WHILE LOOP & NESTED LOOPS & NESTED IF

Abdelrahman A. Mohamed

Main Headlines

- “While” Loop
- “While” with else statement
- “For” with else statement
- “Break” statement
- “Continue” statement
- “Pass” statement
- Nested loop
- Nested if

While Loop

- Used to iterate over a block of code as long as the test expression (condition) is true.
- Syntax of while Loop in Python

```
while test_expression:
    Body of while
```

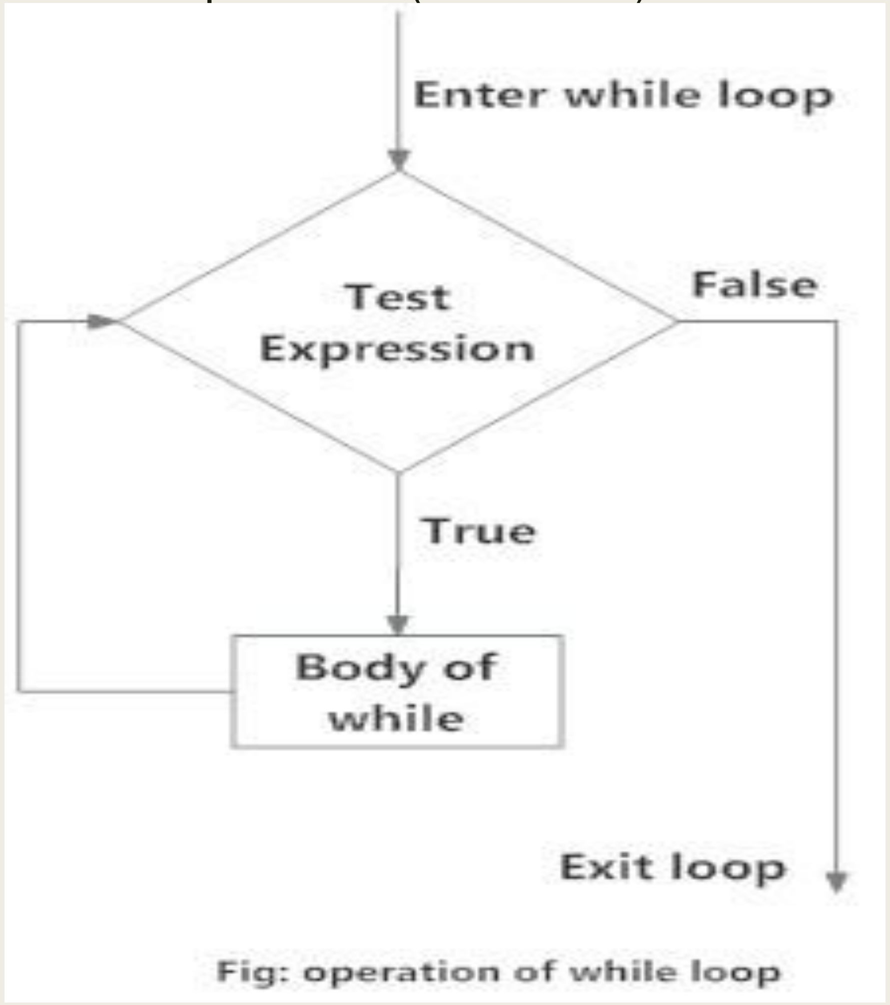
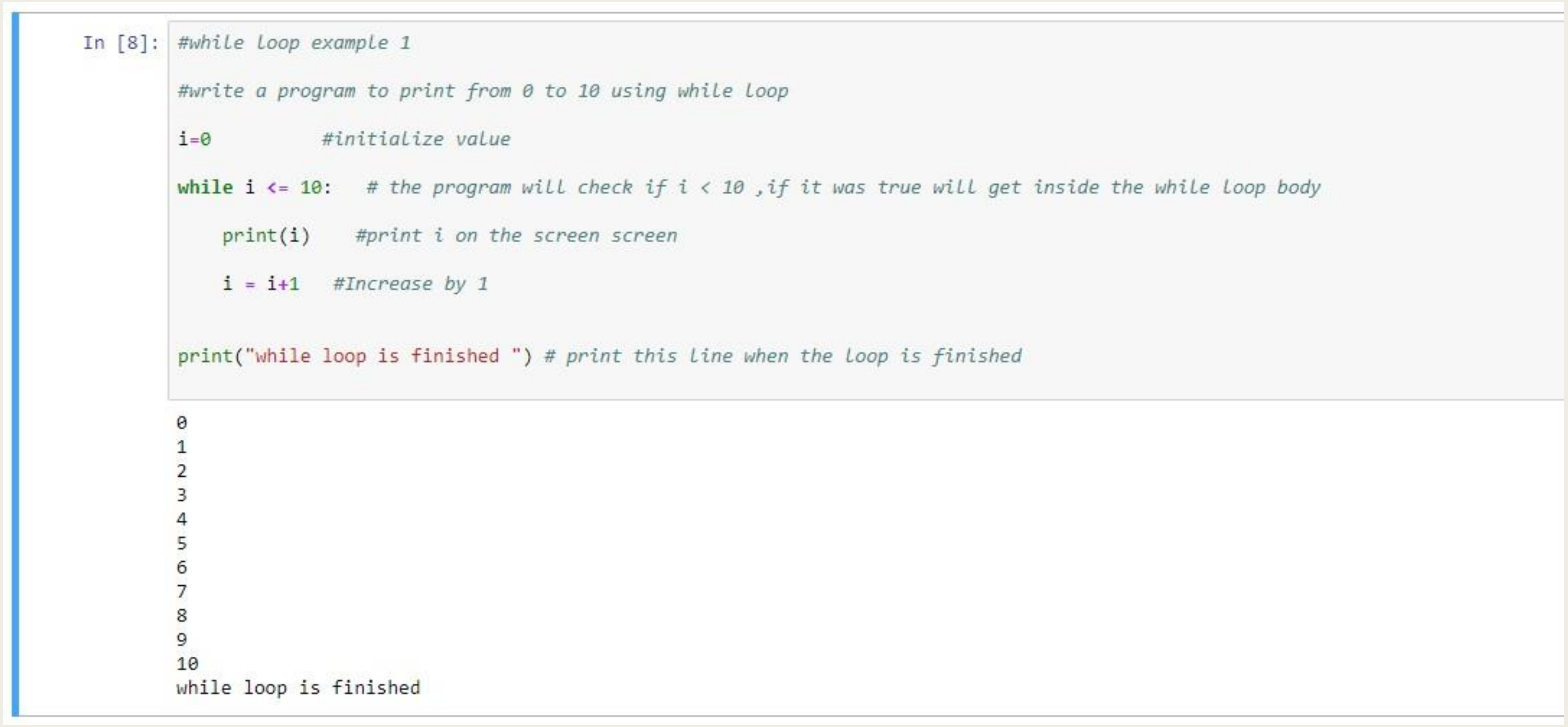


