
GrimScythe2001

Page No.
Date
YOUVA

1. What is C language? Who developed C language?

→ The C language is a general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, while a static type system prevents unintended operations.

The C language was developed by Dennis Ritchie at Bell Labs between 1972-73.

2. What do you mean by low level, middle level and high level language? Give example for each.

→ (i) Low level language: A low-level programming language is a programming language that provides little or no abstraction from a computer's instruction set architecture - commands or functions. Generally, this refers to either machine code or assembly language.

(ii) Middle level language: A middle or medium level programming language is that, that interacts with the abstraction layer of a computer system. It binds a gap between a machine level language and high level languages.
for example - C, C++

(iii) High-level language: A high level programming language is a middle level language that takes the concepts of abstraction and high-level constructs to the extreme. They are the logical semantic evolution of mid-level languages for e.g Python, Ruby, SQL

3. Write short note on structure oriented, object oriented and non structure oriented oriented programming language.

→ (i) Structure Oriented - It is a programming paradigm aimed at improving the clarity, quality and development time of a computer program by making extensive use of the structured control flow constructs of selection and repetition, block structures, and subroutines. It follows top-down approach. In structure oriented programming or module oriented programming language, programs are divided into small self contained functions.

(ii) Object oriented - It is a programming designed to focus on data and supports inheritance, encapsulation, abstraction, polymorphism etc. It follows bottom-top approach. In object oriented programming, programs are divided into small entities called objects. For example - C++, JAVA

(iii) Non-Structure oriented - It is historically earliest programming paradigm capable of creating Turning-complete algorithms. There is no specific structure for programming this language. For example - BASIC, COBOL, FORTRAN.

4. What
inte
→ (i)

(ii)

(iii)

4. What do you mean by compiler, assembler and interpreter?

→ (i) Compiler - Compilers are used to convert high level languages (or middle level languages like C, C++) into machine code. This language processor reads the complete source program as a whole in one go and translates it into an equivalent program in machine language. For example - C compiler, C++, Java (GCC).

(ii) Assembler - It is used to translate the program written in Assembly language into machine code. The source program is an input of assembler that contains assembly language instructions.

(iii) Interpreter - The translation of ~~a~~ single statement of source program into machine code is done by language processor and executes it immediately before moving on to the next line is called an interpreter. If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message. The interpreter moves on to the next line for execution only after removal of the error.
For example - python, XML, Matlab

5. What is the only function all C function must contain? What punctuation is used to signal the beginning and end of code blocks? What punctuation ends most lines of C code?

→ The prime function all C programs must contain is main().

The curly brackets { } signal the beginning and end of code blocks.

The semicolon (;) ends most lines of Code

6. What is data type? What is the size of int, float and char data types?

→ Data types refer to an extensive system used for declaring variables or functions of different types. The sizes of various data types are:

(i) int - 2 bytes
(ii) float - 4 bytes
(iii) char - 1 byte

7. Is C programming language structure oriented or procedure oriented language?

→ C programming language is procedure oriented

8. What is the difference between a variable and a constant in C and how do we define them? Show with example.

→ The value for variable varies, however the value for a constant, once defined, never changes throughout the program unless changed from the main source.

9. →

10. →

11. →

variable can be defined normally with only datatype and variablename, however, a constant value requires a keyword 'const' infront of a datatype.

For example- int a = 5; // variable
const int a = 5; // constant

9. What do you mean by declaration and initialisation?

→ Allocating a memory space to a variable is known as declaration, and giving it a first value is known as initialisation.

10. Can we run a program without main function? What is library function?

→ No, we can not run a program without main() function.

Library functions are inbuilt functions which are grouped together and placed in a common place called library.

11. What do you mean by "stdio.h"? Why do we write "return 0" in main function?

→ "stdio.h" is a standard input output header file which tells the compiler to insert the contents of stdio at that particular place.

"return 0" is return value which must be included to return an integer value when the main function is of type int.

12. What is compile time error and run time error?

→ A compile time error is a problem or a bug such as a syntax error or missing file reference that prevents the program from successfully compiling. The compiler produces compile time errors and usually indicates what line of the source code is causing the problem.

