CIT-112 Viva Questions for Lab Final Exam (2)

Curated by Kazi Rifat Morshed · GitHub
CSE, KU

1

Top 75 C Interview Questions and Answers 2023 | Teachics

Source: https://teachics.org/interview-questions/c-interview-questions-and-answers/

Top 75 C Interview Questions and Answers

C is a procedural programming language initially developed by Dennis Ritchie at Bell Laboratories in 1972. The syntax of many programming languages including Java, PHP, and JavaScript is primarily based on the C language.

Here are the most important **C interview questions and answers**. The questions are divided into three sections; Basic C Interview Questions and Answers, Intermediate C Interview Questions and Answers, and Advanced C Interview Questions and Answers.

Beginner C Interview Questions and Answers

#1. What is C?

C is a high-level, general-purpose, procedural programming language.

#2. Who developed C language?

Dennis Ritchie.

#3. When was C language developed?

C language was developed in 1970 at Bell Laboratories.

#4. What is a C token?

The smallest individual units in C are known as C tokens. There are six types of tokens in C,

- Keywords
- Constants
- Identifiers
- Strings
- Operators
- Special symbols.

#5. What is a C keyword?

Predefined reserved words with fixed meanings are called keywords. Keywords are the building blocks for program statements.

#6. How many reserved keywords are there in C?

There are 32 reserved keywords in C. They are

auto	break	case	char
const	continue	do	default
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

#7. What is an identifier in C?

The names of variables, functions, and arrays are referred to as identifiers. These are user-defined names consisting of a sequence of letters and digits. Identifiers must be unique.

#8. What is a variable?

A variable is a user-defined name given to memory locations that can be used to store data. A variable may take different values at different times during the execution of a program.

#9. What do you mean by the scope of a variable?

The part of the code where a declared variable can be accessed directly is called the scope of the variable.

#10. What is a constant?

Fixed values that do not change throughout program execution are called constants.

#11. What is meant by variable initialization?

Assigning a value to a variable once declared is called the initialization of a variable.

#12. What are the data types present in C?

C supports five primary (fundamental) datatypes namely integer(int), character(char), floating-point(float), double-precision floating-point(double) and void(void).

Derived types are data types that are derived from fundamental data types. Arrays, pointers, function types, structures, and so on are examples.

User-defined datatypes – structures, unions, enum.

#13. What is an operator?

An operator is a symbol that instructs a computer to perform certain operations. Operators are typically used as part of mathematical or logical expressions.

#14. What are the operators available in C?

Aritmetic Operators	+ – * / %
Relational Operators	< <= > >= == !=
Logical Operators	&& | !
Assignment Operator	+= -= *= /= %=
Increment and Decrement Operator	++ —
Bitwise Operator	& | ^ ` << >>
Conditional Operator	?:
Special Operator	sizeof . ->

#15. What is operator precedence?

The order in which C evaluates expressions is called operator precedence.

#16. What is the modulus(%) operator?

The modulo division (%) operator produces the remainder of an integer division.

#17. What is the difference between integer arithmetic, real arithmetic, and mixed-mode arithmetic?

When the operands in a single arithmetic expression are both integers, then the operation is known as **integer arithmetic**. Integer arithmetic always produces an integer value as output. All the examples in the above table are integer arithmetic. For example: `12 / 10 = 1`

Real arithmetic refers to arithmetic operations that will use only real operands. The operator % cannot be used with real operands. For example: `3.0 / 2.0 = 1.5`

Mixed-mode arithmetic expressions are those in which one of the operands is real and the other is an integer. Since one of the operands is of real type, then the result is always a real number. For example: `12 / 10.0 = 1.2`

#18. What is the difference between = and == operator in C?

`==` is the comparison operator used to check whether the value or expression on the

#19. What is the use of printf() and scanf() functions in C?

The `scanf()` function is used to input data through the keyboard. It is a general input function available in C. The `printf()` function can be used to display values of variables and results of expressions on the screen.

#20. What are the different forms of if statements that can be implemented in C?

- Simple if statement.
- if….else statement.

- Nested if….else statement.
- else if ladder.

#21. What is looping?

The process of executing a sequence of statements repeatedly until some conditions for termination of the loop is satisfied is called looping.

#22. What are the loop control statements provided in C?

- The `while` statement.
- The `do` statement.
- The `for` statement.

#23. What is the use of the break statement?

The `break` statement can be used to accomplish an early exit from a loop. When a `break` statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop.

#24. What is the use of the continue statement?

The `continue` statement is used to skip a part of a loop. It causes the loop to be continued with the next iteration after skipping any statements in between.

#25. What is an array?

An Array is a collection of variables of a similar type that are referenced by a common name. There are three types of arrays, namely,

- One Dimensional Array
- Two Dimensional Array
- Multi DImensional Array

#26. Define String?

A sequence of characters that are treated as a single data item is called a string.

#27. What is the important string handling functions in C?

- `strcpy()` – String assignment function.
- `strlen()` – Returns the number of characters in the string.
- `strcmp()` – Compares two strings.
- `strcat()` – Concatenate two strings.
- `strstr()` – Used to locate substring.

#28. What is a function?

A block of program code that performs a particular task is called a function.

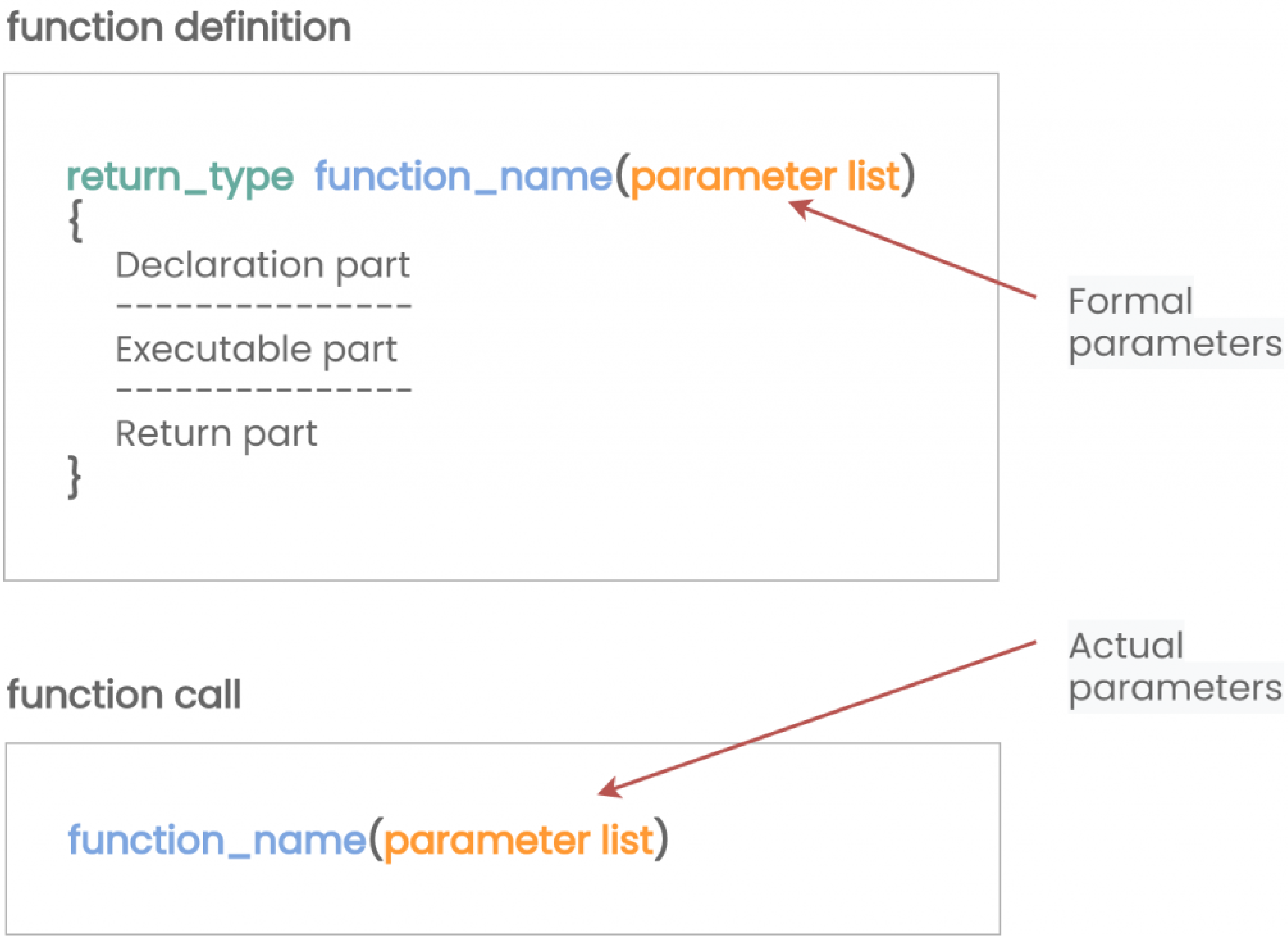
#29. What is the difference between library functions and user-defined functions?

