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

- **Portability** – Platform independent language.
- **Modularity** – Possibility to break down large programs into small modules.
- **Flexibility** – The possibility to a programmer to control the language.
- **Speed** – C comes with support for system programming and hence it is compiling and executes with high speed when comparing 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?

- **Int –** Represent 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?

Ans) The mistakes when creating a program called syntax errors. Misspelled commands or incorrect case commands, an incorrect number of parameters when called a method /function, data type mismatches can identify as common examples for syntax errors.

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

Ans) ***There are two possible methods to perform this task.***

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

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

2) Use conventional + or – sign.

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?

Ans) The words that are part of the slandered **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?

Ans) When there is a pointer with pointing to a memory address of any variable, but

Q #7) Describe static function with its usage?

Ans) A function, which has a function definition prefixed with a static keyword is defined as a static function. The static function should call within the same source code.

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

Ans) 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?

Ans) Uninitialized pointers in the C code are known as **Wild Pointers**. These are a 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++?

Ans) '++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?

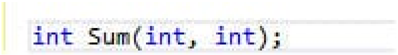
Ans) '==' is the comparison operator which is use to compare the value or expression on the left-hand side with the value or expression on the right-hand side.

'=' is the assignment operator which is use 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?

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 integer data type and it accepts two integer parameters.

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

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

This is called as cyclic nature and 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?

Ans) The file contains 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 the stdio.h.

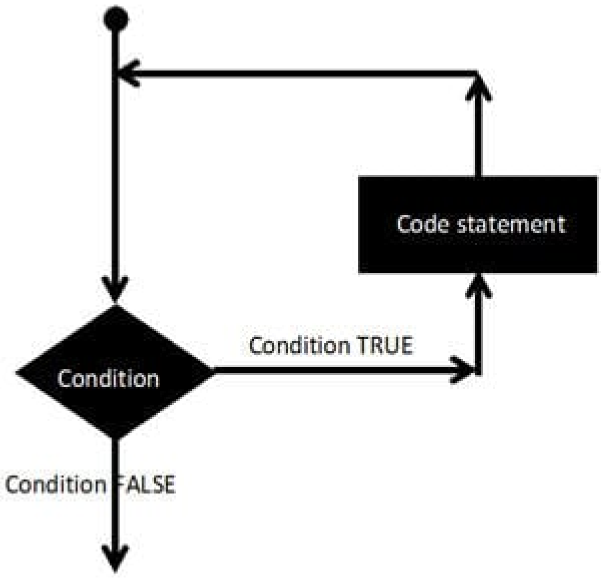
Q #15) There is a practice in coding to keep some code blocks in comment symbols than delete it when debugging. How this affect when debugging?

Ans) This concept called as commenting out and 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 uncomment.

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

Ans) A statement that allows executing statement or group of statements in repeated way is defined as a loop. Following diagram explains

Following diagram explains a general form of a loop.



There are 4 types of a loop statement in C.

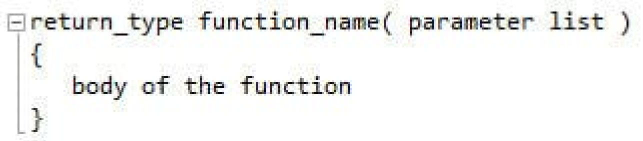
- **For Loop**
- **Do…While Loop**
- **Nested Loop**

Q #17) What is a nested loop?

Ans) A loop running within another loop is referred as a **nested loop**. The first loop is called Outer loop and inside the loop is called Inner loop. Inner loop executes the number of times define an outer loop.

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

Ans) Function definition in C contains four main sections.

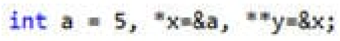


```
return_type function_name( parameter list )
{
   body of the function
}
```

- **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 need to use perform the required action.
- **Function Body** -> Collection of statement that needs to perform the required action.

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

Ans) 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 value of the variable a.

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

Ans) 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 include header file in double quotes (“”) and angular braces (<>)?

Ans) When Header file include within double quotes (“”), compiler search first in the path. But when Header file include within angular braces (<>), the compiler only search in the working directory for the particular header file.

Q #22) What is a sequential access file?

Ans) In general programs store data into files and retrieve existing data from files. With the sequential access file such data saved in a sequential pattern. When retrieving data from such files each data need to read one by one until required information find.

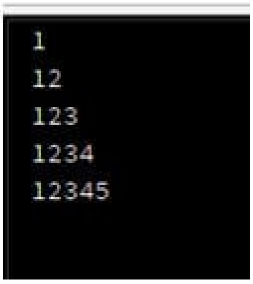
Q #23) What is the method to save data in stack data structure type?

Ans) Data is stored in Stack data structure type using **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 as a POP.

