Steps of programming development

1. Problem definition and analysis
2. Program design
3. Coding
4. Compilation
5. Debugging and testing
6. Documentation
7. Implementation and Maintenance

1.Problem definition and analysis

The first phase in developing a program for a particular problem is to identify and define the problem. We analyze it by using the basic steps *i.e.* input, processing and output. Input is the data to solve the problem, processing is the steps to convert input to output and output is the results obtained after solving problem. Therefore the problem to be solved is defined by identifying it.

2. Program design

The second phase is program design. In this phase behaviour of the problem is to be understood in terms of the necessary inputs and the required outputs. Then the programmer designs a process through different techniques like algorithm or flowchart or pseudocode which will give the required outputs from the given inputs.

Algorithm

After properly defining the problem, a detailed, finite, step by step procedure for solving it must be developed called as an algorithm. Algorithm can be written in ordinary language that lies between ordinary language and programming language. It can be easily converted into a program in any programming language.

Efficient algorithm should consists of

- Each procedure should specify the inputs and outputs available.
- All the variables should be properly defined.
- The flow of the progam should be forward except for looping etc.
- Indents should be properly placed so that it can be easily understood.
- Documentation should be there but short and meaningful.
- Use subprograms or subroutines whenever required.

An algorithm is "a effective procedure for solving the problem in a finite number of steps."
It is effective, which means that an answer is found and it has a finite number of steps.

Characteristics of an algorithm

1. Input : There are zero or more values which are externally supplied as input.
2. Output : Atleast one value is produced as an output.
3. Definiteness : Each step must be clear and unambiguous.
4. Finiteness : An algorithm must terminate after a finite number of steps.
5. Effectiveness : Each step must be not only definite but also be feasible.

An algorithm can be stated using sequence, decision and repetition.

- **Sequence:** Sequence means that each step or process in the algorithm is executed in the specified order.

  An Algorithm to find the sum of any two numbers.

  1. Start
  2. Declare A,B,C
  3. Print "Enter two numbers"
  4. Input the value of A and B
  5. C = A+B
  6. Print C
  7. Stop

- **Decision:** In algorithms when the result is based on some condition and the outcome is either true or false and the flow of control has to move according to the decision.

  An algorithm that compare the two numbers and print the message.

  1. Start
  2. Print "Enter two Numbers"
  3. Input A and B
  4. If A > B then
     Print "A is Greater than B"
  5. If B > A then
     Print "B is Greater than A"
  6. If A = B then
  7. Print "Both are Equal"
  8. Stop.

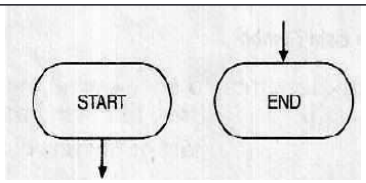
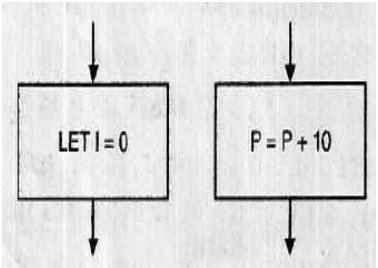
- **Repetition:** The Repeat loop is used to iterate or repeat a process or sequence of processes until some condition become true.

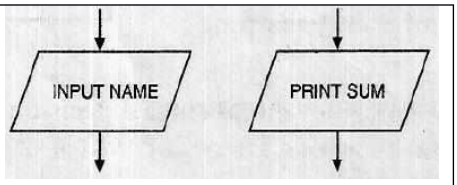
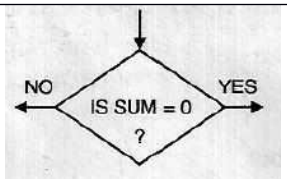
An algorithm for calculating in factorial of a given number N.

1. Start
2. Declare N,F,C
3. Print Enter number
4. Input N
5. F = 1
6. C= 1
7. F = F * C
8. C = C + 1
9. If C < = N then Goto step 7
10. Print F
11. Stop

Flowchart

Flowchart is the pictorial or graphical representation of the step by step procedure of a problem to be solved. A flowchart shows the actual flow of the logic of a program. It is made up of symbols for various types of operations used in the program.

S.no	Symbol	Use	Ex
	Oval or Terminal Symbol	Used as the first or last symbol in a program to start the flowchart or at end.	
	Rectangle or Process Symbol	Used to represent any kind of processing activity *i.e.* calculation, initialization	
		Used to enter data from	

	Parallelogram or Input/ output Symbol	keyboard or to print the results on screen.	
	 **Diamond box or Decision Symbol**	Used where a decision has to be made in selecting the different paths to be followed. In this there is only one entry point and at least two exits.	
	Arrow or flow data Symbol	Used to show the flow of data from one point to another	
	 **Circle or connector Symbol**	Used to enter from or exit to another part of the flowchart	

Advantage of using flowcharts

• ***Communication*** Flowchart are a better way of communicating the logic of a system to all concerned.

