

COMPUTER SCIENCE

Class-VIII (Jan)

Chapter-13: Algorithms

Prepared by:

Md. Ariful Islam

1

Algorithm

Algorithm is a procedure or formula for solving a problem, based on conducting a sequence of specified actions.

A computer program can be viewed as an elaborate algorithm. In mathematics and computer science, an algorithm usually means a small procedure that solves a recurrent problem.

Criteria for successful algorithm:

- ❑ **Accuracy** - it must lead to the expected outcome.
- ❑ **Consistency** - it must produce the same result each time it is run.
- ❑ **Efficiency** - it must solve the problem in the shortest possible time using as few computer resources as possible. In this example the mapping software is replacing a manual method, and if it were no faster than looking in an atlas then it would not be an improvement on the older method. Later in the chapter there is a section on algorithms used to sort and search data. Some of these algorithms are more efficient than others and will sort the data far more quickly.

Relationship between algorithm & Program

- ❖ Algorithms and programs are closely related, but they are not the same.
- ❖ An algorithm is a detailed design for a solution; a program is the implementation of that design.
- ❖ Algorithm solves a specific problem, it is focused on a specific whole concept.
- ❖ A program can be an implementation of many algorithms, or an algorithm can be implemented by patching together many programs. A program can even contain no algorithms.

Flowcharts

Flowchart is a graphical representation of an algorithm. Each step in the algorithm is represented by a symbol. Symbols are linked together with arrows showing the order in which steps are executed.

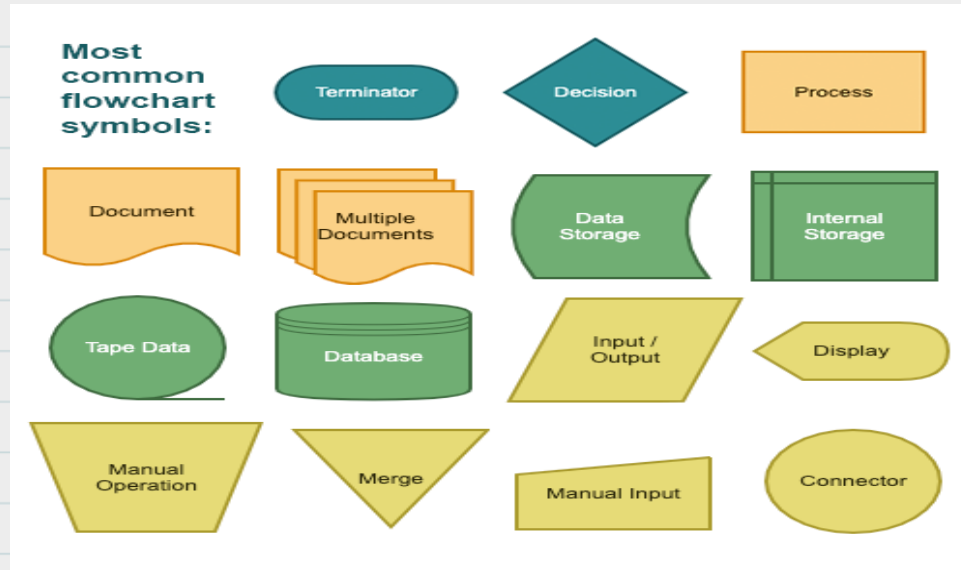


Fig: common flowchart symbols

Algorithm, for making a cup of coffee

Written Description:

- Fill kettle with water
- Turn on kettle
- place coffee in cup
- wait for water to boil
- pour water into cup
- add milk and sugar
- stir



Pseudo-code

Pseudo-code is a plain language description of the steps in an algorithm or another system.

- ❑ In addition to flowcharts and written descriptions, algorithms can also be expressed in pseudo-code. The pseudo-code can be used to code the solution in an actual programming language.
- ❑ It allows the developer to concentrate on the logic and efficiency of the algorithm without having to bother about the rules of any particular programming language.
- ❑ It is relatively straightforward to translate an algorithm written in pseudo-code into any high-level programming language.

