A computer performs three basic functions : it accepts data (input), processes on such data by performing arithmetic and logical operations (processing), finally it generates desired data (output). To solve a problem, it can perform above functions with high speed and accuracy. However, a computer can not think, not even take decisions of its own. It can not also decide a method for solving a given problem. Users have to instruct the computers how or in what way the task is to be performed. A set of such instructions is called a program. The most critical and basic part in a programming is a careful and systematic analysis of the whole problem about the solution of the given problem. While attempting to solve a given physical problem using a computer, following steps should be followed in order :

① Convert the given physical problem to a mathematical problem.

② Formulate the corresponding mathematical data or equations.

③ Choose an appropriate method to solve those equations.

④ Proceed for a systematic step by step analysis to solve the problem using above method (algorithm).

⑤ Represent all the steps in the above algorithm graphically. Thus a detailed chart is drawn (flow chart).

⑥ Convert the flow chart into a source program by using a high level programming language (*e.g.*, FORTRAN 77/90).

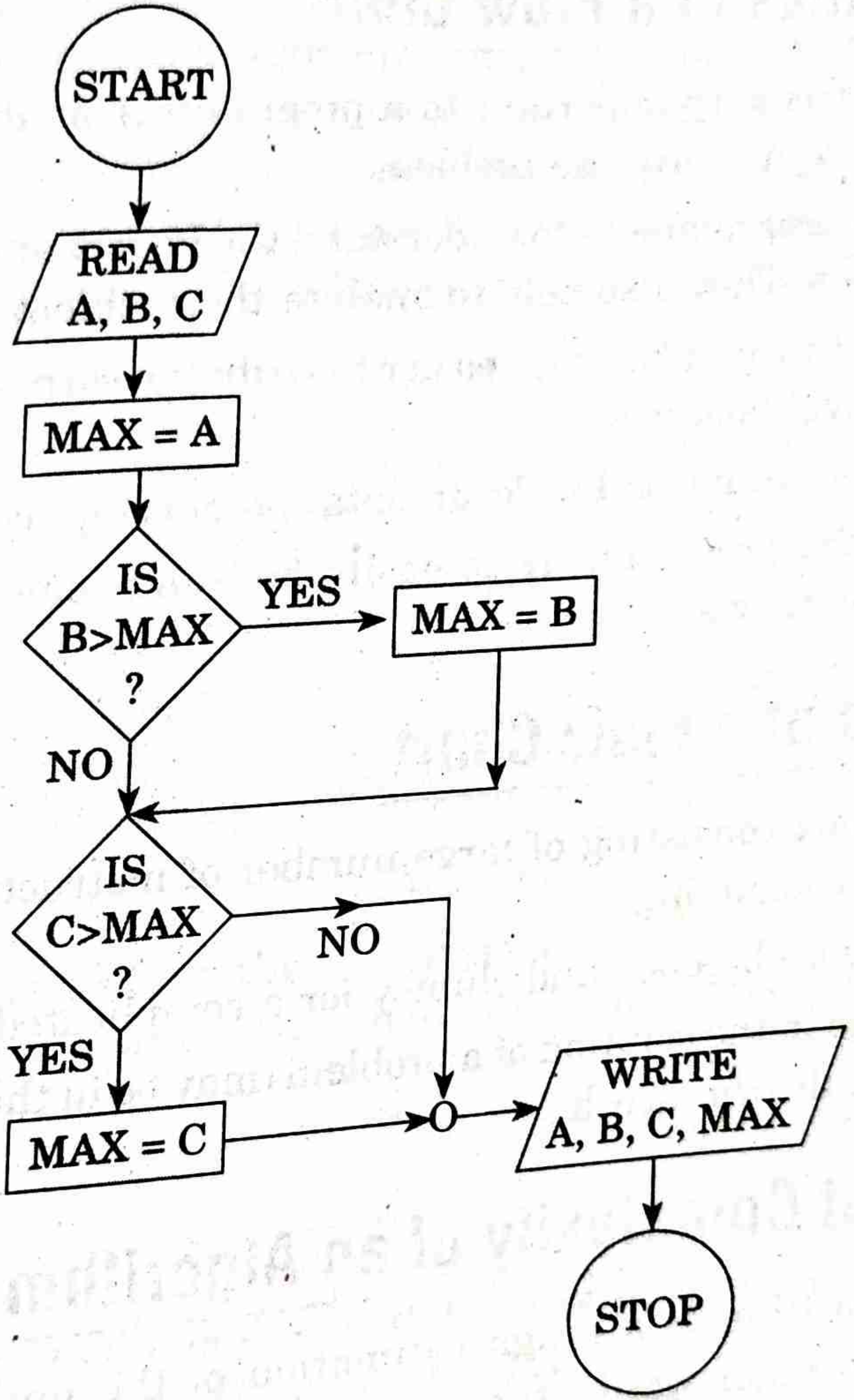
⑦ Compile and execute the program using corresponding compiler (at first object program then executable file will be generated).

