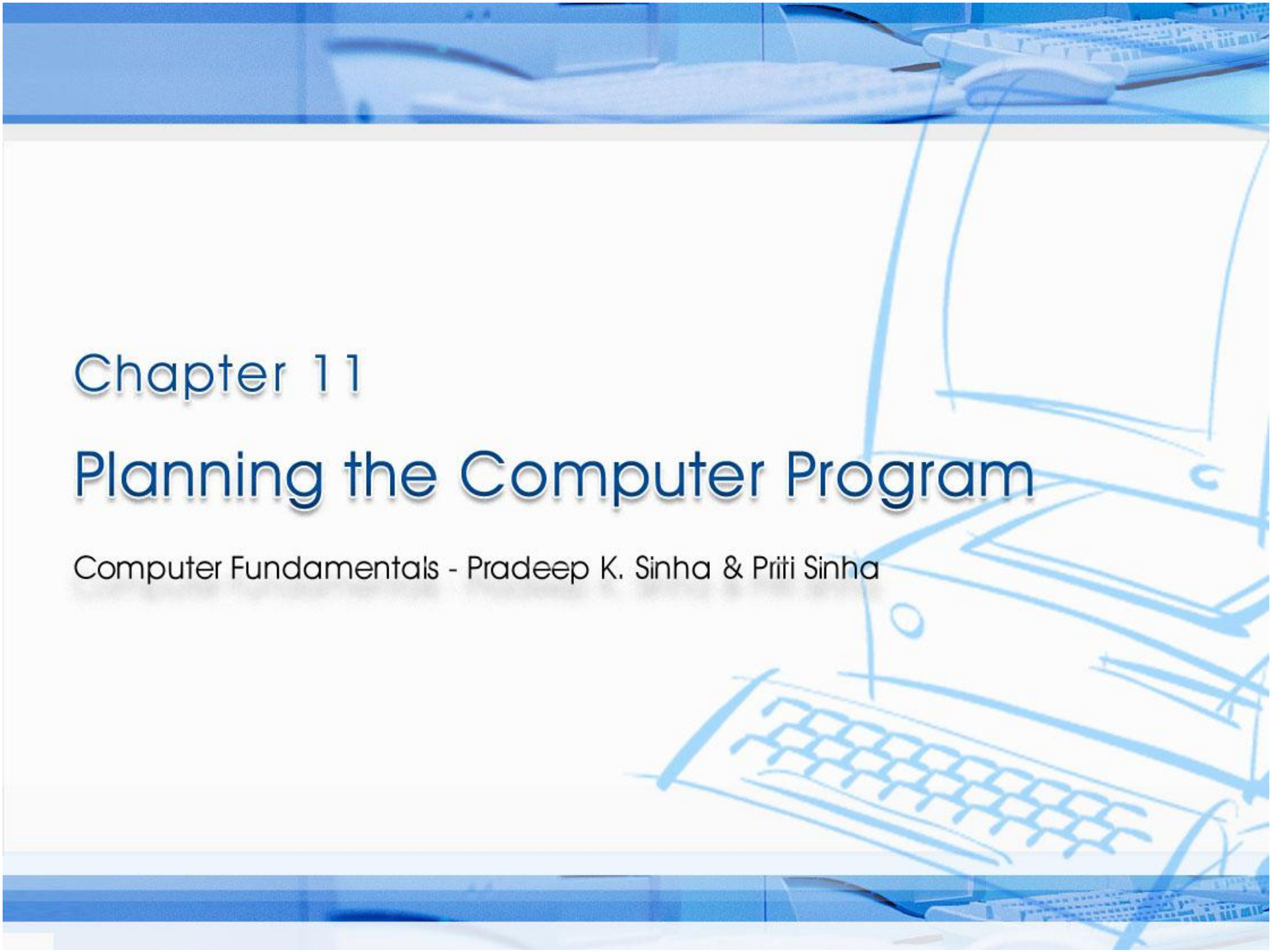


Chapter 11

Planning the Computer Program

Computer Fundamentals - Pradeep K. Sinha & Priti Sinha

Learning Objectives

In this chapter you will learn about:

- Programs must be planned before they are written
- Algorithm
- Flowchart
- Pseudocode
- Plan the logic of a computer program
- Commonly used tools for program planning and their use

Purpose of Program Planning

- To write a correct program, a programmer must write each and every instruction in the correct sequence
- Logic (instruction sequence) of a program can be very complex
- Hence, programs must be planned before they are written to ensure program instructions are:
  - Appropriate for the problem
  - In the correct sequence

Algorithm

- Refers to the logic of a program and a step-by-step description of how to arrive at the solution of a given problem
- In order to qualify as an algorithm, a sequence of instructions must have following characteristics:
  - Each and every instruction should be precise and unambiguous
  - Each instruction should be such that it can be performed in a finite time
  - One or more instructions should not be repeated infinitely. This ensures that the algorithm will ultimately terminate
  - After performing the instructions, that is after the algorithm terminates, the desired results must be obtained

Sample Algorithm (Example 1)

There are 50 students in a class who appeared in their final examination. Their mark sheets have been given to you.

The division column of the mark sheet contains the division (FIRST, SECOND, THIRD or FAIL) obtained by the student.

Write an algorithm to calculate and print the total number of students who passed in FIRST division.

Sample Algorithm (Example 1)

(contd...)

Step 1: Initialize Total_First_Division and Total_Marksheets_Checked to zero.

Step 2: Take the mark sheet of the next student.

Step 3: Check the division column of the mark sheet to see if it is FIRST, if no, go to Step 5.

Step 4: Add 1 to Total_First_Division.

Step 5: Add 1 to Total_Marksheets_Checked.

Step 6: Is Total_Marksheets_Checked = 50, if no, go to Step 2.

Step 7: Print Total_First_Division.

Step 8: Stop.

Sample Algorithm (Example 2)

There are 100 employees in an organization. The organization wants to distribute annual bonus to the employees based on their performance. The performance of the employees is recorded in their annual appraisal forms.

Every employee's appraisal form contains his/her basic salary and the grade for his/her performance during the year. The grade is of three categories – 'A' for outstanding performance, 'B' for good performance, and 'C' for average performance.

It has been decided that the bonus of an employee will be 100% of the basic salary for outstanding performance, 70% of the basic salary for good performance, 40% of the basic salary for average performance, and zero for all other cases.

Write an algorithm to calculate and print the total bonus amount to be distributed by the organization.

Sample Algorithm (Example 2)

(contd...)

Step 1: Initialize Total_Bonus and Total_Employees_Checked to zero.

Step 2: Initialize Bonus and Basic_Salary to zero.

Step 3: Take the appraisal form of the next employee.

Step 4: Read the employee's Basic_Salary and Grade.

Step 5: If Grade = A, then Bonus = Basic_Salary. Go to Step 8.

Step 6: If Grade = B, then Bonus = Basic_Salary x 0.7. Go to Step 8.

Step 7: If Grade = C, then Bonus = Basic_Salary x 0.4.

Step 8: Add Bonus to Total_Bonus.

Step 9: Add 1 to Total_Employees_Checked.

Step 10: If Total_Employees_Checked < 100, then go to Step 2.

Step 11: Print Total_Bonus.

Step 12: Stop.

