Q. What is the C Language?

C is a procedural programming language. It was initially developed by Dennis Ritchie in the year 1972. It was mainly developed as a system programming language to write an operating system. The main features of the C language include low-level memory access, a simple set of keywords, and a clean style; these features make C language suitable for system programmings like an operating system or compiler development.
Many later languages have borrowed syntax/features directly or indirectly from the C language. Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on the C language. C++ is nearly a superset of C language (Few programs may compile in C, but not in C++).

Q. What is the character set in C Language and why use it?

As every language contains a set of characters used to construct words, statements, etc., C language also has a set of characters which include **alphabets, digits**, and **special symbols**. C language supports a total of 256 characters.
Every C program contains statements. These statements are constructed using words and these words are constructed using characters from C character set. C language character set contains the following set of characters...

1. Alphabets
2. Digits
3. Special Symbols

Alphabets

C language supports all the alphabets from the English language. Lower and upper case letters together support 52 alphabets.

Lower case letters - **a to z**

UPPER CASE LETTERS - **A to Z**

Digits

C language supports 10 digits which are used to construct numerical values in C language.

Digits - **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

Special Symbols

C language supports a rich set of special symbols that include symbols to perform mathematical operations, to check conditions, white spaces, backspaces, and other special symbols.

Special Symbols - **~ @ # $ % ^ & * ( ) _ - + = { } [ ] ; : ' " / ? . > , < \ | tab newline space NULL bell backspace vertical tab etc.,**

Q. Explain the tokens in c language?

A token is the smallest element of a program that is meaningful to the compiler. Tokens can be classified as follows:

1. Keywords
2. Identifiers
3. Constants
4. Strings
5. Special Symbols
6. Operators

Keywords: Keywords are pre-defined or reserved words in a programming language. Each keyword is meant to perform a specific function in a program. Since keywords are referred names for a compiler, they can't be used as variable names because by doing so, we are trying to assign a new meaning to the keyword which is not allowed. You cannot redefine keywords. However, you can specify the text to be substituted for keywords before compilation by using

C/C++ preprocessor directives. **C** language supports **32** keywords which are given below:

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

Constants: Constants are also like normal variables. But, the only difference is, their values cannot be modified by the program once they are defined. Constants refer to fixed values. They are also called literals. Constants may belong to any of the data type

Types of Constants:

1. Integer constants – Example: 0, 1, 1218, 12482
2. Real or Floating-point constants – Example: 0.0, 1203.03, 30486.184
3. Octal & Hexadecimal constants – Example: octal: $(013)_8$ = $(11)_{10,}$ Hexadecimal: $(013)_{16} = (19)_{10}$
4. Character constants -Example: 'a', 'A', 'z'
5. String constants -Example: "GIRI'S TECH HUB"

Q. what is the data type and why use it?

Data type is used for decide which kind of information we want to use in program means using data type we can specify which type of data want to use in program.

There are three types of data type in c language

1) Built In Data Type: Built in data type means those data type already present in c language called as built in data types.

There are three types of built in data type

a) Integer

b) Float

c) Character.

2) User Defined Data Type: those data type create by the user for its own purpose called as defined data type and it is combination of different type of data.

There are three types of user defined data type

a) Structure

b) Union

c) Enum

3) Derived Data Type: Derived data type means those data type can define with other data type means derived data type can always use with other data type**.**

There are three types of derived data type

a) pointer

b) Function

c) Array

Q. What is the variable and why use it?

Variables are the some identifier which is used for store the information in memory means a variable is block of memory and its size is depend on its data type and which is used for store the data as per its type and variable can modify or change the value in future.

Syntax of variable declaration: datatype variablename;

e.g int a;

As per the above statement here we declare the variable a means we have the block in memory of 2 byte and its name is a and we can store the integer values in it.

If we want to work with a variable or declare the variable we have the some important rules of variable declaration

---

1) Variable name start with alphabet or underscore character

e.g. int a, _b; is valid variable declaration

2) Variable name cannot start with digit but digit may be come later.

3) Cannot use the any special symbol in variable name except underscore character.

4) Upper case and lower case variable consider as different variable.

