CH # 2

USER INTERFACE

2.1 INPUT / OUTPUT (I/O) FUNCTION

2.1.1 PRINTF ()
2.1.2 FORMAT SPECIFIERS
2.1.3 SCANF ()
2.1.4 GETCH ()
2.1.5 STATEMENT TERMINATOR
2.1.6 ESCAPE SEQUENCE

LONG QUESTIONS

1. Explain *printf* ( ) function with some examples. (K.B+U.B+A.B)

Ans. *printf* is a built-in function in 'C' programming language to show output on screen. Its name comes from "***Print formatted***" that is used to print the formatted output on screen. All data types can be displayed with ***printf*** function.

Syntax:

printf (control string, list of arguments) The control string consists of the following:

- Text
- The format specifier
- The escape sequence.

Text is written within double quotes. It specifies the message that is to be printed along with the values. **Format specifier** specifies the format according to which a value is to be printed. **Escape sequence** controls the printing on the output device. **List of arguments** consists of a list of variables or arithmetic expressions, separated by commas, whose values are to be printed. The values are printed according to the corresponding format specifier. The first format specifier applies to the first argument, the second to the second argument and so on. The arguments the ***printf* ( )** function are optional. When ***printf* ( )** function is used to print only a message, then the arguments are omitted.

Example:

```
# include <stdio. h>
void main ()
{
printf ("Hello World");
}
```

Output:

Hello World

In this example, printf function is used to display **Hello World** on computer screen. Whatever we write inside the double quotes (" ") in the *printf* () function, gets displayed on computer screen.

2. Explain format specifier in C language in detail. (K.B+U.B)

Ans: A format specifier is computer code that tells about the data type, field width and the format according to which a value is to be printed or read from an input device. A list of commonly used format specifiers is given below.

- %d (decimal integer)
- %i (integer)
- %1d (long decimal integer)

- %f floating-point (decimal notation)
- %g floating-point (exponential notation)
- %e floating-point (%f or %g, whichever is shorter)
- %c (single character)
- %s (string)

Example:

```
#include < stdio.h>
void main ()
{
float height = 5.8
int age = 35;
printf ("My age is %d and my height is %f", age,height);
}
```

Output:

My age is 35 and my height is 5.800000.We can observe that while displaying output, first format specifier is replaced with the value of first variable/ data after the ending quotation mark i.e ***age*** in the above example, and second format specifier is replaced with the second variable/ data i.e ***height***.

3. Explain *scanf* ( ) function with examples. (K.B+U.B)

Ans:

SCANF ( ) FUNCTION

scanf is a built-in function in 'C' language that takes input from user into the variables. We specify the expected input data type in scanf function with the help of format specifier. If user enters integer data type, format specifier mentioned in scanf must be %d or %i.

Syntax:

scanf ("control/prompt string", list of variables);

The control string specifies the format specifiers. It is written within double quotes. The control string in ***scanf*** **( )** function is different from ***printf*** **( )** function. In scanf **( )** function, strings cannot be given.

We can take multiple inputs using a single scanf function e.g. look at the following statement. scanf ("%d%d%f", &a, &b, &c);

It takes input into two integer type variables a and b, and one float type variable c. After each input, user should enter a "space" or "enter" key. After all the inputs user must press enter key.

Example:

```
#include <stdio.h>
void main ()
{
char grade;
scanf ("%c",&grade);
}
```

In this example, %c format specifier is used to specify character type for the input

variable. Input entered by user is saved in variable grade.

Solution:

```
# include <stdio.h>
void main ()
{
int roll_number;
float percentage;
char grade;
print (“Enter your roll number”);
scanf (“%d, & roll_number);
printf (“Enter percentage of marks”);
printf (“Enter grade”);
scanf (“%c”, & grade);
printf (“Roll no \t: \t %d \n”, roll_number);
printf (“Percentage \t: \t %f%”, percentage);
printf (“Grade \t %c”, grade);
}
```

4. What is *getch* () function? (K.B+A.B)

Ans:

Getch ( ) Function

getch () function is used to read a character from user. The character entered by user does not displayed on screen. This function is generally used to hold the execution of program because the program does not continue further until the user types a key. To use this function, we need to include the library conio.h in the header section of program.

Example Code

```
# include <stdio.h>
# include <conio.h>
void main ()
}
printf (“Enter any key if you want to exit program”);
getch ()
}
```

5. Explain escape sequence in detail with one program example. (K.B+U.B+A.B)

Ans:

Escape Sequence

The special characters used in ‘C’ language to control printing on the output device are called escape sequences. Escape sequences are used in *printf* function inside the double quotes (“ ”). They force *printf*() to change its normal behavior of showing output. These characters are not printed. These are used inside the control string.

Control Character:

An escape sequence is a combination of a backslash (\) and a code character. The backslash is called the control character. A list of commonly used escape sequences is given below with their meanings.

Commonly used Escape sequences are shown in the table

Sequence	Purpose	Sequence	Purpose

\'	Displays Single Quote	\a	Generates an alert sound
\\	Displays Back slash (\)	\b	Removes previous char
\n	Creates new Line	\t	Moves to next tab

Example:

```
# include <stdio.h>
void main ()
{
printf ("My name is Ali, \n");
printf ("I live in Lahore,");
}
```

Output

My name is Ali
I live in Lahore.

SHORT QUESTIONS

Q.1 What is meant by input and output in programming? (K.B)

Ans:

Input & Output

In programming, input means to enter or feed data in a computer program and output is what the computer produces after processing the data. 'C' language provides many input/output functions to provide interaction between a program and user.

Q.2 Define output function in 'C' language. (K.B)

Ans:

Output Function

"In computer programming, output means to display information on screen or print on printer. 'C' language provides the ***printf*( ), *putchar*( )** and ***puts*( )** functions to output information on computer screen".

Q.3 Name some output function in 'C' language. (K.B)

Ans:

Output Function

Some commonly used output function in C language are:

- printf( )
- putchar( )
- puts( )

Q.4 Define printf( ) function? (K.B)

Ans:

printf( ) Function

printf is a built-in function in 'C' programming language to show output on screen. Its name comes from "Print formatted" that is used to print the formatted output on screen. All data types can be displayed with printf function.

Syntax:

printf ("control string", list of arguments);

Q.5 What is input function in 'C' language? (K.B)

Ans:

scanf Function

In computer programming input means to feed data into program through an input device. 'C' language provides the scanf( ), getch( ), getchar( ) and gets( ) functions to input data.

Q.6 Enlist some input functions in C language. (K.B+U.B+A.B)

Ans: **Input Function**

Some commonly used input functions in 'C' language are:

- scanf( )
- getch( )
- getchar( )
- gets( )

Q.7 What is role of format specifier in 'C' language? (K.B)

Ans: **Format Specifier**

A format specifier is computer code that tells about the data type, field width and the format according to which a value is to be printed or read from an input device

Q.8 Enlist commonly used format specifiers. (K.B)

Ans: **Format Specifiers**

A list of commonly used format specifiers is given below:

- %d decimal integer
- %i integer
- %ld long decimal integer
- %f floating-point (decimal notation)
- %g floating-point (exponential notation)
- %e floating-point (%f of %g, whichever is shorter)
- %c single character
- %s string

Q.9 Differentiate between integer and floating-point format specifier. (K.B+U.B)

Ans: **DIFFERENTIATION**

Following are the differences between integer and floating-point format specifier:

Integer Format Specifier	Floating Point Format Specifier
• The format specifier %d is used to read or print a decimal integer.	• The format specifier %f is used to read and print floating point numbers in decimal notation with a precision of 6 digits after the decimal point.
• The format specifier %ld is used with long integers.	• The format specifier %e is used to read and print floating-point numbers in exponential notation.
• The format specifier %i is used to read or print an integer.	• The format specifier %g is used to print floating-point numbers in decimal or exponential notation whichever is shorter.