Representation of Algorithms

- As programs
- As flowcharts
- As pseudocodes

When an algorithm is represented in the form of a programming language, it becomes a program

Thus, any program is an algorithm, although the reverse is not true

Flowchart

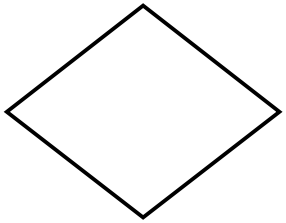
- *Flowchart* is a pictorial representation of an algorithm
- Uses symbols (boxes of different shapes) that have standardized meanings to denote different types of instructions
- Actual instructions are written within the boxes
- Boxes are connected by solid lines having arrow marks to indicate the exact sequence in which the instructions are to be executed
- Process of drawing a flowchart for an algorithm is called *flowcharting*

Basic Flowchart Symbols

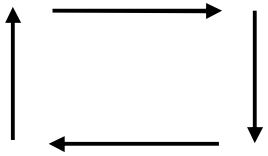
Terminal

Input/Output

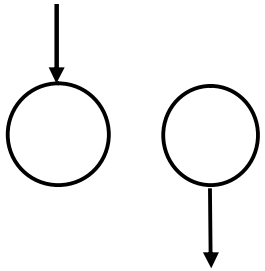
Processing



Decision

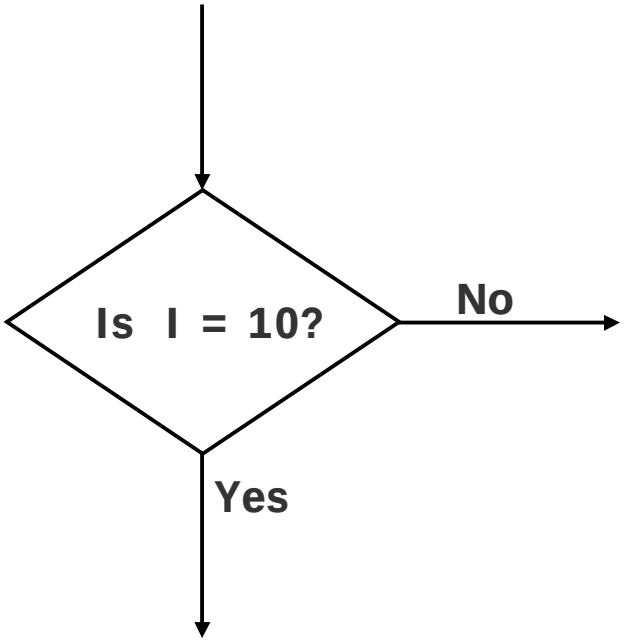


Flow lines

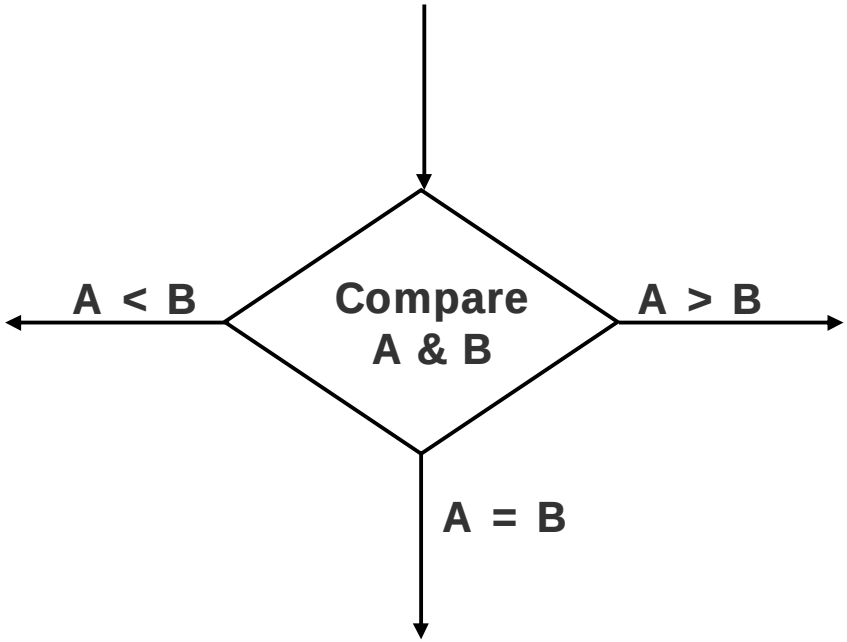


Connectors

Examples of Decision Symbol



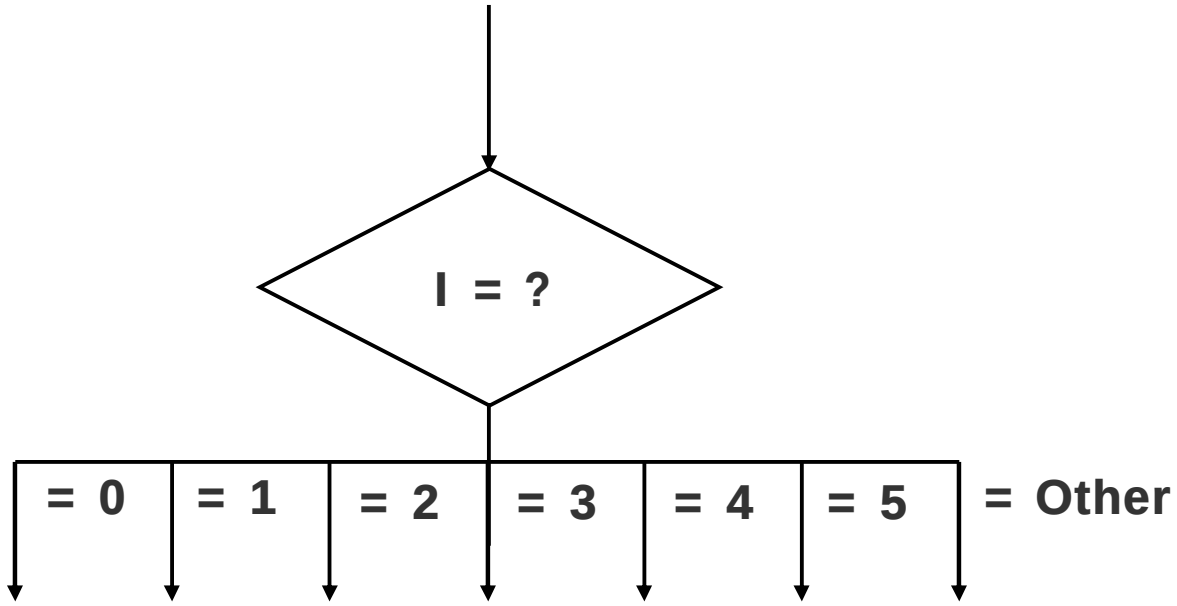
(a) A two-way branch decision.



(b) A three-way branch decision.

Examples of Decision Symbol

(contd...)



(c) A multiple-way branch decision.