Fig: operation of while loop

Abdelrahman A. Mohamed

While Loop Example



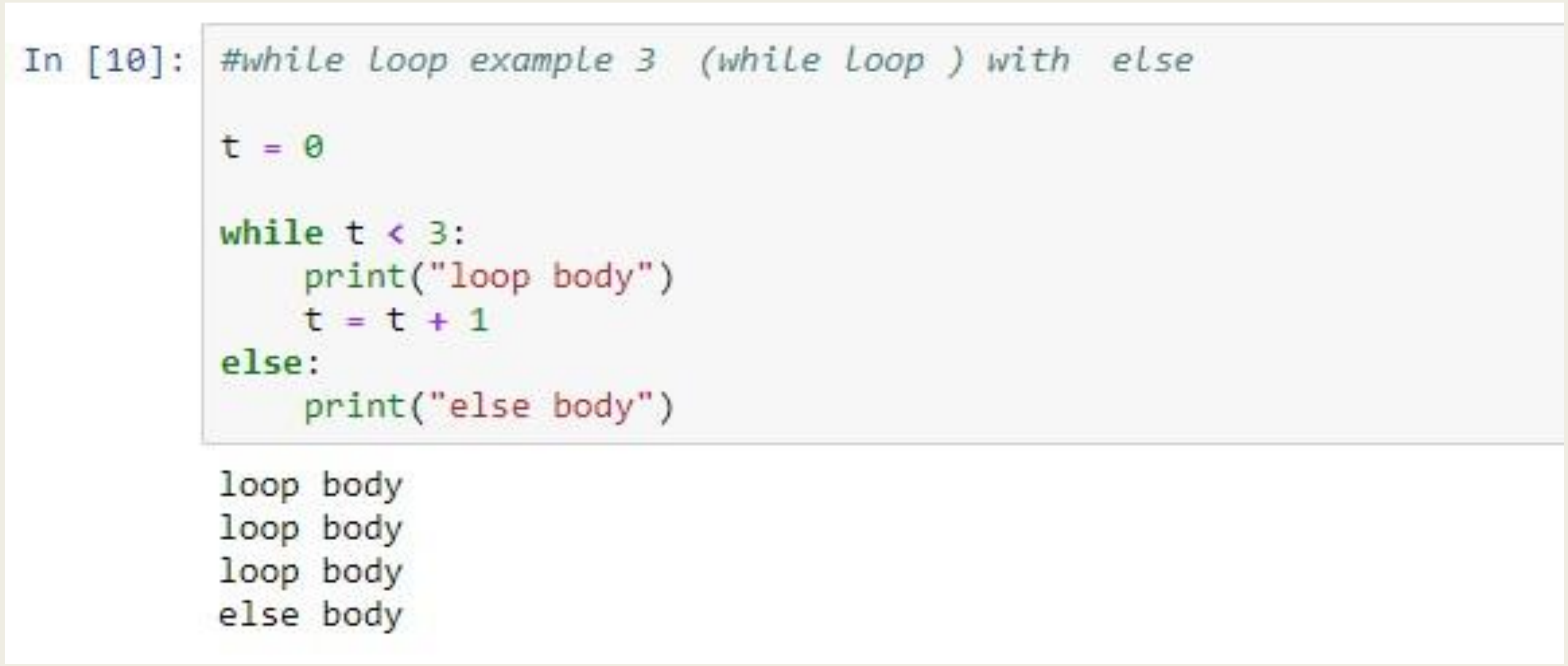
```
In [8]: #while loop example 1

        #write a program to print from 0 to 10 using while loop

        i=0          #initialize value

        while i <= 10:   # the program will check if i < 10 ,if it was true will get inside the while loop body

            print(i)    #print i on the screen screen

            i = i+1   #Increase by 1


        print("while loop is finished ") # print this line when the loop is finished
```

```
0
1
2
3
4
5
6
7
8
9
10
while loop is finished
```

Abdelrahman A. Mohamed

“While” Loop with else

- If the body of “while loop” evaluates true will implement it .
- If evaluates False will implement the else body.

