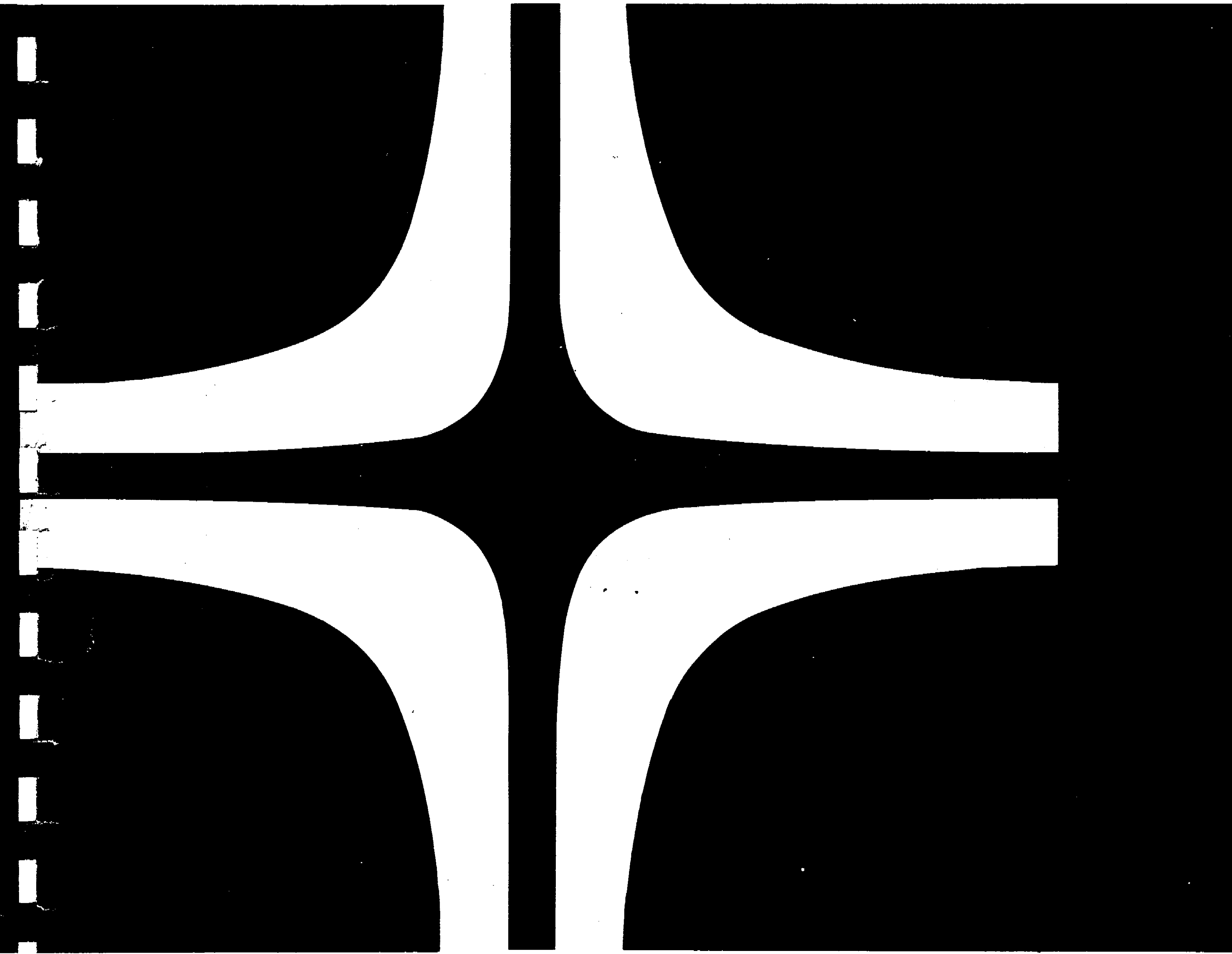
UNIVAC 9000 CARD ASSEMBLER

Programmed Instruction Course

Book 1 - Introduction



SPERRY UNIVAC
COMPUTER SYSTEMS

EDUCATION CENTER

UE-686.1

UNIVAC 9000 CARD ASSEMBLER PROGRAMMED INSTRUCTION COURSE

Introduction to
UNIVAC 9000 Series Computer Programming

Book 1

CONTENTS

INTRODUCTION

This text is the first of a series of programmed instruction manuals designed to teach 9000 Series Card Assembler programming. Successful completion of this text and the self-test evaluation are prerequisites for starting Book 2 of the course.

In this introductory text, the novice acquires the basic computer programming concepts he will need before he begins to learn card assembler language coding.

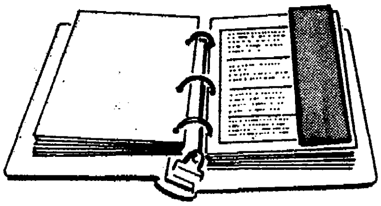
HOW TO STUDY THIS TEXT

This is a programmed text designed for self-study and self-evaluation. Each section is made up of numbered teaching units called frames. An EXPRESS STOP frame precedes each section. This is a self-evaluation frame that permits you to test your knowledge of the material to be covered in the section. If all of your responses to the EXPRESS STOP frame are correct, you may skip to the next EXPRESS STOP frame. If any response is incorrect you are expected to study the frames that follow.

Each EXPRESS STOP frame is followed by a PREVIEW frame. This frame introduces the material to be covered in the subsequent teaching frames. The PREVIEW frame requires no response.

The following types of teaching frames are used:

> Simple response frame (A blank is filled in with a missing word, phrase, or symbol.)
>
> Multiple choice frame (The correct response is selected from two or more alternate responses.)
>
> Matching frame (Corresponding items from two lists are matched.)
>
> Macro frame (A paragraph of text material, covering several related teaching points, is followed by a series of simple response frames that spotlight the teaching points.)



The correct response to each frame is printed in the right-hand margin. Mask the response column with a blank keypunch card as shown in the illustration. Then, check your response with the correct response by lowering the mask as you work down the page.

Note:

EXPRESS STOP frame 1 is a pretest of the material covered in frames 2 through 22. If you do not know this material, do not attempt to guess the correct responses. Skip frame 1 and proceed to frame 2.

1. EXPRESS STOP

Match each of the following basic data processing activities to its corresponding function:

A. Classifying

B. Sorting

C. Calculating

D. Summarizing

E. Recording

______ Information is arranged in a desired numeric or alphabetic sequence.

______ Information is grouped for numeric or alphabetic identification.

______ The results of calculations are written on a physical medium such as a ledger or a check.

______ Addition, subtraction, multiplication, or division is performed.

______ The results of processing are arranged in condensed form.

Match each of the following functions with a corresponding device:

A. Input

B. Output

C. Input/Output

D. Processing

______ Card reader

______ Card punch

______ Magnetic tape unit

______ Magnetic disc unit

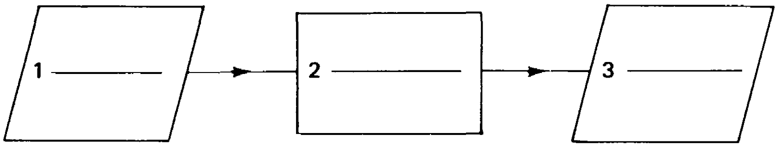
______ Paper tape unit

______ Central processor

______ Printer

______ Console typewriter

Name the three basic data processing functions represented by the symbols below:



B

A

E

C

D

A

B

C

C

C

D

B

C

1. Input
2. Processing
3. Output

1. EXPRESS STOP (Continued)

A standard EAM punched card can record a maximum of:

□ 80 characters.

□ 12 characters.

Data stored on tape is normally arranged in:

□ sequential order.

□ random order.

Data stored on magnetic discs may be arranged in:

□ sequential order only.

□ random order only.

□ sequential or random order.

Which of the following devices provides faster input speed?

□ Card reader

□ Magnetic tape unit

80 characters

sequential order

sequential or random order

Magnetic tape unit

IF YOU ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 23 ON PAGE 1-21.

2. PREVIEW

Data processing is basically a logical problem-solving activity. Whether performed manually by a clerk or automatically by a computer, each step in solving a data processing problem is performed in logical sequence. In the following frames we will study basic data processing procedure and the basic functions performed by computer input, processing, and output devices.

3. Manual data processing is generally performed in the following sequence:

    1. Classifying (Information is grouped for numeric or alphabetic identification.)

    2. Sorting (Information is arranged in a desired numeric or alphabetic sequence.)

    3. Calculating (Processing operations such as addition, subtraction, multiplication, or division are performed.)

    4. Summarizing (The results of calculations are summarized or condensed.)

    5. Recording (Data is written on a physical medium such as a ledger or a check.)

    List the five basic manual data processing activities in the sequence they are generally performed:

    ____

    ____

    ____

    ____

    ____

Classifying

Sorting

Calculating

Summarizing

Recording

4. Although the method of performing these activities is automatic rather than manual, electronic data processing (EDP) also involves classifying, sorting, calculating, summarizing, and ____________ data.

recording

5. Data processing includes three basic functions:

- Input (sales slips, time cards, stock cards, or similar source documents).

- Processing (manipulating data and performing calculations).

- Output (printed reports, checks, etc.).

When sales slips are recorded in a ledger, the sales data represents ______________.

input

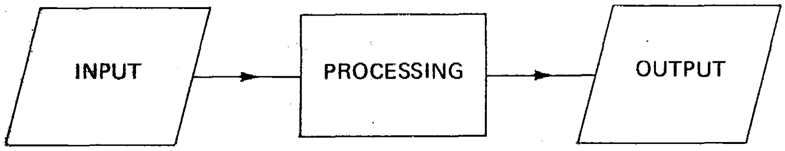
When a clerk computes weekly earnings by multiplying the hourly rate of an employee by the hours worked, this operation represents ______________.

processing

Printing paychecks represents the ______________ function of data processing.

output

6. The three basic data processing functions can be represented by the following simplified diagram.



Whether data processing functions are performed manually or by a computer, the sequence is always the same. Which of the following is the correct sequence?

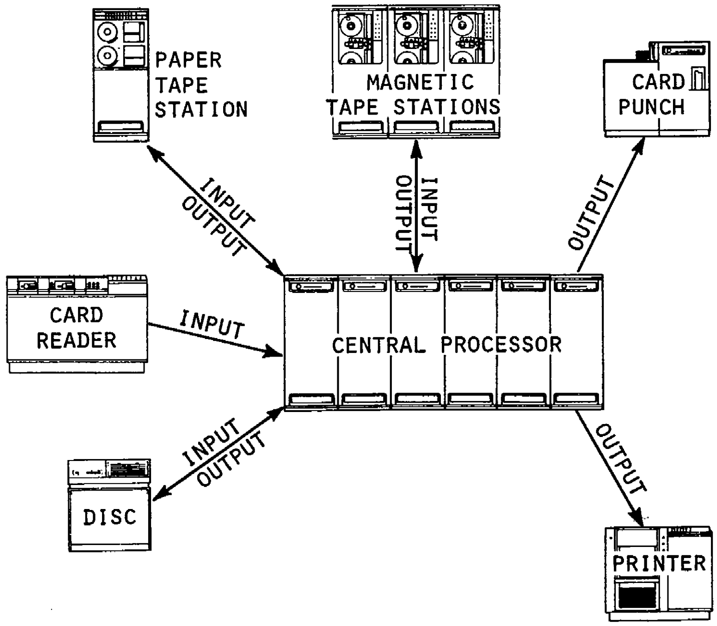
□ Processing, Input, Output

□ Input, Processing, Output

□ Input, Output, Processing

Input, Processing, Output

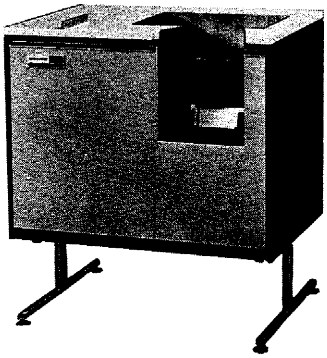
7. A typical electronic data processing (EDP) system including input and output (peripheral) devices is shown below. The peripheral devices are electrically connected directly to the Central Processor.



Refer to the illustration above and check the function(s) of each device listed below:

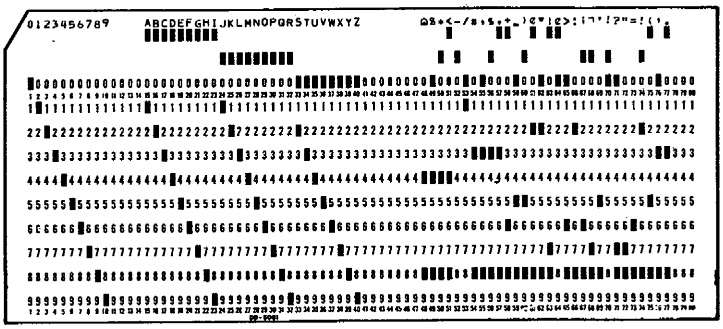
	Input	Output	Processing	
1. Card reader	□	□	□	Input
2. Card punch	□	□	□	Output
3. Magnetic tape units	□	□	□	Input/Output
4. Disc units	□	□	□	Input/Output
5. Central processor	□	□	□	Processing
6. Printer	□	□	□	Output
7. Paper tape unit	□	□	□	Input/Output

8.



Card Reader

The card reader is an input device that recognizes a code punched into cards. It converts the alphabetic, numeric, and special symbols represented by the punched holes into electrical pulse patterns (machine code). The sample punched card below shows the punched-hole code representing numeric, alphabetic, and special characters. This code is called Hollerith code.



A card reader converts:

□ electrical pulses to punched-hole code.

□ punched-hole code to electrical pulses.

The card reader is an:

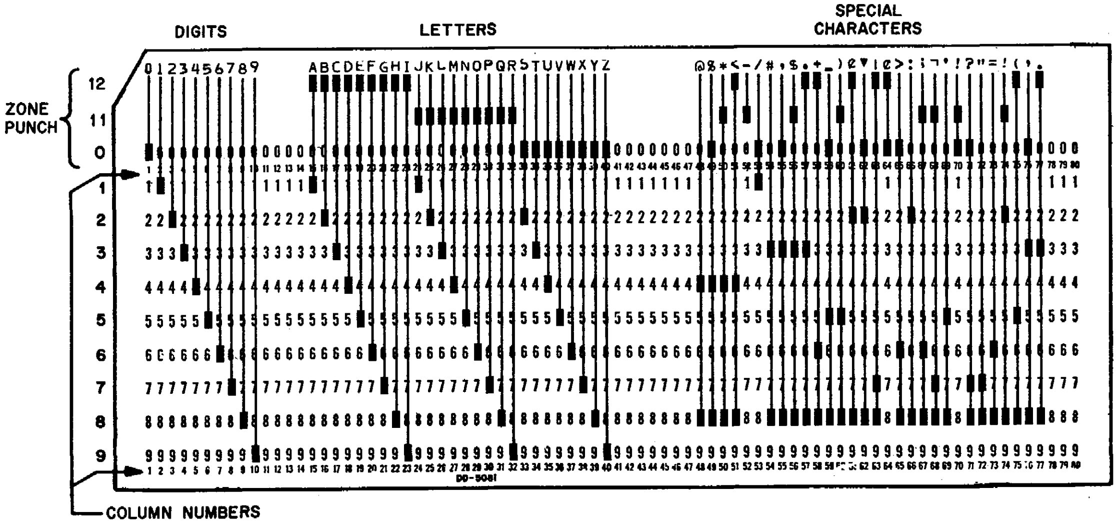
□ input device.

□ output device.

punched-hole code to electrical pulses

input device

9.



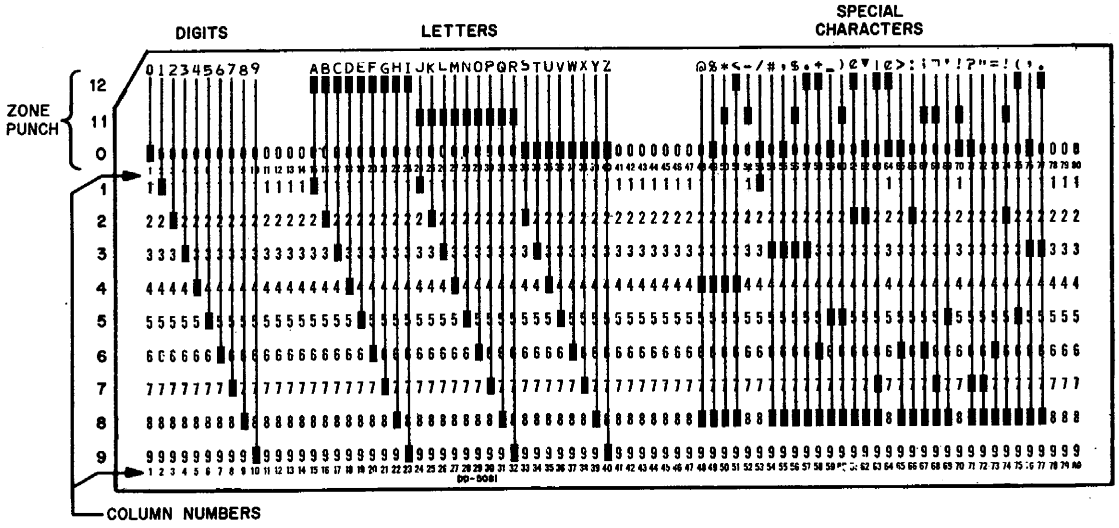
Each numeric, alphabetic, and special character can be represented in Hollerith punched hole code by one, two, or three holes punched in a column as shown in the standard Electrical Accounting Machine (EAM) card above:

A zone punch (12, 11, or 0) is included in the punched-hole code of each alphabetic and special character.

Indicate the zone punch associated with the following characters:

	Zone Punch	Zone Punch
A,B,C,D,E,F,G,H,I	________	12
J,K,L,M,N,O,P,Q,R	________	11
S,T,U,V,W,X,Y,Z	________	0
+	________	12
$	________	11
%	________	0

10.



The standard EAM punched card, as shown above, has 12 horizontal rows and 80 vertical columns. One character can be punched into each column. The maximum number of characters that can be punched into an EAM card is:

□ 80 characters.

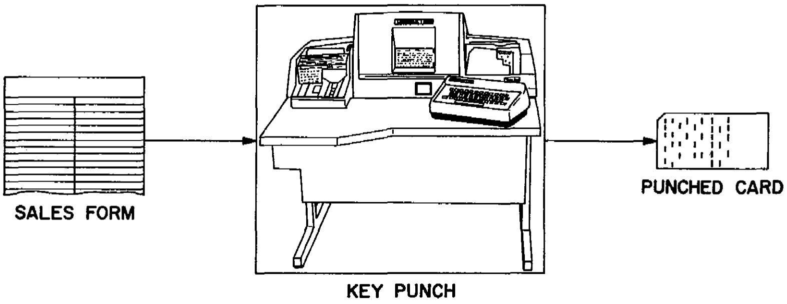
□ 12 characters.

80 characters

Identify the row and column numbers of the holes punched in the above card for each of the following characters:

Character	Row	Column	Row	Column
9	______	______	9	10
A	______	______	12,1	15
J	______	______	11,1	24
S	______	______	0,2	33
$	______	______	11,3,8	56

11. Punched cards are produced by a keypunch machine equipped with a keyboard similar to that of a typewriter. The keypunch operator types source data (sales slips, hours worked, etc.) from a document and produces a punched card as shown in the simplified drawing below. The keypunch is not directly connected to a computer and is thus an off-line device.



Source data is converted to punched hole code by means of a:

□ keypunch.

□ card reader.

The keypunch is used to:

□ prepare source data for input.

□ read output data.

keypunch

prepare source data for input

12.

Card Punch

The card punch is an output device. It operates at a rate of 75 to 200 cards per minute on a column-by-column basis (12 punching positions per column).

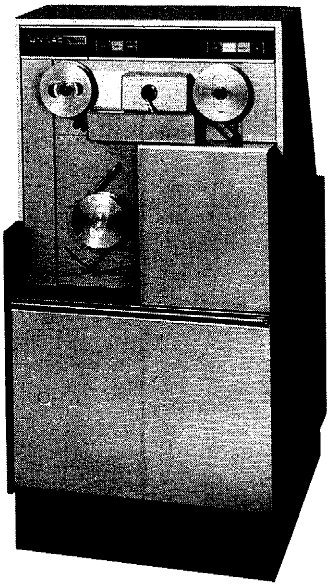
The card punch converts:

□ punched-hole code to machine code.

□ machine code to punched-hole code.

machine code to punched-hole code.

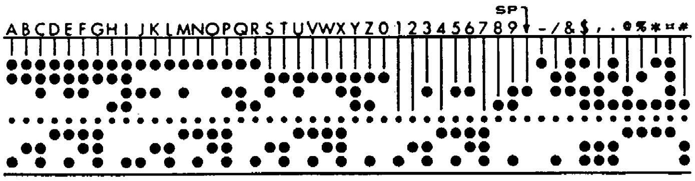
13.



Paper Tape Reader/Punch

The paper tape unit is an input/output (I/O) device. It translates punched-hole code into electrical pulses (computer input) and also converts the electrical pulse output of a computer into punched-hole code. The paper tape is similar to that used in an adding machine.

The segment of punched tape below shows the punched-hole code for numeric, alphabetic, and special characters.


SP
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 - / & $, . @ % * ¤ #

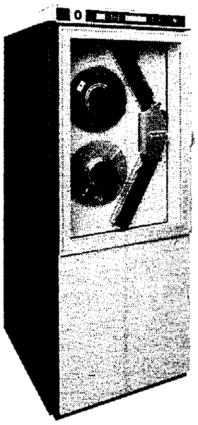
The paper tape unit operates as an <u>input</u> device when it converts:

□ electrical pulses to punched-hole code.

□ punched-hole code to electrical pulses.

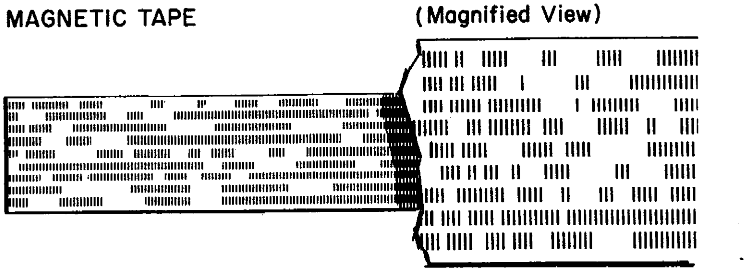
punched-hole code to electrical pulses

14.



Magnetic Tape Unit

The magnetic tape unit shown above is a high speed input/output (I/O) device that provides auxiliary storage on magnetic tape. Information is recorded as magnetized spots on a ferrous oxide coated tape. The tape medium is similar to that used to record sound. Numeric and alphabetic characters are represented by magnetized spot patterns as shown in the simplified drawing of a tape segment below.



Alphanumeric information is recorded on tape just as songs are recorded on tape, one after the other (serially). If the data from 2000 sales slips is recorded on a tape, the magnetized spot patterns representing sales slip No. 1000 can be read and processed:

□ only after all sales slips from No. 1 thru No. 999 are searched sequentially.

□ directly without searching sequentially thru the tape.

Each character in the tape segment above is represented by a pattern of:

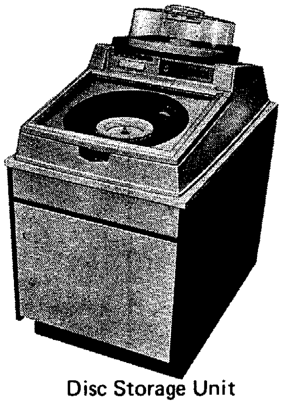
□ magnetized spots.

□ punched holes.

only after all sales slips from No. 1 thru No. 999 are searched sequentially.

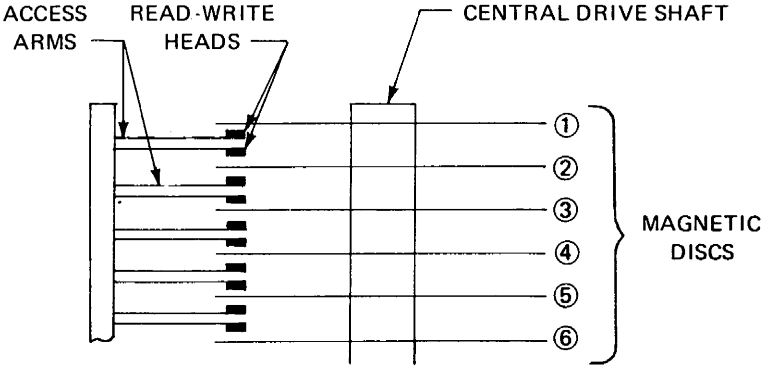
magnetized spots

15.



Disc Storage Unit

The magnetic disc unit is another high-speed input/output (I/O) device. It provides auxiliary storage by means of magnetized spots on ferrous oxide coated metal discs. The discs are vertically stacked in a disc pack like records in a juke box. Each disc surface has 200 concentric tracks on which information is recorded and from which any desired information can be accessed directly by means of multiple read/write heads as shown in the simplified drawing below.