Sample Flowchart (Example 3)

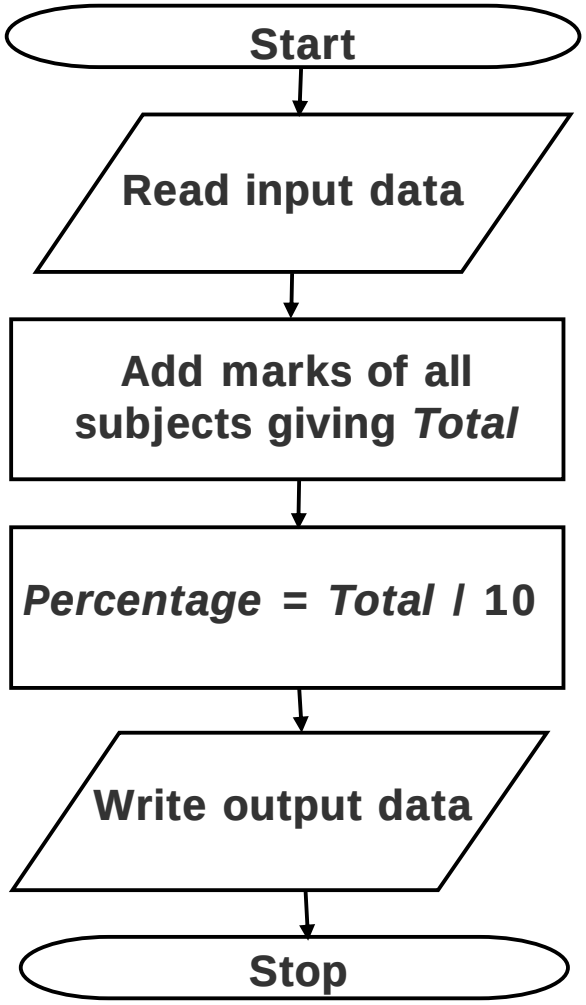
A student appears in an examination, which consists of total 10 subjects, each subject having maximum marks of 100.

The roll number of the student, his/her name, and the marks obtained by him/her in various subjects are supplied as input data.

Such a collection of related data items, which is treated as a unit is known as a record.

Draw a flowchart for the algorithm to calculate the percentage marks obtained by the student in this examination and then to print it along with his/her roll number and name.

Sample Flowchart (Example 3)



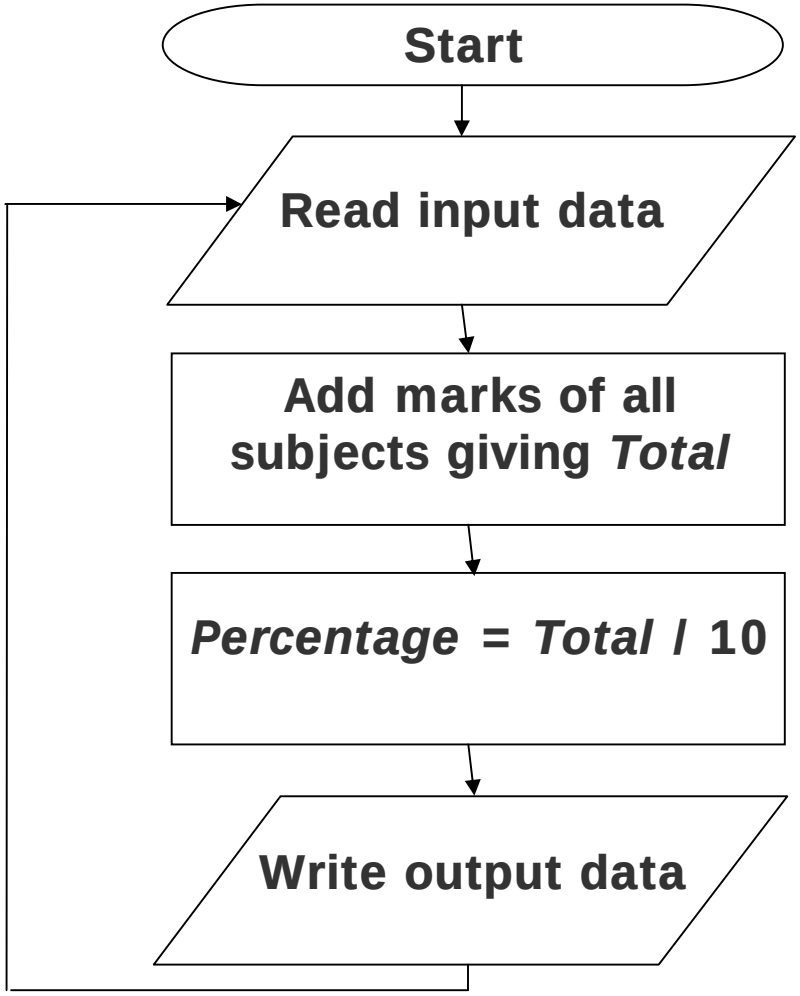
(contd...)

Sample Flowchart (Example 4)

50 students of a class appear in the examination of Example 3.

Draw a flowchart for the algorithm to calculate and print the percentage marks obtained by each student along with his/her roll number and name.

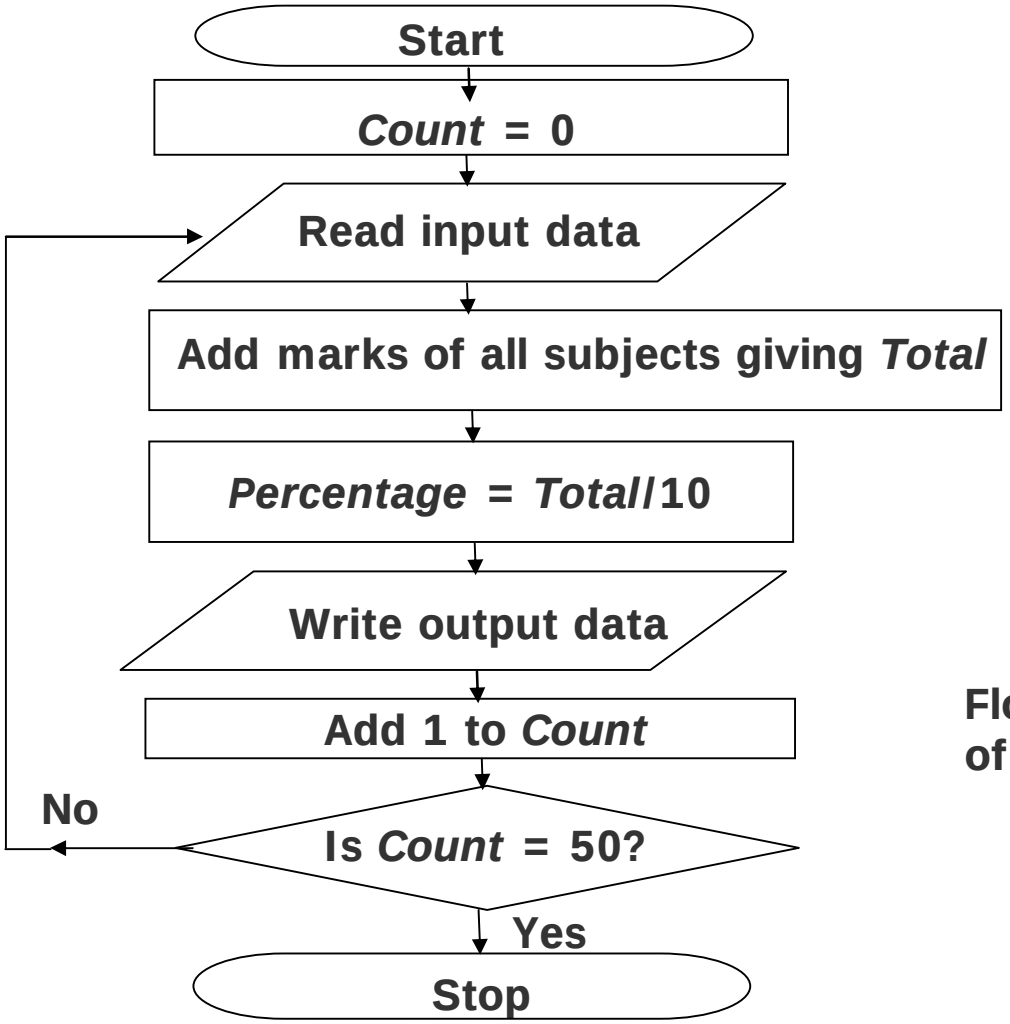
Sample Flowchart (Example 4)



(contd...)