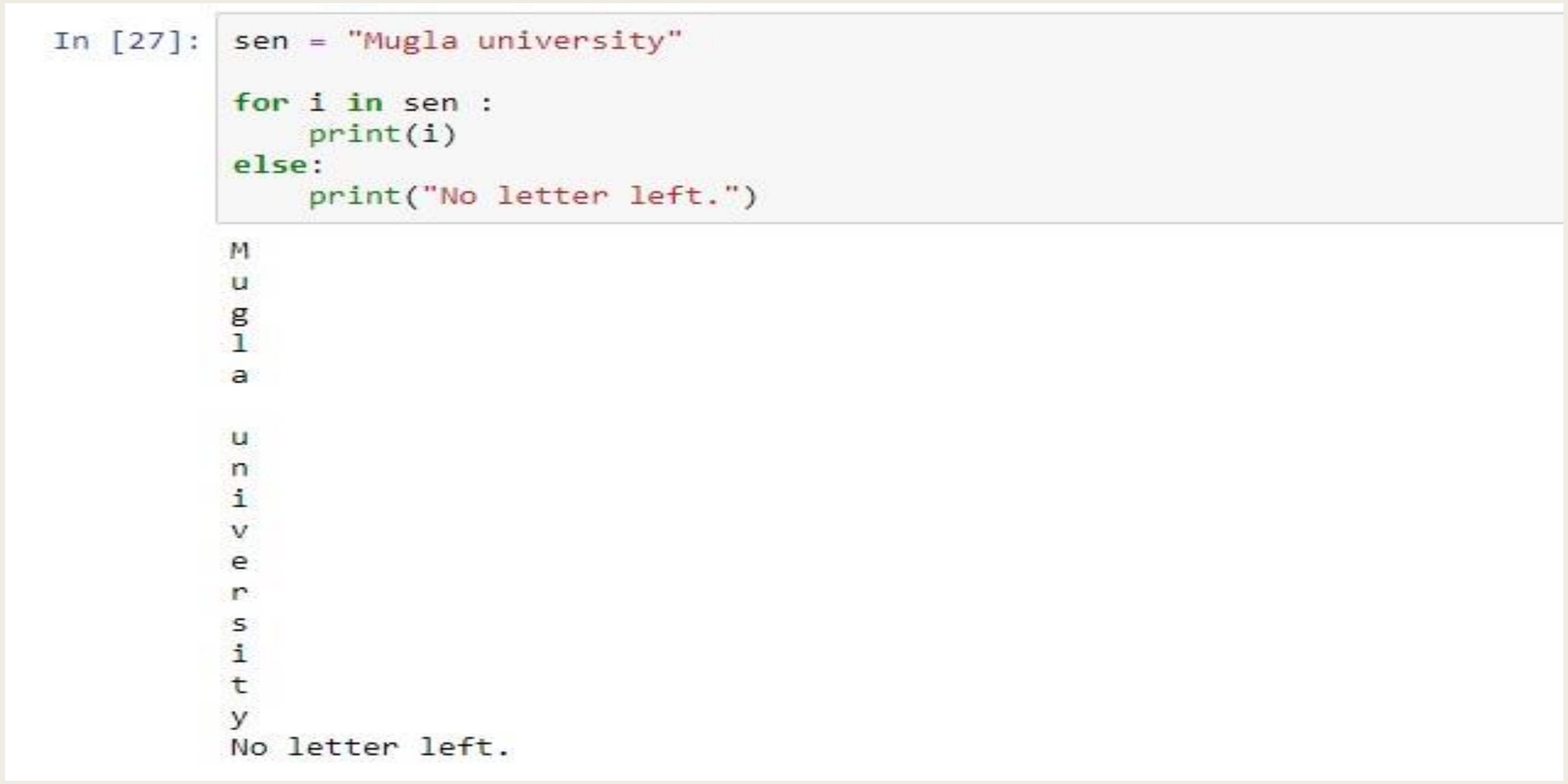


```
In [10]: #while loop example 3  (while loop ) with  else

         t = 0

         while t < 3:
             print("loop body")
             t = t + 1
         else:
             print("else body")
```

```
loop body
loop body
loop body
else body
```

“For” Loop with “else”

- As the “while” loop ,If the body of “for loop” evaluates true will implement it .
- If evaluates False will implement the else body.

