

Top 100 + C Programming Interview Questions & Answers

1) How do you construct an increment statement or decrement statement in C?

There are actually two ways you can do this. One is to use the increment operator ++ and decrement operator -. For example, the statement “x++” means to increment the value of x by 1. Likewise, the statement “x-” means to decrement the value of x by 1. Another way of writing increment statements is to use the conventional + plus sign or - minus sign. In the case of “x++”, another way to write it is “x = x + 1”.

2) What is the difference between Call by Value and Call by Reference?

When using Call by Value, you are sending the value of a variable as parameter to a function, whereas Call by Reference sends the address of the variable. Also, under Call by Value, the value in the parameter is not affected by whatever operation that takes place, while in the case of Call by Reference, values can be affected by the process within the function.

3) Some coders debug their programs by placing comment symbols on some codes instead of deleting it. How does this aid in debugging?

Placing comment symbols /* */ around a code, also referred to as “commenting out”, is a way of isolating some codes that you think maybe causing errors in the program, without deleting the code. The idea is that if the code is in fact correct, you simply remove the comment symbols and continue on. It also saves you time and effort on having to retype the codes if you have deleted it in the first place.

4) What is the equivalent code of the following statement in WHILE LOOP format?

```
for (a=1; a<=100; a++)  
printf ("%d\n", a * a);  
Answer:
```

```
a=1;  
while (a<=100) {  
printf ("%d\n", a * a);  
a++;  
}
```

5) What is a stack?

A stack is one form of a data structure. Data is stored in stacks using the FILO (First In Last Out) approach. At any particular instance, only the top of the stack is accessible, which means that in order to retrieve data that is stored inside the stack, those on the upper part should be extracted first. Storing data in a stack is also referred to as a PUSH, while data retrieval is referred to as a POP.

6) What is a sequential access file?

When writing programs that will store and retrieve data in a file, it is possible to designate that file into different forms. A sequential access file is such that data are saved in sequential order: one data is placed into the file after another. To access a particular data within the sequential access file, data has to be read one data at a time, until the right one is reached.

7) What is variable initialization and why is it important?

This refers to the process wherein a variable is assigned an initial value before it is used in the program. Without initialization, a variable would have an unknown value, which can lead to unpredictable outputs when used in computations or other operations.

8 What is spaghetti programming?

Spaghetti programming refers to codes that tend to get tangled and overlapped throughout the program. This unstructured approach to coding is usually attributed to lack of experience on the part of the programmer. Spaghetti programming makes a program complex and analyzing the codes difficult, and so must be avoided as much as possible.

9) Differentiate Source Codes from Object Codes ?

Source codes are codes that were written by the programmer. It is made up of the commands and other English-like keywords that are supposed to instruct the computer what to do. However, computers would not be able to understand source codes. Therefore, source codes are compiled using a compiler. The resulting outputs are object codes, which are in a format that can be understood by the computer processor. In C programming, source codes are saved with the file extension .C, while object codes are saved with the file extension .OBJ

10) In C programming, how do you insert quote characters (‘ and “) into the output screen?

This is a common problem for beginners because quotes are normally part of a printf statement. To insert the quote character as part of the output, use the format specifiers \' (for single quote), and \" (for double quote).

11) What is the use of a ‘\0’ character?

It is referred to as a terminating null character, and is used primarily to show the end of a string value.

12) What is the difference between the = symbol and == symbol?

The = symbol is often used in mathematical operations. It is used to assign a value to a given variable. On the other hand, the == symbol, also known as “equal to” or “equivalent to”, is a relational operator that is used to compare two values.

13) What is the modulus operator?

The modulus operator outputs the remainder of a division. It makes use of the percentage (%) symbol. For example: $10 \% 3 = 1$, meaning when you divide 10 by 3, the remainder is 1.

14) What is a nested loop?

A nested loop is a loop that runs within another loop. Put it in another sense, you have an inner loop that is inside an outer loop. In this scenario, the inner loop is performed a number of times as specified by the outer loop. For each turn on the outer loop, the inner loop is first performed.

15) Which of the following operators is incorrect and why? (>=, <=, <>, ==)

<> is incorrect. While this operator is correctly interpreted as “not equal to” in writing conditional statements, it is not the proper operator to be used in C programming. Instead, the operator != must be used to indicate “not equal to” condition.

16) Compare and contrast compilers from interpreters.

