

HAMMAD PER

ID# 6961

VISUAL PROGRAMMING

①

Q3(A)

Ans While LOOP:-

The test condition is given in the beginning of the loop and all statements are executed till the given boolean condition satisfies when the condition becomes false the control will be out from the while loop.

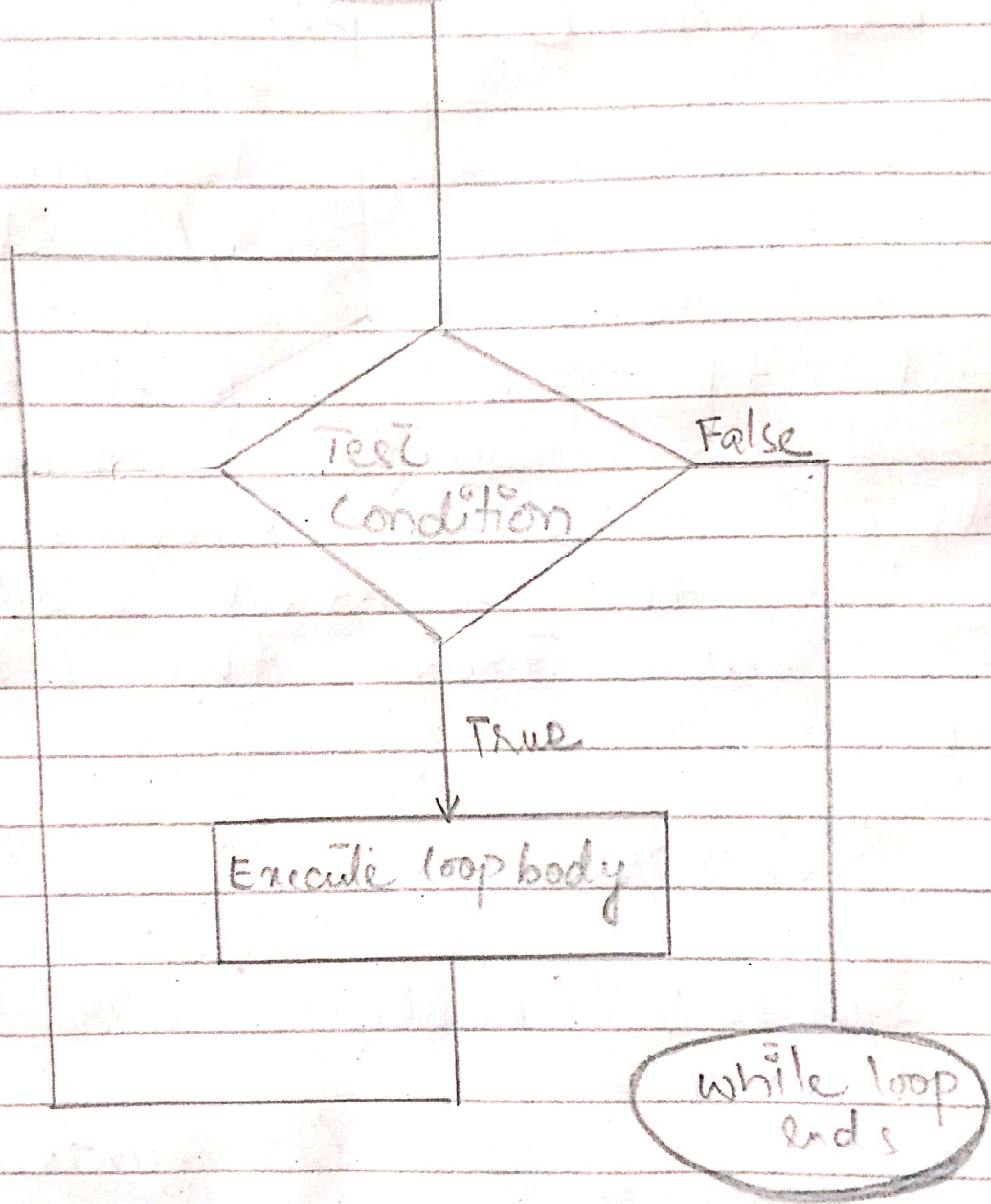
SYNTAX:-

```
while (boolean condition)
{
    loop statements...
}
```