1. BASIC
2. FORTRAN
3. LISP
4. COBOL
5. Pascal
6. C
7. C ++
8. Java
9. Python
10. C#

Algorithm I :

```
STEP 1 : START
STEP 2 : READ A, B, C
STEP 3 : MAX = A
STEP 4 : IF (B > MAX) MAX = B
STEP 5 : IF (C > MAX) MAX = C
STEP 6 : WRITE A, B, C, MAX
STEP 7 : STOP
```

- *Flow chart 1 :*

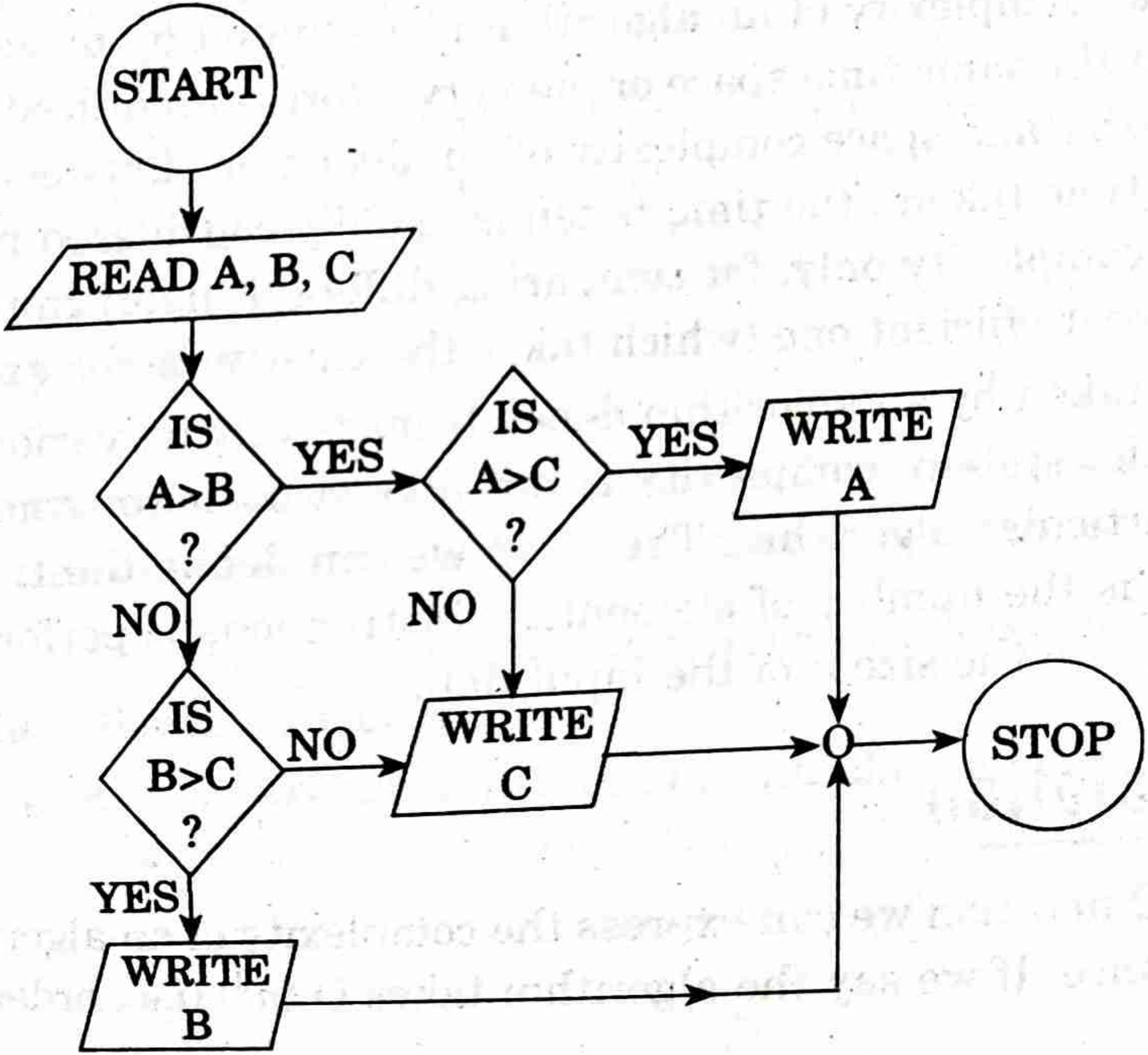


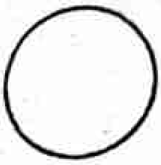
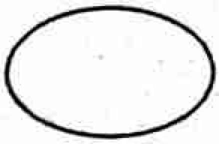
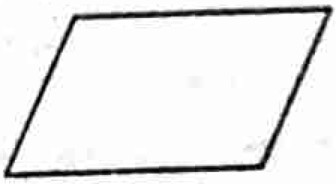
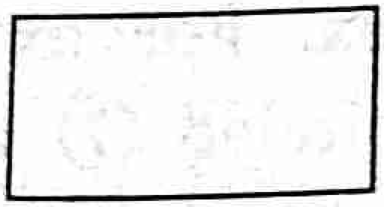
START
READ A, B, C
MAX = A
IS B>MAX ?
YES
MAX = B
NO
IS C>MAX ?
NO
YES
MAX = C
WRITE A, B, C, MAX
STOP

Algorithm 2 :