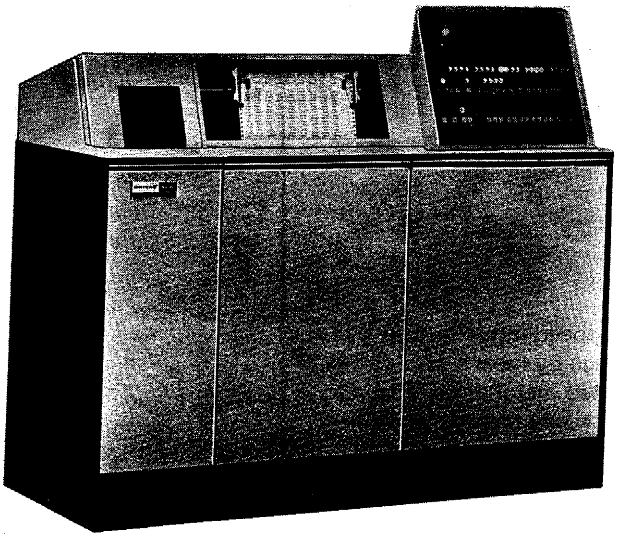
If the data from 20,000 sales slips is written on a disc pack, the magnetized spot pattern representing sales slip No. 16001 can be accessed:

□ directly.

□ only after searching sequentially.

directly

16.



Bar Printer

The bar printer is an output device. The type bar oscillates horizontally in front of the paper. The printing speed is 250 lines per minute with a 63-character font. The paper speed (form skip speed) is 25 inches per second.

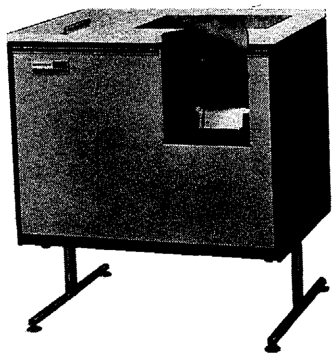
The printer:

□ converts machine code to human-readable language.

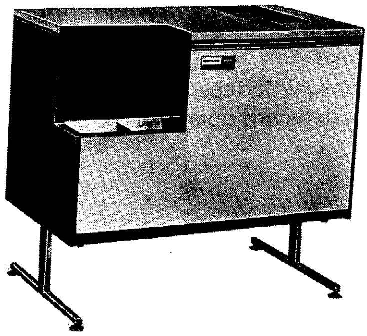
□ converts machine code to punched-hole code.

converts machine code to human-readable language

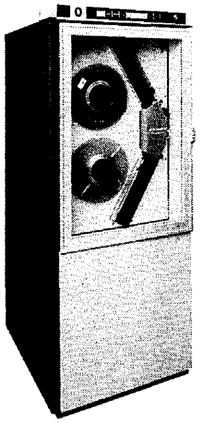
17. Label the peripheral units illustrated below:



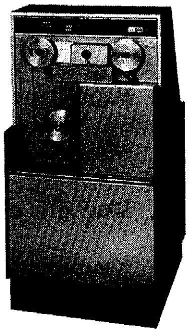
A



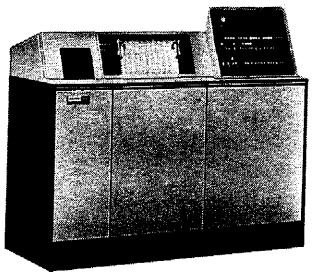
B



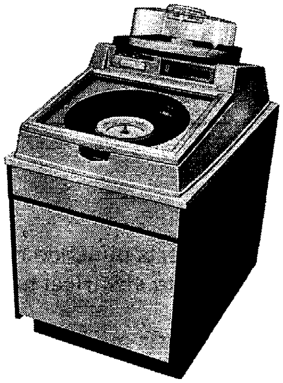
C



D



E



F

A. ____________________

B. ____________________

C. ____________________

D. ____________________

E. ____________________

F. ____________________

Card Reader

Card Punch

Magnetic Tape Unit

Paper Tape Punch/Reader

Printer

Magnetic Disc Unit

18. Input or output operations involving punched cards or punched tape are relatively slow compared with magnetic tape or disc operations. This is due to the speed limitations of electromechanical hole-sensing devices.

Compared with the speed of magnetic tape operation, the input or output speed of punched-card operation is:

□ slow.

□ fast.

slow

19. Match the following:

A. Off-line device	________	Card reader	B
B. Input device	________	Paper tape unit	D
C. Output device	________	Magnetic tape unit	D
D. Input/Output device	________	Magnetic disc unit	D
	________	Keypunch	A
	________	Printer	C
	________	Card punch	C
	________	Console typewriter	D

20. Match the following:

A. Card reader

B. Card punch

C. Paper tape unit

D. Magnetic tape unit

E. Magnetic disc unit

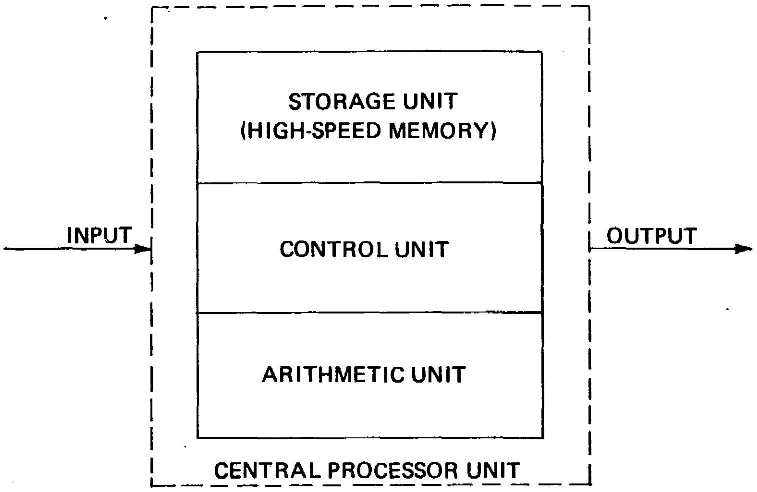
F. Printer

G. Keypunch

H. Console typewriter

________	Converts punched-hole code to electrical pulses.	A,C
________	Converts magnetized spots to electrical pulses.	D,E
________	Converts electrical pulses to punched holes.	B,C
________	Converts source data to punched holes.	G
________	Converts electrical pulses to printed characters.	F,H
________	Converts operator instructions to electrical pulses.	H

21. **The Central Processor Unit (CPU) is the main storage and operational control center of the computer. The CPU consists functionally of three subsections as shown in the following simplified diagram.**



- The Storage Unit provides high-speed memory for storing instructions and data.

- The Control Unit directs and controls input and output functions. It also interprets and executes instructions.

- The Arithmetic Unit performs calculations and such functions as comparing.

The subsections of the Central Processor Unit are the:

____

____

____

Storage Unit

Control Unit

Arithmetic Unit

22. REVIEW

Grouping information in categories for numeric or alphabetic identification is called ________________.

classifying

Arranging data in ascending or descending numeric order for input to a computer is called ________________.

sorting

Multiplying hours worked by the hourly rate is an operation called ________________.

calculating

The totals produced at the end of a monthly payroll report are an example of the ________________ function of data processing.

summarizing

The printing of a payroll check and the attached stub listing gross earnings, deductions, and net earning is an operation called ________________.

recording

Source data read from punched cards is ________________.

input

When a computer operates on data, the operation is called ________________.

processing

When processing is completed, the results are printed or recorded as ________________.

output

Devices electrically connected directly to the Central Processor are referred to as ________________ devices.

peripheral

A keypunch is referred to as an ________________device.

off-line

Punched-hole code in cards can be sensed by a ________________.

card reader

Character-coded holes are punched into cards to represent computer output by a ________________.

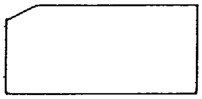
card punch

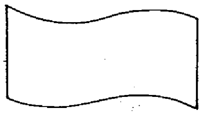
The maximum number of characters that can be punched in a standard EAM card is ________________.

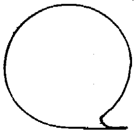
80

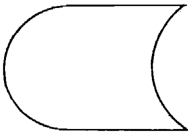
23. EXPRESS STOP

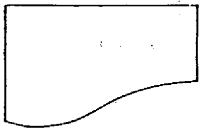
Name the medium represented by each of the following symbols:

 ____________________

 ____________________

 ____________________

 ____________________

 ____________________

A single punched card usually represents one:

- □ field.
- □ record.
- □ file.

A group of punched cards representing data related to the same subject is a:

- □ field.
- □ record.
- □ file.

Punched card

Punched paper tape

Magnetic tape

Magnetic disc

Printed document

record

file

23. EXPRESS STOP (Continued)

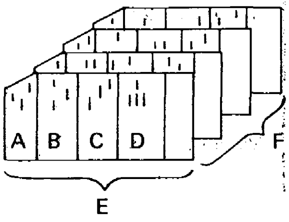
A unit of data in a punched card is a:

□ field.

□ record.

□ file.

Match each letter below to a corresponding designation:



_______ Field

_______ Record

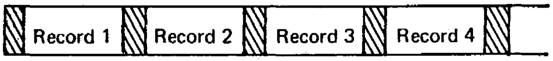
_______ File

Blocked tape format is represented below by:

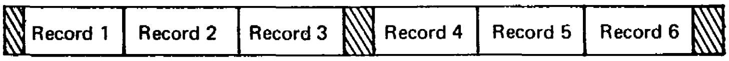
□ tape segment A.

□ tape segment B.

A



B



field

A, B, C, D

E

F

tape segment B

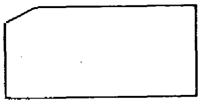
IF YOU ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 42 ON PAGE 1-35.

24. PREVIEW

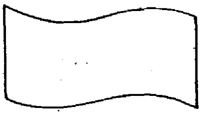
In this section you will learn the symbols that represent data recording media and the devices on which the media are used. The organization of data into fields, records, and files will be discussed. Machine code will be introduced as the binary representation of electrical pulses in patterns of 1's and 0's. External and internal labels for identifying tape files will also be discussed.

25. Each data recording medium is represented graphically in a system flowchart diagram by a standard symbol. Refer to Panel 1 on page 1-135 for a list of media symbols and match each symbol with a corresponding medium.

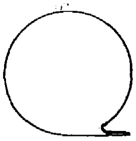
A.



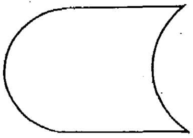
B.



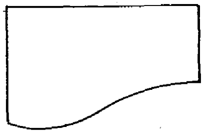
C.



D.



E.



______ Magnetic tape

______ Document

______ Magnetic disc

______ Punched tape

______ Punched card

C

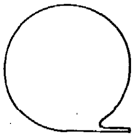
E

D

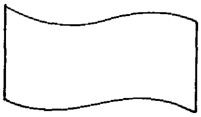
B

A

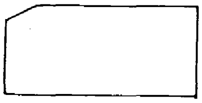
26. Each I/O device is represented in a system flowchart by its corresponding media symbol. Fill in the device name represented by each symbol below:



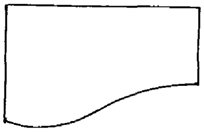
____



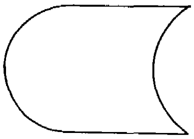
____



____



____



____

Magnetic tape unit

Paper tape unit

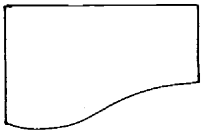
Card reader/Card punch

Printer

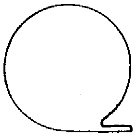
Magnetic disc unit

27. Match each of the following:

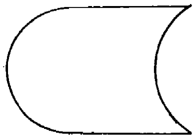
A.



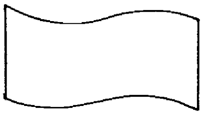
B.



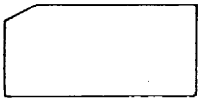
C.



D.

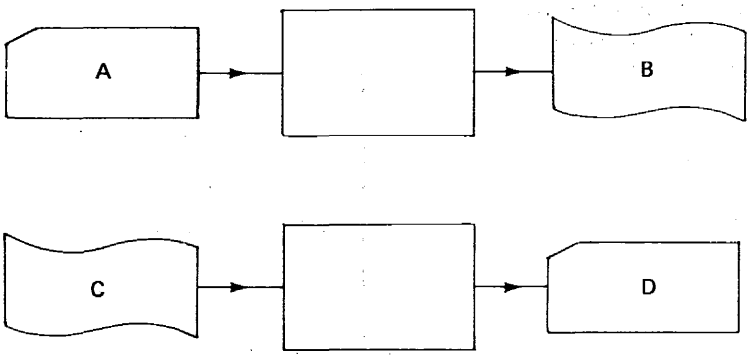


E.



______ Punched card	E
______ Paper tape	D
______ Magnetic tape	B
______ Printed document	A
______ Magnetic disc	C
______ Card punch	E
______ Paper tape punch/reader	D
______ Magnetic tape unit	B
______ Magnetic disc unit	C
______ Printer	A

28.

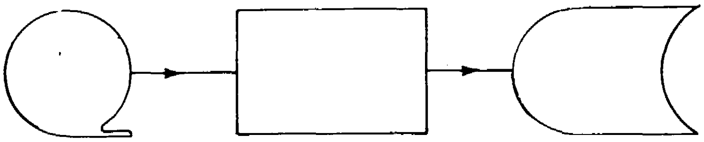


In the flowchart symbols above, select the letters that represent the following devices:

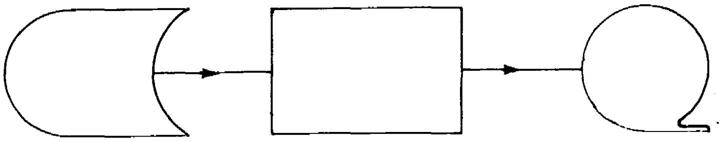
______ Card punch	D
______ Card reader	A
______ Paper tape reader	C
______ Paper tape punch	B

29.

A



B



In the above illustration, magnetic tape is the input medium in:

□ diagram A.

□ diagram B.

diagram A

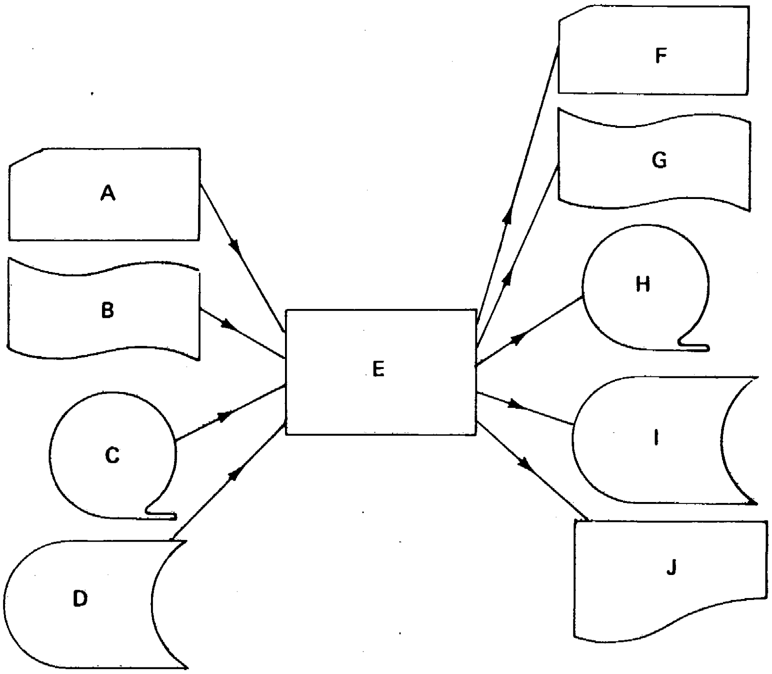
A magnetic disc unit is the input device in:

□ diagram A.

□ diagram B.

diagram B

30.



Name the device represented by each symbol above:

A ______________________________

B ______________________________

C ______________________________

D ______________________________

E ______________________________

F ______________________________

G ______________________________

H ______________________________

I ______________________________

J ______________________________

Card reader

Paper tape input device

Magnetic tape input device

Magnetic disc input device

Central processor unit

Card punch

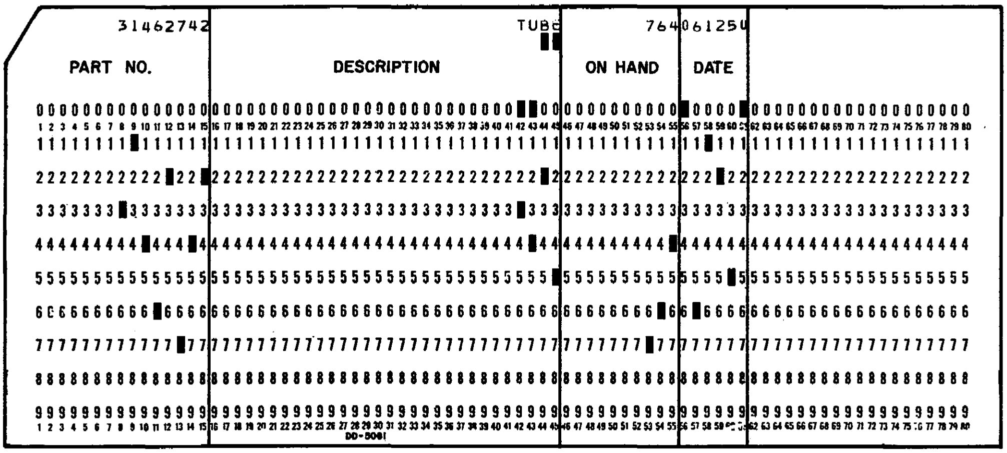
Paper tape output device

Magnetic tape output device

Magnetic disc output device

Printer

31.



The punched card shown above is an inventory stock card. Each unit of data (Part No., Description, On Hand, Date) is called a field. The four fields represent the complete data for this stock item. The complete data punched into this card is called a record.

In the punched card above, the characters that represent the PART NO. form a ________________ and are contained within card columns ________________.

field
1-15

The characters that represent the DESCRIPTION of the stock item form a ________________ within card columns ________________.

field
16-45

ON HAND and DATE are the two remaining fields in the ________________.

record

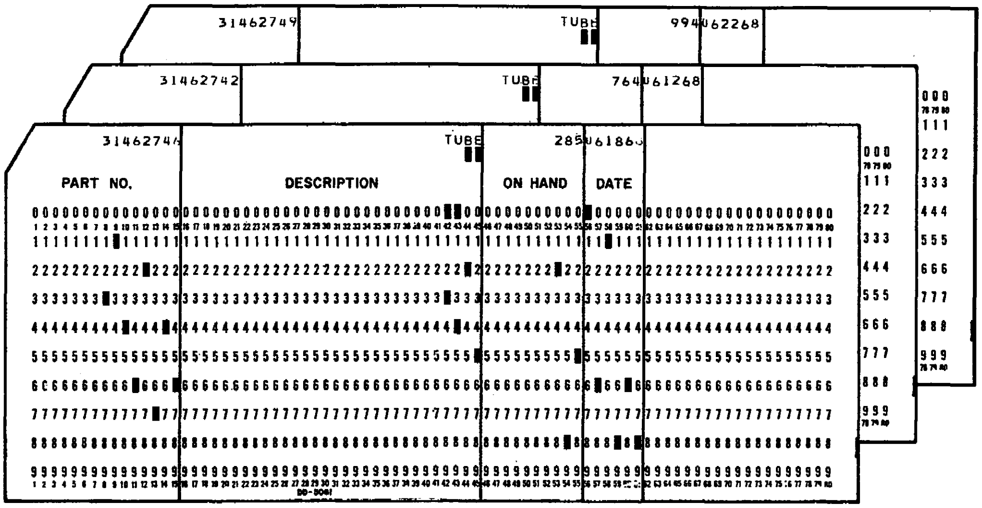
The characters 7 6 4 in the field named ON HAND are punched into columns ______________ .

53, 54, 55

The characters in the field named DATE are punched into columns ________________.

56-61

32.



A group of records representing data related to the same subject forms a file. The three punched-card records above are stock cards from an electron tube inventory ______

□ field.

□ record.

□ file

file

33. Match the following:

A. Field

B. Record

C. File

______ A group of punched cards representing data related to the same subject.

C

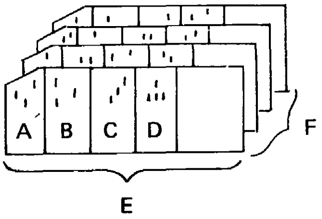
______ A unit of data on a punched card.

A

______ The complete data represented by all the data units in a punched card.

B

34. Match each letter below to a corresponding designation:



_______ Field

_______ Record

_______ File

A,B,C,D

E

F

35. Records and files may be maintained on magnetic tape as well as on cards. A single reel of magnetic tape is generally referred to as a volume and may contain many small files or one large file. A file may be continued from one volume to another.

While card files may be visually identified, tape files cannot be examined visually. To assure adequate volume and file identification of tape files, both external and internal labels are required. An external written label is provided on each container. Internal labels are recorded on tape and checked by the computer. Each internal volume or file label is an 80 character record.

The first label recorded on a volume is a volume (VOL) label record. The data in each file is preceded by a header (HDR) label record and is followed by an end-of-file (EOF) trailer label record.

VOL	HDR	INVENTORY FILE C	EOF

In the above simplified drawing of a single file stored in one volume:

The first record in the volume is a ______________ label record.

The second record in the volume is a ______________ label record.

The first record in the file is a ______________ label record.

The last record in the file is an ______________ label record.

VOL

HDR

HDR

EOF

36. When a file is stored in two volumes, an end of volume (EOV) trailer label designates the end of the first volume and indicates to the computer that the file is continued on volume 2.

Volume 1 of 2.

VOL1	HDR1	INVENTORY FILE A	EOV1

Volume 2 of 2.

VOL2	HDR1	INVENTORY FILE A (CONTINUED)	EOF1

Volume 1 above is terminated by label record:

□ EOF1.

□ EOV1.

EOV1

Volume 2 is terminated by label record:

□ EOF1.

□ EOV1.

EOF1

If a file requires two volumes to store all the data, how many label records will be required?

□ Two

□ Four

□ Six

Six

37.

VOL1	HDR1	FILE A	EOF1	HDR2	FILE B	EOF2

How many files are stored in the above volume? ___________

Two

How many label records are there? ______________________

Five

How many header and trailer label records will be written on a volume that stores three files?

□ Two

□ Five

□ Seven

Seven

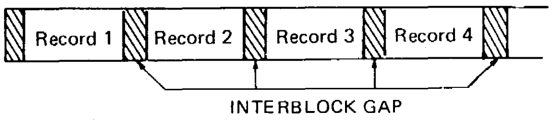
38. Records on magnetic tape are structured and organized as follows:

- Records may be fixed or variable in length.

- Records may be blocked or unblocked. (A magnetic tape block is a string of characters between interblock gaps. An interblock gap is a section of blank tape with no recorded data.)

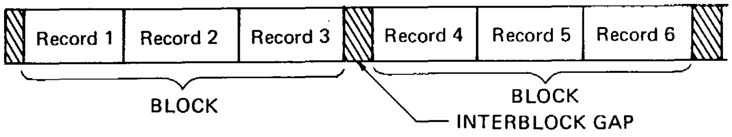
Fixed-Length Records, Unblocked

A



Fixed-Length Records, Blocked

B



The records in tape segment A above are:

□ blocked.

□ unblocked.

Each block in tape segment B above'is:

□ separated by an interblock gap.

□ not separated by an interblock gap.

The records in A and B are:

□ fixed-length.

□ variable-length

unblocked

separated by an interblock gap

fixed-length

39.
- Punched-card files are not labeled internally since they can be identified visually.
- A punched-card file is treated as an unlabeled file contained on a single volume.
- The end of a punched-card file is indicated by an end-of-file (EOF) card punched with special characters designated for this purpose.

A punched-card file is treated as:

□ an unlabeled file.

□ a labeled file.

an unlabeled file

Labels are used to identify a:

□ punched-card file.

□ magnetic tape file.

magnetic tape file

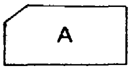
The end of a punched-card file is indicated by:

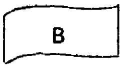
□ an end-of-file (EOF) card.

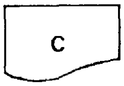
□ a trailer label.

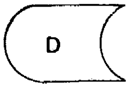
EOF card

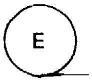
40.











Which of the above symbols represents an output device only?________________

C

41. **REVIEW**	
A unit of data on a punched card is called a ____________.	field
The complete data on a punched card is called a ____________.	record
A group of punched cards containing data related to the same subject is called a ____________.	file
A single reel of magnetic tape is called a ____________.	volume
The first record in a volume is called a ____________ label.	volume
Each file in a volume is preceded by a ____________ label.	header
Each file in a volume is followed by an ____________ label.	EOF
The end of an intermediate volume in a multivolume file is indicated by an ____________ label.	EOV
The end of a volume that stores one or more complete files is indicated by an ____________ label.	EOF
The standard length of a label record is ____________ (how many) characters.	80
Are punched-card files labeled?____________	No
The end of a punched-card file is indicated by an ____________ card.	EOF

NOTES:

42. EXPRESS STOP	
Binary digits are called _______________.	bits
Binary code consists of _____________ (how many) symbols.	two
The presence of a pulse in computer input represents a:	
□ 0 bit.	
□ 1 bit.	1 bit
High-speed memory is made up of:	
□ plated wires.	piated wires
□ data.	
□ instructions.	
Instructions are interpreted and executed by:	
□ the arithmetic unit of a CPU.	
□ the control unit of a CPU.	the control unit of a CPU
□ the storage unit of a CPU.	
A program is a logical sequence of:	
□ data.	
□ instructions.	instructions
The 9200/9300 programmer writes instructions in:	
□ machine code.	
□ symbolic code.	symbolic code.

42. EXPRESS STOP (Continued)

A character is represented in memory by:

□ one bit.