Flowchart for the solution of Example 4 with an infinite (endless) process loop.

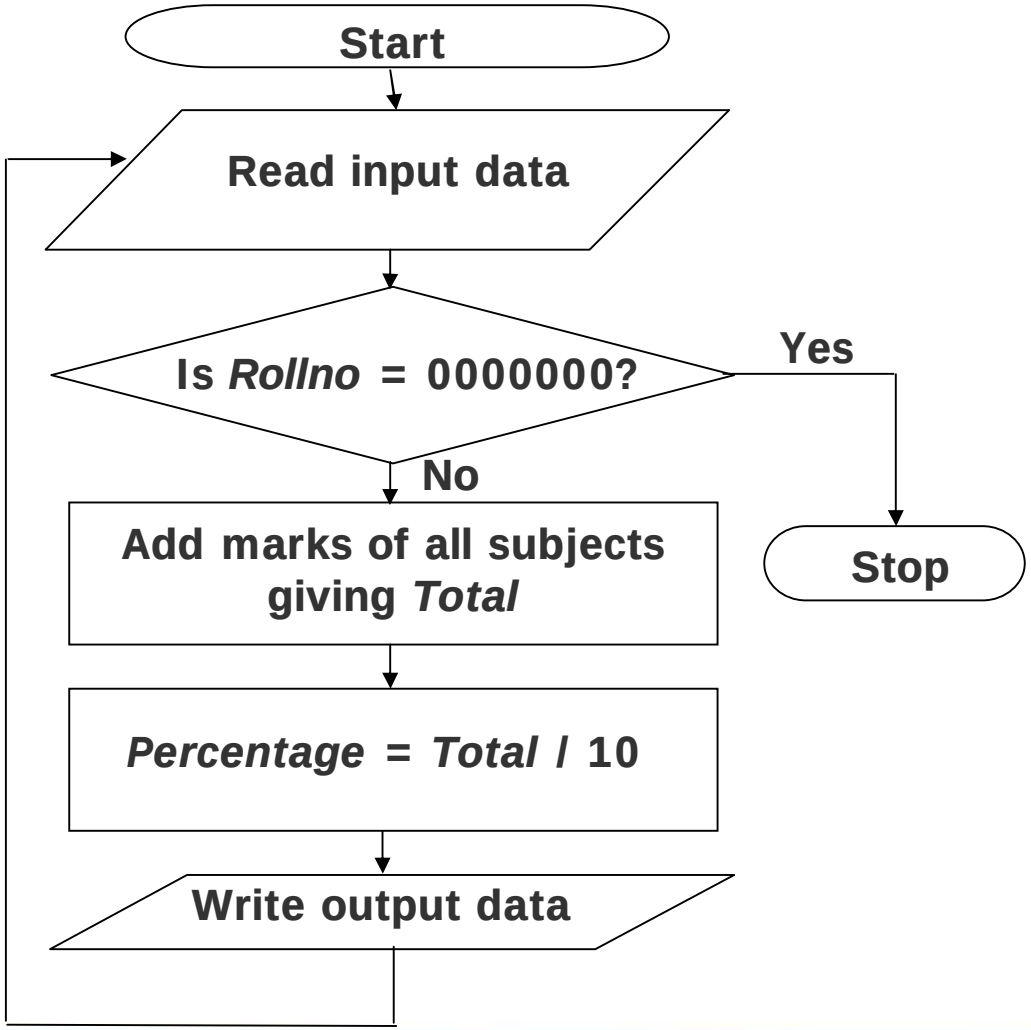
Sample Flowchart (Example 4)



(contd...)

Flowchart for the solution of Example 4.

Sample Flowchart (Example 4)



(contd...)

Generalized flowchart for the solution of Example 4 using the concept of trailer record. Here the process loop is terminated by detecting a special non-data record.

Sample Flowchart (Example 5)

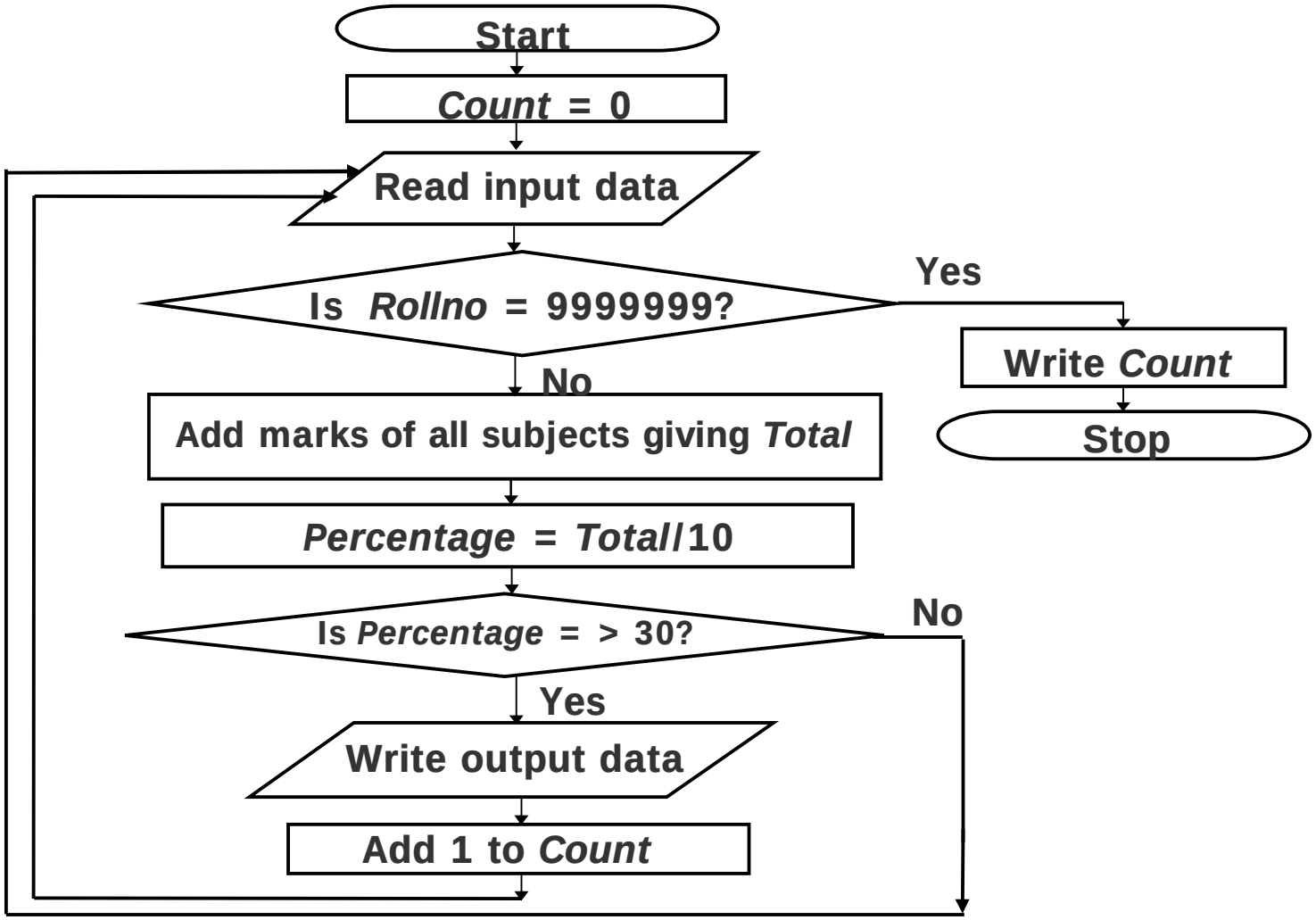
For the examination of Example 3, we want to make a list of only those students who have passed (obtained 30% or more marks) in the examination.

