INTERVIEW QUESTIONS ON C PROGRAMMING

1) What is C language?

C is a mid-level and procedural programming language. The Procedural programming language is also known as the structured programming language is a technique in which large programs are broken down into smaller modules, and each module uses structured code. This technique minimizes error and misinterpretation.

2) Why is C known as a mother language?

C is known as a mother language because most of the compilers and JVMs are written in C language. Most of the languages which are developed after C language has borrowed heavily from it like C++, Python, Rust, javascript, etc. It introduces new core concepts like arrays, functions, file handling which are used in these languages.

3) Why is C called a mid-level programming language?

C is called a mid-level programming language because it binds the low level and high - level programming language. We can use C language as a System programming to develop the operating system as well as an Application programming to generate menu driven customer driven billing system.

4) Who is the founder of C language?

Dennis Ritchie.

5) When was C language developed?

C language was developed in 1972 at bell laboratories of AT&T.

6) What are the features of the C language?

The main features of C language are given below:

- **Simple:** C is a simple language because it follows the structured approach, i.e., a program is broken into parts
- **Portable:** C is highly portable means that once the program is written can be run on any machine with little or no modifications.
- **Mid-Level :** C is a mid-level programming language as it combines the low-level language with the features of the high-level language.
- **Structured**: C is a structured language as the C program is broken into parts.
- **Fast Speed**: C language is very fast as it uses a powerful set of data types and operators.
- **Memory Management:** C provides an inbuilt memory function that saves the memory and improves the efficiency of our program.

- **Extensible**: C is an extensible language as it can adopt new features in the future.

7) What is the use of printf() and scanf() functions?

printf(): The printf() function is used to print the integer, character, float and string values on to the screen.

Following are the format specifier:

- %d: It is a format specifier used to print an integer value.
- %s: It is a format specifier used to print a string.
- %c: It is a format specifier used to display a character value.
- %f: It is a format specifier used to display a floating point value.

scanf(): The scanf() function is used to take input from the user.

8) What is the difference between the local variable and global variable in C?

Following are the differences between a local variable and global variable:

Basis for comparison	Local variable	Global variable
Declaration	A variable which is declared inside function or block is known as a local variable.	A variable which is declared outside function or block is known as a global variable.
Scope	The scope of a variable is available within a function in which they are declared.	The scope of a variable is available throughout the program.
Access	Variables can be accessed only by those statements inside a function in which they are declared.	Any statement in the entire program can access variables.
Life	Life of a variable is created when the function block is entered and destroyed on its exit.	Life of a variable exists until the program is executing.
Storage	Variables are stored in a stack unless specified.	The compiler decides the storage

		location of a variable.

9) What is the use of a static variable in C?

Following are the uses of a static variable:

- A variable which is declared as static is known as a static variable. The static variable retains its value between multiple function calls.
- Static variables are used because the scope of the static variable is available in the entire program. So, we can access a static variable anywhere in the program.
- The static variable is initially initialized to zero. If we update the value of a variable, then the updated value is assigned.
- The static variable is used as a common value which is shared by all the methods.
- The static variable is initialized only once in the memory heap to reduce the memory usage.

10) What is the use of the function in C?

Uses of C function are:

- C functions are used to avoid the rewriting the same code again and again in our program.
- C functions can be called any number of times from any place of our program.
- When a program is divided into functions, then any part of our program can easily be tracked.
- C functions provide the reusability concept, i.e., it breaks the big task into smaller tasks so that it makes the C program more understandable.

11) What is the difference between call by value and call by reference in C?

Following are the differences between a call by value and call by reference are:

	Call by value	Call by reference
Description	When a copy of the value is passed to the function, then the original value is not modified.	When a copy of the value is passed to the function, then the original value is modified.

Memory location	Actual arguments and formal arguments are created in separate memory locations.	Actual arguments and formal arguments are created in the same memory location.
Safety	In this case, actual arguments remain safe as they cannot be modified.	In this case, actual arguments are not reliable, as they are modified.
Arguments	The copies of the actual arguments are passed to the formal arguments.	The addresses of actual arguments are passed to their respective formal arguments.

Example of call by value:

```
1. #include <stdio.h>
2. void change(int,int);
3. int main()
4. {
5.     int a=10,b=20;
6.     change(a,b); //calling a function by passing the values of variables.
7.     printf("Value of a is: %d",a);
8.     printf("\n");
9.     printf("Value of b is: %d",b);
10.    return 0;
11. }
12. void change(int x,int y)
13. {
14.    x=13;
15.    y=17;
16. }
```

Output:

```
Value of a is: 10
Value of b is: 20
```

Example of call by reference:

```
1. #include <stdio.h>
2. void change(int*,int*);
3. int main()
4. {
5.     int a=10,b=20;
6.     change(&a,&b); // calling a function by passing references of variables.
7.     printf("Value of a is: %d",a);
8.     printf("\n");
9.     printf("Value of b is: %d",b);
10.    return 0;
11. }
12. void change(int *x,int *y)
13. {
14.    *x=13;
15.    *y=17;
16. }
```

```
Value of a is: 13
Value of b is: 17
```

12) What is recursion in C?

When a function calls itself, and this process is known as recursion. The function that calls itself is known as a recursive function.

Recursive function comes in two phases:

1. Winding phase
2. Unwinding phase

Winding phase: When the recursive function calls itself, and this phase ends when the condition is reached.

Unwinding phase: Unwinding phase starts when the condition is reached, and the control returns to the original call.

Example of recursion

```
1. #include <stdio.h>
2. int calculate_fact(int);
3. int main()
4. {
5.  int n=5,f;
6.  f=calculate_fact(n); // calling a function
7.  printf("factorial of a number is %d",f);
8.   return 0;
9. }
10. int calculate_fact(int a)
11. {
12.  if(a==1)
13.  {
14.      return 1;
15.  }
16.  else
17.  return a*calculate_fact(a-1); //calling a function recursively.
18.   }
```

Output:

```
factorial of a number is 120
```

13) What is an array in C?

An Array is a group of similar types of elements. It has a contiguous memory location. It makes the code optimized, easy to traverse and easy to sort. The size and type of arrays cannot be changed after its declaration.

Arrays are of two types:

- One-dimensional array: One-dimensional array is an array that stores the elements one after the another.

Syntax:

```
1. data_type array_name[size];
```

- Multidimensional array: Multidimensional array is an array that contains more than one array.

Syntax:

```
1. data_type array_name[size];
```

Example of an array:

```
1. #include <stdio.h>
2. int main()
3. {
4.     int arr[5]={1,2,3,4,5}; //an array consists of five integer values.
5.     for(int i=0;i<5;i++)
6.     {
7.         printf("%d ",arr[i]);
8.     }
9.      return 0;
10. }
```

Output:

```
1 2 3 4 5
```

14) What is a pointer in C?

A pointer is a variable that refers to the address of a value. It makes the code optimized and makes the performance fast. Whenever a variable is declared inside a program, then the system allocates some memory to a variable. The memory contains some address number. The variables that hold this address number is known as the pointer variable.

For example:

```
1. Data_type *p;
```

The above syntax tells that p is a pointer variable that holds the address number of a given data type value.

Example of pointer

```
1. #include <stdio.h>
2. int main()
3. {
4.    int *p; //pointer of type integer.
5.    int a=5;
6.    p=&a;
7.    printf("Address value of 'a' variable is %u",p);
8.     return 0;
9. }
```

Output:

```
Address value of 'a' variable is 428781252
```

15) What is the usage of the pointer in C?

- Accessing array elements: Pointers are used in traversing through an array of integers and strings. The string is an array of characters which is terminated by a null character '\0'.
- Dynamic memory allocation: Pointers are used in allocation and deallocation of memory during the execution of a program.
- Call by Reference: The pointers are used to pass a reference of a variable to other function.

- Data Structures like a tree, graph, linked list, etc.: The pointers are used to construct different data structures like tree, graph, linked list, etc.

16) What is a NULL pointer in C?

A pointer that doesn't refer to any address of value but NULL is known as a NULL pointer. When we assign a '0' value to a pointer of any type, then it becomes a Null pointer.

17) What is a far pointer in C?

A pointer which can access all the 16 segments (whole residence memory) of RAM is known as far pointer. A far pointer is a 32-bit pointer that obtains information outside the memory in a given section.

18) What is dangling pointer in C?

- If a pointer is pointing any memory location, but meanwhile another pointer deletes the memory occupied by the first pointer while the first pointer still points to that memory location, the first pointer will be known as a dangling pointer. This problem is known as a dangling pointer problem.
- Dangling pointer arises when an object is deleted without modifying the value of the pointer. The pointer points to the deallocated memory.

Let's see this through an example.

```
1. #include<stdio.h>
2. void main()
3. {
4.         int *ptr = malloc(constant value); //allocating a memory space.
5.         free(ptr); //ptr becomes a dangling pointer.
6. }
```

In the above example, initially memory is allocated to the pointer variable ptr, and then the memory is deallocated from the pointer variable. Now, pointer variable, i.e., ptr becomes a dangling pointer.

How to overcome the problem of a dangling pointer

The problem of a dangling pointer can be overcome by assigning a NULL value to the dangling pointer. Let's understand this through an example:

```
1. #include<stdio.h>
2.       void main()
```

```
3.  {
4.      int *ptr = malloc(constant value); //allocating a memory space.
5.      free(ptr); //ptr becomes a dangling pointer.
6.      ptr=NULL; //Now, ptr is no longer a dangling pointer.
7.  }
```

In the above example, after deallocating the memory from a pointer variable, ptr is assigned to a NULL value. This means that ptr does not point to any memory location. Therefore, it is no longer a dangling pointer.

19) What is pointer to pointer in C?

In case of a pointer to pointer concept, one pointer refers to the address of another pointer. The pointer to pointer is a chain of pointers. Generally, the pointer contains the address of a variable. The pointer to pointer contains the address of a first pointer. Let's understand this concept through an example:

```
1. #include <stdio.h>
2.  int main()
3. {
4.     int a=10;
5.     int *ptr,**pptr; // *ptr is a pointer and **pptr is a double pointer.
6.     ptr=&a;
7.     pptr=&ptr;
8.     printf("value of a is:%d",a);
9.     printf("\n");
10.    printf("value of *ptr is : %d",*ptr);
11.    printf("\n");
12.    printf("value of **pptr is : %d",**pptr);
13.    return 0;
14. }
```

In the above example, pptr is a double pointer pointing to the address of the ptr variable and ptr points to the address of 'a' variable.

20) What is static memory allocation?

- In case of static memory allocation, memory is allocated at compile time, and memory can't be increased while executing the program. It is used in the array.
- The lifetime of a variable in static memory is the lifetime of a program.
- The static memory is allocated using static keyword.
- The static memory is implemented using stacks or heap.
- The pointer is required to access the variable present in the static memory.
- The static memory is faster than dynamic memory.
- In static memory, more memory space is required to store the variable.

1. For example:
2. int a[10];

The above example creates an array of integer type, and the size of an array is fixed, i.e., 10.

21) What is dynamic memory allocation?

- In case of dynamic memory allocation, memory is allocated at runtime and memory can be increased while executing the program. It is used in the linked list.
- The malloc() or calloc() function is required to allocate the memory at the runtime.
- An allocation or deallocation of memory is done at the execution time of a program.
- No dynamic pointers are required to access the memory.
- The dynamic memory is implemented using data segments.
- Less memory space is required to store the variable.

1. For example
2. int *p= malloc(sizeof(int)*10);

The above example allocates the memory at runtime.

22) What functions are used for dynamic memory allocation in C language?

1. malloc()
    - The malloc() function is used to allocate the memory during the execution of the program.
    - It does not initialize the memory but carries the garbage value.
    - It returns a null pointer if it could not be able to allocate the requested space.

    Syntax

    1. ptr = (cast-type*) malloc(byte-size) // allocating the memory using malloc() function.

2. calloc()
    1. The calloc() is same as malloc() function, but the difference only is that it initializes the memory with zero value.

    Syntax

    1. ptr = (cast-type*)calloc(n, element-size);// allocating the memory using calloc() function.

2. realloc()
    1. The realloc() function is used to reallocate the memory to the new size.
    2. If sufficient space is not available in the memory, then the new block is allocated to accommodate the existing data.

    Syntax

    1. ptr = realloc(ptr, newsize); // updating the memory size using realloc() f unction.

    In the above syntax, ptr is allocated to a new size.

2. free():The free() function releases the memory allocated by either calloc() or malloc() function.

    Syntax

    1. free(ptr); // memory is released using free() function.

    The above syntax releases the memory from a pointer variable ptr.

23) What is the difference between malloc() and calloc()?

	calloc()	malloc()
Description	The malloc() function allocates a single block of requested memory.	The calloc() function allocates multiple blocks of requested memory.
Initialization	It initializes the content of the memory to zero.	It does not initialize the content of memory, so it carries the garbage value.
Number of arguments	It consists of two arguments.	It consists of only one argument.
Return value	It returns a pointer pointing to the allocated memory.	It returns a pointer pointing to the allocated memory.

24) What is structure?

- The structure is a user-defined data type that allows storing multiple types of data in a single unit. It occupies the sum of the memory of all members.
- The structure members can be accessed only through structure variables.
- Structure variables accessing the same structure but the memory allocated for each variable will be different.