□ four bits.

□ eight bits.

A character stored in memory:

□ requires no addressable location.

□ will have an addressable location.

7	3	1
1000	1001	1002

The numbers 1000, 1001, and 1002 above represent:

□ content of memory.

□ addressable locations in memory.

Name the two parts of an instruction:

____

____

Mnemonic code is used to represent:

□ an operand.

□ an operation code.

eight bits

will have an addressable location

addressable locations in memory

Operation code

Operand

an operation code.

IF YOU ANSWERED ALL OF THE ABOVE QUESTIONS CORRECTLY, SKIP TO FRAME 62 ON PAGE 1-47.

43. PREVIEW

Information is stored in computers in a binary form called machine code. This will be discussed in relation to bits, bytes, and characters. We will also discuss addressing, the parts of an instruction, and symbolic language.

44. Just as the language of telegraphy consists of patterns of dots and dashes, the language of computers consists of patterns of ones (1's) and zeros (0's). In computer (machine) code each numeric, alphabetic, or special character is represented by a unique coded pattern of 1's and 0's. Computer code is a binary language based on the binary numbering system, which consists of the two digits 1 and 0. Each binary digit is called a bit. (The word bit is a contraction of the two words binary digit).

The decimal numbering system is based on ten digits. The binary numbering system is based on:

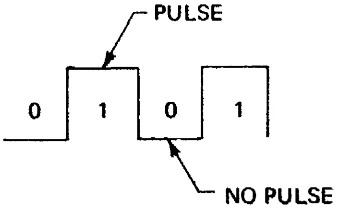
- □ one digit.
- □ two digits.
- □ three digits.

two digits

A binary digit is called a ______________.

bit

45. When character coded punched holes in a punched card are converted into patterns of electrical pulses, each pulse pattern read into the computer is internally stored in memory as a pattern of 1's and 0's. A 1 bit is stored when a pulse is present. A 0 bit is stored when no pulse is present. For example, the pulse pattern shown below corresponds to the binary bit pattern representing the decimal number 5. (We will study the binary numbering system later.)



The presence of a pulse corresponds to a:

□ 0 bit.

□ 1 bit.

1 bit

The absence of a pulse corresponds to a:

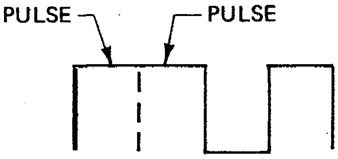
□ 1 bit.

□ 0 bit.

0 bit

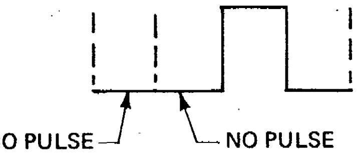
46. The binary bit configuration 1101 is represented by the pulse pattern below:

A



The binary bit configuration 0010 is represented by the pulse pattern below:

B



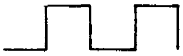
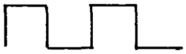
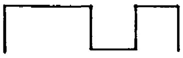
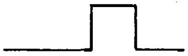
The pulse pattern in illustration A above represents which of the following bit configurations?

□ 0010

□ 1101

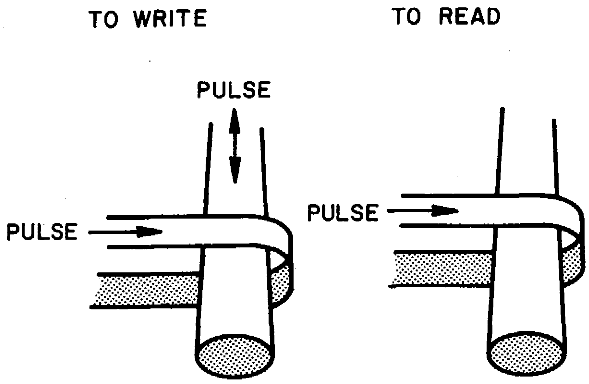
1101

47. Match each of the following bit patterns to its corresponding pulse pattern:

A. 0101	_____		A
B. 1010	_____		B
C. 0010	_____		D
D. 1101	_____		C

48. The UNIVAC 9000 series of computers use integrated circuits and plated-wire memory units. A characteristic of the plated wire memory is the non-destructive read-out. This feature accelerates the memory access time because the bits of information do not have to be restored after reading.

A memory unit is made up of nine planes. Eight planes hold data bits; the ninth plane holds parity bits. Each position of memory consists of a nine-bit unit called a byte. In this course, we will not be concerned with parity bits.



A memory unit is made up of:

□ 7 data bits

□ 8 data bits

□ 9 data bits

8 data bits

Each plane in storage can magnetically store:

□ a 0 bit only.

□ a 1 bit only.

□ either a 1 bit or a 0 bit.

either a 1 bit or a 0 bit

49. Both data and instructions are stored in memory in binary code. An alphabetic character is represented by eight bits. For example, the letter A is represented in memory as 11000001. (This is Extended Binary-Coded-Decimal Interchange Code and will be discussed later.)

Each alphabetic character in memory is stored in:

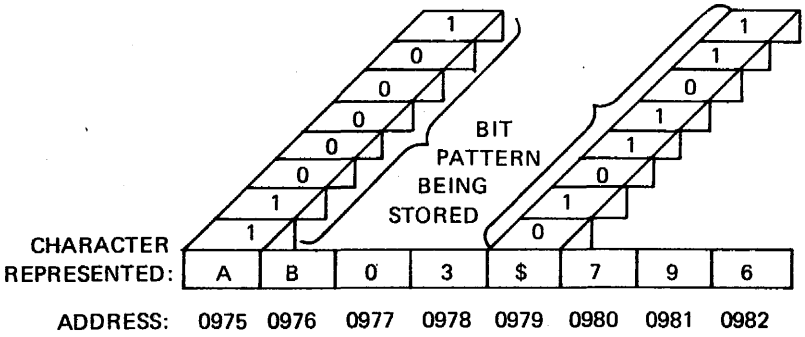
□ one bit

□ four bits

□ eight bits

eight bits

50. When stored in memory, each character has an addressable location as shown in the diagram below.



In the above diagram:

The address of the character A is ________________ .

0975

How many bits are required to store the character A?

____

eight

The number 0982 represents the:

□ content of a memory location.

□ address of a memory location.

address of a memory location

The address of the character $ in storage is__________.

0979

51. An eight-bit character stored in memory is called a byte.

1	1	0	0	0	0	0	1

BYTE

The character A, shown above, is stored in memory as a ________________.

Each byte of stored data occupies ____________ (how many) bits in memory.

byte

eight

52.

4	3	2
2100	2101	2102

The numbers 2100, 2101, and 2102 above represent:

- □ content of memory.
- □ addressable locations in memory.

The numbers 432 above represent:

- □ content of memory.
- □ addressable locations in memory.

addressable locations in memory

content of memory

53.
- Operation of a computer is automatically directed by a program.
- A program is a specified sequence of instructions written by a programmer to operate on data to solve a problem.
- Each instruction defines an operation to be performed and defines the location of the data or specifies a device to be used.

Computer operation is normally directed:

□ automatically by a program.

□ manually by an operator.

A program is a specified sequence of:

□ data.

□ instructions.

An instruction (check one or more):

□ specifies an operation.

□ defines the storage location of the instruction.

□ specifies the data to be used.

□ defines the storage location of the data.

□ can specify a device to be used.

automatically by a program

instructions

specifies an operation

defines the storage location of the data

can specify a device to be used

54. Instructions are written by the 9200/9300 programmer in:

□ machine code.

□ symbolic code.

Instructions are stored in memory in:

□ machine code.

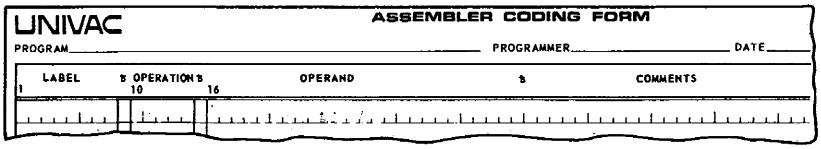
□ symbolic code.

symbolic code

machine code

55.

Form A



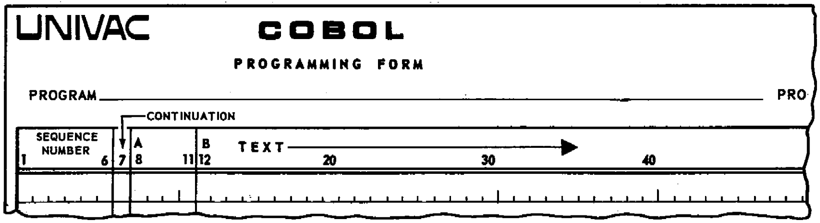
UNIVAC ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

LABEL b OPERATION b OPERAND b COMMENTS

1 10 16

Form B



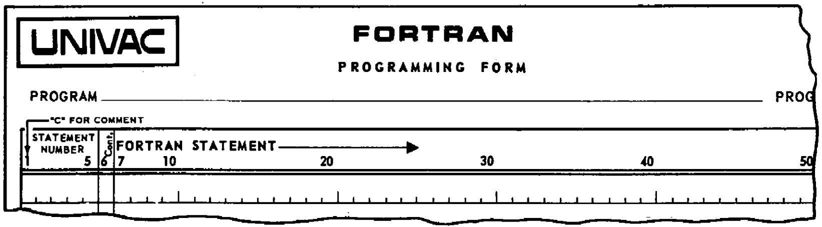
UNIVAC COBOL PROGRAMMING FORM

PROGRAM ______ PRO

CONTINUATION

SEQUENCE NUMBER 1 6 | 7 | A 8 11 | B 12 TEXT 20 30 40

Form C



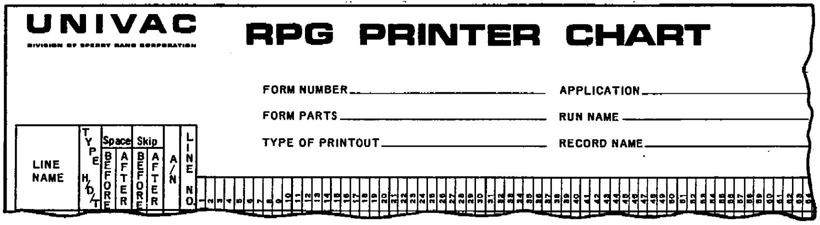
UNIVAC FORTRAN PROGRAMMING FORM

PROGRAM ______ PROG

"C" FOR COMMENT

STATEMENT NUMBER 1 5 | Cont. 6 | FORTRAN STATEMENT 7 10 20 30 40 50

Form D



UNIVAC DIVISION OF SPERRY RAND CORPORATION RPG PRINTER CHART

FORM NUMBER ______ APPLICATION ______

FORM PARTS ______ RUN NAME ______

TYPE OF PRINTOUT ______ RECORD NAME ______

LINE NAME | TYPE H/D/T | Space BEFORE AFTER | Skip BEFORE AFTER | A/N | LINE NO.

Instructions may be coded in any one of several symbolic languages available to the programmer. Commonly used symbolic languages include the following:

Assembly Language

COBOL (Common Business Oriented Language)

FORTRAN (Formula Translation) language

RPG (Report Program Generator) language

Name the symbolic language used with each of the coding forms illustrated above:

Form A ______________________

Form B ______________________

Form C ______________________

Form D ______________________

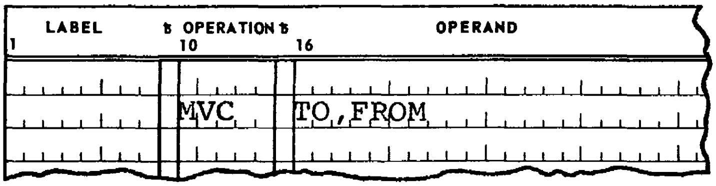
Assembly language

COBOL

FORTRAN

RPG

56. Instructions direct computers to read, write, edit, move data, compare, branch, add, subtract, multiply, divide, etc. Such operations are written in mnemonic code. (Mnemonic is pronounced "nē-mon΄-ik" and means easy to remember.) In the coding example below, the mnemonic MVC means MoVe Character.



```
LABEL      OPERATION   OPERAND
1          10          16
           MVC         TO,FROM
```

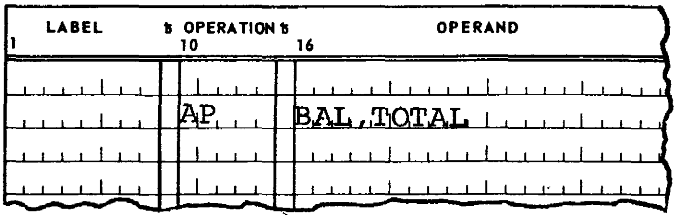
The mnemonic operation code for an Assembly language Move Character instruction is ________________.

MVC

57. • The two parts of an instruction include the operation code and one or more operands.

• The operation code specifies the operation to be performed.

• An operand defines the storage address (location) of data in memory.



```
LABEL      OPERATION   OPERAND
1          10          16
           AP          BAL,TOTAL
```

In the above Add Packed Decimal instruction the mnemonic AP is the ____________ code.

operation

The symbolic names (tags) BAL,TOTAL represent the addresses of data in the above instruction and are called ________________.

operands

The operation code AP is written in ______________ code.

mnemonic

58. The operation code specifies:

□ the address of the instruction.

□ the operation to be performed.

An operand defines:

□ the address of an instruction.

□ the address of data or specifies a device.

the operation to be performed

the address of data or specifies a device

59. Match the following:

A. Program

B. Instruction

C. Operation code

D. Operand

________ A set of coded instructions.

________ Specifies operation to be performed and defines address of data or specifies a device.

________ Defines storage address of data or specifies a device.

________ Specifies operation to be performed.

A

B

D

C

60. The symbolic representation of an operation code is written in ________________ code.

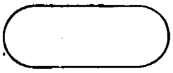
mnemonic

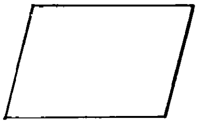
61. REVIEW

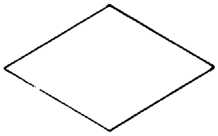
A binary digit is called a ________________.	bit
Binary code consists of ________________(how many) symbols.	two
The presence of a pulse represents a ________________ bit.	1
The absence of a pulse represents a________ bit.	0
An alphabetic character is represented by ________________ (how many) bits in memory.	eight
An eight-bit character configuration is called a ____________.	byte
Instructions and data are internally stored in the high-speed ________________unit of the CPU.	memory
The solution of a problem by a computer is automatically directed by a ________________.	program
A program is a specified sequence of ________________ .	instructions
Instructions are written by a programmer in ______________ code.	symbolic
The operation to be performed by an instruction is specified by the ________________.	operation code
The address of data is defined by an ________________.	operand
The operation code of an instruction is written in ________________code.	mnemonic

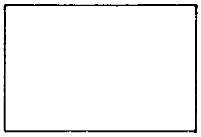
62. EXPRESS STOP

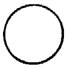
Label the following flowchart symbols.


____


____

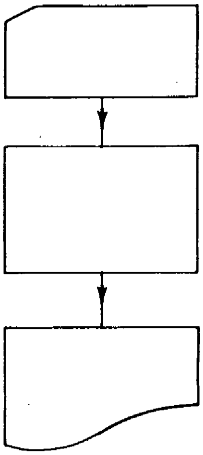

____


____

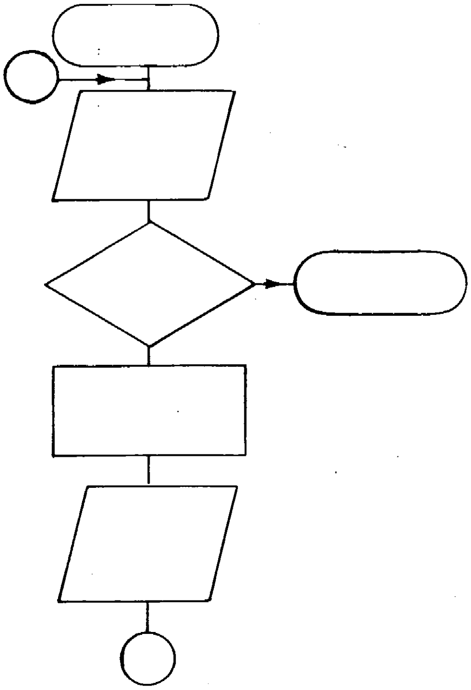

____

Which diagram below is a process flowchart? ______________

A



B



Terminal

Input/Output

Decision

Process

Connector

Diagram B

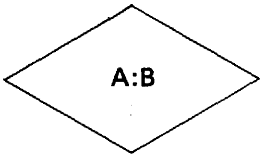
62. EXPRESS STOP (Continued)

The decision block below illustrates that:

□ A is compared to B.

□ A is less than B.

□ A is greater than B.



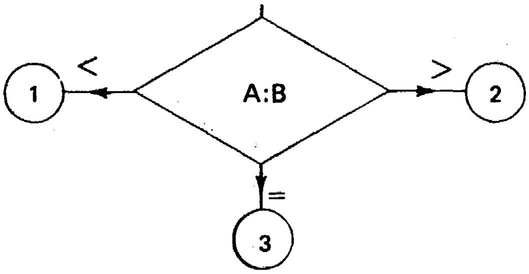
____

Assume in the decision block below that A is greater than B. The flow path will then branch to:

□ ①

□ ②

□ ③



____

Label the following symbols.

: ______________________

≠ ______________________

> ______________________

< ______________________

≤ ______________________

≥ ______________________

A is compared to B

②

Compare

Not equal

Greater than

Less than

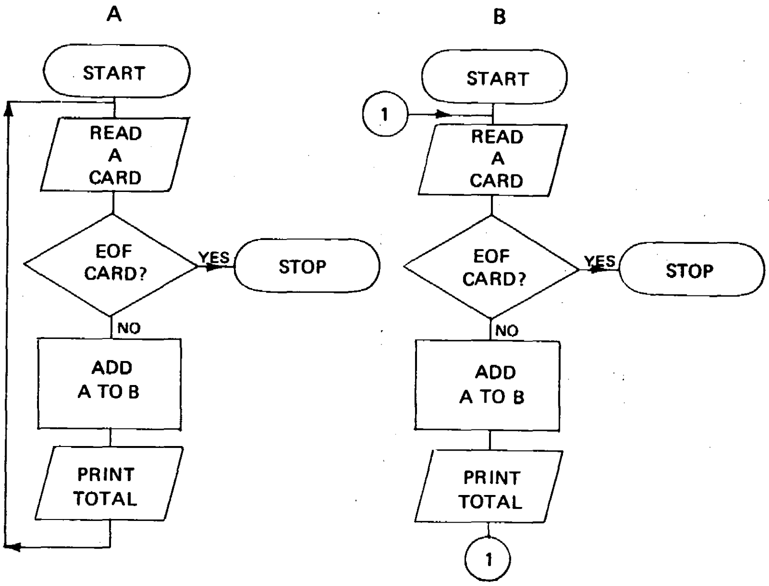
Equal to or less than

Equal to or greater than

62. EXPRESS STOP (Continued)

Which flowchart below provides for processing to be repeated until an EOF card is read?

- □ A only
- □ B only
- □ A and B



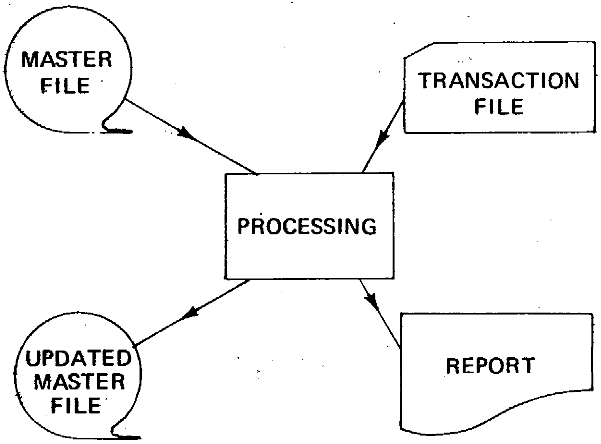
A and B

IF YOU ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 80 ON PAGE 1-63.

63. PREVIEW

In this section we will introduce flowcharting as a tool used in the preliminary phases of the solution of a data processing problem.

64. Just as a layout drawing of a house is prepared by an architect to illustrate the general plan, the systems analyst prepares a system flowchart that illustrates the EDP problem, defines the input and output, and specifies the media. An example of a simplified system flowchart is shown below. (Refer to the media symbols in Panel 1 on page 1-135).



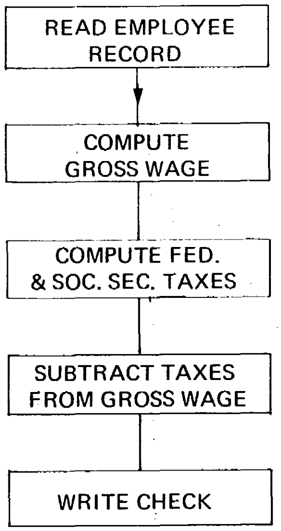
The input and output symbols in the above system flowchart represent:

□ media.

□ processing steps.

media

65. The system flowchart illustrates the problem, defines the input and output, and specifies the media. How the problem is to be solved is the responsibility of the programmer. The basic processing requirements are generally planned first by the programmer at the manual level as shown in the block diagram of a simplified payroll problem below.


READ EMPLOYEE RECORD
COMPUTE GROSS WAGE
COMPUTE FED. & SOC. SEC. TAXES
SUBTRACT TAXES FROM GROSS WAGE
WRITE CHECK

The illustration above is a:

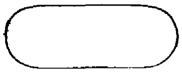
□ system flowchart.

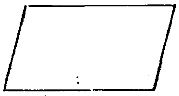
□ block diagram.

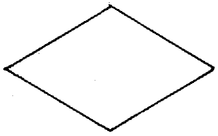
block diagram

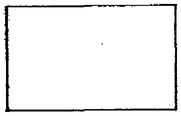
66. When the block diagram has been checked for completeness and accuracy, the programmer is ready to develop the process flowchart. The process flowchart illustrates the step-by-step coded operational instructions the computer will be directed to perform. The standard symbols used to construct the process flowchart represent processing steps and are shown in Panel 1 on page 1-135.

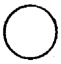
Refer to Panel 1 and label the following symbols.

 ____________________

 ____________________

 ____________________

 ____________________

 ____________________

Terminal

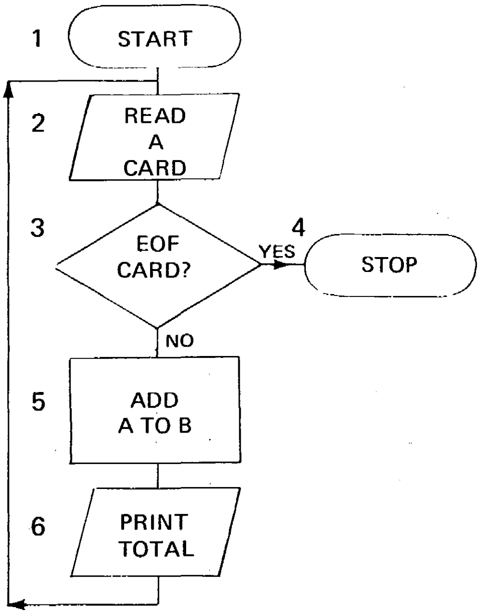
Input/Output

Decision

Process

Connector

67. The following process flowchart illustrates a simple processing problem.



Name the processing step represented by each numbered symbol in the above flowchart.

1. ______________________

2. ______________________

3. ______________________

4. ______________________

5. ______________________

6. ______________________

The above diagram is:

□ a system flowchart.

□ a process flowchart.

Terminal

Input

Decision

Terminal

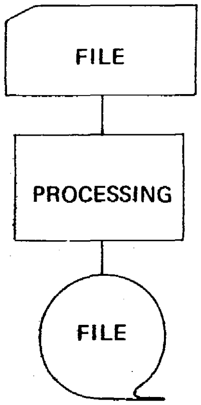
Process

Output

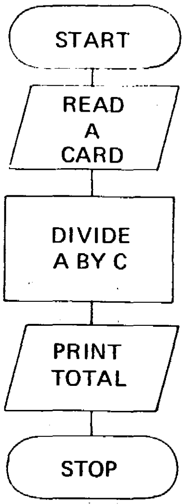
a process flowchart

68.

A



B



Flowchart A above is a:

- □ system flowchart.
- □ process flowchart.