FLOW CHART:-

3

White loop  
start

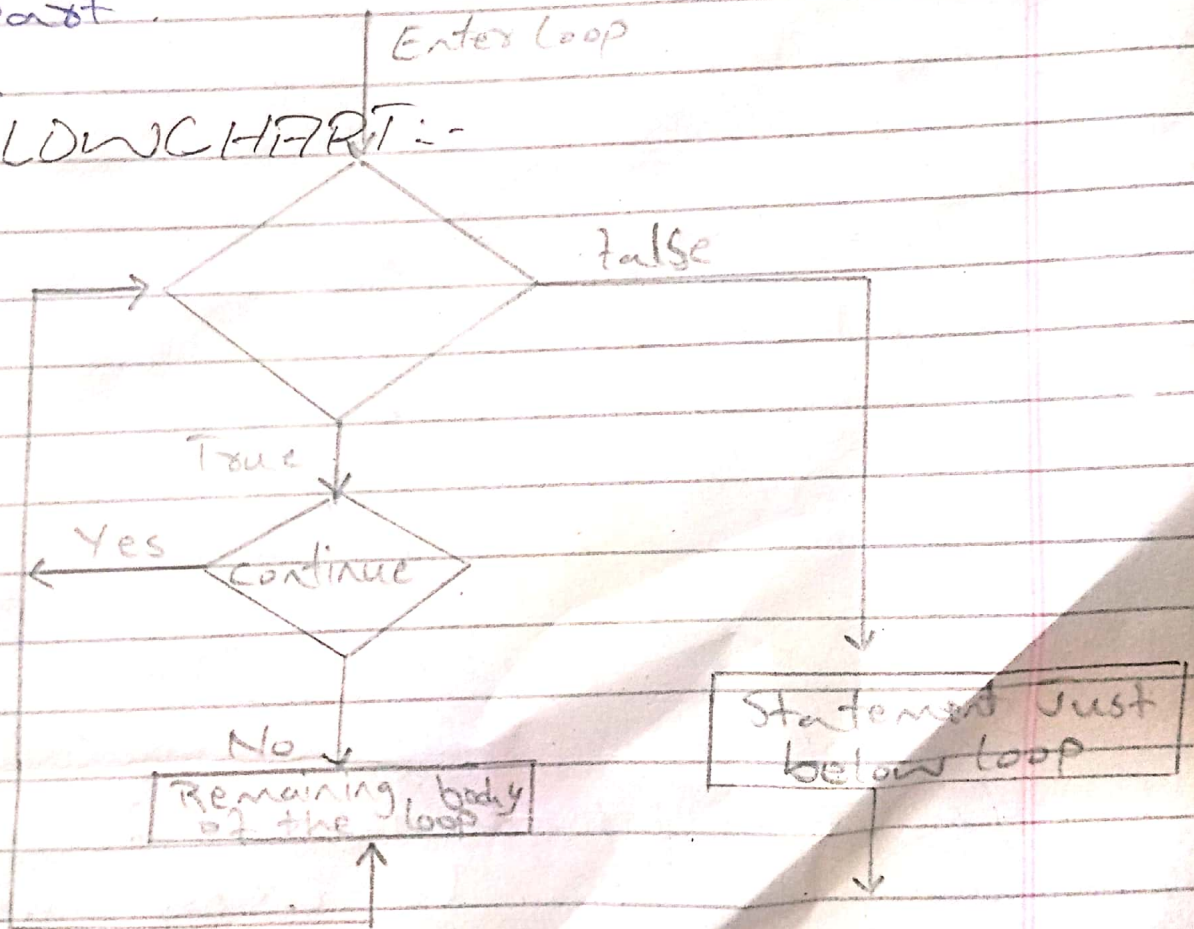


3

## CONTINUOUS STATEMENT:-

Continue Statement is used to skip over the execution part of loop on a certain condition and move the flow to next updation part.

### FLOWCHART:-



Example:-

1) C# program to demonstrate continue using system;

(4)

```
do {  
    statements.  
} while (condition);
```

FLOW CHART:

Example:-

// C++ program to illustrate  
do-while loop using system;

```
{  
    class dowhileloopDemo
```

```
{  
    public static void Main()
```

(5)

For (loop variable initialization;  
testing condition;

increment / decrement

{

// statement to be executed

}

FLOWCHART:-

⑥

Q3(b) FOR LOOP:-

This is most commonly used loop in C language. The syntax and flow of this loop is simple and easy to learn. However there are few cases when you may prefer any other loop, instead of this.

While Loop:-

This is used when you need to execute a block of statements repeatedly until a given condition is met. Read this tutorial to understand the flow of this loop.

Example:-

DO-While loop:-

```
#include <studio.h>
int main()
{
    int count = 1;
    while (count <= 4)
        printf("%d", count);
}
```

(7)

```
count++;  
}  
return 0;  
}
```

Do-while loop:-

It is similar to the while loop, the only difference is that it evaluates the test condition after execution of the statements enclosed in the loop body.

Example:-

```
#include <studio.h>  
int main()  
{  
    int j=0;  
    do  
    {  
        printf("value of  
variable j is : %d\n", j);  
        j++;  
    } while (j=3);  
    return 0;  
}
```



(9)

## Break Statement:-

It is used with various loops (for while and do-while) and switch case statements. When break statement is ~~encountered~~<sup>reached</sup> inside a loop, the control comes out of the switch case and continue execution with the statement following switch case body.

```
#include <studio.h>
int main() {
    (Local variable definition)
    int a = 10;
    (while loop execution)
    while (a < 20) {
```

```
    printf("value of a: %d\n",
a);
```

```
        a++;
        if (a > 15) {
            (terminate the
loop using break statement)
            break;
        }
    }
    return 0;
}
```

(10)