In the end, we also want to print out the total number of students who have passed.

Assuming that the input data of all the students is terminated by a trailer record, which has sentinel value of 9999999 for Rollno, draw a flowchart for the algorithm to do this.

Sample Flowchart (Example 5)

(contd...)



Sample Flowchart (Example 6)

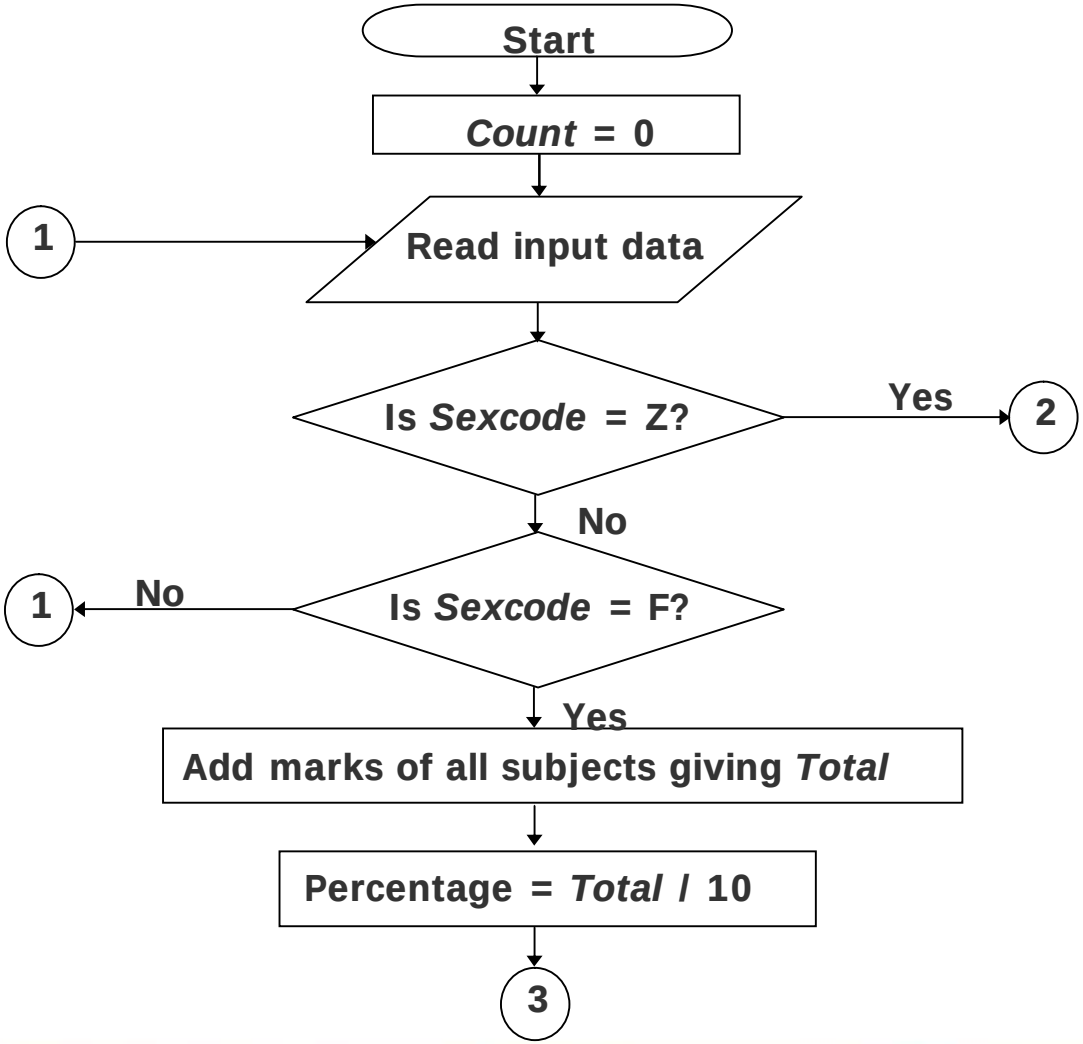
Suppose the input data of each student for the examination of Example 3 also contains information regarding the sex of the candidate in the field named *Sexcode* having values M (for male) or F (for female).

We want to make a list of only those female students who have passed in second division (obtained 45% or more but less than 60% marks).

In the end, we also want to print out the total number of such students.

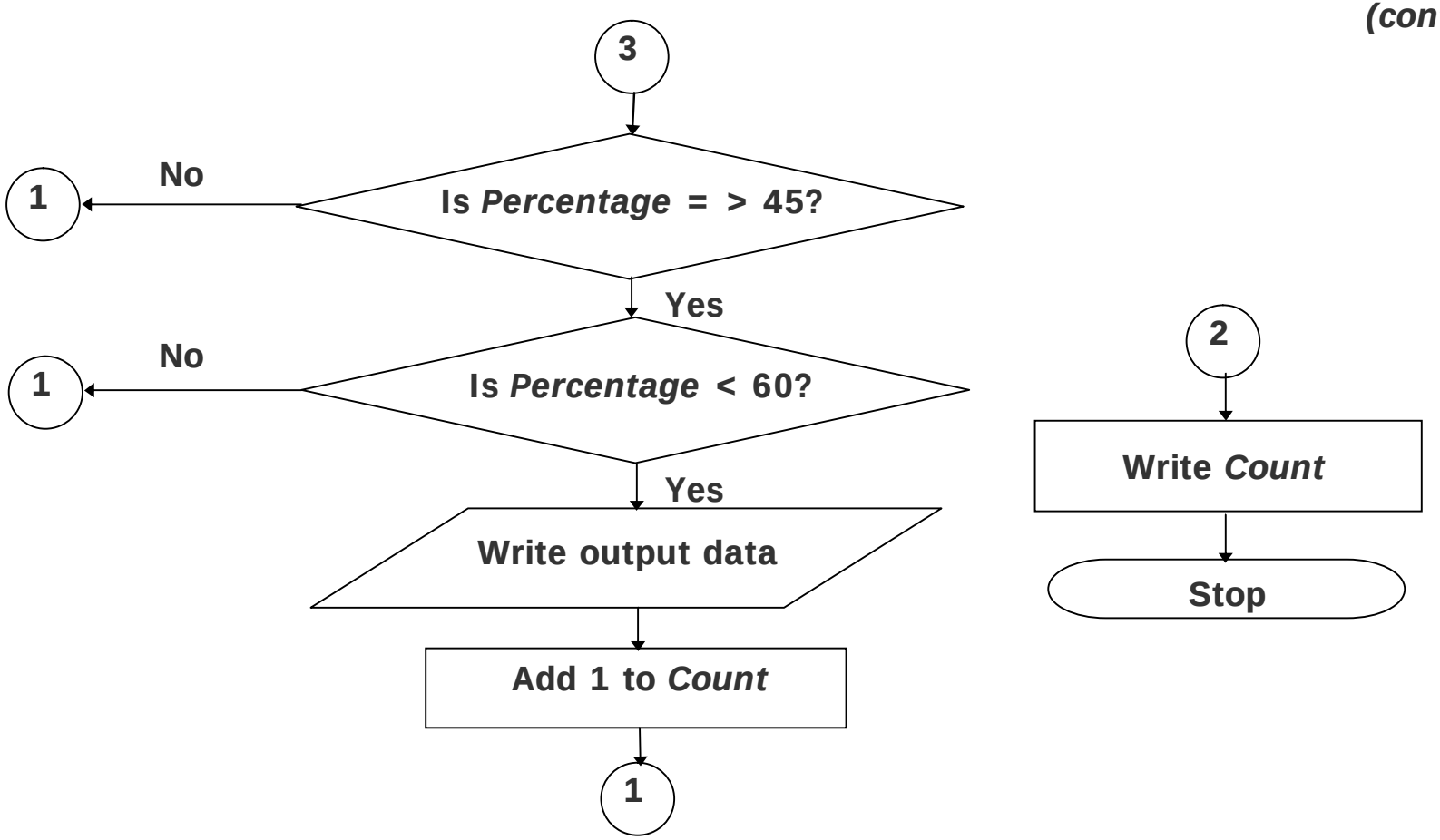
Assuming that the input data of all the students is terminated by a trailer record, which has a sentinel value of Z for Sexcode, draw a flowchart for the algorithm to do this.

