UNIT 1 ALGORITHM AND FLOW CHART

1.0 INTRODUCTION

Intelligence is one of the key characteristics which differentiate a human being from other living creatures on the earth. Basic intelligence covers day to day problem solving and making strategies to handle different situations which keep arising in day to day life. One person goes Bank to withdraw money. After knowing the balance in his account, he/she decides to withdraw the entire amount from his account but he/she has to leave minimum balance in his account. Here deciding about how much amount he/she may withdraw from the account is one of the example of the basic intelligence. During the process of solving any problem, one tries to find the necessary steps to be taken in a sequence.

In this Unit you will develop your understanding about problem solving and aproaches.

1.1 OBJECTIVES

After going through this unit you should be :

- able to define problem;
- able to define algorithm;
- write algorithms for simple problems;
- explain properties of an algorithm;
- the meaning of flowchart;
- explain the need of flow chart;
- explain different symbols used in flow chart;
- draw flow chart for simple problems; and

- convert a flow chart into an algorithm and *vice-versa*.

1.2 PROBLEM SOLVING

Can you think of a day in your life which goes without problem solving? Answer to this question is of course, No. In our life we are bound to solve problems. In our day to day activity such as purchasing something from a general store and making payments, depositing fee in school, or withdrawing money from bank account. All these activities involve some kind of problem solving.

It can be said that whatever activity a human being or machine do for achieving a specified objective comes under problem solving. To make it more clear, let us see some other examples.

Example1: If you are watching a news channel on your TV and you want to change it to a sports channel, you need to do some thing i.e. move to that channel by pressing that channel number on your remote. This is a kind of problem solving.

Example 2: One Monday morning, a student is ready to go to school but yet he/she has not picked up those books and copies which are required as per time-table. So here picking up books and copies as per time-table is a kind of problem solving.

Example 3: If some one asks to you, what is time now? So seeing time in your watch and telling him is also a kind of problem solving.

Example 4: Some students in a class plan to go on picnic and decide to share the expenses among them. So calculating total expenses and the amount an individual have to give for picnic is also a kind of problem solving.

Now, broadly we can say that problem is a kind of barrier to achieve something and problem solving is a process to get that barrier removed by performing some sequence of activities.

Here it is necessary to mention that all the problems in the world can not be solved. There are some problems which have no solution and these problems are called ***Open Problems***. If you can solve a given problem then you can also write an algorithm for it. In next section we will learn what is an ***algorithm.***

1.3 ALGORITHM

Algorithm can be defined as: **"A sequence of activities to be processed for getting desired output from a given input."**

Webopedia defines an algorithm as**: "A formula or set of steps for solving a particular problem. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point".**

There may be more than one way to solve a problem, so there may be more than one algorithm for a problem.

Now, if we take definition of algorithm as: "**A sequence of activities to be processed for getting desired output from a given input.**" Then we can say that:

1. Getting specified output is essential after algorithm is executed.
2. One will get output only if algorithm stops after finite time.
3. Activities in an algorithm to be clearly defined in other words for it to be unambiguous.

Before writing an algorithm for a problem, one should find out what is/are the inputs to the algorithm and what is/are expected output after running the algorithm. Now let us take some exercises to develop an algorithm for some simple problems:

While writing algorithms we will use following symbol for different operations:

'+' for Addition
'-' for Subtraction
'*' for Multiplication
'/' for Division and
'←' for assignment. For example A←X*3 means A will have a value of X*3

1.3.1 Example of Algorithm

Problem 1: Find the area of a Circle of radius r.

Inputs to the algorithm:
Radius r of the Circle.

Expected output:
Area of the Circle

Algorithm:

```
Step1: Read\input the Radius r of the Circle

Step2: Area← PI*r*r  // calculation of area

Step3: Print Area
```

Problem 2: Ravi has to attend at least 70% of Practical Classes for C programming to be eligible to appear in the examination. There are total 50 Practical Classes for C programming. He has attended 20 out of 30 classes held so far. Find, at least how many more classes to be attended by Ravi to be eligible for appearing in Practical Examination.