Q. What is the operator and explain its type

Operators are the some symbols to perform some operation in programming

There are seven types of operator in c language

---

1) Arithmetic Operator

2) Assignment Operator

3) Relational Operator

4) Logical Operator

5) Conditional Operator

6) Increment and Decrement Operator

7) Bitwise Operator

Q. what is the bitwise operator in c language

The bitwise operators are the operators used to perform the operations on the data at the bit-level. When we perform the bitwise operations, then it is also known as bit-level programming. It consists of two digits, either 0 or 1. It is mainly used in numerical computations to make the calculations faster.

Operator	Meaning of operator
&	**Bitwise AND operator**
|	**Bitwise OR operator**
^	**Bitwise exclusive OR operator**
~	**One's complement operator (unary operator)**
<<	**Left shift operator**
>>	**Right shift operator**

Let's look at the truth table of the bitwise operators.

X	Y	X&Y	X|Y	X^Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

Q. what is the constant and why use it as well as explain the types of constant in c language

Constant means once we initialize the value cannot change later the major goal of constant is reuse the values

Basically there five types of constant in c language

1) integer constant: integer constant cannot store the point values in it e.g 100

2) float constant: float constant can store the decimal point values in it e.g 1.5

3) Character constant: this is used for store the character constant in c language

'A','B' etc

4) Octal Constant: octal constant start with 0 in c language and range of octal constant is 0 to 7 e.g. 072 is octal constant

5) Hexa Decimal Constant: Hexa Decimal constant start with 0x and 0X and between 0 to 15 value.

Q. Explain the conditional operator in c language

Conditional operator is used for check the condition and it contains the ternary operator ? And : operator

Syntax : exp1 ? exp2 : exp3;

Exp1 is always condition and exp2 and exp3 are the options if exp1 is true then exp2 get executed and if exp1 is false then exp3 executed.

Q. Explain Integer Data Type in Detail?

Integer Data Type is used for store the non decimal point values

There are four types of integer data type in c language

1) Integer : if we want to use the integer we have the int keyword and memory size of integer is 2 byte or 4 byte is depend on compiler range is

15 31

2 or 2 is also depend on memory size and format specified is %d

2) long integer : long integer is also subtype of integer and it is used for store the decimal values memory size of long integer is 4 or 8 byte is depend on compiler and range is

31 63

2 or 2

3) Short integer: short integer is also subtype of integer and it is used for store the decimal point values and memory size is 2 byte and range

15

2 and format specifier is %i or %d

4) Unsigned integer: unsigned integer is used for store the positive values memory size is 2 byte format specifier is %u and range is

16

2

Q. what is the character data type and why use it?

Character data type is used for store the alpha numeric values means using character data type we can store the alphabetical values as well some time we have the alpha numeric values e.g suppose we have the college PRN number it is combination of character and alphabet TCHJAVA00122.

There are two types of character data type

1) Sign character: if we want to use the sign character we have to use the char in program and memory size of character data type is 1 byte and the range is -128 to 127

2) Unsigned character: if we want to use the unsigned character we have to use the unsigned char and memory size is 1 byte and range is 0 to 255 and format specified is %c

Q. Explain the brief history of C Language

History of C language is interesting to know. Here we are going to discuss a brief history of the c language. **C programming language** was developed in 1972 by Dennis Ritchie at bell laboratories of AT&T (American Telephone & Telegraph), located in the U.S.A. **Dennis Ritchie** is known as the **founder of the c language**. It was developed to overcome the problems of previous languages such as B, BCPL, etc.

Initially, C language was developed to be used in **UNIX operating system**. It inherits many features of previous languages such as B and BCPL. Let's see the programming languages that were developed before C language.

Q. Explain the ASCII In C Language?

The full form of ASCII is the **American Standard Code for information interchange**. It is a character encoding scheme used for electronics communication. Each character or a special character is represented by some ASCII code, and each ASCII code occupies 7 bits in memory.

There is Total 256 Ascii code in c language

Some of the important ASCII code given below

A to Z = 65 to 90

a to z = 97 to 122

0 = 48 to 9 = 57

Q. Explain the Format Specifier in c Language?

The Format Specifier is a string used in the formatted input and output functions. The format string determines the format of the input and output. The format string always starts with a '%' character.