Syntax of structure

```
1. struct structure_name
2. {
3.   Member_variable1;
4.  Member_variable2
5.  .
6.  .
```

7. }[structure variables];

Let's see a simple example.

```
1. #include <stdio.h>
2. struct student
3. {
4.     char name[10];       // structure members declaration.
5.     int age;
6. }s1;      //structure variable
7. int main()
8. {
9.     printf("Enter the name");
10.    scanf("%s",s1.name);
11.    printf("\n");
12.    printf("Enter the age");
13.    scanf("%d",&s1.age);
14.    printf("\n");
15.    printf("Name and age of a student: %s,%d",s1.name,s1.age);
16.    return 0;
17. }
```

Output:

```
Enter the name shikha
Enter the age 26
Name and age of a student: shikha,26
```

25) What is union?

- The union is a user-defined data type that allows storing multiple types of data in a single unit. However, it doesn't occupy the sum of the memory of all members. It holds the memory of the largest member only.

- In union, we can access only one variable at a time as it allocates one common space for all the members of a union.

Syntax of union

```
union union_name
{
Member_variable1;
Member_variable2;
.
.
Member_variable n;
}[union variables];
```

Let's see a simple example

```
#include<stdio.h>
union data
{
    int a;      //union members declaration.
    float b;
    char ch;
};
int main()
{
union data d;       //union variable.
d.a=3;
d.b=5.6;
d.ch='a';
printf("value of a is %d",d.a);
```

```
15. printf("\n");
16. printf("value of b is %f",d.b);
17. printf("\n");
18. printf("value of ch is %c",d.ch);
19. return 0;
20. }
```

Output:

```
value of a is 1085485921
value of b is 5.600022
value of ch is a
```

In the above example, the value of a and b gets corrupted, and only variable ch shows the actual output. This is because all the members of a union share the common memory space. Hence, the variable ch whose value is currently updated.

26) What is an auto keyword in C?

In C, every local variable of a function is known as an automatic (auto) variable. Variables which are declared inside the function block are known as a local variable. The local variables are also known as an auto variable. It is optional to use an auto keyword before the data type of a variable. If no value is stored in the local variable, then it consists of a garbage value.

27) What is the purpose of sprintf() function?

The sprintf() stands for "string print." The sprintf() function does not print the output on the console screen. It transfers the data to the buffer. It returns the total number of characters present in the string.

Syntax

```
1. int sprintf ( char * str, const char * format, ... );
```

Let's see a simple example

```
1. #include<stdio.h>
2. int main()
3. {
4. char a[20];
5. int n=sprintf(a,"javaToint");
```

```
6.  printf("value of n is %d",n);
7.  return 0;}
```

Output:

```
value of n is 9
```

28) Can we compile a program without main() function?

Yes, we can compile, but it can't be executed.

But, if we use #define, we can compile and run a C program without using the main() function. For example:

```
1. #include<stdio.h>
2. #define start main
3. void start() {
4.    printf("Hello");
5. }
```

29) What is a token?

The Token is an identifier. It can be constant, keyword, string literal, etc. A token is the smallest individual unit in a program. C has the following tokens:

1. Identifiers: Identifiers refer to the name of the variables.
2. Keywords: Keywords are the predefined words that are explained by the compiler.
3. Constants: Constants are the fixed values that cannot be changed during the execution of a program.
4. Operators: An operator is a symbol that performs the particular operation.
5. Special characters: All the characters except alphabets and digits are treated as special characters.

30) What is command line argument?

The argument passed to the main() function while executing the program is known as command line argument. For example:

```
1. main(int count, char *args[]){
2. //code to  be executed
```

```
3. }
```

31) What is the acronym for ANSI?

The ANSI stands for " American National Standard Institute." It is an organization that maintains the broad range of disciplines including photographic film, computer languages, data encoding, mechanical parts, safety and more.

32) What is the difference between getch() and getche()?

The getch() function reads a single character from the keyboard. It doesn't use any buffer, so entered data will not be displayed on the output screen.

The getche() function reads a single character from the keyword, but data is displayed on the output screen. Press Alt+f5 to see the entered character.

Let's see a simple example

```
1. #include<stdio.h>
2. #include<conio.h>
3. int main()
4. {
5.
6.  char ch;
7.  printf("Enter a character ");
8.  ch=getch(); // taking an user input without printing the value.
9.  printf("\nvalue of ch is %c",ch);
10. printf("\nEnter a character again ");
11. ch=getche(); // taking an user input and then displaying it on the screen.
12.  printf("\nvalue of ch is %c",ch);
13. return 0;
14. }
```

Output:

```
Enter a character
value of ch is a
Enter a character again a
```

```
value of ch is a
```

In the above example, the value entered through a getch() function is not displayed on the screen while the value entered through a getche() function is displayed on the screen.

3) What is the newline escape sequence?

The new line escape sequence is represented by "\n". It inserts a new line on the output screen.

35) What is the difference between near, far and huge pointers?

A virtual address is composed of the selector and offset.

A near pointer doesn't have explicit selector whereas far, and huge pointers have explicit selector. When you perform pointer arithmetic on the far pointer, the selector is not modified, but in case of a huge pointer, it can be modified.

These are the non-standard keywords and implementation specific. These are irrelevant in a modern platform.

36) What is the maximum length of an identifier?

It is 32 characters ideally but implementation specific.

37) What is typecasting?

The typecasting is a process of converting one data type into another is known as typecasting. If we want to store the floating type value to an int type, then we will convert the data type into another data type explicitly.

Syntax

```
1. (type_name) expression;
```

38) What are the functions to open and close the file in C language?

The fopen() function is used to open file whereas fclose() is used to close file.

39) Can we access the array using a pointer in C language?

Yes, by holding the base address of array into a pointer, we can access the array using a pointer.

40) What is an infinite loop?

A loop running continuously for an indefinite number of times is called the infinite loop.

Infinite For Loop:

```
1. for(;;){
2. //code to be executed
3. }
```

Infinite While Loop:

```
1. while(1){
2. //code to be executed
3. }
```

Infinite Do-While Loop:

```
1. do{
2. //code to be executed
3. }while(1);
```

41) Write a program to print "hello world" without using a semicolon?

```
1. #include<stdio.h>
2. void main(){
3.  if(printf("hello world")){} // It prints the ?hello world? on the screen.
4. }
```

1. What's the value of the expression 5["abxdef"]?

The answer is 'f'.

Explanation: The string mentioned "abxdef" is an array, and the expression is equal to "abxdef"[5]. Why is the inside-out expression equivalent? Because a[b] is equivalent to *(a + b) which is equivalent to *(b + a) which is equivalent to b[a].

2. What is a built-in function in C?

The most commonly used built-in functions in C are sacnf(), printf(), strcpy, strlwr, strcmp, strlen, strcat, and many more.

Built-function is also known as library functions that are provided by the system to make the life of a developer easy by assisting them to do certain commonly used predefined tasks. For example, if you need to print output or your program into the terminal, we use printf() in C.

3. In C, What is the #line used for?

In C, #line is used as a preprocessor to re-set the line number in the code, which takes a parameter as line number. Here is an example for the same.

```
#include <stdio.h>
        /*line 1*/


                        /*line 2*/
```

```
	int main(){
		/*line 3*/


				/*line 4*/
	printf("Hello world\n");				/*line 5*/
	//print current line					/*line
6*/
	printf("Line: %d\n",__LINE__);			/*line 7*/
	//reset the line number by 36			/*line 8*/
	#line 36 /*reseting*/
	//print current line					/*line
36*/
	printf("Line: %d\n",__LINE__);			/*line 37*/
	printf("Bye bye!!!\n");					/*line
39*/


				/*line 40*/
	return 0;
	/*line 41*/
}
				/*line 42*/
```

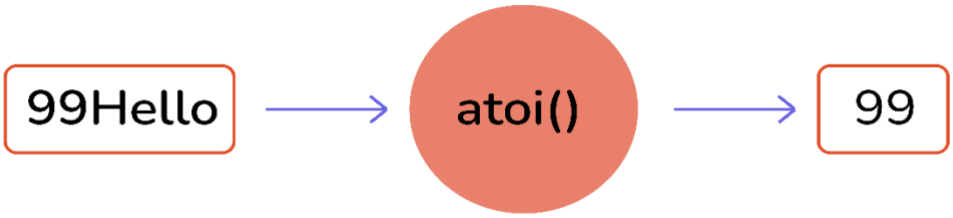
4. How can a string be converted to a number?

The function takes the string as an input that needs to be converted to an integer.

```
int atoi(const char *string)
```

Return Value:

- On successful conversion, it returns the desired integer value
- If the string starts with alpha-numeric char or only contains alpha-num char, 0 is returned.
- In case string starts with numeric character but is followed by alpha-num char, the string is converted to integer till the first occurrence of alphanumeric char.



InterviewBit

5. How can a number be converted to a string?

The function takes a pointer to an array of char elements that need to be converted, and a format string needs to be written in a buffer as a string

```
int sprintf(char *str, const char *format, ...)
```

The output after running the above code:

Output: Value of Pi = 3.141593

6. Why doesn't C support function overloading?

After you compile the C source, the symbol names need to be intact in the object code. If we introduce function overloading in our source, we should also provide name mangling as a preventive measure to avoid function name clashes. Also, as C is not a strictly typed language many things(ex: data types) are convertible to each other in C. Therefore, the complexity of overload resolution can introduce confusion in a language such as C.

When you compile a C source, symbol names will remain intact. If you introduce function overloading, you should provide a name mangling technique to prevent name clashes. Consequently, like C++, you'll have machine-generated symbol names in the compiled binary.

Additionally, C does not feature strict typing. Many things are implicitly convertible to each other in C. The complexity of overload resolution rules could introduce confusion in such kind of language

7. What is the difference between global int and static int declaration?

The difference between this is in scope. A truly global variable has a global scope and is visible everywhere in your program.

```
#include <stdio.h>


int my_global_var = 0;


int
```

```
main(void)


{
  printf("%d\n", my_global_var);
  return 0;
}
```

global_temp is a global variable that is visible to everything in your program, although to make it visible in other modules, you'd need an ”extern int global_temp; ” in other source files if you have a multi-file project.

A static variable has a local scope but its variables are not allocated in the stack segment of the memory. It can have less than global scope, although - like global variables - it resides in the .bss segment of your compiled binary.

```
#include <stdio.h>


int
myfunc(int val)


{
    static int my_static_var = 0;


    my_static_var += val;
    return my_static_var;
}


int
main(void)


{
   int myval;


   myval = myfunc(1);
   printf("first call %d\n", myval);
```

```
  myval = myfunc(10);

  printf("second call %d\n", myval);

  return 0;
}
```

8. Difference between const char* p and char const* p?

const char* p is a pointer to a const char.

char const* p is a pointer to a char const.

Since const char and char const are the same, it's the same.

9. Why n++ execute faster than n+1 ?

n++ being a unary operation, it just needs one variable. Whereas, n = n + 1 is a binary operation that adds overhead to take more time (also binary operation: n += 1). However, in modern platforms, it depends on few things such as processor architecture, C compiler, usage in your code, and other factors such as hardware problems.

While in the modern compiler even if you write n = n + 1 it will get converted into n++ when it goes into the optimized binary, and it will be equivalently efficient.

```
$ time perl -le '$n=0; foreach (1..100000000) { $n++ }'
real    0m2.389s
user    0m2.371s
sys 0m0.015s

$ time perl -le '$n=0; foreach (1..100000000) { $n=$n+1 }'
real    0m4.469s
user    0m4.442s
sys 0m0.020s
```

10. What are the advantages of Macro over function?

Macro on a high-level copy-paste, its definitions to places wherever it is called. Due to which it saves a lot of time, as no time is spent while passing the control to a new function and the control is always with the callee function. However, one downside is the size of the compiled binary is large but once compiled the program comparatively runs faster.

11. Specify different types of decision control statements?

All statements written in a program are executed from top to bottom one by one. Control statements are used to execute/transfer the control from one part of the program to another depending on the condition.

- If-else statement.
    - normal if-else statement.
    - Else-if statement
    - nested if-else statement.
- Switch statement.

12. What is the difference between struct and union in C?

A struct is a group of complex data structures stored in a block of memory where each member on the block gets a separate memory location to make them accessible at once

Whereas in the union, all the member variables are stored at the same location on the memory as a result to which while assigning a value to a member variable will change the value of all other members.

```
/* struct & union definations*/
struct bar {
        int a;      // we can use a & b both simultaneously
        char b;
}       bar;

union foo {
        int a;      // we can't use both a and b simultaneously
        char b;
}       foo;

/* using struc and union variables*/

struct bar y;
y.a = 3;  // OK to use
y.b = 'c'; // OK to use
```

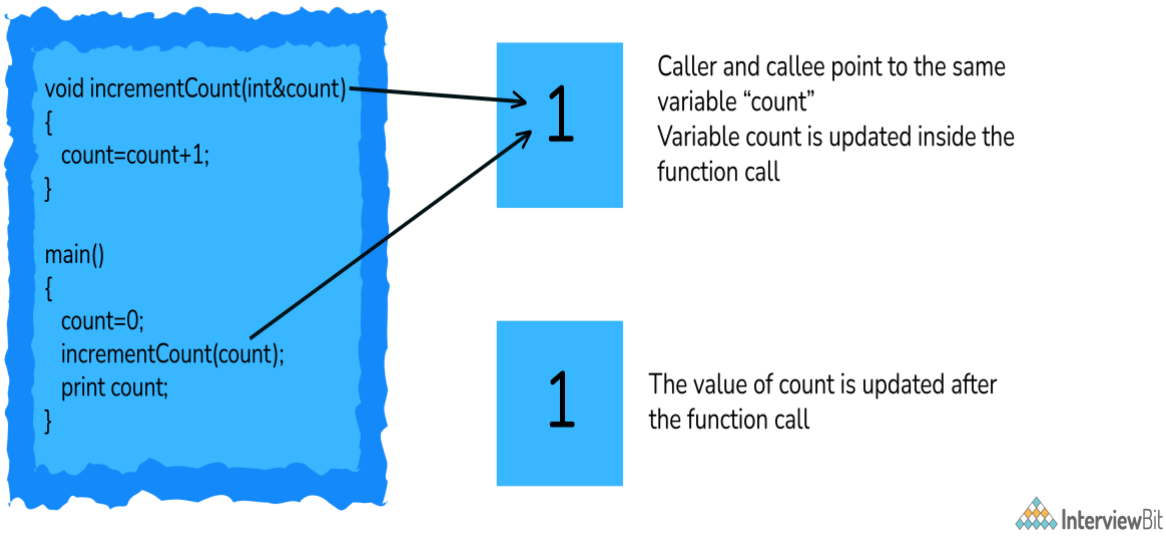
```
union foo x;

x.a = 3; // OK

x.b = 'c'; // NOl this affects the value of x.a!
```

13. What is call by reference in functions?

When we caller function makes a function call bypassing the addresses of actual parameters being passed, then this is called call by reference. In incall by reference, the operation performed on formal parameters affects the value of actual parameters because all the operations performed on the value stored in the address of actual parameters.