Inputs to the algorithm:
i.Percentage of Attendance Required
ii.Total Number of Classes
iii. Number of Classes held so far
iv.Number of classes attended so far by Ravi

Expected output:
Number of classes to be attended to become eligible for appearing in examination

Algorithm:

```
Step1: Read Total Number of Classes C
```

```
Step2: Read Percentage Required P

Step3: Read Classes already attended Ca

Step4: Total Classes to be attended Ct← C*P/100

Step5: More Classes to be attended Cm←Ct-Ca

Step6: Print Cm
```

Problem 3: Convert temperature Fahrenheit to Celsius

Inputs to the algorithm:
Temperature in Fahrenheit

Expected output:
Temperature in Celsius

Algorithm:

```
Step 1: Read Temperature in Fahrenheit F

Step 2: C← 5/9*(F-32)

Step 3: Print Temperature in Celsius: C
```

Problem 4 : Ramshewak goes to market for buying some fruits and vegetables. He is having a currency of Rs 500 with him for marketing. From a shop he purchases 2.0 kg Apple priced Rs. 50.0 per kg, 1.5 kg Mango priced Rs.35.0 per kg, 2.5 kg Potato priced Rs.10.0 per kg, and 1.0 kg Tomato priced Rs.15 per kg. He gives the currency of Rs. 500 to the shopkeeper. Find out the amount shopkeeper will return to Ramshewak. and also tell the total item purchased.

Before we write algorithm for solving above problem let we find out what are the inputs to the algorithm **and** what is expected output.

Inputs to the algorithm are:

1. amount of different items purchase, for example 2.0 kg Apple etc.

2. price of the items, for example Mango is Rs. 35.0 per kg

3. total amount given to the shopkeeper

Expected output:
Amount to be returned by shopkeeper after deducting total price of the purchased vegetables and fruits, and total items purchased.

Algorithm:

```
Step1: Total ← 0, i ← 1;
```

```
Spet2: Read amount purchased and unit price of item$_i$

Step3: Total← Total + amount of item$_i$* price per unit of item$_i$

Step4:  i← i+1

Step5: Repeat Step 2 to 4 for each purchased item

Step6: Read Total amount given to the shopkeeper as GivenAmount

Step7: RefundAmount ← GivenAmount-Total

Step8: Print amount to be refund is Rs.: RefundAmount

Step9: Print total item purchased are: i
```

Problem 5: Find factorial of N.

Inputs to the algorithm are:
1. Number N

Expected output:
Factorial of N

Algorithm:

```
Step 1 : Result R←1

Step 2: I ←1

Step 3: Read the Number N

Step 4: Compare I with N

Step 5:  If I <=N  Then  R ← R*I ,Otherwise GOTO Step 8

Step 6:  I← I+1

Step 7: Repeat Step 4 and 6

Step 8: Print  R
```

Problem 6: Print the Table of N.

Inputs to the algorithm are:
Number N

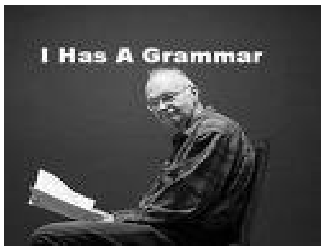
Expected output:
Table of N

Algorithm:

```
Step 1: I←1
Step 2: Read N
Step 3: If I <= 10 then Print I*N otherwise GOTO Step 6
Step 4: I ←I+1
Step 5: Repeat Step 3 and 4
Step 6: Stop
```

1.3.2 Properties of algorithm

Donald Ervin Knuth has given a list of five properties for an algor these properties are:



Donald Ervin Knuth
(Courtesy: http://svana.org)

1) **Finiteness**: An algorithm must always terminate after a finite number of steps. It means after every step one reach closer to solution of the problem and after a finite number of steps algorithm reaches to an end point.

2) **Definiteness**: Each step of an algorithm must be precisely defined. It is done by well thought actions to be performed at each step of the algorithm. Also the actions are defined unambiguously for each activity in the algorithm.

3) **Input**: Any operation you perform need some beginning value/quantities associated with different activities in the operation. So the value/quantities are given to the algorithm before it begins.

4) **Output**: One always expects output/result (expected value/quantities) in terms of output from an algorithm. The result may be obtained at different stages of the algorithm. If some result is from the intermediate stage of the operation then it is known as intermediate result and result obtained at the end of algorithm is known as end result. The output is expected value/quantities always have a specified relation to the inputs.

5) **Effectiveness**: Algorithms to be developed/written using basic operations. Actually operations should be basic, so that even they can in principle be done exactly and in a finite amount of time by a person, by using paper and pencil only.

Any algorithm should have all these five properties otherwise it will not fulfill the basic objective of solving a problem in finite time. As you have seen in previous examples, every step of an algorithm puts you closer to the solution.

While writing an algorithm for a problem if you are writing some operations such as a loop for performing some operation(s) repeatedly, then your loop must stop after finite number of iteration\repetition. While solving complex activity, for example for getting the total sum of all the items purchased from a general store, the better approach is to calculate the price of all the items separately and then add them to get the final sum. By doing this, you assure the finiteness and effectiveness of the operations in your algorithm.

☞ Check Your Progress 1

1) What is an algorithm?

……………………………………………..……………………….

…………………………………………………………………………………………

……………………………………………………………………..

………………………………..

2) Explain need of an algorithm.

……………………………………………..……………………….