Format specifier	Description
%d or %i	It is used to print the signed integer value where signed integer means that the variable can hold both positive and negative values.
%u	It is used to print the unsigned integer value where the unsigned integer means that the variable can hold only positive value.
%o	It is used to print the octal unsigned integer where octal integer value always starts with a 0 value.
%x	It is used to print the hexadecimal unsigned integer where the hexadecimal integer value always starts with a 0x value. In this, alphabetical characters are printed in small letters such as a, b, c, etc.
%X	It is used to print the hexadecimal unsigned integer, but %X prints the

	alphabetical characters in uppercase such as A, B, C, etc.
%f	It is used for printing the decimal floating-point values. By default, it prints the 6 values after '.'.
%e/%E	It is used for scientific notation. It is also known as Mantissa or Exponent.
%g	It is used to print the decimal floating-point values, and it uses the fixed precision, i.e., the value after the decimal in input would be exactly the same as the value in the output.
%p	It is used to print the address in a hexadecimal form.
%c	It is used to print the unsigned character.
%s	It is used to print the strings.
%ld	It is used to print the long-signed integer value.

Q. Explain the Comments in c language?

Comments in C language are used to provide information about lines of code. It is widely used for documenting code. There are 2 types of comments in the C language.

1. Single Line Comments
2. Multi-Line Comments

3. Single Line Comments
4. Single line comments are represented by double slash \\. Let's see an example of a single line comment in C.

```
#include<stdio.h>
  int main(){
  //printing information
     printf("Hello C");
  return 0;
}
```

Multiline Comments

Multi-Line comments are represented by slash asterisk * ... *\. It can occupy many lines of code, but it can't be nested. Syntax:

```
/*
  code
     to be commented
     */
  #include<stdio.h>
     int main(){
        /*printing information
          Multi-Line Comment*/
        printf("Hello C");
     return 0;
  }
```

Q. what is the compiler?

Difference between Compiler and Interpreter

Compiler:

It is a translator which takes input i.e., High-Level Language, and produces an output of low-level language i.e. machine or assembly language.

- A compiler is more intelligent than an assembler it checks all kinds of limits, ranges, errors, etc.
- But its program run time is more and occupies a larger part of memory. It has slow speed because a compiler goes through the entire program and then translates the entire program into machine codes.

Interpreter:

An interpreter is a program that translates a programming language into a comprehensible language. –

- It translates only one statement of the program at a time.
- Interpreters, more often than not are smaller than compilers.

S.No.	Compiler	Interpreter
1.	Compiler scans the whole program in one go.	Translates program one statement at a time.
2.	As it scans the code in one go, the errors (if any) are shown at the end together.	Considering it scans code one line at a time, errors are shown line by line.
3.	Main advantage of compilers is it's execution time.	Due to interpreters being slow in executing the object code, it is preferred less.
4.	It converts the source code into object code.	It does not convert source code into object code instead it scans it line by line
5	It does not require source code for later execution.	It requires source code for later execution.
Eg.	C, C++, C# etc.	Python, Ruby, Perl, SNOBOL, MATLAB, etc.

Q. what will be the output of given code?

```
void main()
{
    int i=0, j=1, k=2, m;
    m = i++ ||  j++ || k++;
    printf("%d %d %d %d", m, i, j, k);
```

```
}
```

Q. What will be the output of given code?

```
void main()
{
    int c = - -2;
    printf("c=%d", c);
}
```

Q. What will be the output of given code ?

```
void main()
{  int c = - -2;
    printf("c=%d", c);
}
```

Q. What will be the output of given code?

```
#include <stdio.h>
   int main()
   {  int i = 3;
      int l = i / -2;
      int k = i % -2;
      printf("%d %d\n", l, k);
      return 0;
   }
```

What will be the output of given code ?

```
#include <stdio.h>
   int main()
   {
      int i = 5;
      i = i / 3;
      printf("%d\n", i);
      return 0;
   }
```

What will be the output of given code ?

```
#include <stdio.h>

   int main()

   { int i = -5;

      i = i / 3;

      printf("%d", i);

      return 0;

   }
```

What will be the output of given code?

```
#include <stdio.h>

   int main()

   {  int i = -5;

      i = i / 3;

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

```
    return 0;
}
```

What will be the output of given code?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

What will be the output of given code ?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

Q. what will be the output of given code ?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

Q. What will be the output of given code?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

Q. what will be the output of given code?