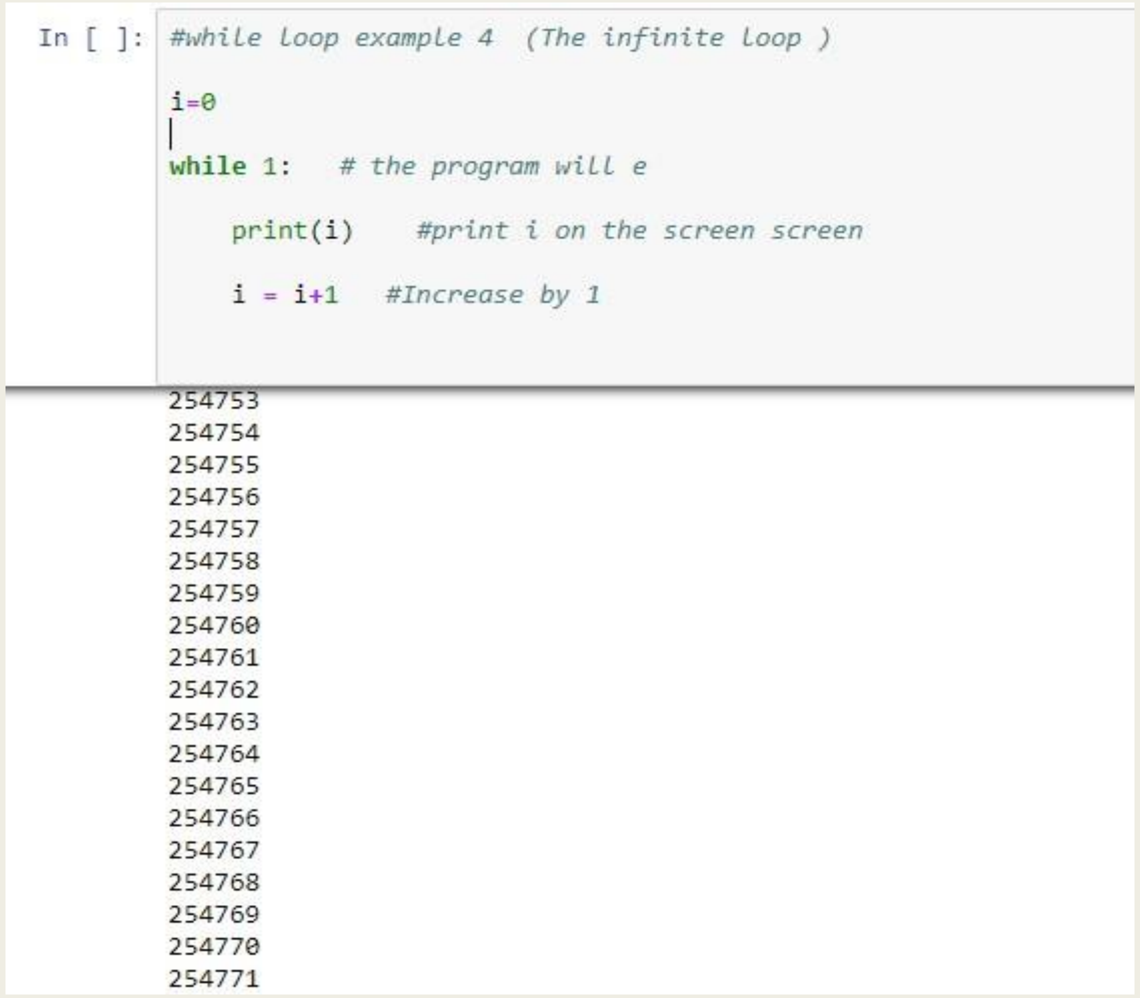


```
In [27]: sen = "Mugla university"

         for i in sen :
             print(i)
         else:
             print("No letter left.")

M
u
g
l
a
 
u
n
i
v
e
r
s
i
t
y
No letter left.
```

Abdelrahman A. Mohamed

The infinite loop

- It’s mean the condition of “while loop” is always True (1).



```
In [ ]: #while loop example 4  (The infinite loop )

        i=0

        while 1:   # the program will e

            print(i)    #print i on the screen screen

            i = i+1   #Increase by 1
```

```
254753
254754
254755
254756
254757
254758
254759
254760
254761
254762
254763
254764
254765
254766
254767
254768
254769
254770
254771
```

