Unit 13 □ Flowchart

13.0 Objectives

The objectives of the Unit are to understand the :

- Meaning of flowchart
- Basic parts of the flowchart such as flowchart symbols and the flow lines connecting these symbols.
- Advantages and limitations of flowchart

13.1 Introduction

The use of data processing equipment has focused attention upon the necessity for an orderly representation of information flow. The sequence in which operations are to be executed should be precisely stated. The data and sequence of operations to be performed upon the data together constitute the information flow.

A flow chart can be customized to fit any need or purpose. For this reason, flow charts can be recognized as a very unique quality improvement method. Flowcharts were used historically in electronic data processing to represent the conditional logic or computer programmes. The characteristics of flowcharts may be summerized as follows :

- A language to specify the logic of algorithms
- Flowcharts are not a programming language, rather it is an algorithm specification language.
- In an algorithm we specify the computation operations (arithmetic, logical, data input, data output) to be done, and their sequence.
- Each type of computation operation is denoted by a special symbol in the flowchart language
- Symbols are connected by arrows to represent the order of the operations (sequence)
- Flowchart are useful during algorithm design
- A flowchart can be relatively easily translated to a C++ programme

13.2 Definition

A flowchart (also spelled flow-chart and flow chart) is a schematic representation of a process. They are commonly used in business/economic presentations to help the audience to visualize the content better, or to find flaws in the process.

Information system flowcharts show how data flows from source documents through the computer to final distribution to users. Programme flowcharts show the sequence of instructions in a single programme or subroutine. Different symbols are used to draw each type of flowchart.

Four particular types of flow charts have proven useful when dealing with a process analysis : top-down flow chart, detailed flow chart, work flow diagrams, and a deployment chart. Each of the different types of flow charts tends to provide a different aspect to a process or a task. Flow charts provide an excellent form of documentation for a process, and quite often are useful when examining how various steps in a process work together.

13.3 Types of Flowchart

The pictorial representation of the programmes or the algorithm is known as flowcharts. It is nothing but a diagrammatic representation of the various steps involved in designing a system. Flowcharts are of three types :
- System flowcharts
- Run flowcharts
- Programme flowcharts

13.3.1 System Flowcharts

System flowchart describes the data flow for a data processing system. It provides a logical diagram of how the system operates. It represents the flow of documents, the operations performed in data processing system. It also reflects the relationship between inputs, processing and outputs. Following are the features of system flowcharts :

● The sources from which data is generated and device used for this purpose

● Various processing steps involved

● The intermediate and final output prepared and the devices used for their storage

The following is a sample of system flowchart for the following algorithm :

Prompt the user for the centigrade temperature.