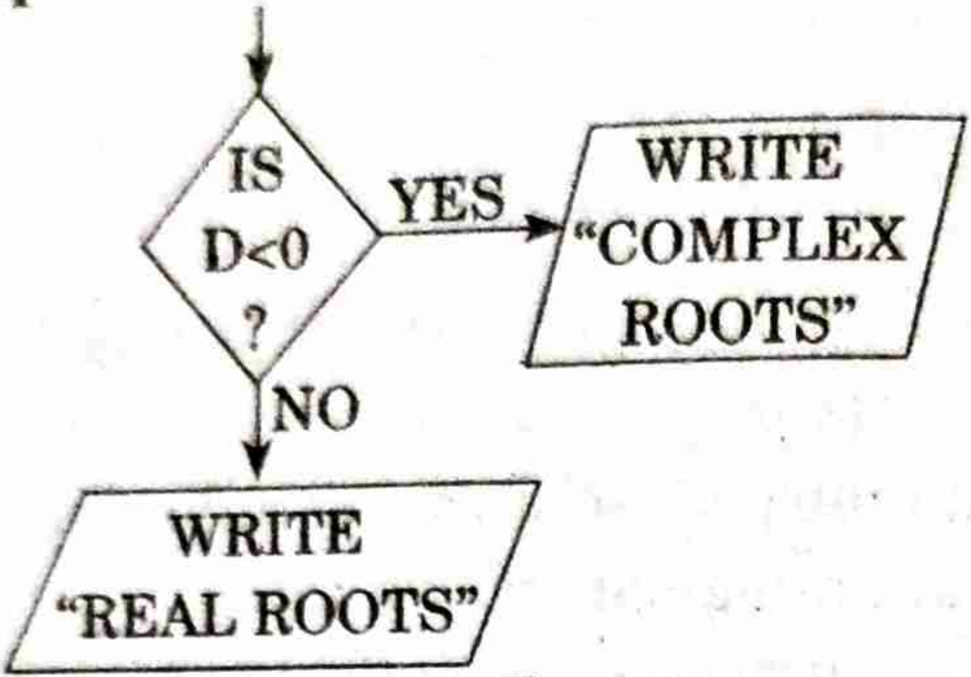
```
STEP 1 : START
STEP 2 : READ A, B, C
STEP 3 : IF (A > B) GO TO STEP 4, OTHERWISE GO TO STEP 5
STEP 4 : IF (A > C) WRITE “A IS MAXIMUM”, OTHERWISE WRITE “C IS MAXIMUM”
         GO TO STEP 6
STEP 5 : IF (B > C) WRITE “B IS MAXIMUM”, OTHERWISE  “C IS MAXIMUM”
STEP 6 : STOP
```

- **(Alternative) Flow Chart 2 :**



Sl. No.	Flow Chart Symbol
1.	 OR 
2.	↓↑←→
3.	
4.	