If a program's source code has already been compiled into an executable program, it may still have bugs that occur while the program is running, for example, features that don't work, unexpected program behaviour or program crashes. These types of problems are called runtime errors since they occur at runtime.

13. What are logical errors and how does it differ from syntax errors?

→ Logical errors occur when a program does not do what the programmer expects it to do. The program will run without interruption, however the outputs could be illogical or unexpected.

Syntax error occurs when a program does not confirm to the grammar of the programming language, and compiler cannot compile the source file. For example- missing a semicolon or a curly bracket where it is supposed to be.

14. What do you understand by identifiers and keywords?

→ An identifier is a unique name given to a particular variable, function or label of class in the program. Keywords are predefined reserved words, which possess special meaning. Each keyword defines the "type" declared data.

15. What are arithmetic operators? Modulus operator (%) can be used in which data type?

→ The arithmetic operators are some of the C Programming Operator, which are used to perform arithmetic operations that include addition, subtraction, multiplication, division and modulus.

Modulus operator (%) can only be used with int data types, not even with float or double.

16. Arithmetic and logical expressions are evaluated from which direction of a code line?

→ Arithmetic and logical expression are evaluated from left to right direction of a code line.

17. What are logical operators, conditional operators and bitwise operators?

→ Logical operators are used to perform logical operations like AND(&&), OR(||) & NOT(!) on the given expressions.

Conditional operators return one value if condition is true and return another value if it is false.
The syntax for conditional operator is:

(condition? true_value : false_value);

Bitwise operators are used to perform bit operations. Decimal values are converted into binary values which are the sequence of bits and bit wise operators like AND(&), OR(|), XOR(^), NOT(~), left shift(<<) and right shift(>>) work on these bits.

18. What do you mean by derived data type? What is local and global variable in C?

→ Derived datatypes are a little twisted or grouped together datatypes like array and pointer.

Local variable is a variable declared inside a scope which can ~~only~~ be called and used only within that function or scope. Global variable, on the other hand, is declared outside the scope and it can be called and used by any scope ~~from~~ anywhere within a program.

19. What is enum in C?

→ Enumeration (or enum) is a user defined data type in C. It is mainly used to assign names to integral constants, the names make a program easy to read and maintain.

20. What is an operator and operand?

→ An operator is the 'function' that performs the operation, whereas the operand is the input to that function. For example, in the expression, $5+8=13$, '+' is an operator and digits '5' and '8' are operands. In $a=b+c$, '+' is an operator and b and c are operands.

21. Wh
→ In
all
Th
de

22. Wh
→ A
of
ev
Co
or

23. Wh
→ Co
th
in
ex
to
on

24. Wh
→ in
Il
le
of

25. Ho
→ op
Th
loo
&c.

21. What is the use of #define in C?

→ In the C Programming Language, the #define directive allows the definition of macros within your source code. These macro definitions allow constant values to be declared for use throughout our code.

22. What is an endless loop?

→ An endless loop or an infinite loop is a sequence of instructions in a computer program which loops endlessly, either due to the loop having no terminating condition, having one that can never be met, or one that causes the loop to start over.

23. What are control statements?

→ Control statements enable programmers to specify the flow of program control, ie, the order in which the instructions in a program must be executed. They make it possible to make decisions, to perform tasks repeatedly or to jump from one section of code to another.

24. What is || operator and how does it function in a program?

→ || means a logical OR, so it's true if at least one of the many terms is true, false otherwise.

25. How to print 1 to 100 without using conditional operator or conditional statement?

→ This can be done by two ways; one is by using loop, however loop uses conditional statement in it. So, the only way would be to print 1 to 100 individually

26. Which bitwise operator is suitable for
(a) turning on a particular bit in a number
(b) checking whether a particular bit is on or off?

→ (a) Bitwise operator | is suitable for turning on a particular bit in a number.
(b) Bitwise operator & is suitable for checking whether a particular bit is on or off.

28. Describe the difference between = and == symbols in C programming language?

→ '=' symbol is an assignment operator and is used to assign the value of variable or expression '==' symbols is an equal to operator and it is a relation operator used for comparision.

29. Which of the following operator is incorrect and why? ( >= , <= , <> , == ).

→ The operator '<>' is incorrect because it has no operational meaning.