The symbols used in flowchart B represent:

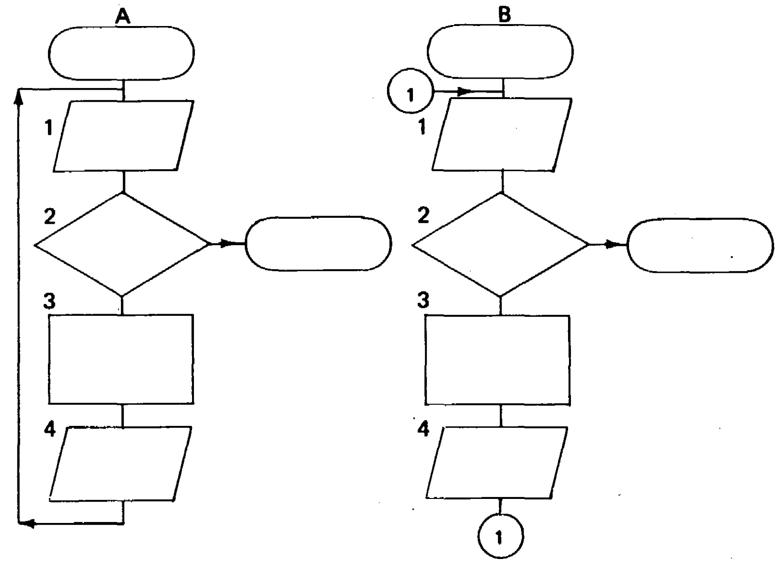
- □ media.
- □ processing steps.

system flowchart

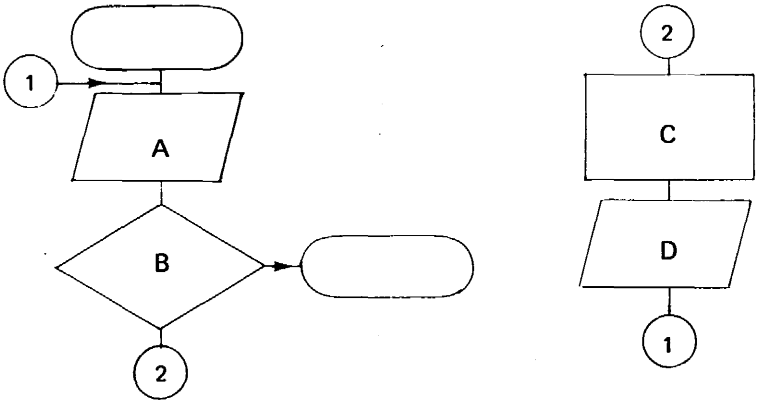
processing steps

69. **A loop is used to indicate that the process described in a flowchart is automatically repeated** until a terminating condition halts the processing. The connector symbol ○ is commonly used in flowcharts to avoid the need for drawing a loop. The loop shown between blocks 4 and 1 in flowchart A below is replaced in flowchart B by two ________________.

connectors



70. Connector symbols also are used where space limitations require a flowchart to be sectionalized as shown below.



In the above sectionalized process flowchart, the program continues from block B to:

- □ block A.
- □ block C.

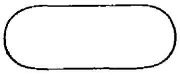
block C

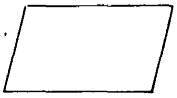
The program loops from block D to:

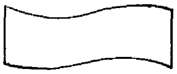
- □ block A.
- □ block C.

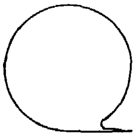
block A

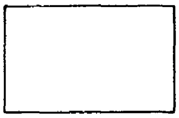
71. Match the following:

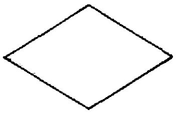
A 

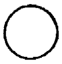
B 

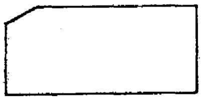
C 

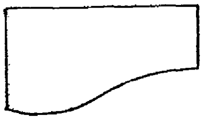
D 

E 

F 

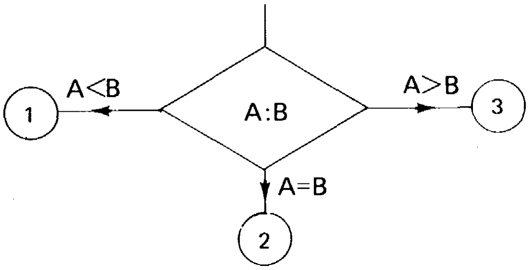
G 

H 

I 

______ Terminal	A
______ Connector	G
______ Decision	F
______ Punched card	H
______ Punched tape	C
______ Document	I
______ Magnetic tape	D
______ Input/Output	B
______ Process	E

72.



In the above decision block, the value of A is compared to the value of B. One of three conditions is possible:

A is greater than B (A>B).

A is less than B (A<B).

A is equal to B (A=B).

The symbol > above indicates that A is:

□ greater than B.

□ less than B.

greater than B

The symbol < above indicates that A is:

□ greater than B.

□ less than B.

less than B

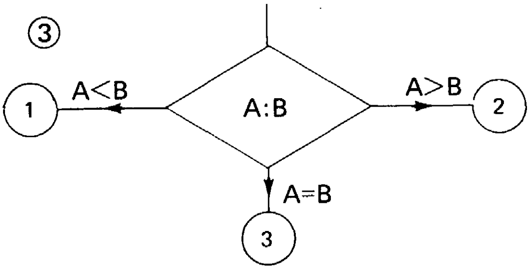
73. In the diagram below we will assume that A is greater than B. Thus the flow path will branch to:

□ ①

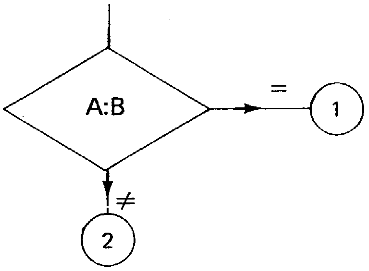
□ ②

□ ③

②



74.



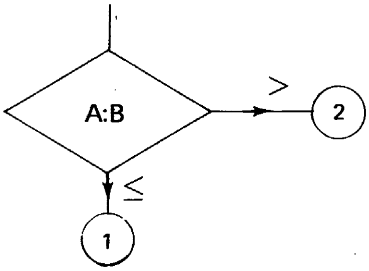
The symbol ≠ above means not equal. If A is not equal to B, the flow path will branch to:

□ ①

□ ②

②

75.



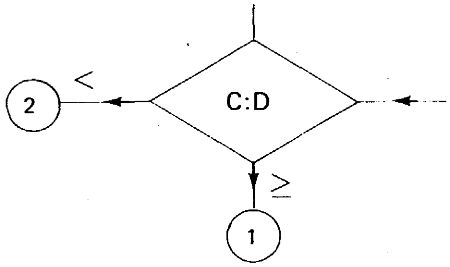
The symbol ≤ above means less than or equal to. If A is less than or equal to B, the flow path will branch to:

□ ①

□ ②

①

76.



The symbol ≥ above means greater than or equal to. If C is greater than or equal to D, the flow path will branch to:

□ ①

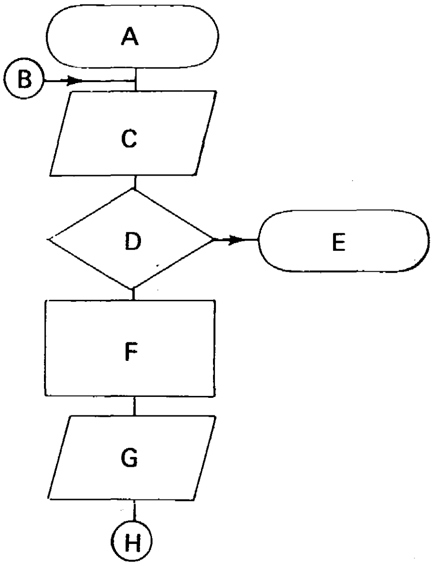
□ ②

①

77. Match the following:

A. S>T	____ S is less than T.	B
B. S<T	____ S is greater than T.	A
C. S≤T	____ S is less than or equal to T.	C
D. S≥T	____ S is compared to T.	E
E. S:T	____ S is not equal to T.	F
F. S≠T	____ S is greater than or equal to T.	D

78.



Match each symbol in the above flowchart with the corresponding symbol name.

________ Input

________ Decision point

________ Process

________ Output

________ Terminal

________ Connector

Match each symbol in the above flowchart with a corresponding processing step:

________ START

________ STOP

________ EOF CARD?

________ ADD A TO B

________ READ A CARD

________ PRINT RESULT

C

D

F

G

A,E

B,H

A

E

D

F

C

G

79. REVIEW

Illustrating the problem, defining the input and output, and specifying the media to be used are the responsibilities of the ________ analyst.	systems
The processing steps required to solve a problem can be planned at the manual level by means of a ________ diagram.	block
The detailed process flowchart is prepared by the ________.	programmer
The symbols used in a system flowchart represent ________.	media
The symbols used in a process flowchart represent ________.	processing steps
Connectors are used to represent a ________.	loop or a continuation
A test point to determine the direction of a branch in the flow path is illustrated in a flowchart by a ________.	decision block
The symbol > represents ________.	greater than
The symbol < represents ________.	less than
The symbol ≥ represents ________.	greater than or equal to
The symbol ≤ represents ________.	less than or equal to
The symbol : represents ________.	compare
The symbol ≠ represents ________.	not equal
The symbol ○ represents ________.	a connector

NOTES:

80. EXPRESS STOP

Assume that the letters A through E will be read into an input area that contains data from a previous program as shown below. What characters will be stored in this area after the new data is read into memory?

BEFORE

E	*	J	*	R

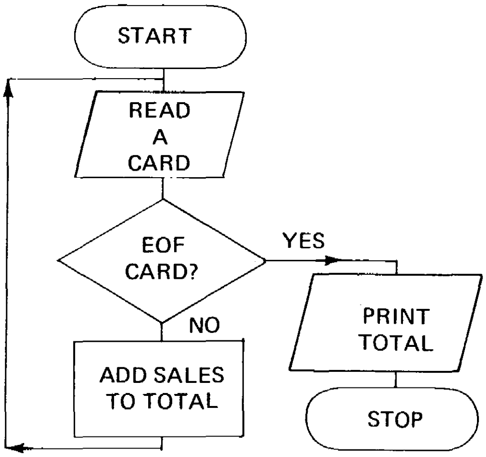
AFTER

____

Does the program represented by the flowchart below require a housekeeping operation?

□ Yes

□ No



How many times will the PRINT operation be performed?

____

When will the PRINT operation be performed?

____

A	B	C	D	E

Yes

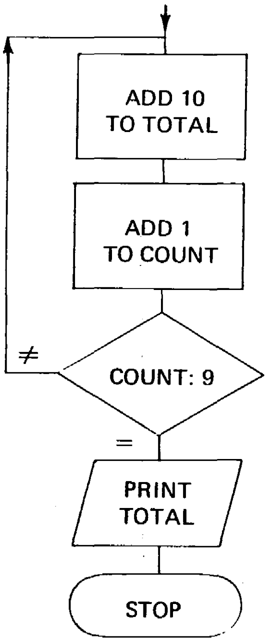
Once

When the last card is read.

80. EXPRESS STOP (Continued)

Assume that COUNT has been set to zero in the program represented by the flowchart below. How many times will the ADD 1 TO COUNT operation be performed?____________

Nine times

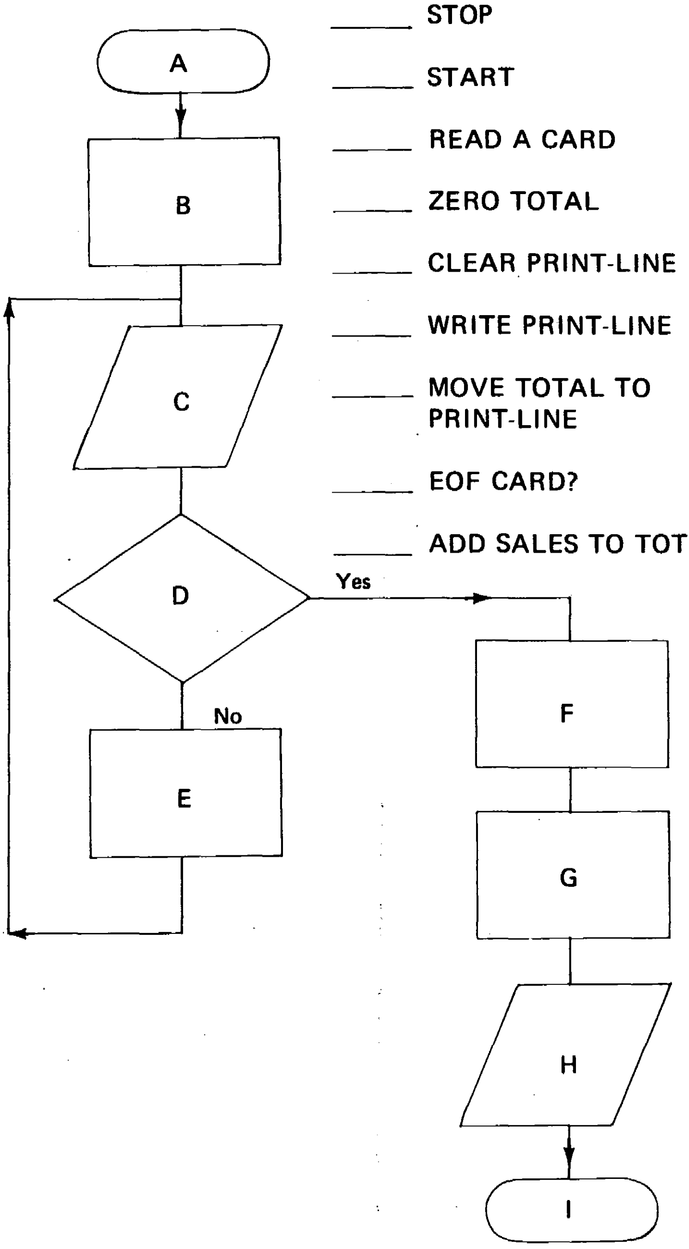


As shown in the above flowchart, how many times will the ADD 10 TO TOTAL operation be performed?__________

Nine times

80. EXPRESS STOP (Continued)

Match each processing step listed below to the appropriate symbol in the following flowchart:



_____ STOP	I
_____ START	A
_____ READ A CARD	C
_____ ZERO TOTAL	B
_____ CLEAR PRINT-LINE	F
_____ WRITE PRINT-LINE	H
_____ MOVE TOTAL TO PRINT-LINE	G
_____ EOF CARD?	D
_____ ADD SALES TO TOTAL	E

IF YOUR ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 102 ON PAGE 1-83.

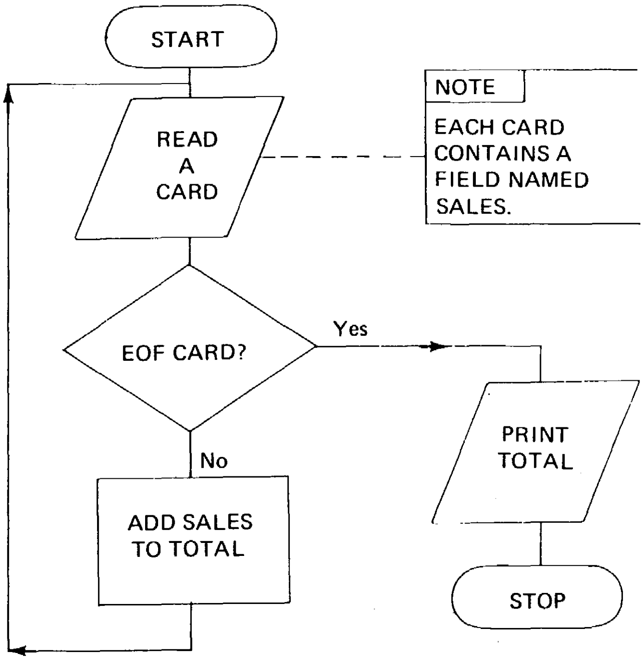
81. PREVIEW

A process flowchart must contain all of the information that a programmer will need to write a usable program. Usually each step to be coded is represented by one or more symbols.

Some of the symbols represent data manipulation activities. Others represent operations that are required by the machine. Housekeeping activities such as setting counters or clearing output areas are typical examples of operations that are necessary. These operations are not usually obvious from the statement of a problem or from the system flowchart. Yet the programmer must know when these activities are required and the exact point at which they must be included in the flowchart.

We will examine flowcharting techniques and machine considerations in the following frames.

82. Carefully examine the following flowchart.



You have not had an opportunity to read the statement or specification of the problem for the above flowchart. However, you should be able to follow the logic and answer the following questions.

What input device will be required? ______________________

What output device will be required? ______________________

Will all of the cards be tested for an "EOF CARD" condition?

____

How many times will the Print operation be performed?

____

When will the Print operation be performed?

____

Card reader

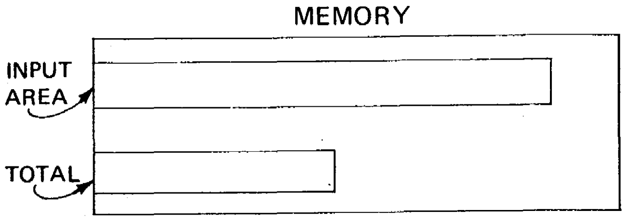
Printer

Yes

Once

When the EOF card is detected.

83. The flowchart in the preceding frame implies that the programmer must write code that will reserve two storage areas: One input area will receive the sales data from a card, and one area will accumulate the sum in the field named TOTAL.



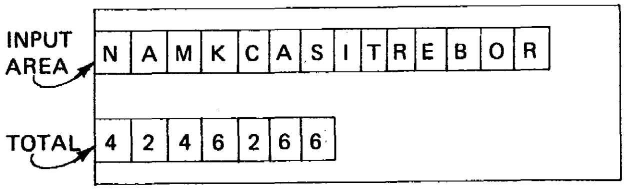
Is the programmer's task of reserving the storage areas illustrated in the flowchart?

□ Yes

□ No

No

Before the first card is read into the input area, information is contained in this location from the previous program as shown in the simplified illustration below. When the first card is read into memory, the previous contents of memory are overlayed and destroyed. Similarly the data from the first card is overlayed and destroyed when the second card is read into memory.



When new data is read into memory, it appears in the above location designated:

□ Input Area.

□ TOTAL.

Input Area

When processing is performed, the resulting sum will be accumulated in the location designated:

□ Input Area.

□ TOTAL.

TOTAL

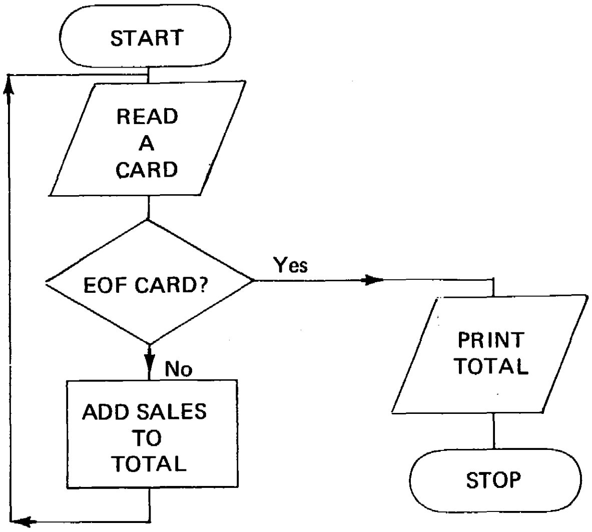
84. In the flowchart below, when the ADD SALES TO TOTAL operation is performed, the computer totals the contents of the two data areas and places the sum in TOTAL.

What will be the content of TOTAL at the instant the first card is read into the input area?

□ Zeros.

□ Data from a previous program.

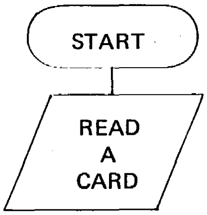
Data from a previous program



The flowchart □ will □ will not explicitly state the number of storage areas required.

will not

85. When data is read or moved into a location, the new data overlays (destroys) the previous contents. Assume that the first card contains the letters A through K. Show the contents of the input area after the first card is read.



Input area before READ

*	*	E	D	*	W	A	R	*	D	J

Input area after READ

A	B	C	D	E	F	G	H	I	J	K

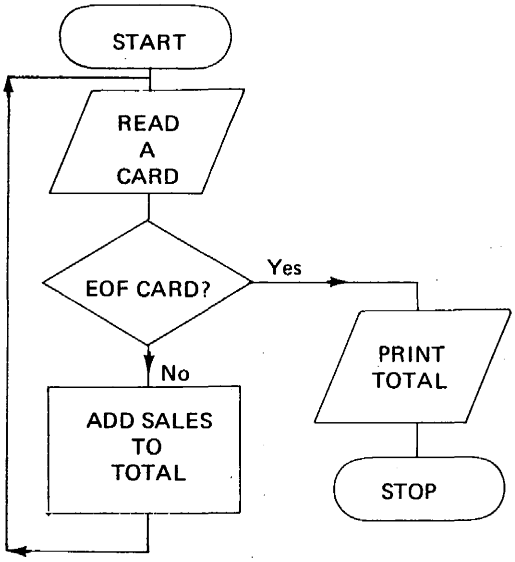
86. The flowchart below indicates that the ADD SALES TO TOTAL will add the contents of the field named SALES to the contents of the field named TOTAL.

Since we do not want to add the sales amount of the first card to the unknown value contained in TOTAL, the programmer must perform a housekeeping chore by filling TOTAL with zeros.

Is the housekeeping function illustrated in this flowchart?

□ No

□ Yes



We have seen that three functions that must be coded are not illustrated in the flowchart above. CHECK these three operations from the following list:

□ ADD SALES TO TOTAL.

□ Reserve memory space for the input.

□ Reserve memory space for TOTAL.

□ READ A CARD.

□ Zero the TOTAL field.

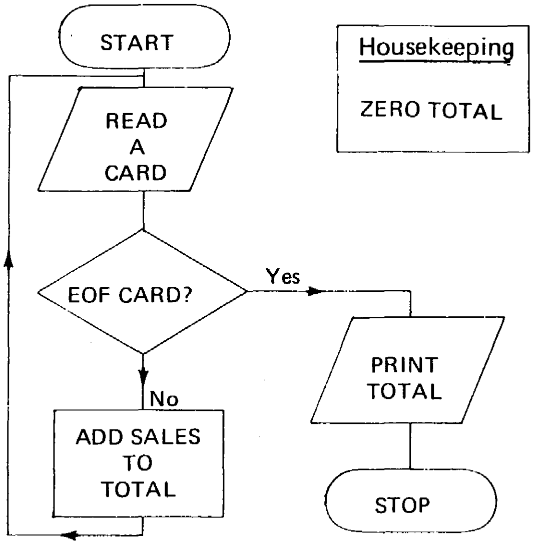
No

Reserve memory space for the input

Reserve memory space for TOTAL

Zero the TOTAL field

87. The allocation of storage areas is usually not illustrated in the flowchart. The input/output and work areas that are required are implied. However, housekeeping activities should always be explicitly indicated in the flowchart.



The housekeeping operation ZERO TOTAL should be included in the above flowchart. It will be performed only once. Therefore the housekeeping block should be inserted after the:

□ START block.

□ READ block.

□ ADD block.

START block

How many times will the above housekeeping function be performed?

□ Only once.

□ After each card.

Only once

88. A blank or space is not usually considered to be a numeric value. An area that is to be used in calculations should always be cleared with:

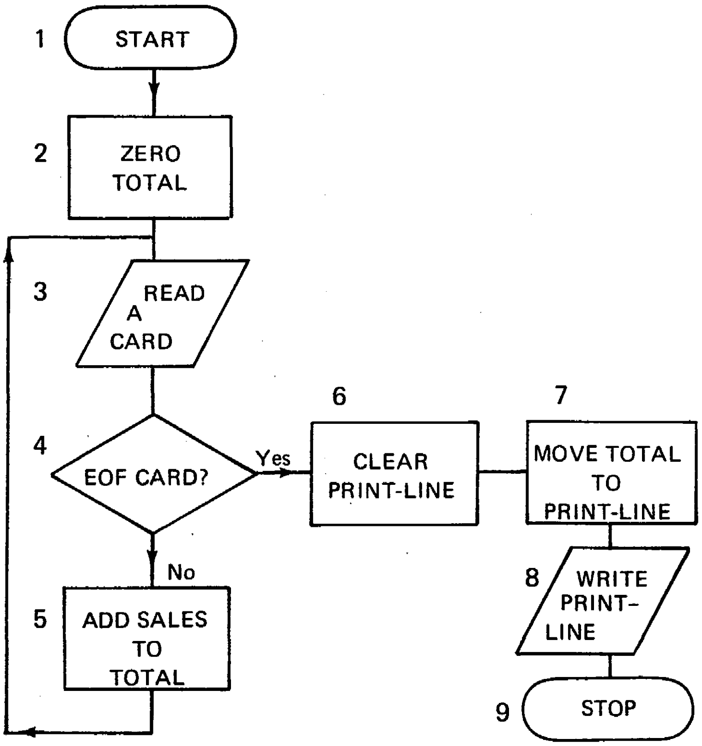
- □ an alphabetic character.
- □ zeros.
- □ blanks.

zeros

89. Housekeeping functions, such as clearing a print-line or setting a counter to zero, frequently appear as the first block in the flowchart. In some programs, housekeeping blocks appear at several key points.

In the following flowchart, which blocks indicate housekeeping operations? ________________

Blocks 2 and 6



90. The housekeeping functions that are performed only once are usually placed at:

□ the beginning of the program.

□ the end of the program.

The "CLEAR PRINT-LINE" operation in the previous flowchart is performed only once. Therefore, we can combine this block with the:

□ READ A CARD block.

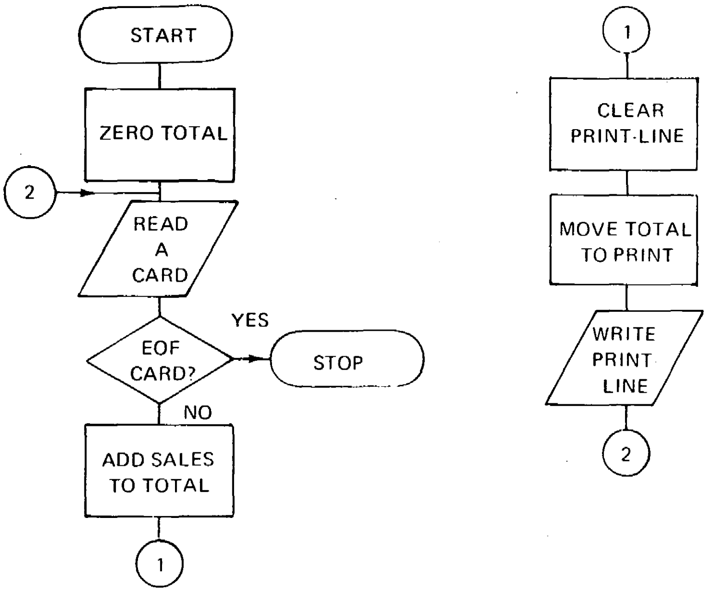
□ ZERO TOTAL block.

