**Loops:**

Looping statement: A loop statements is used to execute a statement or group of statements multiple times

**For:** for loop is used for sequential traversal example traversing a list or string or range etc.

Syntax:

For iterator-var in sequence:

statements

Program :

n=int(input())

for I in range (1, n+1):

print(I, end=””)

input: 10

output: 1,2,3,4,5,6,7,8,9,10

**break statement:** break statement is used to terminate the loop before the ending of the sequence it brings control out of the loop

syntax:

for I in range(1,10):

if I == 5:

break

print(I, end=””)

**continue statement:** it returns the control to the beginning of the loop

Program:

for I in range(1,10):

if i%2 == 0 or i%3 == 0:

continue

print(I, end=””)

output: 1, 5, 7

**pass statement:** pass statement is used to represent empty control statements , functions and classes.

Syntax:

For I in range (1,10):

If i%3==0 or i% 2 == 0:

Pass

else:

print(I, end=” ”)

output: 1, 5, 7

**for else:** for else is used to avoid flag variables else part will execute when the loop is terminated due to end of the sequences. else part will not execute if loop is terminated due to break statement.

Syntax:

for iteration\_var in sequence:

if condition:

break

else:

statements

**Nested for:** for written a for every iteration of the outer loop, inner loop will execute completely

Syntax:

for iteration\_var in sequence:

for iter\_var in sequence:

statements

**while:** if the number of iterations doesn’t know in advance or if the sequence needs to update inside of the loop then while is useful

syntax:

while condition:

statements

Program:

n=int(input())

i=1

while i<=n:

print(I, end=” “)

i+=1

**Nested loop:** A looping statement inside of another looping statement is called nested loop that may be for inside for , while inside while, while inside for , for inside while

Program:

n=int(input())

while n>9:

s=0

while n!=0

s=s+n%10

n=n//10

n=s

print(n)

input: 276

output: 6