Algorithm:
A step-by-step procedure for solving a problem in a finite amount of time.
Algorithms can be represented using Flow Charts.

**CHARACTERISTICS OF AN ALGORITHM:**

1. Algorithms always have a definite starting point and an end point. These points are generally marked with the words like Start, Begin, End, Stop etc.
2. They consist of finite number of steps.
3. They always relate to a specific problem or you can say that they are written for a given problem.
4. They serve as foundation stone for programming.
5. They are written in easy language.

---

**Example 1:** Algorithm for converting from Celsius to Fahrenheit can be represented as:

1. \[ F = \left(\frac{9}{5}\right) C + 32 \]  
   (algebraic equation)
2. “Multiply the temperature reading in Celsius by 9/5 and then add 32 to the product”.

---

**Example 2:** Find minimum in a set of numbers

1. Take a look at the first number; memorize it
2. Take a look at the next number
3. Is it less than the number we just memorized?
   - **Yes:** Memorize this one instead
   - **No:** Ignore this number
4. Are more numbers are left?
   - **Yes:** Goto step 2
   - **No:** Read out the memorized number. That’s the required minimum.
Example 3.
Write algorithm for swapping contents of two variables.

Algorithm:

1. Start
2. Read value of A
3. Read value of B
4. C=A
5. A=B
6. B=C
7. Print A
8. Print B
9. End

This algorithm makes use of intermediate variable C. It holds value of A in it, temporarily and later assigns it to B variable. Thus values of variable A and B get interchanged. There is another way of interchanging the values i.e. without making use of intermediate variable. Algorithm for it is given below.

Algorithm:

1. Start
2. Read value of A and B
3. A=A+B
4. B=A-B
5. A=A-B
6. Print A, B
7. End
Problem 4.
Write an algorithm to find the largest of given three numbers.

Algorithm:

1. Start
2. Read three numbers A, B and C
3. Let Big=0
4. IF A>B Then Big=A Else Big=B
5. IF C>Big Then Big=C
6. Print Big
7. End

Problem 5.
Write an algorithm for calculating and printing factorial (!) of a given number.

Algorithm:

1. Start
2. Read a number, A
3. Let I=1
4. Let Sum=1
5. Sum=Sum*I
6. I=I+1
7. If I is not > A perform step 5, 6 and 7
8. Print Sum
9. End
Problem 6.
Write an algorithm for solving a given quadratic equation, \( ax^2 + bx + c = 0 \). Note that roots are determined by following formula:

\[
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\]

Algorithm:

1. Start
2. Read value of \( a \), \( b \) and \( c \)
3. If \( a=0 \) Stop
4. Calculate values of discriminant \( D = b^2 - 4ac \)
5. If \( D=0 \) then there is one root \( p = \frac{-b}{2a} \)
6. If \( D>0 \) then there are two real roots
7. If \( D<0 \) then there are two complex roots
   \[
   p = \frac{b + \sqrt{D}}{2a} \quad \text{and} \quad q = \frac{-b - \sqrt{D}}{2a}
   \]
8. Print \( p \) and \( q \)
9. Stop

PROBLEMS WITH ALGORITHMS:

As mentioned earlier, algorithms are written in English like sentences. Sentences are always subject to misinterpretation. If the sentence is complex, different readers may interpret it differently. This definitely leads to problems at the time of coding. To overcome this problem, often the solution of the problem is provided in pictorial form. Pictures carry more meaning and don't lead to ambiguity. Such type of step-by-step solution, provided in pictorial form is called flowchart.
FLOW CHARTS

When a step-by-step solution of a given problem is illustrated in the form of graphical chart that chart is called flowchart.

FLOWCHARTS SYMBOLS

Flowchart is a universal tool. A flowchart consists of special geometric symbols connected by arrows. Within each symbol is a phrase presenting the activity at that step. The shape of the symbol indicates the type of operation that is to occur. For instance, the parallelogram denotes input or output. The arrows connecting the symbols, called flowlines, show the progression in which the steps take place. Flowcharts should “flow” from the top of the page to the bottom.