…………………………………………………………………………………………

………………………………………………………….

……………………………………..…………..

3) Can there be more than one algorithm for a single problem? Give reason for your answer.

……………………………………………..……………………….

…………………………………………………………………………………………

……………………………………………………….... ……………..

………………………………..

4) Write an algorithm to find average age of a group of 10 players.

……………………………………………..……………………….

…………………………………………………………………………………………

……………………………………………………………………..

………………………………..

5) State True/False for the following statements:

i) An algorithm should take finite amount of time to compete.

ii) Ambiguous steps are allowed in an algorithm.

iii) Complex problems can not have an algorithm.

iv) For any problem algorithm can be developed.

v) A correct algorithm always gives expected output.

1.4 FLOWCHART

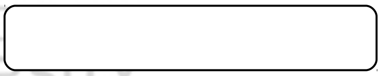
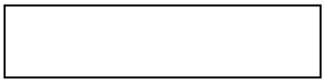
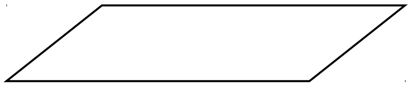
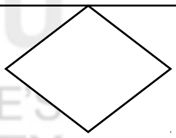
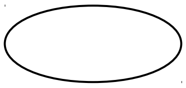
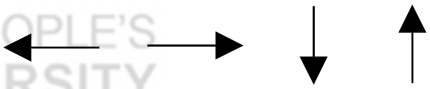
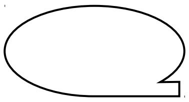
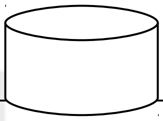
The flowchart is a diagram which visually presents the flow of data through processing systems. This means by seeing a flow chart one can know the operations performed and the sequence of these operations in a system. Algorithms are nothing but sequence of steps for solving problems. So a flow chart can be used for representing an algorithm. A flowchart, will describe the operations (and in what sequence) are required to solve a given problem.

You can see a flow chart as a blueprint of a design you have made for solving a problem. For example suppose you are going for a picnic with your friends then you plan for the activities you will do there. If you have a plan of activities then you know clearly when you will do what activity. Similarly when you have a problem to solve using computer or in other word you need to write a computer program for a problem then it will be good to draw a flowchart prior to writing a computer program. Flowchart is drawn according to defined rules.

1.4.1 Flowchart Symbols

For drawing flow chart standard symbols are used. These symbols are given in Table 1.

Table1: Flow chart Symbols

Symbols	**Meaning/Used for**
	Start or end of the programme
	Used for writing steps of operations/action or processing function of a programme
	Input or output operation
	Decision making and branching operations
	Connector or joining of two parts in a flow chart
	Flow line used for showing flow of data
	Magnetic Tape used for secondary storage/Backup
	Magnetic Disk used for secondary storage/Backup

1.4.2 Meaning of Flowchart

A flowchart is a diagrammatic representation of algorithm. It clearly illustrates the sequence of operations to be performed for getting the solution of a problem. For simple problems flow charts may not be very useful but for complex and large problems flow chars are very helpful in understanding the logic of the problem. Flowcharts are used as a link of communication between programmers and clients for whom the program to be developed. If you are having a flowchart for your program then you can use it in explaining the program to others. Once the flowchart is drawn, it becomes easy to write the computer program. Flowcharts can be used for preparing a better documentation of a complex problem.

1.4.3 Guidelines for drawing A Flowchart

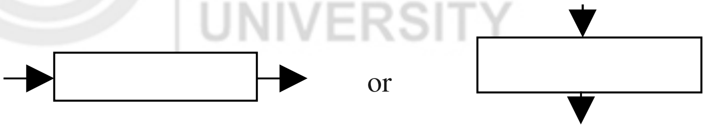
Flow charts are drawn using slandered flowchart symbols. While drawing flowchart some guideline to be followed. Below are some guidelines for drawing flowchart:

1) First of all list all necessary requirements in a logical order.

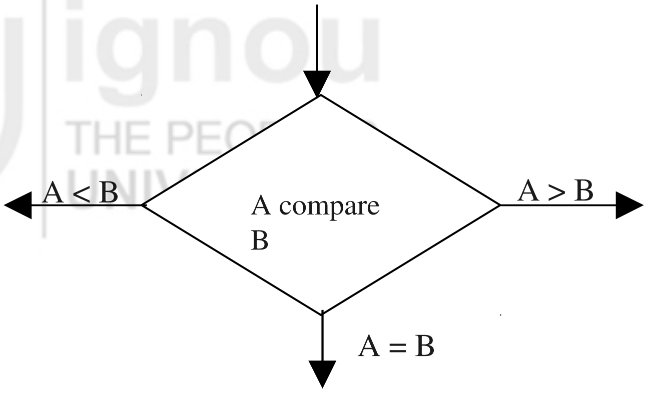
2) The flowchart should be clear and easy to understand. There should not be any ambiguity in understanding the flowchart. For doing this it is necessary to have all the steps and operation very simple.