Sample Flowchart (Example 6)



Sample Flowchart (Example 4)

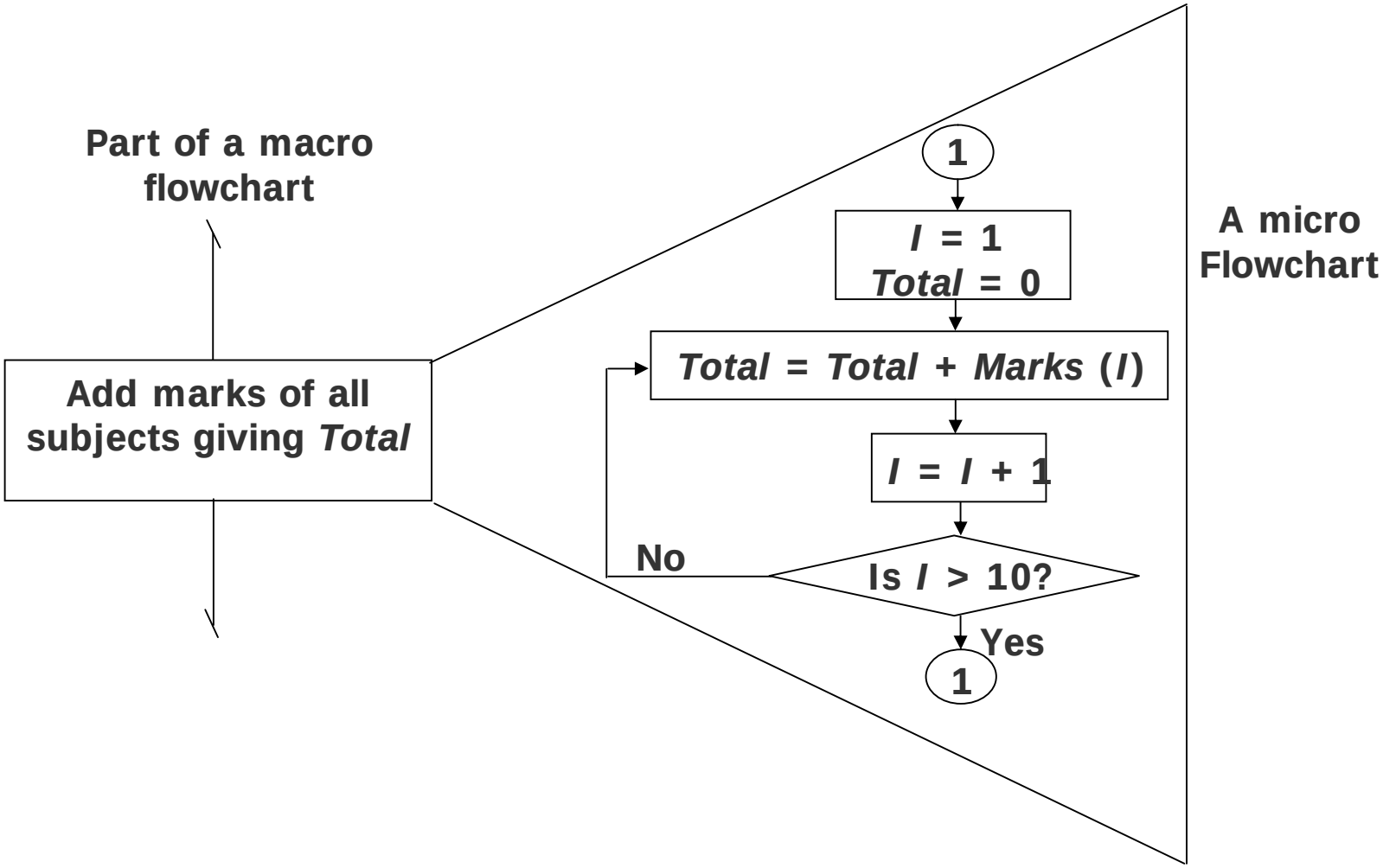
(contd...)



Levels of Flowchart

- Flowchart that outlines the main segments of a program or that shows less details is a *macro flowchart*
- Flowchart with more details is a *micro flowchart*, or detailed flowchart
- There are no set standards on the amount of details that should be provided in a flowchart

Example of Micro Flowchart



Flowcharting Rules

- First chart the main line of logic, then incorporate detail
- Maintain a consistent level of detail for a given flowchart
- Do not chart every detail of the program. A reader who is interested in greater details can refer to the program itself
- Words in the flowchart symbols should be common statements and easy to understand

Flowcharting Rules

(contd...)

- Be consistent in using names and variables in the flowchart
- Go from left to right and top to bottom in constructing flowcharts
- Keep the flowchart as simple as possible. Crossing of flow lines should be avoided as far as practicable
- If a new flowcharting page is needed, it is recommended that the flowchart be broken at an input or output point.
- Properly labeled connectors should be used to link the portions of the flowchart on different pages

Advantages of Flowchart

- Better Communication
- Proper program documentation
- Efficient coding
- Systematic debugging
- Systematic testing

Limitations of Flowchart

- Flowcharts are very time consuming and laborious to draw (especially for large complex programs)
- Redrawing a flowchart for incorporating changes/ modifications is a tedious task
- There are no standards determining the amount of detail that should be included in a flowchart

Pseudocode

- A program planning tool where program logic is written in an ordinary natural language using a structure that resembles computer instructions
- " Pseudo" means imitation or false and " Code" refers to the instructions written in a programming language. Hence, pseudocode is an imitation of actual computer instructions
- Because it emphasizes the design of the program, pseudocode is also called *Program Design Language (PDL)*

Basic Logic (Control) Structures

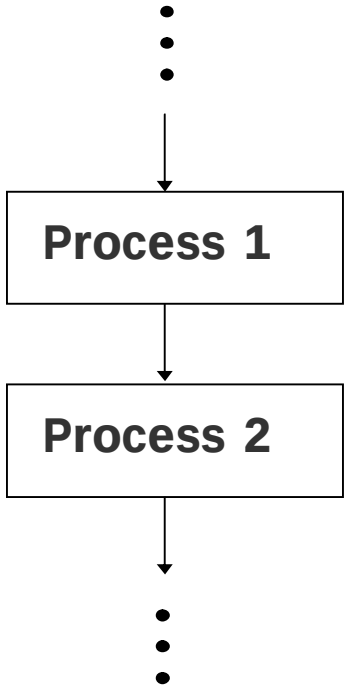
Any program logic can be expressed by using only following three simple logic structures:

1. Sequence logic,
2. Selection logic, and
3. Iteration (or looping) logic

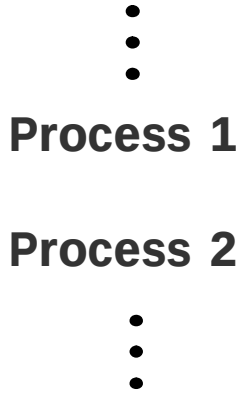
Programs structured by using only these three logic structures are called ***structured programs***, and the technique of writing such programs is known as ***structured programming***

Sequence Logic

It is used for performing instructions one after another in sequence.