14. What is pass by reference in functions?

In Pass by reference, the callee receives the address and makes a copy of the address of an argument into the formal parameter. Callee function uses the address to access the actual argument (to do some manipulation). If the callee function changes the value addressed at the passed addre it will be visible to the caller function as well.



15. What is a memory leak? How to avoid it?

When we assign a variable it takes space of our RAM (either heap or RAM) dependent on the size of data type, however, if a programmer uses a memory available on the heap and forgets to a delta it, at some point all the memory available on the ram will be occupied with no memory left this can lead to a memory leak.

```
int main()

{

   char * ptr = malloc(sizeof(int));
```

```
    /* Do some work */
    /*Not freeing the allocated memory*/
    return 0;
}
```

To avoid memory leaks, you can trace all your memory allocations and think forward, where you want to destroy (in a good sense) that memory and place delete there. Another way is to use C++ smart pointer in C linking it to GNU compilers.

17. What is typedef?

typedef is a C keyword, used to define alias/synonyms for an existing type in C language. In most cases, we use typedef's to simplify the existing type declaration syntax. Or to provide specific descriptive names to a type.

```
typedef <existing-type> <new-type-identifiers>;
```

typedef provides an alias name to the existing complex type definition. With typedef, you can simply create an alias for any type. Whether it is a simple integer to complex function pointer or structure declaration, typedef will shorten your code.

18. Why is it usually a bad idea to use gets()? Suggest a workaround.

The standard input library gets() reads user input till it encounters a new line character. However, it does not check on the size of the variable being provided by the user is under the maximum size of the data type due to which makes the system vulnerable to buffer overflow and the input being written into memory where it isn't supposed to.

We, therefore, use fgets() to achieve the same with a restricted range of input

Bonus: It remained an official part of the language up to the 1999 ISO C standard, but it was officially removed by the 2011 standard. Most C implementations still support it, but at least GCC issues a warning for any code that uses it.

19. What is the difference between #include "..." and #include <...>?

In practice, the difference is in the location where the preprocessor searches for the included file.

For #include <filename> the C preprocessor looks for the filename in the predefined list of system directories first and then to the directories told by the user(we can use -I option to add directories to the mentioned predefined list).

For #include "filename" the preprocessor searches first in the same directory as the file containing the directive, and then follows the search path used for the #include <filename> form. This method is normally used to include programmer-defined header files.

20. What are dangling pointers? How are dangling pointers different from memory leaks?

The dangling pointer points to a memory that has already been freed. The storage is no longer allocated. Trying to access it might cause a Segmentation fault. A common way to end up with a dangling pointer:

```
#include<stdio.h>
#include<string.h>


char *func()
{
   char str[10];
   strcpy(str,"Hello!");
   return(str);
}
```

You are returning an address that was a local variable, which would have gone out of scope by the time control was returned to the calling function. (Undefined behavior)

```
*c = malloc(5izeof(int));
free(c);
*c = 3; //writing to freed location!
```

In the figure shown above writing to a memory that has been freed is an example of the dangling pointer, which makes the program crash.

A memory leak is something where the memory allocated is not freed which causes the program to use an undefined amount of memory from the ram making it unavailable for every other running program(or daemon) which causes the programs to crash. There are various tools like O profile testing which is useful to detect memory leaks on your programs.

```
void function(){
        char *leak = malloc (10); //leak assigned but not freed
}
```

21. What is the difference between 'g' and "g" in C?

In C double-quotes variables are identified as a string whereas single-quoted variables are identified as the character. Another major difference being the string (double-quoted) variables end with a null terminator that makes it a 2 character array.

22. Can you tell me how to check whether a linked list is circular?

Circular linked list is a variation of a linked list where the last node is pointing to the first node's information part. Therefore the last node does not point to null.

Algorithm to find whether the given linked list is circular

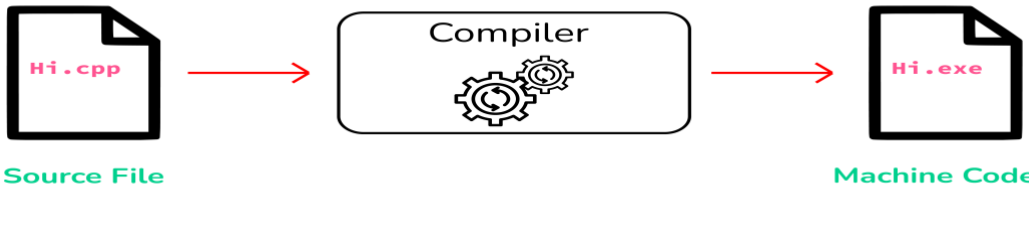
A very simple way to determine whether the linked list is circular or not

- Traverse the linked list
- Check if the node is pointing to the head.
- If yes then it is circular.

23. What is the use of a semicolon (;) at the end of every program statement?

It is majorly related to how the compiler reads(or parses) the entire code and breaks it into a set of instructions(or statements), to which semicolon in C acts as a boundary between two sets of instructions.

24. Differentiate Source Codes from Object Codes

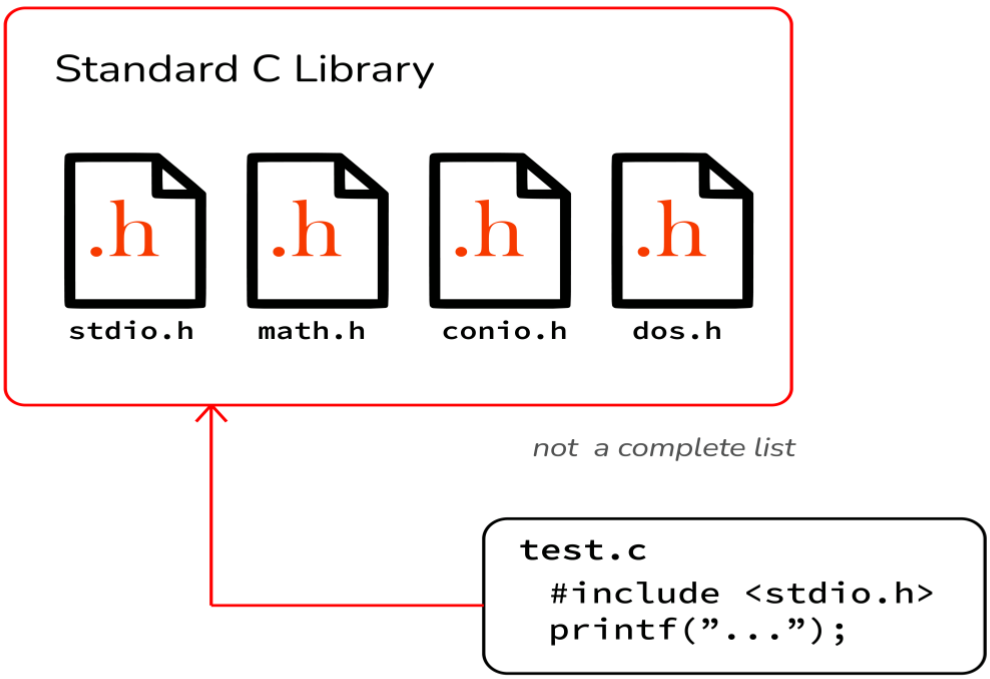


Source Code and Object Code Difference

The difference between the Source Code and Object Code is that Source Code is a collection of computer instructions written using a human-readable programming language while Object Code is a sequence of statements in machine language, and is the output after the compiler or an assembler converts the Source Code.

The last point about Object Code is the way the changes are reflected. When the Source Code is modified, each time the Source Code needs to be compiled to reflect the changes in the Object Code.

25. What are header files and what are its uses in C programming?



InterviewBit

Header Files in C

In C header files must have the extension as .h, which contains function definitions, data type definitions, macro, etc. The header is useful to import the above definitions to the source code using the #include directive. For example, if your source code needs to take input from the user do some manipulation and print the output on the terminal, it should have stdio.h file included as #include <stdio.h>, with which we can take input using scanf() do some manipulation and print using printf().

26. When is the "void" keyword used in a function

The keyword “void” is a data type that literally represents no data at all. The most obvious use of this is a function that returns nothing:

```
void PrintHello()
{
        printf("Hello\n");
        return;             // the function does "return", but no value is returned
}
```

Here we’ve declared a function, and all functions have a return type. In this case, we’ve said the return type is “void”, and that means, “no data at all” is returned. The other use for the void keyword is a void pointer. A void pointer points to the

memory location where the data type is undefined at the time of variable definition. Even you can define a function of return type void* or void pointer meaning “at compile time we don’t know what it will return” Let’s see an example of that.

```
void MyMemCopy(void* dst, const void* src, int numBytes)
{
        char* dst_c = reinterpret_cast<char*>(dst);
        const char* src_c = reinterpret_cast<const char*>(src);
        for (int i = 0; i < numBytes; ++i)
                dst_c[i] = src_c[i];
}
```

27. What is dynamic data structure?

A dynamic data structure (DDS) refers to an organization or collection of data in memory that has the flexibility to grow or shrink in size, enabling a programmer to control exactly how much memory is utilized. Dynamic data structures change in size by having unused memory allocated or de-allocated from the heap as needed.

Dynamic data structures play a key role in programming languages like C, C++, and Java because they provide the programmer with the flexibility to adjust the memory consumption of software programs.

Q1. What are the basic Datatypes supported in C Programming Language?

Ans: The Datatypes in C Language are broadly classified into 4 categories. They are as follows:

- Basic Datatypes
- Derived Datatypes
- Enumerated Datatypes
- Void Datatypes

The Basic Datatypes supported in C Language are as follows:

Datatype Name	Datatype Size	Datatype Range
short	1 byte	-128 to 127
unsigned short	1 byte	0 to 255
char	1 byte	-128 to 127
unsigned char	1 byte	0 to 255

int	2 bytes	-32,768 to 32,767
unsigned int	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,14
unsigned long	4 bytes	0 to 4,294,967,295
float	4 bytes	3.4E-38 to 3.4E+38
double	8 bytes	1.7E-308 to 1.7E+308
long double	10 bytes	3.4E-4932 to 1.1E+49

Q3. What do you mean by the Scope of the variable? What is the scope of the variables in C?

Ans: Scope of the variable can be defined as the part of the code area where the variables declared in the program can be accessed directly. In C, all identifiers are lexically (or statically) scoped.

Q4. What are static variables and functions?

Ans: The variables and functions that are declared using the keyword Static are considered as Static Variable and Static Functions. The variables declared using Static keyword will have their scope restricted to the function in which they are declared.

Q5. Differentiate between calloc() and malloc()

Ans: calloc() and malloc() are memory dynamic memory allocating functions. The only difference between them is that calloc() will load all the assigned memory locations with value 0 but malloc() will not.

Q6. What are the valid places where the programmer can apply Break Control Statement?

Ans: Break Control statement is valid to be used inside a loop and Switch control statements.

Q7. How can we store a negative integer?

Ans: To store a negative integer, we need to follow the following steps. Calculate the two's complement of the same positive integer.

Eg: 1011 (-5)

Step-1 − One's complement of 5: 1010

Step-2 − Add 1 to above, giving 1011, which is -5

Q8. Differentiate between Actual Parameters and Formal Parameters.

Ans: The Parameters which are sent from main function to the subdivided function are called as Actual Parameters and the parameters which are declared a the Subdivided function end are called as Formal Parameters.

Q9. Can a C program be compiled or executed in the absence of a main()?

Ans: The program will be compiled but will not be executed. To execute any C program, main() is required.

Q10. What do you mean by a Nested Structure?

Ans: When a data member of one structure is referred by the data member of another function, then the structure is called a Nested Structure.

Q11. What is a C Token?

Ans: Keywords, Constants, Special Symbols, Strings, Operators, Identifiers used in C program are referred to as C Tokens.

Q12. What is Preprocessor?

