PPS LAB VIVA QUESTION AND ANSWERS

1) What is C language?
Ans: C is a programming language developed at AT & T's Bell Laboratories of USA in 1972. The C programming language is a standardized programming language developed in the early 1970s by Ken Thompson and Dennis Ritchie for use on the UNIX operating system. It has since spread to many other operating systems, and is one of the most widely used programming languages.

2) What is an algorithm?
Ans: An algorithm is a step-by-step method of performing any task.

3) What is a flow chart?
Ans: A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows.

4) What is a C Preprocessor?
Ans: C Preprocessor is a program that processes our source program before it is passed to the compiler.

5) What is the use of header files as used in C programming?
Ans: Header files are used to have declarations. It is simple to include a single header file than writing all the needed functions prototypes.

6) What is the Structure of a C Program?
Ans: Documentation Section, Linking Section, Definition Section, Global declaration Section, main function, subprogram section.

8) What is the use of main() function?
Ans: main() is the starting point of program execution.

9) What are the types of constants in c?
Ans: C constants can ba divided into two categories:
1) Primary constants (Numerical)
2) Secondary constants (Character)

10) What is a Compiler?
Ans: A **compiler** is a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code).

11) What is a Translator?
Ans: A **translator** is a computer program that translates a program written in a given programming language into a functionally equivalent program in a different language.

12) What is a Interpreter?
Ans: An Interpreter is a computer program that directly executes, i.e. *performs*, instructions written in a programming or scripting language, without previously batch- compiling them into machine language.

MD AFZAL
Asst. Professor
Mob: 8179700193
mdafzal.mbnr@gmail.com

13) What is a Token in C?
Ans: A Token is the basic building block of a **C. (or)** the basic element recognized by the compiler is the "**token**."
C Tokens are: Key Words, Identifier, Constants, String – literal, Operator, Punctuators

14) What are Printf() and scanf() Functions :
Ans: printf() and scanf() functions are inbuilt library functions in C which are available in which are available in " stdio.h" header file.

- printf() function is used to print the "character, string, float, integer, octal and hexadecimal values" onto the output screen.
- scanf() function is used to read character, string, numeric data from keyboard

15) What is a Data Type and List the different Data types?
Ans: C data types are defined as the data storage format that a variable can store a data to perform a specific operation.
List of Data Types:

1) Basic Data Types: Int, Float, Char, Double, long int
2) Enumeration Data Type: enum
3) Derived Data Type: Pointer, array, structure, union
4) Void Data Type: void

18) What is a Void Data Type?
Ans: Void is an empty data type that has no value.

19) What is a comment in C?
Ans: Comments are like helping text in your C program and they are ignored by the compiler. We can write in between /* and */ or // (Line Comments).

20) What is an Identifier in C?
Ans: A C identifier is a name used to identify a variable, function, or any other user- defined item. An identifier starts with a letter A to Z or a to z or an underscore _ followed by zero or more letters, underscores, and digits (0 to 9). C does not allow punctuation characters such as @, $, and % within identifiers. C is a **case sensitive** programming language.

21) What is a Key word in C?
Ans: Keywords are reserved words in C and Keywords are may not be used as constant or variable or any other identifier names.

23) Define a Variable?
Ans: A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in C has a specific type, which determines the size and layout of the variable's memory.
Syntax for variable Declaration:

```
DataType VariableList;
int i,j;
cahr c, ch;
```

24) What are the steps to develop a C Program?

Ans:

1) Specifying the problem statement
2) Designing an algorithm
3) Coding
4) Debugging
5) Testing and Validating
6) Documentation and Maintenance.

25) What is the process of debugging or compilation for C Programs?

Ans:

To check the errors in a program is called debugging or compilation. This process done by 3 stage:

1) **Checking Syntactic Errors:** These errors occur due to the usage of wrong syntax for the statements. (Short cut Key : ALT+F9)
2) **Checking Run time Errors/ Linkage Errors:** These Errors are determined at the execution time of the program (EX: Divide a number by Zero, finding logarithm of negative number, finding square root of negative number ,etc.)
3) **Checking Linker Errors:** These errors are occurred due to the linking of header file to the current program (Short cut Key : CTRL+F9)
4) **Checking Logical Errors:** These Errors occur due to incorrect usage of the instruction in the program. Logical Errors are determined by analyzing the outputs for different possible inputs that can be applied to the program.

Difference between compiler and debugger:

A compiler converts programs written in a language a human being can (hopefully) understand into the only language a computer understands (binary code). So to execute your code, a compiler is the only thing you need.
A debugger on the other hand is a tool that helps you identify bugs (errors) in your program.