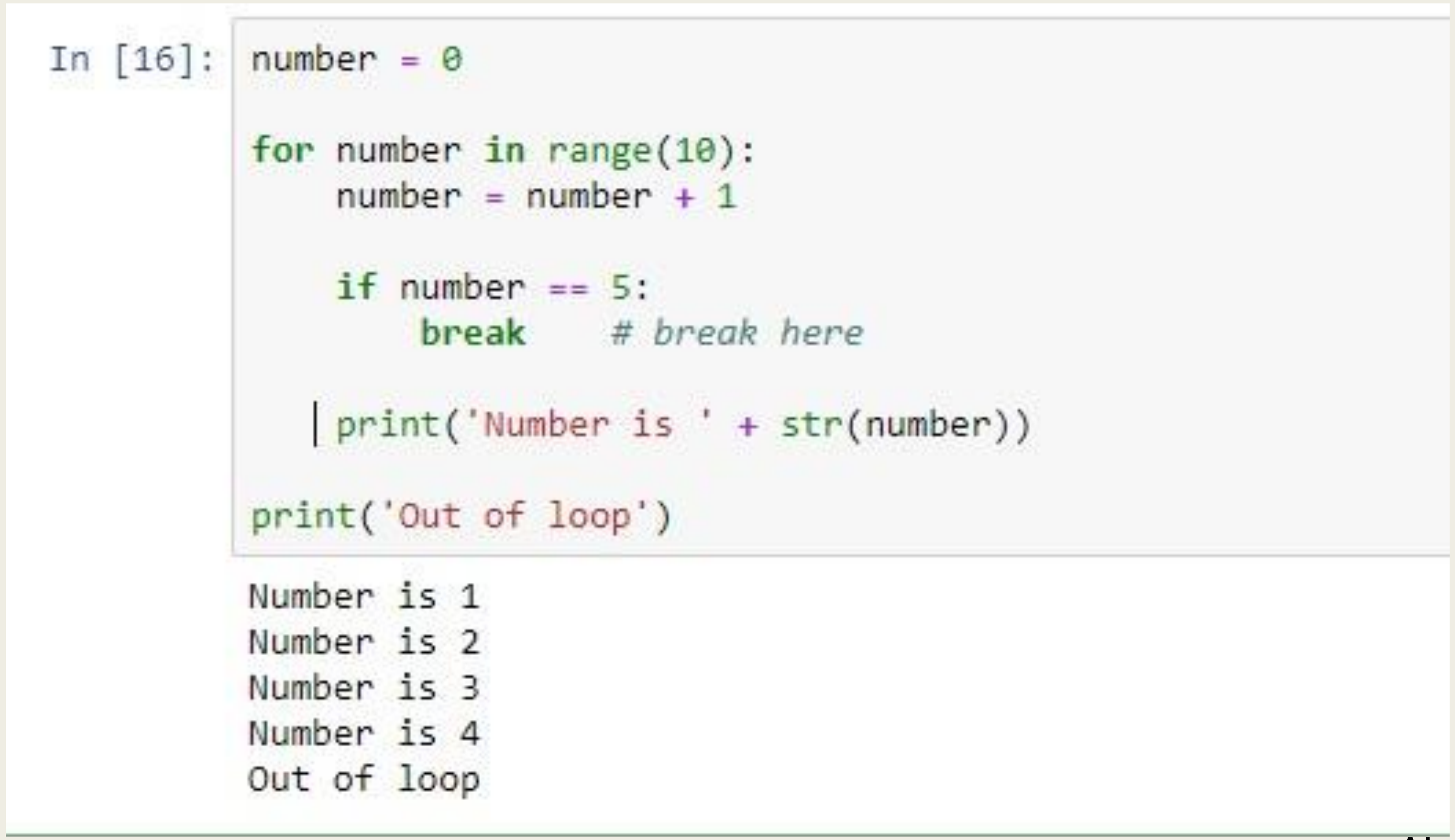
Abdelrahman A. Mohamed

Break, Continue and Pass statements.

- In “for” and “while” loops sometimes an external factor may influence the way your program runs. When this occurs, you may want your program to exit a loop completely, skip part of a loop before continuing, or ignore that external factor. You can perform these actions with **break**, **continue**, and **pass** statements.

Break statement

- Give us opportunity to exit out of a loop when an external condition is triggered
- usually be after a conditional if statement.

