INTRODUCTION TO FLOWCHARTING

1.0 OBJECTIVES

Friends,After studying this topic you will be able to -

- describe problem solving
- describe the meaning of flowcharts and flowcharts as a tool to represent program logic sequence.
- explain types of flowcharts and flowchart symbols.
- state uses of flowcharts and advantages of flowcharts
- describe develop flowcharts for problem solving.
- describe the advanced flowcharting techniques involved in flowcharts for computations, decision making, loops, predefined process etc.

1.1 INTRODUCTION

Computers are capable of handling various complex problems which are tedious and routine in nature. In order that a computer solve a problem, a method for the solution and a detailed procedure has to be prepared by the programmer. The problem solving Involves :

- Detailed study of the problem
- Problem redefinition
- Identification of input data, output requirements and conditions and limitations
- Alternative methods of solution
- Selection of the most suitable method
- Preparation of a list of procedures and steps to obtain the solution
- Generating the output

The preparation of lists of procedures and steps to obtain the result introduces the algorithmic approach to problem solving. The algorithm is a sequence of instructions designed in such a way that if the instructions are executed in a specific sequence the desired results will be obtained. The instructions should be precise and concise and the result should be obtained after a finite execution of steps. This means that the algorithm should not repeat one or more instructions infinitely. It should terminate at some point and result in the desired output.

An algorithm should possess the following characteristics :

- Each and every instruction should be precise and clear
- Each instruction should be performed a finite number of times
- The algorithm should ultimately terminate
- When the algorithm terminates the desired result should be obtained.

1.2 FLOWCHARTS

Before you start coding a program it is necessary to plan the step by step solution to the task your program will carry out. Such a plan can be symbolically developed using a diagram. This diagram is then called a flowchart. Hence a flowchart is a symbolic representation of a solution to a given task. A flowchart can be developed for practically any job. Flowcharting is a tool that can help us to develop and represent graphically program logic sequence. It also enables us to trace and detect any logical or other errors before the programs are written.

1.3 TYPES OF FLOWCHARTS

Computer professionals use two types of flowcharts viz :

- Program Flowcharts.
- System Flowcharts

1.3.1 Program Flowcharts :

These are used by programmers. A program flowchart shows the program structure, logic flow and operations performed. It also forms an important part of the documentation of the system. It broadly includes the following:

- Program Structure.
- Program Logic.
- Data Inputs at various stages.
- Data Processing
- Computations and Calculations.
- Conditions on which decisions are based.
- Branching & Looping Sequences.
- Results.
- Various Outputs.

The emphasis in a program flowchart is on the logic.

1.3.2 System Flowcharts :

System flowcharts are used by system analyst to show various processes, sub systems, outputs and operations on data in a system.

In this course material we will be discussing program flowcharts only.

1.2 & 1.3 Check Your Progress.

Answer in 1-2 sentences :

a) What is an algorithm?

.....

.....

b) What is a flowchart?

.....

.....

c) What are the types of flowcharts?

.....

.....

d) List any two steps involved in problem solving.

.....

.....

1.4 FLOWCHART SYMBOLS

Normally, an algorithm is expressed as a flowchart and then the flowchart is converted into a program with the programming language. Flowcharts are independent of the programming language being used. Hence one can fully concentrate on the logic of the problem solving at this stage. A large number of programmers use flowcharts to assist them in the development of computer programs. Once the flowchart is fully ready, the programmer then write it in the programming language. At this stage he need not concentrate on the logic but can give more attention to coding each instruction in the box of the flowchart in terms of the statements of the programming language selected.

A flowchart can thus be described as the picture of the logic to be included in the computer program. It is always recommended for a beginner, to draw flowcharts prior to writing programs in the selected language. Flowcharts are very helpful during the testing of the program as well as incorporating further modifications.