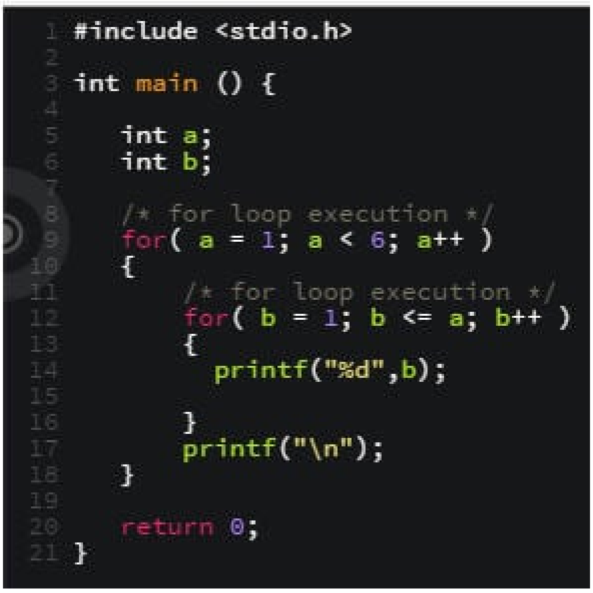
Q #24) What is the significance of C program algorithms?

Ans) The algorithm needs to create first and it contains step by step guidelines on how the solution should create. Also, it contains the steps to consider and the required calculations/operations within the program.

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



```
1
12
123
1234
12345
```

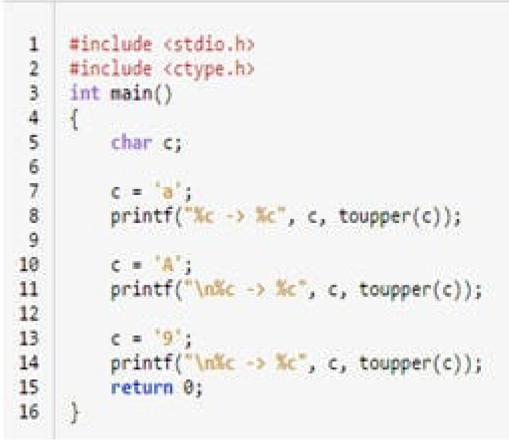


```
#include <stdio.h>

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 and example code?

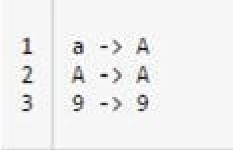
Ans) Toupper() function is use to convert the value to uppercase when it uses with characters.

Code –



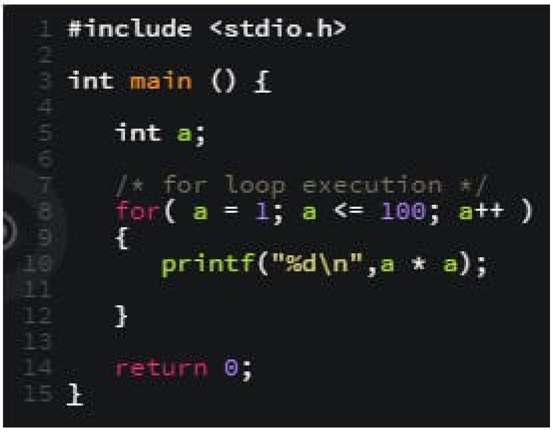
```
#include <stdio.h>
#include <ctype.h>
int main()
{
    char c;

    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 –

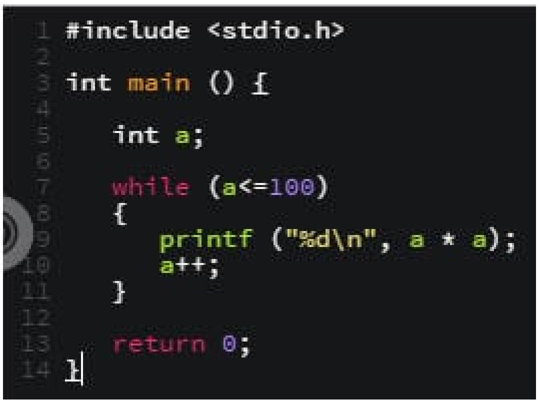


```
a -> A
A -> A
9 -> 9
```

Q #27) What is the code in while loop that returns the output of 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;
}
```



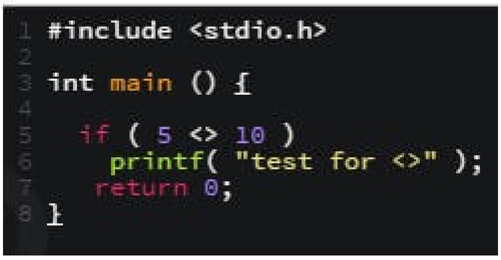
```
#include <stdio.h>

int main () {

   int a;