Q.10 What is the use of character format specifier? (K.B+U.B)

Ans: **Format Specifier**

The character format specifier, %c is used to read or print a single character.

Q.11 What is *scanf( )* function? (K.B+U.B)

Ans: **Scanf ( ) Function**

scanf is a built in function in C language that takes input from user into the variables. We specify the expected input data type in *scanf* function with the help of format specifier. If user enters integer data type, format specifier mentioned in scanf must be %d or %i.

Syntax:

scanf ("control string", "list of variables");

Q.12 What is *getch*( ) functions? (K.B+U.B)

Ans: **Getch ( ) Functions**

getch () function is used to read a character from user. The character entered by user does not displayed on screen. This function is generally used to hold the execution of program because the program does not continue further until the user types a key.

Q.13 What is meant by statement terminator in C language? (K.B+U.B)

Ans: **Statement Terminator**

A statement terminator is identifier for compiler which identifies end of a line. In 'C' language semi colon (;) is used as statement terminator. If we do not end each statement with a **(;)** it results into an error/s.

Q.14 Enlist commonly used escape sequence. (K.B+U.B+A.B)

Ans: **Escape Sequence**

Following are the some commonly used escape sequence in 'C' language:

Sequence	Purpose	Sequence	Purpose
\'	Displays Single Quote	\a	Generates an alert sound
\\	Displays Back slash (\)	\b	Removes previous char
\n	Creates new Line	\t	Moves to next tab

Q.15 What is the purpose of escape sequence in 'C' language? (K.B+A.B)

Ans: **Escape Sequence**

The special characters used in 'C' language to control printing on the output device are called escape sequences. Escape sequences are used in ***printf*** function inside the "and." they force ***printf*()** to change its normal behavior of showing output. These characters are not printed. These are used inside the control string.

MUTLIPLE CHOICE QUESTIONS

1. Which operations are involved in communication between computer and user? (K.B)

(A) Input/output Operation (B) Storage Operation
(C) Processing Operation (D) Controlling Operation

2. 'C' language uses function to provide interaction between program and user: (K.B)

(A) Input Function (B) Output Function (C) Format Specifier (D) Both A & B

3. Which one of the following is not a output function: (K.B+U.B)

(A) printf( ) (B) putchar ( ) (C) puts ( ) (D) getch( )

4. In 'C' program to display text, variable, constant and expression we use: (K.B)

(A) printf( ) (B) puts( ) (C) putchar( ) (D) strcpy( )

5. Which function is used to get values into variables from the keyboard during the execution of a program: (K.B+U.B)

(A) scanf( ) (B) getche( ) (C) printf( ) (D) getchec( )

6. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main()
{
printf("Hello World! %d \n", x);
return 0;
}
```

(A) Hello World! x; (B) Hello World! followed by a Junk Value
(C) Compile Time Error (D) Hello World!

7. The format identifier '%i' is also used for data type: (K.B)

(A) char (B) int (C) float (D) double

8. Find the output of the following program. (A.B)

```
void main()
{
int i=065, j=65;
printf("%d %d", i, j);
}
```

(A) 53 65 (B) 65 65 (C) 065 65 (D) 053 65

9. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int var = 010;
printf("%d", var);
}
```

(A.B)

(A) 2 (B) 8 (C) 9 (D) 10

10. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
printf("sanfoundary\rclass\n");
return 0;
}
```

(A) sanfoundaryclass (B) sanndry (C) sanfoundarclass (D) sanfoundary

11. *scanf*( ) returns as its value:

(A) Number of Successfully Matched and Assigned Input Items
(B) Nothing
(C) Number of Characters Properly Printed
(D) Error

12. '%f' access specifier is used for: (A.B)

(A) Strings (B) Integral Types (C) Floating Types (D) Character Types

13. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
printf("%.0f",2.89);
```

(A) 2.890000 (B) 2.89 (C) 2 (D) 3

14. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h> int main( )
{
float a = 2.455555555555
printf("%f", a);
}
```

(A) 2.455555 (B) 2.455556 (C) 2.456 (D) 2.46

15. What will be the output of following 'C' code? (K.B+U.B+A.B)

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

(A) Compile Time Error (B) -1
(C) 1 (D) Implementation Defined

16. What will be the output of following 'C' code? (K.B+U.B+A.B)

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

(A) Compile Time Error (B) 3
(C) 1 (D) Implementation Defined

17. A symbol used in 'C' with *scanf*( ) function: (K.B+A.B)
(A) # (B)! (C) & (D) i

18. Which one of the following symbol is used as a statement terminator? (K.B)
(A) : (B) # (C) ; (D) }

19. A computer code that tells about data type field width and function is called _____.(K.B)
(A) Reserve Word (B) Format Specifier (C) Escape Sequence (D) String Field

20. Which format specifier is used for decimal integer data? (K.B+A.B)
(A) % (B) % Id (C) %d (D) %C

21. A format specifier use for integer_____. (K.B+A.B)
(A) %d (B) i% (C) %f (D) %c

22. A format specifier use for long decimal integer _____. (K.B+A.B)
(A) %d (B) %i (C) %id (D) %f

23. Format specifier used for string variable _____. (K.B+A.B)
(A) %s (B) %c (C) %e (D) %g

24. A format specifier used for floating point exponential notation _____. (K.B+A.B)
(A) %e (B) %g (C) %c (D) %f

25. A format specifier is used for single character _____. (K.B+A.B)
(A) %c (B) %s (C) %e (D) %

26. The special character used in C language to control printing on output devices are called _____ (K.B+A.B)
(A) Format Specifier (B) Escape (C) String Format (D) Control String

27. Which Escape sequence is used to produce a test (bell) sound? (K.B+A.B)
(A) \a (B) \b (C) \n (D) \r

28. A escape sequence used to move cursor back word by one position. _____. (K.B+A.B)
(A) \b (B) \n (C) \r (D) \t

29. Which escape sequence is used to move cursor to the beginning? (K.B+A.B)

(A) \a (B) \b (C) \n (D) \r

30. A escape sequence used to move cursor to the next horizontal tabular position:(K.B+A.B)
(A) \t (B) \\ (C) \n (D) \

31. Which escape sequence produced a back slash? (K.B+A.B)
(A) \ (B) \\ (C) \' (D) \"

32. Escape sequences are prefixed with _____. (K.B+A.B)
(A) % (B) \ (C) \\ (D) \"

33. Format specifiers are used for _____. (K.B+A.B)
(A) Variable (B) Constants
(C) Both (D) All of these

34. We can take_____ inputs using single scan function. (K.B+A.B)
(A) Single (B) Multiple (C) Ten (D) None of these

35. If we forget '&' symbol in scanf function it is _____. (K.B+A.B)
(A) Error (B) Not an error (C) Code (D) None of these

36. Escape sequence used for new line _____. (K.B+A.B)
(A) \a (B) \b (C) \n (D) None of these

37. Escape sequence to display a single quote _____. (K.B+A.B)
(A) \a (B) \' (C) \, (D);

38. A tab is a collection of _____ spaces. (K.B+A.B)
(A) 8 (B) 12 (C) 16 (D) 32

2.2 OPERATORS

LONG QUESTIONS

1. What is the use of arithmetic operators? Also enlist its types. (K.B+A.B)

Ans:

ARITHMETIC OPERATORS

Arithmetic operators are used to perform arithmetic operations that include addition, subtraction, multiplication, division and also to find the remainder obtained when an integer is divided by another integer.

Types:

The types of arithmetic operators used in C language are described here:

Operator	Name	Description
/	Division Operator	It is used to divide the value on left side by the value on right side.
*	Multiplication Operator	It is used to multiply two values.
+	Addition Operator	It is used to add two values.
-	Subtraction Operator	It is used to subtract the value on right side from the value on left side.
%	Modulus Operator	It gives remainder value after dividing the left operand by right operand.

Program