Algorithm for adding two numbers:

Written Description:

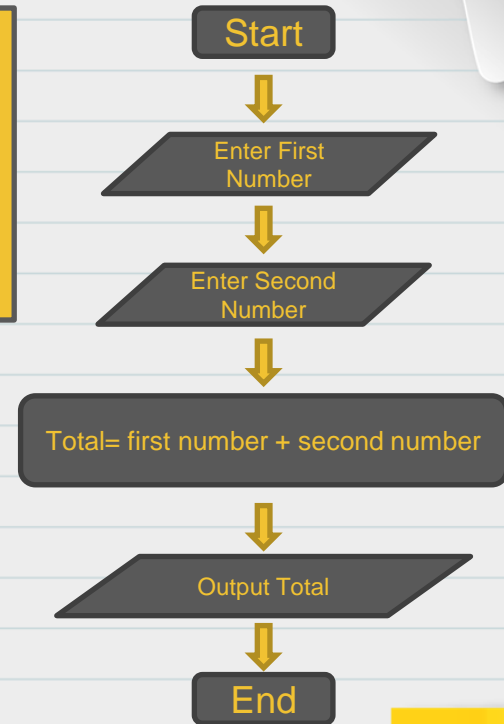
- Enter First number
- Enter second number
- Calculate total by adding two numbers
- Output total

Pseudo-code:

```
SEND 'enter first number' TO DISPLAY  
RECEIVE FIRST NUMBER FROM KEYBOARD  
SEND 'enter second number' TO DISPLAY  
RECEIVE SECOND NUMBER FROM KEYBOARD  
SET total TO FirstNumber+SecondNumber  
SEND TOTAL TO DISPLAY
```

The pseudo-code spells out the step-by-step instructions that the computer will be expected to carry out. It also introduces some important programming concepts.

- The numbers entered by the user are stored in two variables with the identifiers 'firstNumber' and 'second Number'.
- The result of adding the numbers together is stored in the variable 'total'.
- When some text is to be displayed, for example 'Please enter the first number.' it has to be enclosed in quotation marks, either single or double.
- When a variable is to be displayed, the quotation marks are not used. If they were, then, in the last instruction, the word 'total' would be displayed and not the number it represents.
- Arithmetic operators are used to perform calculations. This box shows the arithmetic operators.



Arithmetic Operators

Operator	Mathematical function	Example
+	Addition	$1+2=3$
-	Subtraction	$4-1=3$
^	Exponential	$2^4=16$
*	Multiplication	$4*3=12$, $(5*6)2=60$
/	Division	$12/4=3$
Mod	Modulus (return the remainder from an integer division)	$15 \text{ Mod } 4=3$ $255 \text{ mod } 10=5$
\	Integer Division (discards the decimal places)	$19\backslash 4=4$

Variable: A 'container' used to store data. The data stored in a variable is referred to as a value. The value stored in a variable is not fixed. The same variable can store different values during the course of a program and each time a program is run.

Identifier: a unique name given to a variable or a constant. Using descriptive names for variables makes code much easier to read.

Arithmetic operator: an operator that performs a calculation on two numbers.

Variables: Variables play an important role in algorithms and programming. The value stored by a variable can change as a program is running. Variables are extremely useful in programming because they make it possible for the same program to process different sets of data.

Constant: Constant is the opposite of a variable. It is a 'container' that holds a value that always stays the same. Constants are useful for storing fixed information, such as the value of pi, the number of liters in a gallon or the number of months in a year.

Variables: Variables play an important role in algorithms and programming. The value stored by a variable can change as a program is running. Variables are extremely useful in programming because they make it possible for the same program to process different sets of data.

Constant: Constant is the opposite of a variable. It is a 'container' that holds a value that always stays the same. Constants are useful for storing fixed information, such as the value of pi, the number of liters in a gallon or the number of months in a year.

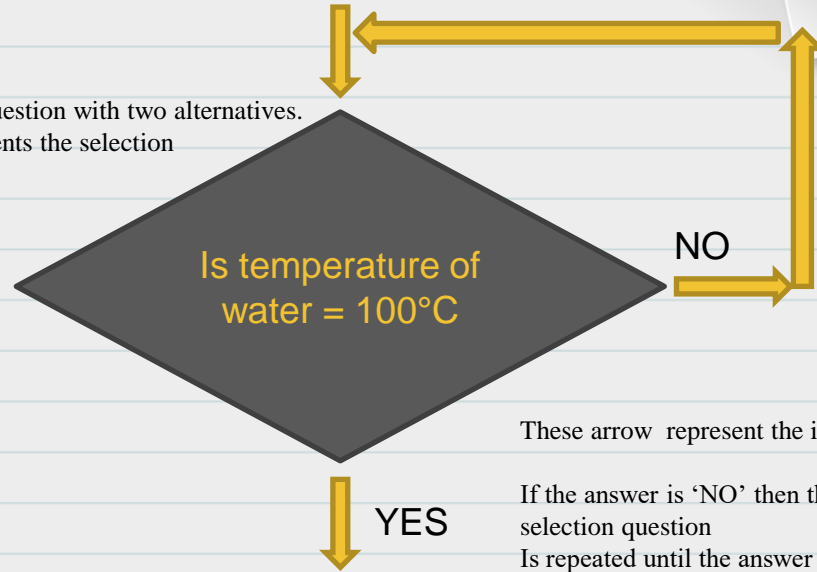
Construct: Construct a component from which something is built. Letters and numbers (.e. a to z and 0 to 9) are the constructs we use to build our language and convey meaning.