30. Explain increament and decrement operators. what is the difference between the expressions "++a" and "a++"?

→ Increment operator is used to increase the value of a variable by 1, whereas decrement operator is used to decrease the value of a variable by 1.

"++a" expression is pre-increment expression which first increases the value by 1 then does any sort of operation, whereas "a++" is post-increment expression which first performs any sort of operation

and then increases the value by 1.

31. What does the format %.10.2 mean when included in a printf statement?

→ The format "%10.2f" defines that the whole string occupies 10 characters with 2 decimal places. If there is not enough numbers, then spaces are used to the left of the numbers. It is a float type

32. What will be the outcome of the following conditional statement if the value of variables is 10? $ >= 10 && $ < 25 && $ != 12

→ $ >= 10 && $ < 25 && $ != 12

⇒ 10 >= 10 && 10 < 25 && 10 != 12

⇒ 1 && 1 && 1

⇒ 1

33. Give the benefits of if else over conditional operators

→ Conditional operator is generally limited, whereas if else statement is more compact, efficient and convenient. Using conditional operators in more complex code can make code horribly difficult to read, which is not the case with if else conditional statement

34. Differentiate between if else and switch case with an example.

→ The basic difference between switch and if else is that the switch case statement tests the value of variable or expression against a series of different cases or values, until a match is found. If there are no matches found, the optional default case

is executed.

35. Why most of the time, it is necessary to use break statement in switch case?

→ The break statement prevents the execution of the block following the current one, therefore it is necessary to use break in switch case.

36. c = !a ? 10 : 100 ; convert this statement into if else form.

→
```
if (!a)
    c = 10;
else
    c = 100;
```

37. Can we use String string as a case variable? if yes, give an example.

→ No, in C, String cannot be used as a switch case variable.

38. Can we use continue instead of break to move program's execution to the start of switch? Explain.

→ No, continue statement can only be used within a loop. The function of continue is to skip certain statements of the loop to exit out of that loop and start over to the same previous loop. When used with switch case, an error message is displayed asking programmer to use continue only within loops.

39. What
if
x = a

→ x =

40. Is it
it was
value

provide

→ Yes, it
the

After
the

41. Define
has to

→ The te
based
takes
makes
block
Simila
In exp
have to

39. What value will be assigned to the variable x if a = 10, b = 20, c = 30, d = 40 for the expression x = a/b + c*d - c?

→ x = a/b + c*d - c
= 10/20 + 30*40 - 30
= 0 + 1200 - 30
= 1170

∴ The value 1170 will be assigned to variable X.

40. Is it possible to initialise the variable at the time it was declared? For initialisation a=2, c=1, the value of a and c after this code will be

c = (c)? a=0 : 2;

provided a = int, var1 = double, var2 = float.

→ Yes, it is possible to initialise the variable at the time it was declared.

After the execution of code c = (c)? a=0 : 2, the new value of a and c will be 0,
i.e., a=0 and c=0;

41. Define ternary operator in C. Which expression has to be present in the following?

exp1 ? exp2 : exp3

→ The ternary operator is used to execute code based on the result of a binary condition. It takes in a binary condition as input, which makes it similar to an 'if-else' control flow block. It also, however, returns a value, behaving similar to a function.
In expression exp1? exp2 : exp3, all three have to be present necessarily for operator to work.

42. Value of c after the following expression (initialisations a=1, b=2, c=1);

```
c += (-c) ? a : b;
```

→ The value of c will be 2 (i.e., b) because,

$c += (-c)\ ?\ a : b$

$\Rightarrow c = c + (-c)\ ?\ a : b$

$\Rightarrow c = 1 - 1\ ?\ a : b$

$\Rightarrow c = 0\ ?\ a : b$

where, 0 denotes false

So, $c = b$

i.e., $c = 2$.

43. What is the type of the below assignment expression if x is of type float, y is of type int?

→ Since y is of type int and the final answered is saved under y, the type of the assignment expression `y = x + y;` would be int.