1. Store the value in C

2. Set F to 32+(9' C/5)

3. Print the value of C, F

4. Stop
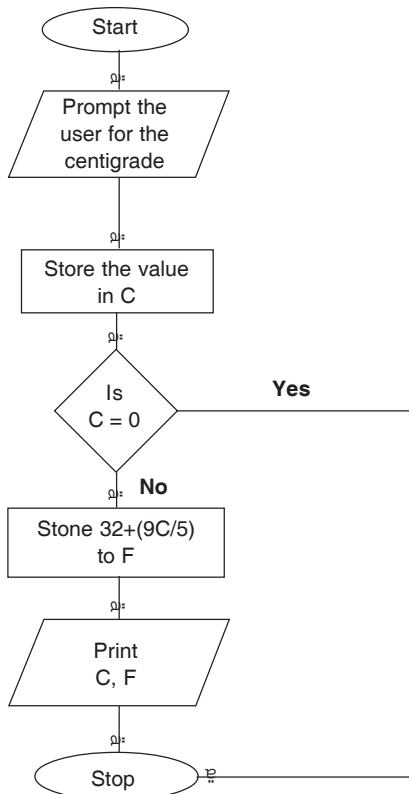
13.3.2 Run flowcharts

Run flowcharts are used to represent the logical relationship of computer routines along with inputs, master files, transaction files and outputs. The following illustrates a run flowchart.
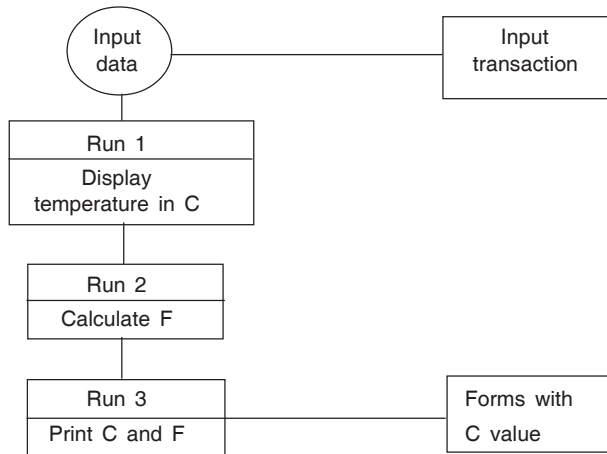
13.3.3 Programme flowcharts

A programme flowchart represents, in detail, the various steps to be performed within the system for transforming the input into output. The various steps are logical/ arithmetic operations, algorithms etc. It serves as the basis for discussions and communication between the system analysts and the programmers. Programme flow-charts are quite helpful to programmers in organizing their programming efforts. These flowcharts constitute an important components of documentation for an application.

The following figure represents a programme flowcharts for finding the sum of first five natural numbers (1-5).
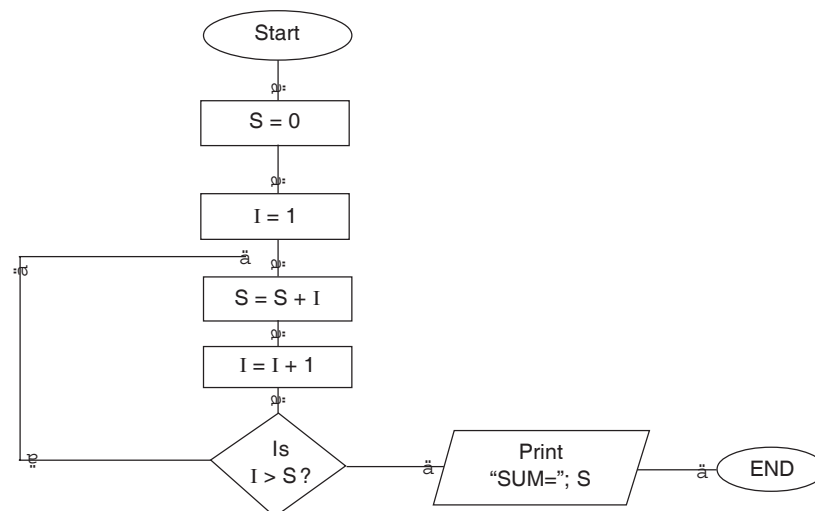
13.4 Guidelines for drawing a flowchart

Flowcharting is a tool for analyzing processes. It allows breaking any process down into individual events or activities and to display these in shorthand form showing the logical relationships between them. Constructing flowcharts promotes better understanding of processes and better understanding of processes is a pre-requisite for improvement.

Flowcharts are usually drawn using some standard symbols; however, some special symbols can also be developed when required. Some standard symbols, which are frequently, required for flowcharting many computer programmes are shown in the following table :

Symbols	Explanation
	Start or end of the programme
	Use it to represent an event, which occurs automatically. Such an event will trigger a subsequent action, for example 'receive telephone call', or describe a new state of affairs.
	Data
	Input or output operation
	Stored data
	Off-page connector
	Decision making and branching. Use it to represent a decision point in the process. Typically, the statement in the symbol will require a 'yes' or 'no' response and branch to different parts of the flowchart accordingly.
	Manual input
	Predefined process
	Connector or joining of two parts of programme. Use it to represent a point at which the flowchart connects with another process. The name or reference for the other process should appear within the symbol.
	Sequential Access Storage (Magnetic Tape)
	Magnetic Disk
	Direct Access storage
	Flow lines
	Display

217

There are no hard and fast rules for constructing flowcharts, but there are guidelines, which are useful to bear in mind. Here are six steps, which can be used as a guide for completing flowcharts.

- In drawing a proper flowchart, all necessary requirements should be listed out in logical order.

- The flowchart should be clear, neat and easy to follow. There should not be any room for ambiguity in understanding the flowchart.