Selection: Selection is a construct that allows a choice to be made between different alternatives.

Iteration: a construct that means the repetition of a process. An action is repeated until there is a desired outcome or a condition is met. It is often referred to as a loop

Representation of selection & iteration in flowchart

There is a question with two alternatives.
This represents the selection



These arrow represent the iteration.

If the answer is 'NO' then the selection question is repeated until the answer is 'yes' the desired outcome.

Fig: Selection and iteration in flowchart

Representation of selection & iteration in Pseudo-code

Selection in pseudo-code is represented exactly using an IF.....THEN.....ELSE Statement.

IF.....THEN.....ELSE STATEMENT

```
IF temperature = 100 °C THEN
    Switch off kettle
ELSE
    Keep waiting
END IF
```

IF.....THEN STATEMENT

```
IF score >= 90 THEN
    SEND 'Excellent' TO DISPLAY
END IF
```

Relational operator is an operator that compares two values.

Relational Operators:

= (equal to)
> (greater than)
>= greater than or equal to
< less than
<= less than or equal to
<> Not equal to

```
SET passmark TO 75
  RECEIVE mark FROM KEYBOARD
IF mark >= passmark THEN
  SEND 'Well done. You've passed.' TO DISPLAY
ELSE
  SEND 'bad luck. You've failed.' TO DISPLAY
END IF
```


Nested Selection

The IF...THEN....ELSE statement allows a choice to be made between two possible alternatives. Sometime there are more than two possibilities. This is where a nested IF statement comes to use.

