Program development cycle: -

- Problem analysis
- Task analysis
- Algorithm development
- Algorithm testing
- Coding
- Testing and debugging
- Documentation
- Implementation
- Maintenance and enhancement

Problem analysis: -

The developer wants to analyse the program precisely and completely. Based on understanding, the developer wants to know about which problem needs to be developed.

Task analysis: -

After developing the problem, the developer wants to develop various solution for the given problem. From these solutions, the optimum solutions are chosen. Which can solve the problem economically and comfortably.

Define algorithm: -

An algorithm defines as a finite sequence of an explicit instructions that when provided with a set of input values produces an output then terminates.

Advantages.

1) It is a step-by-step procedure of a solution to a given problem, and it is easy to understand.
2) It is easy to develop an algorithm and it is converted into a flowchart and into a computer program.
3) It has got a definite procedure.
4) It is easy to debug as every step got its own logical sequence.

Disadvantages: -

1) Creating an algorithm is time consuming and complex.

Define flowchart: -

A graphical representation of a programme is called flowchart.

6 symbols of flowchart: -

Flow lines - it is used to connect symbols. And it is used for indicate the directions.

Terminal (START - STOP): - This is used to represent the start and end of the flow chart.

Input/output: -

List and explain the features of c: -

- C is a robust
- Middle-level language combining features of high-level and low-level languages.
- It is highly suited for system software and application packages.
- It has rich data types and powerful operators
- It is platform-independent and highly portable.
- It allows pointer arithmetic and manipulation and expression compactness.
- Its library can be extended with user-defined functions.
- It provides manipulation of processor registers.

Give Structure of C program and explain each section: -

Documentation section: -

It consists of comment lines, programmer name, author, time & date. it gives an overview of a program.

Link section: -

Link section has header files of functions used in program, providing instructions to compiler to link functions from system library.

Definition section: -

All the symbolic constants are written in definition section. Macros are known as symbolic constants.

Global Declaration section: -

The global variables that can be used anywhere in the program are declared in global declaration section. This section also declares the user defined functions.

main() Function Section: -

Every C program must have one main() function section, which contains two parts: declaration and executable. In the declaration part, all variables used in the executable part are declared. The opening and closing braces must be used to signify the beginning and end of the main() section. Each statement in the main() section must end with a semicolon (;). Execution of the program starts at the opening brace and ends at the closing brace.

Subprogram Section: -

The subprogram section contains all the user-defined functions needed to perform a task. These functions are called in the main() function.

Write a note on Tokens in 'C': -

The smallest individual elements or units in a program are called as Tokens. C has following tokens. Identifiers, Keywords, Constants, Operators, Special character.

Identifiers: -

Identifiers refer to the name given to the programming elements such as variables, functions, arrays, etc...

Constants: -

Constants like regular variables except can't be modified after definition.

Variables: -

Variables are identifiers whose value can change during program execution. Data types decide what values can be stored in variables, which must be declared before use in C.

Constants: -

Constants refer to fixed values that can't be changed during execution. Types include int, float, char, string, and Enum. Constants are like variables but cannot be modified. There are four types of constants in C:

1) Integer Constants.
2) Real Constants.
3) Character Constants.
4) String Literals.

Explain in brief classification of a) integer types and b) floating point types c) Character data types: -

OR

List and explain primary data types available in 'C': -

Integer types:-

Integer is a data type in C programming language which represents whole numbers (no fractional or decimal part). It can represent values in the range of -32768 to 32767. The size of an integer is platform-dependent, although a maximum value of about two billion is common.
There are 3 types of integer data types:

1) Short int: It is a 16-bit signed integer.
2) int: It is a 32-bit signed integer.
3) Long int: It is a 64-bit signed integer.

Floating point types: -

Floating point numbers require 32 bits (float) or 64 bits (double) for precision up to 6 or 14 digits. Long double uses 80 bits for extended precision. Range of values for double is 1.7E-308 to 1.7E+308.

Character point types: -

Keyword 'char' defines a character data type. 8 bits of storage needed. Qualifier 'signed' or 'unsigned' can be used for char. Unsigned chars 0-255, signed chars -128 to 127.