□ ADD SALES TO TOTAL block.

the beginning of the program

ZERO TOTAL block

91.



Some housekeeping operations must be repeated.

How many housekeeping operations are indicated in the above flowchart? ____________

The CLEAR PRINT operation is performed:

□ once for each card.

□ only once.

Could we combine the housekeeping operations into one block?

□ Yes

□ No

Two

once for each card

No

92. When a printed report is required as a result, the flowchart will usually include housekeeping operations that must be repeated.

TURN TO PANEL 2 ON PAGE 1-136.

PART 1 of PANEL 2 shows an area of memory that has been reserved by the programmer. This area will always be referred to as PRT-LINE. All data to be printed will be moved into this area before it is written out to the printer.

How many character positions of memory have been reserved for PRT-LINE? ____________________

132

PART 2 illustrates the headings that must be printed on the first page of the report.

How many times must PRT-LINE be filled with blank spaces before the headings are completely printed?

□ One

□ Two

□ Three

Three

93. Areas of memory that are reserved for use in calculations are always cleared by filling with zeros. What must the programmer use to clear the printer output area?

□ Zeros

□ Blanks

□ Alphabetics

□ Numerics

Blanks

94. The following process flowchart contains the programming steps required to read a file of punched cards and place the data on magnetic tape.

Assume that the cards are arranged in ascending sequence by an account number punched into the card. The program must check this sequence before the data is written to tape. If a card is out of sequence, it is to be omitted from the tape file. (Rejected cards will be included in a subsequent run.)

The program must compare the second account number to that of the first card and the third account number to that of the second card etc. It is necessary, therefore, to store the earlier number so that it may be compared to the later number. The name TEST has been assigned to this memory area. Write each step listed in the column below in the appropriate symbol of the following flowchart.

REJECT CARD

WRITE TO TAPE

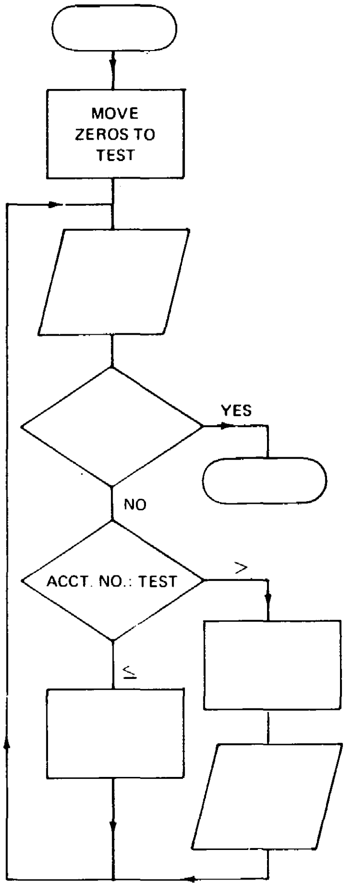
EOF CARD?

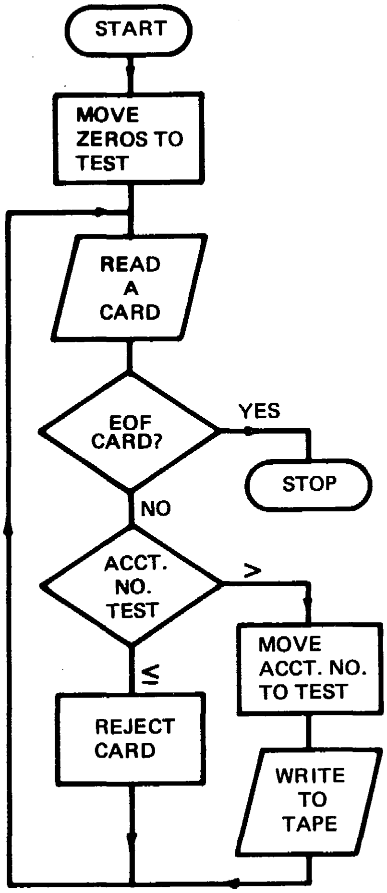
START

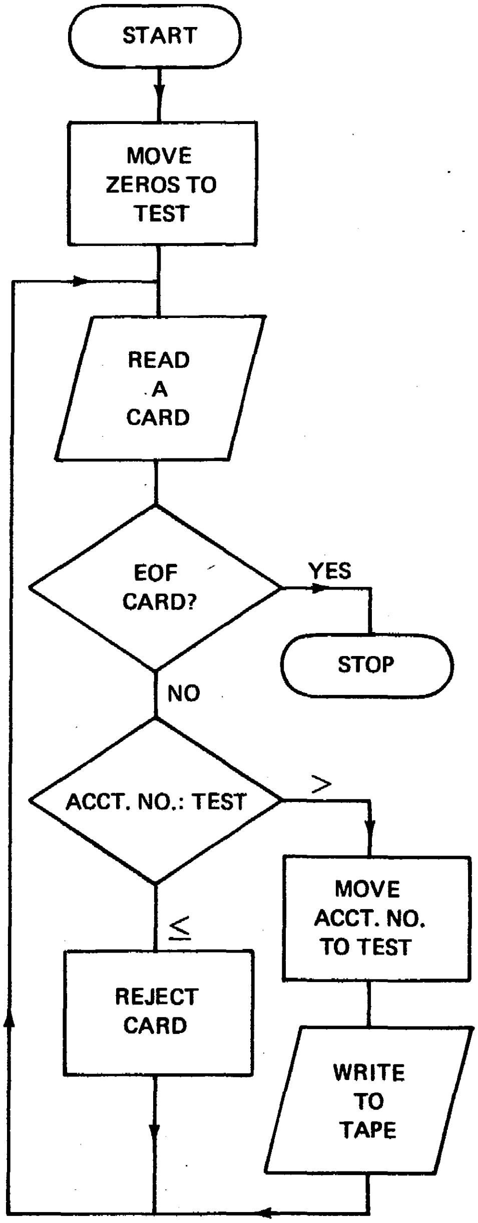
MOVE ACCT. NO. TO TEST

STOP (END OF JOB)

READ A CARD



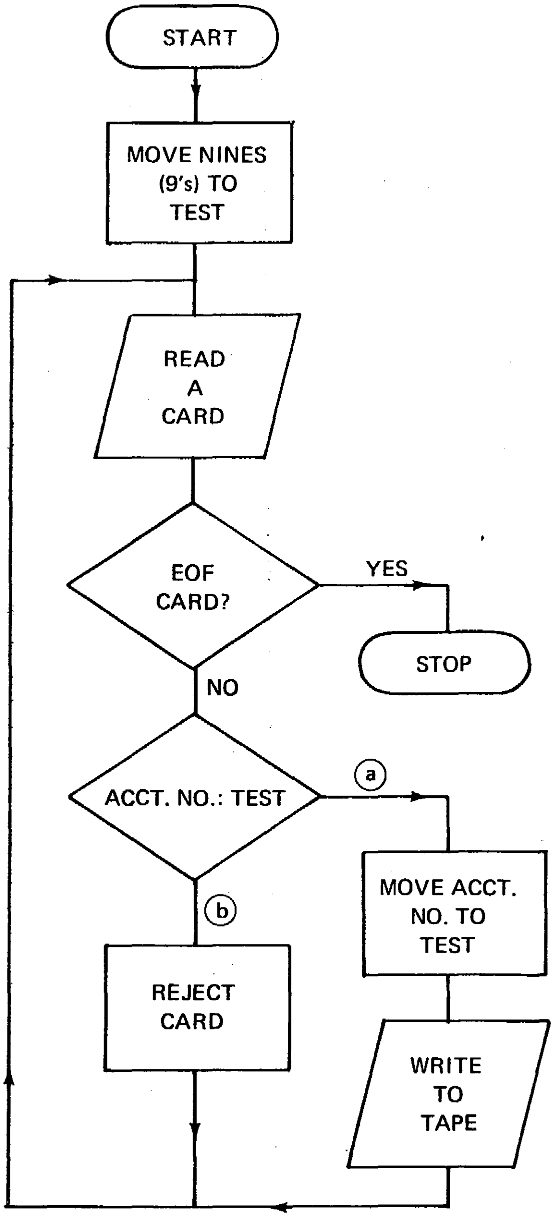



START
MOVE ZEROS TO TEST
READ A CARD
EOF CARD?
YES
STOP
NO
ACCT. NO.: TEST
>
MOVE ACCT. NO. TO TEST
≤
REJECT CARD
WRITE TO TAPE

95. As illustrated in the flowchart on the facing page, when the first card is read into memory, the account number punched into this card is compared with the contents of location TEST. What value will TEST contain at this time?	
□ The account number from a previous card.	
□ Zeros.	Zeros
□ Data left from a previous program.	
What value will TEST contain when the first card has been written to tape?	
□ Zeros.	
□ The account number of the first card.	The account number of the first card.
□ The account number of the next (second) card to be read.	
Will the second card be read before or after the first card is written to tape?	
□ Before	
□ After	After
How many times will the housekeeping activity of moving zeros to TEST be performed? ______________________	Once
What value will TEST contain after the 10th card has been written to tape?	
□ The account number of the 9th card.	
□ The account number of the 11th card.	
□ The account number of the 10th card.	The account number of the 10th card

96. The flowcharting technique illustrated in the previous frame is used frequently to test the sequence of input data.

The previous example assumed that the input data was arranged in ascending order; field TEST was thus loaded with zeros. In the example below, the input data is assumed to be arranged in descending numeric order; field TEST is thus filled with`nines (99999). This value is greater than any account number that will appear on a card.



In the above flowchart, indicate the correct symbol for:

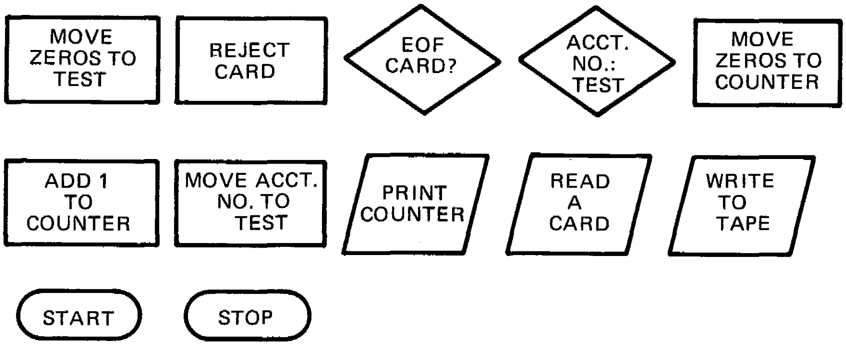
decision point ⓐ ____________

decision point ⓑ ____________

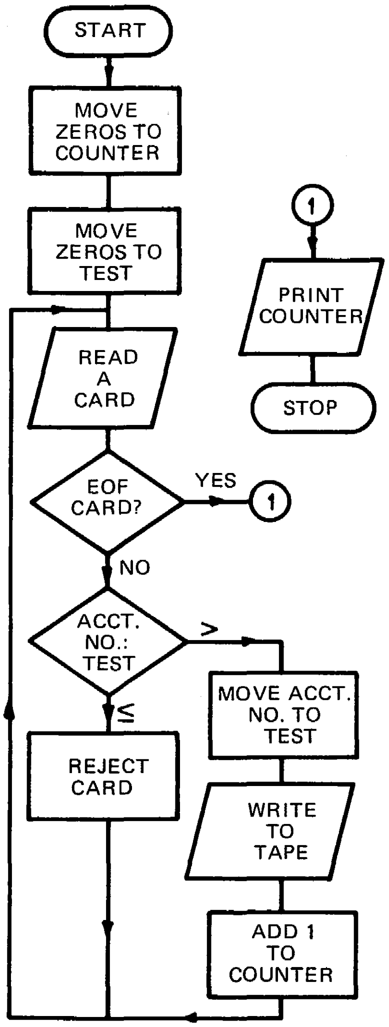
<

≥

97. Processing problems often reqüire the programmer to establish a field in memory that may be used as a counter. Printing the page number on a report or determining the number of records processed are examples of this.

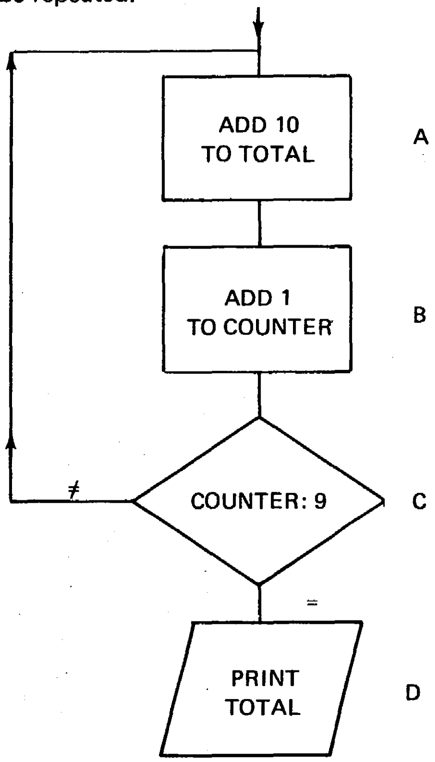


Using the above processing symbols, draw a flowchart in the space below to represent a program in which the number of records that have been successively written to tape are counted. This counter value should be printed before the program terminates.



98. A loop may be used to indicate that a sequence of instructions is to be repeated. Because the computer reads only one card at a time, a loop makes it possible to read subsequent cards in a file.

A counter is frequently used to control the number of times a loop will be repeated.



Which step(s) provide(s) loop control in the flowchart above?

□ Step A

□ Step B

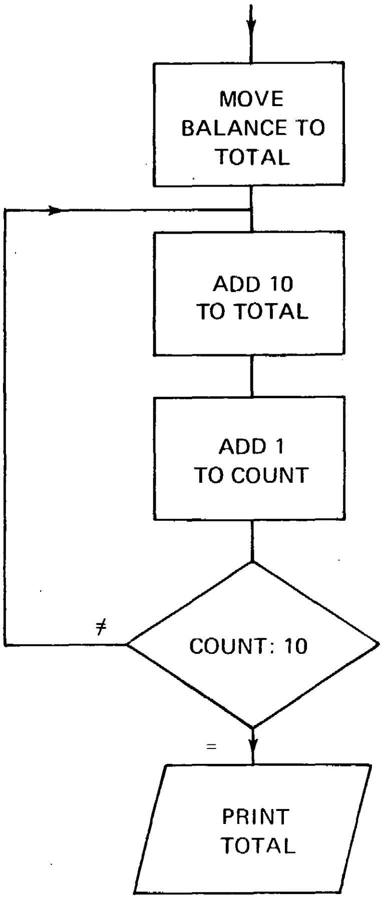
□ Step C

□ Step D

Step B

Step C

99.



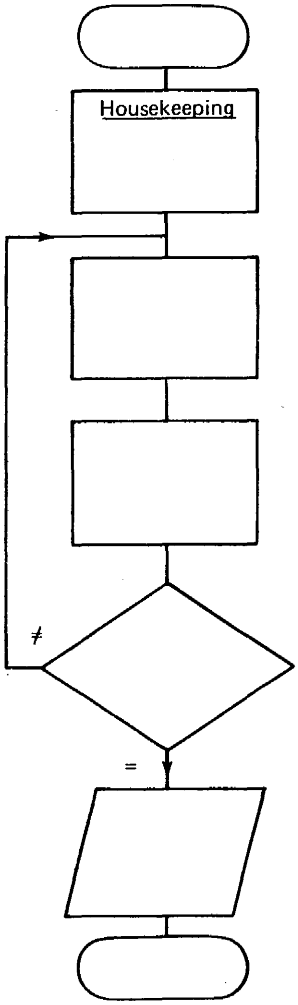
Assume that the COUNT field has been set to zero in the above flowchart. How many times will the ADD 10 TO TOTAL operation be performed? ______________

10 times

What value will be printed if COUNT is originally set to zero and BALANCE contains the value 240? ______________

340

100. Multiplication is simply a series of additions. We can easily construct a loop that will calculate 648 x 7 and print the result. Write each operational step listed in the column below in the appropriate symbol of the following flowchart:



STOP

ADD 648 TO TOTAL

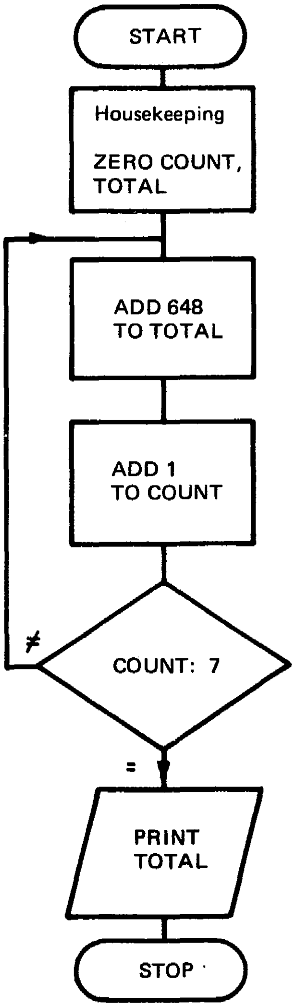
START

ZERO COUNT, TOTAL

COUNT:7

PRINT TOTAL

ADD 1 TO COUNT



101. In the program illustrated in the preceding frame the ADD 648 TO TOTAL operation will be performed ________________ (how many) times.

seven

What value must the COUNT field contain before the program will drop through and perform the print instruction? ________

7

The first time that ADD 648 TO TOTAL is performed, COUNT will contain:

□ 1

□ 0

□ 6

0

102. EXPRESS STOP

Match the following:

A. Source program	_______	Written by programmer.	A
B. Assembly program	_______	Supplied by manufacturer.	B
C. Object program	_______	Output of assembly run.	C
	_______	In machine language.	B,C
	_______	In symbolic language.	A
	_______	Directs assembly process.	B

An Assembly run will usually result in two outputs. Name them.

1. ________________	Object program
2. ________________	Program listing

Arrange the following data processing activities in the order in which they are performed:

Assembly run	1. ________________	Process flowchart
Process flowchart	2. ________________	Write source code
Test run	3. ________________	Assembly run
Debug assembly run	4. ________________	Debug Assembly run
Production run	5. ________________	Test run
Write source code	6. ________________	Production run

IF YOU ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 119 ON PAGE 1-91.

103. **PREVIEW**

The first steps in program preparation are frequently the responsibility of the systems analyst. If we assume that the analyst has supplied the programmer with a system flowchart and a definition of the problem in which all input and output formats are specified, we may then trace the programmer through the steps required to solve the problem when a symbolic language is used.

In the following frames you will learn to identify the steps required to produce a usable program.

104. Before the programmer starts to code the program, he must illustrate the logic of his program by drawing a:

□ system flowchart

□ process flowchart

process flowchart

105. TURN TO PANEL 3 (Page 1-137).

A programmer must be capable of performing all of the steps required to solve a data processing problem. Steps 1 and 2 of the flowchart in Panel 3 show activities that are usually performed by a systems analyst. The remaining steps are usually assigned to a programmer.

Notice that steps 5 and 7 are performed by the computer. What is the input to the Assembly run, step 5?

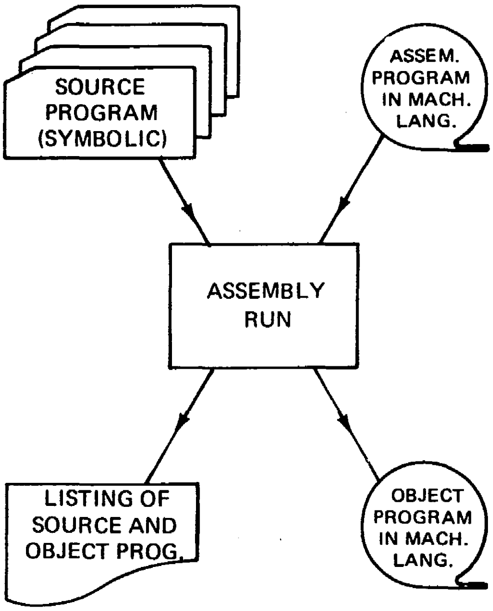
□ Cards punched from the programmer's coding form.

□ Test data.

Cards punched from the programmer's coding form

106. If the programmer makes a clerical error when coding the program, it will be detected during the Assembly run. What are the next two steps to be performed when an error is indicated? (Hint: look at Panel 3) 1. ____________________ 2. ____________________	Correct errors Assembly run
107. Is it always necessary to reassemble the program when an error is detected? □ Yes □ No	Yes
108. Suppose an error is detected during the test run, step 7, what are the next three steps to be performed? 1. ____________________ 2. ____________________ 3. ____________________	Correct errors Assembly run Any errors?
109. The computer manufacturer supplies an Assembly program that will translate the symbolic language Source program into a form that can be executed by the computer. The program that is produced by this Assembly run is called an Object program. To produce the Object program, two inputs are required. What are they called? 1. ____________________ 2. ____________________	Source program Assembly program

110.


SOURCE PROGRAM (SYMBOLIC)
ASSEM. PROGRAM IN MACH. LANG.
ASSEMBLY RUN
LISTING OF SOURCE AND OBJECT PROG.
OBJECT PROGRAM IN MACH. LANG.

The illustration above shows two inputs. One input is the deck of cards punched from the coding sheet (one card for each line of symbolic code). This input is considered to be data. The other input is supplied by ______________________.

Which input is considered to be data?

____

In which language, symbolic or machine, is each of the following?

Source program: ________________

Object program: ________________

Assembly program: ________________

the computer manufacturer

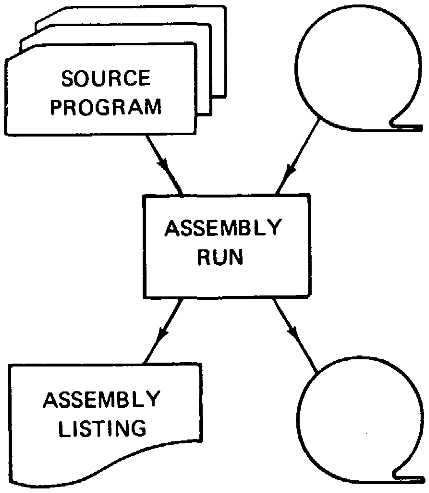
Source program

symbolic language

machine language

machine language

111. As you can see in the illustration below, the Assembly run has two inputs and two outputs. Name the two that are frequently resident on magnetic tape.



1. ____________

2. ____________

Which is the program written on the coding form by the programmer? ____________

If errors are detected during the Assembly run shown above, they will be printed on the ____________.

ASSEMBLY PROGRAM

OBJECT PROGRAM

Source program

Assembly Listing

112. It is your job as a programmer to locate any error and correct it. When all errors are corrected, the program must then be:

□ reassembled.

□ scrapped.

reassembled

113. The listing produced by the Assembly run will identify errors made by the programmer. If the listing indicates the program is error-free the programmer may:

□ stop; his task is complete.

□ prepare to take the next step.

prepare to take the next step

114. When your listing is error-free you are ready to test your program with data. The computer is then ready to read your object card deck (or tape) and follow the instructions you have written for it. What is this next computer run called?

□ Assembly run

□ Production run

□ Test run

Test Run

115. Data that the programmer makes up to test his program looks like real data but offers many advantages over the use of real data, for example:

- It is available when you need it.
- The programmer can make sure that all of the paths in the program are tested.
- The amount of data needed to test the program can be limited by the programmer.

Checking the test output is as important as writing the program. When a program has an error and does not work properly, it is said to have "bugs." The process of finding the error and correcting it is known as "debugging."

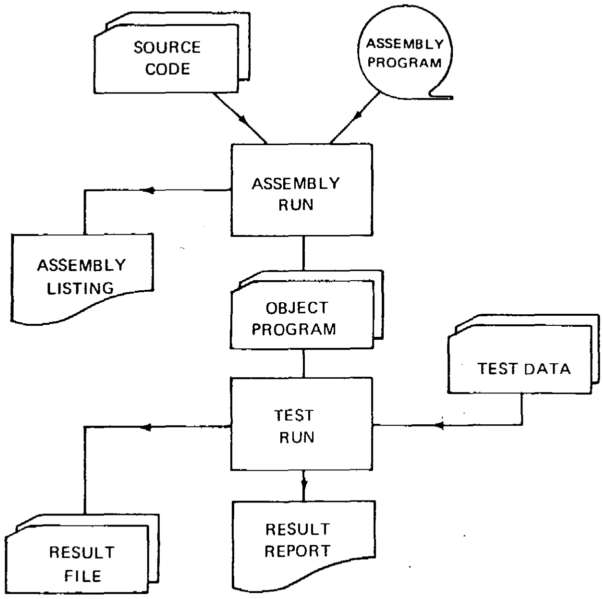
Is it necessary to repeat the Assembly run after debugging the program?

□ Yes

□ No

Yes

116.