The functions that are already defined in the C library are called library functions(Built-in functions). They are stored in different header files and these functions cannot be modified by the user. To use a library function in a program, the user has to include the corresponding header file in the program. Some of the library functions in C include `printf()`, `scanf()`, `getch()`, `sqrt()`, etc.

The function that has to be defined by the user at the time of writing a program is called **user-defined functions**.

#30. What is the difference between a formal parameter and an actual parameter?

The parameters used in function definitions are called formal parameters and those used in function calls are called actual parameters. The formal parameters list declares the variables that will receive the data sent by the calling program.



#31. What is recursion?

A technique where a function calls itself is known as recursion.

#32. Distinguish between automatic and static variables.

All local variables that are declared inside a function are known as auto variables unless not specified. That is, by default a local variable is an auto variable. An auto variable is created each time when the function is called and destroyed when the program's execution leaves the function.

A static variable is similar to an automatic variable. A static variable is declared once and only destroys when the execution of the program finishes.

Automatic Variables	Static Variables
All local variables are automatic by default. Using the keyword auto is optional.	static keyword must be used to declare a static varibale.
The scope of an automatic variable is always local to the function in which it is declared.	The scope of an static variable is always local to the function.
Automatic variables are created when the function is called.	A static variable is only initialized once, when the program is compiled.
Automatic variables are destroyed when the execution of the function is completed.	The value of a static variable persists until the end of the program.

#33. What do you mean by scope, visibility, and lifetime of a variable?

The region of a program in which a variable is available for use is called the scope of the variable.

The ability of a program to access a variable from the memory is the visibility of a variable.

The duration of time in which a variable exists in the memory during execution is the lifetime of a variable.

#34. How does a structure differ from an array?

An array is a collection of data elements of the same type. Structures can have elements of different types.

An array is a derived data type whereas a structure is a user-defined data type.

An array can be used as a built-in datatype. All we have to do is declare and use the array. But in the case of a structure, we have to design and declare it as a data structure before the structure variables are declared and used.

#35. Distinguish between a structure and union.

Both structure and union are user-defined data types in C. The major difference between a structure and a union is in terms of storage. In structures, each member has its own storage location, but all the members of a union share the same location. That is a union can handle only one member at a time.

#36. What are the size of a structure and a union?

The total size of the structure is the sum of the size of every data member. The total size of a union is the size of the largest data member.

#37. What is meant by nested structures and arrays of structures?

The individual members of a structure can be other structures as well, such structure is called a nested structure. That is, a structure may contain another structure as its member.

It is possible to declare an array of structures. The array will have individual structures as its elements.

#38. What is a pointer?

Pointers are a special type of variable that is used to store the address of another variable as their values.

#39. What is type conversion?

The process of converting a variable of one data type into another is referred to as typecasting. Casting allows you to make this type of conversion explicit, or to force it when it wouldn’t normally happen.

#40. What is the difference between a void pointer and a null pointer?

A pointer that holds a null value is called a null pointer. When a pointer has a null value it is not pointing anywhere. A void pointer is a type of pointer that points to some data location in storage, which doesn't have any specific type.

That is null pointer is a value, while the void pointer is a type.

#41. How does an append mode differ from a write mode in file management?

When the mode is 'write', a file is created if the file does not exist. If that file already exists, then the contents are deleted before starting writing.

When the mode is 'append', the file is opened with current contents safe. If the file does not exist, then a file with the specified name is created.

#42. What is the significance of EOF in files?

'EOF' is a specific designation for a file marker that indicates the end of the data.

#43. What is dynamic memory allocation?

The process of allocating memory at runtime is known as dynamic memory allocation.

#44. What is the difference between a macro and a function in C?

Macros	Functions
All the macros will be processed before the program compiles.	Functions are not preprocessed but compiled.
Macros do not check for compilation errors which leads to unexpected output.	Function checks for compilation errors.
Before Compilation, the macro name is replaced by macro value.	During function call, transfer of control takes place.
Faster in execution.	A bit slower in execution than Macros.

#45. What is a double pointer?

When a pointer holds the address of another pointer then such type of pointer is known as double pointer.

Advanced C Interview Questions and Answers

#46. What are the two ways of passing parameters to functions?

- Pass by Value – The values of actual parameters are copied to the variables in the parameter list of the called function. Any changes to the parameter in the called function have no effect on data in the calling function. That is the called function works on the copy and not on the original values.
- Pass by Reference – The memory addresses of the variable rather than the copies of values are sent to the called function. Thus any changes made to the parameter are also made to the original variable.

#47. What are the rules for initializing structures?

The rules to keep in mind while initializing structure variables are,

- We cannot initialize individual data members inside the structure definition.
- The order of values enclosed in braces must match the order of members in the structure definition.
- It is permitted to initialize only the first few members and leave the remaining blank. The uninitialized members should be at the end of the list.
- The uninitialized members will have default values based on the data type.

#48. What is conditional compilation?

Conditional compilation is a feature of the C preprocessor which can be used to switch on or off a particular line or group of lines in a program. The compiler control directives used for this are,

- `#if`
- `#else`
- `#elif`
- `#endif`
- `#ifdef`
- `#ifndef`
- `#undef`
- `#pragma`
- `#error`

#49. What is the difference between functions and parameterized macros?

Macros	Functions
All the macros will be processed before the program compiles.	Functions are not preprocessed but compiled.
Macros do not check for compilation errors which leads to unexpected output.	Function checks for compilation errors.
Before Compilation, the macro name is replaced by macro value.	During function call, transfer of control takes place.
Faster in execution.	A bit slower in execution than Macros.

#50. What are the typical applications of pointers in developing programs?

- To pass arguments by reference.
- For accessing array elements.
- To return multiple values.
- Dynamic memory allocation.
- To implement data structures.
- To do system-level programming where memory addresses are useful.

#51. What are near, far, and huge pointers?

Near pointer is used to store 16-bit addresses means within the current segment on a 16-bit machine. The limitation is that we can only access 64kb of data at a time.
A far pointer is typically 32-bit, which includes a segment selector, making it possible to point the addresses outside of the default segment.
The huge pointer is also 32-bit and can access outside segments. In far pointer, the segment part cannot be modified, but in Huge it can be.

#52. What is a dangling pointer in C?

A pointer pointing to a non-existing memory location is called a dangling pointer. A dangling pointer is a pointer that has a value that refers to some memory location that is not valid or does not exists.