(a) Flowchart



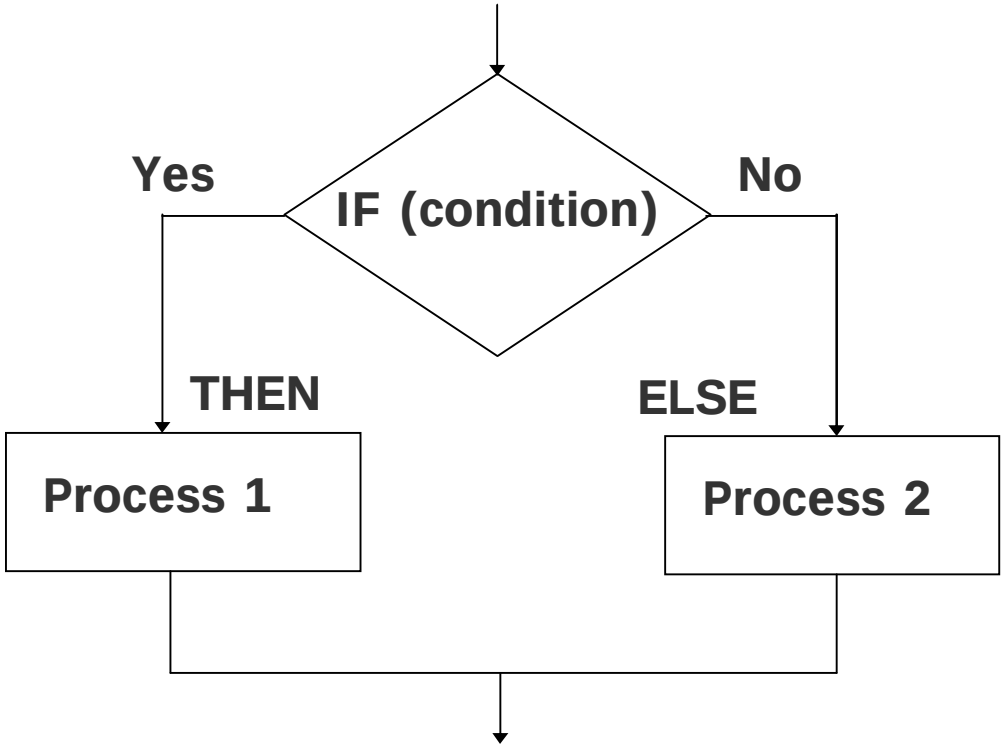
(b) Pseudocode

Selection Logic

- Also known as decision logic, it is used for making decisions

- Three popularly used selection logic structures are
  1. IF...THEN...ELSE
  2. IF...THEN
  3. CASE

Selection Logic (IF...THEN...ELSE Structure)



(a) Flowchart

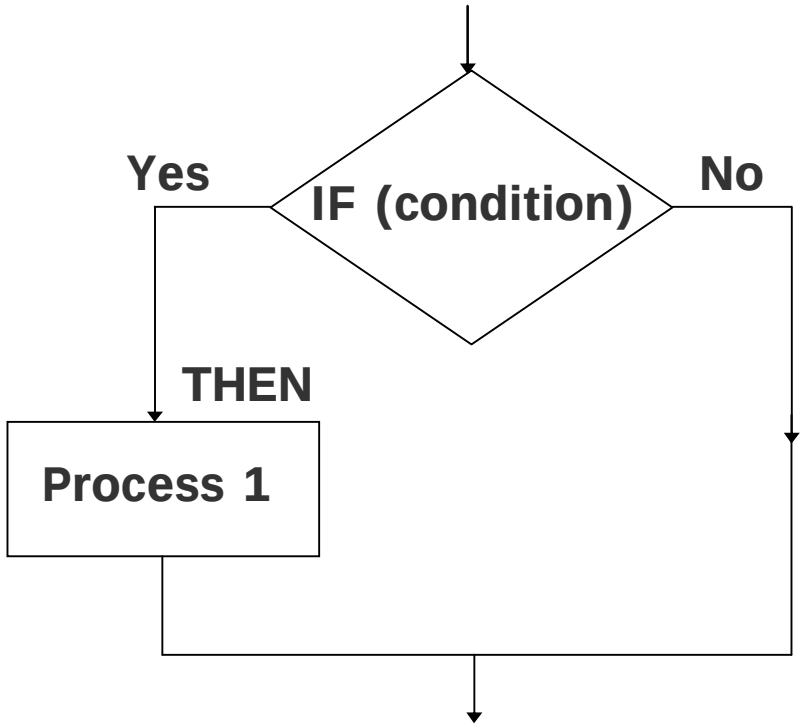
```
    .
    .
    .
IF Condition

    THEN    Process 1
    ELSE    Process 2

ENDIF
    .
    .
    .
```

(b) Pseudocode

Selection Logic (IF...THEN Structure)

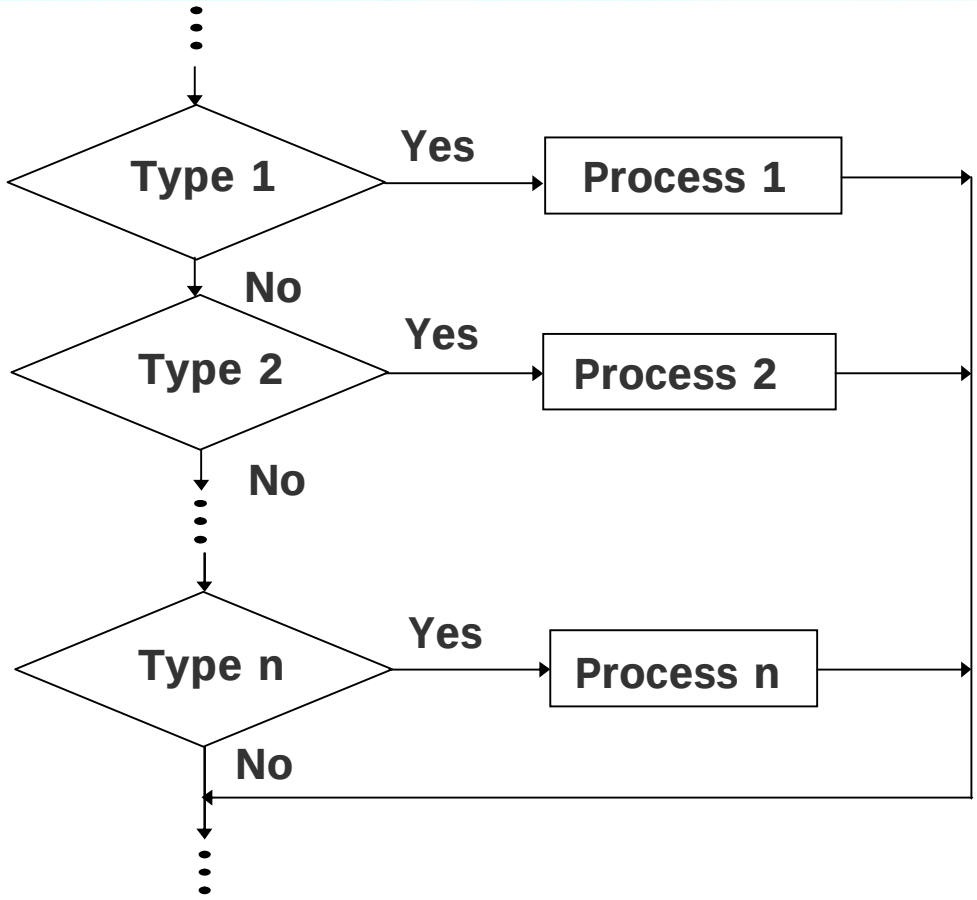


(a) Flowchart

```
:
IF Condition
    THEN    Process 1
ENDIF
:
```

(b) Pseudocode

Selection Logic (CASE Structure)