3) Usually direction of the flow of data /procedure in the system should be from left to right or top to bottom.

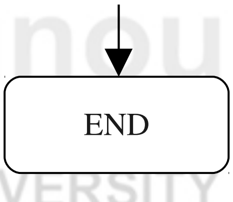
4) Only one flow line should come out from a process symbol.



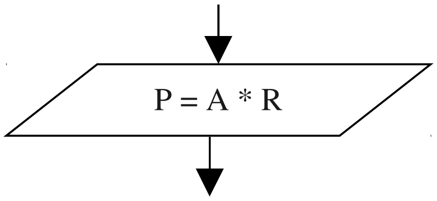
5) In the case of use of decision symbol,only one flow line should enter to it(decision symbol), but there may be two or three flow lines coming out of the decision symbol, one for each possible answer.



6) In a flowchart only one flow line should come to the end symbol.



7) While writing steps in side the processing symbol, steps should be brief and if necessary, you can use the annotation symbol to describe data or processing l steps more clearly.



Where P = Price, A = Amount, and R = Rate

8) In the case of complex flowchart connector symbols to be used for reducing the number of flow lines in the flowchart.

9) Intersection of flow lines should be avoided to make a flowchart more effective and for better way of communication.

10) A flowchart must have a logical start and end.

11) Once a flowchart is drawn its validity should be tested by passing through it with a simple set of test data.

1.4.4 Some examples of Flowcharts

Now, we will discuss some examples on flowcharting. These examples will help in proper understanding of flowcharting technique. This will help you in program development process in next unit of this block.

For example following is a Flowchart fo[illegible].

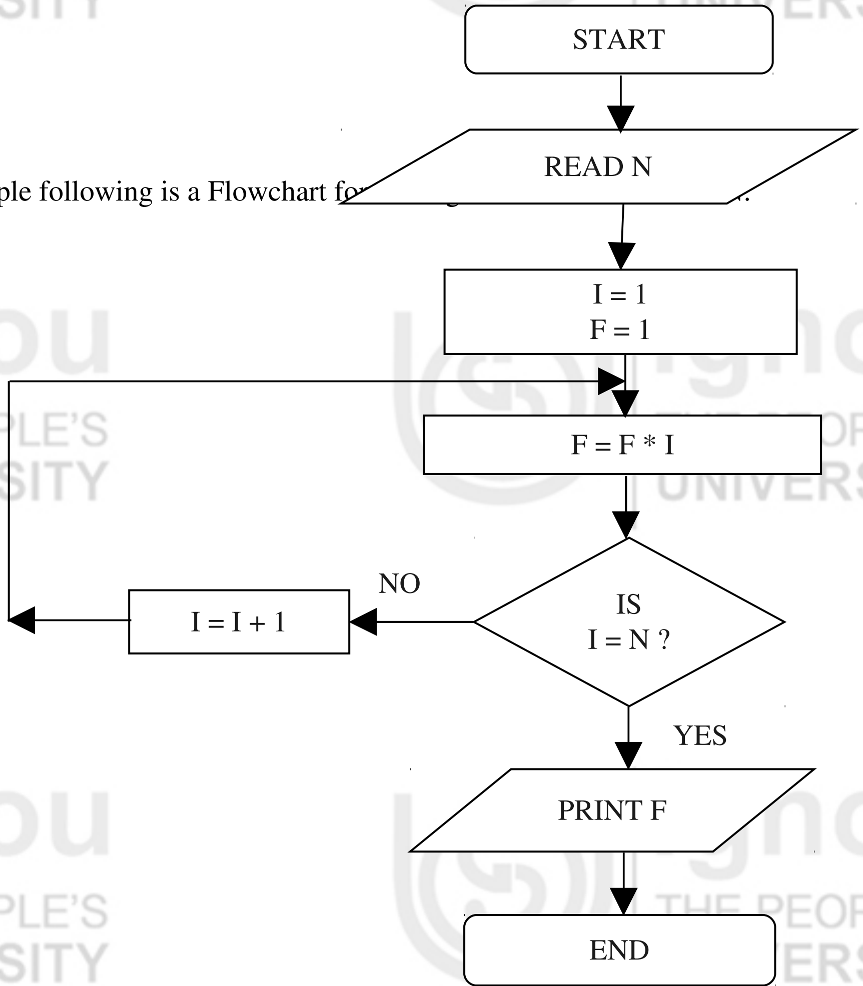


Figure 1: Flowchart for factorial N

Some more examples of flowchart:

Problem 1: Find the area of a circle of radius r.