- The usual direction of the flow of a procedure or system is from left to right or top to bottom.

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

or
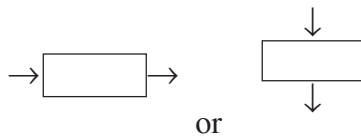
- Only one flow line should enter a decision symbol, but two or three flow lines, one for each possible answer, should leave the decision symbol.

< 0 > 0 < 0 > 0

$= 0$ $= 0$

- Only one flow line is used in conjunction with terminal symbol.

- Write within standard symbols briefly. As necessary, you can use the annotation symbol to describe data or computational steps more clearly.

------ This is top secrert data
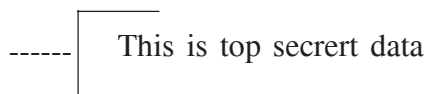
- If the flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines. Avoid the intersection of flow lines if you want to make it more effective and better way of communication.

- Ensure that the flowchart has a logical start and finish.

- It is useful to test the validity of the flowchart by passing through it with a simple test data.

218

A flowchart consists of a set of 'flowchart symbols' connected by arrows. Each symbols contains information about what must be done at that point & the arrow shows the 'flow of execution' of the algorithm i.e. they show the order in which the instructions must be executed. The purpose of using flowcharts is to graphically present the logical flow of data in the system and defining major phases of processing along with the various media to be used.

13.5 Advantages

The benefits of flowcharts are as follows :

- Communication : Flowcharts are better way of communicating the logic of a system to all concerned.

- Effective analysis : With the help of flowchart, problem can be analyzed in more effective way.

- Proper documentation : Programme flowcharts serve as a good programme documentation, which is needed for various purposes.

- Efficient Coding : The flowcharts act as a guide or blueprint during the systems analysis and programme development phase.

- Proper Debugging : The flowchart helps in debugging process.

- Efficient Programme Maintenance : The maintenance of operating programme becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part.

13.6 Limitations

- Complex logic : Sometimes, the programme logic is quite complicated. In that case, flowchart becomes complex and clumsy.

- Alterations and Modifications : If alterations are required the flowchart may require re-drawing completely.

- Reproduction : As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.

- The essentials of what is done can easily be lost in the technical details of how it is done.

13.7 Creating flowcharts on a computer

There are various packages for creating flowcharts, according to different standards. The most common is UML, for which there are abundant packages for various platforms. See UML article for list. The creation of simple flowcharts on a computer is fairly easy with any vector-based drawing programme, but Microsoft Word (versions 97 through 2003) and Openoffice. Org (Draw app) both have specialized tools for making consistent charts. For Mac OS X OmniGraffle is an excellent application.

13.8 Examples

13.8.1 Draw a flowchart to find the sum of first 50 natural numbers.

```
        ┌──────────────┐
        │    START     │
        └──────┬───────┘
               │
        ┌──────┴───────┐
        │   SUM = 0    │
        └──────┬───────┘
               │
        ┌──────┴───────┐
        │    N = 0     │
        └──────┬───────┘
               │
        ┌──────┴───────┐
  NO    │   N = N + 1  │
        └──────┬───────┘
               │
        ┌──────┴───────┐
        │ SUM = SUM + N│
        └──────┬───────┘
               │
          ◇ IS N = 50 ? ◇
               │ YES
        ╱──────────────╱
       ╱  PRINT SUM   ╱
      ╱──────────────╱
               │
        ┌──────┴───────┐
        │    END       │
        └──────────────┘
```

