

Q. No. 2 Part (i)

Program:

Number: Positive or negative.

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

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("\nEnter a number:");
```

```
    scanf("%d", &n);
```

```
    if (n > 0)
```

```
        printf("\nIt is a positive number!");
```

```
    else if (n < 0)
```

```
        printf("\nIt is a negative number!");
```

```
    else
```

```
        printf("\nIt is zero!");
```

```
}
```

Q. No. 2 Part (ii)

Differentiation:

Break	Continue
1 Break statement is used to exit from a loop as soon as certain condition appears.	• Continue statement is used to shift the control to the beginning of loop as soon as condition appears.
2 Break is used with loops as well as with switch statement.	• Continue is only used with loops.
3 Syntax: It has following syntax: <code>break;</code>	• Syntax: It has following syntax: <code>continue;</code>

Q. No. 2 Part (iii)

Escape Sequences:

Escape sequences are basically used to control printing on output device. These are a combination of backslash (\) and control character.

⇒ Uses of Escape Sequences:

<u>Escape Sequence</u>	<u>Use</u>
\b	Used to move backward by one tab position
\n	Used to move on next/new line
\"	Used to produce double quotes.
\t	Used to move in tabular position.

Q. No. 2 Part (iv)

Evaluation:

$$(a) 2 * (3+5) - 4/2 + 6$$

$$2 * 8 - 4/2 + 6$$

$$16 - 4/2 + 6$$

$$16 - 2 + 6$$

$$14 + 6$$

$$20$$

$$(b) 7 + 4 * 3 - 2^2$$

$$7 + 12 - 2^2$$

$$7 + 12 - 4$$

$$19 - 4$$

$$5$$

Q. No. 2 Part (v)

Algorithm: ^{Area of circle}

❖ Planning the solution:

Input : Radius of circle

Required Output: Area of circle

Processing : Take square of radius and multiply with π ($\frac{22}{7}$)

❖ Algorithm:

STEP 1: START

Let radius = $r = 5\text{cm}$

STEP 2: Calculate Area:

$$\text{Area} = \pi * r * r$$

STEP 3: OUTPUT Area.

STEP 4: STOP

Q. No. 2 Part (vi)

Program:

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

```
int main()
```

```
{ int a, b, c, largest;
```

```
printf ("\n Enter three numbers:");
```

```
scanf ("%d %d %d", &a, &b, &c);
```

```
if (a > b)
```

```
{ if (a > c)
```

```
printf ("%d\n largest = %d", a);
```

```
}
```

```
else if ((b > a) && (b > c))
```

```
printf ("%d\n largest = %d", b);
```

```
else
```

```
printf ("%d\n largest = %d", c); }
```

Q. No. 2 Part (vii)

Main Components of Loop:

Loop is a structure used in programming for repeating a set of statements until a particular condition is met. Loops are very often used in programming to make programming easier and efficient. Following are two main components of loop:

i- Block of statements which form body of loop and ~~is~~ executed repeatedly until a particular condition is met.

ii. Loops terminate based on test conditions.

Q. No. 2 Part (viii)

Conversion: ^{to} while loop

```
#include <stdio.h>
int main()
{
    int i = 1;
    while (i <= 5)
    {
        printf ("%d", i);
        i++;
    }
    return 0;
}
```

Q. No. 2 Part (ix)

Code for Given Output:

```
#include <stdio.h>
int main ()
{
    int j, k;
    for (j = 1; j <= 3; j++)
    {
        for (k = 1; k <= j; k++)
            printf ("%d ", k);
        printf ("\n");
    }
}
```

Q. No. 2 Part (x)

Truth Table:

As indicated by given logic circuit the function should be $X = AB + C$.

Following is the truth table for $X = AB + C$:

A	B	C	AB	F / $AB + C$
1	1	1	1	1
1	1	0	1	1
1	0	1	0	1
1	0	0	0	0
0	1	1	0	1
0	1	0	0	0
0	0	1	0	1
0	0	0	0	0

Q. No. 2 Part (xi)

Code: First ten natural numbers.