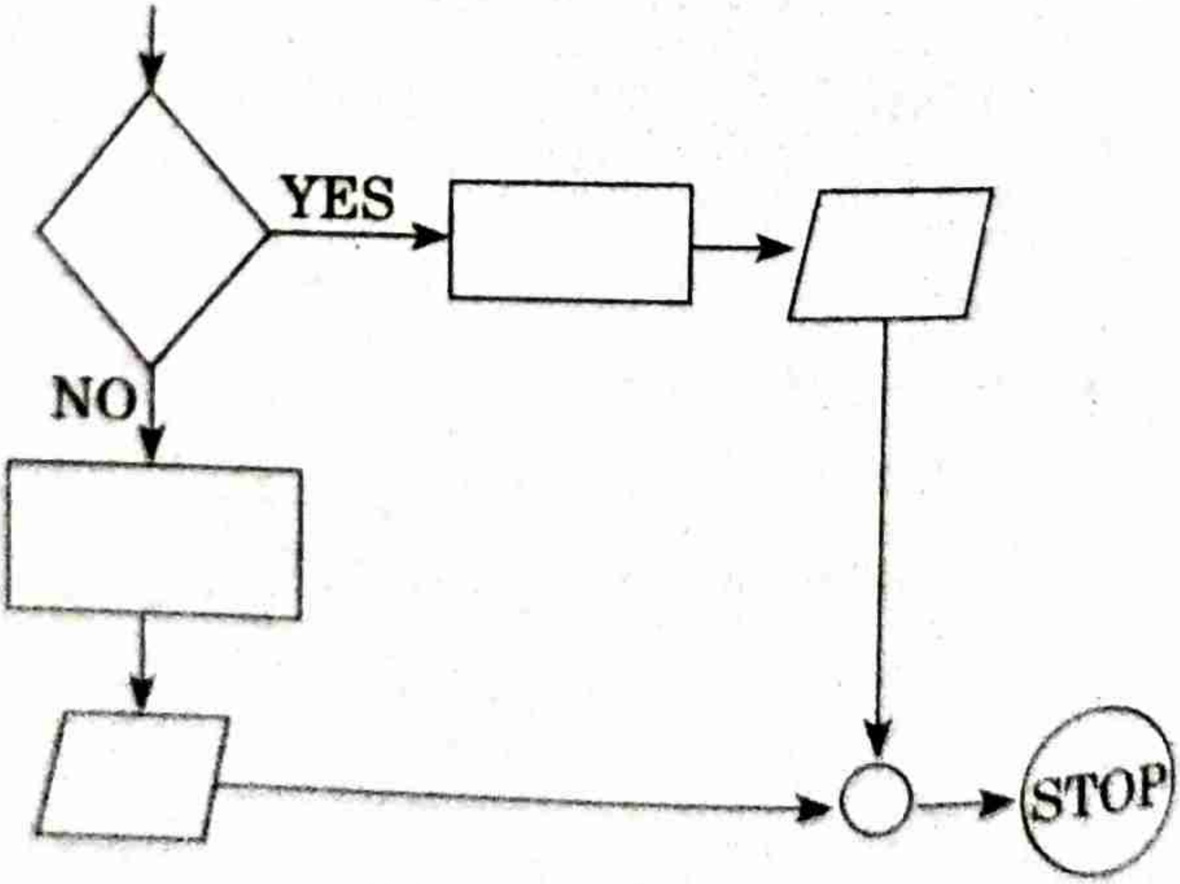
(iv) **Example :**

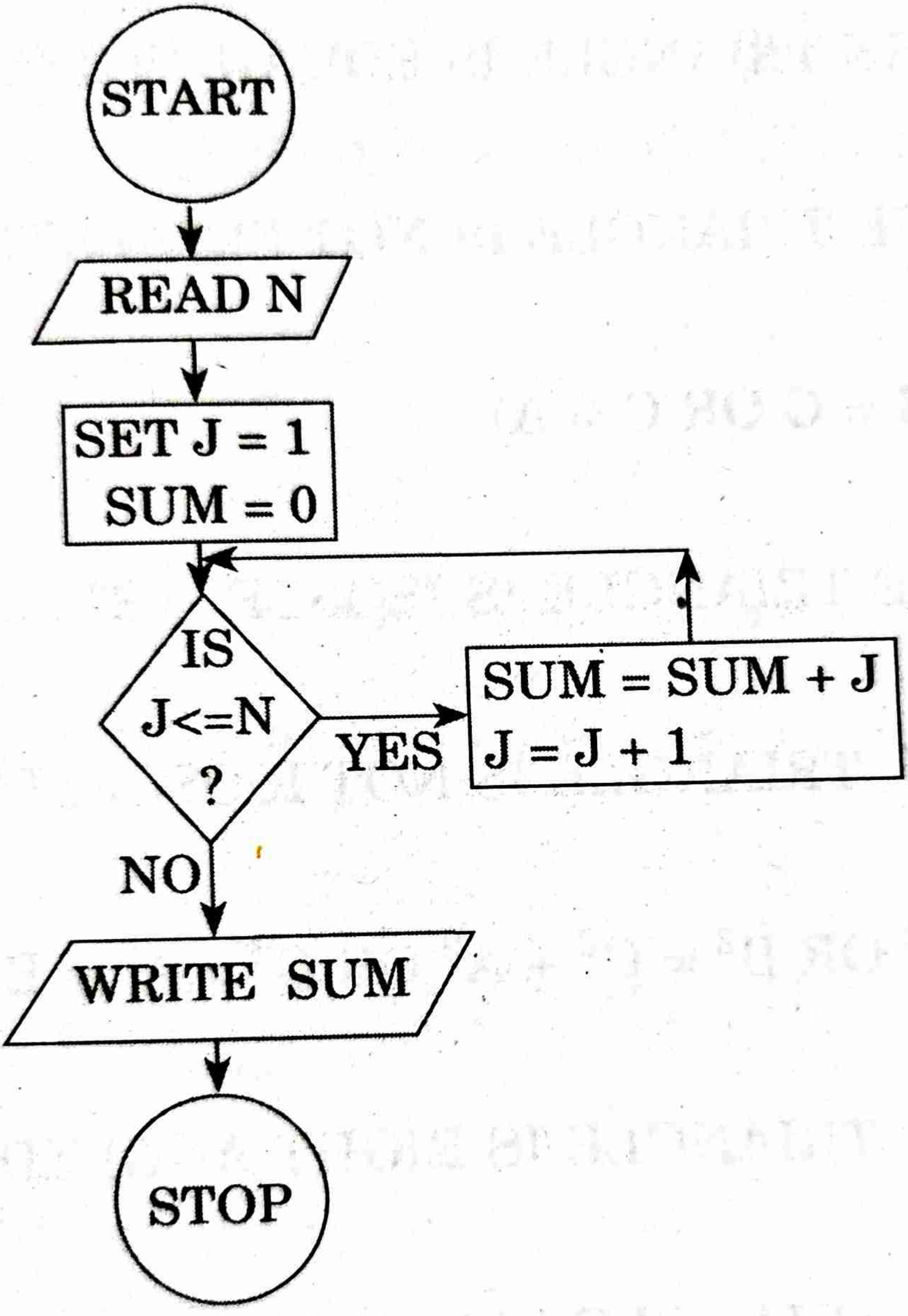


(i) This a symbol (a small circle) in flow chart used as
connector

(ii) It allows the flow chart to be drawn without intersecting
lines or without a reverse flow

Example :

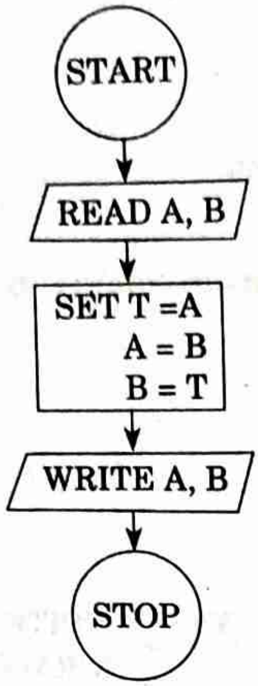



START
READ N
SET J = 1
SUM = 0
IS
J<=N
?
YES
SUM = SUM + J
J = J + 1
NO
WRITE SUM
STOP

7. **Draw a flow chart to interchange or swap two numbers.**

Solution Let us denote the two numbers by A and B.

Flow Chart 6 :



8. **Write an algorithm to determine the quotient Q and remainder R after dividing M by N (M > N, M, N : positive integers)**

Solution **Algorithm 9 :**

```
STEP 1 : START
STEP 2 : READ M, N
STEP 3 : Calculate Q = Integral Part (M / N)
STEP 4 : Calculate R = M – N × Q
STEP 5 : WRITE Q, R
STEP 6 : STOP
```