Compilers and interpreters often deal with how program codes are executed. Interpreters execute program codes one line at a time, while compilers take the program as a whole and convert it into object code, before executing it. The key difference here is that in the case of interpreters, a program may encounter syntax errors in the middle of execution, and will stop from there. On the other hand, compilers check the syntax of the entire program and will only proceed to execution when no syntax errors are found.

17) How do you declare a variable that will hold string values?

The char keyword can only hold 1 character value at a time. By creating an array of characters, you can store string values in it. Example: “char MyName[50]; ” declares a string variable named MyName that can hold a maximum of 50 characters.

18) Can the curly brackets { } be used to enclose a single line of code?

While curly brackets are mainly used to group several lines of codes, it will still work without error if you used it for a single line. Some programmers prefer this method as a way of organizing codes to make it look clearer, especially in conditional statements.

19) What are header files and what are its uses in C programming?

Header files are also known as library files. They contain two essential things: the definitions and prototypes of functions being used in a program. Simply put, commands that you use in C programming are actually functions that are defined from within each header files. Each header file contains a set of functions. For example: `stdio.h` is a header file that contains definition and prototypes of commands like `printf` and `scanf`.

20) What is syntax error?

Syntax errors are associated with mistakes in the use of a programming language. It maybe a command that was misspelled or a command that must was entered in lowercase mode but was instead entered with an upper case character. A misplaced symbol, or lack of symbol, somewhere within a line of code can also lead to syntax error.

21) What are variables and it what way is it different from constants?

Variables and constants may at first look similar in a sense that both are identifiers made up of one character or more characters (letters, numbers and a few allowable symbols). Both will also hold a particular value. Values held by a variable can be altered throughout the program, and can be used in most operations and computations. Constants are given values at one time only, placed at the beginning of a program. This value is not altered in the program. For example, you can assigned a constant named `PI` and give it a value `3.1415` . You can then use it as `PI` in the program, instead of having to write `3.1415` each time you need it.

22) How do you access the values within an array?

Arrays contain a number of elements, depending on the size you gave it during variable declaration. Each element is assigned a number from 0 to number of elements-1. To assign or retrieve the value of a particular element, refer to the element number. For example: if you have a declaration that says `int scores[5];`, then you have 5 accessible elements, namely: `scores[0]`, `scores[1]`, `scores[2]`, `scores[3]` and `scores[4]`.

23) Can I use “int” data type to store the value 32768? Why?

No. “int” data type is capable of storing values from -32768 to 32767. To store 32768, you can use “long int” instead. You can also use “unsigned int”, assuming you don’t intend to store negative values.

24) Can two or more operators such as `\n` and `\t` be combined in a single line of program code?

Yes, it's perfectly valid to combine operators, especially if the need arises. For example: you can have a code like `printf("Hello\n\n'World'")` to output the text "Hello" on the first line and "World" enclosed in single quotes to appear on the next two lines.

25) Why is it that not all header files are declared in every C program?

The choice of declaring a header file at the top of each C program would depend on what commands/functions you will be using in that program. Since each header file contains different function definitions and prototype, you would be using only those header files that would contain the functions you will need. Declaring all header files in every program would only increase the overall file size and load of the program, and is not considered a good programming style.

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

When declaring functions, you will decide whether that function would be returning a value or not. If that function will not return a value, such as when the purpose of a function is to display some outputs on the screen, then "void" is to be placed at the leftmost part of the function header. When a return value is expected after the function execution, the data type of the return value is placed instead of "void".

27) What are compound statements?

Compound statements are made up of two or more program statements that are executed together. This usually occurs while handling conditions wherein a series of statements are executed when a TRUE or FALSE is evaluated. Compound statements can also be executed within a loop. Curly brackets { } are placed before and after compound statements.

28) What is the significance of an algorithm to C programming?

Before a program can be written, an algorithm has to be created first. An algorithm provides a step by step procedure on how a solution can be derived. It also acts as a blueprint on how a program will start and end, including what process and computations are involved.

29) What is the advantage of an array over individual variables?

When storing multiple related data, it is a good idea to use arrays. This is because arrays are named using only 1 word followed by an element number. For example: to store the 10 test results of 1 student, one can use 10 different variable names (grade1, grade2, grade3... grade10). With arrays, only 1 name is used, the rest are accessible through the index name (grade[0], grade[1], grade[2]... grade[9]).