```
#include <stdio.h>
int main ()
{
    int j;
    for (j=1 ; j<=10 ; j++)
        printf ("\n j = %d", j);
}
```

Use for rough work

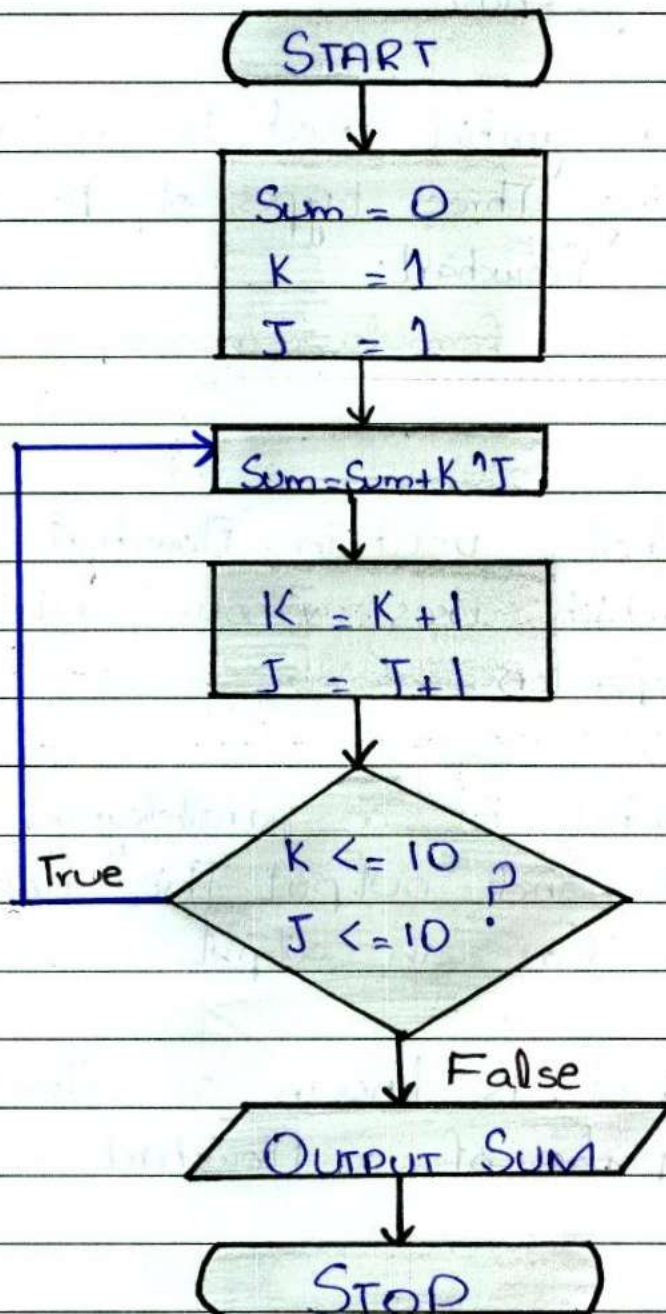
Flowchart:

Flowchart is the diagrammatic representation of algorithm. It describes and illustrates sequence of operations to solve a specific problem. It uses symbols to represent various functions.

Series:

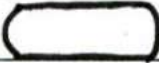
$$\text{Sum} = 1^1 + 2^2 + 3^3 + 4^4 + \dots + 10^{10}$$

Flowchart for this series is:



⇒ Symbols used:

Above flowchart used following symbols:

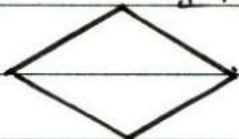
1:  :-

This symbol is used at the start and end of flowchart ; it basically indicates start and stop. A flowchart can have one start but it may have many ends.

2:  :-

The second symbol used is rectangle which uses processings. Three types of processes are used in this flowchart:

Initialization, formula, increment.

3:  :-

Third symbol, used in flowchart is diamond symbol which uses various decisions. It includes relational operators $<$, $>$, $<=$, $>=$, $=$, $!$ etc.

4:  :-

Next symbol is a parallelogram which represents input and output. This flowchart has no input but has an output.

5: $\rightarrow$:-

Last symbol is known as flowline which connects parts of a flowchart.

Switch Statement:

Switch statement is just like if-else-if statement used in HLL's. It checks only for equality. It does not check for a range nor uses relational operators. It is multi-cases statement. It makes decision based on several cases.

❖ Syntax:

```
switch (expression)
{
    case const - 1:
        statements;
        break;
    case const - 2:
        statements;
        break;
    :
    case const - n:
        statements;
        break;
    default:
        statements;
}
```

❖ Example:

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter a character:");
    scanf("%c", &ch);
```

```
switch (ch)
{
    case 'a':
    case 'e':
    case 'i':
    case 'o':
    case 'u':
        printf ("It is a vowel!");
        break;
    default:
        printf ("It is a consonant!");
}
```

❖ Working:

An expression or variable is used with switch which after evaluating moves to inner side and is matched with results given with keywords case.

If any case is matched, it executes statements under that specific case, if none of the case is matched it executes statements under default. Use of default is optional.

❖ Significance of break statement:

Break statement basically exits case block. When break is used, it skips the statements and ends case block and shifts control back to the beginning of switch statement. Break statement is significant in separating case blocks from one another.

Program:

1) For Loop:

For is a looping structure which is used in a situation when number of iterations are known in advance. It continues ^{until} a fixed number of times therefore it is also known as "counter loop."

2) Syntax:

For loop has the following syntax:

```
for (initialization ; test condition ; Increment/decrement)
{
    Body of loop;
}
```

3) Program:

Following program demonstrates use of for loop to print multiplication table:

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    int n, j;
    printf ("Enter any number:");
    scanf ("%d", &n);
    for (j = 1 ; j <= 10 ; j++)
        printf ("%2d X %2d = %3d", n, j, n*j);
}
```

⇒ Output:

Output of the program written above is:

Enter any number : 5

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

⇒ Explanation:

The above output shows the multiplication table of a number which is entered by the user. The program of this output uses "for" as a looping statement, where j is used as a loop variable. The last printf statement used $\%2d$, $\%2d$ and $\%3d$ as format specifiers for n , j and $n*j$ respectively. The 2, 2 and 3 with d means to print values of n , j and $n*j$ with field width of 2, 2 and 3 respectively.

HTML Code:

The table given as output of required HTML code uses 5 rows and 3 columns with specific headings. So, to write html code of this output table tag with its attributes will be used.

⇒ Code:

```
<html>
```

```
<head>
```

```
<title> Creating Table </title>
```

```
</head>
```

```
<body>
```

```
<table border="1">
```

```
<tr>
```

```
<th> Sr.No. </th>
```

```
<th> Name </th>
```

```
<th> Total Marks </th>
```

```
</tr>
```

```
<tr> 1
```

```
<td> Drake </td>
```

```
<td> Drake </td>
```

```
<td> 28 </td>
```

```
</tr>
```

```
<tr>
```

```
<td> 2 </td>
```

```
<td> Sillick </td>
```

```
<td> 26 </td>
```

```
</tr>
```

```
<tr>
```

```
<td> 3 </td>
<td> Tim </td>
<td> 27 </td>
</tr>
<tr>
<td> 4 </td>
<td> Nilly </td>
<td> 30 </td>
</tr>
</table>
</body>
</html>
```

⇒ Output:

The html code used above displays following table as its output:

Sr. No.	Name	Total marks
1	Drake	28
2	Slick	26
3	Tim	27
4	Nilly	30

The html code of this output basically used table tag, row tag, heading tag, column tag and one border attribute.