```
/* This program takes an input the price of a box of chocolates and the total number of chocolates in the box. The program finds and displays the price of one chocolate */
#include <stdio.h>
void main ()
{
```

```
float box_price, num_of_chocolates, unit_price;
print ("Please enter the price of whole box of chocolates:");
scanf ("%", & box_price);
printf ("Please enter the number of chocolates in the box:");
scanf ("&f", &num_of_chocolates);
unit_price = box_price / num_of_chocolates;
printf ("The price of a single chocolate is %f", unit_price);
}
```

Output:

Please enter the price of whole box of chocolates: 150
Please enter the number of chocolates in the box: 50
The price of a single chocolate is 3.000000

2. Explain relational operators in detail. (K.B+U.B)

Ans:

RELATIONAL OPERATORS

Relational operators compare two values to determine the relationship between values. Relational operators identify either the values are equal, not equal, greater than or less than one another. 'C' language allows us to perform relational operators on numeric and char type data. Table presents relational operators in C language and their descriptions:

Relational Operator	Description
= =	Equal to
! =	Not equal
>	Greater than
<	Less than
> =	Greater than equal to
< =	Less than equal to

Relation operators perform operations on two operands and return the result in Boolean expression (***TRUE OR FALSE***). A true value is represented by 1, whereas a false value is represented by a 0. This concept is further illustrated as followed

Examples:

Relation Expression	Explanation	Result
5 = =5	5 is equal to 5 ?	TRUE
5 ! = 7	5 is not equal to 7?	TRUE
5 > = 5	5 is greater than or equal to 5?	TRUE
5 < = 4	5 is less than or equal to 4?	FALSE

3. Explain Logical operators in detail (K.B+U.B)

LOGICAL OPERATORS

Logical operators are used for building compound conditions. We have seen before that a single condition is built using a relational operator in an expression. If we need to build more than one condition for some action to take place in programming, then we have to form compound condition.

Types of Logical Operators:

There are three types of logical operators. These are:

Operator	Definition
& &	AND
||	OR
!	NOT

AND Operator (&&):

AND operator && takes two Boolean expressions as operands and produces the result TRUE if both of its operands are TRUE. It returns FALSE if any of the operands is FALSE. Table shows the truth table for AND operator.

Expression	Result
FALSE && FALSE	FALSE
FALSE && TRUE	FALSE
TRUE && FALSE	FALSE
TRUE && TRUE	TRUE

OR Operator (||):

OR operator accepts Boolean expression and returns TRUE if at least one of the operands it TRUE. Table shows that truth table for OR operator.

Expression	Result
FALSE || FALSE	FALSE
FALSE || TRUE	TRUE
TRUE || FALSE	TRUE
TRUE || TRUE	TRUE

NOT Operator (!):

NOT operator negates or reverses the value of Boolean expression. It makes it TRUE, if it is FALSE and FALSE if it is TRUE. Table present the truth table for Not operator.

EXPRESSION	Result
!(TRUE)	FALSE
!(FALSE)	TRUE

Examples of Logical Operators:

Logical Expression	Explanation	Result
3 < 4 && 7 > 8	3 is less than 4 AND 7 is greater than 8?	FALSE
3 = = 4 || 3 > 1	3 is equal to 4 OR 3 is greater than 1?	TRUE
!(4 > 2 || 2 = = 2)	NOT (4 is greater than 2 OR 2 is equal to 2)?	FALSE

Example:

The expression:

!(a<b)

Will be true if **a** is not less than **b**. In other words, the condition will be true if a is greater than or equal to **b**. The same condition can also be written as **(a>=b)** which is easy to understand.

SHORT QUESTIONS

Q.1 What is an expression? (K.B)

Ans:

EXPRESSION

Expressions consist of constants and variables combined together with operators.

Example:

Percentage is = obtained_marks / total_marks*100;

Q.2 Define operators in 'C' language.

Ans: **OPERATORS IN 'C' LANGUAGE**

"An operator is a symbol used to command the computer to perform a certain mathematical or logical operation. Operators are used to operate on data and variables".

Types:

Some commonly used operators in 'C' language are:

- Arithmetic operators
- Assignment operators
- Relational operators
- Logical operators

Q.3 Name some operators of 'C' language. (K.B)

Ans: **'C' LANGUAGE OPERATORS**

Following are some commonly used operators in C language:

- Arithmetic operators
- Assignment operators
- Relational operators
- Logical operators

Q.4 What are arithmetic operators in 'C' language? (K.B)

Ans: **ARITHMETIC OPERATORS**

Arithmetic operators are used to perform arithmetic operations that include addition, subtraction, multiplication, division and also to find the remainder obtained when an integer is divided by another integer.

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder (Modulus) Operator

Q.5 What is Modulus Operator? (K.B)

Ans: **MODULUS OPERATOR**

Modulus operator (%) performs division of left operand by the right operand and returns the remainder value after division. Modulus operator works on integer data types.

- **int REM = 14 % 3;**

As, when we divide 14 by 3, we get a remainder of 2, so the value stored in variable REM is 2.

Q.6 What are logical operators in 'C' language? (K.B)

Ans: **LOGICAL OPERATORS**

Logical operators are used for building compound conditions. We have seen before that a single condition is built using a relational operator in an expression. If we need to build more than one condition for some action to take place in programming, then we have to form compound condition.

Types:

There are three logical operators:

- AND (&&)
- OR (||)
- NOT (!)

Q.7 Differentiate between relational and logical operators. (K.B+U.B)

Ans: **DIFFERENTIATION**

Following are the differences between relational and logical operators:

Relational Operator	Logical Operator
• Relational operators are used to compare two values of the same types.	• Logical operators are used for building compound conditions.
• After evaluation of a relational expression, the result produced is either True or False.	• These operator always evaluates results in the form of true and false.
• Relational operator include ==, !=, <, >, <= and >=.	• Logical operators include, AND (&&), OR (||), and NOT (!).

Q.8 Enlist name of Relational operators. **(K.B)**

Ans: **RELATIONAL OPERATORS**

Relational Operator	Description
= =	Equal to
! =	Not equal
>	Greater than
<	Less than
> =	Greater than equal to
< =	Less than equal to

Q.9 Enlist name of logical operators. **(K.B)**

Ans: **LOGICAL OPERATORS**

There are three types of logical operators. These are described below.

Operator	Definition
& &	AND
||	OR
!	NOT

Q.10 Draw Table for AND Operator. **(K.B)**

Ans: **AND OPERATOR**

Expression	Result
FALSE && FALSE	FALSE
FALSE && TRUE	FALSE
TRUE && FALSE	FALSE
TRUE && TRUE	TRUE

Q.11 Draw Table for OR Operator. **(K.B)**

Ans: **OR OPERATORS**

Expression	Result
FALSE || FALSE	FALSE
FALSE || TRUE	TRUE
TRUE || FALSE	TRUE
TRUE || TRUE	TRUE

Q.12 Draw Table for NOT Operator. **(K.B)**

Ans: **NOT OPERATORS**

Expression	Result
!(TRUE)	FALSE
!(FALSE)	TRUE

Q.13 Write down the differences between Assignment operator (=) and Equal to operators (= =)? (K.B+U.B)

Ans: Following are the differences between assignment operator and equal to operator.

Assignment Operator (=)	Equal to Operators (= =)
Assignment operator is used to assign a value to a variable, or assign a value of variable to another variable.	In C language, = = operator is used to check for equality of two expressions
Equal sign (=) is used as assignment operator in C.	Double Equal sign (= =) is used as equal to operator
Consider the following example: int sum = 5;	Consider the following example: 3+2 = = 5;

Q.14 Differentiate between Unary and Binary operators. (K.B+U.B)

Ans:

Unary Operators:	Binary Operators:
Unary operators are applied over one operand only	Binary operators require two operands to perform the operation
Example Logical not (!) operator has only one operand. Sign operator (-) is another example of a unary operator e.g. -5.	**Example** All the arithmetic operators, and relational operators are binary operators. The logical operator && and || are also binary operators.

Q.15 What do you know about operator precedence in 'C' language? (K.B+U.B)

Ans: **ORDER OF PRECEDENCE OF OPERATORS**

If there are multiple operators in an expression, the question arises that which operator is evaluated first. To solve this issue, a precedence has been given to each operator. An operator with higher precedence is evaluated before the operator with lower precedence. In case of equal precedence, the operator at left side is evaluated before the operator at right side.

Example:

Result = 18 / 2 * 3 + 7% 3 + (5 * 4);

Operator	Precedence
()	1
!	2
*, /, %	3
+, -	4
>, <, >=, <=	5
==, !=	6
&&	7
||	8
=	9

MULTIPLE CHOICE QUESTIONS

1. Which of the following is related to expression? (K.B)
(A) Control / Variable (B) Operators
(C) Mathematical Operation (D) All of these

2. 'C' language has ___ main types of operators. (K.B)
(A) 3 (B) 4 (C) 5 (D) 6

3. In 'C' language we used to program arithmetic operations by using _____. (K.B)
(A) Arithmetic Operators (B) Assignment
(C) Relational Operator (D) Logical Operator

4. Assignment operators is 'C' language is denoted by _____. (K.B+U.B)
(A) = (B) == (C) <=i (D) =>

5. In relational operators equal to is denoted by: (K.B+U.B)
(A) == (B) = (C)! = (D) >=

6. Which operators are used to build compound condition? (K.B+U.B)
(A) Assignment (B) Relational (C) Logical (D) Increment

7. In 'C' language "AND" logical operator is denoted by using symbol _____.(K.B+U.B)
(A) && (B) // (C)! (D) #

8. In 'C' language "OR" logical operator is denoted by _____. (K.B+U.B)
(A) && (B) || (C) # (D)!

9. Which logical operator uses single expression? (K.B+U.B)
(A) AND (B) OR (C) NOT (D) NAND

10. Expression !(a<b) is an example of _____. (K.B+U.B)
(A) AND (B) OR (C) NOT (D) XOR

11. Which one operator has highest order of precedence in 'C'? (K.B+U.B)
(A) ++, -- (B) *, / % (C) +, - (D) ||

12. Which one of the following operators of C has lowest order of precedence?(K.B+U.B)
(A) Logical Operator (B) Assignment
(C) Relational Operators (D) Increment / Decrement Operator

13. What will be the output value of x in the following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int x = 5 * 9 / 3 + 9;
}
```