Flowcharting has many standard symbols. Flowcharts use boxes of different shapes to denote different types of instructions. The actual instruction is written in the box. These boxes are connected with solid lines which have arrowheads to indicate the direction of flow of the flowchart. The boxes which are used in flowcharts are standardised to have specific meanings. These flowchart symbols have been standardised by the American National Standards Institute. (ANSI).

While using the flowchart symbols following points have to be kept in mind:

- The shape of the symbol is important and must not be changed.
- The size can be changed as required.
- The symbol must be immediately recognizable.
- The details inside the symbol must be clearly legible.
- The flow lines, as far as possible, must not cross.

Terminal Symbol:

Every flowchart has a unique starting point and an ending point. The flowchart begins at the start terminator and ends at the stop terminator. The Starting Point is indicated with the word START inside the terminator symbol. The Ending Point is indicated with the word STOP inside the terminator symbol. *There can be only one*

START and one STOP terminator in you entire flowchart. In case a program logic involves a pause, it is also indicated with the terminal symbol.

Input/Output Symbol :

This symbol is used to denote any input/output function in the program. Thus if there is any input to the program via an input device, like a keyboard, tape, card reader etc. it will be indicated in the flowchart with the help of the Input/Output symbol. Similarly, all output instructions, for output to devices like printers, plotters, magnetic tapes, disk, monitors etc. are indicated in the Input/Output symbol.

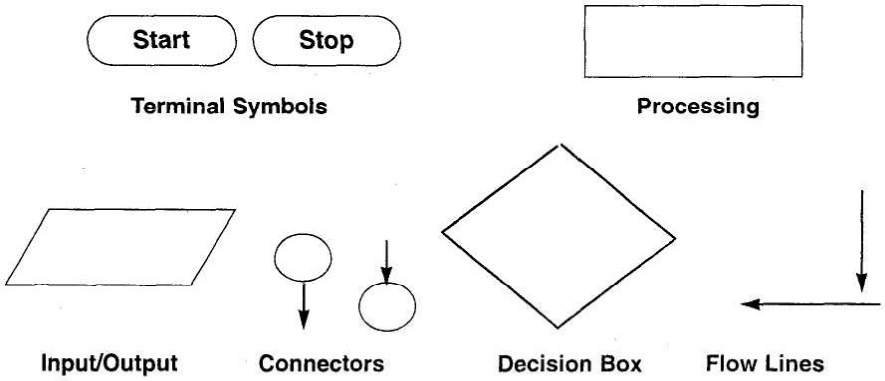


Fig. 1 : Flowchart Symbols

Process Symbol :

A process symbol is used to represent arithmetic and data movement instructions in the flowchart. All arithmetic processes of addition, subtraction, multiplication and division are indicated in the process symbol. The logical process of data movement form one memory location to another is also represented in the process box. If there are more than one process instructions to be executed sequentially, they can be placed in the same process box, one below the other in the sequence in which they are to be executed.

Decision Symbol :

The decision symbol is used in a flowchart to indicate the point where a decision is to be made and branching done upon the result of the decision to one or more alternative paths. The criteria for decision making is written in the decision box. All the possible paths should be accounted for. During execution, the appropriate path will be followed depending upon the result of the decision.

Flowlines :

Flowlines are solid lines with arrowheads which indicate the flow of operation. They show the exact sequence in which the instructions are to be executed. The normal flow of the flowchart is depicted from top to bottom and left to right.

Connectors :

In situations, where the flowcharts becomes big, it may so happen that the flowlines start crossing each other at many places causing confusion. This will also result in making the flowchart difficult to understand. Also, the flowchart may not fit in a single page for big programs. Thus whenever the flowchart becomes complex and spreads over a number of pages connectors are used. The connector represents entry from or exit to another part of the flowchart. A connector symbol is indicated by a circle and a letter or a digit is placed in the circle. This letter or digit indicates a link. A pair of such identically labelled connectors are used to indicate a continued flow in situations where flowcharts are complex or spread over more than one page. Thus a connector

indicates an exit from some section in the flowchart and an entry into another section of the flowchart. If an arrow enters a flowchart but does not leave it, it means that it is an exit point in the flowchart and program control is transferred to an identically labelled connector which has an outlet. This connector will be connected to the further program flow from the point where it has exited. Connectors do not represent any operation in the flowchart. Their use is only for the purpose of increased convenience and clarity.

