*Loops :-

A loop is a programming construct that allows you to repeat a block of code multiples times until a certain condition is met. Loops help avoid writing the same code and again.

Types of loops :-

1) for loop
2) while loop.
3) Nested loops.
   - * for loop inside for loop
   - * while loop inside while loop
   - * For loop inside while loop
   - * while loop inside for loop

1) for loop :-

A for loop is a control flow statement used to repeat a block of code a specific number of times or iterate over a sequence such as a list, tuple, or string.

It is widely used when the number of iterations is known in advance.

syntax :-

```
for variable in sequence:
    #code to execute
```

ex:-

```
for i in range(5):
    print("Hello",i)
```

o/p:-

```
Hello 0
Hello 1
Hello 2
Hello 3
Hello 4
```

(or)

* for loop with sequence

syntax:

```
for var in sequence name: #var = ittirative variable
    statements.
```

* for loop with range( )

syntax:

in C

(initialization ; condition; incrementation)

(i=0;i<=0;i=i+1)

in python

range(start value, stop value, step size)

default start value is : 0

step size is : 1

Ex:-

```
value = "Besant Tech"
for a in value:
    print(a)
```

o/p:-

```
B
e
s
a
n
```

T

```
* for i in range(0,10,1): #i=0,1,2,3,...9
      print('Besant Tech")
```

```
o/p:- Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
      Besant Tech
```

```
* value = "Besant Tech"
    print (value [0])
```

```
o/p:- B
```

```
* animals = ['monkey', 'loin', 'dog', 'cat', 'rabbit', 'cow']
  print (animals)
```

```
o/p:- ['monkey', 'loin', 'dog', 'cat', 'rabbit', 'cow']
```

```
* for i in range (1,11,1):
    if (i%2==0):
        print (f"{i}= even number')
    else:
        print(f"{i}= odd number")
```

```
o/p:- 1= odd number
      2 = even number
      3 = odd number
      4 = even number
      5 = odd number
      6 = even number
      7 = odd number
      8 = even number
      9 = odd number
      10 = even number
```

* write a (factorial) program to find factorial of given number. ex 5! = 5*4*3*2*1 = 120

```
num=int(input("Enter a value:"))
fact=1
for i in range(1,num+1,1):
    print("i=",i)
    fact=fact*i
    print(fact)
```

```
o/p:- enter a value: 5
      i=1
      1
      i=2
      2
      i=3
      6
      i=4
      24
      i=5
      120.
```

2) while loop :-

A while loop is a control flow statement that allows code to execute repeatedly based on a boolean condition. The loop continues executing the block of statements until the condition evaluates to false. proper use of while loops is essential when the number of iterations is not predetermine and depends on dynamic conditions.

Syntax:-

```
while
initialization of while loop
while (condition):
    statements
    inc/dec while loop.
```

Ex:- i=1

```
while (i<=10):
    print ("hi good Morning")
    i= i+1
```

o/p:-

```
Hi good Morning
hi good Morning
      |
      |
      |
      |
      |
hi good Morning.
```

Q:-

```
'''1x1=1
1x2=2
.....
1x10= '''

num = int(input("enter a value:"))
i=1
while (i<=10):
    print (f"(num)x{i}= {num*i}")
    i+=1
```

o/p:- enter a value=9

```
9x1=9
9x2=18
9x3=27
9x4=36
9x5=45
9x6=54
9x7=~~6[illegible]~~63
9x8=~~78~~72
9x9=81
9x10=90.
```

*

```
x=1
while (x<=10):
    if (x%2 ==0):
        print(x,"=even number");
    else:
        print(x, "=odd number");
    x=x+1
```

o/p:-

```
1= odd number
2 = even number
3 = odd number
4 = evennumber
  |
  |
  |
10 = even number.
```

③ Nested loops:-

A nested loop means putting one si+loop inside another loop. The inner loop runs completely every time the outer loop runs once.

Ex:-

```
for i in range(1,4):
    for j in range (1,3):
        print (i,j)
```

o/p:-

```
1 1
1 2
2 1
2 2
3 1
3 2
```

* for loop inside for loop :-

A for loop inside another for loop is called a nested for loop
It means one loop runs inside another.
The outer loop runs first, and for each time it runs, the inner loop runs completely.

