C PROGRAMMING

-TAMIL TELEGRAM TECH INTERVIEWERS.

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.

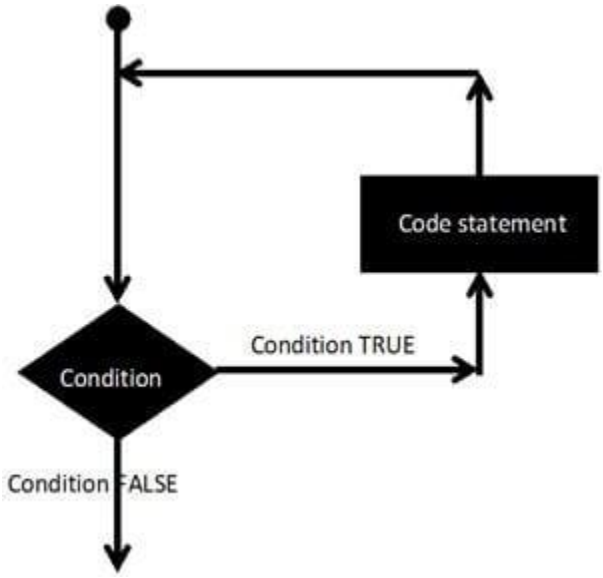
Q #15) There is a practice in coding to keep some code blocks in comment symbols than delete it when debugging. How this affects when debugging?

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.

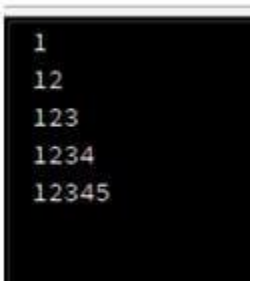
Q #23) What is the method to save data in a stack data structure type?

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?



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;
}
```

```c
#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:

```c
#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?

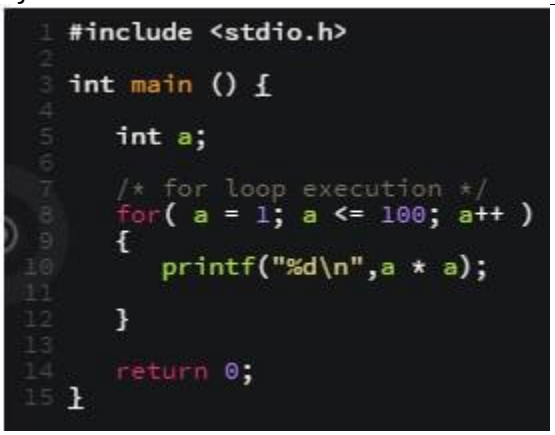
```c
#include <stdio.h>
```

```c
int main () {

      int a;

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

      return 0;
}
```



```c
#include <stdio.h>

int main () {

   int a;

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

   }

   return 0;
}
```

Answer:

```c
#include <stdio.h>

int main () {

     int a;

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

```c
#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:

```c
#include <stdio.h>

int main () {

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

```c
#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;
}
```

```c
#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
/*
* 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.