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

   return 0;
}
```

☰ Menu

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

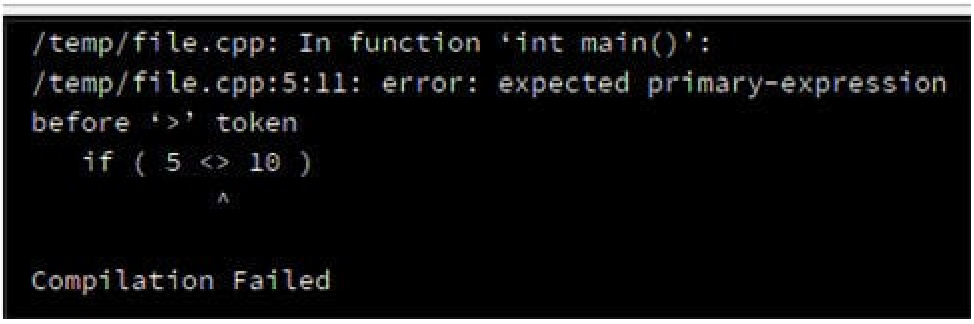
Ans) Incorrect operator is ‘<>’.This is the format correct when writing conditional statements, but it is not a correct operation to indicate not equal in C programming and it gives compilation error as follows.

Code –



```
#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 single line code in C program?

Ans) Yes, it is working 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?

Ans) 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 32-bit processor storage space for 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?

Ans) ***There are 5 modifiers available in C programming language as follows.***

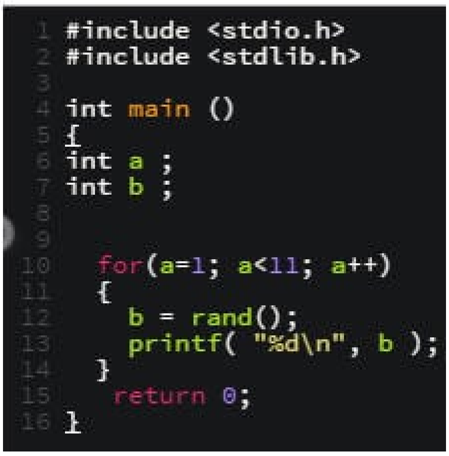
☰ Menu

- Long
- Signed
- Unsigned
- long long

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

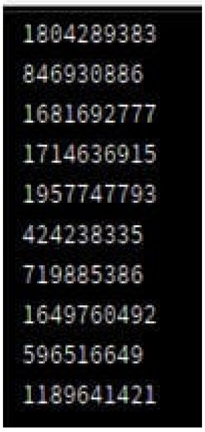
Ans) The command rand() is available to use for this purpose. The function returns any integer number beginning from zero(0). Following sample code demonstrate the use of rand().

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;
}
```

Output –

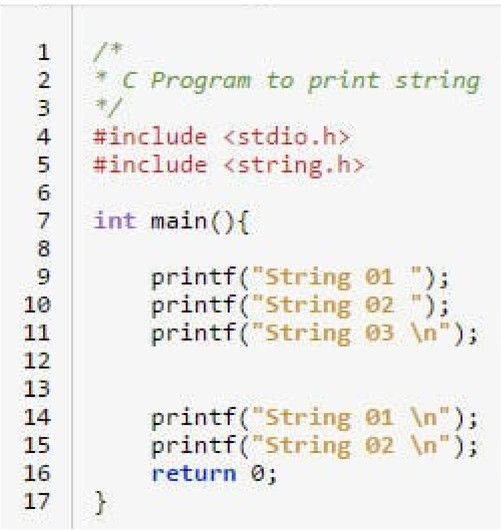


```
1804289383
846930886
1681692777
1714636915
1957747793
424238335
719885386
1649760492
596516649
1189641421
```

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

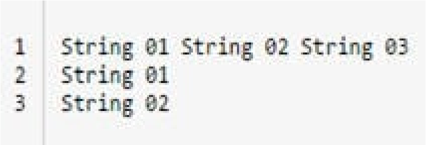
Ans) New line escape sequence is represented by \n. This indicates the point that new line need to start to the compiler and the output creates accordingly. Following sample program demonstrate the use of newline escape sequence.

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 screen



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

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

Ans) Int data type only capable of storing values between – 32768 to 32767.To store 32768 a modifier needs to use with int data type. Long Int can use and also if there is no any negative values unsigned int is also possible to use.

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

Ans) It is possible and easy to create a new header file. Create a file with function prototypes that needs to use inside the program. Include the file in '#include' section from its name.

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

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

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

Ans) 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?

Ans) 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 indirect reference. But when we

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

Ans) ***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?

Ans) The process of dividing the main program into executable subsection is called module programming. This concept promotes the 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 Basic C program structure. Theatrical and practical knowledge of the candidate is examined with the questions.

a Share **8** k Share **15** f +1 **10** d Tweet **33** SHARES

FREE EBook Download!

Testing Help & Tutorials

QA Training
Selenium Tutorials
QTP Tutorials
ISTQB Study Guide
ISTQB Premium Study Guide
QC Tutorials
HP LoadRunner Tutorials
JMeter Tutorials
JIRA Tutorials
VBScript Tutorials
Best Test Management Tools
Appium Studio Tutorials

☰ Menu