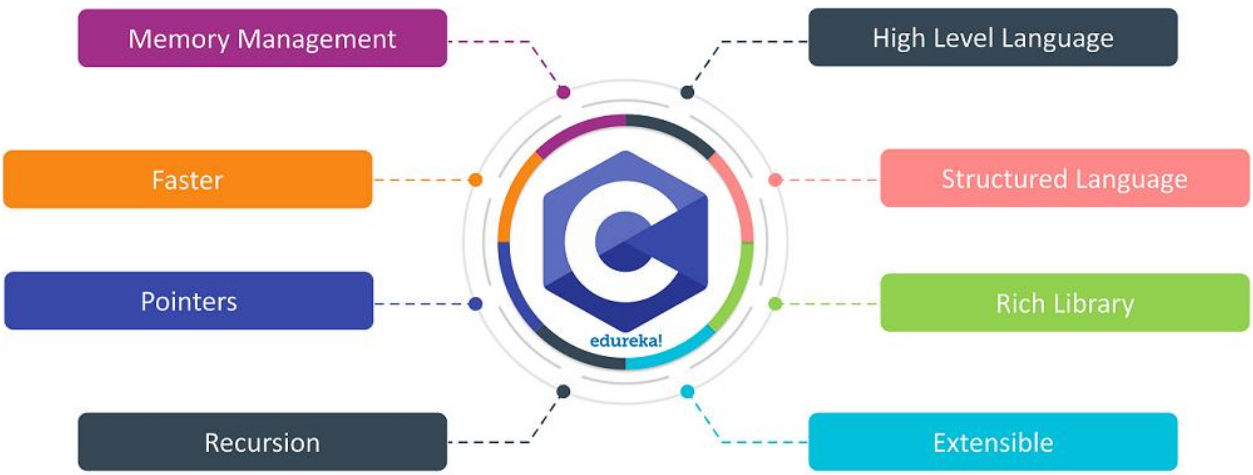
Ans: A Preprocessor Directive is considered as a built-in predefined function or macro that acts as a directive to the compiler and it gets executed before the actual C Program is executed.

In case you are facing any challenges with these C Programming Interview Questions, please write your problems in the comment section below.

Q13. Why is C called the Mother of all Languages?

Ans: C introduced many core concepts and data structures like arrays, lists, functions, strings, etc. Many languages designed after C are designed on the basis of C Language. Hence, it is considered as the mother of all languages.

Q14. Mention the features of C Programming Language.



Q17. What is /0 character?

Ans: The Symbol mentioned is called a Null Character. It is considered as the terminating character used in strings to notify the end of the string to the compiler.

Q18. What is the main difference between the Compiler and the Interpreter?

Ans: Compiler is used in C Language and it translates the complete code into the Machine Code in one shot. On the other hand, Interpreter is used in Java Programming Langauge and other high-end programming languages. It is designed to compile code in line by line fashion.

Q19. Can I use int datatype to store 32768 value?

Ans: No, Integer datatype will support the range between -32768 and 32767. Any value exceeding that will not be stored. We can either use float or long int.

Q20. How is a Function declared in C Language?

Ans: A function in C language is declared as follows,

```
1return_type function_name(formal parameter list)
2{
3       Function_Body;
4}
```

Q23. Where can we not use &(address operator in C)?

Ans: We cannot use & on constants and on a variable which is declared using the register storage class.

Q26. Differentiate between getch() and getche().

Ans: Both the functions are designed to read characters from the keyboard and the only difference is that

getch(): reads characters from the keyboard but it does not use any buffers. Hence, data is not displayed on the screen.

getche(): reads characters from the keyboard and it uses a buffer. Hence, data is displayed on the screen.

//Example

```
1  #include&lt;stdio.h&gt;
2  #include&lt;conio.h&gt;
3  int main()
```

```
4  {
5       char ch;
6       printf("Please enter a character ");
7       ch=getch();
8       printf("nYour entered character is %c",ch);
9       printf("nPlease enter another character ");
10      ch=getche();
11      printf("nYour new character is %c",ch);
12      return 0;
13}
```

//Output

Please enter a character
Your entered character is x
Please enter another character z
Your new character is z

Q27. Explain toupper() with an example.

Ans. toupper() is a function designed to convert lowercase words/characters into upper case.

//Example

```
1#include&lt;stdio.h&gt;
2#include&lt;ctype.h&gt;
3int main()
4{
5    char c;
6    c=a;
7    printf("%c after conversions  %c", c, toupper(c));
8    c=B;
9    printf("%c after conversions  %c", c, toupper(c));
```

//Output:

a after conversions A
B after conversions B

Q28. Write a code to generate random numbers in C Language.

Ans: Random numbers in C Language can be generated as follows:

```
1  #include&lt;stdio.h&gt;
2  #include&lt;stdlib.h&gt;
3  int main()
4  {
5      int a,b;
6      for(a=1;a&lt;=10;a++)
7      {
8          b=rand();
9          printf("%dn",b);
10     }
11     return 0;
12}
```

//Output

1987384758
2057844389
3475398489
2247357398
1435983905

Q29. Can I create a customized Head File in C language?

Ans: It is possible to create a new header file. Create a file with function prototypes that need to be used in the program. Include the file in the '#include' section in its name.

Q31. Explain Local Static Variables and what is their use?

Ans: A local static variable is a variable whose life doesn't end with a function call where it is declared. It extends for the lifetime of the complete program. All calls to the function share the same copy of local static variables.

```
1  #include&lt;stdio.h&gt;
2  void fun()
3  {
4      static int x;
```

```
5      printf("%d ", x);
6      x = x + 1;
7  }
8  int main()
9  {
10     fun();
11     fun();
12     return 0;
13}
```

//Output

0 1

Q32. What is the difference between declaring a header file with < > and ” “?

Ans: If the Header File is declared using < > then the compiler searches for the header file within the Built-in Path. If the Header File is declared using ” ” then the compiler will search for the Header File in the current working directory and if not found then it searches for the file in other locations.

Q33. When should we use the register storage specifier?

Ans: We use Register Storage Specifier if a certain variable is used very frequently. This helps the compiler to locate the variable as the variable will be declared in one of the CPU registers.

Q34. Which statement is efficient and why? x=x+1; or x++; ?

Ans: x++; is the most efficient statement as it just a single instruction to the compiler while the other is not.

Q35. Can I declare the same variable name to the variables which have different scopes?

Ans: Yes, Same variable name can be declared to the variables with different variable scopes as the following example.

```
1int var;
2void function()
3{
4   int variable;
5}
```

```
6int main()
7{
8   int variable;
9}
```

Q36. Which variable can be used to access Union data members if the Union variable is declared as a pointer variable?

Ans: Arrow Operator(->) can be used to access the data members of a Union if the Union Variable is declared as a pointer variable.

Q37. Mention File operations in C Language.

Ans: Basic File Handling Techniques in C, provide the basic functionalities that user can perform against files in the system.

Function	Operation
fopen()	To Open a File
fclose()	To Close a File
fgets()	To Read a File
fprint()	To Write into a File

In case you are facing any challenges with these C Programming Interview Questions, please write your problems in the comment section below.

Q38. What are the different storage class specifiers in C?

Ans: The different storage specifiers available in C Language are as follows:

- auto
- register
- static
- extern

Q39. What is typecasting?

Ans: Typecasting is a process of converting one data type into another is known as typecasting. If we want to store the floating type value to an int type, then we will convert the data type into another data type explicitly.

Syntax:

```
1(type_name) expression;
```

Q48. Which structure is used to link the program and the operating system?

Ans: The answer can be explained through the following points,

- The structure used to link the operating system to a program is file.
- The file is defined in the header file “stdio.h”(standard input/output header file).
- It contains the information about the file being used, its current size and its location in memory.
- It contains a character pointer that points to the character that is being opened.
- Opening a file establishes a link between the program and the operating system about which file is to be accessed.

Q49. What are the limitations of scanf() and how can it be avoided?

Ans: The Limitations of scanf() are as follows:

- scanf() cannot work with the string of characters.
- It is not possible to enter a multiword string into a single variable using scanf().
- To avoid this the gets( ) function is used.
- It gets a string from the keyboard and is terminated when enter key is pressed.
- Here the spaces and tabs are acceptable as part of the input string.

Q50. Differentiate between the macros and the functions.

Ans: The differences between macros and functions can be explained as follows:

- Macro call replaces the templates with the expansion in a literal way.
- The Macro call makes the program run faster but also increases the program size.
- Macro is simple and avoids errors related to the function calls.
- In a function, call control is transferred to the function along with arguments.
- It makes the functions small and compact.
- Passing arguments and getting back the returned value takes time and makes the program run at a slower rate.

Q51. Suppose a global variable and local variable have the same name. Is it is possible to access a global variable from a block where local variables are defined?

Ans: No. It is not possible in C. It is always the most local variable that gets preference.

1. What do you understand by calloc()?

calloc() is a dynamic memory allocation function that loads all the assigned memory locations with 0 value.

2. What happens when a header file is included with-in double quotes ““?

When a header file in c++ is included in double-quotes, the particular header file is first searched in the compiler’s current working directory. If not found, then the built-in include path is also searched.

3. Define <stdio.h>.

It is a header file in C that contains prototypes and definitions of commands such as scanf and printf.[2] [MOU3]

4. One of the most common c interview questions is to define the What is the use of static functions.?

When we want to restrict access to functions, we need to make them static. Making functions static allows us to reuse the same function in C programming name in multiple files.

5. Name the four categories in which data types in the C programming language are divided in.

Basic data types - Arithmetic data types, further divided into integer and floating-point types

Derived datatypes -Arithmetic data types that define variables and assign discrete integer values only

Void data types - no value is available

Enumerated data types -Array types, pointer types, function, structure and union types

6. What is the function of s++ and ++s?

s++ is a single machine instruction used to increment the value of s by 1. (Post increment). ++s is used to carry out pre-increment.

7. What is the use of the ‘==’ symbol?

The ‘==’ symbol or “equivalent to” or “equal to” symbol is a relational operator, i.e., it is used to compare two values or variables.

8. What is the output of the following code snippet?

```
#include <stdio.h>

void local_static()

{

static int a;

printf("%d ", a);

a= a + 1;

}

int main()

{

local_static();

local_static();

return 0;

}
```

0 1

10. Name a ternary operator in the C programming language.

The conditional operator (?:)

1. Why is int known as a reserved word?

As int is a part of standard C language library, and it is not possible to use it for any other activity except its intended functionality, it is known as a reserved word.

2. This is one of the most commonly asked C programming interview questions. What will this code snippet return?

```
void display(unsigned int n)
```

```
{
if(n > 0)
{
display(n-1);
printf("%d ", n);
}
}
```

Prints number from 1 to n.

3. Another frequent c interview question is what is meant by Call by reference?

When a variable's value is sent as a parameter to a function, it is known as call by reference. The process can alter the value of the variable within the function.

4. What information is given to the compiler while declaring a prototype function?

The following information is given while declaring a prototype function:

- Name of the function
- Parameters list of the function
- Return type of the function.[6]

5. Why are objects declared as volatile are omitted from optimization?

This is because, at any time, the values of the objects can be changed by code outside the scope of the current code.

6. Give the equivalent FOR LOOP format of the following:

```
a=0;
while (a<=10) {
printf ("%d\n", a * a);
a++;
}
for (a=0; a<=10; a++)
```

printf ("%d\n", a * a);

7. What is a modifier in the C programming language? Name the five available modifiers.

It is used as a prefix to primary data type to indicate the storage space allocation's modification to a variable. The available modifiers are:

1. short
2. long
3. long long
4. signed
5. unsigned

8. A pointer *a points to a variable v. What can 'v' contain?

Pointers is a concept available in C and C++. The variable 'v' might contain the address of another memory or a value.

9. What three parameters fseek() function require to work after the file is opened by the fopen() function?

The number of bytes to search, the point of origin of the file, and a file pointer to the file.

10. Name a type of entry controlled and exit controlled loop in C programming.

Entry controlled loop- For loop (The condition is checked at the beginning)

Exit Controlled loop- do-while loop.[7] (The condition is checked in the end, i.e. loop runs at least once)

1. Give the output of the following program:

```
#include <stdio.h>

int main(void)

{

int arr[] = {20,40};

int *a = arr;
```

```
*a++;

printf("arr[0] = %d, arr[1] = %d, *a = %d",

arr[0], arr[1], *a);

return 0;

}
```

arr[0] = 20, arr[1] = 40, *p = 40

2. One of the frequently asked c interview questions could be to explain Canif you can we free a block of memory that has been allocated previously? If yes, how?

A block of memory previously allocated can be freed by using free(). The memory can also be released if the pointer holding that memory address is: realloc(ptr,0).

3. How to declare a variable as a pointer to a function that takes a single character-pointer argument and returns a character?

char (*a) (char*);

4. What is the stack area?

The stack area is used to store arguments and local variables of a method. It stays in memory until the particular method is not terminated.

5. What is the function of the following statement?

sscanf(str, “%d”, &i);

To convert a string value to an integer value.

6. Will the value of ‘a’ and ‘b’ be identical or different? Why?

```
float num = 1.0;

int a = (int) num;

int b = * (int *) &num;
```

The variable stores a value of num that has been first cast to an integer pointer and then dereferenced.

7. What are huge pointers?

Huge pointers are 32-bit pointers that can be accessed outside the segment, and the segment part can be modified, unlike far pointers.

8. What will be the output of the following?

```
#include<stdio.h>

main()

{

char *a = "abc";

a[2] = 'd';

printf("%c", *a);

}
```

The program will crash as the pointer points to a constant string, and the program is trying to change its values.

9. What is a union?

A union is a data type used to store different types of data at the exact memory location. Only one member of a union is helpful at any given time.

10. How is import in Java different from #include in C?

Import is a keyword, but #include is a statement processed by pre-processor software. #include increases the size of the code.

Q3). What do you mean by Data Type in C?

Ans: Data Type defines which type of value a variable can store. In C

programming language each variable is associated with a data type and each data

type requires different size of memory.

Some most usable data type:

char: char shows the variable of type char stores single character. char requires a

one byte of memory.

Ex: char ch;

Here ch is variable and having data type “char”.

int: int shows the variable having data type int stores integer value. int requires a four byte of memory.

Ex: int i;

here i is variable of type integers.

float: float is used to store decimal numbers with single precision. float requires a four byte of memory.

