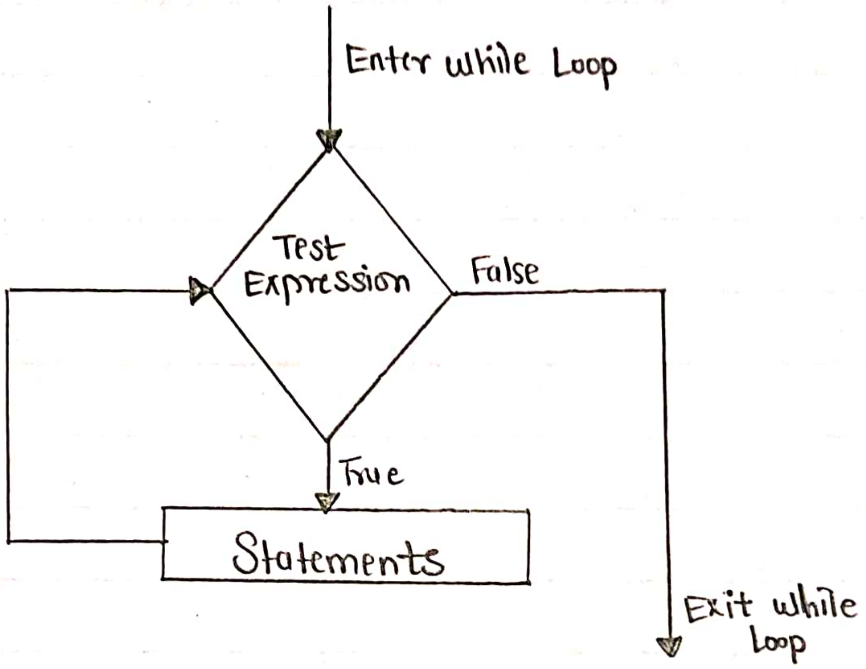
Practical No. 2

Date:

Aim:	Write a program in python to demonstrate execution of Loops.
Flowchart:-	While Loop's general flow chart:



Practical No. 1

Date :

Aim : Write a program in python to demonstrate execution of Loops.

Theory : There are two primitive Loops in python :

- While Loop.
- For Loop.

— While Loop : This Loop comes under the category of indefinite iteration i.e the number of times the loop is executed isn't specified explicitly in advances. While Loop is used to executed a block of statements repeatedly until a given condition is satisfied. When the condition becomes false, the Line immediately after the Loop in the program is executed.

Syntax :

```
while expression :
    statements;
```

ARISE & SHINE

Algorithm : For the code : Python program to print Numbers from 1-10 using while Loop.

Step 1 : Start

Step 2 : Check if value of var "i" is less than or equal to 10, while i is less than equal to 10 go to step 3. otherwise go to step 5.

Step 3: print the value of "i".
Step 4: increment "i" by 1.
Step 5: STOP.

Code 1: Python program to print Numbers from 1 to 10 using While Loop.

```
i=1;
while(i<=10):
    print(i);
    i=i+1;
```

Algorithm: For the code: Python program to print odd numbers using while loop. (1-50).

Step 1: START
Step 2: Check if the value of var "i" is less than equal to 50, while "i" is less than equal to 50 go to step 4 otherwise go to step 5.
Step 3: increment the value of "i" by 2
Step 4: print the Value of "i", while (i%2 != 0)
Step 5: STOP

Code 2: Python program to print all odd numbers between 1-50 using the while Loop.

```
i = 0;
while (i<=50):
    if (i % 2 == 0):
        print (i);
    i = i+1;
```

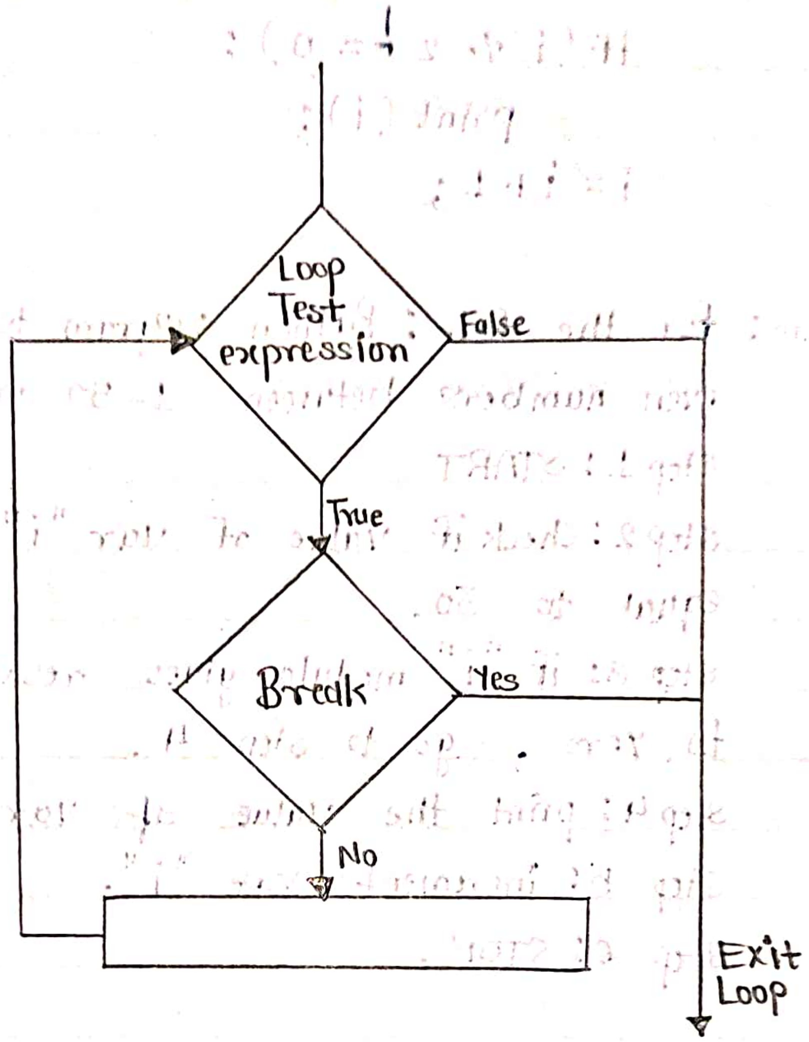
Algorithm: for the Code : Python program to print all ~~odd~~ even numbers between 1-50 using ~~for~~ while Loop

```
Step 1 : START
Step 2 : Check if value of var "i" is less than or equal to 50.
Step 3: if "i" modulo gives result remainder equal to zero, go to step 4.
Step 4 : print the value of var "i".
Step 5: increment var "i".
Step 6 : STOP.
```