#53. What is a wild pointer in C?

A pointer in C that is not initialized either by the compiler or programmer is known as a wild pointer. Uninitialized pointers behavior is totally undefined because it may point to some arbitrary location that can be a cause of the program crash.

#54. What is the difference between malloc() and calloc()?

`malloc()` allocates the requested size of bytes and returns a pointer to the first byte of the allocated space.

`calloc()` allocates space for an array of elements, initializes them to zero, and then returns a pointer to the memory.

#55. What do you mean by Memory Leak?

A memory leak occurs when memory is allocated but not released back to the operating system. Memory leakage increases unwanted memory usage.

#56. What are command line arguments?

Command line arguments are simply arguments that are specified after the name of the program in the system's command line, and these argument values are passed on to your program during program execution.

2

Top 40 C Programming Interview Questions and Answers

Source : https://www.softwaretestinghelp.com/c-programming-interview-questions/

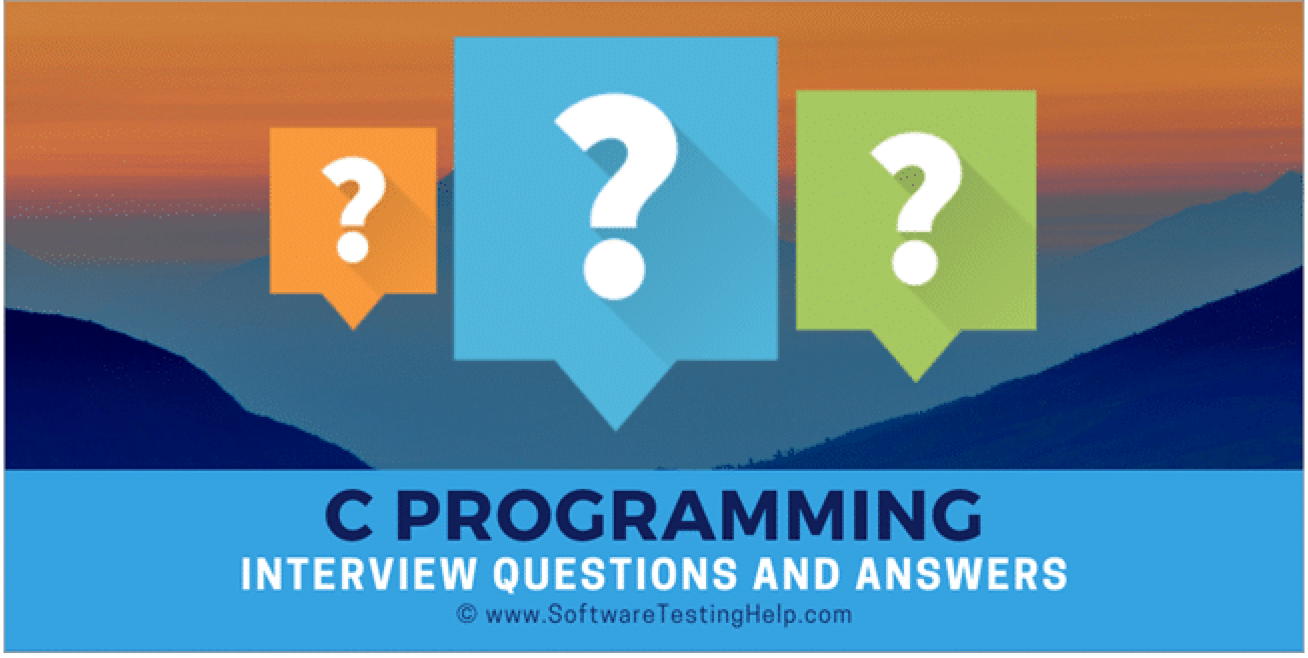
WARNING: THIS FOLLOWING SEGMENT(copied from softwaretestinghelp website) HAS GOT SOME ISSUES (SOME TEXT?CODE OR PICTURES MAY BE MISSING), IT IS SUGGESTED TO VISIT THE MAIN SOURCE (EVEN EXPORT AS PDF HAS SUCH GAPS)
SOURCE: https://www.softwaretestinghelp.com/c-programming-interview-questions/

June 28, 2023

Most frequently Asked C Programming Interview Questions and Answers:

C programming language was developed between 1969 and 1973 by Dennis Ritchie at Bell Labs. He uses this new programming language to re-implement UNIX operating system.

C is a high-level structured oriented programming language used for general-purpose programming requirements. Basically, C is a collection of its library functions. It is also flexible to add user-defined functions and include those in the C library.


C PROGRAMMING
INTERVIEW QUESTIONS AND ANSWERS
© www.SoftwareTestingHelp.com

The main usage of C programming language includes Language Compilers, Operating Systems, Assemblers, Text Editors, Print Spoolers, Network Drivers, Modern Programs, Data Bases, Language Interpreters, and Utilities.

Most Common C Programming Interview Questions

Here we go.

Q #1) What are the key features in the C programming language?

Answer: Features are as follows:

- **Portability**: It is a platform-independent language.
- **Modularity:** Possibility to break down large programs into small modules.
- **Flexibility:** The possibility of a programmer to control the language.
- **Speed:** C comes with support for system programming and hence it compiles and executes with high speed when compared with other high-level languages.
- **Extensibility**: Possibility to add new features by the programmer.

Q #2) What are the basic data types associated with C?

Answer:

- **Int** – Represent the number (integer)
- **Float** – Number with a fraction part.
- **Double** – Double-precision floating-point value
- **Char** – Single character
- **Void** – Special purpose type without any value.

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 #4) What is the process to create increment and decrement statement in C?

Answer: There are two possible methods to perform this task.

- Use increment (++) and decrement (-) operator.

Example When x=4, x++ returns 5 and x- returns 3.

- Use conventional + or – sign.

Example When x=4, use x+1 to get 5 and x-1 to get 3.

Q #5) What are reserved words with a programming language?

Answer: The words that are a part of the standard C language library are called **reserved words**. Those reserved words have special meaning and it is not possible to use them for any activity other than its intended functionality.

Example: void, return int.

Q #6) What is the explanation for the dangling pointer in C?

Answer: When there is a pointer pointing to a memory address of any variable, but after some time the variable was deleted from the memory location while keeping the pointer pointing to that location is known as a dangling pointer in C.

Q #7) Describe static function with its usage?

Answer: A function, which has a function definition prefixed with a static keyword is defined as a static function. The static function should be called within the same source code.

Q #8) What is the difference between abs() and fabs() functions?

Answer: Both functions are to retrieve absolute value. abs() is for integer values and fabs() is for floating type numbers. Prototype for abs() is under the library file < stdlib.h > and fabs() is under < math.h >.

Q #9) Describe Wild Pointers in C?

Answer: Uninitialized pointers in the C code are known as **Wild Pointers**. They point to some arbitrary memory location and can cause bad program behavior or program crash.

Q #10) What is the difference between ++a and a++?

Answer: '++a" is called prefixed increment and the increment will happen first on a variable. 'a++' is called postfix increment and the increment happens after the value of a variable used for the operations.

Q #11) Describe the difference between = and == symbols in C programming?

Answer: '==' is the comparison operator which is used to compare the value or expression on the left-hand side with the value or expression on the right-hand side.

'=' is the assignment operator which is used to assign the value of the right-hand side to the variable on the left-hand side.

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.