Ex:float f;

Here f is variable of type float.

double: double is used to store decimal numbers with double precision.

Ex: double d;

Here d is variable of type double.

Q4). How many types of data type in C Language?

Ans: C language supports 2 type of data types:

a). Primary data types:

Primary data types are known as fundamental data types.Some primary data types are integer(int), floating point(float), character(char) and void.

b).Derived data types:

Derived data types are derived from primary datatypes. Some derived data types are array, structure, union and pointer.

Q5). What is constant and variable in C .

Ans: Constant: A constant is a value or an identifier whose value cannot be changed in a program. identifier also can be defined as a constant.

For example:

const double PI = 3.14

Here, PI is a constant. And the PI contains 3.14 for this program.

Variables

A variables is a value or an identifier whose value can be changed in a program.

Variable is a container to hold data.

For example:

int score = 6;

Here, score is a variable of integer type and it contains value 6. The value of score can be change so here score is variable.

Q6) What is the use of printf() and scanf() functions?

Ans: printf(): The printf() function is used to print values on to the screen(console). Value may be anythings like integer, float, character and string.

scanf():The scanf() function is used to take input from the user.

Q7). What is the difference between the local variable and global variable in C.

Ans: local variable: It is defined inside a function. It is accessible only in the function in which it is defined.

Global variable : It is defined outside of all the functions in a program. It is available to all the functions in the program.

Q8) Write format specifiers in C.

Ans: %d: It is used to print an integer value.

%s: It is used to print a string.

%c: It is used to display a character value.

%f: It is used to display a floating point value.

Q9) How many spaces a Tab has in C.

Ans: A Tab has 8 spaces.

Q10) What is static variable in C? What is its use?

Ans: static keyword is used to declared static variable. Static variables preserves their value outside of their scope. Hence, static variables preserve their previous value in their previous scope and are not initialized again in the new scope.

Scope of the static variable is available in the entire program. So, we can access a static variable anywhere in the program.

The initial value of the static variable is zero. The static variable shares common value with all the methods.The static variable is initialized only once in the memory heap to reduce the memory usage.

Q11) Can we compile and execute a program without main() function?

Ans: Yes, we can compile and execute using macros.

```
#include <stdlib.h>
#include <stdio.h>
#define fun main
int fun(void)
{
   printf("Quescol");
```

```
    return 0;


}
```

Output:

```
Quescol
```

Q15). What is the difference between the = symbol and == symbol?

Ans: The single = symbol is an assignment operator which is used to assign value to a variable whereas the double == symbol is an relational operator which is used to compare the two values, is also called “equal to” or “equivalent to”.

Q16). What is C Token?

Ans: In c, Keywords, Constants, Special Symbols, Strings, Operators, Identifiers are referred as Tokens.

Q17). Where can we not use &(address operator in C)?

Ans: We cannot use ‘&’ on constants and on a variable which is declared using the register storage class.

Q18). What is l-value and r-value?

Ans: The term l-value and r-value are named because of their appearance in the expression.

L-value appeared on the left side of the assignment operator. Basically in general case variable is considered as L-value in which we are going to assign some values.

R-value appeared on the right side of the assignment operator.

Example: In this expression “x = 15”, x is considered as l-value and 15 is as r-value.

Q19). What is the difference between puts() and printf()?

Ans:

puts()	printf()
1. The puts() function only allows you to print a string on the console output screen.	1. The printf() function can print strings as well as mixed types of variables on the console output screen.
2.The syntax for puts: puts(str)	2.The syntax for printf: printf(str)
3. The puts() function prints string and automatically adds one new character at the end of the string that is a new line character(\n).	3.The printf() function prints text, string, text+ value, etc, without adding extra character at the end.
4. Its implementation is simpler than printf.	4. Its implementation is complex as compared to puts.

Q1). Explain IF-Else Statement in C Language.

Ans:

IF-Else Statement works as a control flow statement in C. if statement is followed by an optional else statement.

Syntax :

```
if(boolean_expression) {

   /* statement s1 will execute if the boolean_expression is true */

} else {

   /* statement s2 will execute if the boolean_expression is false */

}
```

If the Boolean_expression is true, then the if block will be executed, otherwise, the else block will be executed.

Q3). Break and continue in C Language with example.

Ans:

break: break statement terminates the loop when it is encountered. The break statement can be used with decision making statement such as if…else and loop like for, while and do..while.

```
#include <stdio.h>
int main()
{
    int i, n, sum = 0;
    for(i=1; i <= 5; ++i)
```

```
    {
        printf("Enter a n%d: ",i);
        scanf("%d",&n);
        if(n > 6)
        {
            break;
        }
        sum += n;
    }
    printf("Sum = %d",sum);
    return 0;
}
```

Output:

```
Enter a n1: 2
Enter a n2: 5
Enter a n3: 2
Enter a n4: 1
Enter a n5: 7
Sum = 10
```

continue :

The continue statement is used to skips some statements inside the loop. The continue statement is used with decision making statement such as if…else.

```
# include <stdio.h>
int main()
{
    int i,n, sum = 0;
    for(i=1; i <= 5; ++i)
    {
        printf("Enter a n%d: ",i);
```

```
      scanf("%d",&n);
      if(n < 5)
      {
         continue;
      }
      sum += n;
   }
   printf("Sum = %d",sum);
   return 0;
}
```

Output:

```
Enter a n1: 4
Enter a n2: 3
Enter a n3: 9
Enter a n4: 1
Enter a n5: 8
Sum = 17
```

Q4). What are the valid places where the programmer can apply Break Control Statement?

Ans: It is valid to be used inside a loop and Switch control statements.

Q5). Switch in C Language with example.

A switch statement allows to test a variable for equality against a list of cases. It is good practise to use switch statement than nested if…else . Because switch is faster than nested if…else(not always).

```
#include <stdio.h>
int main () {
   int division = 3;
   switch(division) {
```

```
    case 1 :
       printf("1st Division\n" );
       break;
    case 2 :
          printf("2nd Division\n" );
       break;
    case 3 :
       printf("3rd Division\n" );
       break;
    default :
       printf("runner up\n" );
  }
  return 0;
}
```

Output:

```
3rd Division
```

Q6). While Loop in C Language with example.

A while loop is used to executes a statement multiple time as long as a given condition is true.

```
#include <stdio.h>
int main () {
  int a = 0;
  while( a < 10 ) {
    printf("value of a: %d\n", a);
    a++;
  }
  return 0;
```

```
}
```

Output:

```
value of a: 0
value of a: 1
value of a: 2
value of a: 3
value of a: 4
value of a: 5
value of a: 6
value of a: 7
value of a: 8
value of a: 9
```

Q7). What is the difference between while (0) and while (1)?

The term while is conditional based loop, While(0) means the condition of while loop is false so it will never execute a particular section of the code inside the program. Whereas while(1) is the opposite of while(0) and it will execute the particular section of code infinite times.

Q8). Difference b/w Entry controlled and Exit controlled loop?

In Entry controlled loop first the loop condition is checked then after the body of the loop is executed whereas in Exit controlled loop first the body of the loop is executed then after condition is checked.

Entry Controlled Loops are : for, while

Exit Controlled Loop is : do while

Q9). Can we use continue statement without using a loop?

No, we cannot use continue statement without using a loop, it can be used within any loop whether its while , for , or Do-while loop. In case if we try to use continue without using loop, there will be a compile error i.e"Misplaced continue".

Q10) Explain for Loop in C Language with example.

A for loop is used to executes a statement for a specific number of times.

```
#include <stdio.h>
int main () {
   int i;
   for( i = 0; i < 10; i++ ){
      printf("value of a: %d\n", i);
   }
   return 0;
}
```

Output:

```
value of a: 0
value of a: 1
value of a: 2
value of a: 3
value of a: 4
value of a: 5
value of a: 6
value of a: 7
value of a: 8
value of a: 9
```

Q11). What is the difference between ++a and a++.

'++a' is called prefix increment. In this, first value of the variable "a" gets incremented and then assigned to the variable a. On the other hand "a++" is called post increment.The value stored in the variable "a" gets incremented after the execution of the particular line.

Q2). What is recursion in C?

Ans: When a function calls itself, and this process is known as recursion. The function that calls itself is known as a recursive function.

```
#include<stdio.h>
int fibonacci(int);
int main(){
        int n, i, k;
        printf("Enter maximum length of fibonacci series\n");
        scanf("%d",&n);
        for(i=0;i<n;i++) {
                printf("%d ",fibonacci(i));
        }
}

int fibonacci(int i){
        if(i==0)
           return 0;
        else if(i==1)
           return 1;
        else
           return (fibonacci(i-1)+fibonacci(i-2));

}
```

Output:

```
Enter maximum length of fibonacci series
5
0 1 1 2 3
```

Q3). What are called and calling functions?

Ans: Parent function which calls any function is known as called function and the function which is being called is known as calling function.

```
int main()
```

```
{
   name_Func()
}
```

Here, name_Func() is called within the main() function, so main() is called function, while name_Func() is calling function.

Q4). Are exit () and return statements same in function definition?

Ans: No, both are different as we used exit() to immediately exit from the program, where as return is used to return the control of programs execution from the called function to calling function.

Q5). When is the "void" keyword used in a function?

Ans: When we declare the function we have to decide whether we want some return value from the function or not, if we only want to print some value or statements on the screen we use void on the left side of function name otherwise we place the data type of the value on the left side of function name.

Q6). Differentiate between call by value and call by reference.

Ans:

Factor	Call by Value	Call by Reference
Safety	In call by value actual parameters are safe because operations are performed on formal parameter.	In call by value operations performed on actual parameters so i not safe here.

Memory Location	In this separate memory locations are created for actual and formal parameters	In this actual and Formal parameters share the same memory space.
Parameters	In this copy of actual parameters are passed.	In this actual parameters are passed.

Q3) What is a NULL pointer in C?

Ans: A pointer that doesn't refer to any address of value but NULL is known as a NULL pointer. When we assign a '0' value to a pointer of any type, then it becomes a Null pointer.

Q4) What is a far pointer in C?

Ans: A pointer which can access all the 16 segments of RAM is known as far pointer. A far pointer is a 32-bit pointer that obtains information outside the memory in a given section.

Q5) What is a dangling pointer in C?

Ans: A pointer that doesn't refer to any address of value but NULL is known as a NULL pointer. When we assign a '0' value to a pointer of any type, then it becomes a Null pointer.

Q6). What is Wild Pointers in C.

Ans: Such pointers in C which are Uninitialized in c code are called Wild Pointers.

In general these pointers are pointing to the arbitrary memory location and cause program to crash.

Q7) can a pointer access the array?

Ans: Pointers can access the array by holding the base address of the array.

8). Why a[5] == 5[a]?

Ans: As we know that

a[b] == *(a+b)

so

a[2] == *(a+2)

therefore,

2[a] = *(2+a)

Here is basically commutative law which is followed by it.

Q3). Difference between malloc() and calloc() functions?

Ans:

Malloc()	Calloc()
1. The term malloc() stands for memory allocation.	1. The term malloc() stands for contiguous allocation.

2. It takes one argument, that is, the number of bytes.	2. It takes two arguments i.e number of blocks and size of each block.
3.syntax of malloc(): void *malloc(size_t n); Where n indicates allocates bytes of memory. If the allocation succeeds, a void pointer to the allocated memory is returned. Otherwise, NULL is returned.	3.syntax of calloc(): void *calloc(size_t n, size_t size); Allocates a contiguous block of memory large enough to hold n elements of size bytes each. The allocated region is initialized to zero.
4. It is faster than calloc.	4. It is somewhat slower than malloc() as it takes extra steps to initialize the allocated memory zero.

Q4). What do you mean by a Nested Structure?

```
struct Date
{
int birth_date;
int birth_month;
int birth_year;
};
struct personal_recod
{
```

```
char name[25];
struct Date birth_date;
float salary;
}person;
```

Q #3) What is the description for syntax errors?

Answer: The mistakes/errors that occur while creating a program are called syntax errors. Misspelled commands or incorrect case commands, an incorrect number of parameters in calling method /function, data type mismatches can be identified as common examples for syntax errors.

Q #12) What is the explanation for prototype function in C?

Answer: Prototype function is a declaration of a function with the following information to the compiler.

- Name of the function.
- The return type of the function.
- Parameters list of the function.

```
int Sum(int, int);
```

In this example Name of the function is Sum, the return type is the integer data type and it accepts two integer parameters.

Q #13) What is the explanation for the cyclic nature of data types in C?

Answer: Some of the data types in C have special characteristic nature when a developer assigns value beyond the range of the data type. There will be no compiler error and the value changes according to a cyclic order. This is called cyclic nature. Char, int, long int data types have this property. Further float, double and long double data types do not have this property.

Q #14) Describe the header file and its usage in C programming?

Answer: The file containing the definitions and prototypes of the functions being used in the program are called a header file. It is also known as a library file.

Example: The header file contains commands like printf and scanf is from the stdio.h library file.

Q #15) There is a practice in coding to keep some code blocks in comment symbols than delete it when debugging. How this affects when debugging?

Answer: This concept is called commenting out and this is the way to isolate some part of the code which scans possible reason for the error. Also, this concept helps to save time because if the code is not the reason for the issue it can simply be removed from comment.

Q #16) What are the general description for loop statements and available loop types in C?

Answer: A statement that allows the execution of statements or groups of statements in a repeated way is defined as a loop.

The following diagram explains a general form of a loop.

There are 4 types of loop statements in C.

- While loop
- For Loop
- Do…While Loop
- Nested Loop

Q #17) What is a nested loop?

Answer: A loop that runs within another loop is referred to as a nested loop. The first loop is called the Outer Loop and the inside loop is called the Inner Loop. The inner loop executes the number of times defined in an outer loop.