26) List some Syntactic Errors or Syntax Errors?

Ans:

1) Missing semicolon (Statement Missing ;)
2) Undeclared a variable name or Undefined symbol (Then check variable declaration syntax and check for header file for some keywords)
3) ')' expected (Then check the no of parenthesis opened and closed)
4) Illegal string constant (check for the last double quote in a string)
5) printf and scanf arguments should be placed in ( )
6) Compound statement missing (Check for the no of { and } are opened and closed)
7) Proto type missing error (Check for Function declaration statement)
8) Forgetting to put &,", and comma operator in specific places.
9) Comparing strings with == operator

27) List some Warnings?

Ans:

1) Miss use of = and ==
2) Loop has no body (remove the semicolon at last of the for loop or while loop)
3) Uninitialized a variable

28) What is an Operator and list different types of operators in C?
Ans: The symbols which are used to perform logical and mathematical operations in a C program are called C operators. The different operators in C are: Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment and Decrement operators, Conditional operators, Bitwise operators, Special operators.

29) What are Arithmetic Operators?
Ans: Arithmetic operators are used to perform mathematical calculations like addition, subtraction, multiplication, division and modulus in C programs. (Like +,-,*,/,%)

30) What are Assignment Operators?
Ans: Values for the variables are assigned using assignment operators. There are two types assignment operators are there:

1) **Simple Assignment**(=)
2) **Compound Assignment** (+= , -=, *=, %=)

31) What are Relational Operators?
Ans: Relational operators are used to find the relation between two variables. i.e. to compare the values of two variables in a C program.. (Like >, <, >=, <=, !=, ==)

32) What are Logical Operators?
Ans: These operators are used to perform logical operations on the given expressions. (Like &&, ||, !)

33) What are Bitwise Operators?
Ans: These operators are used to perform bit operations. Decimal values are converted into binary values which are the sequence of bits and bit wise operators work on these bits. (Like &, |, ~, ^, <<(Left Shift), >> (Right Shift))

34) What are Conditional (Ternary) Operators?
Ans: Conditional operators return one value if condition is true and returns another value is condition is false. (Like: ? and :)

35) What are Increment / Decrement Operators?
Ans: Increment operators are used to increase the value of the variable by one and decrement operators are used to decrease the value of the variable by one in C programs. (Like ++, –)

36) What are Special Operators?
Ans: &(Address Operator), * (Pointer Operator), and Sizeof()

37) What is sizeof operator?
Ans: It returns the number of bytes the operand occupies.

38) What is pre-increment or post-increment?
Ans: ++n (pre increment) increments n before its value is used in an assignment operation or any expression containing it. n++ (post increment) does increment after the value of n is used.

39) What is type casting?
Ans: Converting a variable of one type to another type.

40) What are the Format Specifiers or Type Specifiers or Conversion Specifiers?
Ans: %d (Integer), %f (Float), %c (Character), %l (Long Integer), %s (Strings), %u (Address with decimal value), %p (Address with Hexa Decimal Value in Small Letters), %x ((Address with Hexa Decimal Value in Capital Letters)

41) What is a Statement in C?
Ans: A statement is a block of code that does something.

42) Different Types of Statements?
Ans: Null Statement, Expression Statement, Compound Statement, Return Statement, Conditional Statements, Iterative or Looping Statements, Unconditional Statements.

43) Define Null Statement?
Ans: A "null statement" is a statement containing only a semicolon;

44) Define Expression Statement?
Ans: When an expression statement is executed, the expression is evaluated according to the rules outlined in Expressions and Assignments

45) Define Compound Statement?
Ans: A compound statement (also called a "block") typically appears as the body of another statement which is in between {and}

46) Define Return Statement?
Ans: The **return** statement terminates the execution of a function and returns control to the calling function. A **return** statement can also return a value to the calling function.

47) Define Conditional Statements and give the list of them?
Ans: Conditional Statements which allows to perform actions depending upon some conditions provided by the programmer. The Different types of conditional statements are:

1) If Statement
2) If else statement
3) Nested- if else statement
4) Switch Statement

48) Write the Syntax for IF Statement?
Ans:

```
if ( <expression> )
<statement>
```

49) Write the Syntax for IF- ELSE Statement?
Ans:

```
if ( <expression> )
<statement 1>
else
<statement 2>
```

50) Write the Syntax for NESTED- IF – ELSE Statement?
Ans:
```
if ( <expression1> )
{
        <Statements>
}
else if(<expression2>)
{
        <Statements>
}
else
{
        <Statements>
}
<statement>
```