(A) 3.75 (B) Depends on Compiler
(C) 24 (D) 3

14. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int x = 5.3 % 2;
printf("value of x is %d", x);
```

(A) Value of x is 2.3 (B) Value of x is 1
(C) Value of x is 0.3 (D) Compile Time Error

15. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int y = 3;
int x = 5 % 2 * 3 / 2;
printf(:Value of x is %d:, x);
}
```

(A) Value of x is 1 (B) Value of x is 2
(C) Value of x is 3 (D) Compile Time Error

16. The precedence of arithmetic operators is (from Highest to Lowest): (K.B+U.B+A.B)
(A) %, *, /, +, - (B) %, +, /, *, - (C) +, -, %, *, / (D) %, +, -, *, /

17. Which one of the following is not an arithmetic operation? (K.B+U.B+A.B)
(A) a*=10; (B) a/=10; (C) a != 10; (D) a%=10;

18. Which one of the following data type will throw an error on modulus operation (%)?(K.B+U.B+A.B)
(A) Char (B) Short (C) int (D) float

19. Which one of the following are the fundamental arithmetic operators?(K.B+U.B+A.B)
(A) +, - (B) +,-,% (C) +,-,*,/ (D) +,-,*,/,%

20. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int a = 10;
double b = 5.6;
int c;
c = a + b;
printf("%d", c);
}
```

(A) 15 (B) 16 (C) 15.6 (D) 10

21. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int a = 10, b = 5 , c = 5;
int d;
d = a == (b+c);
printf("%d", d);
```

(A) 1 (B) 5 (C) 5 (D) 25

22. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int x = 1, y= 0, z=5;
int a = x && y || z++;
printf("%d", z);
}
```

(A) 6 (B) 5 (C) 0 (D) Varies

23. What will be the output of following 'C' code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int x = 1, z= 3;
int y = x << 3;
printf("%d\n", y);
}
```

(A) -2147483648 (B) -1 (C) 1 (D) Runtime Error

24. Result of a logical or relational expression in C language is: (K.B+U.B+A.B)

(A) True or False

(B) 0 or 1

(C) 0 If Expression is False and any Positive Number if Expression is True

(D) Both a & b

25. What will be the value of d in the following C program? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int a = 10,   b = 5 , c = 5;
int d
d = b + c == a;
printf("%d", d);
}
```

(A) 1 (B) 10 (C) 10 (D) 25

26. Which among the following is NOT a logical or relational operator? (K.B+U.B+A.B)

(A)! = (B) == (C) || (D) =

27. Relational operators cannot be used on: (K.B+U.B+A.B)

(A) Structure (B) Long (C) String (D) float

28. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int x = 2,    y = 0;
int z = (y++) ? 2 : y == 1 && x ;
printf("%d\n", z);
return 0;
}
```

(A) 0 (B) 1 (C) 2 (D) Undefined Behavior

29. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
int main( )
{
int y = 2;
int z = y + (y = 10);
printf("%d\n", z);
}
```

(A) 12 (B) 20 (C) 4 (D) Either 12 or 20

30. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int b = 5 & 4 & 6;
printf("%d", b);
}
```

(A) 3 (B) 4 (C) 5 (D) 6

31. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int b = 5 & 4 | 6;
printf("%d", b);
}
```

(A) 0 (B) 4 (C) 1 (D) 6

32. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int b = 5 + 7 * 4- 9 * (3, 2);
printf("%d", b);
}
```

(A) 6 (B) 13 (C) 15 (D) 21

33. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int h = 8;
int b = 4 * 6 + 3 * 4 < 3 ? 4 : 3;
printf("%d\n", b);
}
```

(A) 3 (B) 33 (C) 24 (D) 11

34. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
int a = 2 + 3 - 4 + 8 – 5 % 4;
printf("%d\n", a);
}
```

(A) 0 (B) 8 (C) 9 (D) 11

35. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
chr  a = '0';
chr  b = 'm';
int c = a && b | | '1';
printf("%d\n", c);
}
```

(A) 0 (B) 1 (C) a (D) m

36. What will be the output of following C code? (K.B+U.B+A.B)