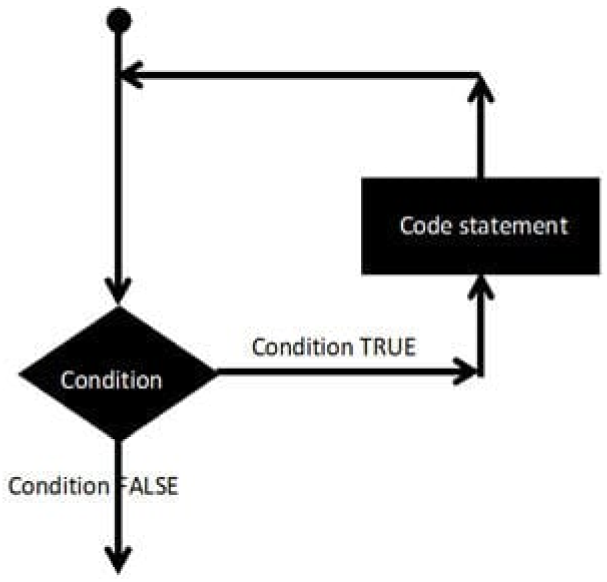
Q #15) There is a practice in coding to keep some code blocks in comment symbols than delete it when debugging. How this affects when debugging?

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?

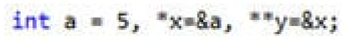
Answer: The function definition in C contains four main sections.

```
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 #19) What is a pointer on a pointer in C programming language?

Answer: A pointer variable that contains the address of another pointer variable is called pointer on a pointer. This concept de-refers twice to point to the data held by a pointer variable.



```
int a = 5, *x=&a, **y=&x;
```

In this example y returns the value of the variable a.**

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.

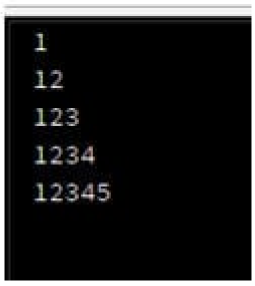
Q #23) What is the method to save data in a stack data structure type?

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.

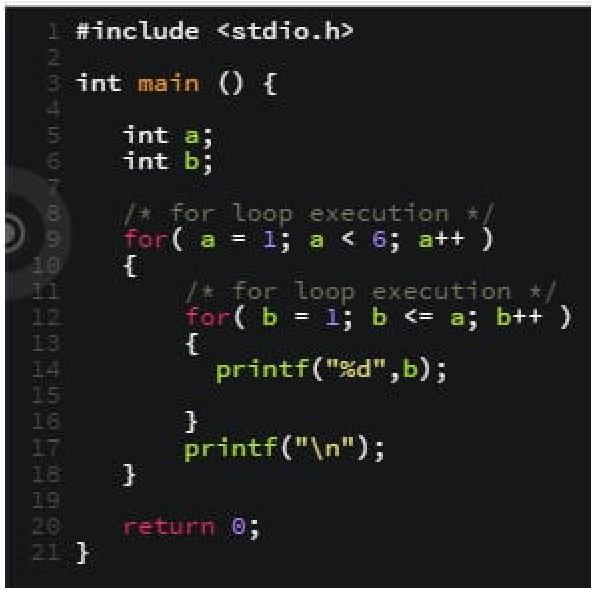
Q #25) What is the correct code to have the following output in C using nested for loop?



```
1
12
123
1234
12345
```

Answer:

```
#include <stdio.h>

int main () {

   int a;

   int b;

   /* for loop execution */

   for``( a = 1; a < 6; a++ )

   {

      /* for loop execution */

      for ( b = 1; b <= a; b++ )

      {

         printf``(``"%d"``,b);

      }

      printf``(``"\n"``);

   }

   return 0;

}
```



```
#include <stdio.h>

int main () {

   int a;
   int b;

   /* for loop execution */
   for( a = 1; a < 6; a++ )
   {
       /* for loop execution */
       for( b = 1; b <= a; b++ )
       {
         printf("%d",b);

       }
       printf("\n");
   }

   return 0;
}
```

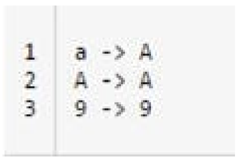
Q #26) Explain the use of function toupper() with an example code?

Answer: Toupper() function is used to convert the value to uppercase when it used with characters.

Code:

```
#include <stdio.h>

#include <ctype.h>

int main()

{

   char c;

   c = 'a'``;

   printf``(``"%c -> %c"``, c, toupper``(c));

   c = 'A'``;

   printf``(``"\n%c -> %c"``, c, toupper``(c));

   c = '9'``;

   printf``(``"\n%c -> %c"``, c, toupper``(c));

   return 0;

}
```

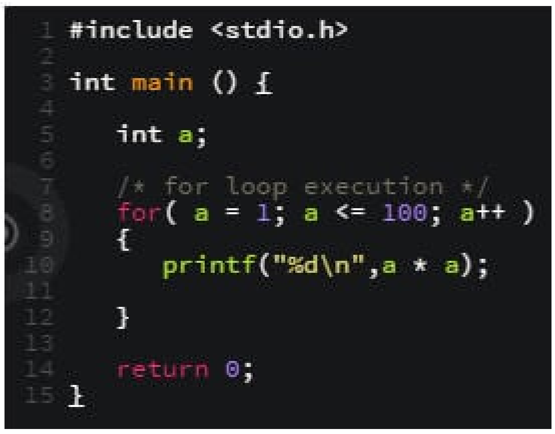
Result:



```
a -> A
A -> A
9 -> 9
```

Q #27) What is the code in a while loop that returns the output of the given code?

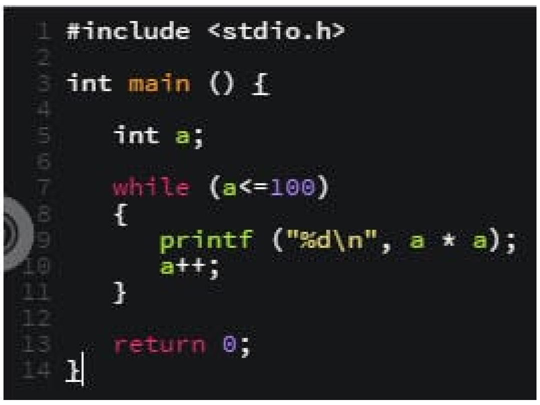
```
#include <stdio.h>

int main () {

    int a;

    /* for loop execution */

    for``( a = 1; a <= 100; a++ )

    {

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

    }

    return 0;

}
```



```
#include <stdio.h>

int main () {

   int a;

   /* for loop execution */
   for( a = 1; a <= 100; a++ )
   {
      printf("%d\n",a * a);

   }

   return 0;
}
```

Answer:

```
#include <stdio.h>

int main () {

   int a;

   while (a<=100)

   {

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

      a++;

   }

   return 0;
}
```



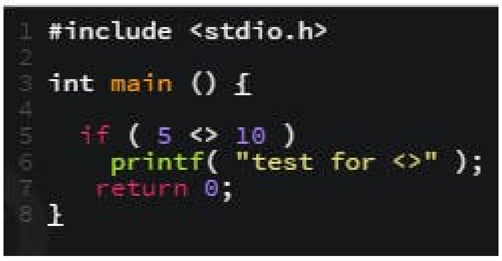
```
#include <stdio.h>

int main () {

   int a;

   while (a<=100)
   {
      printf ("%d\n", a * a);
      a++;
   }

   return 0;
}
```

Q #28) Select the incorrect operator form in the following list(== , <> , >= , <=) and what is the reason for the answer?

Answer: Incorrect operator is ‘<>’. This format is correct when writing conditional statements, but it is not the correct operation to indicate not equal in C programming. It gives a compilation error as follows.

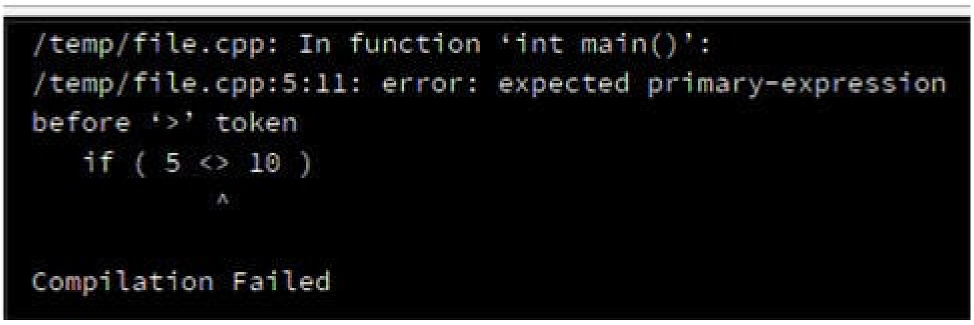
Code:

```
#include <stdio.h>

int main () {

   if ( 5 <> 10 )

     printf``( "test for <>" );

    return 0;

}
```



```
#include <stdio.h>

int main () {

  if ( 5 <> 10 )
    printf( "test for <>" );
   return 0;
}
```

Error:



```
/temp/file.cpp: In function ‘int main()’:
/temp/file.cpp:5:11: error: expected primary-expression
before ‘>’ token
   if ( 5 <> 10 )
          ^

Compilation Failed
```

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:

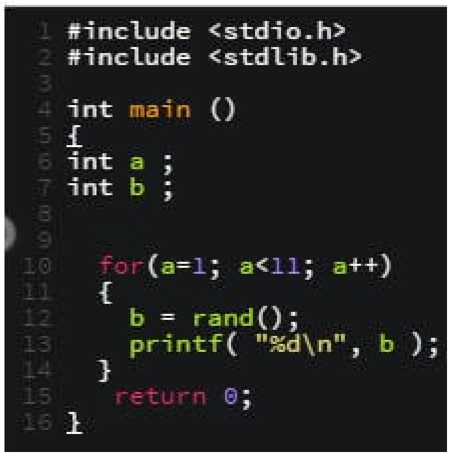
- Short
- Long
- Signed
- Unsigned
- long long

Q #32) What is the process to generate random numbers in C programming language?

Answer: The command rand() is available to use for this purpose. The function returns an integer number beginning from zero(0). The following sample code demonstrates the use of rand().

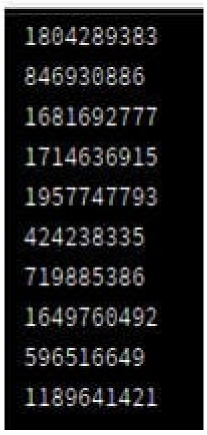
Code:

```
#include <stdio.h>

#include <stdlib.h>

int main ()

{

int a;

int b;

  for``(a=1; a<11; a++)

  {

     b = rand``();

     printf``( "%d\n"``, b );

   }

   return 0;
}
```



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

int main ()
{
int a ;
int b ;


  for(a=1; a<11; a++)
  {
    b = rand();
    printf( "%d\n", b );
  }
   return 0;
}
```

Output:



```
1804289383
846930886
1681692777
1714636915
1957747793
424238335
719885386
1649760492
596516649
1189641421
```

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. The following sample program demonstrates the use of the newline escape sequence.

Code:

```
/*

* C Program to print string

*/

#include <stdio.h>

#include <string.h>

int main(){

   printf``(``"String 01 "``);

   printf``(``"String 02 "``);

   printf``(``"String 03 \n"``);
```

```
    printf``(``"String 01 \n"``);

    printf``(``"String 02 \n"``);

    return 0;

}
```

Output:

```
String 01 String 02 String 03
String 01
String 02
```

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 #36) Describe dynamic data structure in C programming language?

Answer: Dynamic data structure is more efficient to memory. The memory access occurs as needed by the program.

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.

Q #40) What is the explanation for modular programming?

Answer: The process of dividing the main program into executable subsection is called module programming. This concept promotes reusability.

Conclusion

The questioner is based on the C programming language concepts including memory management with pointers, the knowledge of its syntax and some example programs that use the Basic C program structure. Theatrical and practical knowledge of the candidate is examined with the questions.

Recommended Reading

- Interview Questions and Answers
- Some Tricky Manual Testing Questions & Answers
- C Vs C++: 39 Main Differences Between C and C++ With Examples

---

3

Source: https://www.studocu.com/in/document/kalinga-institute-of-industrial-technology/programming-in-c/c-programming-viva-questions/29118795

C programming Viva Questions

1. Which type of language is C?

Ans- C is a high–level language and general-purpose structured programming language.

Ans- Compile is a software program that transfer program developed in a high-level language

into executable object code.

Ans- The approach or method that is used to solve the problem is called an algorithm.

4. What is a c token and types of c tokens?

Ans- The smallest individual units are known as C tokens. C has six types of tokens Keywords,

Constants, Identifiers, Strings, Operators and Special symbols.

5. How many Keywords (reserve word)are in C?

Ans- There are 32 Keywords in the C language.

6. What is an identifier?

Ans- Identifiers are user-defined names given to variables, functions and arrays.

7. What are the Back Slash character constants or Escape sequence characters available in C?

Ans- Back Slash character constant are \t, \n, \0.

Ans- Variables are user-defined names given to memory locations and are used to store values.

A variable may have different values at different times during program execution.

9. What are the Data Types present in C?

Ans- Primary or Fundamental Data types (int, float, char), Derived Datatypes(arrays, pointers) and User-Defined data types(structures, unions, enum)

10. How to declare a variable?

Ans- The syntax for declaring variable isdata type variable_name-1,

variable_name-2,….variable_name-n;

11.What is meant by initialization and how do we initialize a variable?

Ans- While declaring a variable assigning value is known as initialization. Variable can be

initialized by using assignment operator (=).

12.How many types of operators or there in C?

Ans- C consist Arithmetic Operators (+, -, *, /,%), Relational Operators (<, <=, >, >=, !=), Logical

Operators (&&, ||, !), Assignment Operators (=, +=, -=, *=, /=), Increment and Decrement

Operators (++, –), Conditional Operator(?:), Bitwise Operators(<<, >>, ~, &, |, ^) and Special

Operators (. , ->, &, *, size of)

13. What is a Unary operator and what are the unary operators present in C?

Ans- An operator which takes only one operand is called a unary operator. C unary operators

are Unary plus (+), Unary minus (-), Increment and Decrement operators (++,–), Address of

operator (&), Value at operator (*), sizeof operator, ones complement operator (~)

14. What is the use of the modulus (%) operator?

Ans- The modulus operator produces the remainder of an integer division. It cannot be used on

15. What is the use of printf and scanf functions in C?

Ans-Values of variables and results of expressions can be displayed on the screen using printf

functions. Values to variables can be accepted through the keyboard using scanf function

16. Forms of IF statements?

Ans- Simple IF statement, IF-ELSE statement, NESTED IF-ELSE statement and ELSE IF ladder

17. Forms of IF statements?

Ans- Simple IF statement, IF-ELSE statement, NESTED IF-ELSE statement and ELSE IF ladder

18. What is a goto statement?

Ans- GOTO is an unconditional branching statement which transfers control to the specified

Ans- Loop is a sequence of statements that runs repeatedly.

20. What are loop control statements in C?

Ans- While do-while and for.

21. What are sections present in for loop?

Ans- Initialization section, conditional section and increment/decrement section.

22. What is the use of break statements?

Ans- The break statement is used to exit from the loop.

Ans-The array is a collective name given to similar elements.