1.4 Check Your Progress.

1. Answer in 1- 2 sentences

a) What is the use of decision box in flowcharts?

.....

.....

b) What do flowlines show?

.....

.....

2. Match the following :

Column A	Column B
a) Connector	(i) show the sequence of instructions execution
b) Input/Output	(ii) represent arithmetic and data movement instructions
c) Process	(iii) represents entry or exit to another part of flowchart.
d) Flow Lines	(iv) denote any input/output function

1.5 ADVANTAGES OF FLOWCHARTS

There are a number of advantages when using flowcharts in problem solving. They provide a very powerful tool to programmers to first represent their program logic graphically and independent of the programming language.

- Developing the program logic and sequence. A macro flowchart can first be designed to depict the main line of logic of the software. This model can then be broken down into smaller detailed parts for further study and analysis.

- A flowchart being a pictorial representation of a program, makes it easier forthe programmer to explain the logic of the program to others rather than a program

- It shows the execution of logical steps without the syntax and languagecomplexities of a program.

- In real life programming situations a number of programmers are associatedwith the development of a system and each programmer is assigned a specific task of the entire system. Hence, each programmer can develop his own flowchart and later on all the flowcharts can be combined for depicting the overall system. Any problems related to linking of different modules can be detected at this stage itself and suitable modifications carried out. Flowcharts can thus be used as working models in design of new software systems.

- Flowcharts provide a strong documentation in the overall documentation of thesoftware system.

- Once the flowchart is complete, it becomes very easy for programmers to writethe program from the starting point to the ending point. Since the flowchart is a detailed representation of the program logic no step is missed during the actual program writing resulting in error free programs. Such programs can also be developed faster.

- A flowchart is very helpful in the process of debugging a program. The bugs canbe detected and corrected with the help of a flowchart in a systematic manner.- A flowchart proves

to be a very effective tool for testing. Different sets of data are fed as input to program for the purpose

1.5 Check Your Progress.

1. **Give any two advantages of flowcharts.**

   .....

   .....

1.6 DEVELOPING FLOWCHARTS

In developing the flowcharts following points have to be considered:

- Defining the problem.
- Identify the various steps required to form a solution.
- Determine the required input and output parameters.
- Get expected input data values and output result.
- Determine the various computations and decisions involved.

With this background of flowcharts and flowchart symbols let us now draw some sample flowcharts. First we shall write the steps to prepare the flowchart for a particular task and then draw the flowchart.

Example : To prepare a flowchart to add two numbers. (Fig. 2a.) The steps are :

1. Start.
2. Get two numbers N1 and N2.
3. Add them.
4. Print the result.
5. Stop.

Example : To prepare a flowchart to determine the greatest of two numbers. Here we use the decision symbol. We also combine the two reads for numbers A and B in one box.