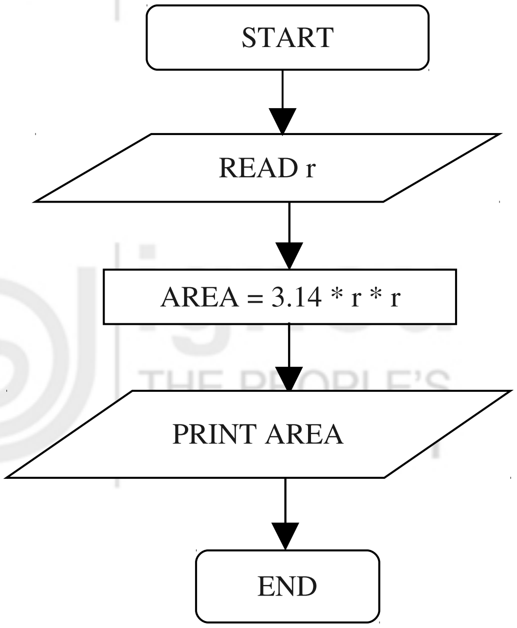


Figure 2: Flowchart rcle.

Problem 2: Ravi has to attend at least 70% of Practical Classes for C Programming to be eligible to appear in the exa al 50 Practical Classes for C Programming. He has already a es held so far. Find at least how many classes to be attended by earing in Practical Examination.

- * Total number of classes **C**
- * Percentage Required **P**
- * Number of Classes Already A
- * Number of Classes to be Atte
- * More classes to be attended $\mathbf{C_m}$

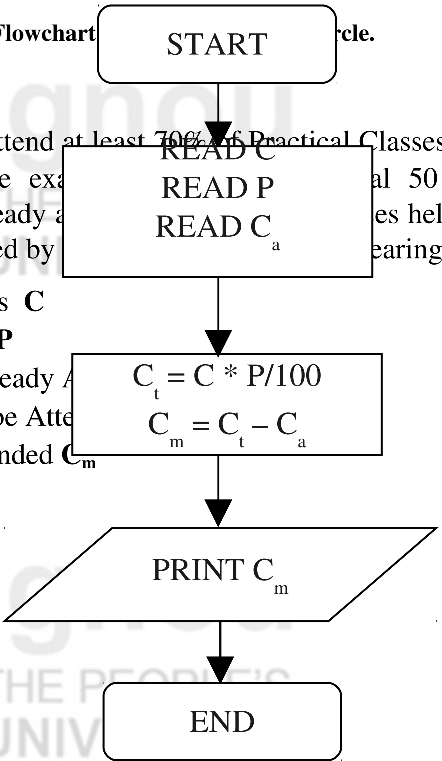


Figure 3: Flowchart for checking eligibility to appear in practical exam.

Problem 3: Convert temperature Fahrenheit to Celsius.

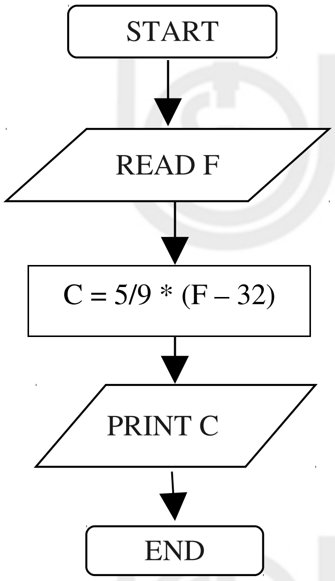


Figure 4: Flowchart for convert I to C.

Problem 4: Ramesh goes to market for buying some fruits and vegetables. He is having a currency of Rs 500 with him for marketing. From a shop he purchase 2.0 kg Apple priced Rs. 50.0 per kg, 1.5 kg Mango priced Rs. 35.0 per kg, 2.5 kg Potato priced Rs. 10.0 per kg, and 1.0 kg Tomato priced Rs. 15 per kg. He give the currency of Rs. 500 to the shopkeeper. Find the amount shopkeeper will return to Ramesh and also tell the total item purchased. Before we make flowchart for solving above problem let us find out what are the inputs and what is the expected output.