51) Write the Syntax for Switch Case Statement?
Ans:
switch (<expression>) (<case list> }
where
<case list>
is a sequence of
case <value>: <statement list>
break;
and optionally one
default: <statement list>
break;

52) What are Iterative or Looping Statements?
Ans: Iterative or Looping statement which executes the statements within the compound statement by checking the condition, and performs same set of statements as a iterative process or as a loop until the condition false.
The Iterative or Looping Statements are: While, do-while, for

53) Write Syntax for WHILE Statement?
Ans:
```
        while ( <expression> )
        <statement>
```

54) Write Syntax for DO- WHILE Statement?
Ans:
```
        do <statement>
        while ( <expression> );
```

55) Write Syntax for FOR Statement?
Ans:
```
        for ( <Initialization>;<Condition>;<Increment/Decrement>)
        <statement>
```

56) What is the difference between for loop and while loop?
Ans: For Loop is used to execute a set of statements in fixed number of times. We use While loop when the number of iterations to be performed is not known in advance we use while loop.

57) What are Unconditional Statements?
Ans: goto and labeled statements, break Statement continue Statement.

58) Define goto and labeled Statement, Write syntax for goto and labeled statements?
Ans: The **goto** statement transfers control to a label. The given label must reside in the same function and can appear before only one statement in the same function.
Syntax:

goto <label>;
<label>: <Statement>;

59) Define Break Statement, Write syntax for Break statement?
Ans: The **break** statement terminates the execution of the nearest enclosing **do**, **for**, **switch**, or **while** statement in which it appears. Control passes to the statement that follows the terminated statement.
Syntax:

break;

60) Define Continue Statement, Write syntax for Continue statement?
Ans: The **continue** statement passes control to the next iteration of the nearest enclosing **do**, **for**, or **while** statement in which it appears
Syntax:

continue;

61) Define Type Qualifiers, and list them?
Ans: The keywords which are used to modify the properties of a variable are called type qualifiers. There are two types of qualifiers available in C language. They are,1) const 2) volatile.

62) Define const Keyword with Syntax?
Ans: Constants are also like normal variables. They refer to fixed values. They are also called as literals.
Syntax:

const data_type variable_name; (or) const data_type *variable_name;

63) Define volatile Keyword with Syntax?
Ans: When a variable is defined as volatile, the program may not change the value of the variable explicitly.
Syntax:

volatile data_type variable_name; (or) volatile data_type *variable_name;

Note: volatile specifies a variable whose value may be changed by processes outside the current program

64) What is a Macro?
Ans: Macros are the identifiers that represent statements or expressions. To associate meaningful identifiers with constants, keywords, and statements or expressions.

65) What is the difference between #include< > and #include " "?
Ans:

#include‹ ›
Specifically used for built in header files.

#include " "
Specifically used for used for user defined/created n header file.

66) Define Storage class with Syntax?
Ans: Storage class specifiers in C language tells the compiler where to store a variable, how to store the variable, what is the initial value of the variable and life time of the variable.
Syntax: storage_specifier data_type variable _name

68) Define Array?
Ans: C Array is a collection of variables belongings to the same data type. OR An Array is a collection of Homogeneous or similar data type elements having unique values, and stored ata different locations. You can store group of data of same data type in an array.

69) Types of an Array?
Ans:
There are 2 types of C arrays. They are,
1) One dimensional array
2) Multi-dimensional array
1. i) Two dimensional array
2. ii) Three dimensional array, four dimensional array etc…

70) What are the Characteristics of Arrays?
Ans: An array holds elements that have the same data type. Array elements are stored in subsequent memory locations
Two-dimensional array elements are stored row by row in subsequent memory locations. Array name represents the address of the starting element

73) How we can read or print the array elements?
Ans: To read or print the array elements we need to use FOR Statement.

74) Define a String?
Ans: C Strings are nothing but array of characters ended with null character ('\0'). Strings are always enclosed by double quotes. Whereas, character is enclosed by single quotes in C.
Example:

```
char string[20] = { 'H' , 'e' , 'l' , 'l' , 'o' , '\0'}; (or)
char string[20] = "Hello";              (or)
char string []  = "Hello";
```

75) Header used for String Functions?
Ans: #include<string.h>

76) List String Handling Functions or String Manipulation Functions?
Ans: strcat(), strncat(), strcmp(), strncmp(), strcmpi(), strcpy(), strncpy(), strlen(), strchr(), strrchr(), strstr(),strrstr(), strrev()

77) What is the difference between strings and arrays?
Ans: String is a sequence of characters ending with NULL .it can be treated as a one dimensional array of characters terminated by a NULL character.