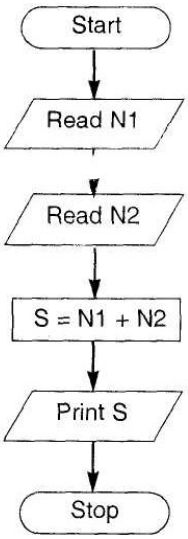


Fig. 2a) Flowchart to add two numbers N1 and N2

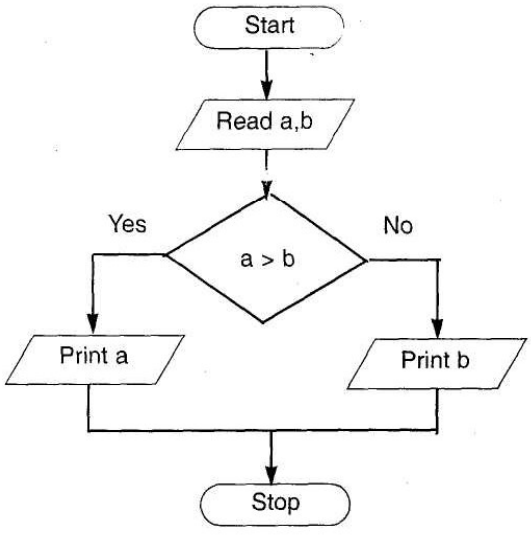


Fig. 2b) Flowchart to determine the greater of two numbers a and b

The steps are :

1. Start
2. Get two number A and B.
3. If A > B then print A else print B.
4. Stop.

Note that in the first example, we have used two separate input/output boxes to read the numbers N1 and N2. In the second example, both the numbers a and b are read in the same box. Thus if more than one instructions of the same kind follow one another then they can be combined in the same box.

1.6 Check Your Progress.

1. Write the steps and draw the flowcharts for the following :

a) Find the average of three numbers a, b and c.

b) Find the area of a rectangle whose length and breadth are given.

2. Answer the following :

a) What are the points to be considered in developing flowcharts?

.....

.....

1.7 TECHNIQUES

In this section we shall cover the various flowcharting techniques viz.

- flowcharts for computations
- flowcharts for decision making
- flowcharts for loops
- Predefined Process

1.7.1 Flowcharts for Computations :

Computers are used to perform many calculations at high speed. When you develop a program it also involves several calculations.

The general format of the flowcharting steps for computations is :

- Create memvars used in calculations and read operation.
- Get required data input using memvars.
- Perform the necessary calculations.
- Print the result.

Programming considerations while using computation techniques : Most languages have provision for creating memvars. The exact syntax depends on the language used. In most cases (but not all) your programs have to create and initialize the memvars before you can use them.

The following examples show the usage of flowcharts in computations. The flowcharts are shown in Fig.3a and Fig. 3b.

Example : Flowchart for a program that converts temperature in degrees Celsius to degrees Fahrenheit.

First let us write the steps involved in this computation technique.

1. Start.
2. Create memvars F and C (for temperature in Fahrenheit and Celsius).

2. Read degrees Celsius into C.
3. Compute the degrees Fahrenheit into F.
4. Print result (F).
5. Stop.

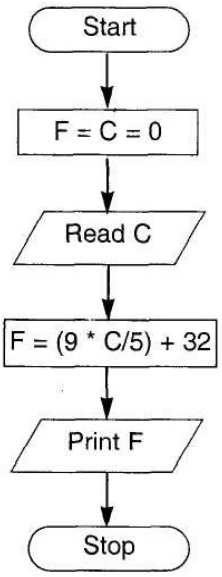


Fig. 3a) Flowchart to convert temperature from Celsius to Fahrenheit

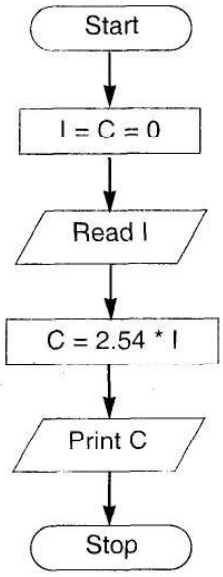


Fig. 3b) Flowchart to convert inches to centimetres

Example : Flowchart for a program that converts inches to centimeters First let us write the steps involved in this computation technique.

1. Start.
2. Create memvars C and I (for Centimeters and Inches respectively).

2. Read value of Inches into I 3.

Compute the Centimeters into C.

4. Print result (C).
5. Stop.

1.7.2 Flowcharts for decision making :

Computers are used extensively for performing various types of analysis. The decision symbol is used in flowcharts to indicate it.

The general format of steps for flowcharting is as follows:

- Perform the test of the condition.
- If condition evaluates true branch to Yes steps.
- If condition evaluates false branch to No steps.