Q #18) What is the general form of function in C?

Answer: The function definition in C contains four main sections.

```
return_type function_name( parameter list )
{
    body of the function
}
```

- Return Type: Data type of the return value of the function.
- Function Name: The name of the function and it is important to have a meaningful name that describes the activity of the function.
- Parameters: The input values for the function that are used to perform the required action.
- Function Body: Collection of statements that performs the required action.

Q #20) What are the valid places to have keyword "Break"?

Answer: The purpose of the Break keyword is to bring the control out of the code block which is executing. It can appear only in looping or switch statements.

Q #21) What is the behavioral difference when the header file is included in double-quotes ("") and angular braces (<>)?

Answer: When the Header file is included within double quotes (" "), compiler search first in the working directory for the particular header file. If not found, then it searches the file in the include path. But when the Header file is included within angular braces (<>), the compiler only searches in the working directory for the particular header file.

Q #22) What is a sequential access file?

Answer: General programs store data into files and retrieve existing data from files. With the sequential access file, such data are saved in a sequential pattern. When retrieving data from such files each data is read one by one until the required information is found.

Q #23) What is the method to save data in a stack data structure type?

Answer: Data is stored in the Stack data structure type using the First In Last Out (FILO) mechanism. Only top of the stack is accessible at a given instance. Storing mechanism is referred as a PUSH and retrieve is referred to as a POP.

Q #24) What is the significance of C program algorithms?

Answer: The algorithm is created first and it contains step by step guidelines on how the solution should be. Also, it contains the steps to consider and the required calculations/operations within the program.

Q #29) Is it possible to use curly brackets ({}) to enclose a single line code in C program?

Answer: Yes, it works without any error. Some programmers like to use this to organize the code. But the main purpose of curly brackets is to group several lines of codes.

Q #30) Describe the modifier in C?

Answer: Modifier is a prefix to the basic data type which is used to indicate the modification for storage space allocation to a variable.

Example– In a 32-bit processor, storage space for the int data type is 4.When we use it with modifier the storage space change as follows:

- Long int: Storage space is 8 bit
- Short int: Storage space is 2 bit

Q #31) What are the modifiers available in C programming language?

Answer: There are 5 modifiers available in the C programming language as follows:

- Short
- Long
- Signed
- Unsigned
- long long

Q #33) Describe the newline escape sequence with a sample program?

Answer: The Newline escape sequence is represented by \n. This indicates the point that the new line starts to the compiler and the output is created accordingly.

Q #34) Is that possible to store 32768 in an int data type variable?

Answer: Int data type is only capable of storing values between – 32768 to 32767. To store 32768 a modifier needs to used with the int data type. Long Int can use and also if there are no negative values, unsigned int is also possible to use.

Q #35) Is there any possibility to create a customized header file with C programming language?

Answer: Yes, it is possible and easy to create a new header file. Create a file with function prototypes that are used inside the program. Include the file in the '#include' section from its name.

Q #37) Is that possible to add pointers to each other?

Answer: There is no possibility to add pointers together. Since pointer contains address details there is no way to retrieve the value from this operation.

Q #38) What is indirection?

Answer: If you have defined a pointer to a variable or any memory object, there is no direct reference to the value of the variable. This is called the indirect reference. But when we declare a variable, it has a direct reference to the value.

Q #39) What are the ways to a null pointer that can be used in the C programming language?

Answer: Null pointers are possible to use in three ways.

- As an error value.
- As a sentinel value.
- To terminate indirection in the recursive data structure.

42) Can the "if" function be used in comparing strings?

No. "if" command can only be used to compare numerical values and single character values. For comparing string values, there is another function called strcmp that deals specifically with strings.

43) What are preprocessor directives?

Preprocessor directives are placed at the beginning of every C program. This is where library files are specified, which would depend on what functions are to be used in the program. Another use of preprocessor directives is the declaration of constants.Preprocessor directives begin with the # symbol.

44) What will be the outcome of the following conditional statement if the value of variable s is 10?

s >=10 && s < 25 && s!=12

The outcome will be TRUE. Since the value of s is 10, s >= 10 evaluates to TRUE because s is not greater than 10 but is still equal to 10. s< 25 is also TRUE since 10 is less then 25. Just the same, s!=12, which means s is not equal to 12, evaluates to TRUE. The && is the AND operator, and follows the rule that if all individual conditions are TRUE, the entire statement is TRUE.

45) Describe the order of precedence with regards to operators in C.

Order of precedence determines which operation must first take place in an operation statement or conditional statement. On the top most level of precedence are the unary operators !, +, – and &. It is followed by the regular mathematical operators (*, / and modulus % first, followed by + and -). Next in line are the relational operators <, <=, >= and >. This is then followed by the two equality operators == and !=. The logical operators && and || are next evaluated. On the last level is the assignment operator =.

46) What is wrong with this statement? myName = "Robin";

You cannot use the = sign to assign values to a string variable. Instead, use the strcpy function. The correct statement would be: strcpy(myName, "Robin");

47) How do you determine the length of a string value that was stored in a variable?

To get the length of a string value, use the function strlen(). For example, if you have a variable named FullName, you can get the length of the stored string value by using this statement: I = strlen(FullName); the variable I will now have the character length of the string value.

48) Is it possible to initialize a variable at the time it was declared?

Yes, you don't have to write a separate assignment statement after the variable declaration, unless you plan to change it later on. For example: char planet[15] = "Earth"; does two things: it declares a string variable named planet, then initializes it with the value "Earth".

50) What are the different file extensions involved when programming in C?

Source codes in C are saved with .C file extension. Header files or library files have the .H file extension. Every time a program source code is successfully compiled, it creates an .OBJ object file, and an executable .EXE file.

51) What are reserved words?

Reserved words are words that are part of the standard C language library. This means that reserved words have special meaning and therefore cannot be used for purposes other than what it is originally intended for. Examples of reserved words are int, void, and return.

55) Not all reserved words are written in lowercase. TRUE or FALSE?

FALSE. All reserved words must be written in lowercase; otherwise the C compiler would interpret this as unidentified and invalid.

57) What would happen to X in this expression: X += 15; (assuming the value of X is 5)

X +=15 is a short method of writing X = X + 15, so if the initial value of X is 5, then 5 + 15 = 20.

58) In C language, the variables NAME, name, and Name are all the same. TRUE or FALSE?

FALSE. C language is a case sensitive language. Therefore, NAME, name and Name are three uniquely different variables.

59) What is an endless loop?

An endless loop can mean two things. One is that it was designed to loop continuously until the condition within the loop is met, after which a break function would cause the program to step out of the loop. Another idea of an endless loop is when an incorrect loop condition was written, causing the loop to run erroneously forever. Endless loops are oftentimes referred to as infinite loops.

60) What is a program flowchart and how does it help in writing a program?

A flowchart provides a visual representation of the step by step procedure towards solving a given problem. Flowcharts are made of symbols, with each symbol in the form of different shapes. Each shape may represent a particular entity within the entire program structure, such as a process, a condition, or even an input/output phase.

61) What is wrong with this program statement? void = 10;

The word void is a reserved word in C language. You cannot use reserved words as a user-defined variable.

62) Is this program statement valid? INT = 10.50;

Assuming that INT is a variable of type float, this statement is valid. One may think that INT is a reserved word and must not be used for other purposes. However, recall

that reserved words are express in lowercase, so the C compiler will not interpret this as a reserved word.

63) What are actual arguments?

When you create and use functions that need to perform an action on some given values, you need to pass these given values to that function. The values that are being passed into the called function are referred to as actual arguments.

65) What is output redirection?

It is the process of transferring data to an alternative output source other than the display screen. Output redirection allows a program to have its output saved to a file. For example, if you have a program named COMPUTE, typing this on the command line as COMPUTE >DATA can accept input from the user, perform certain computations, then have the output redirected to a file named DATA, instead of showing it on the screen.

66) What are run-time errors?

These are errors that occur while the program is being executed. One common instance wherein run-time errors can happen is when you are trying to divide a number by zero. When run-time errors occur, program execution will pause, showing which program line caused the error.

67) What is the difference between functions abs() and fabs()?

These 2 functions basically perform the same action, which is to get the absolute value of the given value. Abs() is used for integer values, while fabs() is used for floating type numbers. Also, the prototype for abs() is under <stdlib.h>, while fabs() is under <math.h>.

68) What are formal parameters?

In using functions in a C program, formal parameters contain the values that were passed by the calling function. The values are substituted in these formal parameters and used in whatever operations as indicated within the main body of the called function.

69) What are control structures?

Control structures take charge at which instructions are to be performed in a program. This means that program flow may not necessarily move from one statement to the next one, but rather some alternative portions may need to be pass into or bypassed from, depending on the outcome of the conditional statements.

71) When is a “switch” statement preferable over an “if” statement?

The switch statement is best used when dealing with selections based on a single variable or expression. However, switch statements can only evaluate integer and character data types.

72) What are global variables and how do you declare them?

Global variables are variables that can be accessed and manipulated anywhere in the program. To make a variable global, place the variable declaration on the upper portion of the program, just after the preprocessor directives section.

73) What are enumerated types?

Enumerated types allow the programmer to use more meaningful words as values to a variable. Each item in the enumerated type variable is actually associated with a numeric code. For example, one can create an enumerated type variable named DAYS whose values are Monday, Tuesday… Sunday.

74) What does the function toupper() do?

It is used to convert any letter to its upper case mode. Toupper() function prototype is declared in <ctype.h>. Note that this function will only convert a single character, and not an entire string.

75) Is it possible to have a function as a parameter in another function?

Yes, that is allowed in C programming. You just need to include the entire function prototype into the parameter field of the other function where it is to be used.

77) Which function in C can be used to append a string to another string?

The strcat function. It takes two parameters, the source string and the string value to be appended to the source string.

79) Do these two program statements perform the same output? 1) scanf("%c", &letter); 2) letter=getchar()

Yes, they both do the exact same thing, which is to accept the next key pressed by the user and assign it to variable named letter.

80) What are structure types in C?

Structure types are primarily used to store records. A record is made up of related fields. This makes it easier to organize a group of related data.

81) What does the characters "r" and "w" mean when writing programs that will make use of files?

"r" means "read" and will open a file as input wherein data is to be retrieved. "w" means "write", and will open a file for output. Previous data that was stored on that file will be erased.

82) What is the difference between text files and binary files?

Text files contain data that can easily be understood by humans. It includes letters, numbers and other characters. On the other hand, binary files contain 1s and 0s that only computers can interpret.

83) is it possible to create your own header files?

Yes, it is possible to create a customized header file. Just include in it the function prototypes that you want to use in your program, and use the #include directive followed by the name of your header file.

86) What is the general form of a C program?

A C program begins with the preprocessor directives, in which the programmer would specify which header file and what constants (if any) to be used. This is followed by the main function heading. Within the main function lies the variable declaration and program statement.

87) What is the advantage of a random access file?

If the amount of data stored in a file is fairly large, the use of random access will allow you to search through it quicker. If it had been a sequential access file, you would have to go through one record at a time until you reach the target data. A random access file lets you jump directly to the target address where data is located.

88) In a switch statement, what will happen if a break statement is omitted?

If a break statement was not placed at the end of a particular case portion? It will move on to the next case portion, possibly causing incorrect output.

89) Describe how arrays can be passed to a user defined function

One thing to note is that you cannot pass the entire array to a function. Instead, you pass to it a pointer that will point to the array first element in memory. To do this, you indicate the name of the array without the brackets.

90) What are pointers?

Pointers point to specific areas in the memory. Pointers contain the address of a variable, which in turn may contain a value or even an address to another memory.

91) Can you pass an entire structure to functions?

Yes, it is possible to pass an entire structure to a function in a call by method style. However, some programmers prefer declaring the structure globally, then pass a variable of that structure type to a function. This method helps maintain consistency and uniformity in terms of argument type.

92) What is gets() function?

The gets() function allows a full line data entry from the user. When the user presses the enter key to end the input, the entire line of characters is stored to a string variable. Note that the enter key is not included in the variable, but instead a null terminator \0 is placed after the last character.

93) The % symbol has a special use in a printf statement. How would you place this character as part of the output on the screen?

You can do this by using %% in the printf statement. For example, you can write printf("10%%") to have the output appear as 10% on the screen.

94) How do you search data in a data file using random access method?

Use the fseek() function to perform random access input/ouput on a file. After the file was opened by the fopen() function, the fseek would require three parameters to work: a file pointer to the file, the number of bytes to search, and the point of origin in the file.

95) Are comments included during the compilation stage and placed in the EXE file as well?

No, comments that were encountered by the compiler are disregarded. Comments are mostly for the guidance of the programmer only and do not have any other significant use in the program functionality.

96) Is there a built-in function in C that can be used for sorting data?

Yes, use the qsort() function. It is also possible to create user defined functions for sorting, such as those based on the balloon sort and bubble sort algorithm.

97) What are the advantages and disadvantages of a heap?

Storing data on the heap is slower than it would take when using the stack. However, the main advantage of using the heap is its flexibility. That's because memory in this structure can be allocated and remove in any particular order. Slowness in the heap can be compensated if an algorithm was well designed and implemented.

98) How do you convert strings to numbers in C?

You can write you own functions to do string to number conversions, or instead use C's built in functions. You can use atof to convert to a floating point value, atoi to convert to an integer value, and atol to convert to a long integer value.

99) Create a simple code fragment that will swap the values of two variables num1 and num2.

```
int temp;


temp = num1;


num1 = num2;


num2 = temp;
```

Question: Please explain what is a header file in C? What will happen if we include a header file twice in a C program?