syntax:-

```
for var_a in range( ): #outer loop
      for var_b in range( ): # inner loop
            statement in inner loop
      statements in outer loop.
```

ex:-

```
-> for i in range(1,4):                 o/p:-  1 1
      for j in range(1,3):                     1 2
         print (i,j)                           2 1
                                               2 2
                                               3 1
                                               3 2
```

```
-> for row in range(1,4):
      for col in range(1,4):                  o/p:- (1,1)  (1,2)  (1,3)
         print(f"({row},{col})",end=" ")          (2,1)  (2,2)  (2,3)
   print( )                                       (3,1)  (3,2)  (3,3)
```

```
-> for i in range(1,6,1):                 o/p:-  1   2    3    4    5
      for j in range(1,6,1):                     2   4    6    8    10
         print(i*j, end="\t")                    3 / 6    9    12   15
      print( )                                   4   8    12   16   20
                                                 5   10   15   20   25
```

```
-> for i in range(1,6,1):
      for j in range(1,6,1):              o/p:-   1 1 1 1 1
         print(i,end=" ")                         2 2 2 2 2
      print( )                                    3 3 3 3 3
                                                  4 4 4 4 4
                                                  5 5 5 5 5
```

```
-> for i in range(1,6,1):                 o/p:-
      for j in range(1,6,1):                           2           4
         if((i*j)%2==0):                         2     4     6     8    10
            print(i*j,end="\t")                        6           12
      else:                                      4     8     12    16   20
         Print(" ", end="\t")                          10          20
```

Pattern programs:-

```
→ for i in range(1,6,1):
      for j in range(1,i+1,1):
          print("*", end=" ")
      print
```

o/p:-

```
*
* *
* * *
* * * *
* * * * *
```

```
→ for i in range(1,6,1):
      for j in range(6,i,-1):
          print("*",end=" ")
      print( )
```

o/p:-

```
* * * * *
* * * *
* * *
* *
*
```

```
→ for i in range(1,6,1):
      for j in range(1,i+1,1):
          print(j,end=" ")
      print( )
```

o/p:-

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
→ for i in range(1,6,1):
      for j in range(1,7-i,1):
          print(j,end=" ")
      print( )
```

o/p:-

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

* while loop inside while loop:-

A while loop inside another loop is called a nested while loop. The outer loop runs first, and for each time it runs, the inner loop runs completely.

- The outer while loop controls the number of Main iterations.
- The inner while loop executes completely for each iteration of the outer loop.

ex:-
```
i=1                                   1 1
while i<=3:  #outer loop              1 2
    j=1                               2 1
    while j<=2:  #inner loop          2 2
        Print(i,j)                    3 1
        j+=1                          3 2
    i+=1
```

syntax:-

```
initialization of outer loop
while(condition of outerloop):
    initialization of inner loop
    while(condition of inner loop):
        statements of inner loop.
        incldec of inner loop.
    statements of outer loop
    incldec of outer loop.
```

->
```
i=1
while(i<=10):
    j=1
    while(j<=10):
        Print(i*j ,end="\t")
        j=j+1
    Print()
    i=i+1
```

o/p:-
```
1  2  3  4  5  6  7. 8  9
2  4  6  8  10 12 14 16 18
3  6  9  12 15 18 21 24 27
4  8  12 16 20 24 28 32 3
5  10 15 20 25 30 35 40 45
6  ,  ,  ,  ,  ,  ,  ,  ,
7  ,  ,  ,  ,  ,  ,  ,  ,
8  ,  ,  |  ,  ,  ,  ,  ,
9  ,  ,  ,
10 20 30 40 50 60 70 80 90 100
```

->
```
i=1
while (i<=5):
    j=1
    while(j<=i):
    Print (i,end=" ")
        j=j+1
    Print()
    i=i+1
```

o/p:-
```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
-> num=2
   i=1
   while (i<=5):
       j=1
       while (j<=i):
           print(num, end=" ")
           num=num+2
           j= j+1
       print()
       i=i+1
```

o/p:-

```
2
4 6
8 10 12
14 16 18 20
22 24 26 28 30
```

```
-> i=1
   while (i<=5):
       j=1
       while(j<=i):
           print("*", end=" ")
           j+=1
       print()
       i+=1
  i=1
  while (i<=4):
      j=4
      while(j>=i):
          print("*", end=" ")
          j=1
      print()
      i+=1
```

o/p:-

```
 *
 **
 ***
****
*****
*****
***
**
*
```

```
-> for i in range(1,6,1):
       print("*"*i, end=" ")
       print(" "*(2*(5-i)), end=" ")
       print("*"*i, end=" ")
       print()
   for i in range(4,0,-1):
       print("*"*i, end=" ")
       print(" "*(2*(5-i)), end=" ")
       print("*"*i, end=" ")
       print()
```

o/p:-

```
 *                 *
 **               **
 ***             ***
****           ****
**********
****          ****
***            ***
**              **
*                *
```

```
→ for i in range(1,6,1):
      for j in range(5,i,-1):
          print(" ",end=" ")
      for k in range(1,i+1, 1):
          print("*",end=" ")
      print( )
```

op

```
        *
       * *
      * * *
     * * * *
    * * * * *
```

```
→ for i in range(1,6,1):
      for j in range(5,i,-1):
          print(" ",end=" ")
      for k in range(1,i+1,1):
          print("*",end=" ")
      print()
  for i in range(4,0,-1):
      for j in range(5,i,-1):
          print(" ",end=" ")
      for k in range(1,i+1,1):
          print("*",end=" ")
      print( )
```

op :-