Programming Considerations :

Most programming languages have commands for performing test and branching. The exact commands and syntax depends on the language used. Some of

the conditional constructs available in programming languages for implementing decision making in programs are as follows:

- If
- If - else - endif
- If - elseif - endif
- Do case - endcase.
- Switch.

All languages do not support all of the above constructs.
The operators available for implementing the decision test are as follows:

- Relational Operators (which determine equality or inequality)
- Logical Operators, (useful for combining expressions)

The branching to another set of commands can be implemented by using functions, procedures etc.

Example: Flowchart to get marks for 3 subjects and declare the result. If the marks >= 35 in all the subjects the student passes else fails.

The steps involved in this process are :

1. Start.
2. Create memvars m1, m2, m3.
3. Read marks of three subjects m1, m2, m3.
4. If m1 >= 35 goto step 5 else goto step 7
5. If m2 >= 35 goto step 6 else goto step 7
6. If m3 >= 35 print Pass. Goto step 8
7. Print fail
8. Stop

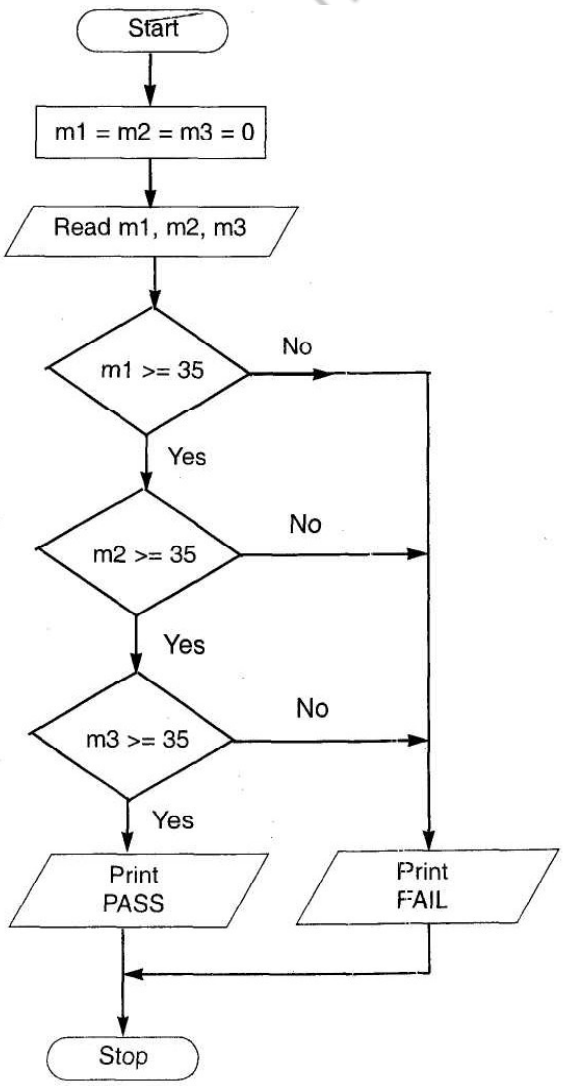


Fig. 4 Flowchart to determine whether pass or fail making using of decision box

The flowchart is shown in Fig. 4.

An alternative method is the one in which you can combine all the conditions

with the AND operator. The steps then would be :

1. Start
2. Create memvars m1, m2, m3.
3. Read marks of three subjects into m1, m2 and m3.
4. If m1 >= 35 and m2 >= 35 and m3 >= 35 print Pass. Otherwise goto step 5.
5. Print Fail.
6. Stop

Developing this flowchart is left as an exercise to the student.

1.7.3 Flowcharts for loops

Looping refers to the repeated use of one or more steps. i.e. the statement or block of statements within the loop are executed repeatedly. There are two types of loops. One is known as the fixed loop where the operations are repeated a fixed number of times. In this case, the values of the variables within the loop have no effect on the number of times the loop is to be executed. In the other type which is known as the

variable loop, the operations are repeated until a specific condition is met. Here, the number of times the loop is repeated can vary.