Code 3: Python program to print all ~~Ev~~ even numbers between 1-50 Using while Loop.

```
i=0;
~~while (i % 2 == 0):~~
while (i <= 50):
    if (i % 2 == 0):
        print (i);
    i = i + 1;
```

Flowchart: Break statement general Flowchart :



Theory: **Break Statements :**

Break statement Is used to terminate the execution of the Loop. Break statement is used to bring the control out of the Loop when some external condition is triggered. break statement is put inside the loop body. It terminates the current loop ie the loop in which it appears & resumes execution at the next statement immediately after the end of that loop. If the break statement is inside a nested loop, the break statement will terminate the innermost loop.

Syntax :

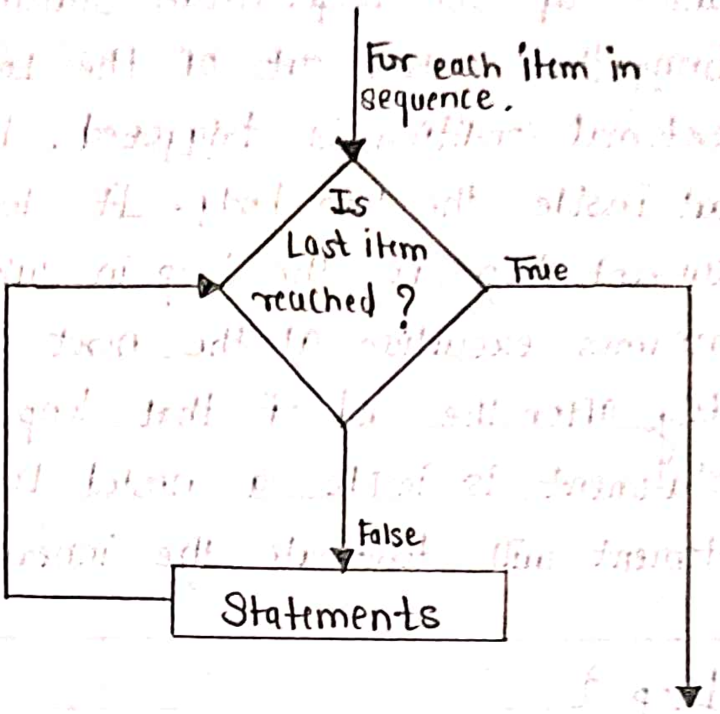
```
Loop {
        condition :
            break;
}
```

Code 4 : Python program to demonstrate the use of break statement. Code to print only 3 numbers in a loop, conditioned at 10 iterations ?

```
i = 1;
while (i <= 10):
        print (i);
        i = i + 1;
        }
```

```
if (i ==4):
        break;
```

Flowchart: For Loop general flowchart :



For loop :

It has the abality to iterate over the items of any sequence, such as List or a String. A for Loop is used for iterating over a sequence (that is a list, a tupple, a dictionary, a set or a string). This is less like the "for" keyword in other programming langauge & works more like an iterator method as found in other object oriented programming Langauges. With the for loop we can execute a set of statement, once for each item in a List, tupples or a set.

Syntax: for iterating_var in sequence :
            statements;

Algorithm: For the code : Python program to print the predefined Array

```
Step 1: START
Step 2: Initialize the list called 'fruits' containing the elements.
Step 3: For each element in x in the 'fruits' list:
        print 'x' followed by a space.
step 4: For each character 'x' in the string 'Banana':
        print 'x' followed by a space.
step 5: STOP
```

Conclusion: The practical has been successfully studied.

Code 5:

```
fruits = ["Banana", "Apple", "Gauva"]
for x in fruits :
    print(x, end = " ");
for x in "Banana":
    print(x);
```

Algorithm :- For the code: Python program to demonstrate the use of "range" Function.

```
Step 1: START
Step 2: initialize a range starting from 5, ending before 100, with a step of 5.
Step 3: For each value 'x' in this range:
            print 'x' followed by a space.
Step 4: STOP.
```

Code 6:

```
for x in range (10) :            // 0 .... 9
    print (x, end = " ");
```

ARISE & SHINE

Code 7:

```
for x in range (5, 10):          // 5 .... 9
    print (x, end = " ");
```

Code 8:

```
for x in range (5, 100, 5):      // 5, 10, 15 ...., 95
    print (x, end = " ");
```

Conclusion: The practical has been successfully studied.