**Fundamentals of C Language**

**Structure of a C program:**

C language does not actually follow any specific method of writing a program. It is a case sensitive language. All the statements must be written in lower case letters (small latters). The structure of a C program is as follows:

< Header files >

< Global declaration of variables >

Main ( )

{

 < Local declaration of variables >

- - - - - - - - - - -

< Statements >

- - - - - - - - - - -

}

< sub programs - function blocks >

Q: What are the basic steps involved in writing a computer program?

Writing of programs may involved the following steps:

* 1. Understanding the problem.
	2. Planning the method of solution.
	3. Development of the methods by using suitable algorithms, flowcharts, etc.
	4. Coding the instructions in a programming language.
	5. Transforming the instructions into machine readable form (using an input medium)
	6. Program testing and debugging.
	7. Documentation of the works involved in the production of the program.

**What is a Pseudo code:**

Pseudo code consists of English-like statements describing an algorithm. It is written using simple phrases and avoids cryptic symbols. It is independent of high level languages and is a very good means of expressing an algorithm. It is written in a structured manner and indentation is used to increase clarity. As an example, the following algorithm finds and returns the maximum of n given numbers:

1 Algorithm Max(A,n)

2 /\* A is an array of size n with index starts from 1 \*/

3 {

4 Result = A[1];

5 for i= 2 to n do

6 if A[i] > Result then Result := A[i];

1. return Result

8 }

**C-preprocessor:**

The C-preprocessor is a collection of special statements, called directives, which are executed at the beginning of the program compilation. The commands for the preprocessor are inserted in C source-code that is (.c) files and called compiler directives. Each compiler directive is prefixed by a hash sign (#).

Examples of preprocessor directives are #include, #define, #if, #elif, #endif, #undef etc.

**#define (Macro Directive):** ‘C’ allows defining an identifier having constant value using #define directive. This is called a preprocessor directive as if it is not part of a C program. This directive is placed at the beginning of a C program. The symbol # occurs in the first column and no semicolon is allowed at the end.

For example,

 #define PI 3.14

 #define MAXIMUM 100

**#include (File Directive):** #include is also a processor directive. This directive causes one file to be included in another for example #include<stdio.h> which appears at the start of the program.

**Library function:**

Library functions are prewritten routines that carry out various commonly used operations or calculations. They are contained within one or more library files that accompany each C compiler.

Some library functions are described below:

**fclose( ):** The function fclose( ) closes a stream. The fclose( ) function returns 0 if the close operation is successful, otherwise it returns -1.

**printf( ):** The printf( ) function writes formatted output to stdout i.e. standard output device.

**strcpy( ):** strcpy( ) function copies one string to another string.

**&&&**