The loop process in general includes :

- Setting and initialising a counter
- execution of operations
- testing the completion of operations
- incrementing the counter

The test could either be to determine whether the loop has executed the specified number of times, or whether a specified condition has been met.

Programming considerations :

Most of the programming languages have a number of loop constructs for efficiently handling repetitive statements in a program. These include :

- do-while loop
- while loop
- for loop
- for-next loop

In most of the looping situations, we make use of counters. In situations where the loop is to be repeated on the basis of conditions, relational operators are used to check the conditions.

Example : To find the sum of first N numbers. This example illustrates the use of a loop for a specific number of times. Fig. 5a. The steps are :

1. Start
2. Create memvars S , N, I
3. Read N
4. Set S (sum) to 0
5. Set counter (I) to 1.
6. S = S + I
7. Increment I by 1.
8. Check if I is less than or equal to N. If no, go to step 6.
9. Print S 10. Stop

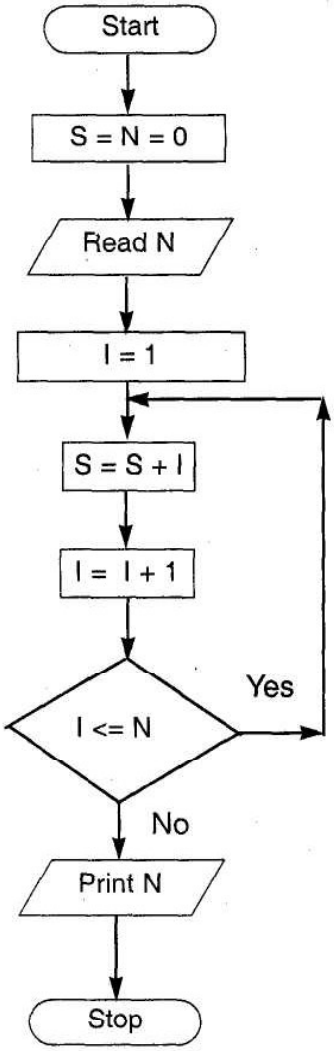


Fig.5a Flowchart to find sum of first N numbers

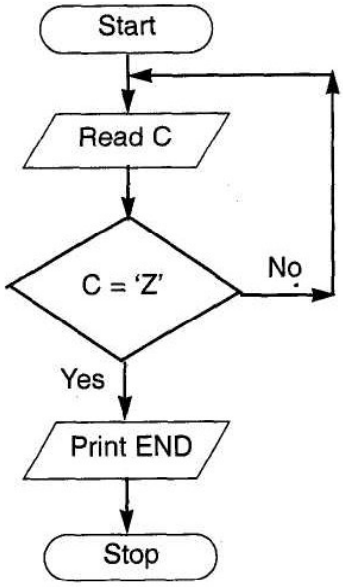


Fig.5b Flowchart to read a character till Z is input

The flowchart is shown in Figure 5a.

Example : To check whether character read from keyboard is Z. If it is Z then print END, else read another character. This example shows the test which executes till a particular condition is satisfied.

The steps are :

1. Start
2. Create memvar C
3. Read C,
4. Check if C = 'Z'. If no goto step 3.
5. Print END
6. Stop

The flowchart is shown if figure 5b.

1.7.4 Predefined Process

In a large application, we use programs written by others. Also when we invoke a library routine of a language, we are using predefined process. In a predefined process the required inputs are known and the expected output. We do not know how the routine handles the task. For us it is a like a black box. Predefined processes have great use as they enable us to use programs written by others and save a lot of time.

It also permits the integration of various parts of the software into a single unit. The predefined routine can be put at any place in the flowchart. It is a single symbol of flowchart that represents an entire flowchart created elsewhere.

Programming Considerations:

Today structured and modular programming is accepted as the best way to developed software applications. Each module treats the other module as a predefined process. The development of the library routines also envisions the use of predefined process. It prevents us from having to write separate programs again and again each time to do the same task, in different applications.