44. Find the error in the following programs:

→ (a)

```
#include<stdio.h>
int main()
{
    char ch;
    int i;
    scanf("%d", &i);
    scanf("%c", &ch);
    printf("%c %d", ch, i);
    return 0;
}
```

(b)
```
# include <stdio.h>
~~int main()~~                    // main() function error
void main()
{
  printf("Okay");
}
```

(c) ~~# include <stdio.h>~~
~~int # include <~~

(c)
```
# include <stdio.h>
int main()
{
    int x = 10;
    Static int y = 10 ;    // Static variable needs initialisation
    if ( x == y)
        printf(" Equal");
    else if ( x > y)
    printf(" Greater");
     else
        printf(" Less");
     getchar();
}
```

45. What is the output of the following C code?

```
# include <stdio.h>
int main()
{
    int x = 2, y = 0;
    int z = (y++) ? y == 188 x : 0;
    printf("%d\n", z);
    return 0;
}
```

→ 0

46. What is the output of this C code?

```
# include <stdio.h>
void main()
{
    int a = 3;
    int b = ++a + a++ + --a;
    printf("Value of b is %d", b);
}
```

→ Value of b is 13. // Gcc Compiler

47. What are the outputs of the following partial programs?

```
char ch = 'A';
printf("%s", ch);
```

→ The GCC compiler does not show any output, otherwise the output would be a garbage value or in some cases, error.

48. Find the output of given C code.

→
```
#include <stdio.h>
int main()
{
    int number = 3, che_ = 3;
    if (number < 0 || che_)
    {
        printf ("You entered %d.\n", number);
    }
    printf ("The if Statement is easy.");
    return 0;
}
```

→ You entered 3.
The if Statement is easy.

49. Find the output of given C code.

```
#include <stdio.h>
int main()
{
    int i = 5, j = 6, k = 7;
    if ( i > j == k)
        printf ("%d %d %d", i++, ++j, --k);
    else
        printf ("%d %d %d", i, j, k);
    return 0;
}
```

→ 5 6 7

50. What will be the output of the C program?

```
# include <stdio.h>
int main()
{
    int i = 5;
    if (i== 3,4)
        printf("Hi");
    else
        printf("No Hi");
    return 0;
}
```

→ Hi

51. What will be the output of the C program?

```
# include <stdio.h>
int main()
{
    int i = 1;
    i++;
    Switch(i--)
    {
        case 1:
            printf("case 1 executed");
            break;
        case 2:
            printf("case 2 executed");
            break;
        default:
            printf("default block executed");
            break;
    }
    return 0; }
```

→ case 2 executed

52. What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    switch (312)
    {
      case 1:
        printf("case 1 executed");
      case 2:
        printf(" case 2 executed");
        break;
      default:
        printf("Default block executed");
    }
    return 0;
}
```

→ case 1 executed case 2 executed.

53. What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    switch (25)
    {
        case 25L : printf("25L");
                   break;
        case 25.0 : printf("25.0");
                    break;
        default : printf("Nothing");
                  break;
```

```
    }
    return 0;
}
```

→ Error message displayed for not allocating float value to be defined as case variable. However, if the program were to run, the output would be 25L.

54. What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    if (printf("0"))
        printf("inside if block");
    else
        printf("inside else block");
    return 0;
}
```

→ 0 inside if block

55. What will be the output of the C program?

```
#include<stdio.h>
int main()
{
    int i = 5, j=4;
    if (!printf(""))
        printf("%d %d", i, j);
    else
        printf("%d %d", i++, ++j);
    return 0;
}
```

→ 5 4

56. What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    int i=1, j=0;
    if ( i-- == j)
        printf("i = %d ", --i);
    else
        printf ("j = %d ", ++j );
    return 0;
}
```

→ j = 1

57. What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    float me = 5.25;
    double you = 5.25;
    if (me == you)
    printf ("Appy is best");
    else
        break;
    return 0;
}
```

→ Error message displayed for break; statement to be used only with loop or switch cases.
However, if the program were to run, output:
Appy is best.

58. What will be the output of the C program?

```
# include <stdio.h>
int main()
{
    if ("May I Get in")
        printf ("yes, Get in");
    else
        printf ("No");
    return 0;
}
```

→ Yes, Get in

59. What will be the output of the C program?