220
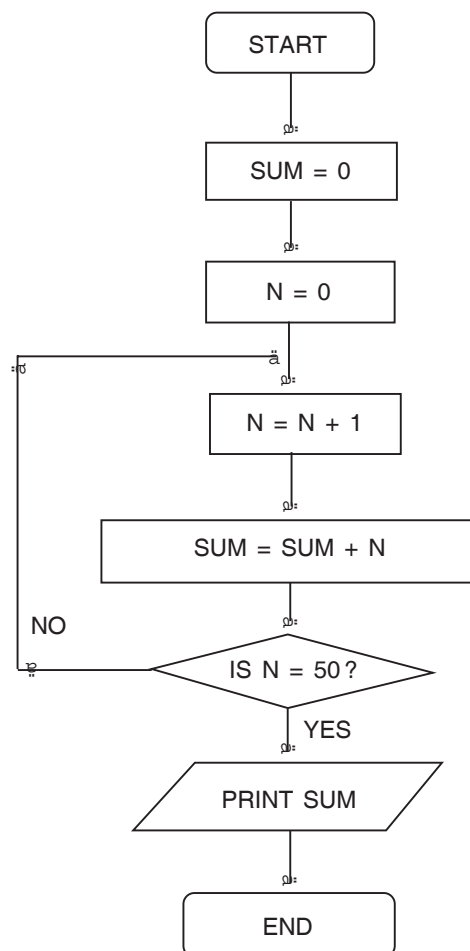
13.8.2 Draw a flowchart to find the largest of three numbers A, B, and C.

```
                         ┌──────────┐
                         │  START   │
                         └────┬─────┘
                              │
                      ╱───────────────╲
                     ╱  READ A, B, C    ╲
                     ╲                   ╱
                      ╲─────────┬───────╱
                                │
         ┌─────────────┐   ┌──────────┐   ┌──────────┐
  YES    │     IS      │NO │    IS    │YES│    IS    │ YES
 ◄───────┤   B > C ?   ├───┤  A > B?  ├───┤  A > C?  ├───►
         └──────┬──────┘   └─────┬────┘   └─────┬────┘
                │              NO │           NO │
                │                 │              │
        ╱───────────╲    ╱───────────╲   ╱───────────╲    ╱───────────╲
       ╱  PRINT B    ╲  ╱  PRINT C    ╲ ╱  PRINT C    ╲  ╱  PRINT A    ╲
       ╲─────┬───────╱  ╲─────┬───────╱ ╲─────┬───────╱  ╲─────┬───────╱
             │                │               │                │
             └────────────────┴───────┬───────┴────────────────┘
                                       │
                                  ┌─────────┐
                                  │   END   │
                                  └─────────┘
```
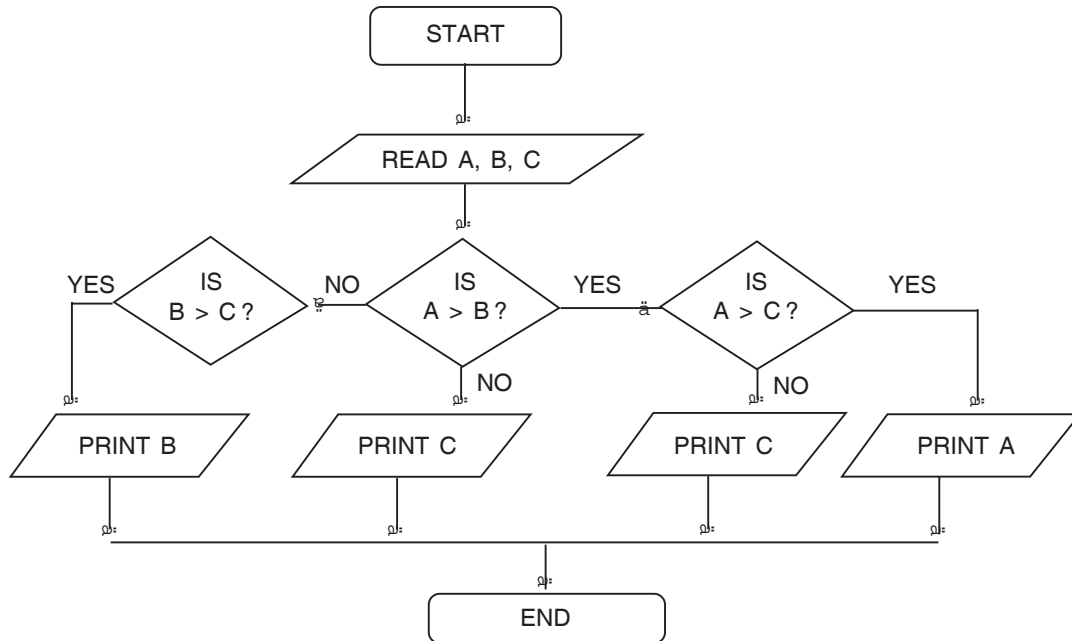
13.8.3 Calculate the average from 25 examination scores

```
                  ╭──────────╮
                  │  START   │
                  ╰────┬─────╯
                       │
                ┌──────────────┐
                │   Sum = 0    │
                │    C = 0     │
                └──────┬───────┘
                       │
              ╱─────────────────╲
             ╱   Enter            ╲
            ╱  Examination         ╲
            ╲  Score  S            ╱
             ╲───────┬────────────╱
                     │
              ┌──────────────┐
              │   Sum =      │
              │   Sum + S    │
              └──────┬───────┘
                     │
              ┌──────────────┐
              │  C = C + 1   │
              └──────┬───────┘
                     │
              ╱───────────────╲
     No      ╱   Is C > 25     ╲    Yes
    ◄───────┤        ?          ├───────►  ┌──────────────┐    ╱──────────╲    ╭────────╮
            ╲                   ╱           │  Average =   │   ╱  Output    ╲   │  Stop  │
             ╲─────────────────╱            │   Sum / 25   ├──╱   Average    ╲──│        │
                                            └──────────────┘  ╲───────────────╱  ╰────────╯
```