1.7 Check Your Progress.

1. **Write the programming considerations for the following** :

a) Computations :

.....

.....

b) Decision making :

.....

.....

2. Draw the flowcharts for the following :

a) Printing the first five odd numbers.

b) Read age of a person. If age less than 60 then print “Not a senior citizen” otherwise print “Senior Citizen”.

1.8 SUMMARY

In this chapter we learnt the concept of Algorithm & flowchart

" The algorithm is a sequence of Instructions designed in such a way that if the instructions are executed in a specific sequence the desired result will be obtained.

" A Flowchart is a symbolic representation of a solution to a given task.

" Program flowchart & System flowchart are two types of flowcharts

" Flowchart uses many symbols/shapes to denote different types of instructions.

" Flowchart symbols have been standardized by the American Standard Institute.

At the end we studied Flowcharts for Computation, Flowcharts for decision making, flowcharts for loops and flowchart for predefined process.

Source *http://jayaram.com.np* (e book)

1.9 CHECK YOUR PROGRESS - ANSWERS

1.2 & 1.3

1. a) An algorithm is a sequence of instructions designed in such a way that if the instructions are executed in a specific sequence the desired results will be

obtained. The instructions should be precise and concise and the result should be obtained after a finite execution of steps.

b) A flowchart is a symbolic representation of a solution to a given task. Flowcharting is a tool that can help us to develop and represent graphically program logic sequence.

c) There are two types of flowcharts : Program Flowcharts which are used by the programmers and which show the program structure, logic flow and operations performed. It also forms an important part of the documentation of the system and system flowcharts which are used by system analyst to show various processes, sub systems, outputs and operations on data in a system.

d) Any two steps in problem solving are :
(i) Detailed study of the problem
(ii) Identification of input data, output requirements and conditions and limitations

1.4

1. a) The decision box is used in a flowchart to indicate the point where a decision is to be made and branching done upon the result of the decision to one or more alternative paths. The criteria for decision making is written in the decision box. All the possible paths should be accounted for. During execution, the appropriate path will be followed depending upon the result of the decision.

b) Flowlines are solid lines with arrowheads which are used to indicate the flow of operation. They show the exact sequence in which the instructions are to be executed.

2. a - (iii) b - (iv) c - (ii) d - (i)

1.5

1. Advantages of Flowcharts :

(i) A flowchart shows the execution of logical steps without the syntax and language complexities of a program.

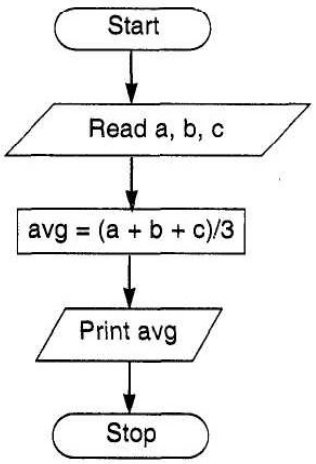
(ii) In real life programming situations a number of programmers are associated with the development of a system and each programmer is assigned a specific task of the entire system. Hence, each programmer can develop his own flowchart and later on all the flowcharts can be combined for depicting the overall system. Any problems related to linking of different modules can be detected at this stage itself and suitable modifications carried out. Flowcharts can thus be used as working models in design of new software systems.

1.6

1. a) To find average of three numbers a, b ,c.

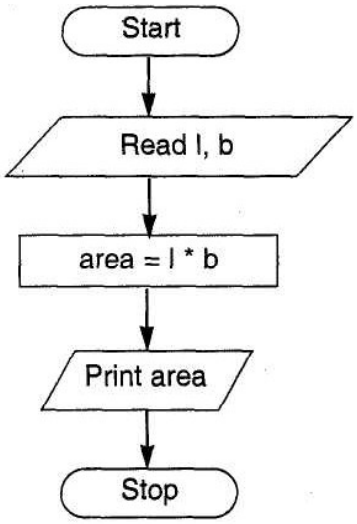
b) To find area of a rectangle whose length and breadth area read.