78) Define Pointer with Syntax and example?
Ans: C Pointer is a variable that stores/points the address of another variable. C Pointer is used to allocate memory dynamically i.e. at run time.
Syntax: data_type* var_name;
Example: int* p;
char* p;

79) What are the uses of Pointers?
Ans:
- ✓ *Pointer is used in the following cases*
- ✓ *It is used to access array elements.*
- ✓ *It is used for dynamic memory allocation.*
- ✓ *It is used in Call by reference.*
- ✓ *It is used in data structures like trees, graph, linked list etc.*

80) What is the invalid pointer Arithmetic?
Ans:
- ✓ Adding, multiplying and dividing two pointers.
- ✓ Shifting or masking pointer.
- ✓ Addition of float or double to pointer
- ✓ Assignment of a pointer of one type to a pointer of another type

81) What is a pointer value and address?
Ans: A pointer value is a data object that refers to a memory location. Each memory location is numbered in the memory. The number attached to a memory location is called the address of the location.

82) How are Pointer Variables initialized?
Ans: Pointer variable are initialized in two ways:
1) Static memory allocation
2) Dynamic memory allocation

83) What is a pointer to pointer?
Ans: If a pointer variable points another pointer value. Such a situation is known as a pointer to a pointer.
Example:
int p1,*
*int** p2;*
v=10;
P1=&v;
p2=&p1;

Here p2 is a pointer to a pointer.

84) What are the advantages of using array of pointers to string instead of an array of strings?
Ans: Efficient use of memory.
Easier to exchange the strings by moving their pointers while sorting.

85) What are the pointer declarations used in C?
Ans:

- ✓ Array of pointers, e.g , int* a[10]; Array of pointers to integer
- ✓ Pointers to an array,e.g , int (*a)[10]; Pointer to an array of into Function returning a pointer,e.g, float *f( ) ;
- ✓ Function returning a pointer to float Pointer to a pointer ,e.g, int **x;
- ✓ Pointer to apointer to int pointer to a data type ,e.g, char *p; pointer to char

86) What are the Advantages of Functions?
Ans:

- ✓ It reduces the Complexity in a program by reducing the code.
- ✓ Function are easily understanding and reliability and execution is faster.
- ✓ It also reduces the Time to run a program.
- ✓ In other way, it's directly proportional to Complexity.
- ✓ It's easy to find-out the errors due to the blocks made as function definition outside the main function.

87) What is Recursion?
Ans: A recursion function is one which calls itself either directly or indirectly it must halt at a definite point to avoid infinite recursion.

88) What is the difference between malloc( ) and calloc( ) Functions?
Ans: Malloc is used for memory allocation and initialize garbage values.malloc () for allocating the single block of memory.
Syntax: *ptr-variable=(type-casting*)malloc(n*sizeof(DataType))
Example: *ptr=(int*)malloc(5*sizeof(int));
Calloc is same as malloc but it initialize 0 value.calloc () for allocating multiple blocks of memory.
Syntax: *ptr-variable=(type-casting*)calloc(n,sizeof(DataType))
Example: *ptr=(int*)calloc(5,sizeof(int));

89) What is the purpose of realloc?
Ans: It increases or decreases the size of dynamically allocated array. The function realloc (ptr,n) uses two arguments. The first argument ptr is a pointer to a block of memory for which the size is to be altered. The second argument specifies the new size. The size may be increased or decreased. If sufficient space is not available to the old region the function may create a new region.

90) What is an Argument?
Ans: An argument is an entity used to pass data from the calling to a called function.

91) What are Built-in-Functions/ Pre Defined Functions / Library Functions?
Ans: The functions that are predefined and supplied along with the compiler are known as built in functions. They are also known as library functions.

92) What are the uses of Functions?

Ans:

- ✓ C functions are used to avoid rewriting same logic/code again and again in a program.
- ✓ There is no limit in calling C functions to make use of same functionality wherever required.
- ✓ We can call functions any number of times in a program and from any place in a program.
- ✓ A large C program can easily be tracked when it is divided into functions.
- ✓ The core concept of C functions are, re-usability, dividing a big task into small pieces to achieve the functionality and to improve understandability of very large C programs.

94) Define Call by Value?

Ans: In call by value method, the value of the variable is passed to the function as parameter. The value of the actual parameter cannot be modified by formal parameter. Different Memory is allocated for both actual and formal parameters.

Actual parameter – This is the argument which is used in function call. Formal parameter – This is the argument which is used in function definition

95) Define Call by Reference?

Ans: In call by reference method, the address of the variable is passed to the function as parameter. The value of the actual parameter can be modified by formal parameter. Same memory is used for both actual and formal parameters since only address is used by both parameters.