24. How can we initialize an array?

Ans -The initializer for an array is a comma-separated list of constant expressions enclosed in

({ }). The initializer is preceded by an equal sign (=). You do not need to initialize all elements

25. What is the difference between normal variable and array variable?

Ans- A variable can store only one value at a time whereas an array variable can store several

26. What are the types of Arrays?

Ans- A one-Dimensional array, Two-Dimensional array and Multi-Dimensional array

27. What is a character array?

Ans- An array which can store several characters is called character array

Ans- The function is a self-contained block of the statement which is used to perform a certain

29. What are the types of functions?

Ans- C functions are divided into two categories user-defined function and built-in functions

30. Which are called built-in functions?

Ans- Printf, scanf, clrscr, gotoxy, string handling functions and file handling functions

31. What is a recursive function?

Ans- A function calling itself is called function recursion

32. How to pass an array to a function?

Ans- Arrays are passed to a function by sending its address

33. What is a pointer variable?

Ans- A pointer variable is a variable that can store the address of another variable

34. How can we store the address of a variable is a pointer?

Ans- By using the address of the operator we can store the address of a variable in a pointer

35. How many bytes does a pointer variable occupies in memory?

Ans- A pointer variable irrespective of its type it occupies two bytes in memory

36. What are storage classes available in C?

Ans- Auto, static, extern and register

The structure is user-defined data typed. The structure is a collective name given to

38. How to excess structure members?

Ans- Structure members can be accessed using the dot operator

39. What are the differences between structures and arrays?

Ans- Structures store dissimilar values whereas arrays stores similar values. One structure

variable can be assigned to another structure variable whereas one array variable cannot be

assigned to another array variable

40. What is the size of a structure?

Ans- Sum of all the members size is becomes structure size

Ans- Union is a user-defined data type that can store a value of different data types

42. What are the types of files we can create using C?

Ans-We can create text and binary files using C

43. What are the file-handling functions present in C?

Ans- fopen, fclose, fgetc, fputc, fgets, fputs, fprintf, fscanf, fread, fwrite, fseek

44. What are the file opening modes present in C?

Ans- r, w, a, r+, w+, a+, rb, wb, rb+, wb+

45. How to access structure members by their pointer?

Ans- We can use structure members using arrow operator with its pointer

46. What is the difference between normal variable and array variable?

Ans- A variable can store only one value at a time whereas an array variable can store several

47. How to declare a variable?

Ans- The syntax for declaring the variable is

Datatype variable_name-1,variable_name-2…variable_name-n

Get instant help powered by AI

source: https://lastmomenttuitions.com/c-programming-viva-questions-2/

1. Which type of language is C?
   Ans- C is a high–level language and general-purpose structured programming language.
2. What is a compiler?
   Ans- Compile is a software program that transfer program developed in a high-level language into executable object code.
3. What is an algorithm?
   Ans- The approach or method that is used to solve the problem is called an algorithm.
4. What is a c token and types of c tokens?
   Ans- The smallest individual units are known as C tokens. C has six types of tokens Keywords, Constants, Identifiers, Strings, Operators and Special symbols.
5. How many Keywords (reserve word)are in C?
   Ans- There are 32 Keywords in the C language.
6. What is an identifier?
   Ans- Identifiers are user-defined names given to variables, functions and arrays.
7. What are the Back Slash character constants or Escape sequence characters available in C?
   Ans- Back Slash character constant are \t, \n, \0.
8. What is a variable?
   Ans- Variables are user-defined names given to memory locations and are used to store values. A variable may have different values at different times during program execution.
9. What are the Data Types present in C?
   Ans- Primary or Fundamental Data types (int, float, char), Derived Datatypes(arrays, pointers) and User-Defined data types(structures, unions, enum)
10. How to declare a variable?
    Ans- The syntax for declaring variable isdata type variable_name-1, variable_name-2,….variable_name-n;
11. What is meant by initialization and how do we initialize a variable?
    Ans- While declaring a variable assigning value is known as initialization. Variable can be initialized by using assignment operator (=).
12. How many types of operators or there in C?
    Ans- C consist Arithmetic Operators (+, -, , *,/,%), Relational Operators (<, <=, >, >=, !=), Logical Operators (&&, ||, !), Assignment Operators (=, +=, -=,* =, /=), Increment and Decrement Operators (++, –), Conditional Operator(?:), Bitwise Operators(<<, >>, ~, &, |, ^) and Special Operators (. , ->, &, *, size of)
13. What is a Unary operator and what are the unary operators present in C?
    Ans- An operator which takes only one operand is called a unary operator. C unary operators are Unary plus (+), Unary minus (-), Increment and Decrement operators (++,–), Address of operator (&), Value at operator (*), sizeof operator, ones complement operator (~)
14. What is the use of the modulus (%) operator?
    Ans- The modulus operator produces the remainder of an integer division. It cannot be used on floating-point data
15. What is the use of printf and scanf functions in C?
    Ans-Values of variables and results of expressions can be displayed on the screen using printf functions. Values to variables can be accepted through the keyboard using scanf function
16. Forms of IF statements?
    Ans- Simple IF statement, IF-ELSE statement, NESTED IF-ELSE statement and ELSE IF ladder
17. Forms of IF statements?
    Ans- Simple IF statement, IF-ELSE statement, NESTED IF-ELSE statement and ELSE IF ladder
18. What is a goto statement?
    Ans- GOTO is an unconditional branching statement which transfers control to the specified label.
19. What is a loop?
    Ans- Loop is a sequence of statements that runs repeatedly.

20. What are loop control statements in C?
    Ans- While do-while and for.
21. What are sections present in for loop?
    Ans- Initialization section, conditional section and increment/decrement section.
22. What is the use of break statements?
    Ans- The break statement is used to exit from the loop.
23. What is an array?
    Ans-The array is a collective name given to similar elements.
24. How can we initialize an array?
    Ans- The initializer for an array is a comma-separated list of constant expressions enclosed in braces ({ }). The initializer is preceded by an equal sign (=). You do not need to initialize all elements in an array.
25. What is the difference between normal variable and array variable?
    Ans- A variable can store only one value at a time whereas an array variable can store several value at a time.
26. What are the types of Arrays?
    Ans- A one-Dimensional array, Two-Dimensional array and Multi-Dimensional array
27. What is a character array?
    Ans- An array which can store several characters is called character array
28. What is a function?
    Ans- The function is a self-contained block of the statement which is used to perform a certain task
29. What are the types of functions?
    Ans- C functions are divided into two categories user-defined function and built-in functions
30. Which are called built-in functions?
    Ans- Printf, scanf, clrscr, gotoxy, string handling functions and file handling functions
31. What is a recursive function?
    Ans- A function calling itself is called function recursion
32. How to pass an array to a function?
    Ans- Arrays are passed to a function by sending its address
33. What is a pointer variable?
    Ans- A pointer variable is a variable that can store the address of another variable
34. How can we store the address of a variable is a pointer?
    Ans- By using the address of the operator we can store the address of a variable in a pointer
35. How many bytes does a pointer variable occupies in memory?
    Ans- A pointer variable irrespective of its type it occupies two bytes in memory
36. What are storage classes available in C?
    Ans- Auto, static, extern and register
37. What is a structure?
    Ans- The structure is user-defined data typed. The structure is a collective name given to dissimilar elements
38. How to excess structure members?
    Ans- Structure members can be accessed using the dot operator
39. What are the differences between structures and arrays?
    Ans- Structures store dissimilar values whereas arrays stores similar values. One structure variable can be assigned to another structure variable whereas one array variable cannot be assigned to another array variable
40. What is the size of a structure?
    Ans- Sum of all the members size is becomes structure size
41. What is a union?
    Ans- Union is a user-defined data type that can store a value of different data types
42. What are the types of files we can create using C?
    Ans-We can create text and binary files using C
43. What are the file-handling functions present in C?
    Ans- fopen, fclose, fgetc, fputc, fgets, fputs, fprintf, fscanf, fread, fwrite, fseek