* Apple Price $\mathbf{A_P}$
* Mango Price $\mathbf{M_P}$

* Apple Amount $\mathbf{A_W}$
* Mango Amount $\mathbf{M_W}$

- * Potato Price $\mathbf{P_P}$
- * Tomato Price $\mathbf{T_P}$
- * Amount Given $\mathbf{A_M}$
- * Potato Amount $\mathbf{P_W}$
- * Tomato Amount $\mathbf{T_W}$
- * Return Amount $\mathbf{R_{AM}}$

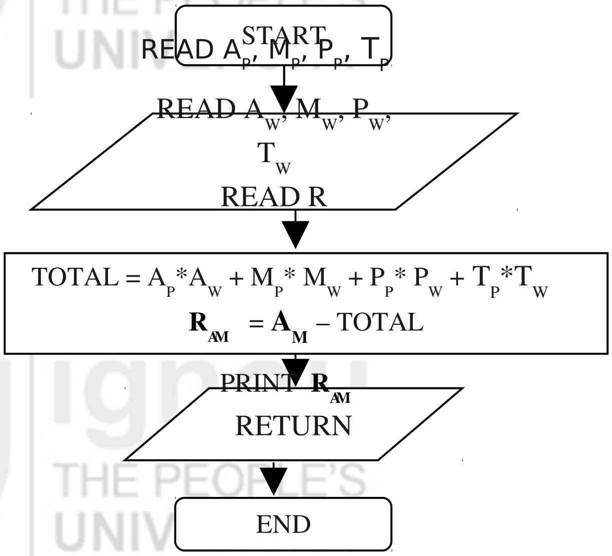

START
READ A_P, M_P, P_P, T_P
READ A_W, M_W, P_W, T_W
READ R
TOTAL = A_P*A_W + M_P* M_W + P_P* P_W + T_P*T_W
$\mathbf{R_{AM}}$ = $\mathbf{A_M}$ – TOTAL
PRINT $\mathbf{R_{AM}}$
RETURN
END

Figure 5: Flowchart for cost calculation and return amount

Problem 6: Print the Table of N

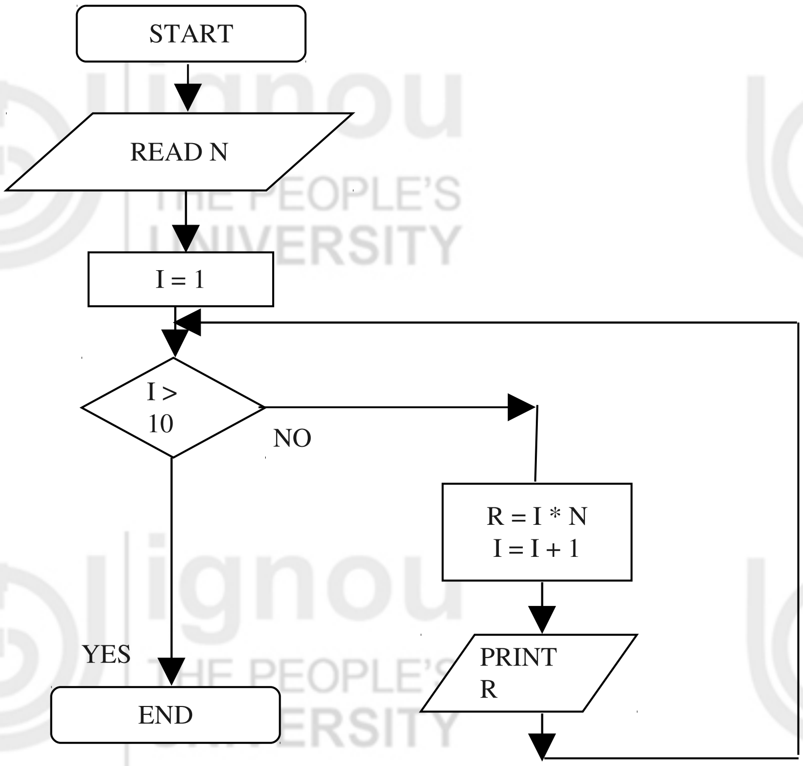


Figure 6: Flowchart for printing table of N

1.4.5 Advantages of using Flowcharts

As we discussed flow chart is used for representing algorithm in pictorial form. This pictorial representation of a solution/system is having many advantagrs.These advantages are as follows:

1) **Communication**: A Flowchart can be used as a better way of communication of the logic of a system and steps involve in the solution, to all concerned particularly to the client of system.

2) **Effective analysis**: A flowchart of a problem can be used for effective analysis of the problem.

3) **Documentation of Program/System**: Program flowcharts are a vital part of a good program documentation. Program document is used for various purposes like knowing the components in the program, complexity of the program etc.

4) **Efficient Program Maintenance**: Once a program is developed and becomes operational it needs time to time maintenance. With help of flowchart maintenance become easier.

5) **Coding of the Program**: Any design of solution of a problem is finally converted into computer program. Writing code referring the flowchart of the solution become easy.

1.4.6 Limitations of using Flowcharts

1) Complexity of Logic: If program logic is complex then flowchart of the program becomes complicated.

2) Alterations and Modifications in Logic: any alterations in the program logic may require re-drawing of flowchart completely.

