EXPERIMENT NO : 1C

Python programs to implement Looping in Python

(while loop ,for loop, nested loop)

NAME : AKASH RAMKRIT YADAV	**ID.NO: VU4F2122016**
BATCH : A BRANCH : IT DIV : A	

Aim:-To implement Looping in Python : (while loop , for loop, nested loop)

THEORY:

OUTPUT:

```
Python 3.11.0a4 (main, Jan 17 2022, 12:57:32) [MSC v.1929 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
#AKASH YADAV      ID.NO:VU4F2122016      EXP:1C     DATE:23/1/2023
```

# Python program to illustrate
# while loop

A]

```
a=0
while a<3:
    a=a+1
    print("HELLOW AKASH!")


HELLOW AKASH!
HELLOW AKASH!
HELLOW AKASH!
```

B]

```
# checks if list still
# contains any element
A=[1,2,3,45,6,7,8,]
```

```
while a:
    print(A.pop())
```

```
8
7
6
45
3
2
1
```

#For loop

A]simple for loop programme

A `for` loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

```
Friends = ["akash", "viram", "suraj"]
for x in Friends:
 print(x)
```

```
akash
viram
suraj
```

B] Looping Through a String

Even strings are iterable objects, they contain a sequence of characters:

```
for x in "akash":
  print(x)
```

```
a
k
a
s
h
```

C] The break Statement:

With the `break` statement we can stop the loop before it has looped through all the items:

```
Friends = ["akash", "viram", "suraj"]

for x in Friends:
  print(x)
  if x == "viram":
    break

akash
viram
```

D] The range() Function

To loop through a set of code a specified number of times, we can use the `range()` function,

The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
1]for x in range(5):
  print(x)


0
1
2
3
4


2] for x in range(2,5):
    print(x)


2
3
4
```

E]Else in For Loop

The `else` keyword in a `for` loop specifies a block of code to be executed when the loop is finished

```
for x in range(7):

  print(x)

else:

  print("Finally finished!")
```

```
0

1

2

3

4

5

6

Finally finished!
```

F] The pass Statement

`for` loops cannot be empty, but if you for some reason have a `for` loop with no content, put in the `pass` statement to avoid getting an error.

```
for x in [0, 2, 3]:
  pass
```

#Nested Loops

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

A]

```
NAME = ["AKASH", "VIRAM", "SURAJ"]

SURNAME = ["YADAV", "YADVANSHI", "AHIR",]


for x in NAME:

  for y in SURNAME:

    print(x, y)
```

```
AKASH YADAV

AKASH YADVANSHI

AKASH AHIR

VIRAM YADAV

VIRAM YADVANSHI

VIRAM AHIR

SURAJ YADAV

SURAJ YADVANSHI

SURAJ AHIR
```

B] Printing multiplication table using Python nested for loops :

```
# Running outer loop from 2 to 3
  for i in range(4, 6):
  # Printing inside the outer loop
  # Running inner loop from 1 to 10
    for j in range(1, 11):
        # Printing inside the inner loop
         print(i, "*", j, "=", i*j)
    # Printing inside the outer loop
    print()
```

```
4 * 1 = 4        5 * 1 = 5
4 * 2 = 8        5 * 2 = 10
4 * 3 = 12        5 * 3 = 15
4 * 4 = 16        5 * 4 = 20
4 * 5 = 20        5 * 5 = 25
4 * 6 = 24        5 * 6 = 30
4 * 7 = 28        5 * 7 = 35
4 * 8 = 32        5 * 8 = 40
4 * 9 = 36        5 * 9 = 45
4 * 10 = 40       5 * 10 = 50
```