2  
Q(b)

```
#include <stdio.h>
void main()
{
    int tmp;
    printf("input days temperature:");
    scanf("%d", &tmp);
    if (tmp < 0)
        printf("freezing
weather.\n");
    else if (tmp < 10)
        printf("very cold
weather.\n");
    else if (tmp < 20)
        printf("cold
weather.\n");
    else if (tmp < 30)
        printf("Normal in temp.\n");
    else
        if (tmp < 40)
            printf("its Hot.\n");
        else
            printf("its Hot.\n");
        else
            printf("its very hot.\n");
}
```

Q4:-

Ans:-

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax:-

The syntax of a for loop in C++ is -

```
for (init; condition;
     increment) {
    statements (s);
}
```

Here is the flow of control in a for loop -

- The init step is executed first, and only once. This step allows you to declare and variables. You are not to put a statement here as long as a semicolon appears.
- Next, the condition is evaluated. If it is false, the body of the loop does not execute and flow of control jumps to the next statement.

- Just after the for loop. After the body of the for loop executes the flow of control jumps back up to the increment statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.
- The condition is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then increment step, and then again testing for condition). After the condition becomes false the for loop terminates. Flow Diagram.

Example:-

```

Live Demo
Using System;
namespace Loops {
    class Program {
        static void Main(string[]
args) {

```

(13)

```
for loop execution /  
for int a=10; a < 20; a  
= a+1) {  
    Console.WriteLine ("value  
of a {0}", a);  
}  
    Console.ReadLine();  
}
```

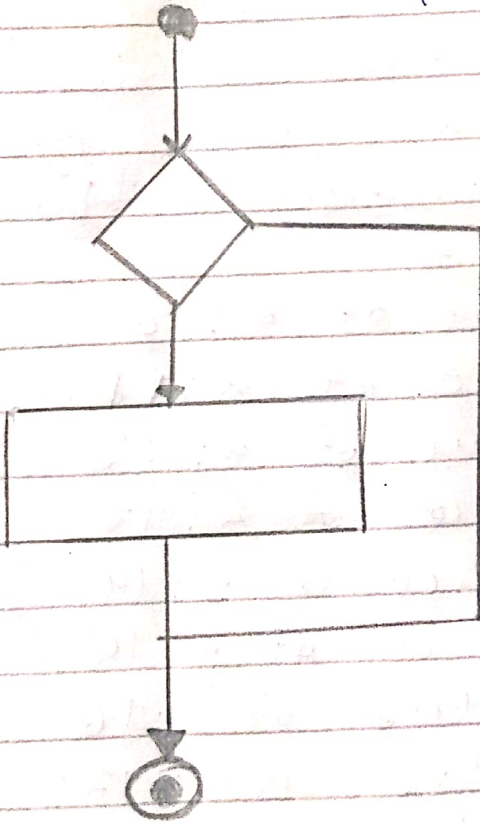
When the above code is compiled and executed, it produces the following result-

value of a: 10  
value of a: 11  
value of a: 12  
value of a: 13  
value of a: 14  
value of a: 15  
value of a: 16  
value of a: 17  
value of a: 18  
value of a: 19

(14)

Q. What is decision making in C # explain with the help of flow charts?

Ans Decision making structures requires the programmer to specify one or more conditions to be evaluated or tested by the program



There are three types of decision making statements.

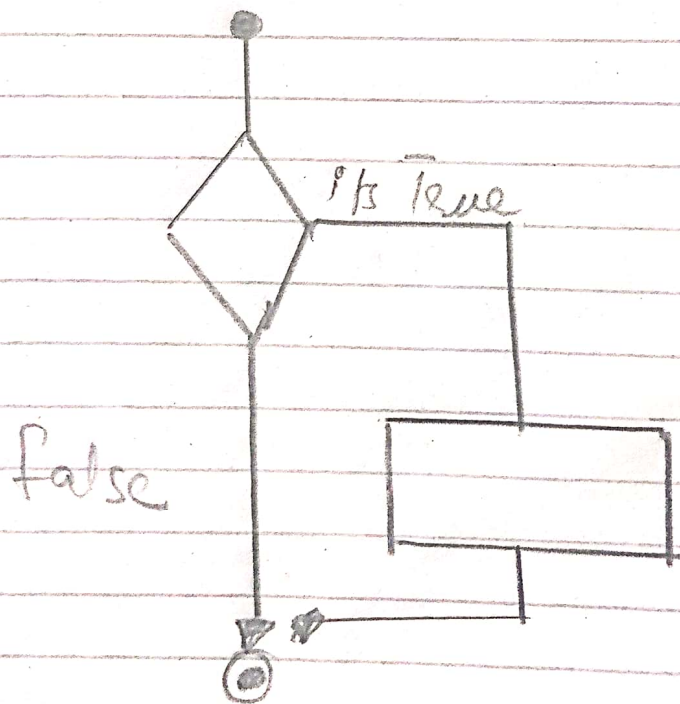
- 1- if statement
- 2- else statement
- 3- else if statement.

(15)

If statements

If the expression evaluates to True, then the block of code inside the if statement be executed.

If expression evaluates to false, then the first set of code after the end of the statement (after the closing curly brace) is executed.



If else statement