Answer: Header files store the definitions and set of rules governing different built-in functions of the C programming language. For instance, the printf() and scanf() functions are defined in the stdio.h header file.

Every header file contains a set of predefined functions, meant to make programming in C simpler. You need to include the specific header file in your C program to be able to use the functions defined in it. For example, you can't use printf() and scanf() functions without including the stdio.h header file.

When a header file is included twice in a C program, the second one gets ignored. In actual, the #, called the include guard, preceding a header file ensures that it is included only once during the compilation process.

Question: What are some of the limitations of C language?

Answer: As everything has a finite potential, so the C language stands in no exception. The following are some of the drawbacks of C languages:

- Concept of OOPs: C language prohibits the concept of OOPs as it is based on the procedural approach. (Inheritance, Polymorphism, Encapsulation, Abstraction, Data Hiding).
- Run Time Checking: C language does not do the running checking which means that errors are not detected after every line of coding, but only once the complete coding is done making it inconvenient to correct the bugs
- Concept of the Namespace: C language does not exhibit the property of Namespace, so there cannot be two variables with the same name in the C language program.
- Lack of Exception Handling: The language doesn't exhibit the important feature of exception handling. The feature of exception handling doesn't allow the user to detect the errors and bugs while compiling the code.
- Insufficient Level for Abstraction: C language doesn't have a very wide data handling capacity, which poses a threat to the security of the language.

Question: What is the objective of the main () function in C?

Answer: The main () function in C language to the inlet to the C program. It is the entry point where the process of execution of the program starts. When the execution of the C program initiates, the control of the program is directed towards the main () function. It is mandatory that every C language program has a main () function. Although it is the function that indicates the programming process, it is not the first function to be executed.

Question: What can be understood by variable and constant?

Answer: In C language, both constant and variable is widely used while designing a program. The major difference is between variables, and constant is that variable can alter its assigned value at any point of the program. In contrast, the value of the constant remains unaltered during the entire program. The value of the constant is locked during the execution of the program. For example, the value of pi can be set as a constant during the entire course of the program.

Question: How are global variables different from static variables?

Answer: Global variables are variables with global scope, i.e., they are accessible throughout the program, unless shadowed. These variables are defined outside a function or code block. Static variables are variables allocated statically, i.e., their value can't be changed. It is fixed for the entire run of a program. They can be defined outside as well as inside functions. Moreover, they can be accessed from anywhere inside the program.

Question: Explain the role of protected access specifier?

Answer: The privacy of a protected keyword lies somewhere between the keywords private and public . If a class is marked as protected, it can be accessed by its member functions, classes derived with public or protected access, privately derived classes and friends of the class that declared these members.

Question: What is the keyword volatile used for?

Answer: Volatile prevents the compiler from optimizing the variable or object in question. Any code can change the variable's ((for example, thread)) value outside the scope of current code at any time. This implies that the compiler has to keep the value of a volatile variable in all the local copies of the variable.

Question: Explain the purpose of the 'delete' operator?

Answer: Delete removes all the objects created by the new expression, i.e. frees memory in the heap space. The array objects are deleted using the [] operator:

```
delete[] array;
NULL or void Pointer can be deleted as:
delete ptr;
The same is applicable for user-defined data types as well. For example,
int *var = new int;
delete var;
```

Question: Explain the purpose of extern storage specifier.

Answer: The extern storage specifier helps declare objects that can be used by many source files. It describes a variable that is externally defined. The definition can appear at the beginning of a block or outside a function. There is only one declaration of the extern variable. If any other instance is found, it is considered the same as the first one.

Extern variables can have block scope or file scope depending on where they are defined.

Question: Define preprocessor.

Answer: A preprocessor is a program that produces an output which is used by some other program as an input. For example, translation is a preprocessing step after which the code is sent for compilation.

Question: **List out the differences between reference and Pointer?**

Reference	Pointer
It is an alternative name for a variable.	Stores the address of a variable.
Declared using * : int *ptr.	Declared using & : int &refvar.
Cannot have null values.	Can have null values assigned.
Can be accessed through pass by value.	Uses pass by reference.
Must be initialized upon declaration, i.e. int &ref; will give an error.	No need for initialization during declaration itself, i.e. int *ptr is correct.
Shares same memory address as the original variable and takes up some space on the stack.	Has its size and memory address on the stack.

Question: Explain the difference between delete and delete[]?

Answer: Delete removes a single object from memory, whereas delete[] is used for de-allocating memory of an array of objects. The importance of having delete[] is that if we have a pointer (say ptr) to an array of size 10 (new myarr[10]) and simply give delete ptr, since we don't know how many objects ptr is pointing to, and thus delete will only delete the first item. The remaining 9 items will not be deleted. This will cause a memory leak. Example:

```
// delete
int *var = new int;
delete var;
// delete[]
```

```
int *arr = new int[1];
delete[] arr;
```

Question: Please explain the concept of Dangling Pointer in C? In how many ways can a pointer act as a Dangling Pointer?

Answer: A pointer that points to a memory location that is already deleted is called a dangling pointer. As per another definition, a dangling pointer is a pointer that points to a dereferenced memory location. A pointer acts as a dangling pointer in three cases:

1. Deallocation of memory
2. When the local variable isn't static
3. When the variable goes out of scope

Initially, the dangling pointer holds a valid memory address. Later, however, the held address is released.

The Dangling Pointer Problem:

When A pointer is pointing to a memory location, and a B pointer deletes the memory occupied by the A pointer, then the A pointer still points to the memory location, which is no longer available. This is called the dangling pointer problem.

Question: Compare arrays with pointers in the C programming language?

Answer: Following are the various differences between arrays and pointers in C:

- Definition - An array is a form of data structure that stores multiple, homogeneous elements at contiguous memory locations. A pointer, however, is a variable that stores or points to the memory address of some other variable.
- Initialization - It is possible to initialize arrays at the definition. Pointers, however, can't be initialized at the definition.
- Size - While a pointer can store the address of only a single variable, an array is capable of storing multiple elements. The total number of elements stored by arrays is determined by the array size.

Question: What are the important differences between a structure and a union in C?

Answer: Following are the three important differences between a structure and a union in the C programming language:

- For the same data type, a structure requires more memory than a union.
- Modifying the value of a member of a structure doesn't affect other members. Doing the same in a union, however, results in affecting all the members of the union.
- While only one element can be accessed at a time in the union, it is possible to access all elements of a structure simultaneously.

Question: How is pass by value different from a pass by reference?

Answer: Pass by value and pass by reference are also called call by value and call by reference, respectively. In the call by value, values are send/passed to the function as parameters. Pass by value is used when there is a requirement of not modifying the actual parameters. Address pertaining to the actual parameters are send/passed to the function in the call by reference. Pass by reference is used when there is a need for modifying the actual parameters. Changes made to the arguments in the called function are not reflected in the calling function in a pass by value. Opposite to this, changes made to the arguments in the called function are reflected in the calling function in a pass by reference.

Know about pass by reference in detail.

Question: How will you resolve the scope of a global symbol?

Answer: We can use the extern storage specifier for resolving the scope of a global symbol. The extern keyword is used for extending the visibility or scope of variables and functions. As C functions are visible throughout the program by default, its use with function declaration or definition is not required.

Question: When do we use the register keyword?

Answer: The register storage specifier, i.e., the register keyword, is used for storing a variable in the machine register. This is typically used for heavily-used variables to the likes of a loop control variable. The main intent behind using the register keyword is to speed-up the program by minimizing variable access time.

Question: What do you understand by rvalue and ivalue?

Answer: The expression on the left of the assignment operator (=) is called an ivalue. An rvalue is an expression on the right side of the assignment operator, and it is assigned to an ivalue.

For instance,

```
int a = 25;
```

int a is the ivalue in the above-mentioned example while 25 is the rvalue. While an ivalue persists beyond a single expression, the rvalue doesn't persist beyond the expression using it.

.**Question: Please explain tokens in C.**

Answer: Tokens are the smallest, indivisible units of a C program with distinct meanings. Following are the various types of tokens in C:

- Constants - Fixed values that can't be changed during the program execution.
- Identifiers - This refers to the name of the functions, variables, arrays, structures, etc.

- Keywords/Reserved Names - Predefined words with special meanings that can't be used as variable names.
- Operators - Symbols that tell the C compiler to perform specific logical, mathematical, or relational operations.
- Special Characters - All characters excluding the alphabets and digits are special characters.

Question: Briefly explain the various file opening modes in C?

Answer: Files are opened in a C program using the fopen() function. It is defined in the stdio.h header file. The general syntax of fopen() is:

```
ptr = fopen ("file address", "mode");
```

It is possible to open a file in the following 12 different opening modes in a C program:

1. r - Opens a file for reading.
2. rb - Opens a file for reading in binary mode.
3. w - Opens a file for writing.
4. wb - Opens a file for writing in binary mode.
5. a - Opens a file for appending i.e. adding data to the end of the file.
6. ab - Opens a file for appending in binary mode.
7. r+ - Opens a file for both reading and writing.
8. rb+ - Opens a file for both reading and writing in binary mode.
9. w+ - Opens a file for both reading and writing.
10. wb+ - Opens a file for both reading and writing in binary mode.
11. a+ - Opens a file for both reading and appending.
12. ab+ - Opens a file for both reading and appending in binary mode.

Leave for r, rb, r+, and rb+ modes; the file is created if it is not found when tried to open in other modes. The fopen() returns NULL if the file doesn't exist in these 4 file opening modes.

Question: Compare local variables and global variables.

Answer: Following are the various differences between the local and global variables:

1. Declaration

Local variables are declared inside a function, while global variables are the variables declared outside the functions.

2. Life

The life of a local variable is tied with the function block where it is declared. The variable is destroyed when the function exits. Global variables remain for the entire lifetime of the program.

3. Scope

The scope of a local variable is confined to the function or code block where it is declared. Global variables have global scope, i.e., they are available throughout the program.

4. Storage

Unless specified explicitly, local variables are stored in a stack. The storage for a global variable is decided by the compiler itself.

Question: What is a far pointer in C?

Answer: A far pointer is a 32-bit pointer capable of accessing all the 16 segments, i.e., the whole residence memory of RAM. It can access information outside the computer memory in a given segment. To use the far pointer, it is required to:

- Allocate the sector register to store data address in the segment, and
- Store another sector register within the most recent sector

Question: Please explain the auto keyword in C.

Answer: auto is the default storage class of all the variables declared inside a code block or function. Hence, local variables can also be referred to as automatic or auto variables. If no value is stored in an auto variable, then it gets a garbage value. Auto variables are called so because these variables allocate and deallocate memory upon entering and exiting the code block or function in which they are declared, respectively. Typically, there is no need to mention the auto keyword explicitly.

Question: Explain the difference between near, far, and huge pointers.

Answer: Any virtual address has the selector and offset. While a near pointer doesn't have explicit selector, far and huge pointers do. Performing pointer arithmetic on the far pointer doesn't result in modifying the selector. It does, however, in the case of a huge pointer.

Question: Please define typecasting.

Answer: Typecasting is the process of converting one data type into another. It is of two types:

1. Implicit type casting - Also known as an automatic conversion, implicit type conversion is performed automatically by the C compiler, i.e., it doesn't require a casting operator. For example:

```
#include <stdio.h>
#include <conio.h>
void main ()
```

```
{
   int x = 22;
   float b = x; //implicit type conversion
   printf("%f", b);
}
```

Output: 22.000000

2. Explicit type casting - Unlike implicit type conversion, explicit type casting is performed by the programmer. A type casting operator is used for telling the compiler to convert (cast) one data type into another. For example:

```
#include <stdio.h>
#include <conio.h>
void main ()
{
   float x = 22.22;
   int b = (int) x; //explicit type conversion
   printf("%d", b);
}
```

Output: 22

Here, (int) is the typecasting operator.

Question: What are the distinct ways of passing parameters to the function and state along with which is to be used when?

Answer: There are primarily two ways of passing parameters to the function which are listed as follows:

- Call by Reference- In this technique, the address of the actual parameter is sent instead of the values. It is chosen in the case when the user doesn't want the actual parameter to be altered with the formal parameter.
- Call by Value- In this technique, we only send values to the functions as the parameters. It is chosen in the case the user does want the actual parameter to be altered with the formal parameter but just to be used.

Question: What is the utilization of #undef preprocessor?

Answer: As per the C programming language, the main aim of the #undef directive is that it acts as a guide to the preprocessor to get rid of all the definitions for the particular macro. If a macro is non-specific, and #ifdef directive on that specified macro will show the result as false.

Question: Elaborate the process by which we can apply quote character (and) in output screen in C programming?

Answer: We all know that applying quote character is the part of printf statement an in order to apply this character in output screen, you can use the format of specifiers/ ‘ this is for a single quote and /” for a double quote.

Question: What is the use of modulus operation in C programming?

Answer: Modulus operator in C programming is very important operator in this programming because this modulus operation tell the division operation in the programming function. It basically shows the remainder value left after the division between two numbers. It is represented with the symbol of (%).

For example:

1. 20 % 3 = 2, this will be expressed as, when we divided 20 by 3 we are left with the remainder of 2.
2. 11 % 2 =1, this means that when we divided 11 by 2 we get the remainder of 1.

Question: What is the general form that is supported by C programming? Explain its concept in detail.

Answer: It commences its process with the preprocessor directives and these directives perform a lot of function like it responsible for specifying the header files and it also tells about the constants to be used in the C programming process. This whole process is followed by the main function of this programming and that is the heading function. This function performs the variable declaration and the most important program statements.

Question: Random file access in C programming plays a very important part. Why? list some of its merits.