```
# include <stdio.h>
int main()
{
    int i = 2;
    if (i == (1,2))
    printf ("Hai");
    else
    printf ("No Hai");
    return 0;
}
```

→ Hai

60. What will be the output of the following C code?

```
#include <stdio.h>
void main()
{
    if (!print(""))
        printf("Okkk");
    else
        printf("Hiii");
}
```

→ Okkk

61. What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
    int a = -20;
    if (a>0)
        printf("%d is a positive integer\n", a);
    else if (a<0)
        printf("%d is a negative integer\n", a);
    else
        printf("Its ZERO\n");
    return 0;
}
```

→ -20 is a negative integer.

62. What will be the output of the following C code:

```
#include <stdio.h>
void main()
{
    int a=5, b=10, c=1;
    if (a&&b>c)
    {
        printf("It's too easy you can do it");
    }
    else
    {   break;
    }
}
```

→ It's too easy you can do it

63. Explain the output of following code in 2-3 line

```
#include <stdio.h>
void main()
{
    if (sizeof(void))
        printf("IIT");
    else
        printf("NIT");
}
```

→ IIT

64. Using Switch statement, write a program that displays the following menu for the food items available to take order from the customer:

- B = Burger
- F = French Fries
- P = Pizza
- S = Sandwiches

The program inputs the type of food and quantity. It finally displays the total charges for the order according to following criteria:

- Burger = Rs. 200
- French fries = Rs. 50
- Pizza = Rs. 500
- Sandwiches = Rs. 15

→

```
#include <stdio.h>
void main()
{
    char opt;
    int qty;
    printf(" *** MENU ***");
    printf("\nBurger (B) ");
    printf ("\n French Fries (F)");
    printf("\n Pizza (P)");
    printf ("\n Sandwiches (S)");
    printf("\n\n Select your order: ");
    scanf(" %c ",&opt);
    if ( opt >= 97 && opt <= 122)
        opt = opt - 32;
    printf(" Enter quantity: ");
    scanf(" %d ", &qty);
```

```
        switch ( opt)
        {
            case 'B' : printf(" Total amount = %d", (200*qty));
                       break;
            case 'f' : printf(" Total amount= %d", (50*qty));
                       break;
            case 'P' : printf("Total amount= %d", (500*qty));
                       break;
            case 'S' : printf(" Total amount= %d", (15*qty));
                       break;
        }
}
```

65. What is the output of the following program?

```
void main()
{
    int x = 40, y = 30 , z = 80;
    if (x < y < z)
        printf (" \n Hello world ");
    else
        printf (" \n Goodbye ");
}
```

→ Helloworld

66. What will be the output of the C program?

```
# include <stdio.h>
int main()
{
    short int si = 1;
    switch ( ++si - si++ )
    {
        case 1L:
            printf("First");
            break;
        case 2L:
            printf("Second");
            break;
        default:
            printf("Bye");
            break;
    }
    return 0;
}
```

→ Bye

67. What will be the output of the C program?

```
#include <stdio.h>
int main()
{
    int num = 0;
    if (num++, num--, ++num)
    switch (num)
    {
        case 1: printf ("case one");
                break;
        case 2: printf("case two");
                break;
        default: printf(" default block");
                break;
    }
    return 0;
}
```

→ case one

68. Spot the error in the given code. Explain:

```
#include <stdio.h>
int main() {
int i;
if (true)
    printf ("This will work");
else
    printf(" This will not work");
return 0;
}
```

→ The line if (true) has error as the variable true has not been declared or defined. It will, however, work if instead of

true, "true" is written.

69. What will be the output when you will execute following C code and explain its output?

```
#include <stdio.h>
void main()
{
    int a=5;
    a = a>=4;
    Switch (2)
    {
        case 0: int a=8;
        case 1: int a=10;
        case 2: ++a;
        case 3: printf ("%d", a);
    }
}
```

→ Correction in case 0 and case 1 because multiple declaration of variable a as int. However, neglecting the error, the output will be:

2

70. What will be the output of following program?

```
# include <stdio.h>
# define FALSE -1
# define NULL 0
# define TRUE 1
int main()
{
    if (NULL)
        printf ("NULL");
    else if (FALSE)
        printf ("TRUE");
    else
        printf ("FALSE");
    return 0;
}
```

→ TRUE