C VIVA QUESTION AND ANSWERS

1. Who developed C language?
   Dennis Ritchie.
2. How do you make comments in C program?
   / Comments */*
3. What is preprocessor statement?
   Preprocessor statements start with #. They are
   1. Inclusion statements (#include <stdio.h>)
   2. Definition Statements (#define PI 3.14)
   3. Conditional compilation statements (#ifdef, #ifndef, #endif, #if, #else)
4. Differentiate between constant and variable.
   Constant : It's value will not change during the execution of the program while the variable might change during the execution of the program
5. What is data type?
   Data type refers to the different classes of data that can be manipulated by C.
6. Name the basic data types of C.
   int, float ,char, double, long
7. What is the range of integer, char, float for a 16-bit computer?
   int – 2 bytes, char - 1 byte, float – 4 bytes.
8. What is a header file?
9. Name any 3 header files?

10. What are qualifiers?
    Size Qualifier – long, short. Eg: long int.
    Sign Qualifier – signed, unsigned. Eg: unsigned int.
11. What is a statement?
    Statement is an instruction that can be an expression or function call in the source program.
12. What is a keyword?
    Keywords are reserved words which have predefined meaning in C. There are 32 keywords in C.
13. What is an identifier?
    Identifier is one which is used in a program to represent variables, constants, array names, function names.
14. What are escape sequences?
    To print special characters like newline, blank space, tab space escape sequences are used in C. Newline - \n tab - \t null character - \0
15. How do you classify C operators?
    Arithmetic, relational, bitwise, logical.
16. What is the use of modulus operator?
    Modulus operator '%' gives the remainder after the division operation is performed.
    5 / 2 = 2 (quotient) where as 5 % 2 = 1 (remainder)
17. What are bitwise operators?
    Bitwise Operators enable us to perform manipulations over data at bit level. They work only on integers. &, |, >>, <<, ^
18. What is unary operator?

It is defined over a single operand. Eg: address operator(&), a++, a—(Increment and Decrement operators), ! (Logical NOT operator)

19. What is binary operator?

*It is defined over 2 operands. Eg: Arithmetic operators(+, -, *, /, %), relational operators(>, < >=, =, !=) logical operator(&&, ||) and Bitwise operators(&, |)*

20. What is a conditional / ternary operator?

It is defined over 3 operands.
(Expression 1) ? (Expression 2) : (Expression 3);
Example : max = a > b ? a : b;

21. Differentiate between && and & operators.

&& - Logical AND operator.
& - Bitwise AND operator.

22. What are different format specifiers?

Format specifiers are used in printf() and scanf() statements to print/read different data types.
%d – integers %f -floats %c – characters %s – string
%ld – long integers %lf - double floats

23. Differentiate between pre-increment/decrement & post- increment/decrement.

Pre-increment/decrement: The variable is first incremented/decremented and then used in the expression.
Post-increment/decrement: The variable is first used in the expression and then incremented.
Pre or Post increment/decrement of a variable does not make any difference when used as independent statement.

24. Differentiate between Unformatted and formatted I/O statements.

Formatted: printf(), scanf() - Displays the output in more readable manner.
Unformatted: putchar(), puts() – Displays a characters or strings.

25. How do you classify the control structures/statements?

Control statements determine the 'flow of control' of the program.
1. *Sequence control statements*
2. *Selection control statements if, if/else, if/else if/else.*
3. *Looping control statements. For, while, do/while.*
4. *Case control statements switch*

26. Differentiate between while and do-while loop.

While(test) – The test expression is first evaluated before entering the body of the loop. If the test evaluates to false in the beginning itself, the body of the loop is not entered at all.
Do-while – The test expression is evaluated at the end. So the statements in the body of the loop are guaranteed to be executed at least once.

27. Differentiate between break and continue.

Break - causes premature exit of the loop. The control is transferred to the statement following the loop. It can be used in switch.
Continue – causes skipping of the statements following it in the body of the loop. Control is transferred back to the loop. It cannot be used in switch.

28. When do you prefer for loop statement?

For looping 'for' statement is preferred as it has all the three things: initialization, test-expression and updation in one single line.

29. What is looping?

The repetition of execution of a block of statements as long as some condition is true is called as looping.

30. What is an array?
Array is a group of logically related data items of similar type stored in contiguous memory locations sharing a common name but distinguished by subscript.
31. Differentiate between an array and an ordinary variable.
Variable – a named memory location.
Array – a collection of data items which are of same type and share a common name.
32. What are character arrays?
Character arrays are nothing but strings (Collection of characters).

33. When do you use two-dimensional character array?
When you want to store multiple rows of elements like matrix. If you want to store 3integer elements in 4 rows then you would declare a two-dimensional array:
Int array[4][3].
Array[0][0] would be
34. Name the different string handling functions?
strlen(), strcpy(), strrev(), strcat(),strcmp()
35. Differentiate between standard functions & user-defined functions?
Standard or built-in-functions are those which are available in C Library. They can be used by any programmer. Eg. Scanf(), printf(), strlen(), strcpy().
User-defined functions are those which are written by a user to solve a problem. Eg. main() is a user defined function.
36. What is iteration?
37. What is the difference between iteration and recursion?
38. What are arguments?
Arguments facilitates the data communication between calling function and called function.
Int sum(int a[], int n) Here a[] and n are arguments to the function sum. This function sum is returning an integer.
39. List different types of storage classes.
1. *Auto*
2. *Static*
3. *Extern / global*
4. *Register*
40. Differentiate between local and global variables.