Explain with examples declaring, initializing and assigning value to variable: -

Declaration of variable: -

Declare variables in C using:
data_type variable_name; (or data_type variable1, variable2,...,variablen);

```
e.g.    int a;
        float variable;
        float a, b;
```

Explain the any 4 advantages of flowchart.

1) **Clarify complex processes:** Flow charts provide a visual representation of the steps in a process. This makes it easier to understand complex processes and provides a clear structure.
2) **Identify problems:** Flow charts help identify areas of a process that may be inefficient or require improvement. This can be done by clearly mapping out the steps of a process and looking for any inconsistencies or areas that could be improved.
3) **Document processes:** Flow charts are an effective way to document processes so they can be easily understood and followed by others. This is especially helpful when training new employees.
4) **Increase productivity:** Flow charts can be used to identify and eliminate any costly or time-consuming steps in a process. This can help increase efficiency and productivity.

. Explain the Limitations of Flowcharts:-

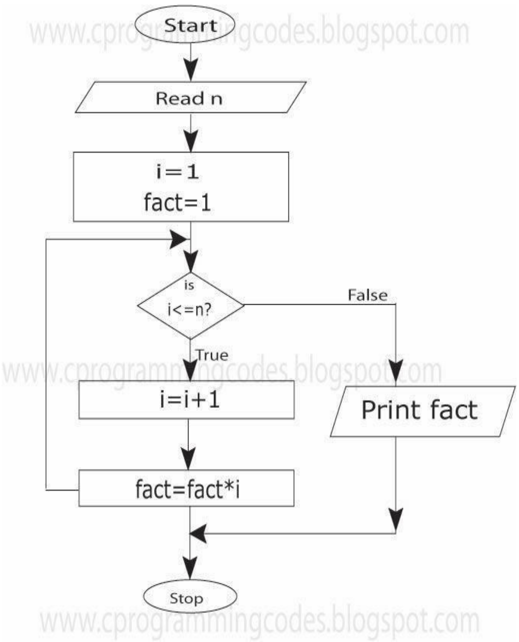
Complex: Flowcharts can become very long and hard to understand when a program is large. They can be time-consuming to create for large programs with many statements.

Costly: Drawing flowcharts can help you solve straightforward logic-based problems quickly. But if you have to create flowcharts for a large application, it could take a lot of time and money, making it expensive.

Difficult to modify: Modifying a flowchart is difficult because it requires redrawing the entire logic. This can be especially challenging if the program is large and complex, as it requires drawing thousands of flow lines and symbols with proper spacing.

No update: Programs are usually updated frequently, but large programs may not have their flowcharts updated to match. This inconsistency can make it difficult to understand the program's logic, which is the main purpose of flowcharts.

Draw the flowchart to find factorial of n number: -



UNIT 2

List and explain arithmetic, logical operators available in C: -

Arithmetic operators: -

Arithmetic operators in C are used to perform basic mathematical operations on variables. The following are the arithmetic operators available in C:

- Addition (+): Adds two operands
- Subtraction (-): Subtracts one operand from another
- Multiplication (*): Multiplies two operands
- Division (/): Divides one operand by another
- Modulus (%) : Returns the remainder when one operand is divided by another
- Increment (++): Increases the value of an operand by one
- Decrement (--): Decreases the value of an operand by one

Logical operators: -

Logical operators in C are operators that perform logical operations and return a Boolean value (true or false). The logical operators available in C are:

1) Logical AND (&&) - Returns true if both operands are true; returns false if either operand is false.
2) Logical OR (||) - Returns true if either operand is true; returns false if both operands are false.
3) Logical NOT (!) - Returns true if the operand is false; returns false if the operand is true.

List and explain a) conditional operator, b) increment & decrement operators available in C: -

Conditional operators: -

In C, the "?:" operator pair is used to create conditional expressions. Syntax: expr?expr1:expr2; If expr is true, the value of the expression is expr1. If expr is false, the value of the expression is expr2. These operators are also known as ternary operators.

Increment & decrement operators: -