44. What are the file opening modes present in C?
    Ans- r, w, a, r+, w+, a+, rb, wb, rb+, wb+
45. How to access structure members by their pointer?
    Ans- We can use structure members using arrow operator with its pointer
46. What is the difference between normal variable and array variable?
    Ans- A variable can store only one value at a time whereas an array variable can store several value at a time.
47. How to declare a variable?
    Ans- The syntax for declaring the variable is Datatype variable_name-1,variable_name-2…variable_name-n

---

5

C Language or C Programming Viva Questions

source: https://www.efaculty.in/c-language-or-c-programming-viva-questions/
Afaq Ahmad Khan — Read time: 14 minutes

C Language or C Programming Viva Questions

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:

```
#include <stdio.h>
void change(int,int);
int main()
{
   int a=10,b=20;
   change(a,b); //calling a function by passing the values of variables.
   printf(“Value of a is: %d”,a);
   printf(“\n”);
   printf(“Value of b is: %d”,b);
   return 0;
}
void change(int x,int y)
{
   x=13;
   y=17;
}
```

Output:

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

Example of call by reference:

```
#include <stdio.h>
void change(int,int);
int main()
{
    int a=10,b=20;
    change(&a,&b); // calling a function by passing references of variables.
    printf("Value of a is: %d",a);
    printf("\n");
    printf("Value of b is: %d",b);
    return 0;
}
void change(int x,int y)
{
    *x=13;
    *y=17;
}
```

Output:

```
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

```
#include <stdio.h>
int calculate_fact(int);
int main()
{
 int n=5,f;
 f=calculate_fact(n); // calling a function
 printf("factorial of a number is %d",f);
return 0;
}
int calculate_fact(int a)
{
if(a==1)
{
    return 1;
}
else
return a*calculate_fact(a-1); //calling a function recursively.
 }
```

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:

```
data_type array_name[size];
```

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

Syntax:

```
data_type array_name[size];
```

Example of an array:

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

Output:

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:

```
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

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

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.

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

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:

```
#include <stdio.h>
   void main()
   {
       int *ptr = malloc(constant value); //allocating a memory space.
       free(ptr); //ptr becomes a dangling pointer.
       ptr=NULL; //Now, ptr is no longer a dangling pointer.
   }
```

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:

```
#include <stdio.h>
 int main()
{
  int a=10;
  int ptr,**pptr; // ptr is a pointer and **pptr is a double pointer.
  ptr=&a;
  pptr=&ptr;
  printf("value of a is:%d",a);
  printf("\n");
  printf("value of ptr is : %d",ptr);
  printf("\n");
  printf("value of pptr is : %d",pptr);
  return 0;
}
```

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()

- 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()

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

Syntax

```
ptr = realloc(ptr, newsize); // updating the memory size using realloc() function.
```

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

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

Syntax

```
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 the 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

```
struct structure_name
{
  Member_variable1;
 Member_variable2
.
.
}[structure variables];
```

Let's see a simple example.

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

```
struct student
{
   char name[10];       // structure members declaration.
   int age;
}s1;      //structure variable
int main()
{
   printf(“Enter the name”);
   scanf(“%s”,s1.name);
   printf(“\n”);
   printf(“Enter the age”);
   scanf(“%d”,&s1.age);
   printf(“\n”);
   printf(“Name and age of a student: %s,%d”,s1.name,s1.age);
   return 0;
}
```

Output:

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

25- What is a 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);
  printf("\n");
  printf("value of b is %f",d.b);
  printf("\n");
  printf("value of ch is %c",d.ch);
  return 0;
}
```

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

```
int sprintf ( char  str, const char  format, … );
```

Let's see a simple example

```
#include <stdio.h>
int main()
{
 char a[20];
 int n=sprintf(a,"javaToint");
 printf("value of n is %d",n);
 return 0;}
```

Output:

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:

```
#include <stdio.h>
#define start main
void start() {
  printf(“Hello”);
}
```

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:

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

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

```
#include <stdio.h>
#include <conio.h>
int main()
{
 char ch;
 printf(“Enter a character “);
 ch=getch(); // taking an user input without printing the value.
 printf(“\nvalue of ch is %c”,ch);
 printf("\nEnter a character again ");
 ch=getche(); // taking an user input and then displaying it on the screen.
printf("\nvalue of ch is %c",ch);
 return 0;
}
```

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.

33- What is the newline escape sequence?

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

34- Who is the main contributor in designing the C language after Dennis Ritchie?

Brain Kernighan.

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

```
(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:

```
for(;;){
//code to be executed
}
```

Infinite While Loop:

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

Infinite Do-While Loop:

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

41- Write a program to print “hello world” without using a semicolon?

```
#include<stdio.h>
void main(){
 if(printf(“hello world”)){} // It prints the ?hello world? on the screen.
}
```

42- Write a program to swap two numbers without using the third variable?

```
#include<stdio.h>
#include<conio.h>
main()
{
int a=10, b=20;    //declaration of variables.
clrscr();        //It clears the screen.
printf(“Before swap a=%d b=%d”,a,b);

a=a+b;//a=30 (10+20)
 b=a-b;//b=10 (30-20)
 a=a-b;//a=20 (30-10)

 printf(“\nAfter swap a=%d b=%d”,a,b);
 getch();
 }
```

43- Write a program to print Fibonacci series without using recursion?

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int n1=0,n2=1,n3,i,number;
 clrscr();
printf("Enter the number of elements:");
scanf("%d",&number);
printf("\n%d %d",n1,n2);//printing 0 and 1

 for(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already printed
 {
  n3=n1+n2;
  printf(" %d",n3);
  n1=n2;
  n2=n3;
 }
getch();
}
```

44- Write a program to print Fibonacci series using recursion?

```
#include<stdio.h>
#include<conio.h>
void printFibonacci(int n) // function to calculate the fibonacci series of a given number.
{
static int n1=0,n2=1,n3;    // declaration of static variables.
    if(n>0){
         n3 = n1 + n2;
         n1 = n2;
        n2 = n3;
          printf("%d ",n3);
          printFibonacci(n-1);    //calling the function recursively.
    }
}
void main(){
    int n;
    clrscr();
    printf("Enter the number of elements: ");
    scanf("%d",&n);
    printf("Fibonacci Series: ");
    printf("%d %d ",0,1);
    printFibonacci(n-2);//n-2 because 2 numbers are already printed
    getch();
}
```

45- Write a program to check prime number in C Programming?

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,m=0,flag=0;    //declaration of variables.
clrscr();    //It clears the screen.
printf("Enter the number to check prime:");
scanf("%d",&n);
m=n/2;
for(i=2;i<=m;i++)
{
if(n%i==0)
{
printf("Number is not prime");
flag=1;
break;    //break keyword used to terminate from the loop.
```

```c
}
}
if(flag==0)
printf(“Number is prime”);
getch();    //It reads a character from the keyword.
}
```

46- Write a program to check palindrome number in C Programming?

```c
#include<stdio.h>
#include<conio.h>
main()
{
int n,r,sum=0,temp;
clrscr();
printf(“enter the number=”);
scanf(“%d”,&n);
temp=n;
while(n>0)
{
r=n%10;
sum=(sum*10)+r;
n=n/10;
}
if(temp==sum)
printf(“palindrome number “);
else
printf(“not palindrome”);
getch();
}
```

47- Write a program to print factorial of given number without using recursion?

```c
#include<stdio.h>
#include<conio.h>
void main(){
  int i,fact=1,number;
  clrscr();
  printf(“Enter a number: “);
  scanf(“%d”,&number);

  for(i=1;i<=number;i++){
      fact=fact*i;
  }
  printf(“Factorial of %d is: %d”,number,fact);
  getch();
}
```

48- Write a program to print factorial of given number using recursion?