The illustration above represents two kinds of computer runs. During the test run the computer is directed by the

____________________ .

Object program

117. Check the following statements as true (T) or false (F):

T F

□ □ The Assembly listing must be clear of errors before a test run is attempted.

□ □ The Test run is usually performed with real data.

□ □ The Object program is in symbolic language.

True

False

False

Match the following:

A. Assembly program

B. Source program

C. Object program

_____ The output of the translation (Assembly) run.

_____ Supplied by the manufacturer.

_____ Written by the programmer.

C

A

B

118. During a production run the computer executes the Object program in machine language to obtain the final result. Arrange the following activities in the order in which they are performed:

Assembly run 1. ____________________

Flowchart 2. ____________________

Test run 3. ____________________

Write source code 4. ____________________

Production run 5. ____________________

Real data is used during the:

- □ Test run.
- □ Assembly run.
- □ Production run.

What program is in control of the central processor unit during a production run? ____________________

Flowchart

Write source code

Assembly run

Test run

Production run

Production run

Object program

119. EXPRESS STOP

The smallest unit of memory storage that may be addressed in the 9200/9300 computer is called a __________ .

byte

Data fields that are alphabetic or alphanumeric will always be stored in:

□ packed format.

□ unpacked format.

unpacked format

Excluding parity, how many binary bits are required to store one alphabetic character? ________________

Eight

,Check the following statements as true (T) or false (F):

T	F		
□	□	Each magnetized plane of a plated wire can store one bit.	True
□	□	A decimal data field read in from a card character format enters the processor in unpacked format.	True
□	□	A packed field consists of two numerics in all but the most-significant byte of the field.	False
□	□	There are two formats for the storage of decimal data.	True
□	□	In unpacked format each byte contains two decimal digits.	False
□	□	Data is usually represented in Extended Binary-Coded Decimal Interchange Code.	True

IF YOU ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 142 ON PAGE 1-103.

120. PREVIEW

Data may concern inventory, accounts receivable, accounts payable, payroll, and the like. In many cases, processing involves not only performing calculations on some part of each record to arrive at balance, amount, or earnings, but also involves adding, changing, or deleting records as new transactions occur.

The data is made available to the memory of the system for processing by an input device. The information is sensed or read and is converted to a code used within the computer. The following group of frames will discuss the methods of representing data within the computer.

121. REVIEW

Computers function in binary states. This means that each magnetized plane of a plated bit wire can indicate only two conditions that we represent by the symbols 0 and 1. Patterns of these symbols are called binary code.

Information is represented in memory by:

□ punched holes.

□ printed characters.

□ magnetized plated bit wires.

magnetized plated bit wires

The system of symbolizing information with 1's and 0's is called ________________ code.

binary

122. The decimal numbering system uses 10 symbols to represent values. The place (positional) value of the digits signifies units, tens, hundreds, and so on.

Each position is ________(how many) times the value of the position to its right?

The binary number system uses only two symbols. What are they? ____________________

Binary symbols are commonly called ____________.

The positional value of bit symbols is based on the progression of powers of 2. The units position has a decimal value of 1; the next position, the value of 2; the next 4, and so on. A 1 bit represents the presence of a positional value as shown in the examples below:

BINARY POSITIONAL VALUE									
128	64	32	16	8	4	2	1		
0	0	0	0	0	0	0	0	=	0
0	0	0	0	0	0	0	1	=	1
0	0	0	1	0	0	0	0	=	16
0	0	0	1	0	1	0	0	=	20
0	0	1	0	0	0	1	1	=	35
0	1	0	0	0	0	0	0	=	64

Convert each of the following binary numbers to its equivalent decimal value:

0	0	0	1	0	1	1	1	=	____
1	0	0	0	1	0	1	0	=	____
1	1	1	1	1	1	1	1	=	____

10

0 and 1

bits

23

138

255

123. In the 9200/9300, each plated bit wire can store eight data bits.	
How many bits in a byte? ____________	Eight
A byte can store at least one __________ .	character
124. The 9200/9300 uses some numeric data in binary form by grouping 4 bytes (32 bits) into a unit called a word.	
How many bits in a byte? ______________	Eight
How many bytes in a word? ____________	Four
Since a word is four bytes long, how many planes of memory are required to store a word? ______________	32 planes
How many bits in a halfword? ____________________	16 bits
Is it necessary for the planes to be in consecutive memory positions?	
□ Yes	Yes
□ No	
How many consecutive bits of memory are required to store a doubleword? ____________________	64 bits
125. Match the following:	
A. Byte ______ Single magnetized plane.	B
B. Bit ______ Contains four bytes.	C
C. Word ______ Contains eight bits.	A
______ Zero (0) or one (1).	B
______ Contains 32 bits.	C
______ Storage of at least one character.	A

126. The Extended Binary Coded Decimal Interchange Code (EBCDIC) is the principal internal coding scheme used in the 9200/9300 System. Although each byte in EBCDIC is constructed of zeros and ones, the positional values of binary code are not applicable. Instead, unique bit patterns have been assigned to represent specific decimal, alphabetic, or special characters.

In EBCDIC, each decimal value from 0 through 9 is represented as a binary value in the rightmost four bit positions. The four leftmost bit positions are all ones as shown in the following examples:

Decimal	EBCDIC
1	11110001
2	11110010
3	11110011

What is the EBCDIC bit representation for each of the following decimal values?

4 ________

0 ________

8 ________

11110100

11110000

11111000

127. The four leftmost bits in the EBCDIC representation of a decimal value:

□ do not have a unique pattern.

□ have a unique pattern.

Are the four leftmost bits meaningful to the EBCDIC representation of decimal values?

□ Yes

□ No

do not have a unique pattern

No

128. The principal internal coding scheme used in the 9200/9300 System is:

□ Binary.

□ Octal.

□ EBCDIC.

The numeric, alphabetic, and special characters used to print checks or reports are called printer graphics. Turn to Panel 4 on page 1-138 and examine the characters in the column listed under PRINTER GRAPHICS.

EBCDIC

129. Assume the following characters are read in from punched cards. Show in EBCDIC code how they would appear in memory. (Refer to Panel 4.)

Character	EBCDIC Code	
* (asterisk)	______________	01011100
R	______________	11011001
I	______________	11001001
C	______________	11000011
E	______________	11000101
$ (dollar sign)	______________	01011011
4	______________	11110100
6	______________	11110110

130. Check the following statements as true (T) or false (F):

T F

□ □ At present, many EBCDIC patterns have no assigned printable/recognizable symbols in the 9200/9300 System.

True

□ □ The pure binary system can only be used to represent numeric data.

True

□ □ In the binary number system, the position of a bit determines the value of the bit.

True

□ □ The highest decimal value that can be represented in 4 bit positions is 15.

True

131. FOOTNOTE

With only a few exceptions, we will not be concerned with the binary numbering system. The discussion up to this point was intended primarily to acquaint you with what is involved in the internal coding of data and instructions. From this point, we will consider all data to be stored in EBCDIC.

132. When we read the value 367 into memory from a card, it will appear in memory as:

3	6	7
11110011	11110110	________

11110111

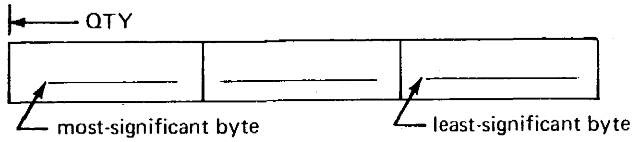
How many bytes of memory are required to store this value? ____________

3

When decimal data is represented as shown above, the format is called unpacked decimal. (Packed and unpacked format will be discussed later.)

133. Suppose we are writing a stock-control program that reads in punched cards containing the number of items added to or removed from stock. One field in this transaction card is called QTY (quantity).

We read a card with the value 829 in the QTY field. How would this appear in memory?



The four leftmost bits of each byte contain:

□ all zeros.

□ all ones.

□ a combination of ones and zeros.

When the four leftmost bits of the least significant byte are all ones, the field is considered to be positive. In the illustration above, the value 829 is:

□ positive (+).

□ negative (-).

The sign of the field is indicated by the four leftmost bits of the:

□ most-significant byte.

□ least-significant byte.

11111000	11110010	11111001

all ones

positive (+)

least-significant byte

134. Numeric data read in from punched cards will appear in memory in unpacked format. EBCDIC representation of decimal data is unpacked. This means that the leftmost four bits of the least-significant byte in the numeric field always indicate the ______________.

sign

The leftmost four bits of the other bytes in the field are called zone bits. The illustration below shows a three-byte unpacked decimal field.

Zone	Numeric	Zone	Numeric	Sign	Numeric

Using our positive value, 829, we could illustrate the data as:

Z	8	Z	2	+	9

The four leftmost bits of the bytes containing the digits 8 and 2 are called ______________ bits.

zone

The four leftmost bits of the byte containing the 9 indicate the ________________ of the field.

sign

Do these leftmost bits have a decimal value?

□ Yes

□ No

No

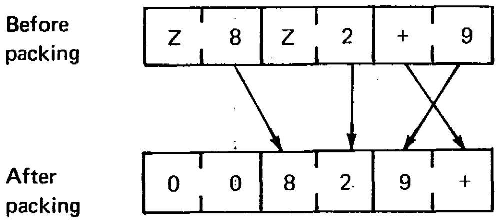
The bits that represent the decimal value in an unpacked byte are the:

□ zone bits.

□ numeric bits.

numeric bits

135. Since zone bits have no decimal value, we can replace them with numeric information by packing two decimal digits into an eight-bit byte. All numeric data used in calculations must be converted to packed format before arithmetic operations are performed. A single instruction can convert a field from unpacked to packed format. The instruction preserves the sign of the unpacked field in the packing operation by reversing the sign and numeric portion of the least-significant byte in the field.



After packing, the sign of the field is located in the:

□ rightmost position of the field.

□ leftmost position of the last byte.

rightmost position of the field.

An unpacked numeric field that is three bytes in length can be packed into ______________ bytes.

two

An unpacked numeric field that has a length of nine bytes can be packed into a field having a length of:

□ eight bytes

□ five bytes

□ four bytes

five bytes

136. Data read from cards enters the computer memory in:

□ packed format.

□ unpacked format.

unpacked format

How many bits are used to represent the sign of the field?

□ Eight

□ Four

□ One

Four

137. All decimal arithmetic performed in the 9200/9300 System requires the numeric data to be in:

□ packed format.

□ unpacked format.

The packed format provides a significant saving in storage because numeric data is stored in approximately half the area that would be required if the unpacked format were used.

packed format

138. Show how a five-column field in a record read from punched cards would appear in memory. Use Z for zone and use the positive value 40637.

Now show this value in packed format.

Z	4	Z	0	Z	6	Z	3	+	7

0	0	0	0	4	0	6	3	7	+

139. MATCH the following:

A. Packed format

B. Unpacked format

____ Data read from cards directly into memory. B

____ Used for arithmetic operations. A

____ Applicable to decimal data only. A

____ Sign represented in the rightmost four bits of field. A

____ Applicable to alphanumeric data. B

140.

Z	N	Z	N	Z	N	S	N

The above illustration symbolizes data in:

□ packed format.

□ unpacked format.

The symbol Z represents the ______________ portion of the byte.

The symbol N represents the ______________ portion of the byte.

The symbol S represents the ______________ portion of the byte.

To store the above data in memory in packed format requires:

□ three bytes

□ four bytes

□ five bytes

unpacked format

zone

numeric

sign

three bytes

141. Packed data must be converted to the unpacked format before it can be punched into cards or printed as output. Again, a single instruction can perform this operation. Usually two storage areas are involved. The data stored in the smaller (packed) area is unpacked into a larger storage area.

Unpack the contents of the field named AREA 1 into AREA 2 below:

AREA 1

N	N	N	N	N	S

AREA 2

Pack the contents of HOT into COLD below:

HOT

Z	4	Z	5	Z	7	+	6

COLD

Z	N	Z	N	Z	N	Z	N	S	N

0	0	0	4	5	7	6	+

142. EXPRESS STOP	
CHECK the following statements as true (T) or false (F):	
T F	
□ □ All input/output devices are controlled by instructions.	True
□ □ Instructions are stored in random sequence.	False
□ □ An instruction can be executed only once because it is destroyed.	False
□ □ Instructions are usually executed in the order in which they are stored.	True
□ □ Data is read into memory before the program is entered.	False
□ □ Instructions can be executed over and over again.	True
□ □ All computer operations are directed by instructions.	True
In what form are instructions stored during the production run?	
□ Machine language	Machine language
□ Symbolic language	
Input data is punched one record per card. What will enter the computer when the first read instruction is executed?	
□ The entire file	
□ Data from one card	Data from one card
Which part of an instruction indicates the action to be taken by the computer?	
□ Operation part	Operation part
□ Operand part	

142. EXPRESS STOP (Continued)

Computer systems that use variable-length instructions provide more space for the storage of data.

□ True

□ False

True

IF YOU ANSWERED ALL QUESTIONS CORRECTLY, SKIP TO FRAME 158 ON PAGE 1-113.

143. PREVIEW

The programmer writes a series of instructions that specify the operations to be performed by the computer. If data is involved, the instructions must direct the computer to the data. If a device is to be controlled, the instruction must specify the device and the operation to be performed.

In the following frames you will learn more about the information contained in instructions and how they are stored and executed.

144. Where are instructions stored during execution of the program?

□ In memory

□ In punched cards

□ On magnetic tape

In memory

After the program is written and assembled, the entire program is usually read into memory before any data is brought in.

In what form are instructions stored during the production run?

□ Symbolic language

□ Machine language

Machine language

145. In general, no particular storage areas are reserved for instructions. They are usually grouped together and placed in ascending sequential locations in the order in which they are entered.

If a program is read into memory and stored in locations 1000 through 2000, the first instruction to be executed will be the one that starts at location ____________.

If the first instruction is six bytes long, the next instruction will start at location:

□ 1006

□ 1007

Are special areas of memory reserved for exclusive use by instructions?

□ Yes

□ No

1000

1006

No

146. The space required in memory to store the many instructions in a program is an important factor in the efficiency of a computer system. If the space required for each instruction is large, less space is left in memory for data.

Some computer systems use fixed-length instructions while others, like 9200/9300, use variable-length instructions.

If the computer design called for each instruction to occupy the same fixed number of memory locations, this fixed-length would have to equal the:

□ longest instruction in the set.

□ shortest instruction in the set.

Which system will provide more space for the storage of data?

□ Fixed-length instructions

□ Variable-length instructions

The number of character locations (bytes) used to store a single instruction in the 9200/9300 is two, four, or six, depending on the type of instruction.

The maximum length of a 9200/9300 instruction is:

□ two bytes.

□ one word.

□ six bytes.

□ four bytes.

The minimum length for 9200/9300 instructions is ____________ bytes.

longest instruction in the set

Variable-length instructions

six bytes

two

147. An instruction usually consists of at least two parts. The operation part designates read, write, add, subtract, move, and so on. The operand part designates the address of the information or device that is needed for the specified operation.

The type of information contained in each part of an instruction is described in the examples below:

OPERATION	OPERAND
Add	quantity stored in memory location 3000 to quantity stored in location 3500.
Read	one record into locations 5000 through 5079.
Branch	to instruction in location 4800.
Move	10 bytes starting at location 1200 to location 1600.

Which part of the instruction indicates the action to be taken?

□ Operation part

□ Operand part

What is the length of the smallest instruction in the 9200/9300 System?

□ Two bytes

□ Four bytes

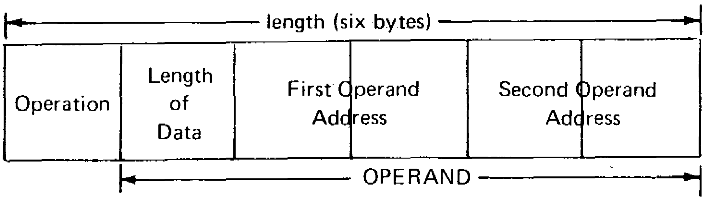
□ One word

□ Six bytes

Operation part

Two bytes

148.



The above format is a typical six-byte 9200/9300 instruction as it appears in memory.

How many bytes of memory are required:

to store the address of the first operand? ______________

Two bytes

to store the address of the second operand? ______________

Two bytes

Which byte of the instruction:

determines the kind of operation to be performed?

____

The first (leftmost) byte

determines how much data will be processed?

____

The second byte

What is the total length of the operand? ______________

Five bytes

149. The symbolic code used by the programmer is translated during the Assembly run. During the production run, the control unit will recognize one machine code pattern of 0 and 1 bits to mean ADD and another pattern will indicate MOVE.

What part of the instruction provides this information?

____

Operation part

150. The programmer uses a mnemonic code to represent the operation he wishes the computer to perform. When "A" is the first letter in the mnemonic code of an instruction, it is interpreted as Add. The letter "B" is interpreted as Branch. What will be the interpretation of the letter "C"?

- □ Subtract
- □ Compare
- □ Move

Compare

The mnemonic code is used in the:

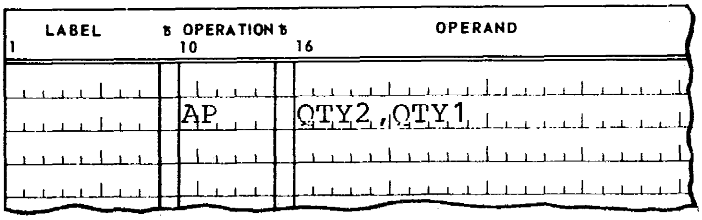
- □ operand part of the instruction.
- □ operation part of the instruction.

operation part of the instruction

151. MATCH the following mnemonic operation codes to the operation they perform:

AP	______ Move character	MVC
SP	______ Edit	ED
MVC	______ Add Packed Decimal	AP
MVI	______ Divide Packed Decimal	DP
CLC	______ Branch on Condition	BC
BAL	______ Subtract Packed Decimal	SP
ZAP	______ Compare Packed Decimal	CP
ED	______ Branch and Load	BAL
BC	______ Move Immediate	MVI
DP	______ Compare Logical	CLC
CP	______ Zero and Add Packed Decimal	ZAP

152.



LABEL 1	ƀ OPERATION ƀ 10	OPERAND 16
	AP	QTY2,QTY1

This is a typical Add Packed Decimal instruction used in programming the 9200/9300. It tells the computer to add the packed decimal data stored in the location designated by the symbol QTY1 to the packed decimal data stored in the location designated by QTY2. (The result is stored in the QTY2 location.)

What information is contained in this instruction? (Check the correct statement(s).)

- □ The number of times the instruction will be executed.
- □ The location of the data to be operated upon.
- □ The operation to be performed.
- □ The format of the data.

What does the operand part of the instruction specify?

____

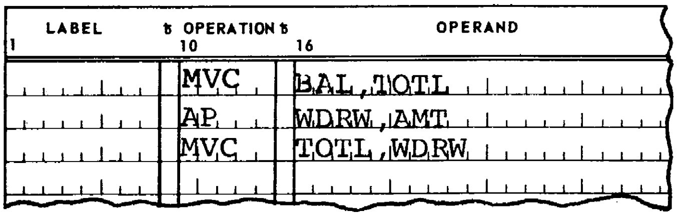
The location of the data to be operated upon.

The operation to be performed.

The format of the data.

The addresses of the data to be operated on.

153. The use of symbols to represent the individual data items relieves the programmer of the task of keeping track of the actual memory locations of the data. As illustrated in the instructions below, the programmer may move data from one location to another without becoming concerned about the address of the memory location he is moving to. The computer relates the symbol to the address and maintains all the records it will need.



LABEL 1	OPERATION 10	OPERAND 16
	MVC	BAL,TOTL
	AP	WDRW,AMT
	MVC	TOTL,WDRW

In the first instruction shown above, the data at location TOTL is moved to location ____________.

Copy the operation code and operands of the last instruction.

OPERATION CODE: __________

OPERAND: ________________

BAL

MVC

TOTL,WDRW

154. Suppose two fields are to be added. FLD1 is three bytes long and FLD2 is two bytes long. Both fields are in packed decimal format.

Of the two instructions shown below, which one contains all of the information required?

□ AP FLD1, FLD2

□ AP 3,2

The operand part of this instruction must contain such information as the memory locations of the data to be added. How is this information included in the instruction?

□ Symbols represent the addresses.

□ By including the actual memory addresses.

AP FLD1, FLD2

Symbols represent the addresses.

155. In addition to instructions that specify internal processing operations, the program usually includes instructions that bring (read) in data. They indicate which input device to bring it in from, and where to store it.

Such read instructions should be brought into memory:

□ after data is brought in.

□ before data is brought in.

before data is brought in

Will all the data of a file be brought into memory at one time?

□ Yes

□ No

No

156. In most data processing applications, the computer operates on the data fields of one record at a time.

If the input data is punched one record per card, what will enter the computer when the first READ instruction is encountered?

□ The entire file.

□ Data from one card.

Data from one card

How much information is usually in memory at one time?

□ Part of the program and all of the data.

□ The entire program and some data.

□ Part of the data and part of the program.

The entire program and some data

157. REVIEW

Check the following statements as true (T) or false (F):

T	F		
□	□	Instructions can be executed over and over again.	True
□	□	A special area of memory is reserved for use by instructions.	False
□	□	Instructions are stored in the same sequence as they are written.	True
□	□	Data is brought into memory before the instructions.	False
□	□	The maximum length of a 9200/9300 Assembly language instruction is six bytes.	True
□	□	An instruction can be executed only once.	False
□	□	Instructions are normally executed in consecutive order.	True
□	□	Instructions contain the symbolic addresses of data in the operand.	True
□	□	The card reader is controlled by instructions.	True
□	□	An instruction contains information that indicates the number of times it will be executed.	False
□	□	The entire program is usually resident in memory before data is brought in.	True

158. EXPRESS STOP

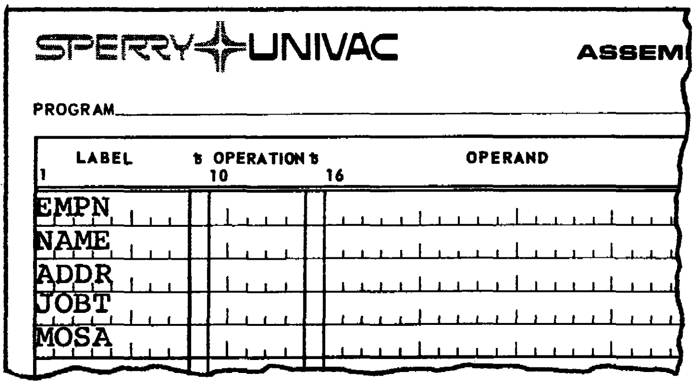
Check the following statements as true (T) or false (F):

T	F		
□	□	The Assembly system converts symbolic language to machine code.	True
□	□	The START and END statements generate machine code in the object program.	False
□	□	Data storage areas are defined by Assembler-directing instructions.	True
□	□	The Object program contains the machine code representation of Assembler-directing instructions.	False
□	□	Data is edited for printing by an Assembler-directing instruction.	False
□	□	The Assembly system is controlled by Assembler-directing instructions.	True
□	□	Data storage areas are reserved by the START and END statements.	False
□	□	Data is moved from one area to another by an Assembler-directing instruction.	False
□	□	The START and END statements perform no function during a production run.	True
□	□	The Assembler assigns a storage address to each symbol written in the label field of the coding form.	True
□	□	The START instruction will indicate when data is to be brought into memory.	False
□	□	Define Storage (DS) instructions are used by the Assembly program.	True

158. EXPRESS STOP (Continued)

In the coding form below, complete the Define Storage (DS) statements to allocate memory for a file in which each record has the following format:

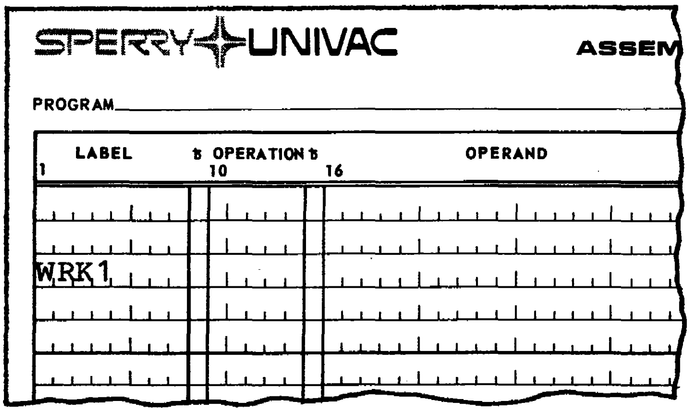
Employee Number	5 characters
Name	20 characters
Address	20 characters
Job Title	30 characters
Monthly Salary	5 characters


SPERRY UNIVAC ASSEM

PROGRAM

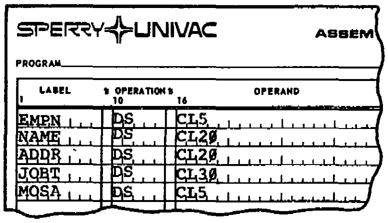
LABEL 1	OPERATION 10	OPERAND 16
EMPN		
NAME		
ADDR		
JOBT		
MOSA		

Assume that the tag WRK1 in the coding form below represents a work area. Complete the Define Storage statement to allocate working storage for each record in the above file.


