wurden zur Verarbeitung der 12 Verbrauchsfelder für jedes dieser Felder die dafür erforderlichen Instruktionen gesondert codiert. Um die Felder bequem ansprechen zu können, wurde jedem von ihnen ein symbolischer Name gegeben.

Daß diese Lösung wegen des vermehrten Bedarfs an Speicherplatz (weniger infolge der vielen Schreibarbeit) und des bei wesentlich mehr Feldern schon gar nicht mehr vorstellbaren Aufwandes nicht praktikabel ist, bedarf aufgrund der nachfolgend gezeigten Lösungen keiner Erläuterung. - Außerdem würde das bedeuten, von einer weittragenden Möglichkeit der elektronischen Datenverarbeitung - während der Ausführung des Programms Adressen zu modifizieren - keinen Gebrauch zu machen.

In den praktikablen Lösungen der Aufgaben 14 und 15 benutzt man diese Modifikation und erreicht dadurch, daß z. B. das Packen aller 12 Felder mit einer einzigen Instruktion durchführbar ist.
Für die Adreßmodifikation wird ein zusätzliches Register (Register 11) als Basisregister gebraucht, das ausschließlich zur Adreßbildung der 12 Felder herangezogen wird. Zur Adressierung sämtlicher anderer Felder dient das bereits vorgesehene Register (Register 10).

Bei der Adreßmodifikation geht es im wesentlichen um zwei Probleme:

a) Wie erreicht man, daß für einen Operanden ein ganz bestimmtes Register zur Adreßbildung eingesetzt wird?

b) Wie kann man die Anfangsadresse (Adresse des 1. Verbrauchsfeldes) in das Register bringen?

Diese Probleme sind in den Aufgaben 14 und 15 in verschiedener Weise gelöst.

Aufgabe 14

zu a) Durch explizite Angabe des entsprechenden Operanden wird das zu verwendende Register (Register 11) bestimmt.

zu b) Die Anfangsadresse wird mittels der Instruktion LA in das Register geladen.

Aufgabe 15

zu a) Gegenüber der Aufgabe 14 wurde auf die explizite Adressierung verzichtet, um den symbolischen Namen (EVERBR) beibehalten zu können. Dies verlangt eine zweite USING-Anweisung, wodurch das Register 11 zur Adreßbildung der betreffenden Felder in der USING-Tabelle zur Verfügung steht. Diese USING-Anweisung wurde zu Beginn des Programms gegeben.

Damit das Register 11 nur für EVERBR eingesetzt wird, ist EVERBR am Schluß des Programms definiert.

zu b) Die Instruktion LA ist für diese Lösung unbrauchbar, da in ihr bereits das Basisregister 11 verwendet würde. Man speichert deshalb die Adresse von EVERBR als Adreßkonstante und lädt sie mit der Instruktion L in das Register.

In Abänderung dieser Lösung wird die Adreßkonstante überflüssig und die Instruktion LA erfüllt ihren Zweck, wenn man die USING-Anweisung an geeigneter Stelle im Programm gibt. Die USING-Anweisung sollte nach der Instruktion LA, aber noch vor den Instruktionen für die Verarbeitung der Felder in die USING-Tabelle aufgenommen werden, damit für das Laden der Adresse nur das Register 10, für die Verarbeitung aber bereits Register 11 zur Verfügung steht. Das erreicht man, wie gezeigt, durch die Stellung dieser USING-Anweisung nach der Instruktion LA.

Anhand dieser Zusammenfassung sollten Sie festgestellt haben, ob Sie mit den wichtigen Aufgaben 14 und 15 zurechtgekommen sind. - Ist Ihnen irgend eine der Lösungen unklar geblieben, so lesen Sie bitte die entsprechende Aufgabe nochmals durch. Ihr Verständnis ist unbedingte Voraussetzung zum Besuch eines Folgelehrgangs.

Entnehmen Sie bitte Ihren Arbeitsmitteln das Merkblatt 999 (3 Seiten).

Sehen Sie sich das Merkblatt genau an. -

Sie haben sich nun den gesamten Lernstoff dieses Lehrgangs erarbeitet.

Damit haben Sie sich die Grundlagen für die Programmierung des Systems /360 in der Assemblersprache erworben. - Außerdem haben Sie nun die Voraussetzungen für die Teilnahme an weiteren Lehrgängen und für das Verständnis der entsprechenden Literatur.

Damit Sie sich jederzeit, insbesondere zur Vorbereitung auf einen der Folgelehrgänge, über den behandelten Stoff informieren können, verbleiben die Arbeitsmittel bei Ihnen.

Bitte sehen Sie sich vor dem Besuch eines Folgelehrgangs besonders die Merkblätter über Instruktionsformate, Operandenformate und Adressierung samt den zu ihrer Durchführung benötigten Anweisungen BALR und USING an.

Sie sind nun am Ende dieses Lehrgangs angelangt. Überprüfen Sie Ihre Kenntnisse, indem Sie die Abschlußfragen selbständig lösen und mit der vorgegebenen Lösung vergleichen. Wir wünschen Ihnen dazu viel Erfolg.

Die Abschlußfragen und deren Lösung finden Sie in den Arbeitsmitteln.