221
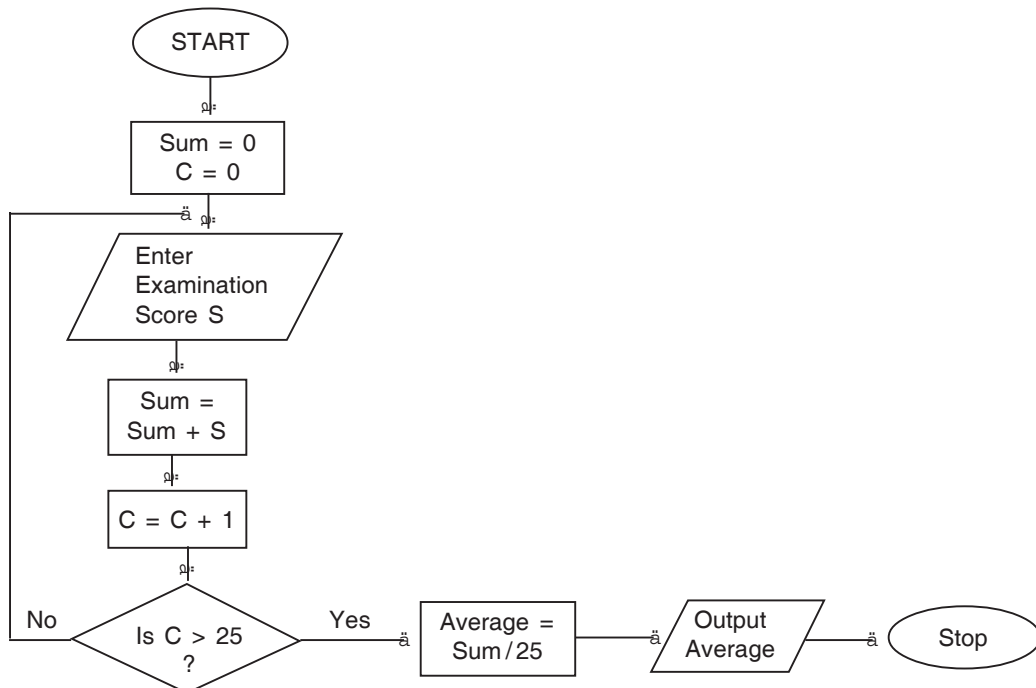
References and Further Readings

1 2005 NOS : Certificate in Computer Application. Lesson 25 : Flowcharting. (http://nos.org/htm/basic2.htm). Visited last : 4/10/2005

2 2005 Flowcharts (http://deming.eng.clemson.edu/pub/tutorials/qctools/flowm.htm) Visited last : 4/10/2005

3 1970 Flowcharting techniques (NY : IBM, 1970). Document Code : C20-8152-1).

13.9 Exercise

1. What is flowchart? Describe various types of flowcharts.

2. Describe how to design flowchart of an activity.

3. Describe important flowchart symbols.