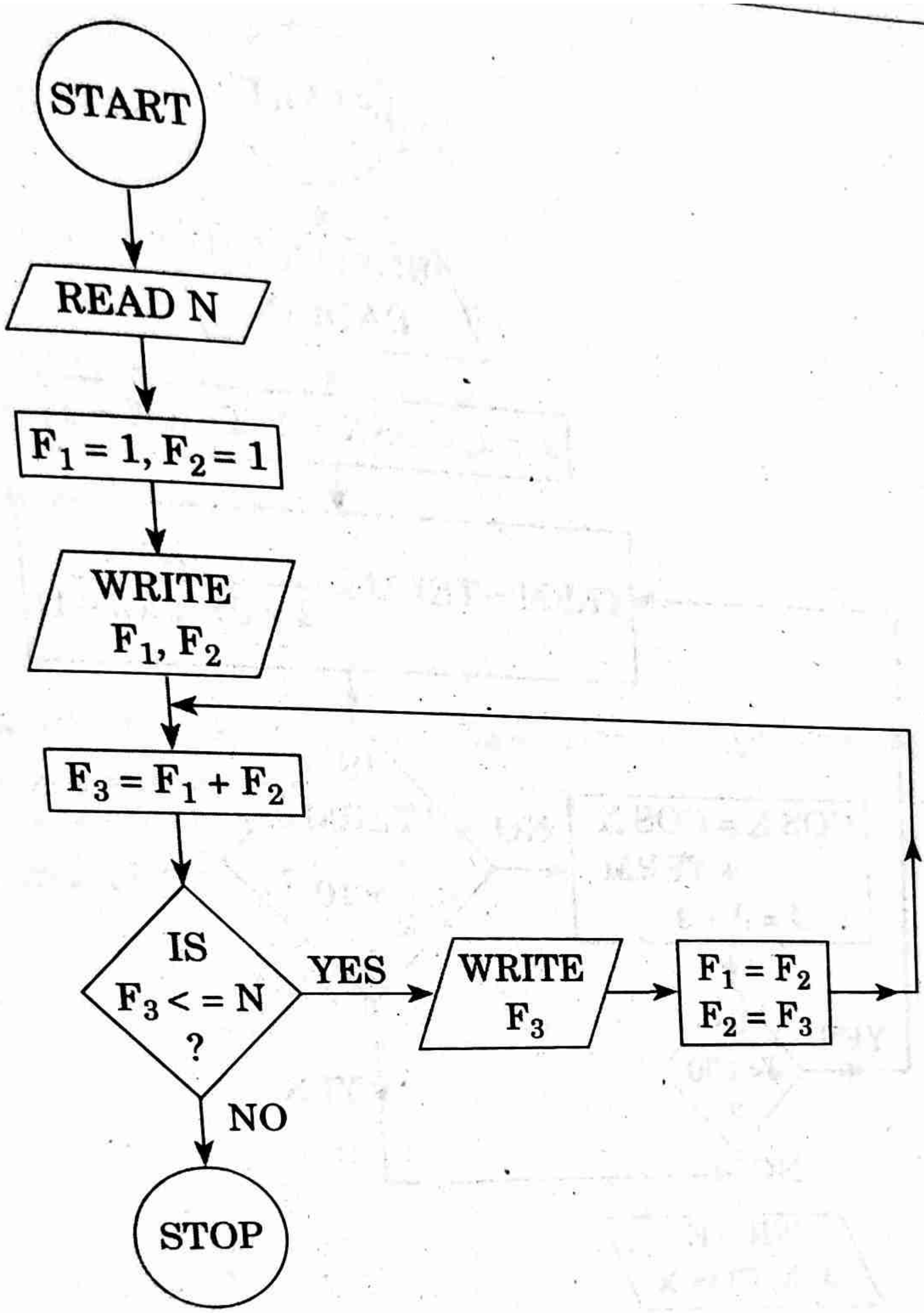
9. **Design an algorithm to compute GCD and LCM of two integers M and N.** [CP 2017]

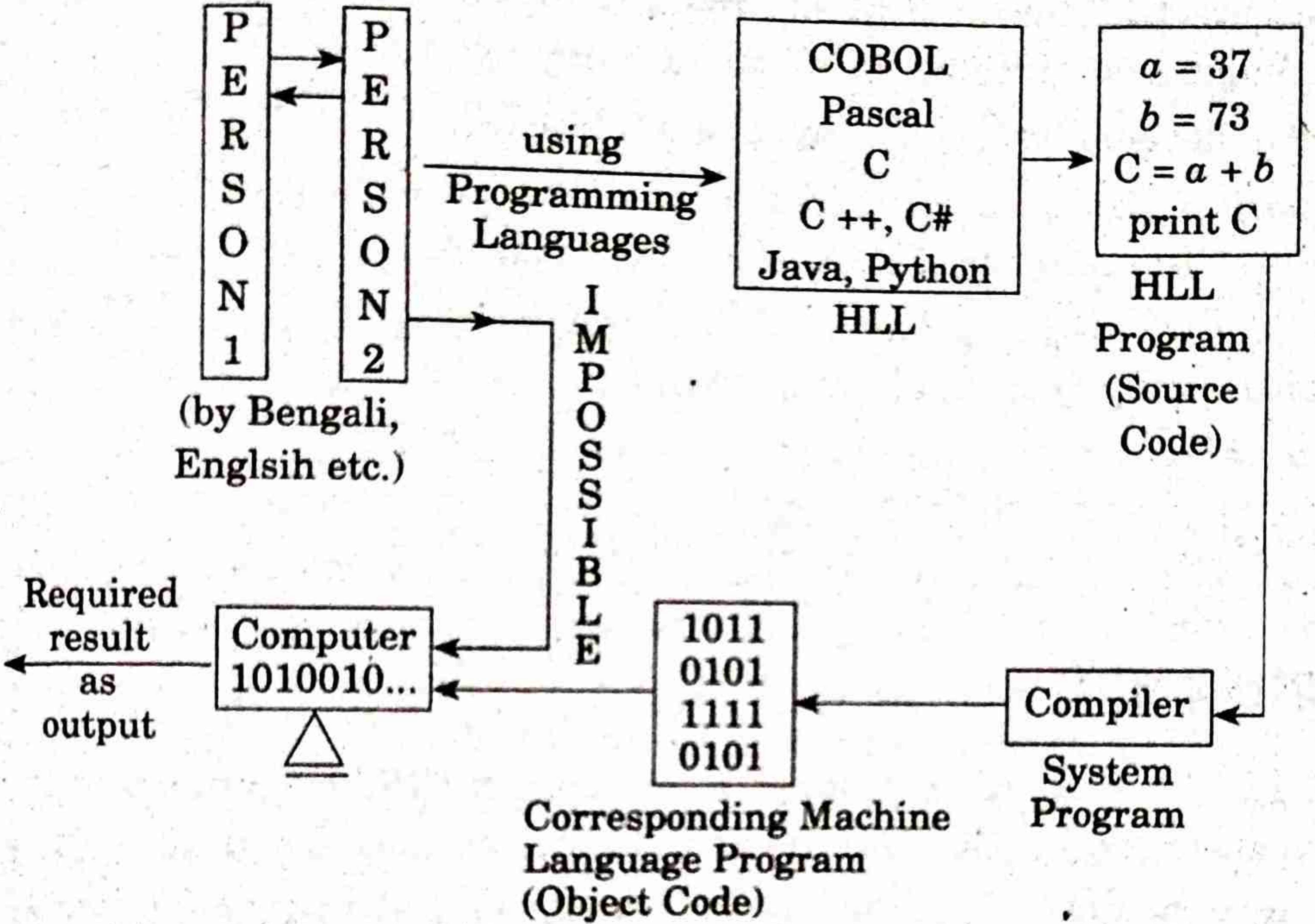
Solution **Algorithm 10 :**

```
STEP 1 : START
STEP 2 : READ M, N
STEP 3 : IF (N > M) EXECUTE STEP 4–6, ELSE GO TO STEP 7
STEP 4 : T = N
STEP 5 : N = M  /* Interchange M and N */
STEP 6 : M = T
STEP 7 : P = M × N  /* product P evaluated */
STEP 8 : Q = INT (M / N)  /* Quotient Q evaluated */
STEP 9 : R = M – N × Q   /* R : Remainder */
STEP 10 : IF (R = 0) THEN GO TO STEP 11, ELSE
            BEGIN
               M = N
               N = R
               GO TO STEP 8
            END
         END IF
STEP 11 : L = P / N        /* L : LCM, N : GCD */
SETP 12 : WRITE N, L
SETP 13 : STOP
```