(a) Flowchart

```
CASE Type
    Case Type 1:    Process 1
    Case Type 2:    Process 2
    Case Type n:    Process n
ENDCASE
```

(b) Pseudocode

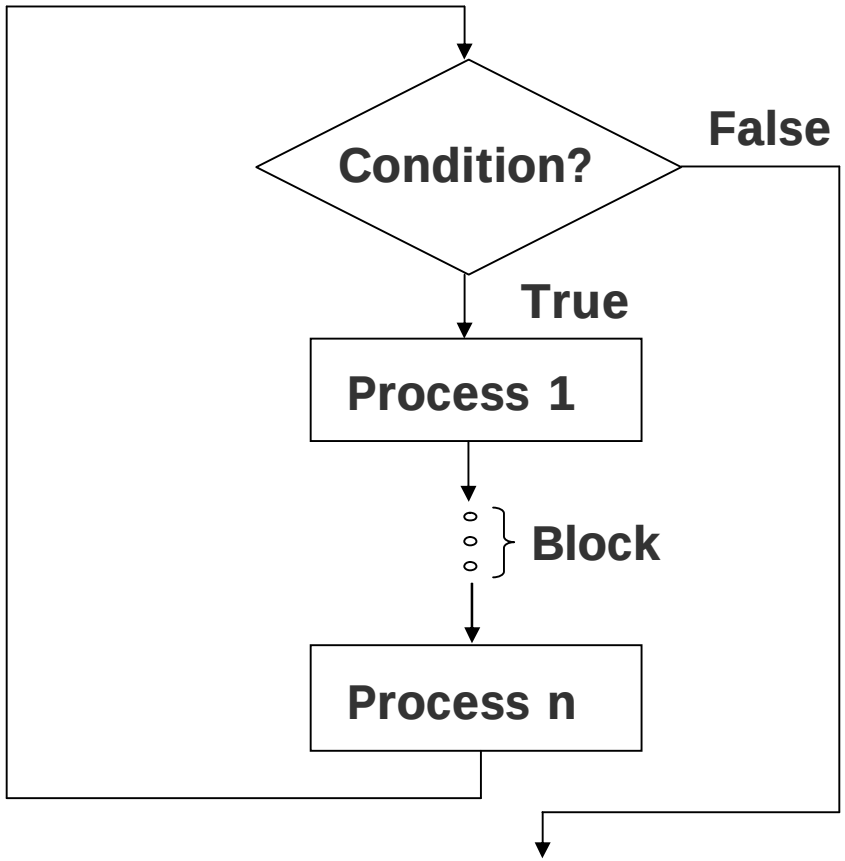
Iteration (or Looping) Logic

- Used to produce loops in program logic when one or more instructions may be executed several times depending on some conditions

- Two popularly used iteration logic structures are

  1. DO...WHILE
  2. REPEAT...UNTIL

Iteration (or Looping) Logic
(DO...WHILE Structure)

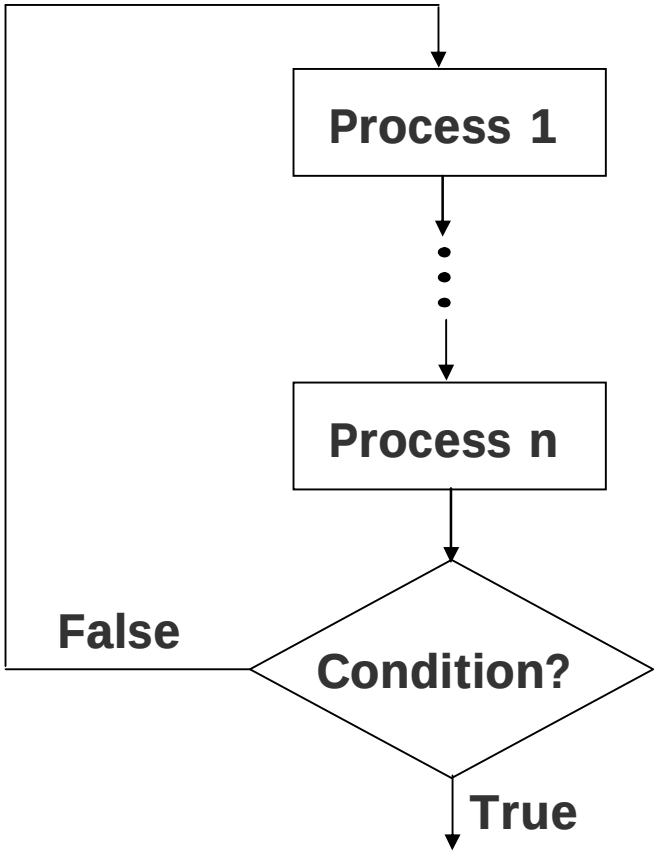


(a) Flowchart

```
:
DO WHILE Condition
    Process 1
    :
    Process n
ENDDO
:
```

(b) Pseudocode

Iteration (or Looping) Logic
(REPEAT...UNTIL Structure)



(a) Flowchart

```
    .
    .
    .
REPEAT
        Process 1
        .
        .
        .
        Process n
UNTIL Condition
    .
    .
    .
```

(b) Pseudocode

Sample Pseudocode (for Example 6)

```
Set Count to zero
Read first student record
DO WHILE Sexcode is not equal to Z
      IF Sexcode = F THEN
            Calculate Percentage
            IF Percentage = > 45 THEN
                  IF Percentage < 60 THEN
                        Write output data
                        Add 1 to Count
                  ENDIF
            ENDIF
      ENDIF
      Read next student record
ENDDO
Write Count
Stop
```

Advantages of Pseudocode

- Converting a pseudocode to a programming language is much more easier than converting a flowchart to a programming language
- As compared to a flowchart, it is easier to modify the pseudocode of a program logic when program modifications are necessary
- Writing of pseudocode involves much less time and effort than drawing an equivalent flowchart as it has only a few rules to follow

Limitations of Pseudocode

- In case of pseudocode, a graphic representation of program logic is not available
- There are no standard rules to follow in using pseudocode
- Different programmers use their own style of writing pseudocode and hence communication problem occurs due to lack of standardization
- For a beginner, it is more difficult to follow the logic of or write pseudocode, as compared to flowcharting

Key Words / Phrases

- Algorithm
- Basic logic structures
- Control structures
- Flowchart
- Iteration logic
- Looping logic
- Micro flowchart
- Macro flowchart
- Pseudocode
- Program Design Language (PDL)
- Sequence logic
- Selection logic
- Sentinel value
- Structured programming
- Trailer record