```
        *
       * *
      * * *
     * * * *
    * * * * *
     * * * *
      * * *
       * *
        *
```

* for loop inside while loop

A for loop inside a while loop means that the while loop runs first, and inside it, the for loop executes completely during each iteration of the while loop.

The outer while loop controls how Many times the inner for loop runs.

Syntax:-

```
initialization of while loop
while (condition):
    for var in range( ):
            statements of for loop
    statements of while loop
    inc/dec of while loop
```

ex:-

```
→ i=1
  while i<=3:                      # outer while loop
     for j in range(1,4):  # Inner for loop
           print(i,j)
     i+=1
```

o/p:-

```
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3
```

```
→ count=1
  while count <=3:                          # outer while loop
       print(f"Table of {count}:")   #display which table is being printed
       for i in range(1,6):                  # Inner for loop.
             print(f"{count} x {i} = {count * i}")
       print()
                                            # Blank line for spacing
       count +=1
```

o/p:-

```
Table of 1:          Table of 2:          Table of 3:

1x1 = 1              2x1 = 2              3x1 = 3
1x2 = 2              2x2 = 4              3x2 = 6
1x3 = 3              2x3 = 6              3x3 = 9
1x4 = 4              2x4 = 8              3x4 = 12
1x5 = 5              2x5 = 10             3x5 = 15
```

*while loop inside for loop:-

A while loop inside for loop means the for loop runs first, and for each iteration of that loop, the while loop executes until its condition becomes false.

The for loop controls how many times the outer process runs, and the while loop handles repeated actions inside it based on a condition.

syntax:-

```
for var in range( )
    initialization of while loop
    while(condition):
          statements of while loop.
          inside of while loop
statements of for loop
```

ex:-

```
for num in range(1,6):                  # outer for loop
    print(f"number: {num}")
    count = num                         # Initialize variable for while loop
    while count >0:                     # Inner while loop.
        print("*", end=" ")
        count -=1                       # Decrease count each time.
    print("\n")                         # Move to next line.
```

o/p:-

```
Number:1   Number:2   Number:3   Number:4   Number:5
*          **         ***        ****       *****
```

1) write a program to check the given number is prime or not.

```
num = int(input("enter a number:"))
count=0
for i in range(2,num,1): #i=2,3,4,5,6,7
    if(num%i==0):
        count = count+1 # =1+1=2
    else:
        count = count # =2 .
print(count)
if(count== 0):
    print("its a prime number")
else:
    print("its not a prime number")
```

o/p:-

```
enter a number:8
2
its not a primenumber
```

2) write a program to count even number, odd number, natural numbers from 1 to 20.

```
en=0
on=0
nn=0
for i in range (1,21,1):
    nn=nn+1
    if (i%2==0):
        en=en+1
    else:
        on=on+1
print (f"natural numbers={nn}\neven numbers={en}\n odd numbers={on}")
```

o/p:-

```
natural numbers=20
even numbers=10
odd numbers=10
```

```
→ r1 = int(input("enter a number:"))
  r2 = int(input("enter a number:"))
  en=0
  on=0
  nn=0
  for i in range(r1,r2+1,1):
      nn=nn+1
      if (i%2==0):
          en=en+1
      else:
          on=on+1
print(f"natural numbers={nn}\neven numbers={en}\nodd numbers={on}")
```

o/p:-

```
enter a number:1
enter a number:10
natural numbers=10
even numbers=5
odd numbers=5.
```

* loop statements:-

Loop statements are used to repeat a block of code multiple time until a specific condition is met.

1. Break statement:-

It stops the loop completely when a certain condition is true.

Ex:-

```
for i in range(1,6):
    if i==3:
        break
    print(i)
```

o/p:-

```
1
2
```

② continue statement:

It skips the current iteration and moves to next one.

Ex:-

```
for i in range(1,6):
    if i==3:
        continue.
    print(i)
```

o/p:-

```
1
2
3
4
```

③ pass statement:-

It is used place holder when you don't want any action in the loop.

eg:-

```
for i in range(1,6)
    if i==3:
        pass
    print(i)
```

o/p:-

```
1
2
3
4
5
```

```
* num=int(input("enter number:"))
  for i in range(2,num,1):
     if (num%i==0):
        print (f"{num}is not a prime number')
        break
  else:
     print (num,"is prime number")
```

o/p:-

```
enter a number:21
21 is not a prime
            number.
```

```
* for i in range(2,11,1): #i=2,3,4,5,6,7,8,9,10
     for j in range(2,i,1):
        if (i%j==0):
           break
  else:
     print(i)
```

o/p:-

```
2
3
5
7.
```