SPERRY UNIVAC ASSEM

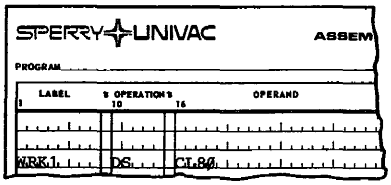
PROGRAM

LABEL 1	OPERATION 10	OPERAND 16
WRK1		


SPERRY UNIVAC ASSEM

PROGRAM

LABEL 1	OPERATION 10	OPERAND 16
EMPN	DS	CL5
NAME	DS	CL2Ø
ADDR	DS	CL2Ø
JOBT	DS	CL3Ø
MOSA	DS	CL5


SPERRY UNIVAC ASSEM

PROGRAM

LABEL 1	OPERATION 10	OPERAND 16
WRK1	DS	CL8Ø

IF YOU ANSWERED ALL QUESTIONS CORRECTLY SKIP TO FRAME 171 ON PAGE 1-125.

159. **PREVIEW**

The programmer must be familiar with the various functional categories of instructions before he can efficiently use reference manuals to select instructions for certain jobs.

In the next group of frames we will examine the processing operations performed by a few of the instructions frequently found under the categories of:

- Assembler-directing instructions
- Logical instructions
- Branching instructions

160. Assembler-directing instructions provide auxiliary functions that assist the programmer to control the assignment of storage addresses, to define data and storage fields, and to control the Assembly System itself. These functions are performed during the Assembly run and, with few exceptions, do not result in the generation of any machine language code. The START and END statements are typical examples.

Does the object program contain a machine-code representation of all the instructions written by the programmer?

□ Yes

□ No

Check the functions below that are performed by Assembler-directing instructions:

□ Define data storage areas.

□ Move data from one memory location to another.

□ Edit data for printing.

□ Control the assignment of storage addresses.

□ Control the Assembly system.

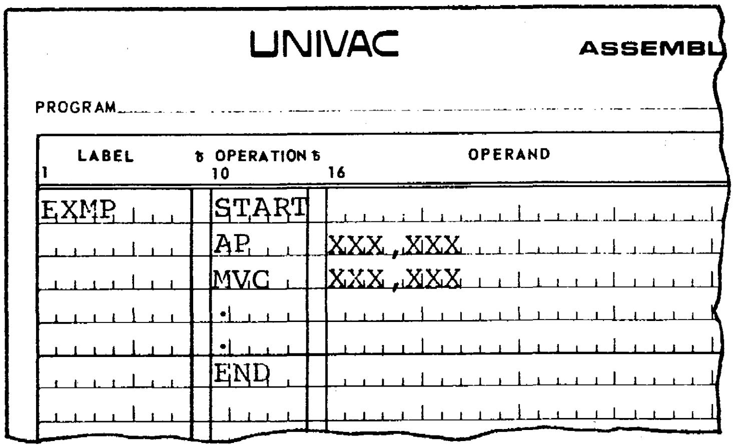
No

Define data storage areas

Control the assignment of storage addresses

Control the Assembly system

161. The START and END statements are examples of Assembler-directing instructions. They define the beginning and end of the program, and are normally the first and last statements written in the program.



UNIVAC ASSEMBL

PROGRAM

LABEL	OPERATION	OPERAND
EXMP	START	
	AP	XXX,XXX
	MVC	XXX,XXX
	•	
	•	
	END	

The START and END statements are necessary because they:

□ reserve data storage areas.

□ inform the Assembler of the beginning and end of the program being assembled.

□ improve documentation.

Will the START and END statements appear in the Object program?

□ Yes

□ No

inform the Assembler of the beginning and end of the program being assembled

No

162. The Assembler system converts the symbolic language statements into machine code, determines memory requirements for data and instructions, performs error checking, and has other responsibilities during several examinations (passes) of the source statements. Several passes are generally required to produce an Object program.

When the computer is examining one source statement after another in order to produce an Object program, it knows when the last statement is encountered because this is a function of the:

□ START statement

□ END statement

What function will the START and END statements perform in the Object program during a production run?

□ They indicate when data is to be brought in or written out.

□ They indicate the amount of memory required to store the data.

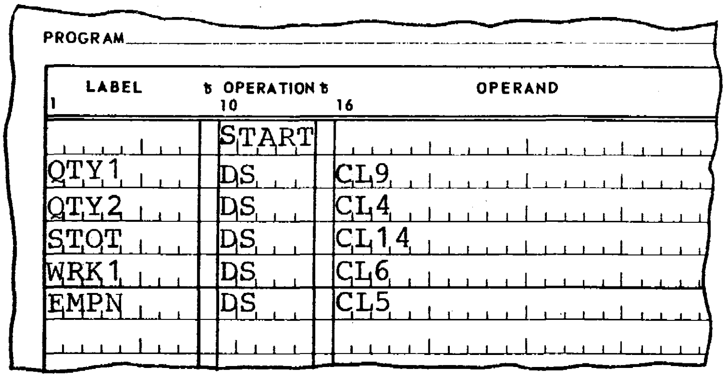
□ They perform no function and are therefore not present in the Object program.

END Statement

They perform no function and are therefore not present in the Object program

163. Following the START statement, the next logical step in coding a program would be to designate the memory area for record processing. An Assembler-controlling code is provided for this purpose. The 9200/9300 uses a Define Storage code identified by the letters DS in the Operation field of the coding sheet. A digit in the Operand field following CL (Character Length) specifies the number of bytes to be allocated.

If we are allocating memory to a data item, we will most likely want to reference this item during our program. The programmer can assign a tag to the item by writing a symbolic name in the Name field of the coding form as shown in the example below:



```
PROGRAM
 LABEL    ␢ OPERATION ␢        OPERAND
1           10         16
            START
QTY1        DS         CL9
QTY2        DS         CL4
STOT        DS         CL14
WRK1        DS         CL6
EMPN        DS         CL5
```

How many bytes of memory are ıeserved for:

EMPN ________________ 5 bytes

STOT ________________ 14 bytes

QTY2 ________________ 4 bytes

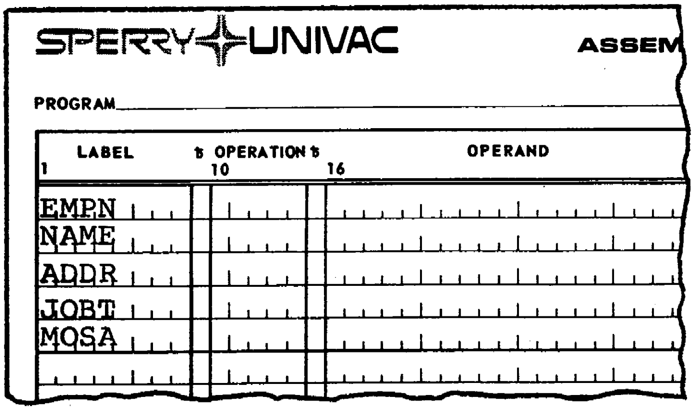
What Assembler-directing mnemonic tells the system to reserve the required bytes? ____________ DS

The symbolic names listed above are called ____________ . tags

164. Write the necessary Define Storage (DS) statements to direct the Assembler to allocate memory for a file in which each record has the following format.

Employee Number	5 characters
Name	20 characters
Address	20 characters
Job Title	30 characters
Monthly Salary	5 characters

We have used a symbolic tag that will describe the data to be stored in the field. Each tag must be unique and must not exceed four characters.



SPERRY UNIVAC ASSEM

PROGRAM

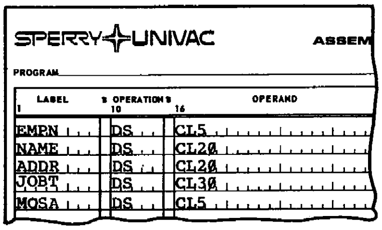
LABEL 1	OPERATION 10	OPERAND 16
EMPN		
NAME		
ADDR		
JOBT		
MOSA		

The Assembler will assign a memory address to each tag in the Label field. What Advantages are gained by using tags instead of actual memory addresses?

□ A name can be chosen that will describe the data to be stored in the field.

□ Words are more easily recalled than numbers.

□ The internal language of the computer is alphabetic.



SPERRY UNIVAC ASSEM

PROGRAM

LABEL 1	OPERATION 10	OPERAND 16
EMPN	DS	CL5
NAME	DS	CL2Ø
ADDR	DS	CL2Ø
JOBT	DS	CL3Ø
MOSA	DS	CL5

A name can be chosen that will describe the data to be stored in the field.

Words are more easily recalled than numbers.

165. In the previous frame you allocated memory for the record of a single employee. What were the total number of bytes required? ______________

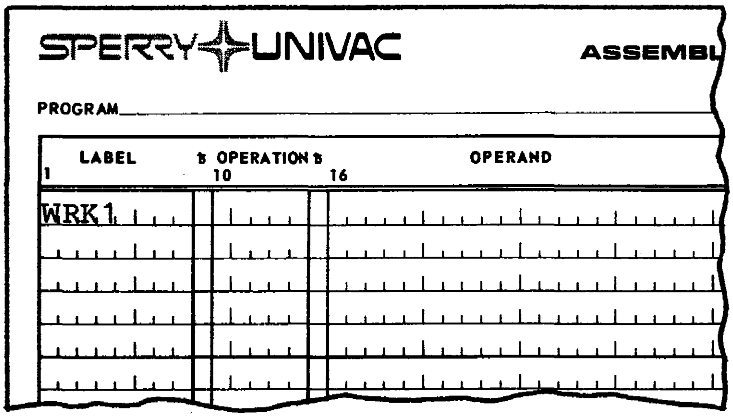
Would it be necessary to write the DS statements for each employee in the file if there were 3000 employees?

□ Yes

□ No

The logic of the program will usually cause the data from a previously processed record to be overlayed by the data from a subsequent record since the same area is used for each record.

Suppose that the logic of the program specifies that after each record is read in from a card it shall be moved to a work area (WRK1) before processing is performed. Write the Define Storage statement to allocate memory for this WRK1 field.



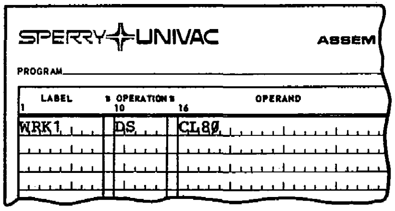
SPERRY UNIVAC ASSEMBL

PROGRAM

LABEL 1	OPERATION 10	OPERAND 16
WRK1		

80

No

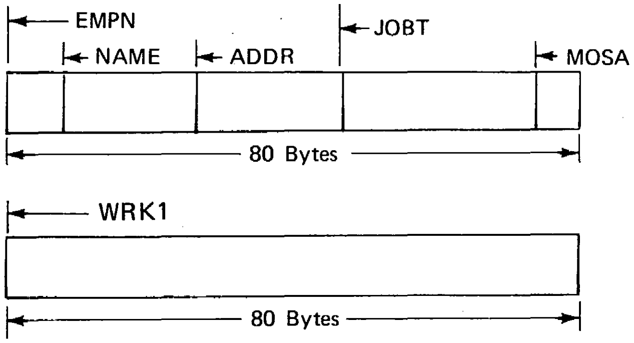


SPERRY UNIVAC ASSEM

PROGRAM

LABEL 1	OPERATION 10	OPERAND 16
WRK1	DS	CL8Ø

166. The two memory areas that were reserved by your DS statements are illustrated below.



The memory areas are identified by their tags. Each tag references the address of the most significant byte of the individual field.

How many individual fields are there in the input area?

□ One

□ Five

□ Six

Each of the fields may be individually addressed by name. The following instruction will move the contents of location EMPN into WRK1

```
MVC WRK1, EMPN
```

What will happen if this instruction is followed by:

```
MVC WRK1, NAME
MVC WRK1, ADDR
MVC WRK1, JOBT
MVC WRK1, MOSA
```

□ The total record will be correctly positioned in the WRK1 field.

□ The first part of the WRK1 field will be overlayed when each subsequent instruction is executed.

Five

The first part of the WRK1 field will be overlayed when each subsequent instruction is executed.

167. We could, of course, define WRK1 as a five-byte field and write four more DS statements directly after it Perhaps they could be named WRK2, WRK3, and so on. This will create five individually addressable fields into which the five items of data could be moved.

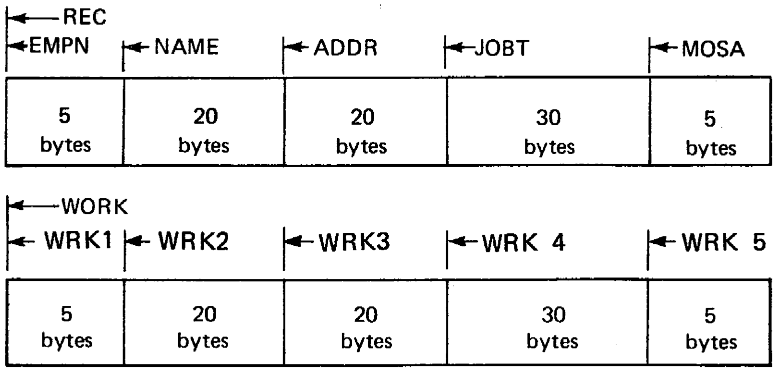
How many Move Character instructions would be required to relocate one employee record? ________

Five

168. Assuming that we are to process thousands of employee records, we would be wasting a considerable amount of time executing five instructions to move the data from an input to a work area. The Assembler-controlling code, ORG, is provided to eliminate this problem. The ORG statement permits us to address the entire unit as a record. Our DS statements can thus be written as follows:

```
EMPN      DS        CL5
NAME      DS        CL20
ADDR      DS        CL20
JOBT      DS        CL30
MOSA      DS        CL5
          ORG       EMPN
REC       DS        CL80
WRK1      DS        CL5
WRK2      DS        CL20
WRK3      DS        CL20
WRK4      DS        CL30
WRK5      DS        CL5
          ORG       WRK1
WORK      DS        CL80
```

Now the programmer can address the entire record or any subsection that has been assigned a name. The illustration below shows the names assigned to the memory locations.



How many bytes of data will be moved when the following instructions are executed?

MVC WRK1, EMPN ____________ 5

MVC WORK, REC ____________ 80

169. FOOTNOTE

The programmer will often use instructions that will guide the Assembly program. He may, for example, wish to have the program assembled in some specific area of memory, or he may wish to enter certain constants as part of the program. A large number of such tasks may be left to the Assembly program.

Assembler-directing instructions are written on the same coding sheets as the symbolic instructions.

NO RESPONSE REQUIRED

170. REVIEW

This frame will give you a chance to review what you have learned about instructions that direct the Assembly program. Check the following statements as true (T) or false (F):

T	F		
□	□	The START and END statements are used by the Assembly program.	True
□	□	The START and END statements will generate machine code in the Object program.	False
□	□	Data storage areas are defined by Assembler-directing instructions.	True
□	□	A symbolic name (tag) can be assigned to a data storage area.	True
□	□	Define Storage (DS) instructions are used by the Assembly program.	True
□	□	The START instruction will indicate when data is to be brought into memory.	False
□	□	The Assembler will assign a memory address to each tag written in the label field of the coding form.	True

NOTES:

171. EXPRESS STOP	
Which of the following are Assembler-directing instructions?	
□ CLC	
□ DS	DS
□ START	START
□ BC	
□ ED	
Match the following:	
1. START ____ Tests the condition code indicator for a specific condition.	2
2. BC ____ Compares the binary value of two fields.	3
3. CLC ____ Marks the beginning of a program.	1
4. DS ____ Allocates memory.	4
When editing data, it is often necessary to suppress zeros. This means that:	
□ all zeros are suppressed.	
□ leading zeros are suppressed.	leading zeros are suppressed
Which kind of data is edited?	
□ Alphanumeric	
□ Numeric	Numeric
□ Alphabetic	

171. **EXPRESS STOP (Continued)**

If the programmer wishes to edit a data field by inserting the $ and decimal point symbols, these symbols must be included in the applicable position of the:

□ data.

□ instruction.

□ mask.

mask

What two functions are performed by the Compare Logical (CLC) instruction?

□ Performs a binary compare of two fields and sets an internal indicator.

□ Sets an indicator and algebraically compares two fields of data.

□ Compares two fields and performs a branch.

Performs a binary compare of two fields and sets an internal indicator

172. **PREVIEW**

Most computer systems contain a group of instructions called logical instructions. Although it is beyond the scope of this chapter to teach the details of logical instructions, we will discuss them briefly.

The logical instruction set of the 9200/9300 Assembly language includes data movement, comparing, editing, bit testing, translating, and shifting instructions. We will discuss several applications of a few of these instructions in the next group of frames. We will also introduce several instructions that are included in the 9200/9300 branching instruction set.

173. An instruction that prepares data for printer output by inserting symbols or eliminating leading zeros is called:

□ an edit instruction.

□ a translate instruction.

□ an addition instruction.

an edit instruction

174. Editing of data is an important function of any program that generates a printed output. A payroll program that produces checks would include editing instructions.

Assume that you are programming such a problem and that you have determined that no check will exceed $999.99. After performing the necessary calculations, the program should print a check for each employee. Using the sample data shown below, write the edited value, including the decimal point and the dollar sign as it should appear on the check for each of the three employees.

Note:

The decimal point is not stored in memory. The symbol ∧ is used to represent the assumed decimal point position of the data in memory.

Data in Memory	Check Amount	
0 7 | 3 ∧ 4 | 3 +	Employee 1 ________	$73.43
7 6 | 7 ∧ 1 | 4 +	Employee 2 ________	$767.14
0 0 | 7 ∧ 0 | 1 +	Employee 3 ________	$7.01

What characters were inserted by the editing operation?
____

Dollar sign ($) and decimal point(.)

What characters were erased (suppressed) by editing?
____

Leading zeros

Is a blank space permitted between the symbol $ and the number 7? ________________

No

175. The edit (ED) instruction is used to prepare numeric data for printing. In the 9200/9300 System this data must be in packed format. The programmer simply designs a mask that is a character by character representation of the data as it is to appear after printing. When the ED instruction is executed, any comma, asterisk, decimal point, or other symbol that has been included in the mask is inserted in its proper position.

To be edited, data must be:

□ alphabetic.

□ numeric.

□ alphanumeric.

The programmer must represent each character position of the data and insert the edit symbols in the relative position in the ________________.

Will it be necessary for the programmer to design and insert a mask for each data item he wishes to edit?

□ Yes

□ No

numeric

mask

Yes

176. Check each of the following statements as true (T) or false (F). An edit instruction:

T	F		
□	□	is used to change instructions.	False
□	□	is considered to be a logical instruction.	True
□	□	is applicable to numeric data only.	True
□	□	will translate internal code to binary.	False
□	□	suppresses leading zeros.	True
□	□	is used when output is on punched cards.	False
□	□	is used to insert symbols in data to be printed.	True
□	□	uses a programmer-designed mask.	True

177. The ability of a computer to change the order in which its instructions are executed has made the computer an extremely flexible data processing tool.

Programs can be written to provide for changes in the order of execution depending on the results of comparisons made between two fields of data. The instruction set of a given computer system will usually offer several different methods of performing this function.

Changing the sequence is usually dependent upon the combined action of two instructions. The first (Compare) will examine the two data fields and set an internal indicator that reflects the outcome of this comparison. The second instruction (Branch on Condition) will test this indicator and will branch or not branch to another point in the program depending on the condition of the indicator.

All Compare instructions in the 9200/9300 will perform two functions. They will compare two fields of data and:

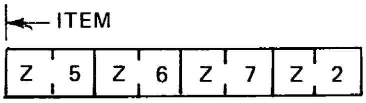
□ add the two data items.

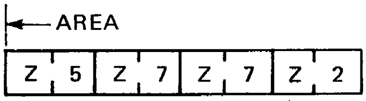
□ set an internal condition code indicator.

□ change the sequence of instructions.

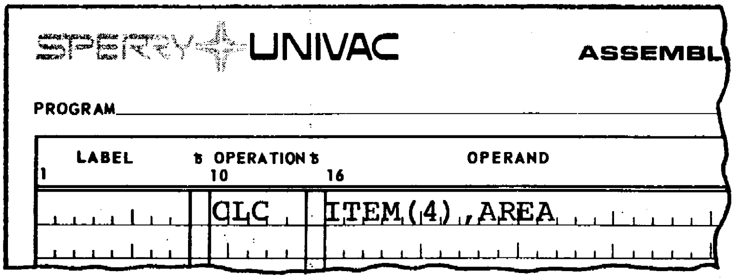
set an internal condition code indicator.

178. The Compare Logical (CLC) instruction tests the binary value of two operands. Assume that the two numeric operands are in memory as shown in the following character (unpacked) format and that they have been read in from cards.





The Compare Logical instruction would be written as shown below:


SPERRY UNIVAC
ASSEMBL
PROGRAM
LABEL
OPERATION
OPERAND
1
10
16

```
         CLC   ITEM(4),AREA
```

What are the two functions performed by this instruction?

1. ____________________

2. ____________________

Which field has the greatest binary value? ______________

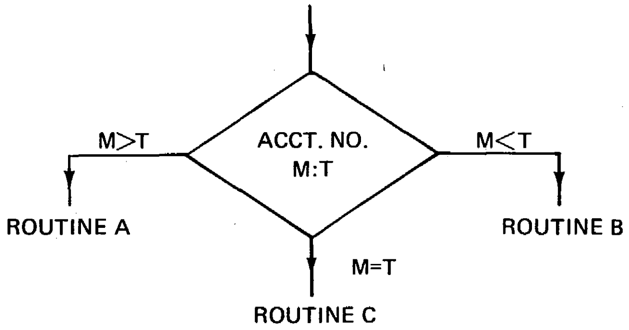
Compares the binary values of the fields.

Sets the condition code indicator

AREA

179. The CLC instruction is useful when comparing two account numbers. If we are posting transactions (T) to a master file (M), we must be certain that each transaction is posted to the correct master record.

The following illustration shows the three possible conditions.



Which routine in the program should be followed if the condition code indicates the values to be equal.

□ ROUTINE A.

□ ROUTINE B.

□ ROUTINE C.

If the master record account number is less than the transaction account number, which routine will be followed? _____

Will the CLC instruction cause the program to branch to the correct routine?

□ Yes

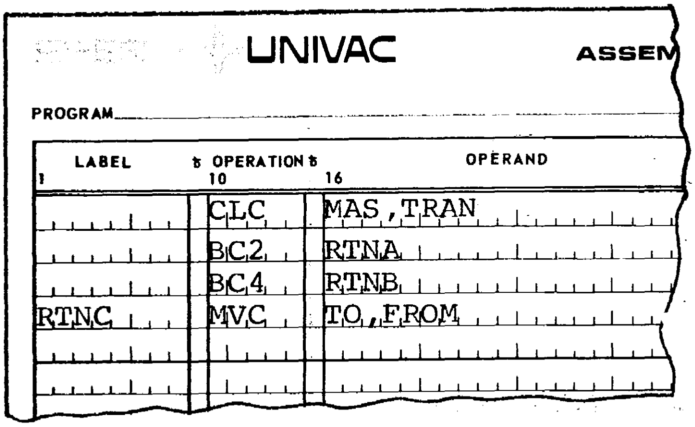
□ No

ROUTINE C

ROUTINE B

No

180. Since three different conditions are possible, we need instructions that will test the condition code indicator and branch to the correct routine. The Branch-on-Condition instructions can test this indicator and the branch will be based on the particular setting being tested. In the following code we have assumed that the length of each account number is four bytes.



UNIVAC ASSEM

PROGRAM

LABEL ␢ OPERATION ␢ OPERAND
1 10 16

```
1        10    16
         CLC   MAS,TRAN
         BC2   RTNA
         BC4   RTNB
RTNC     MVC   TO,FROM
```

The Branch on Condition 2(>) Instruction to Routine B will occur if the condition code is set to indicate:

MAS < TRAN

MAS > TRAN

MAS = TRAN

Neither the Branch on Condition 2 (>) Instruction nor the Branch on Condition 4 (<) Instruction will be executed if there is an ____________ Condition.

MAS < TRAN

equal

181. REVIEW

Which of the following are Assembler-directing instructions?

□ CLC

□ DS

□ START

□ BC

□ ED

MATCH the instruction on the left with the functional description on the right.

Instruction		Function
1. START	______	Tests the condition code indicator for a specific condition.
2. BC		
3. CLC	______	Compares the binary value of two fields.
4. DS	______	Marks the beginning of a program.
	______	Allocates memory.

When editing data, it is often necessary to suppress zeros. This means that:

□ all zeros are suppressed.

□ leading zeros are suppressed.

What kind of data can be edited?

□ Alphanumeric

□ Numeric

□ Alphabetic

DS

START

2

3

1

4

leading zeros are suppressed

Numeric

182. REVIEW

If the programmer wishes to edit a data field by inserting the $ and decimal point symbols, they must be included in the applicable position of the:

□ data.

□ instruction.

□ mask.

mask

What two functions are performed by the Compare Logical (CLC) instruction?