```
#include <stdio.h>
void main( )
{
chr  a = 'A';
chr  b = 'B';
int c = a + b  % 3 - * 2;
printf("%d\n", c);
}
```

(A) 58 (B) 59 (C) 64 (D) 65

37. The variable on which the operations performed are called _____. (K.B+U.B)
(A) Product (B) Expression (C) Operands (D) Operators

38. _____ operators gives the remainder. (K.B+U.B)
(A) Division (B) Modulus (C) Integral (D) None of these

39. If both operands are type the remainder is _____ to give answer in integer. (K.B+U.B)
(A) Repeated (B) Truncated (C) Both A & B (D) None of these

40. The symbol for Modulus operator in C is _____. (K.B+U.B)
(A) * (B) % (C) ! (D) Wood

41. The symbol for multiplication in C is _____. (K.B+U.B)
(A) X (B) x (C) * (D) None of these

42. Relational operators are used to perform operations on _____. (K.B+U.B)
(A) Numeric (B) Characters (C) Strings (D) Both A & B

43. _____ can also be used to show result of relational expression. (K.B+U.B)
(A) printf( ) (B) scanf ( ) (C) Both A & B (D) None of these

44. The symbol used for 'Equal To' operators is _____. (K.B+U.B+A.B)
(A) = (B) == (C) Both A & B (D) None of these

45. The symbol used for 'Not Equal To' operator is _____. (K.B+U.B+A.B)
(A)!= (B) \= (C) ≠ (D) None of these

46. In AND operator's expression FALSE & FALSE results. (K.B+U.B+A.B)
(A) True (B) False (C) Can be both (D) None of these

47. FALS || FALSE always return ____ in OR operator. (K.B+U.B)
(A) False (B) True
(C) can be true or false (D) None

48. ____ is also known as invertor. (K.B)
(A) AND (B) NOT (C) OR (D) None of these

49. The operator which is applied on single operands called _____. (K.B)
(A) Unity Operator (B) Unary operator (C) Binary (D) None of these

50. ____ operator require three operands. (K.B)

(A) Unary (B) Binary (C) Ternary (D) None of these

51. ____ is an example of unary operators (K.B)

(A)! not (B) - (C) Both A & B (D) None of these

52. In AND operator FALSE && TRUE results (K.B+U.B+A.B)

(A) TRUE (B) FALSE

(C) Can be True or False (D) None of these

53. In AND operator the results in only True when both inputs are (K.B+U.B+A.B)

(A) TRUE (B) FALSE

(C) One TRUE one FALSE (D) None of these

54. In OR operator TRUE || FALSE results in _____. (K.B+U.B+A.B)

(A) TRUE (B) FALSE

(C) Can be True or False (D) None of these

55. In OR operator the answer will be FALSE only if both inputs are _____.(K.B+U.B+A.B)

(A) TRUE (B) FALSE (C) TRUE and FALSE (D) None of these

56. Predict the output !(TRUE). (K.B+U.B+A.B)

(A) TRUE (B) FALSE (C) Can be both (D) None of these

57. The variable on which the operations performed are called _____. (K.B+U.B+A.B)

(A) Product (B) Expression (C) Operands (D) Operators

58. Operators are used to perform operations on their _____. (K.B)

(A) Operators (B) Operands (C) Expression (D) None of these

59. _____ operators Gives the remainder. (K.B)

(A) Division (B) Modulus (C) Integral (D) None of these

60. If both the operands of type int, then result of division is also of type integer. Reminder is _____ to give the integer answer. (K.B)

(A) Repeated (B) Truncated (C) Both (D) None of these

61. The symbol used for multiplication in C is _____. (K.B)

(A) X (B) x (C) * (D) None of these

62. The symbol used for Modulus operator in C is _____. (K.B)

(A) * (B) % (C) ! (D) Wood

63. Relational operators are used to perform operations on _____. (K.B)

(A) Numeric (B) Characters (C) Strings (D) Both A & B

PROGRAMMING TIME (A.B)

Programming Time 2.1

Write a program that swaps the values of two integer variables.

Program:

```
void main ()
{
        int a = 2, b = 3, temp;
        temp = a;
        a = b;
        b = temp;
        printf ("Value of a after swapping: %d\n", a);
        printf ("Value of b after swapping: %d\n",  b);
}
```

Programming Time 2.2

/*This program takes as input the price of a box of chocolates and the total number of chocolates in the box. The program finds and displays the price of one chocolate.*/

```
# include <stdio.h>
Void main ( )
{
        float box_price, num_of_chocolates, unit_price;
        printf ("Please enter the price of whole box of chocolates:");
        scanf("%f", &box_price);
        printf ("Please enter the number of chocolates in the box: ");
        scanf("%f, &num_of_chocolates);
        unit_price = box_price / num_of_chocolates;
        printf("The price of a single chocolate is %f, unit_price);
}
```

Output:

```
Please enter the price of whole box of chocolates: 150
Please enter the number of chocolates in the box: 50
The price of a single chocolate is 3.000000
```

Programming Time 2.3

/* Following program takes as input the length and width of a rectangle. Program calculates and displays the area of rectangle on screen. */

```
# include<stdio.h>
void main ()
{
        float length, width, area;
        printf ("Please enter the length of rectangle:");
        scanf ("%f", &length);
        printf ("Please enter the width of rectangle: ");
        scanf ("%f", &width);
        area = length *width;
        printf ("Area of rectangle is : %f", area);
}
```

Output

```
Please enter the length of rectangle: 6.5
Please enter the length of rectangle: 3
Area of rectangle is : 19.500000
```

Programming Time 2.4

```
/* This program takes marks of two subjects from user and displays the sum of marks on console. */
#include <stdio.h>
void main ()
{
        int sum, math, science;
        printf ("Enter marks of Mathematics:");
        scanf ("%d", &math);
        printf ("Enter marks of  Science: ");
        scanf ("%d, &science);
        sum = math + science;
        printf ("Sum of marks is : %d", sum);
}
```

Output

```
Enter marks of Mathematics: 90
Enter marks of Science: 80
Sum of marks is: 170
```

Programming Time 2.5

```
/* This program finds and displays the right most digit of an input number. */
#include <stdio.h>
void  main ()
{
        int num, digit;
        printf("Enter a number:"); scanf("%d, &num);
        digit = num % 10;
        printf("Right most digit of number you entered is: %d", digit);
}
```

Output

```
Enter a number: 789
Right most digit of number you entered is: 9
```

ACTIVITY 2.1

Write down the output of following code:

```
#include <stdio.h>
void main ()
{
   printf ("I am UPPERCASE and this is lowercase");
}
```

Output: I am UPPERCASE and this is lowercase

ACTIVITY 2.2

Write a program that shows your first name in Uppercase and you last name in lower case letters on screen

Solution:

```
#include <stdio.h>
void main()
{
   printf("IMRAN khan");
}
```

Output: IMRAN khan

ACTIVITY 2.3

Write a program that takes roll number, percentage of marks and grade form user as input. Program should display the formatted output like following:

Roll No : input value
Percentage : input value%
Grade : input value

ACTIVITY 2.4

Write a program that takes as input the length of one side of a square and calculates the area of square.

Solution:

```
# include <stdio.h>
void main ()
{
   int Area, Side;
   printf ("enter the length of one side = ");
   scanf ("%d", & Side);
   Area = Side * Side; // calculate area of square
   printf ("Area of square = %d" , Area);
}
```

ACTIVITY 2. 5

Write a program that takes as input the number of balls in jar A and the number of balls in jar B. The program calculates and displays the total number of balls.

Solution:

```
# include <stdio.h>
void main ()
{
   int jar_A, jar_B, Sum;
   printf ("Enter number of balls in Jar A = ");
   scanf ("%d", &jar_A);
   printf ("Enter number of ball in Jar B = ");
   scanf ("%d", & jar_B);
   Sum = jar_A + jar_B;// calculate sum Write print (“sum of jar A and jar B= %d”,sum);
         printf("Sum of jar A and jar B = %d" , Sum);
}
```

ACTIVITY 2.6

Write a program should display the original price of shirt, discount on price and price after discount.

Solution:

```
# include <stdio.h>
void main ()
{
   int original_prince, price, discount_pre; float discounted_price;
   printf ("Enter the original price of shirt ");
   scanf ("%d", &original_prince);
   printf ("enter the discount percentage ");
   scanf ("%d", &discount_pre);
   discounted_price = original_prince * discount_pre /100;
   printf ("original price of shirt = %d discount on price %d and price after discount = %d",
original_prince, discount_pre, discounted_price);
}
```

ACTIVITY 2.7

Write a program that takes 2 digit number from user, computers the product of both digits and show the output.

Solution:

```
# include <stdio.h>
void main ()
{
   int num, rem, prod =1;
   printf ("Enter a number " );
   scanf ("%d" , &num);
   while (num!=0)
   {
      rem = num % 10; // get the right most digit
      prod = prod * rem, // calculate product of digits
      num = num /10; // remove the right most digit
   }
   printf ("%d", prod);
}
```

ACTIVITY 2.8

Write a program that takes second as input and calculates equivalent number of hours, minutes and seconds.

Solution:

```
#include <stdio.h>
void main()
{
   int sec, hour, min, s;
   printf ("input seconds: ");
   scanf ("%d", & sec);
   hour = sec/3600;
   min = (sec-/3600 * hour)160;
   s= (sec - (3600*hour) - (min *60))
   printf ("hour: min:s-;%d: %d:%d\n", hour, min, s);
}
```

ACTIVITY 2.9

Convert the following algebraic expressions into C expressions.

$x = 6y + z$ $\rightarrow x = 6 * y + z$

$x = yz^3 + 3y$ $\rightarrow x = y * z * z * + 3 * y$

$z = x + \frac{y^2}{3x}$ $\rightarrow z = x + \rightarrow y * y / 3 * x$

$z = (x-2)^2 + 3y$ $\rightarrow x * x - 2 * x * 2 + 2 * 2 + 3 * y$

$y = \left(x + \frac{3z}{2}\right) + z^3 + \frac{x}{x}$ $\rightarrow \left(x + (3 * z / 2)\right) + z * z * z + x / z$

ACTIVITY 2.10

Consider the variable x = 3, y =7. Find out the Boolean result of following expression.

(2 + 5) > y	(x + 4) == y
x! = (y - 4)	(y / 2) > = x
-1 < x	(x * 3) < = 20

ACTIVITY 2.11

Assume the following variable values x 4, y = 7 z = 8. Find out the resultant expression.

x = = 2 || y = = 8	7 > = y && z < 5
z > = 5 || x < = -3	y = = y && .(true)
x! = y || y < 5	!(z > x)

ACTIVITY 2.12

Find out the results of the following expression:

Expression	Result
10 / (5 + 3)	2
7 + 3 * (12 + 12)	79
25 %3 * 4	4
34 – 9 * 2 / (3 * 3)	32
18 / (15 – 3 * 2)	2

EXERCISE

Q1. Multiple Choice Questions.

1) 'printf' is used to print _____ type of data. **(U.B+A.B)**
A) *int* B) *float* C) *char* D) All of these

2) 'scanf' is a _____ in C programming language. **(U.B+A.B)**
A) Keyword B) Library C) Function D) None of these

3) getch() is used to take _____ as input from user. **(U.B+A.B)**
A) *int* B) *float* C) *char* D) All of these

4) Let the following part of code, what will be the value of variable '**a**' after execution: **(U.B+A.B)**
int a = 4;
float b = 2.2;
a = a * b;
A) 8.8 B) 8 C) 8.0 D) 8.2

5) Which one of the following is a valid line of code. **(U.B+A.B)**
A) int = 20; B) grade = 'A';
C) line = this is a line; D) none of these

6) Which operator has highest precedence among the following: **(K.B+U.B)**
A) / B) = C) > D) !

7) Which one of the following is not a type of operator: **(K.B+U.B)**
A) Arithmetic operator B) Relational operator
C) Check operator D) Logical operator

8) The operator % is used to calculate _____. **(K.B+U.B)**
A) Percentage B) Remainder C) Factorial D) Square

9) Which one of the following is a valid character: **(K.B+U.B)**
A) here B) "a" C) '9' D) None of these

10) What is true about C language: **(K.B+U.B)**
A) C is not a case sensitive language
B) Keywords can be used as variable names
C) All logical operators are binary operators
D) None of them

ANSWER KEY

1	2	3	4	5	6	7	8	9	10
D	C	C	B	B	D	B	B	C	D

Q2. True or False

1) Maximum value that can be stored by an integer is 32000. **(K.B) T**/F
2) Format specifiers begin with a '%' sign. **(K.B) T**/F
3) Precedence of division operator is greater than multiplication operator. **(K.B)** T/**F**
4) *getch* is used to take all types of data input from user. **(K.B+A.B)** T/**F**
5) scanf is used for output operations. **(K.B+A.B) T**/F

Q3. Define the following. **(K.B)**

1) **Statement Terminator**

Ans: **STATEMENT TERMINATOR**

A statement terminator is an identifier for compiler which identifies end of a line. In C language semicolon (;) is used as statement terminator. If we do not end each statement with a statement terminator it result into error. **Error:** **Missing statement**

2) Format Specifier

Ans:

FORMAT SPECIFIER

A format specifier is computer code that tells about the data type, field width and the format according to which a value is to be printed or read from an input device

A list of commonly used format specifiers is given below:

- %d decimal integer
- %i integer
- %ld long decimal integer
- %f floating-point (decimal notation)
- %g floating-point (exponential notation)
- %e floating-point (%f of %g, whichever is shorter)
- %c single character
- %s string

3) Escape Sequence

Ans:

ESCAPE SEQUENCE

The special characters used in C language to control printing on the output device are called escape sequences. Escape sequences are used in print function inside the "and." they force printf() to change its normal behavior of showing output. These characters are not printed. These are used inside the control string.

4) *scanf*()

Ans:

SCANF () FUNCTION

scanf ()is a built-in function in C language that takes input from user into the variables. We specify the expected input data type in scanf function with the help of format specifier. If user enters integer data type, format specifier mentioned in scanf must be %d or %i.

Syntax:

scanf ("control string", list of variables);

5) Modulus Operator

Ans:

MODULUS OPERATOR

Modulus operator (%) performs division of left operand by the right operand and returns the remainder value after division. Modulus operator works on integer data types.

- **int REM = 14 % 3;**

As, when we divide 14 by 3, we get a remainder of 2, so the value stored in variable REM is 2

Q4. Briefly answer the following questions. (U.3)

1) What is the difference between *scanf* and *getch?*

Ans:

scanf( ) function	getch( ) FUNCTIONS
scanf is a built-in function in C language that takes input from user into the variables. We specify the expected input data type in *scanf* function with the help of format specifier. If user enters integer data type, format specifier mentioned in scanf must be %d or %i.	*getch*() function is used to read a character from user. The character entered by user does not at displayed on screen. This function is generally used to hold the execution of program because the program does not continue further until the user types a key.
Syntax: *scanf*("control string", list of variables seperrated with commas);	**Syntax:** getch();

2) Which function of C language is used to display output on screen?

Ans: **PRINTF ( ) FUNCTION**

printf is built-in function in C programming language to show output on screen. Its name comes from "Print formatted" that is used to print the formatted output on screen. All data types can be displayed with *printf* function.

Syntax:

printf ("control string", list of arguments);

3) Why format specifiers are important to be specified in I/O operations?