Answer - We all know that in every computer we store a large amount of significant information and every time we take a lot of time in finding the key content from the large database but this random access file helps in solving this problem.

1. It helps to find the key data quicker from a large database.
2. It helps indirectly shifting to target address when there are a lot of addresses in the programming.

Question: What is the role of structure types in C programming?

Answer: Major role played by structure types in C programming is to store the records. These records are stored according to their field types. This helps us to easily locate the record at the time of urgent search because we will find all the records in a particular field folder. This helps to save time and money.

Question: Is C language is a middle-level language? If yes then why is it so?

Answer: Yes C language is known to be a middle-level language. This is because it has a very unique and interesting feature that makes it to act like a higher-level language. But at the same time, it facilitates to use low-level methods in interacting with hardware. It also supports the use of the English type of words in its functioning. All such feature makes it to act as high-level language. Moreover, it has a unique feature of storing the memory structure.

Question: What are the functions of reserve words in C programming? Elaborate on them.

Answer: Standard C language library has certain words stored in it all such words are called reserve words. All such words are unique and have a different meaning so they cannot be used for any other kind of purpose rather than its original purpose. Some of the examples are void and return etc.

Question: What are the different control structures that is followed in the C programming?

Answer: In C programming we usually follow three types of control structures and that are sequence, selection and repetition.

1. Sequence: this control structure follows the path that runs from a top to bottom in the process of execution of a program and this flow is in sequence.
2. Selection: In this, the codes are executed on the conditions of evaluation as being true or false.
3. Repetition: In this, the program statement will follow the steps of repetition. The other name of repetition is called a loop structure.

Question: What is the exact meaning of debugging in C programming?

Answer: Debugging means the process that helps in allocating the errors in the programming. It may stop the process of execution. In order to solve this process of execution, debugging will identify those errors and will remove those errors and complications. So it plays a very important role in the execution process of C programming. So it plays a very important role in the execution process of C programming..

Question: What is the exact meaning of C language in computer science?

Answer: C is a middle-level language that is also known as a structural programming language. because it has a very unique feature that makes it to act like a higher-level language. But at the same time, it facilitates to use low-level methods in interacting with hardware. It also supports the use of the English type of words in its functioning. All such feature makes it to act as high-level language. Moreover, it has a unique feature of storing the memory structure.

Question: Explain fast speed and memory management as a unique feature of C language?

Answer:

1. Fast speed: The operators and data types used in C programming are very effective and powerful that increases the speed of working. Hence it helps in fastening up the speed.
2. Memory management: It has a very unique feature of in build memory that can handle a large amount of data very effectively and efficiently. This helps in improving the overall performance of C programming.

Question: Explain the meaning and role of the auto keyword in the C programming?

Answer: Every local variable function present in C programming is called automatic or auto variable. It is very important to know that local variables are those variable that is inside the function block. The other name of the local variable is the auto variable. If the local value does not hold any key value in it then it will only show the garbage value.

Question: What do you know about logical errors and bring out the difference between syntax and logical errors.

Answer:

1. Logical errors: Logical errors always take place in complication process and can pass it very easily but in that case, we will not get the desired results. Such errors occurred when we pass the wrong formula into code.
2. Syntax error: The mistakes that occur in the programming language is called syntax error. It usually takes place when the command is misspelt by the program and because of that, it performs the wrong function. It can also take place when the command is to be given in lower case but by mistake, it is given in the upper case. Moreover, sometimes we make use of wrong symbols. All such problem and mistakes bring errors in the C programming.

Question: What is the exact role played by sequential access file in C programming?

Answer - Its main function in C programming is to store and retrieve the data in the file and then transforming that file into different forms. These are always arranged and stored in sequential order. In order to get access to one particular file, we are facilitated by reading only one data at a time.

Question: What is the importance of linked list in the C programming?

Answer: When nodes are connected with each other in a list that list is called a linked list. This list can only be created by using the pointers. This helps in making efficient use of the memory space in a program.

Question: Explain the meaning of queue and FIFO in C programming?

Answer: Queue is a kind of data structure present in the C programming and all the data present in this queue is stored in the format called FIFO. The full form of FIFO is first-in-first-out. In every queue, the first data is available on the first line.

Question: How binary trees are similar to the linked lists?

Answer: Here binary trees are the extension of the linked lists and linked lists are the nodes which are connected with each other in a list that list is called a linked list. This list can only be created by using the pointers. This helps in making efficient use of the memory space in a program. The binary list also takes the use of two-pointers and those pointers are right and left pointers. So basically every node has two pointers at the ends.

What is a pointer on pointer?

It's a pointer variable which can hold the address of another pointer variable. It de-refers twice to point to the data held by the designated pointer variable.

Eg: int x = 5, *p=&x, **q=&p;

Therefore 'x' can be accessed by **q.

Distinguish between malloc() & calloc() memory allocation.

Both allocates memory from heap area/dynamic memory. By default calloc fills the allocated memory with 0's.

What is keyword auto for?

By default every local variable of the function is automatic (auto). In the below function both the variables 'i' and 'j' are automatic variables.

```
void f() {
  int i;
  auto int j;
}
```

NOTE − A global variable can't be an automatic variable.

What is a NULL pointer?

A pointer pointing to nothing is called so. Eg: char *p=NULL;

What is the purpose of extern storage specifier?

Used to resolve the scope of global symbol.

Eg:

```
main() {
   extern int i;
   Printf("%d",i);
}
int i = 20;
```

Explain the purpose of the function sprintf().

Prints the formatted output onto the character array.

What is the meaning of base address of the array?

The starting address of the array is called as the base address of the array.

When should we use the register storage specifier?

If a variable is used most frequently then it should be declared using register storage specifier, then possibly the compiler gives CPU register for its storage to speed up the look up of the variable.

S++ or S = S+1, which can be recommended to increment the value by 1 and why?

S++, as it is single machine instruction (INC) internally.

What is a dangling pointer?

A pointer initially holding valid address, but later the held address is released or freed. Then such a pointer is called as dangling pointer.

What is the purpose of the keyword typedef?

It is used to alias the existing type. Also used to simplify the complex declaration of the type.

What is the difference between actual and formal parameters?

The parameters sent to the function at calling end are called as actual parameters while at the receiving of the function definition called as formal parameters.

Can a program be compiled without main() function?

Yes, it can be but cannot be executed, as the execution requires main() function definition.

What is the advantage of declaring void pointers?

When we do not know what type of the memory address the pointer variable is going to hold, then we declare a void pointer for such.

Where an automatic variable is stored?

Every local variable by default being an auto variable is stored in stack memory.

What is a nested structure?

A structure containing an element of another structure as its member is referred so.

What is the difference between variable declaration and variable definition?

Declaration associates type to the variable whereas definition gives the value to the variable.

What is a self-referential structure?

A structure containing the same structure pointer variable as its element is called as self-referential structure.

Does a built-in header file contains built-in function definition?

No, the header file only declares function. The definition is in library which is linked by the linker.

What is a preprocessor?

Preprocessor is a directive to the compiler to perform certain things before the actual compilation process begins.

Explain the use of %i format specifier w.r.t scanf().

Can be used to input integer in all the supported format.

How can you print a \ (backslash) using any of the printf() family of functions.

Escape it using \ (backslash).

Does a break is required by default case in switch statement?

Yes, if it is not appearing as the last case and if we do not want the control to flow to the following case after default if any.

When to user -> (arrow) operator.

If the structure/union variable is a pointer variable, to access structure/union elements the arrow operator is used.

What are bit fields?

We can create integer structure members of differing size apart from non-standard size using bit fields. Such structure size is automatically adjusted with the multiple of integer size of the machine.

What are the different ways of passing parameters to the functions? Which to use when?

Call by value − We send only values to the function as parameters. We choose this if we do not want the actual parameters to be modified with formal parameters but just used.

Call by reference − We send address of the actual parameters instead of values. We choose this if we do want the actual parameters to be modified with formal parameters.

What is the purpose of built-in stricmp() function.

It compares two strings by ignoring the case.

Describe the file opening mode "w+".

Opens a file both for reading and writing. If a file is not existing it creates one, else if the file is existing it will be over written.

Where the address of operator (&) cannot be used?

It cannot be used on constants.

It cannot be used on variable which are declared using register storage class.

Is FILE a built-in data type?

No, it is a structure defined in stdio.h.

What is reminder for 5.0 % 2?

Error, It is invalid that either of the operands for the modulus operator (%) is a real number.

How many operators are there under the category of ternary operators?

There is only one operator and is conditional operator (? :).

Which key word is used to perform unconditional branching?

goto

What is a pointer to a function? Give the general syntax for the same.

A pointer holding the reference of the function is called pointer to a function. In general it is declared as follows.

T (*fun_ptr) (T1,T2…); Where T is any date type.

Once fun_ptr refers a function the same can be invoked using the pointer as follows.

fun_ptr();

[Or]

(*fun_ptr)();

Explain the use of comma operator (,).

Comma operator can be used to separate two or more expressions.

Eg: printf(“hi”) , printf(“Hello”);

What is a NULL statement?

A null statement is no executable statements such as ; (semicolon).

Eg: int count = 0;

while(++count<=10) ;

Above does nothing 10 times.

What is a static function?

A function’s definition prefixed with static keyword is called as a static function. You would make a function static if it should be called only within the same source code.

Which compiler switch to be used for compiling the programs using math library with gcc compiler?

Opiton –lm to be used as > gcc –lm <file.c>

Which operator is used to continue the definition of macro in the next line?

Backward slash (\) is used.

E.g. #define MESSAGE "Hi, \

Which operator is used to receive the variable number of arguments for a function?

Ellipses (…) is used for the same. A general function definition looks as follows

void f(int k,…) {

}

What is the problem with the following coding snippet?

char *s1 = "hello",*s2 = "welcome";

strcat(s1,s2);

s1 points to a string constant and cannot be altered.

Which built-in library function can be used to re-size the allocated dynamic memory?

realloc().

Define an array.

Array is collection of similar data items under a common name.

What are enumerations?

Enumerations are list of integer constants with name. Enumerators are defined with the keyword enum.

Which built-in function can be used to move the file pointer internally?

fseek()

What is a variable?

A variable is the name storage.

C is successor of which programming language?

B

What is the full form of ANSI?

American National Standards Institute.

Which operator can be used to determine the size of a data type or variable?

sizeof

Can we assign a float variable to a long integer variable?

Yes, with loss of fractional part.

Is 068 a valid octal number?

No, it contains invalid octal digits.

What it the return value of a relational operator if it returns any?

Return a value 1 if the relation between the expressions is true, else 0.

How does bitwise operator XOR works.

If both the corresponding bits are same it gives 0 else 1.

What is an infinite loop?

A loop executing repeatedly as the loop-expression always evaluates to true such a

```
while(0 == 0) {
}
```

Can variables belonging to different scope have same name? If so show an example.

Variables belonging to different scope can have same name as in the following code snippet.

```
int var;

void f() {
  int var;
}

main() {
  int var;
}
```

What is the default value of local and global variables?

Local variables get garbage value and global variables get a value 0 by default.

Can a pointer access the array?

Pointer by holding array's base address can access the array.

What are valid operations on pointers?

The only two permitted operations on pointers are

Comparision ii) Addition/Substraction (excluding void pointers)

What is a string length?

It is the count of character excluding the '\0' character.

What is the built-in function to append one string to another?

strcat() form the header string.h

Which operator can be used to access union elements if union variable is a pointer variable?

Arrow (->) operator.

Explain about 'stdin'.

stdin in a pointer variable which is by default opened for standard input device.

Name a function which can be used to close the file stream.

fclose().

What is the purpose of #undef preprocessor?

It be used to undefine an existing macro definition.

Define a structure.

A structure can be defined of collection of heterogeneous data items.

Name the predefined macro which be used to determine whether your compiler is ANSI standard or not?

__STDC__

What is typecasting?

Typecasting is a way to convert a variable/constant from one type to another type.

What is recursion?

Function calling itself is called as recursion.

Which function can be used to release the dynamic allocated memory?

free().

What is the first string in the argument vector w.r.t command line arguments?

Program name.

How can we determine whether a file is successfully opened or not using fopen() function?

On failure fopen() returns NULL, otherwise opened successfully.

What is the output file generated by the linker.

Linker generates the executable file.

What is the maximum length of an identifier?

Ideally it is 32 characters and also implementation dependent.

What is the default function call method?

By default the functions are called by value.

Functions must and should be declared. Comment on this.

Function declaration is optional if the same is invoked after its definition.

When the macros gets expanded?

At the time of preprocessing.

Can a function return multiple values to the caller using return reserved word?

No, only one value can be returned to the caller.

What is a constant pointer?

A pointer which is not allowed to be altered to hold another address after it is holding one.

To make pointer generic for which date type it need to be declared?

Void

Can the structure variable be initialized as soon as it is declared?

Yes, w.r.t the order of structure elements only.

Is there a way to compare two structure variables?

There is no such. We need to compare element by element of the structure variables.

Which built-in library function can be used to match a patter from the string?

Strstr()

What is difference between far and near pointers?

In first place they are non-standard keywords. A near pointer can access only 2^15 memory space and far pointer can access 2^32 memory space. Both the keywords are implementation specific and are non-standard.

Can we nest comments in a C code?

No, we cannot.

Which control loop is recommended if you have to execute set of statements for fixed number of times?

for – Loop.

What is a constant?

A value which cannot be modified is called so. Such variables are qualified with the keyword const.

Can we use just the tag name of structures to declare the variables for the same?

No, we need to use both the keyword ‘struct’ and the tag name.

Can the main() function left empty?

Yes, possibly the program doing nothing.

Can one function call another?

Yes, any user defined function can call any function.