```
#include <stdio.h>
    int main()
    {
```

```
    int i = -5;
    i = i / 3;
    printf("%d", i);
    return 0;
}
```

Q. what will be the output of given code ?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

What will be the output of given code ?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
```

```
    return 0;
}
```

what will be the output of given code ?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

what will be the output of given code

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

What will be the output of given code ?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

What will be the output of given code?

```
#include <stdio.h>
    int main()
    {
        int i = -5;
        i = i / 3;
        printf("%d", i);
        return 0;
    }
```

What will be the output of given code

```
#include <stdio.h>
    int main()
    {
```

```
    int i = -5;
    i = i / 3;
    printf("%d", i);
    return 0;
}
```

what will be the output of given program ?

```
#include <stdio.h>
    void main()
    {
        int a = 5, b = -7, c = 0, d;
        d = ++a && ++b || ++c;
        printf("\n%d%d%d%d", a, b, c, d);
    }
```

What will be the output of given code?

```
#include <stdio.h>
  void main()
  {
     int a = -5;
     int k = (a++, ++a);
     printf("%d\n", k);
  }
```

what will be the output statement ?

```
#include <stdio.h>
   int main()
   {
      int x = 2;
      x = x << 1;
      printf("%d\n", x);
   }
```

what will be output of given code ?

```
#include <stdio.h>
   int main()
   {
      int x = -2;
      x = x >> 1;
      printf("%d\n", x);
   }
```

What will be the output of given code ?

```
#include <stdio.h>
   int main()   {
      if (~0 == 1)
         printf("yes\n");
```

```
    else
        printf("no\n");
}
```

What will be the output of given code ?

```
#include <stdio.h>
   int main()
   {
      int y = 0;
      if (1 |(y = 1))
         printf("y is %d\n", y);
      else
         printf("%d\n", y);
   }
```

What will be the output of given code ?

```
#include <stdio.h>
   int main()
   {
      int y = 1;
      if (y & (y = 2))
         printf("true %d\n", y);
      else
```

```
        printf("false %d\n", y);


   }
```

What will be the output of given code?

```
#include<stdio.h>
int main()
{  unsigned int res;
   res = (64 >>(2+1-2)) & (~(1<<2));
   printf("%d\n", res);
   return 0;
}
```

What will be the output of given code?

```
#include<stdio.h>
int main()
{   int i=4, j=8;
   printf("%d, %d, %d\n", i|j&j|i, i|j&&j|i, i^j);
   return 0;
}
```

What will be the output of given code ?

```
#include<stdio.h>
int main()
```

```
{    printf("%x\n", -1>>1);
   return 0;
}
```

What will be the output of given code?

```
#include<stdio.h>
int main()
{
   unsigned int m = 32;
   printf("%x\n", ~m);
   return 0;
}
```

What will be the output of given code ?

```
#include<stdio.h>
int main()
{   printf("%x\n", -1<<3);
   return 0;
}
```

What will be the output of given code ?

```
#include<stdio.h>
int main()
{ printf("%d >> %d %d >> %d\n", 4 >> 1, 8 >> 1);
```

```
   return 0;
}
```

What will be the output of given code?

```
#include<stdio.h>
int main()
{   int i=32, j=0x20, k, l, m;
   k=i|j;
   l=i&j;
   m=k^l;
   printf("%d, %d, %d, %d, %d\n", i, j, k, l, m);
   return 0;
}
```

What will be the output of given code ?

```
#include<stdio.h>
int main()
{  printf("%d %d\n", 32<<1, 32<<0);
   printf("%d %d\n", 32<<-1, 32<<-0);
   printf("%d %d\n", 32>>1, 32>>0);
   printf("%d %d\n", 32>>-1, 32>>-0);
   return 0;
}
```