Algorithm 12 :

```
STEP 1 : START
STEP 2 : READ N
STEP 3 : IF (N < 0)
         THEN
            WRITE "A WRONG VALUE OF N RECEIVED" GO TO STEP 10
         ELSE IF (N = 0)
         THEN
            WRITE "VALUE OF FACT = 1" GO TO STEP 10
         ELSE
            BEGIN
STEP 4 : FACT = 1
STEP 5 : J = 1
STEP 6 : FACT = FACT × J
STEP 7 : J = J + 1
STEP 8 : IF (J ≤  N) THEN GO TO STEP 6; ELSE GO TO STEP 9
STEP 9 : WRITE N, FACT
            END
         ENDIF
STEP 10 : STOP
```


START
READ N
$F_1 = 1, F_2 = 1$
WRITE F_1, F_2
$F_3 = F_1 + F_2$
IS $F_3 < = N$?
YES
WRITE F_3
$F_1 = F_2$
$F_2 = F_3$
NO
STOP


PERSON 1
PERSON 2
(by Bengali, Englsih etc.)
using Programming Languages
COBOL
Pascal
C
C ++, C#
Java, Python
HLL
a = 37
b = 73
C = a + b
print C
HLL Program (Source Code)
IMPOSSIBLE
Required result as output
Computer 1010010...
1011
0101
1111
0101
Corresponding Machine Language Program (Object Code)
Compiler
System Program

9.1 Inportance and Reasons of Learning C

① C is **the most popular programming language** in the world, even in 2020. According to a survey, C is the number one programming language in the sense that maximum number or programs are written today using C. Second position is occupied by Java.

② C is the basis of most of the other programming languages (*e.g.*, Java, .NET, Python, Mysql etc.) *i.e.*, **to learn other languages, learning of C is must**. Because the functalities of other languages are to be learnt from C.

③ Most of the **device drivers** are written in C. Device drivers are systems programs need to be installed in the computer for various devices like printer, scanner etc.

④ The major parts of many operating systems (operating system is also a system program) such as **Windows**, **Unix**, **Linux**, **Android** (popular mobile operating system) are written using C language.

⑤. **Oracle** (one of the biggest softwares in database sector) is written in C. This is a very useful and a successful software.

⑥ The major portion of many famous internet browsers like Google Chrome, Mozilla Firefox, Safari, Opera etc. are written in C.

⑦ C is important to build good programming skills.

⑧ C is very popular for hardware interaction (*i.e.*, to develop hardware level programming).

⑨ The useful electronic devices such as micro oven, washing machine, digital camera are being enriched due to incorporation of a microprocessor, an OS and an in-built embedded program. Those programs are written using C and make the devices fast to run.

⑩ Many professional 3D computer gaming frameworks (*e.g.*, DirectX) are built using C language. DirectX is an Application Program Interface (API) for creating and managing graphic images and multimedia effects in many applications.

⑪ C is used to develop both softwares and applications.

⑫ C is an easy language as compared to any other modern programming languages. It contains very few number of keywords which makes it simple to learn.

⑬ The backbone of Operating System (OS) development is C language. Most of the popular OS are either fully or partially written in C.

① C is a general purpose high level programming language.

② C is very powerful and extensively used language.

③ C programming language is widely known among the programmers around the world. Because of its huge popularity, it is acceptable for most of the system projects.

④ C language is stable (does not get updated much).

⑤ C is portable (both machine and operating system independent language).

⑥ C contains only 32 keywords, so C language is shorter in size.

⑦ C compilation time is faster than other programming languages.

⑧ C is a structured language (to handle a large problem, C divides it into smaller modules called functions/procedures, each of which performs a specific task) and can be understood and modified easily.