```c
#include<stdio.h>
#include<conio.h>
 long factorial(int n)    // function to calculate the factorial of a given number.
{
  if (n == 0)
```

```
    return 1;
else
return(n * factorial(n-1));    //calling the function recursively.
}
 void main()
{
  int number;    //declaration of variables.
  long fact;
 clrscr();
  printf("Enter a number: ");
scanf("%d", &number);
 fact = factorial(number);    //calling a function.
printf("Factorial of %d is %ld\n", number, fact);
 getch();   //It reads a character from the keyword.
}
```

49- Write a program to check Armstrong number in C?

```
#include<stdio.h>
#include<conio.h>
main()
{
int n,r,sum=0,temp;    //declaration of variables.
clrscr(); //It clears the screen.
printf("enter the number=");
scanf("%d",&n);
temp=n;
while(n>0)
{
r=n%10;
sum=sum+(r*r*r);
n=n/10;
}
if(temp==sum)
printf("armstrong  number ");
else
printf("not armstrong number");
getch();  //It reads a character from the keyword.
}
```

50- Write a program to reverse a given number in C?

```
#include<stdio.h>
#include<conio.h>
main()
{
int n, reverse=0, rem;    //declaration of variables.
clrscr(); // It clears the screen.
printf("Enter a number: ");
scanf("%d", &n);
while(n!=0)
{
     rem=n%10;
     reverse=reverse*10+rem;
     n/=10;
}
printf("Reversed Number: %d",reverse);
getch();  // It reads a character from the keyword.
}
```

6

source: https://engineeringinterviewquestions.com/c-language-viva-questions-with-answers-cse/

1. Who developed C language?
   C language was developed by Dennis Ritchie in 1970 at Bell Laboratories.
2. Which type of language is C?
   C is a high – level language and general purpose structured programming language.
3. What is a compiler?
   Compile is a software program that transfer program developed in high level language intoexecutable object code
4. What is IDE?
   The process of editing, compiling, running and debugging is managed by a single integratedapplication known as Integrated Development Environment (IDE)
5. What is a program?
   A computer program is a collection of the instructions necessary to solve a specific problem.
6. What is an algorithm?
   The approach or method that is used to solve the problem is known as algorithm.
7. What is structure of C program?
   A C program contains Documentation section, Link section, Definition section, Globaldeclaration section, Main function and other user defined functions
8. What is a C token and types of C tokens?
   The smallest individual units are known as C tokens. C has six types of tokens Keywords,Constants, Identifiers, Strings, Operators and Special symbols.
9. What is a Keyword?
   Keywords are building blocks for program statements and have fixed meanings and thesemeanings cannot be changed.
10. How many Keywords (reserve words) are in C?
    There are 32 Keywords in C language.
11. What is an Identifier?
    Identifiers are user-defined names given to variables, functions and arrays.
12. What is a Constant and types of constants in C?
    Constants are fixed values that do not change during the program execution. Types of constants are Numeric Constants (Integer and Real) and Character Constants (SingleCharacter, String Constants).
13. What are the Back Slash character constants or Escape sequence charactersavailable in C?
    Back Slash character constant are \t, \n, \0
14. What is a variable?
    Variables are user-defined names given to memory locations and are used to store values. Avariable may have different values at different times during program execution
15. What are the Data Types present in C?
    Primary or Fundamental Data types (int, float, char), Derived Data types(arrays, pointers)and User-Defined data types(structures, unions, enum)
16. How to declare a variable?
    The syntax for declaring variable isdata type variable_name-1, variable_name-2,….variable_name-n;

17. What is meant by initialization and how we initialize a variable?
While declaring a variable assigning value is known as initialization. Variable can beinitialized by using assignment operator (=).
18. What are integer variable, floating-point variable and character variable?
A variable which stores integer constants are called integer variable. A variable which storesreal values are called floating-point variable. A variable which stores character constants arecalled character variables.
19. How many types of operator or there in C?
C consist Arithmetic Operators (+, -, , */,%), Relational Operators (<, <=, >, >=, !=), LogicalOperators (&&, ||, !), Assignment Operators (=, +=, -=, *=, /=), Increment and DecrementOperators (++, –), Conditional Operator(?:), Bitwise Operators(<<, >>, ~, &, |, ^) andSpecial Operators (. , ->, &, *, sizeof)
20. What is RAM ?
RAM – Random Access Memory is a temporary storage medium in a computer. RAM is a volatile memory i.e all data stored in RAM will be erased when the computer is switched off.
21. What do mean by network ?
Computer networking refers to connecting computers to share data, application software and hardware divices. Networks allow sharing of information among various computers and permit users to share files
22. List a few unconditional control statement in C.
break statement
continue statement
goto statement
exit() function
23. What is an array ?
An array is a collection of values of the same data type. Values in array are accessed using array name with subscripts in brackets[]. Synatax of array declaration is
data type array_ name[size];
24. What is Multidimensional Arrays
An array with more than one index value is called a multidimensional array. To declare a multidimensional array you can do follow syntax
data type array_ name[] [] []….;
25. Define string ?
An array of characters is known as a string.for example
char st[8]; this statement declares a string array with 80 characters .
26. Mention four important string handling functions in C languages .
There are four important string handling functions in C languages .
strlen();
trcpy();
strcat();
strcmp();
The header file `#include` is used when these functions are called in a C program.
27. Explain about the constants which help in debugging?
A `#if` directive test can be offered with `#else` and `#else` if directives. This allows conditional branching of the program to run sections of the code according to the result. Constants defined with a `#define` directive can be undefined with the `#undef` directive. The `#ifdef` directive has a companion directive `#ifndef`. These commands can be useful when debugging problem code to hide and unhide sections of the program.
28. Define and explain about ! Operator?
The logical operator ! NOT is a unary operator that is used before a single operand. It returns the inverse value of the given operand so if the variable “c” had a value of true then! C would return value of false. The not operator is very much useful in C programs because it can change the value of variables with successful iterations. This ensures that on each pass the value is changed.
29. What is operator precedence?
Operator precedence defines the order in which C evaluates expressions.

e.g. in the expression a=6+b*3, the order of precedence determines whether the addition or the multiplication is completed first. Operators on the same row have equal precedence.

30. Explain about the functions strcat() and strcmp()?

    This function concatenates the source string at the end of the target string. Strcmp() function compares two strings to find out whether they are the same or different. The two strings are compared character by character until there is a mismatch or end of one of the strings is reached, whichever occurs first. If in case two strings are identical, a value of zero is returned. If there is no matches between two strings then a difference of the two non matching values are returned according to ASCII values.

31. Define function

    A function is a module or block of program code which deals with a particular task. Each function has a name or identifier by which is used to refer to it in a program. A function can accept a number of parameters or values which pass information from outside, and consists of a number of statements and declarations, enclosed by curly braces { }, which make up the doing part of the object

32. Differentiate built-in functions and user – defined functions.

    Built – in functions are used to perform standard operations such as finding the square root of a number, absolute value and so on. These are available along with the C compiler and are included in a program using the header files math.h, s tring.h and so on.

    User defined functions are written by the user or programmer to compute a value or perform a task. It contains a statement block which is executed during the runtime whenever it is called by the main program.

33. Distinguish between actual and formal arguments.

    Actual arguments are variables whose values are supplied to the function in any function call. Formal arguments are variables used to receive the values of actual arguments from the calling program.

34. Explain the concept and use of type void.

    A function which does not return a value directly to the calling program is referred as a void function. The void functions are commonly used to perform a task and they can return many values through global variable declaration.

35. What is recursion ?

    A function calling itself again and again to compute a value is referref to as recursive function or recursion. Recursion is useful for branching processes and is effective where terms are generated successively to compute a value.