Ans: **FORMAT SPECIFIER**

A format specifier is computer code that tells about the data type, field width and the format according to which a value is to be printed or read from an input device

Example:

- %d decimal integer
- %i integer
- %ld long decimal integer
- %f floating-point
- %s string

4) What are escape sequences? Why do we need them?

Ans: **ESCAPE SEQUENCE**

The special characters used in C language to control printing on the output device are called escape sequences. Escape sequences are used in print function inside the "and." they force printf() to change its normal behavior of showing output. These characters are not printed. These are used inside the control string.

5) Which operators are used for arithmetic operations?

Ans: **ARITHMETIC OPERATORS**

Arithmetic operators are used to perform arithmetic operations that include addition, subtraction, multiplication, division and also to find the remainder obtained when an integer is divided by another integer.

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder (Modulus) Operator

6) What are relational operators? Describe with an example.

Ans:

Relational Operator
• Relational operators are used to compare two values of the same types.
• After evaluation of a relational expression, the result produced is either TRUE or FALSE
• Relational operator include ==, !=, <, >, <= and >=.

Examples:

Relation Expression	Explanation	Result
5 = =5	5 is equal to 5 ?	TRUE
5 ! = 7	5 is not equal to 7?	TRUE
5 > = 5	5 is greater than or equal to 5?	TRUE
5 < = 4	5 is less than or equal to 4?	FALSE

7) What are logical operators? Describe with an example.

Ans:

LOGICAL OPERATORS

Logical operators are used for building compound conditions. We have seen before that a single condition is built using a relational operator in an expression. If we need to build more than one condition for some action to take place in programming, then we have to form compound condition.

Types of Logical Operators:

There are three types of logical operators. These are:

Operator	Definition
& &	AND
||	OR
!	NOT

AND operator (&&):

AND operator && takes two Boolean expressions as operands and produces the result TRUE if both of its operands are TRUE. It returns FALSE if any of the operands is FALSE. Table shows the truth table for AND operator.

Expression	Result
FALSE && FALSE	FALSE
FALSE && TRUE	FALSE
TRUE && FALSE	FALSE
TRUE && TRUE	TRUE

OR Operator (||):

OR operator accepts Boolean expression and returns true if at least one of the operands it true. Table shows that truth table for OR operator.

Expression	Result
FALSE || FALSE	FALSE
FALSE || TRUE	TRUE
TRUE || FALSE	TRUE
TRUE || TRUE	TRUE

NOT Operator (!):

NOT operator negates or reverses the value of Boolean expression. It makes it TRUE, if it is FALSE and FALSE if it is TRUE. Table presents the truth table for not operator.

Expression	Result
!(TRUE)	FALSE
!(FALSE)	TRUE

Examples of Logical Operators:

Logical Expression	Explanation	Result
3 < 4 && 7 > 8	3 is less than 4 AND is greater than 8?	FALSE
3 = = 4 || 3 > 1	3 is equal to 4 OR 3 is greater than 1?	TRUE
! (4 > 2 || 2 = = 2)	NOT (4 is greater than 2 OR 2 is equal to 2)?	FALSE

Example:

The expression: **! (a<b) w**ill be true if **a** is not less than **b**. In other words, the condition will be TRUE if **a** is greater than or equal to **b**. The same condition can also be written as **(a>=b)** which is easy to understand.

8) What is the difference between unary operators and binary operators?

Ans:

Unary Operators	Binary Operators
Unary operators are applied over one operand only	Binary operators require two operands to perform the operation
Example Logical not (!) operator has only one operand. Sign operator (-) is another example of a unary operator e.g. -5.	**Example** All the arithmetic operators, and relational operators are binary operators. The logical operator && and || are also binary operators.

9) What are the differences between '=' operators and '==' operators?

Assignment Operator (=)	Equal to Operators (= =)
Assignment operator is used to assign a value to a variable, or assign a value of variable to another variable.	In C language, = = operator is used to check for equality of two expressions
Equal sign (=) is used as assignment operator in C.	Double Equal sign (= =) is used as equal to operator
Consider the following example: int sum = 5;	Consider the following example: 3+2 = = 5;

10) What is meant by precedence of operators? Which operator has the highest precedence in C language?

Ans: If there are multiple operators in an expression, the question arises that with operator is evaluated first. To solve this issue, a precedence has been given to each operator (Table given below). An operator with higher precedence is evaluated before the operator with lower precedence. In case of equal precedence, the operator at left side is evaluated before the operator at right side.

Example:

Result = 18 / 2 * 3 + 7% 3 + (5 * 4);

Operator	Precedence
()	1
!	2
* , / , %	3
+ , -	4
> , < , > = , < =	5
= = , ! =	6
&&	7
||	8
=	9

Q5. Write down output of the following code segments.

	Code	Output

a.	# include<stdio.h>
	void main ()
	{
	int x = 2, y = 3, z = 6;
	int ans1, ans2, ans3;
	ans1= x / z * y;
	ans2 = y + z / y * 2;
	ans3 = z / x +x * y;
	printf("%d %d %d", ans1, ans2, ans3);
	}
		0 7 9

b.	# include<stdio.h>
	void main ()
	{
	printf (“nn \n\n nnn \nn\nt\t”);
	printf (“nn /n/n nn/n/n”);
	}
		**nn
		nnn
		n t
		nn/n/n nn/n**

c.	# include<stdio.h>
	void main ()
	{
	int a = 4, b;
	float c = 2.3;
	b=c*a;
	printf(“%d”,b);
	}
		9

d.	# include<stdio.h>
	void main ()
	{
	int a = 4 * 3 / (5+1) + 7 % 4;
	printf(“%d”,a);
	}
		5

e.	# include<stdio.h>
	
	void main ()
	{
	printf(“%d”,(((5 > 3) && (4 > 6)) || (7 > 3)));
	}
		1

Q6. Identify errors in the following code segments.

	Code	Identified Error/s	Correct Syntax

a.	# include<stdio.h>
	void main ()
	{
	int a , b = 13;
	b = a% 2;
	printf(“Value of b is : %d , b);
	}
		Trace error in line #3 (body of function) “closing” inverted commas missing	printf(“Value of b is : %d” , b);

b.	# include<stdio.h>
	void main ()
	{
	int a , b , c;
	printf(“Enter First Number: ”);
	scanf(“%d” , &a);
	printf(“Enter Second Number: ”);
	scanf(“%d”, &b);
	a + b = c;
	]
		Trace error in closing brace of body of function	Use curly braces **}** instead of that **]**

c.	# include<stdio.h>
	main ()
	{
	int num;
	printf(Enter number: ”
	);
	scanf(“%d, & num);
	};
		Trace error at the end of body of function (curly braces) statement terminator used};	Statement terminator ; can’t be used at the end of body of function

d.	#include<stdio.h>
	int main ()
	{
	float f;
	printf [“Enter First Number: ”];
	scanf(“%c”, &f);
	}
		Trace error in scanf (“%c”, &f);	Float data type always written with the format specifier %f

PROGRAMMING EXERCISES (A.B)

Exercise 1

The criteria for calculation of wages in a company is given below.

Basic Salary	**= Pay Rate Per Hour**	**x**	**Working Hours Of Employee**
Overtime Salary	**= Overtime Pay Rate**	**x**	**Overtime Hours Of Employee**
Total Salary	**= Basic Salary**	**+**	**Overtime Salary**

Write a program that should take working hours and overtime hours of employee as input. The program should calculate and display the total salary of employee.

Solution:

```
# include <stdio.h>
        void main ( )
{
        float working_hours,overtime_hours;
        float pay_per_hour,overtime_per_hour;
        float basic_salary,overtime_salary,Total_salary;
        printf("Enter working hours of employee\n");
        scanf("%f",&working_hours);
        printf("Enter pay rate per hour\n");
        scanf("%f",&pay_per_hour);
        printf("Enter overtime hours of employee");
        scanf("%f",&overtime_hours);
        printf("Enter overtime rate per hour of employee");
        scanf("%f",&overtime_per_hour);
        basic_salary=pay_per_hour*working_hours;
        overtime_salary= overtime_per_hour*overtime_hours;
        Total_salary=basic_salary+overtime_salary;
        printf("Total Salary=%f",Total_salary);
}
```

Exercise 2

Write a program that takes Celsius temperature as input, converts the temperature into Fahrenheit and shows the output. Formula for conversion of temperature from Celsius to Fahrenheit is: $F = \frac{9}{5}C + 32$

Solution:

```
# include <stdio.h>
        void main ( )
{
        float C,F;
        printf(" enter tempratue in celsius\n");
        scanf("%f",&C);
        F = 9/5*C+32;
        printf("Temprature in Farenheit=%f",F);
}
```

Exercise 3

Write a program that displays the following output using single *printf* statement:

*** * * ***

1 2 3 4

Solution:

```
# include <stdio.h>
void main ( )
{
printf("*\t* t*\t*\n1\t2\t3 t4");
}
```

Exercise 4

Write a program that displays the following output using single printf statement:

I am a Boy

I live in Pakistan

I am a proud Pakistani

Solution:

```
# include <stdio.h>
void main ( )
{
printf ("I am a boy\nI live in Pakistan\nI am a proud Pakistani");
}
```

Exercise 5

A clothing brand offer 15% discount on each item. A lady buys 5 shirts from this brand. Write a program that calculate total price after discount and amount of discount availed by the lady. Original prices of the shirts are:

Shirt1 = 423

Shirt2 = 320

Shirt3 = 270

Shirt4 = 680

Shirt5 = 520

Note: Use 5 variables to store the prices of shirts.

Solution:

```
# include <stdio.h>
void  main( )
{
int shirt1=423, shirt2=320, shirt3=270, shirt4=680, shirt5=520;
float Total_Price, Total_Discount, Final_Price;
printf("First Shirt price= %d \t\tDiscount is %.2f \tPayable price= %.2f\n",shirt1, (shirt1*0.15),
(shirt1-(shirt1*0.15)));
printf("Second Shirt price= %d \tDiscount is %.2f \tPayable price= %.2f\n",shirt2,
(shirt2*0.15), (shirt2-(shirt2*0.15)));
printf("Third Shirt price= %d \t\tDiscount is %.2f \tPayable price= %.2f\n",shirt3,
```

```
(shirt3*0.15), (shirt3-(shirt3*0.15)));
printf("Fourth Shirt price= %d \tDiscount is %.2f \tPayable price= %.2f\n",shirt4,
(shirt4*0.15), (shirt4-(shirt4*0.15)));
printf("Fifth Shirt price= %d \t\tDiscount is %.2f \tPayable price= %.2f\n",shirt5,
(shirt5*0.15), (shirt5-(shirt5*0.15)));
printf("\n\n---------------------------------------------------------------------------------
---------\n");
Total_Price= shirt1 + shirt2 + shirt3 + shirt4 + shirt5;
Total_Discount=(shirt1*0.15)+(shirt2*0.15)+(shirt3*0.15)+(shirt4*0.15)+(shirt5*0.15);
Final_Price = (shirt1-(shirt1*0.15))+(shirt2-(shirt2*0.15))+(shirt3-(shirt3*0.15))+(shirt4-
(shirt4*0.15))+(shirt5-(shirt5*0.15));
printf("Total price: %.2f \tTotal Discount is %.2f \tFinal price: %.2f\n",Total_Price,
Total_Discount, Final_Price);
}
```

Exercise 6

Write a program that swaps the values of two integer variables without help of any third variable.

Solution:

```
# include <stdio.h>
void main( )
{
int a, b;
printf("Enter a\n");
scanf("%d", &a);
printf("Enter b\n");
scanf("%d", &b);
a = a - b;
b = a + b;
a = b - a;
printf("After swapping, a = %.2d\n", a);
printf("After swapping, b = %.2d", b);
}
```

Exercise 7

Write a program that takes a 5-digit number as input, calculates and displays the sum of first and last digit of number.

Solution:

```
# include <stdio.h>
void main( )
{
int number;
printf("Enter a number with length 5 digits=\n");
scanf("%d",& number);
printf("The sum of first and 5th digit is=%d", (number/10000)+(number%10));
}
```

Exercise 8

Write a program that takes monthly income and monthly expenses of the user like electricity bill, gas bill, food expense. Program should calculate the following:

- **Total monthly expenses**
- **Total yearly expenses**
- **Monthly savings**
- **Yearly saving**
- **Average saving per month**
- **Average expense per month**

Solution:

```
#include <stdio.h>
void main()
{
   float monthly_income, electricity_bill, gas_bill, food_expense, monthly_Expenses,
monthly_savings;
   printf("Please enter your Monthly Income=");
   scanf("%f",&monthly_income);
   printf("Please enter you Monthly Expenses=");
   printf("\n\n\tYour Electricity Bill=");
   scanf("%f",&electricity_bill);
   printf("\tYour Gas Bill=");
   scanf("%f",&gas_bill);
   printf("\tYour Food Expense=");
   scanf("%f",&food_expense);
   monthly_Expenses=electricity_bill+gas_bill+food_expense;
   printf("\n\n\tTotal Monthly Expenses are=%f", monthly_Expenses);
   printf("\n\tTotal Yearly Expenses are=%f", (monthly_Expenses)*12);
   monthly_savings =monthly_income-monthly_Expenses;
   printf("\n\n\tYour Monthly savings is=%f", monthly_savings);
   printf("\n\tYour Monthly savings is=%f", monthly_savings*12);
   printf("\n\n\tAverage saving per month=%f", monthly_savings);
   printf("\n\tAverage expense per month=%f", monthly_Expenses);
}
```

Exercise 9

Write a program that takes a character and number of steps as input from user. Program should then jump number of steps from the character.

Sample output:

Enter character: a

Enter steps: 2

New character: c

Solution:

```
# include <stdio.h>
void main( )
{
char c;
int steps,x;
printf("Enter character=");
scanf("%c",&c);
printf("Enter Steps=");
scanf("%d",&steps);
 x = c + steps;
printf("New Character: %c",x);
}
```

Exercise 10

Write a program that takes radius of a circle as input. The program should calculate and display the area of circle.

Solution:

```
# include <stdio.h>
void  main( )
{
float r, Area;
printf(“Enter the radius of circle”);
scanf(“%f”,&r);
Area = 3.14*r*r;
printf(“The area of the circle is %f\n”,Area);
}
```

ANSWER KEY

2.1 INPUT / OUTPUT (I/O) FUNCTION

2.1.1 PRINTF ()

2.1.2 FORMAT SPECIFIERS

2.1.3 SCANF ()

2.1.4 GETCH ()

2.1.5 STATEMENT TERMINATOR

2.1.6 ESCAPE SEQUENCE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	D	D	A	A	C	B	B	D	C	A	C	D	A	B
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	C	C	B	C	B	C	A	D	A	B	A	A	C	A
31	32	33	34	35	36	37	38							
B	B	C	B	B	C	B	A							

2.2 OPERATORS OF C LANGUAGE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	C	A	A	A	C	A	B	C	C	C	B	C	B	A
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	C	D	A	A	A	A	C	D	A	D	A	C	B	B
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
D	C	A	B	B	B	C	B	B	B	C	D	A	B	A
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
B	A	B	B	C	A	B	A	A	B	B	C	C	B	B
61	62	63												
C	B	D												