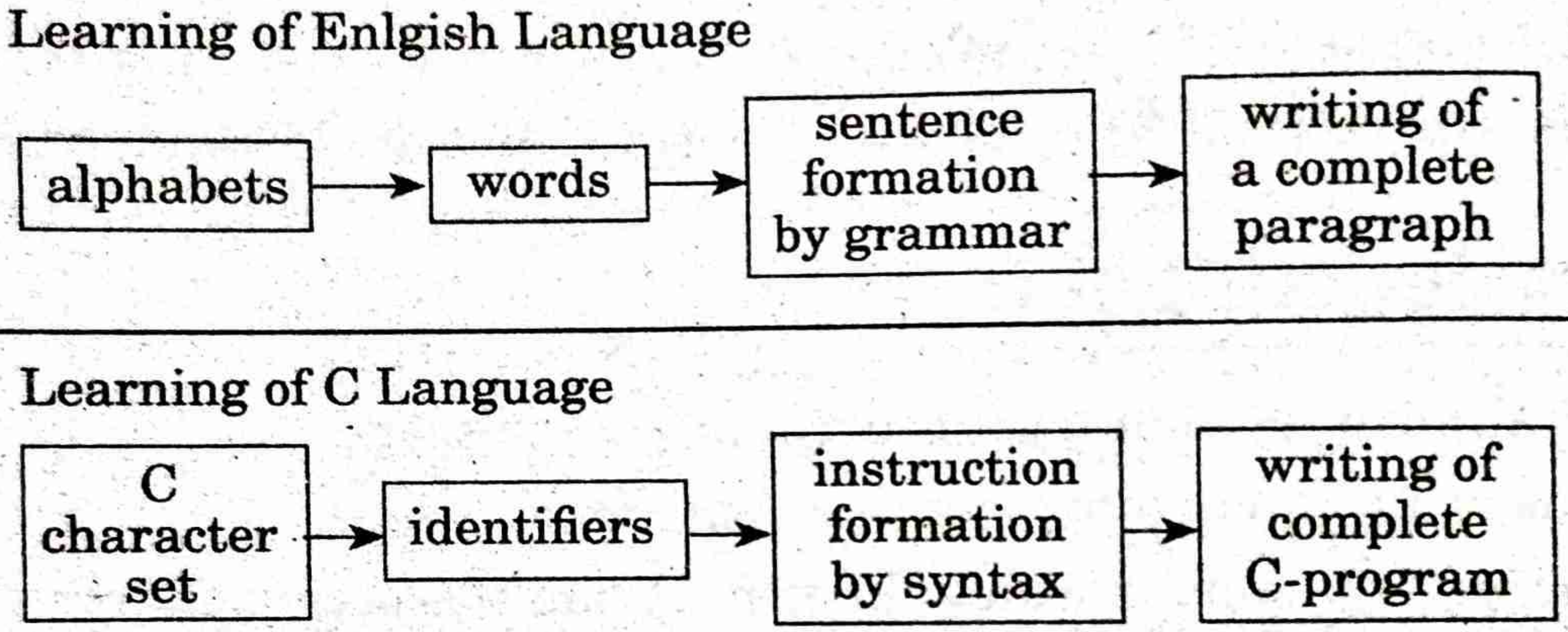
⑨ C supports an effective set of operators and a variety of data types as well.

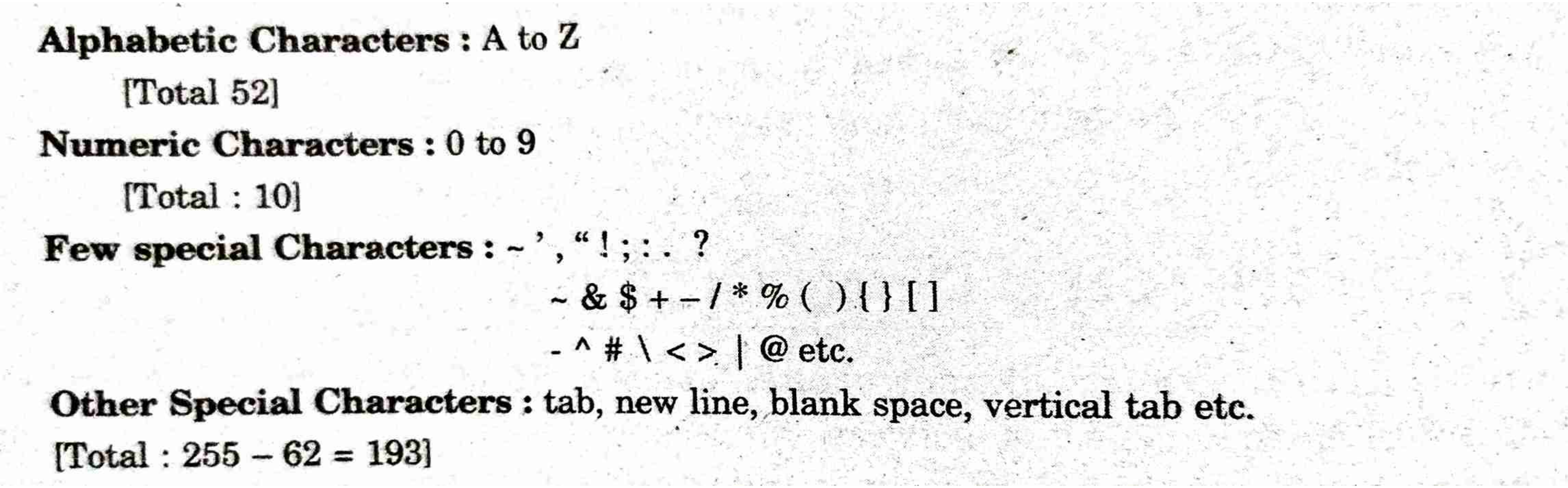
⑩ C has a very good pre-defined library functions.