2. While developing flowcharts the points to be taken into consideration are :



The steps are :
1. Start
2. Read numbers a, b, c
3. Compute the average as (a + b + c)/3
4. Print average
5. Stop.

(i) Defining the problem.



The steps are :
1. Start
2. Read length l and breadth b
3. Compute the area as l * b
4. Print area
5. Stop.

(ii) Identifying the various steps required to form a solution.

(iii) Determining the required input and output parameters.

(iv) Getting expected input data values and output result.

(v) Determining the various computations and decisions involved.

1.7

1.

a) Programming considerations for computation : Most languages have provision for creating memvars. The exact syntax depends on the language used. In most cases your programs have to create and initialize the memvars before you can use them.

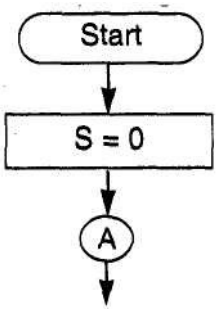
b) Programming considerations for loops : Most programming languages have commands for performing test and branch. Some of the conditional constructs available in programming languages for implementing decision making in programs are as follows: If, If - else - endif, If - elseif - endif, Do case - endcase, Switch.

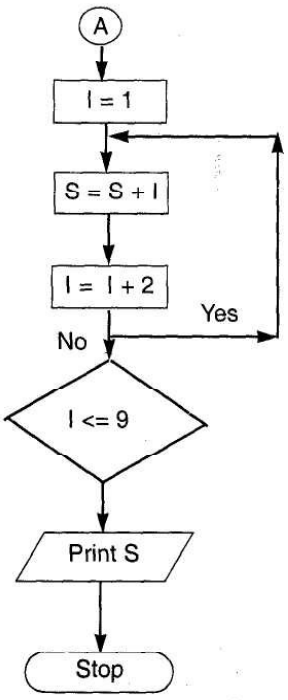
The operators available for implementing the decision test are as follows:

Relational Operators, logical Operators.

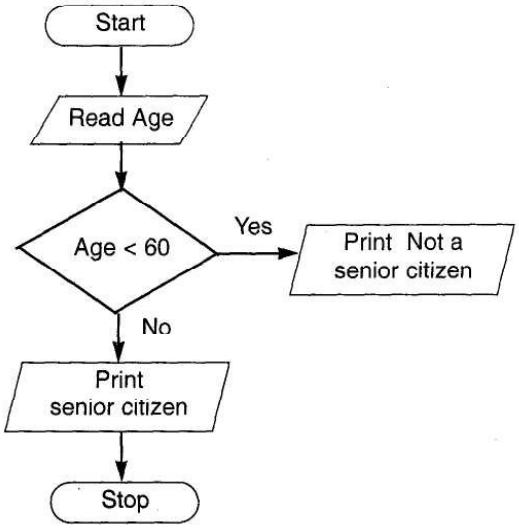
The branching to another set of commands can be implemented by using functions, procedures etc

2.a) Flowchart for printing the first five odd numbers :





b) Read age and print comment



1.10 QUESTIONS FOR SELF-STUDY

1) Answer the following in 7-8 Sentences

a) What characteristics should an algorithm have?
b) State the important points to be considered when developing flowcharts.
c) What is meant by fixed loops and variable loops? **2) Write short note on the following in 10-12 lines**

a) Flowcharts for loops
b) Flowcharts for decision making

3) Write the algorithm and draw flowcharts for the following :-

a) Convert distance entered in Km to Metres.
b) Find the product of the just n numbers.
c) Find the sum of digits of a three digit number.

1.11 SUGGESTED READINGS

The Spirit of C : Mullish cooper

Let us C : Yashwant kanitkar

The C programming Language: Kernigham & Ritchie

◈◈◈

NOTES