3) Reuse is Not Possible: As the flowchart symbols cannot be typed, always reproduction of flowchart symbols are required.

☞ Check Your Progress 2

1) Fill in the blanks:

i) A program flowchart indicates the_________ to be performed and the __________ in which they occur.

ii) A program flowchart is generally read from _____________ to ______________

iii) Flowcharting symbols are connected together by means of_________________

iv) Connectors are used to join ______________portions of a flowchart.

v) The __________is one of the best ways of representing a program.

2) Explain steps involve in drawing of a flowchart.

………………………………………..………………………..

…………………………………………………………………………………………

……………………………………………………….... ……………..

………………………………..

3) Explain uses of Flowchart.

…………………………………………..………………………..

………………………………………………………………………………………

……………………………………………………….. ……………..

………………………………..

4) Draw a flowchart to find the sum of first 100 natural numbers.

…………………………………………..………………………..

………………………………………………………………………………………

……………………………………………………….. ……………..

………………………………..

5) Draw a flowchart to find the largest of three numbers x, y and z.

…………………………………………..………………………..

………………………………………………………………………………………

……………………………………………………….. ……………..

………………………………..

1.5 SUMMARY

In this unit you have learned about an algorithm. How an algorithm to be written. This unit explains the properties of an algorithm such as finiteness, definiteness,and effectiveness. Further this unit covers about the flowcharting. Flowcharting is a way of representing algorithm in the form of a diagram. You have learned about different symbols used in drawing flowcharts. This unit also explains about guidelines for drawing flowcharts. Finally in this unit we have discussed about advantages of flowcharts and limitations of flowcharts.

1.6 SOLUTIONS/ANSWERS

Check Your Progress 1

1) An algorithm is **“A sequence of activities to be processed for getting desired output from a given input.”**These activities may be in the form of some formula, some operation or some input-output activity.

2) Algorithms are developed for solving problems. A problem is a kind of barrier to achieve something and problem solving is a process to get that barrier removed by performing some sequence of activities. Randomly writing any sequence of activity will never solve any problem and selection of activities for solving a problem need to have certain basic norms and those norms are considered as characteristics of an algorithm. So the algorithms are required for solving problems.

3) Yes, there may be more than one algorithm for same problem. As an algorithm states the steps involve in getting the solution of a problem. For example if one wants to go from point X to point Y in a city and there are many alternative paths (ways) which are available for going to point Y from point X. One may choose any of the alternative paths available. Similarly for solving a problem there may be more than

one alternative ways available and each way of solution can be represented in the form of algorithm. So a single problem may have more than one algorithm.

4) **Algorithm:**

```
Step 1 : Sum ←0
Step 2: I ←1
Step 3: Read the age of Player-I as Age(I)
Step 4: Sum ← Sum+Age(I)
Step 5:  I← I+1
Step 6:  If I <=10 Then GOTO Step 3
Step 7: Print Average age is: Sum/10
```

5) i) True
   ii) False
   iii) False
   iv) False
   v) True

Check Your Progress 2

1) i) Steps, order
   ii) Top –bottom, left-right
   iii) Connector
   iv) Two
   v) Flowchart

2) Start the flow chart by drawing the elongated circle shape, and labeling it "START".

   ii) Move to the first action or question, and draw a rectangle or diamond as per need. Write the action or question and then draw an arrow from the start symbol to this shape.

   iii) Put all the actions in your whole process, appropriately in the order they occur, and linking these together using arrows to show the flow of the process.

   iv) For making decision, draw arrows leaving the decision diamond for each possible outcome from the decision, and label them with the outcome.

   v) Finally to show the end of the process using an elongated circle labeled "END".

   vi) After drawing is complete, change your flow chart step to step by asking yourself if you have correctly represented the sequence of activities and decisions involved in the process and if needed make the necessary corrections in the drawing.

3) Flow charts are easy-to-understand diagrams. Flowcharts show how steps in a process are working together. This feature makes flowchart useful tools for communicating

how processes work. Also flowcharts help in clearly documenting how a particular job is done. A flowchart is useful in:

- Defining and analyzing processes of a system;
- For presenting a step-by-step picture of the process for analysis and communication to the clients and users of the system.