30) Write a loop statement that will show the following output:

12

123

1234

12345

Answer:

```
for (a=1; a<=5; i++) {  
    for (b=1; b<=a; b++)  
        printf("%d",b);  
    printf("\n");  
}
```

31) What is wrong in this statement? `scanf("%d",whatnumber);`

An ampersand & symbol must be placed before the variable name `whatnumber`. Placing & means whatever integer value is entered by the user is stored at the “address” of the variable name. This is a common mistake for programmers, often leading to logical errors.

32) How do you generate random numbers in C?

Random numbers are generated in C using the `rand()` command. For example: `anyNum = rand()` will generate any integer number beginning from 0, assuming that `anyNum` is a variable of type integer.

33) What could possibly be the problem if a valid function name such as `tolower()` is being reported by the C compiler as undefined?

The most probable reason behind this error is that the header file for that function was not indicated at the top of the program. Header files contain the definition and prototype for functions and commands used in a C program. In the case of “`tolower()`”, the code “`#include <ctype.h>`” must be present at the beginning of the program.

34) What are comments and how do you insert it in a C program?

Comments are a great way to put some remarks or description in a program. It can serve as a reminder on what the program is all about, or a description on why a certain code or function was placed there in the first place. Comments begin with `/*` and ended by `*/` characters. Comments can be a single line, or can even span several lines. It can be placed anywhere in the program.

35) What is debugging?

Debugging is the process of identifying errors within a program. During program compilation, errors that are found will stop the program from executing completely. At this state, the programmer would look into the possible portions where the error occurred. Debugging ensures the removal of errors, and plays an important role in ensuring that the expected program output is met.

36) What does the && operator do in a program code?

The && is also referred to as AND operator. When using this operator, all conditions specified must be TRUE before the next action can be performed. If you have 10 conditions and all but 1 fails to evaluate as TRUE, the entire condition statement is already evaluated as FALSE

37) In C programming, what command or code can be used to determine if a number of odd or even?

There is no single command or function in C that can check if a number is odd or even. However, this can be accomplished by dividing that number by 2, then checking the remainder. If the remainder is 0, then that number is even, otherwise, it is odd. You can write it in code as:

```
if (num % 2 == 0)

printf("EVEN");

else

printf("ODD");
```

38) What does the format %10.2 mean when included in a printf statement?

This format is used for two things: to set the number of spaces allotted for the output number and to set the number of decimal places. The number before the decimal point is for the allotted space, in this case it would allot 10 spaces for the output number. If the number of space occupied by the output number is less than 10, additional space characters will be inserted before the actual output number. The number after the decimal point sets the number of decimal places, in this case, it's 2 decimal spaces.

39) What are logical errors and how does it differ from syntax errors?

Program that contains logical errors tend to pass the compilation process, but the resulting output may not be the expected one. This happens when a wrong formula was inserted into the code, or a wrong sequence of commands was performed. Syntax errors, on the other hand, deal with incorrect commands that are misspelled or not recognized by the compiler.

40) What are the different types of control structures in programming?

There are 3 main control structures in programming: Sequence, Selection and Repetition. Sequential control follows a top to bottom flow in executing a program, such that step 1 is first

perform, followed by step 2, all the way until the last step is performed. Selection deals with conditional statements, which mean codes are executed depending on the evaluation of conditions as being TRUE or FALSE. This also means that not all codes may be executed, and there are alternative flows within. Repetitions are also known as loop structures, and will repeat one or two program statements set by a counter.

41) What is || operator and how does it function in a program?

The || is also known as the OR operator in C programming. When using || to evaluate logical conditions, any condition that evaluates to TRUE will render the entire condition statement as TRUE.

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 than 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".

49) Why is C language being considered a middle level language?

This is because C language is rich in features that make it behave like a high level language while at the same time can interact with hardware using low level methods. The use of a well structured approach to programming, coupled with English-like words used in functions, makes it act as a high level language. On the other hand, C can directly access memory structures similar to assembly language routines.

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.

52) What are linked list?

A linked list is composed of nodes that are connected with another. In C programming, linked lists are created using pointers. Using linked lists is one efficient way of utilizing memory for storage.

53) What is FIFO?

In C programming, there is a data structure known as queue. In this structure, data is stored and accessed using FIFO format, or First-In-First-Out. A queue represents a line wherein the first data that was stored will be the first one that is accessible as well.