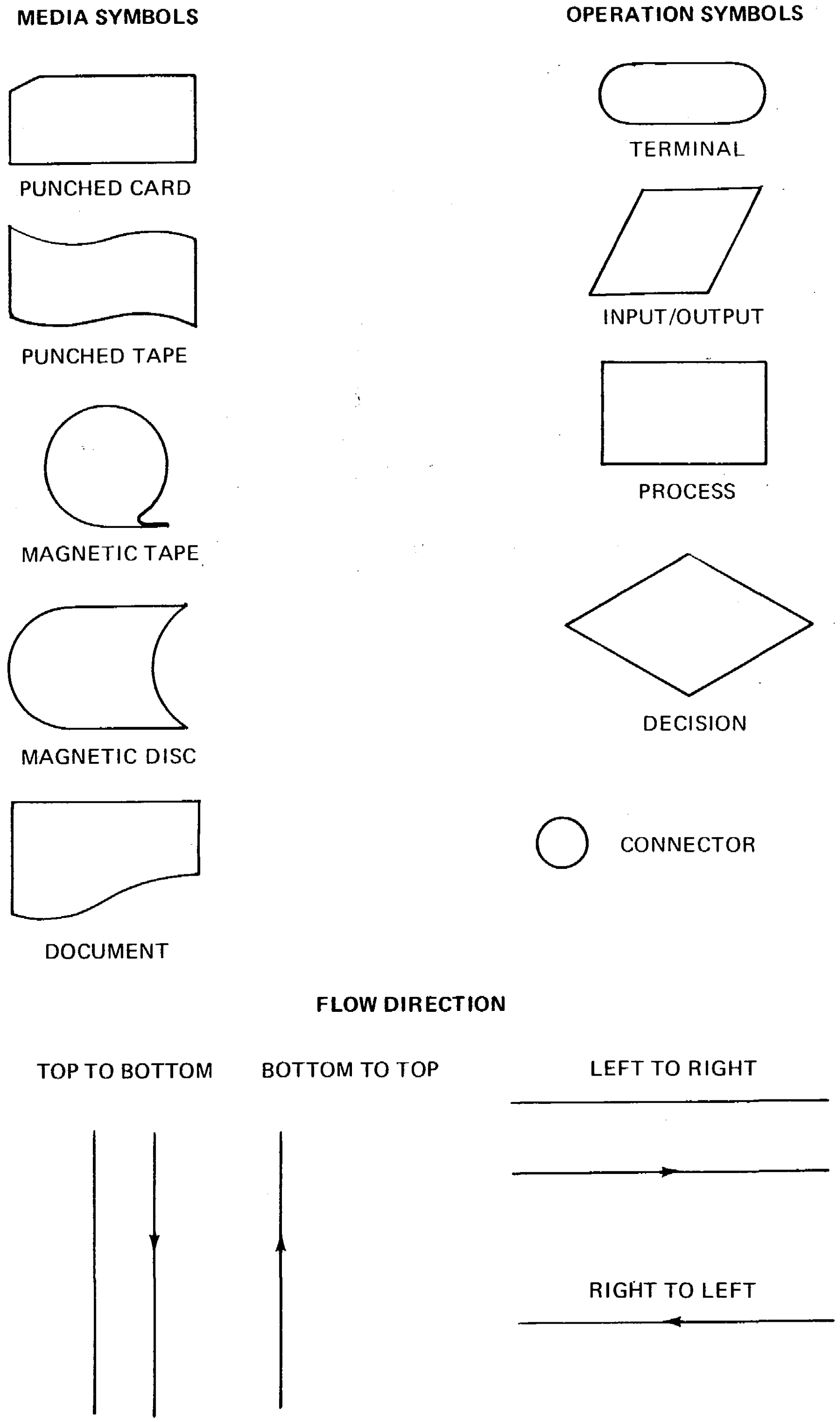
□ Performs a binary compare of two fields and sets an internal indicator.

□ Compares two fields and performs a branch.

Performs a binary compare of two fields and sets an internal indicator

PANEL 1

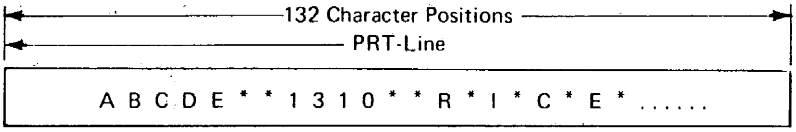
FLOWCHART SYMBOLS



PANEL 2

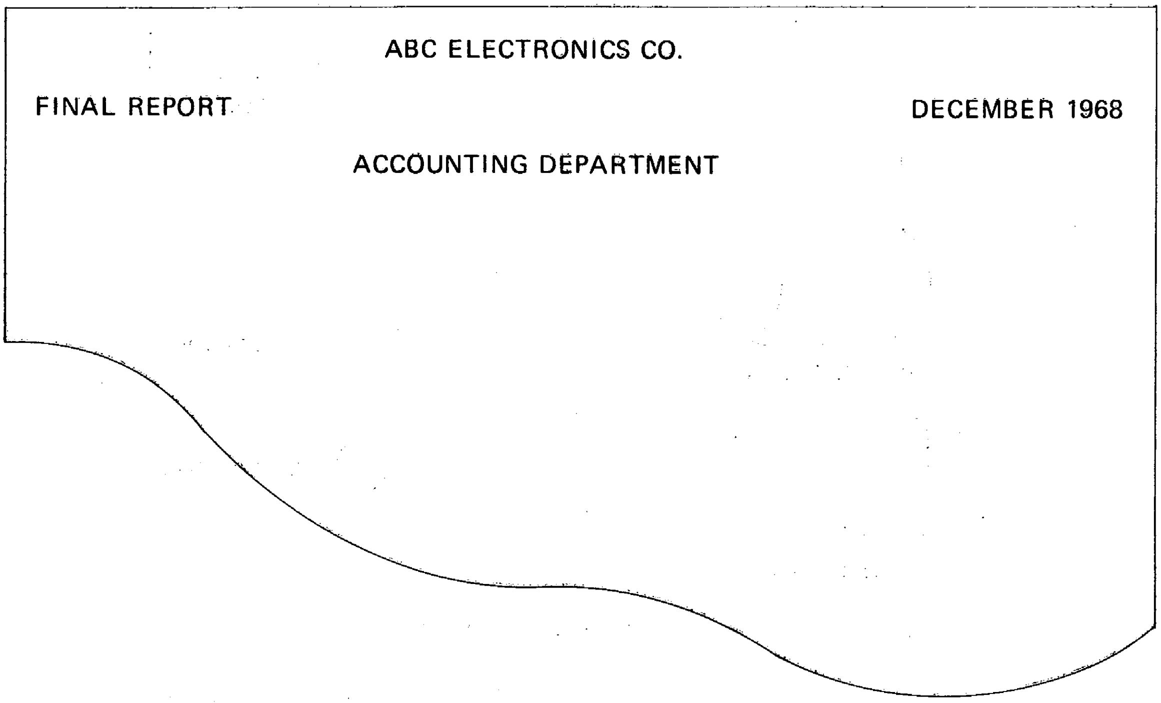
PRINTING

Part 1



This area of memory is equal in length to one line of type on the printer. Each character position across the page is represented.

Part 2



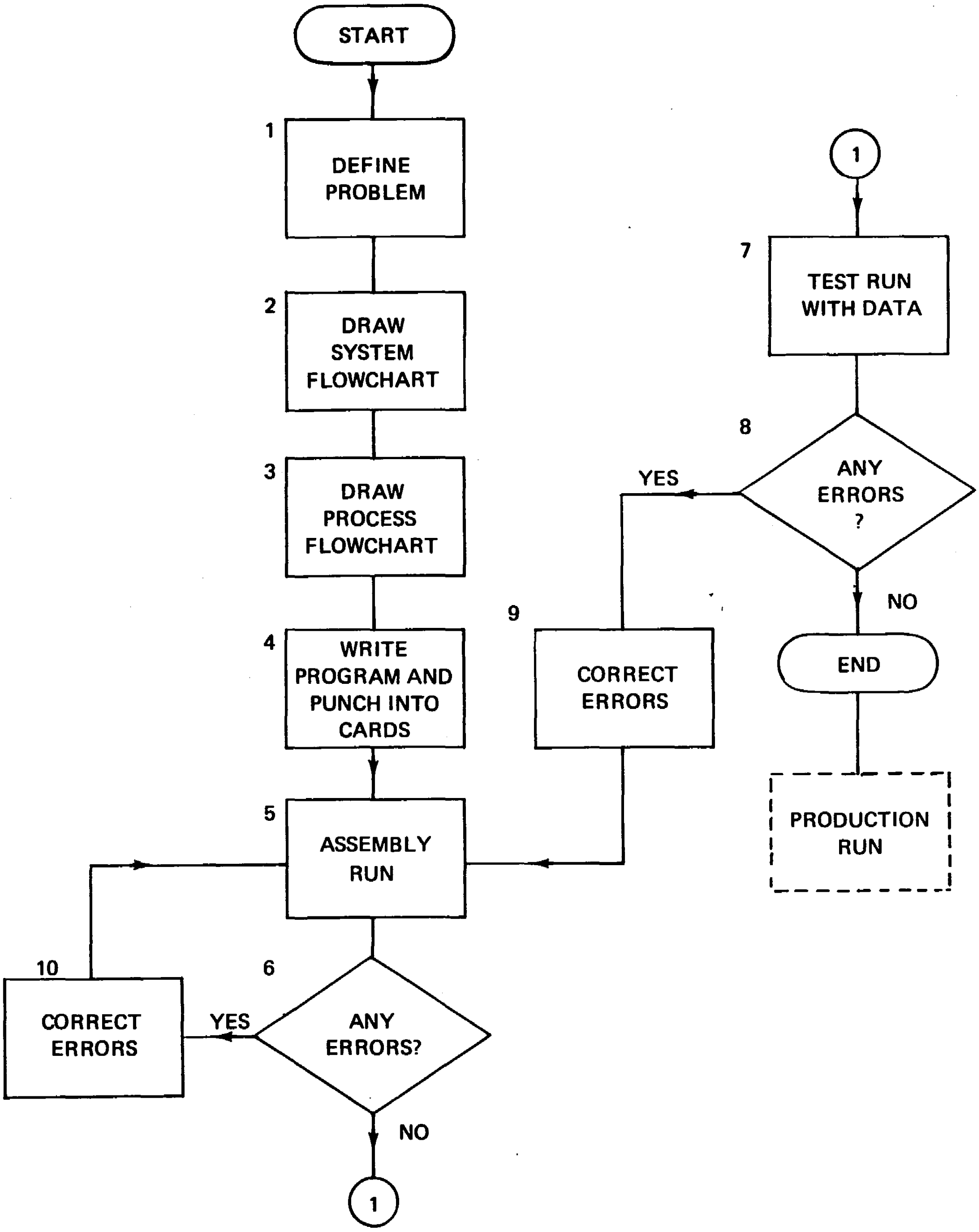
ABC ELECTRONICS CO.

FINAL REPORT DECEMBER 1968

ACCOUNTING DEPARTMENT

PANEL 3

ORGANIZATION



PANEL 4

CHARACTER CODE

EBCDIC	Printer Graphics
1100 0001	A
1100 0010	B
1100 0011	C
1100 0100	D
1100 0101	E
1100 0110	F
1100 0111	G
1100 1000	H
1100 1001	I
1101 0001	J
1101 0010	K
1101 0011	L
1101 0100	M
1101 0101	N
1101 0110	O
1101 0111	P
1101 1000	Q
1101 1001	R
1110 0010	S
1110 0011	T
1110 0100	U
1110 0101	V
1110 0110	W
1110 0111	X
1110 1000	Y
1110 1001	Z
1111 0000	0
1111 0001	1
1111 0010	2
1111 0011	3
1111 0100	4
1111 0101	5
1111 0110	6
1111 0111	7
1111 1000	8
1111 1001	9

EBCDIC	Printer Graphics
1111 1111	■ (lozenge)
0100 0000	Space
0100 1010	¢ (cents)
0100 1011	. (period)
0100 1100	< (Less than)
0100 1101	((open parenthesis)
0100 1110	+ (plus)
0100 1111	| (vertical)
0101 0000	& (ampersand)
0101 1010	! (exclamation)
0101 1011	$ (dollar sign)
0101 1100	* (asterisk)
0101 1101	) (close parenthesis)
0101 1110	; (semicolon)
0101 1111	— (logical NOT)
0110 0000	- (minus)
0110 0001	/ (slash)
0110 1010	∧ (logical AND)
0110 1011	, (comma)
0110 1100	% (percent)
0110 1101	_ (underline)
0110 1110	> (Greater than)
0110 1111	? (Question mark)
0111 1010	: (Colon)
0111 1011	# (Number)
0111 1100	@ (at rate of)
0111 1101	' (Apostrophe)
0111 1110	= (Equal)
0111 1111	" (Quotes)

UNIVAC 9000 CARD ASSEMBLER

PROGRAMMED INSTRUCTION COURSE

BOOK 1 – INTRODUCTION

SELF-TEST

1. Match each of the following basic data processing activities to its corresponding function by writing the appropriate letter in each blank:

A. Classify 1.______Arrange data in a desired numeric or alphabetic sequence.

B. Sort 2.______Arrange data in numeric or alphabetic categories.

C. Calculate 3.______Represent data in or on a physical medium.

D. Summarize 4.______Add, subtract, divide, or multiply.

E. Record 5.______Arrange the results of processing in condensed form.

2. Match each of the following functions with a corresponding device:

A. Input 1.______Card reader

B. Output 2.______Card punch

C. Input/Output 3.______Magnetic tape unit

D. Processing 4.______Magnetic disc unit

E. Off-line 5.______Paper tape unit

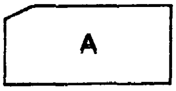
6.______Central Processor

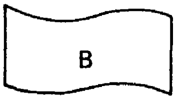
7.______Printer

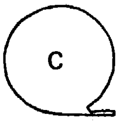
8.______Console typewriter

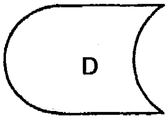
9.______Keypunch

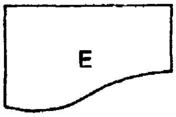
3. Match the following:











1.________Magnetic disc

2.________Punched card

3.________Printed document

4.________Magnetic tape

5.________Punched paper tape

6.________Keypunch

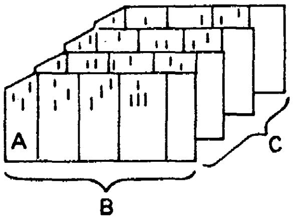
7.________Magnetic tape unit

8.________Printer

9.________Magnetic disc unit

10.________Paper tape unit

4. Match each of the letters in the punched card illustration below with the corresponding designation:



1.________File

2.________Field

3.________Record

5. Match the following:

A. Instruction

B. Operation code

C. Operand

1._______Defines storage address of data.

2._______A mnemonic.

3._______Specifies operation to be performed and defines address of data.

6. Match the following:

A. Symbolic language

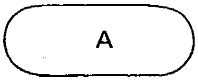
B. Machine language

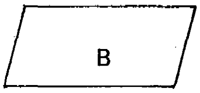
1.______Assembly coding

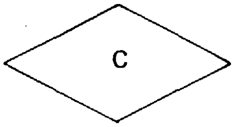
2.______Object program

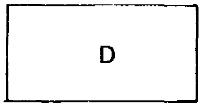
3.______Source program

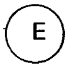
7. Match each of the following symbols to its corresponding function:











1.______Input/Output

2.______Processing

3.______Connector

4.______Terminal

5.______Decision point

8. Match the following:

A. $S > T$

B. $S < T$

C. $S \leq T$

D. $S \geq T$

E. $S : T$

F. $S \neq T$

1.______S is less than T.

2.______S is greater than T.

3.______S and T are unequal.

4.______S is compared to T.

5.______S is greater than or equal to T.

6.______S is less than or equal to T.

9. Match each operational step listed below with the appropriate flowchart symbol:

1.______STOP

2.______READ A CARD

3.______WRITE PRINT LINE

4.______ZERO TOTAL

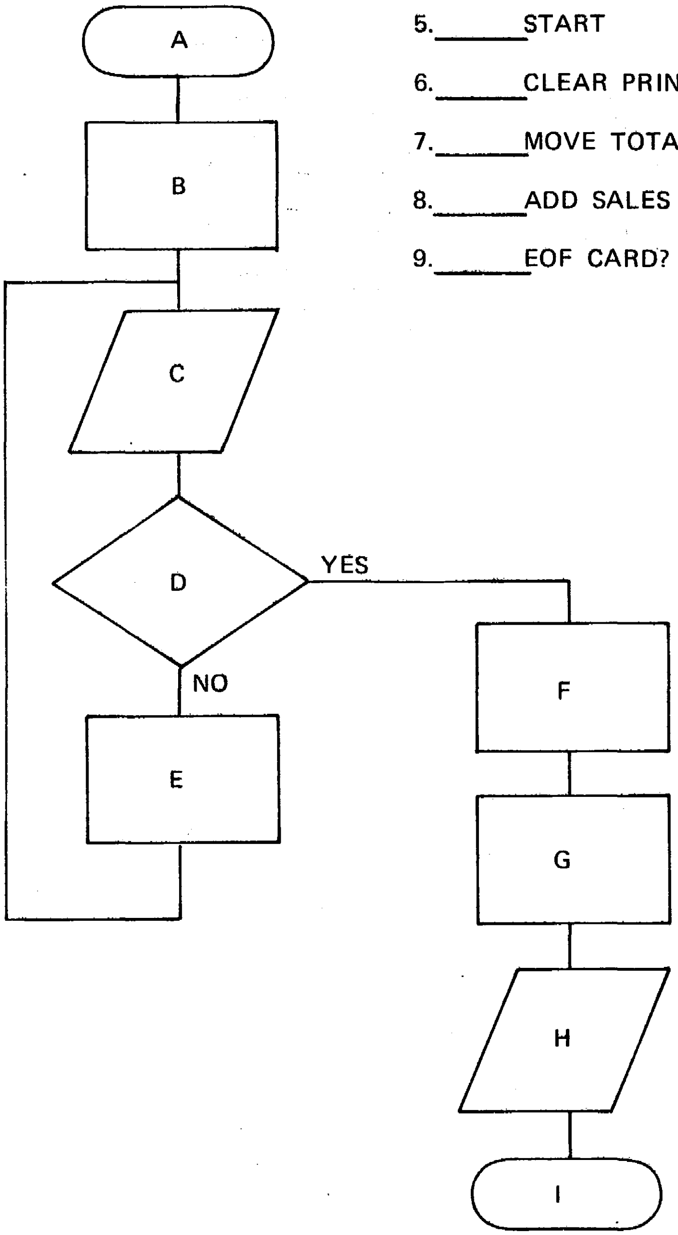
5.______START

6.______CLEAR PRINT LINE

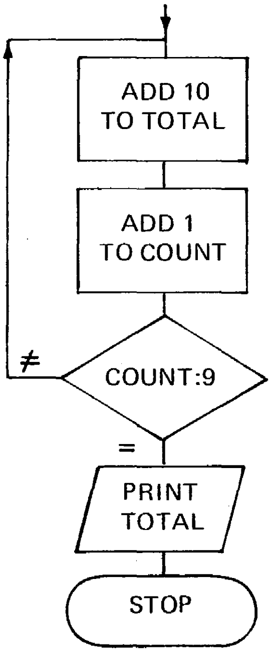
7.______MOVE TOTAL TO PRINT LINE

8.______ADD SALES TO TOTAL

9.______EOF CARD?



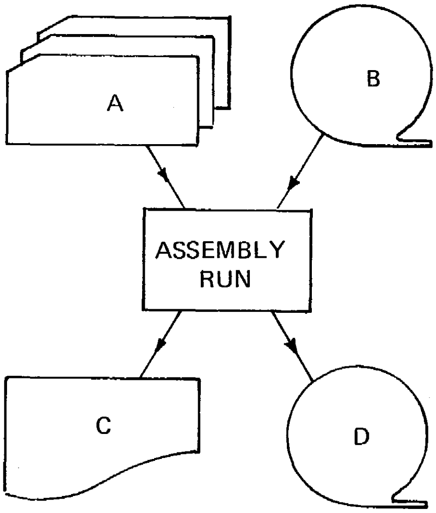
10. Assume that COUNT has been set to zero in the program represented by the flowchart section below.



The ADD 10 TO TOTAL operation will then be performed:

A. 1 time
B. 4 times
C. 9 times
D. 10 times

11. Match the letter in each of the symbols below to a corresponding designation:



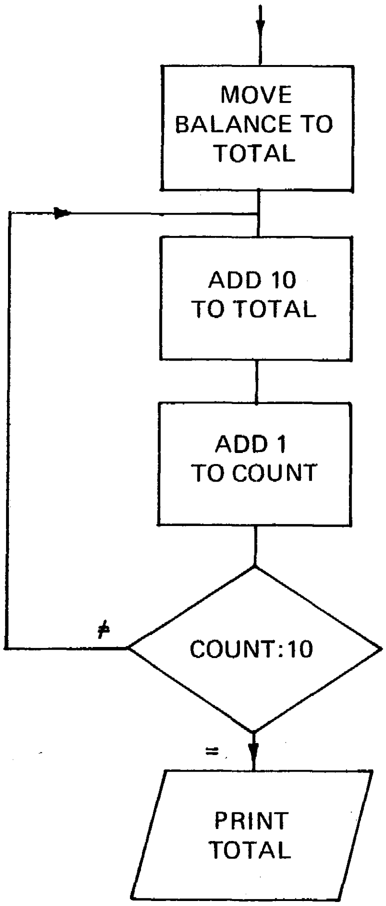
1._______Listing

2._______Assembly program

3._______Object program

4._______Source program

12. As indicated in this flowchart segment, what value will be printed if COUNT is set to zero and BALANCE contains the value 240?



A. 240

B. 300

C. 340

D. 400

13. Number the following 6 functions to indicate the order in which they are performed:

______Assembly run

______Process flowchart

______Test run

______Debug Assembly run

______Production run

______Write Source program

14. An EOF card indicates the end of:

A. Field
B. Record
C. File
D. Program

15. A program is a series of:

A. Questions
B. Instructions
C. Data
D. Operands

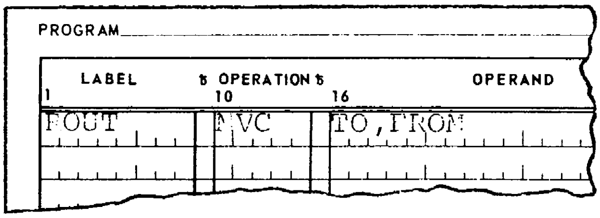
16. The basic function of high-speed memory is to store:

A. Data and instructions
B. Bytes
C. Bits
D. Registers

17. An object program is in:

A. Decimal code
B. Symbolic code
C. Mnemonic code
D. Machine code

18.



PROGRAM

LABEL 1	ƀ	OPERATION 10	ƀ	16 OPERAND
FOUT		MVC		TO,FROM

The above statement represents:

A. A program
B. An instruction
C. An operation code
D. An operand

19. A source program is written in:

A. Decimal code
B. Binary code
C. Symbolic code
D. Machine code

20. A byte contains:

A. 8 bits
B. 4 bits
C. 2 bits
D. 1 bit

21. A process flowchart is prepared by a:

A. Systems analyst
B. Programmer
C. Operator
D. Designer

22. A systems flowchart is generally prepared by a:

A. Manager
B. Operator
C. Programmer
D. Systems Analyst

23. The symbols in a systems flowchart represent:

A. Operands
B. Instructions
C. Media
D. Coding

24. The symbols in a process flowchart represent:

A. Operational steps
B. Media
C. Devices
D. Storage

25. An Assembler program is produced by the:

A. Computer manufacturer
B. Systems Analyst
C. Programmer
D. Computer user

26. A storage area can be cleared for use in arithmetic calculation by filling it with:

A. Numbers
B. Spaces
C. Zeroes
D. Letters

27. A storage area can be cleared for use in a printing operation by filling it with:

A. Numbers
B. Spaces
C. Zeroes
D. Letters

28. The end of a punched card file is indicated by an:

A. EOV label record
B. EOV card
C. EOF label record
D. EOF card

29. Data for arithmetic calculations must be in which format?

A. Unpacked
B. Alphanumeric
C. Packed decimal
D. Hexadecimal

30. Which of the following can be used in a flowchart to represent a repetitive process?

A. Device symbol
B. Media symbol
C. Processing symbol
D. Connector symbol

31. Which of the following is an Assembler-directing instruction?

A. Edit
B. Start
C. Move
D. Branch

UNIVAC 9000 CARD ASSEMBLER

PROGRAMMED INSTRUCTION COURSE

BOOK 1 – INTRODUCTION

SELF-TEST ANSWERS

1	1:B	(1 point each)
	2:A	
	3:E	
	4:C	
	5:D	
2	1:A	(1 point each)
	2:B	
	3:C	
	4:C	
	5:C	
	6:D	
	7:B	
	8:C	
	9:E	
3	1:D	(1 point each)
	2:A	
	3:E	
	4:C	
	5:B	
	6:A	
	7:C	
	8:E	
	9:D	
	10:B	
4	1:C	(1 point each)
	2:A	
	3:B	
5	1:C	(1 point each)
	2:B	
	3:A	
6	1:A	(1 point each)
	2:B	
	3:A	
7	1:B	(1 point each)
	2:D	
	3:E	
	4:A	
	5:C	
8	1:B	(1 point each)
	2:A	
	3:F	
	4:E	
	5:D	
	6:C	
9	1:I	(1-1/3 points each)
	2:C	
	3:H	
	4:B	
	5:A	
	6:F	
	7:G	
	8:E	
	9:D	
10	C	(5 points)
11	1:C	(1 point each)
	2:B	
	3:D	
	4:A	
12	C	(5 points)
13	1:3	(2 points each)
	2:4	
	3:5	
	4:4	
	5:6	
	6:2	
14	C	(1 point)
15	B	(1 point)
16	A	(1 point)
17	D	(1 point)
18	B	(1 point)
19	C	(1 point)
20	A	(1 point)
21	B	(1 point)
22	D	(1 point)
23	C	(1 point)
24	A	(1 point)
25	A	(1 point)
26	C	(1 point)
27	B	(1 point)
28	D	(1 point)
29	C	(1 point)
30	D	(1 point)
31	B	(1 point)