A table of the flowchart symbols adopted by the American National Standards Institute (ANSI) follows.

The main advantage of using a flowchart to plan a task is that it provides a pictorial representation of the task, which makes the logic easier to follow. We can clearly see every step and how each step is connected to the next.

The major disadvantage with flowcharts is that when a program is very large, the flowcharts may continue for many pages, making them difficult to follow and modify.
**Condition Checking / Decision**

It is used for depicting comparison of two values or condition checking or decision-making etc. The condition that is to be checked is written within the figure. It is quite obvious that whenever a condition will be checked there will be two outcomes i.e. either the condition will be true (condition holds good) or it will be false (condition doesn't hold good). These two outcomes are shown as two branches coming out of the symbol. To clearly illustrate, which branch relates to true and which branch relates to false, the words Yes (or Y or True or T) and No (or N or False or F) are written near the branches.

**Processing**

Processing or calculation activities are depicted, using a rectangle. To make the process more illustrative, calculations that are being done are written within the rectangle.
**Start/End**
Like algorithm, flowchart must have a definite starting/Ending point. Starting/Ending point of the flowchart is depicted through a flat oval shape symbol. To make the symbol more illustrative, the word (Start, End) is written within the symbol.

![Start/End](image)

**Flow Lines**
Joining many symbols together makes a flowchart. Each symbol of the chart represents an activity. Which activity will be conducted after which activity, is depicted with help of flow lines. A flow line is a simple line with an arrow at its front end. The head of the arrow depicts the direction of flow. An up arrow is used for depicting a loop in the flowcharts.

![Flow Lines](image)

**Connectors**
Some of the flowcharts may turn out to be quite long. They may extend over many pages. Now the question is, how do you connect the processes, which are either far apart or are off the pages? For such type of requirements, two types of connectors are used. One is called same page connector and another is called "Off Page Connector".

![Connectors](image)
Predefined Process

If the problem to be solved is long then by showing all the steps in the same flowchart may make the chart complex. To avoid the complexity and keep the flowchart simple, you have to decompose the problem. In decomposition, you identify the group of mundane routines, such as initialization of variables etc. and the routines, which are repeatedly required. For example, printing the header lines, calculating a value based on certain formula etc. are the tasks that have to be performed repeatedly. You draw a flowchart for these processes separately and declare them as predefined processes and assign them name for reference. In the main flowchart, instead of writing all the steps, you simply include the symbol of predefined process along with the name. Whenever one wants to go into the details of predefined process, he refers corresponding chart separately.

![Flowchart Diagram]

Note that each process has a unique name. With the help of name you identify the process and refer it in the flowchart. In figure shown two predefined processes with the name “Ph” and “IV”.

From the description of flowchart, given above, it is quite clear that flowchart is nothing but a graphical representation of the solution for a given problem, in which you make use of standard graphical symbols and within the symbol, you write the details of the operation. For communicating the details of the activities of a particular step, you have to include some text within the symbol. For this, you have to often make use of different types of operators.
**Problem (1):**
Draw a flowchart for swapping the contents of two variables.
Flowchart for this problem is shown below fig (a).
Note that this flowchart makes use of intermediate variable, C to interchange the values of two variables A and B.
This flowchart can be redrawn to depict the method, which doesn't make use of intermediate variable. Shown in fig (b).

![Flowchart for swapping two variables](a)

**Problem (2)**
Draw a flowchart for solving a given quadratic equation $ax^2+bx+c=0$.

![Flowchart for solving quadratic equation](b)
PROBLEM (3)
Draw a flowchart to find the larger of the three given numbers.

PROBLEM (4)
Draw a flowchart for calculating and printing factorial of a given number.
Problem (5):
Calculate and report the grade-point average for a class.

The average grade equals the sum of all grades divided by the number of students. We need a loop to read and then add (accumulate) the grades for each student in the class. Inside the loop, we also need to total (count) the number of students in the class.

*Input:* Student grades

*Processing:* Find the sum of the grades; count the number of students; calculate average grade = sum of grades / number of students.

*Output:* Average grade