54) What are binary trees?

Binary trees are actually an extension of the concept of linked lists. A binary tree has two pointers, a left one and a right one. Each side can further branch to form additional nodes, which each node having two pointers as well.

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.

56) What is the difference between the expression “++a” and “a++”?

In the first expression, the increment would happen first on variable a, and the resulting value will be the one to be used. This is also known as a prefix increment. In the second expression, the current value of variable a would be the one to be used in an operation, before the value of a itself is incremented. This is also known as postfix increment.

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 expressed 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.

64) What is a newline escape sequence?

A newline escape sequence is represented by the `\n` character. This is used to insert a new line when displaying data in the output screen. More spaces can be added by inserting more `\n` characters. For example, `\n\n` would insert two spaces. A newline escape sequence can be placed before the actual output expression or after.

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.

70) Write a simple code fragment that will check if a number is positive or negative

```
If (num>=0)

printf("number is positive");

else

printf ("number is negative");
```

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.

76) What are multidimensional arrays?

Multidimensional arrays are capable of storing data in a two or more dimensional structure. For example, you can use a 2 dimensional array to store the current position of pieces in a chess game, or position of players in a tic-tac-toe program.

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.

78) What is the difference between functions getch() and getche()?

Both functions will accept a character input value from the user. When using getch(), the key that was pressed will not appear on the screen, and is automatically captured and assigned to a variable. When using getche(), the key that was pressed by the user will appear on the screen, while at the same time being assigned to a variable.

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.

84) What is dynamic data structure?

Dynamic data structure provides a means for storing data more efficiently into memory. Using dynamic memory allocation, your program will access memory spaces as needed. This is in contrast to static data structure, wherein the programmer has to indicate a fix number of memory space to be used in the program.

85) What are the different data types in C?

The basic data types are `int`, `char`, and `float`. `int` is used to declare variables that will be storing integer values. `float` is used to store real numbers. `char` can store individual character values.

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/output 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 removed 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 your 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;
```

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

It has to do with the parsing process and compilation of the code. A semicolon acts as a delimiter, so that the compiler knows where each statement ends, and can proceed to divide the statement into smaller elements for syntax checking.

101) 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();
```

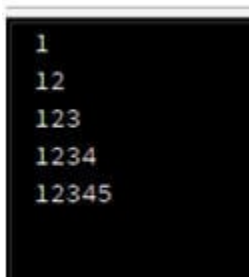
```
local_static();
```

```
return 0;
```

```
}
```

```
0 1
```

102) 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;
}

```



```

1  #include <stdio.h>
2
3  int main () {
4
5      int a;
6      int b;
7
8      /* for loop execution */
9      for( a = 1; a < 6; a++ )
10     {
11         /* for loop execution */
12         for( b = 1; b <= a; b++ )
13         {
14             printf("%d",b);
15         }
16         printf("\n");
17     }
18 }
19
20 return 0;
21 }

```

103) 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:

1	a -> A
2	A -> A
3	9 -> 9

104) 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;
}

```

```

1 #include <stdio.h>
2
3 int main () {
4
5     int a;
6
7     /* for loop execution */
8     for( a = 1; a <= 100; a++ )
9     {
10        printf("%d\n",a * a);
11    }
12
13    return 0;
14 }
15

```

Answer:

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

```

int main () {

    int a;

    while (a<=100)
    {
        printf ("%d\n", a * a);
        a++;
    }
    return 0;
}

```

```

1 #include <stdio.h>
2
3 int main () {
4     int a;
5     while (a<=100)
6     {
7         printf ("%d\n", a * a);
8         a++;
9     }
10
11     return 0;
12 }

```

105) 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;
}

```

```

1 #include <stdio.h>
2
3 int main () {
4     if ( 5 <> 10 )
5     {
6         printf( "test for <>" );
7         return 0;
8     }
9 }

```

Error:

```

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

```

106) 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.

107) 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

108) 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

109) 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;
}
```

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main ()
5 {
6     int a ;
7     int b ;
8
9
10    for(a=1; a<11; a++)
11    {
12        b = rand();
13        printf( "%d\n", b );
14    }
15    return 0;
16 }

```

Output:

```

1804289383
846930886
1681692777
1714636915
1957747793
424238335
719885386
1649760492
596516649
1189641421

```

110) 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:

```

1 String 01 String 02 String 03
2 String 01
3 String 02

```