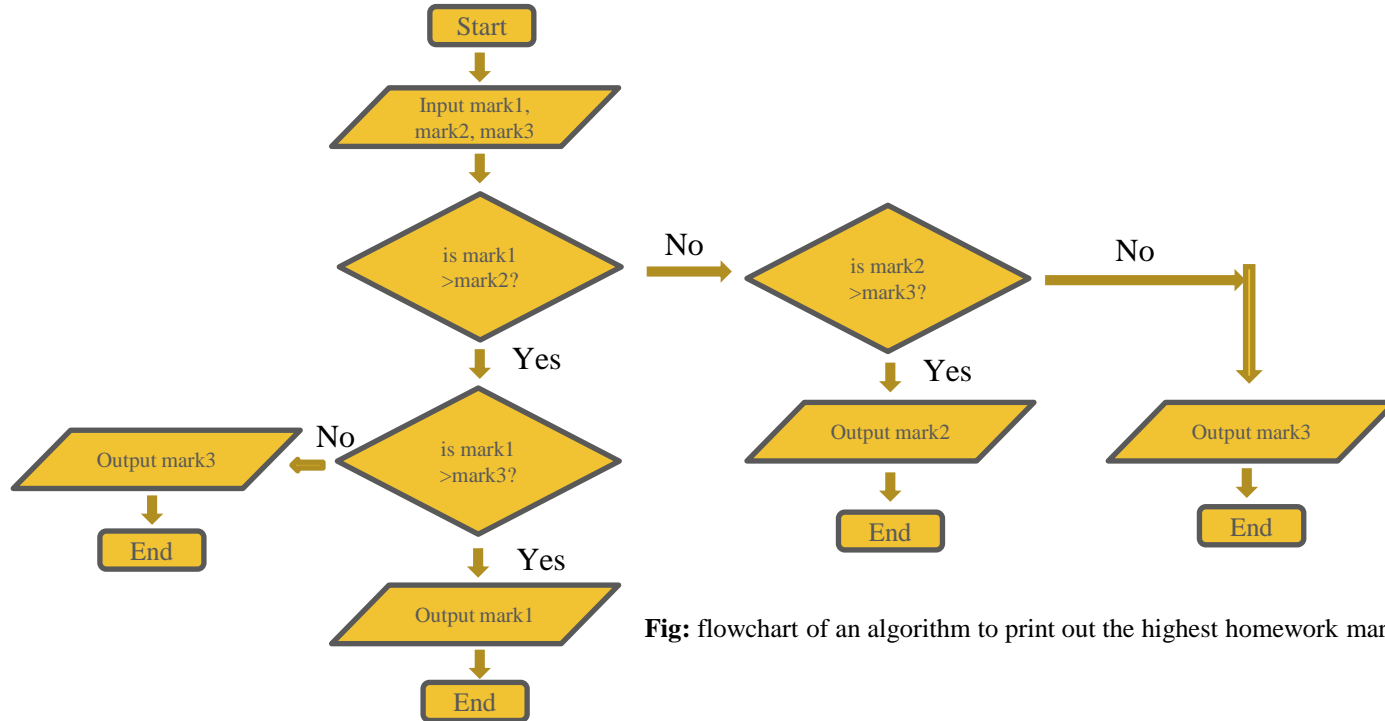


Fig: flowchart of an algorithm to print out the highest homework mark

Nested Selection pseudo-code:

The variables mark1 and mark2 are compared using a relational operator. If mark1 is greater than (>) mark2 then it is compared with mark3. If it is not greater than mark2, then mark2 must be greater than mark1 and it is then compared with mark3. This algorithm can also be expressed in pseudo-code:

```
RECEIVE mark1 FROM KEYBOARD
RECEIVE mark2 FROM KEYBOARD
RECEIVE mark3 FROM KEYBOARD
  IF mark1 > mark2 THEN
    IF mark1 > mark3 THEN
      SEND mark1 TO DISPLAY
    ELSE
      SEND mark3 TO DISPLAY
    END IF
  ELSE
    IF mark2 > mark3 THEN
      SEND mark2 TO DISPLAY
    ELSE
      SEND mark 3 TO DISPLAY
    END IF
  END IF
```

Fig: Pseudo-code of expressed algorithm

Indefinite iteration

Obviously in this example it is not known how many times the program will have to check until the water temperature reaches 100°C. Therefore we have to use indefinite iteration. There are two ways of doing this in pseudo-code: you can use a **REPEAT...UNTIL** loop or a **WHILE...DO** loop.

A **REPEAT...UNTIL** loop checks the condition when it gets to the end of the loop. This means that the statements contained within the loop will be executed at least once. A **WHILE... DO** loop checks the condition at the start of the loop so in some circumstances the statements contained within the loop will not be executed.

Use of REPEAT....UNTIL

```
REPEAT
  RECEIVE temp FROM SENSOR
UNTIL temp = 100
Switch off kettle
```

Use of WHILE.....DO

```
RECEIVE temp FROM SENSOR
WHILE temp <100 DO
  RECEIVE temp FROM SENSOR
END WHILE
Switch off kettle
```

Definite iteration

This is used when the number of iterations, or turns of the loop, is known in advance.

One method is to use a REPEAT.....TIME loop.

A learner is designing a program to help younger children with their times table. When a user enters a number the program will output the times table up to 12.

Use of REPEAT.....TIME

```
RECEIVE number FROM KEYBOARD
FOR index FROM 1 TO 12 DO
  SEND number * index TO DISPLAY
END FOR
```

Logical Operators

A logical operator is a symbol or word used to connect two or more expressions such that the value of the compound expression produced depends only on that of the original expressions and on the meaning of the operator. Common logical operators include AND, OR, and NOT.

- ❑ **AND:** If two conditions are joined by the 'AND' operator, then they must both be true for the whole statement to be true.
- ❑ **OR:** If two conditions are joined by the 'OR' operator, then either one must be true for the whole statement to be true.
- ❑ **NOT:** NOT The NOT operator reverses the logic of the AND and OR statements. The statement 'IF A=3 AND B=6' will be true only if the conditions are met, i.e. A and B are both equal to the values stated. The statement IF NOT (A = 3 AND B=6) will be true whenever both A and B are NOT equal to the values stated i.e. either or both are not equal to those values. The NOT operator is often used when setting a selection criterion, for example IF NOT LENGTH(password) > 8 THEN to ensure a minimum password length.

NESTED LOOPS

A nested loop comprises a loop within a loop. When one loop is nested within another, each iteration of the outer loop causes the inner loop to be executed until completion.

In this example a nested loop is used to calculate and display the average mark achieved by each student in a group of twenty for a series of five tests. The outer loop iterates through each student in turn. The inner loop receives each set of five marks and adds them together. The outer loop calculates and displays the average mark before moving on to the next student.

Random Number

A number within a given range of numbers that is generated in such a way that each number in the range has an equal chance of occurring. There are many devices for generating random numbers. A die is used in games to get a random number from 1 to 6. Computer programming languages have a function for generating random numbers across variable ranges.

Thanks!