	Local/Auto	***Global/Extern***
Default Value --	*Garbage Value*	*Zero*
Scope --	*It is visible in the function In which the variable is defined.*	*It is visible in all the functions of the program.*
Lifetime --	*It is destroyed when the Function exits.*	*It is alive during the lifetime Of the Program.*

41. Differentiate between local and static variables.

	Local/Auto	***Static***

Default Value --	*Garbage Value*	*Zero*
Scope --	*It is visible in the function In which the variable is defined.*	*It is visible in the function in which the variable is defined.*
Lifetime --	*It is destroyed when the Function exits.*	*The value of the static variable is retained on each entry to the function.*

42. Name the different methods of parameter passing?
    Call by value and Call by reference.
43. What do you understand from function prototype?
    Function prototype tells the compiler about
    1. The name of the function
    2. Return type
    3. Arguments passed to the function.

    For eg: ***int sum(int a[], int n);***
    In the above example:
    1. *Sum is the name of the function.*
    2. *int sum****(int a[], int n)*** *a[] and n are arguments to the function sum.*
    3. ***int*** *sum(in a[], int n) This function sum is returning an integer.*
44. What is recursive function?
    A function which calls itself is called as recursive function.
45. What is meant by scope of a variable?
    Scope of the variable means in which functions the value of the variable would be visible or available.
46. What is a structure?
    Structure is a group of logically related data items which may be of different types, stored in contiguous memory locations, sharing a common name but distinguished by its members.
47. Differentiate between array and structure.
    Array - Array is a group of logically related data items of similar type stored in contiguous memory locations sharing a common name but distinguished by subscript.
    Structure - Structure is a group of logically related data items which may be of different types, stored in contiguous memory locations, sharing a common name but distinguished by its members.
48. How do you access the member of a structure?
    The data member of structure can be accessed using dot operator(.).
    Structure variable . data member
    For example: struct student a;
    To access a data member(name) of structure variable student then use a.name;
    If you declare a pointer to a structure then -> operator is used to access the data member of the structure.
    *For example struct student *a;*
    a->name is used to access the data member (name) of the pointer to structure variable.
49. What is union?
    Union is similar to structure. Union is a group of data items which may be of different data types by a common name. In Union – Only one location is large enough to accommodate values of different type at different times one at a time.
50. Differentiate between union and structure.
    Union is similar to structure. Union is also a group of data items which may be of different data types by a common name. But the difference lies in their storage allocation scheme.

In structure – the number of locations allocated would be equal to the number of members in the structure.

In Union – Only one location is large enough to accommodate values of different type at different times one at a time.

51. What is a pointer?

A variable which can store the address of another variable is called pointer.

52. What are pointer operators?

Address operator - & - retrieves the address of a variable.

*Dereference operator - * - used to access the value at a location by means of its address.*

53. What operations can be performed on pointers?

a. *We can subtract two pointer variables, if they point to the same array.*
b. *We can subtract an integer from a pointer variable.*
c. *We can add an integer to a pointer variable.*
d. *We cannot add, multiply or divide two pointer variables.*

Example:

*Int *p, *q;*

p – q	*p – 2*	*p + 2*	*allowed*
p +q	*p * q*	*p / q*	*Not allowed*

54. Differentiate between address operator and dereferencing operator.

Address operator - & - retrieves the address of a variable.

*Dereference operator - * - used to access the value at a location by means of its address.*

55. How do you declare a pointer variable

*datatype * variable;*

56. Differentiate between gets and getchar.

gets() accepts a multi-word string upto a new line character. char gets(char*);*

getchar() is used to read a character through standard input, keyboard.

57. Differentiate between putchar and puts.

puts() displays a string. It also adds the new-line character\n to the string automatically, so the cursor is moved down by one line after the string is displayed.

putchar() displays only a character on standard output, on the screen

58. What is a void return type?

A void return type indicates that a method/function does not return a value.

59. What is the syntax of structure?

```
Struct tag_name
{
Data members;
};
```

For example:

```
Struct  student
{
char name[40];
int rollno;
```

int section;
};

60. What is the syntax of printf and scanf statements?
Printf():
Syntax 1: Printf("format specifiers", list of variables);
For eg: printf("The sum of %d and %d = %d", a,b,sum);
Syantax 2: printf("string");
Eg: printf("Hello World");
Scanf()
Scanf("control string", list of variables);
For eg: scanf("%d%f%d", &x,&y,&z);
61. Write the syntax of **for** statement.
for(initialization; text; updation)
eg: for(i=0; i < 10; i++) { body of the loop}
62. Write the syntax of **while and do/while** statement

```
Initialization;                 initialization;
while(test)                     do
{                           {
Statements;                     statements;
Updation;                   updation;
}                           }  while(test);
```

63. Name the different file handling functions?
fopen(), feof(), fprintf(), fscanf(), fputc(), fgetc(), fread(), fwrite()

64. Name the different functions available in **math.h** header file?
sqrt(), pow(), sin(), cos() etc.
65. Name the different functions available in **ctype.h** header file?
islower(), isdigit(),isalpha(), isupper() etc.
66. Name the different functions available in **string.h** header file?
strlen(), strcmp(),strcat(),strrev().
67. What operations can be performed on strings?
Find the length - strlen(),
Compare two strings - strcmp(),
Concatenate two strings - strcat(),
Reverse a string - strrev().