• ***Effective analysis*** With the help of flowcharts, problems can be analyzed more effectively.

• ***Proper documentation*** Program flowcharts serve as a good program documentation needed for various purposes.

• ***Efficient coding*** Flowcharts act as a guide or blueprint during the systems analysis and program development phase.

• ***Proper debugging*** Flowcharts help in the debugging process.

• ***Efficient program maintenance*** The maintenance of an operating program becomes easy with the help of a flowchart.

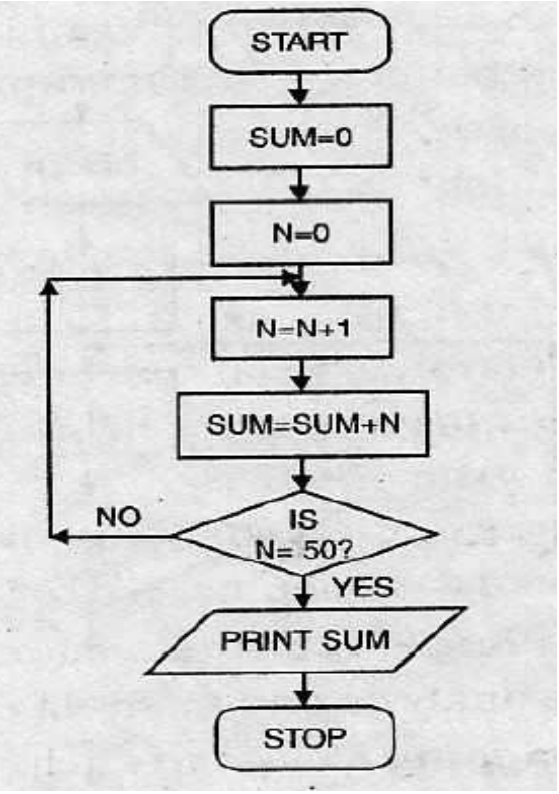
Limitations of using flowchart

• ***Complex logic*** Sometimes, the program logic is quite complicated. In such a case a flowchart becomes complex.

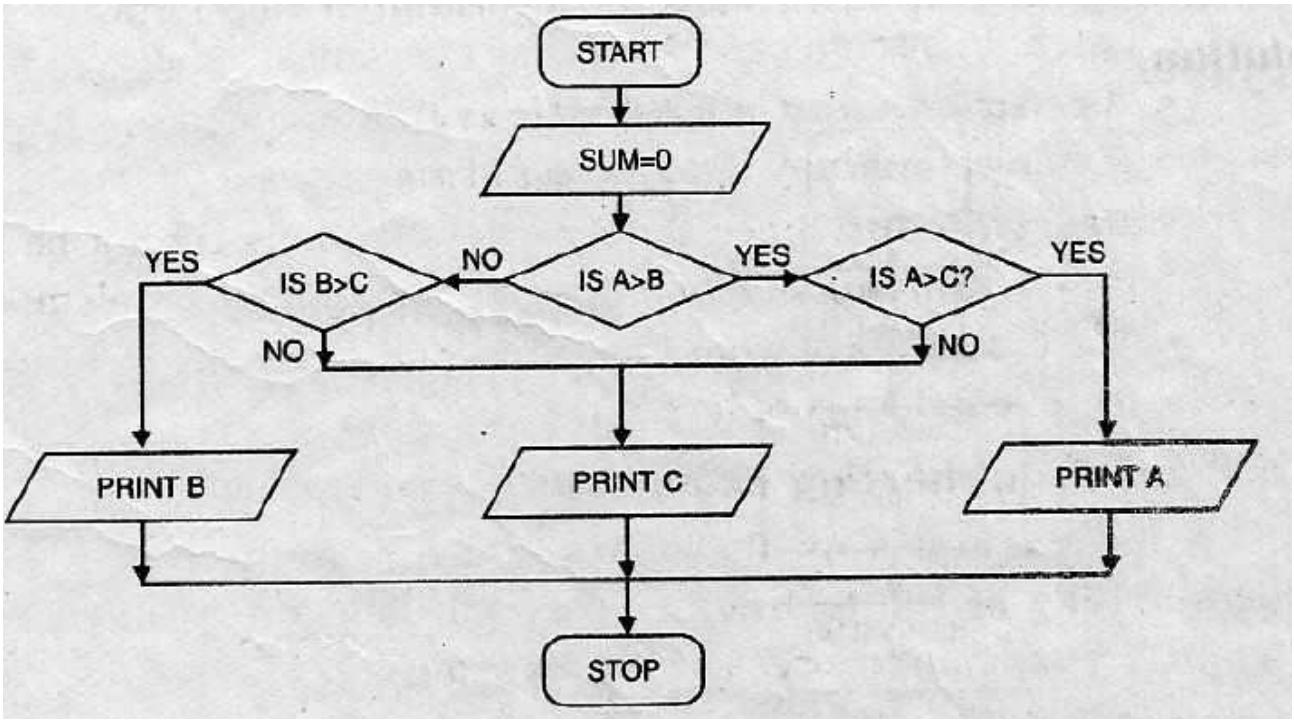
• ***Alterations and modifications*** If alterations are required, the flowchart may need to be redrawn completely.