⑪ C is a powerful, flexible and extensible programming language.

⑫ C is simple and easy to learn. C programs contain English like instructions. Thus using 'C' it is easy to write understand, modify and debug programs.

⑬ C is a case sensitive programming language. C reads lower case letters and upper case letters differently.


Learning of Enlgish Language
alphabets
words
sentence formation by grammar
writing of a complete paragraph
Learning of C Language
C character set
identifiers
instruction formation by syntax
writing of complete C-program



Alphabetic Characters : A to Z

[Total 52]

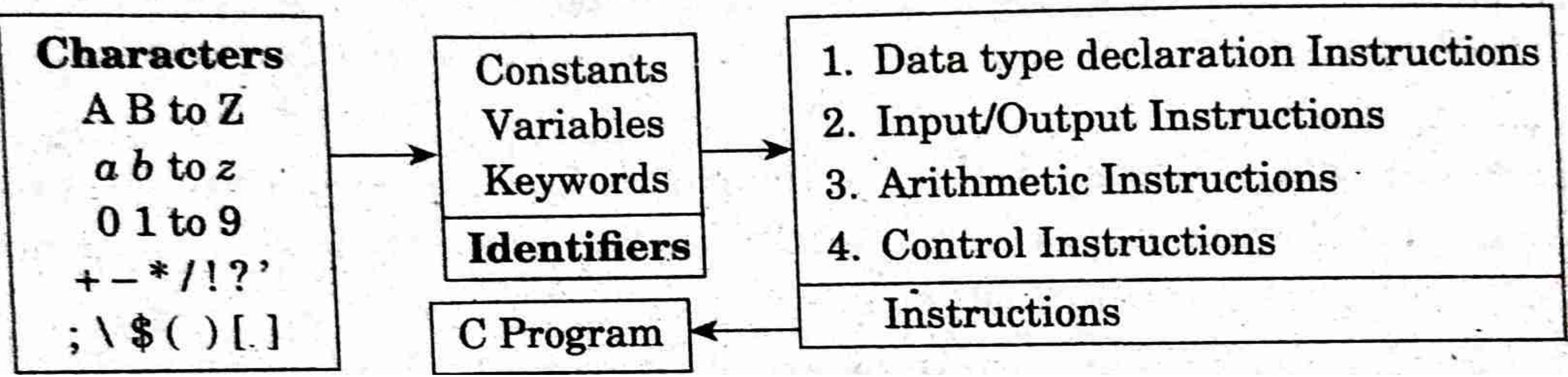
Numeric Characters : 0 to 9

[Total : 10]

Few special Characters : ~ ’ , “ ! ; : . ?
~ & $ + – / * % ( ) { } [ ]
- ^ # \ < > | @ etc.

Other Special Characters : tab, new line, blank space, vertical tab etc.

[Total : 255 – 62 = 193]


Characters
A B to Z
a b to *z*
0 1 to 9
+ – * / ! ? '
; \ $ ( ) [ ]
Constants
Variables
Keywords
Identifiers
1. Data type declaration Instructions
2. Input/Output Instructions
3. Arithmetic Instructions
4. Control Instructions
Instructions
C Program

9.7 Constants, Variables and Keywords in C

Alphabets, numbers and special symbol [like _ (underscore)] are combined to form constants, variables and keywords.

A **constant** as the name indicates is an entity that does not change. Constants can be of three types :

1. **Integer Constants** [a whole number]

   Examples : 73, –37, 0, 32767 etc.

2. **Real (floating point) Constants** [having a decimal point or in exponential form]

   Examples : +98.365, 0.0, –0.473, 1.24e7, –3.4e38, 9999.99, –1.2e–3 etc.

3. **Character Constants** [a single alphabet or a digit or a symbol written within a single quote]

   Examples : ‘J’, ‘j’, ‘=’, “7’ etc.

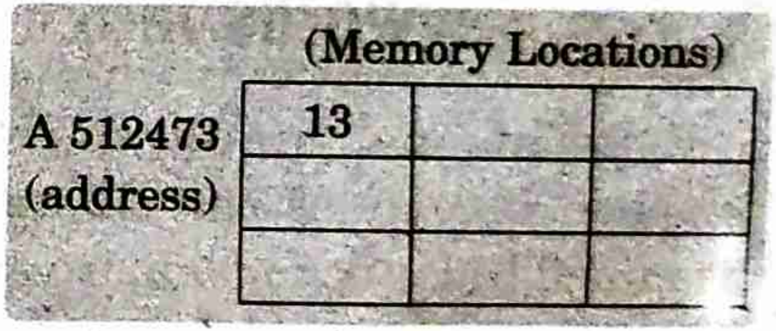


Fig. 9.5

A **variable** as the name indicates is an entity that changes.

During execution of a program we have to store the values of a variable, all intermediate and final results in

Notes

1. In a real constant of exponential form, the part before '*e*' is called **mantissa** (a signed or unsigned real constant of decimal form) and the part after '*e*' is called exponent (an signed or unsigned integer constant).
2. –1.2*e*–1.3 is an invalid real constat, as exponent part –1.3 is not an integer.
3. No commas or blanks are allowed within a real or an integer constant or a variable name.
4. In Turbo C or C++ the range of an integer constant is –32768 to +32767.
5. Range of real constants in exponential form is –3.4*e*38 to 3.4*e*38.
6. A variable name may be any combination of alphabets, numerics and underscores, but first character must be either an alphabet or an underscore.
7. No keywords (*e.g.*, char, int) should be used as variable names.
8. A variable type is nothing but the type of constants the memory cell can store. We use the keywords int, float and char respectively to mean integer type or real type or character types variables.
9. Variable may simply be defined as 'named memory location'.