4) The required flowchart is given in Figure given below:

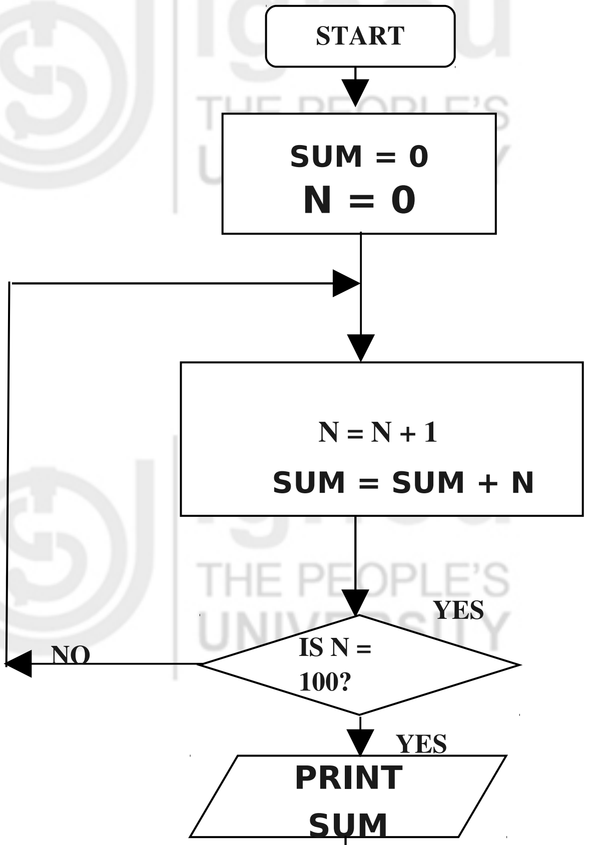


Figure 7: Flo 0 natural numbers

5) A flowchart to find the largest of three numbers x, y and z.

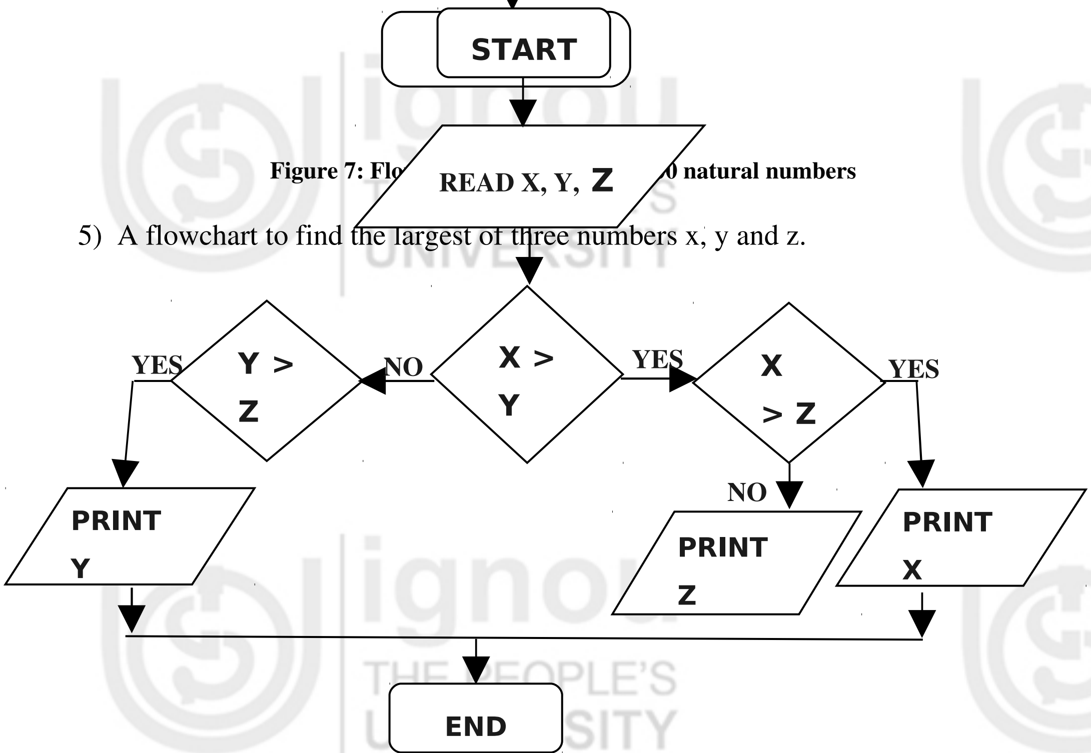


Figure 8: Flowchart for finding out the largest of three numbers

1.7 FURTHER READINGS AND REFERENCES

1) Block 1(Introduction to Algorithms) of MCS-031- Analysis and Design of Algorithm, IGNOU MCA Course, Year 2005.

2) Programming in C by R Subburaj, Vikas Publishing House, Delhi, Year 2000.

3) Complete Knowledge in C by Sukhendu Dey and Debobrataa Dutta, Narosa Publishing House, Delhi, Year 2009.

Reference Web Links:

i) http://www.webopedia.com/TERM/a/algorithm.html

ii) http://www.scriptol.org/computer-algorithm.html

iii) http://www.edrawsoft.com/flowchart-symbols.php

iv) http://www.softpanorama.org/Algorithms/flowcharts.shtml