• ***Reproduction*** Since the flowchart symbols cannot be typed in, the reproduction of a flowchart becomes a problem.

A flowchart to find the sum of the first 50 natural numbers.



A flowchart to find the largest of three numbers A, B, and C.



Tools for design a source code algorithm/flowchart/ analyst

Pseudocode

It is a code which is written in english like form to solve a particular problem. It can have the same style and format of the language but ignores the most punctuation.

Ex:

Pseudocode for printing the bigger out of two numbers.

```
main( )
{
   integer n1, n2
   read nl, n2
   if n1 is bigger than n2
      then print n1
  else print n2
}
```

It is a convenient tool but it is time consuming and lengthy.

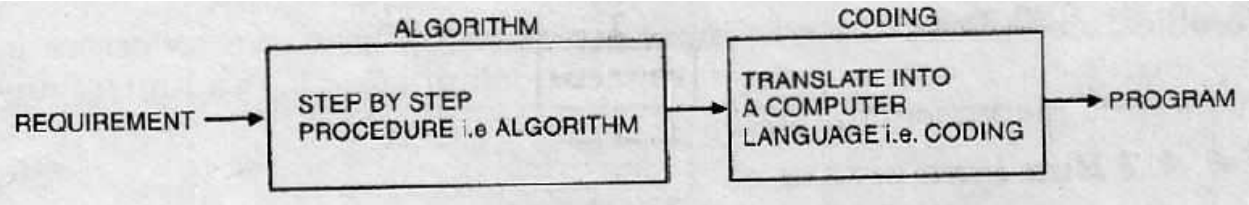
Advantages

1. It is very easy to convert a pseudocode into program of actual computer language.
2. In case of change in the logic of program, pseudocodes are easy to modify.
3. It is very convenient tool.

Disadvantages

1. With pseudocodes there is no graphical representation.
2. There are no standard rules to follows in using pseudocodes.
3. Pseudocode is difficult rules to understand for a beginner and is time consuming.

3. Coding: The third phase is to translates the steps of algorithm or flowchart in a computer program using a language such as C and is called as coding.



Source code is ready to enter it into computer and store it in a disk file with extension .C.

4 Compilation of the program: After coding phase, the next phase is to translate the program into machine language. This translation can be done using a C compiler. C compiler has in-built pre-processor which is used to process the source code before being passed to the compiler for compilation. **Preprocessor** directives are used to tell the pre-processor how to process the source code. It processes the source code and gives as a result the expanded version of source code. C compiler takes this expanded version of source code as an input and produces an object code *i.e.* machine code of the program with the same name and .obj extension.

There may be some type of errors in your program and the compiler prints the appropriate messages for the errors.

Type of errors:

1 Compilation time errors :Compilation time errors abort the compilation process.No object code is produced. These errors result from syntax errors in the source code. *e.g.* missing statement delimiters.

2 Run time errors: Run time errors may be caused due to a number of factors like incorrect input, syntax errors, incorrect control flow, data type mismatch etc.

3 Logical errors: Logical errors are the errors which are caused due to wrong program design and can not be detected by compiler. They can be

detected by watching the printed output *i.e.* we are not getting the results which
are desired.

4 Syntax errors: Syntax errors occur when a program violates the rules of grammar and expression of a computer language. Syntax errors are detected by system software *e.g.* compilers. Syntax errors are common errors but they are very easy to find out.

5. Debugging and testing: A bug is an error in a computer program. The process of eliminating the errors in the program is called as "Debugging". There are various categories of errors in a program. The program can be tested using dummy data or real data and the output is correct then the program is tested and the program is found to be correct.

6. Documentation: In this phase each instruction of the program and its purpose is written in easily understandable form so that a lay man can also understand the purpose of the program's instructions. This phase is also used to
maintain and upgrade the program in future.

7. Implementation and Maintenance: In this phase developed program is installed at the site of the user. And at regular interval maintenance of the program is also provided so that objective of the program is best utilized and fulfilled.