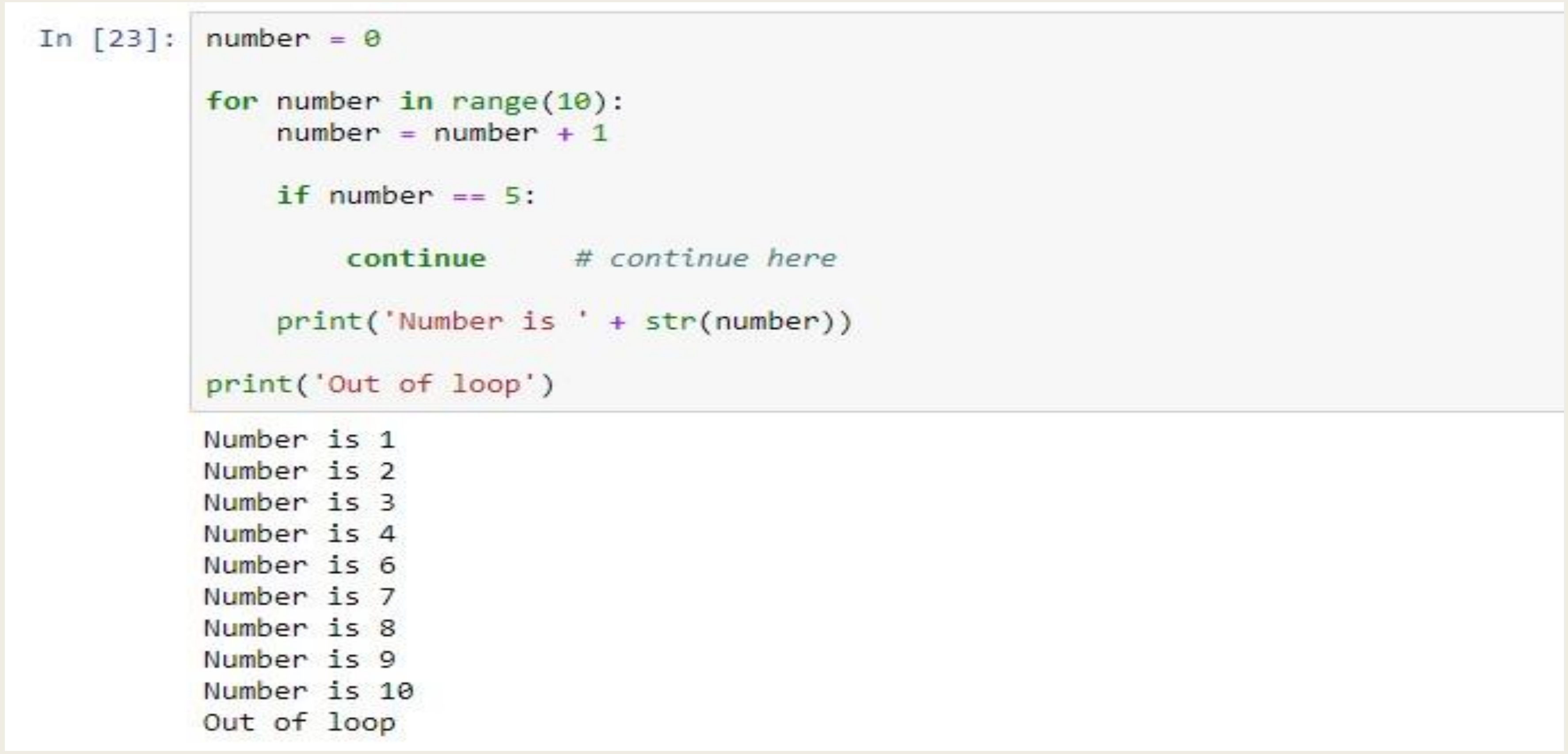


```
In [16]: number = 0

         for number in range(10):
             number = number + 1

             if number == 5:
                 break    # break here

             print('Number is ' + str(number))

         print('Out of loop')
```

```
Number is 1
Number is 2
Number is 3
Number is 4
Out of loop
```

Continue statement

- Gives us the option to skip over the part of a loop where an external condition is triggered, but to go on to complete the rest of the loop
- usually be after a conditional if statement.



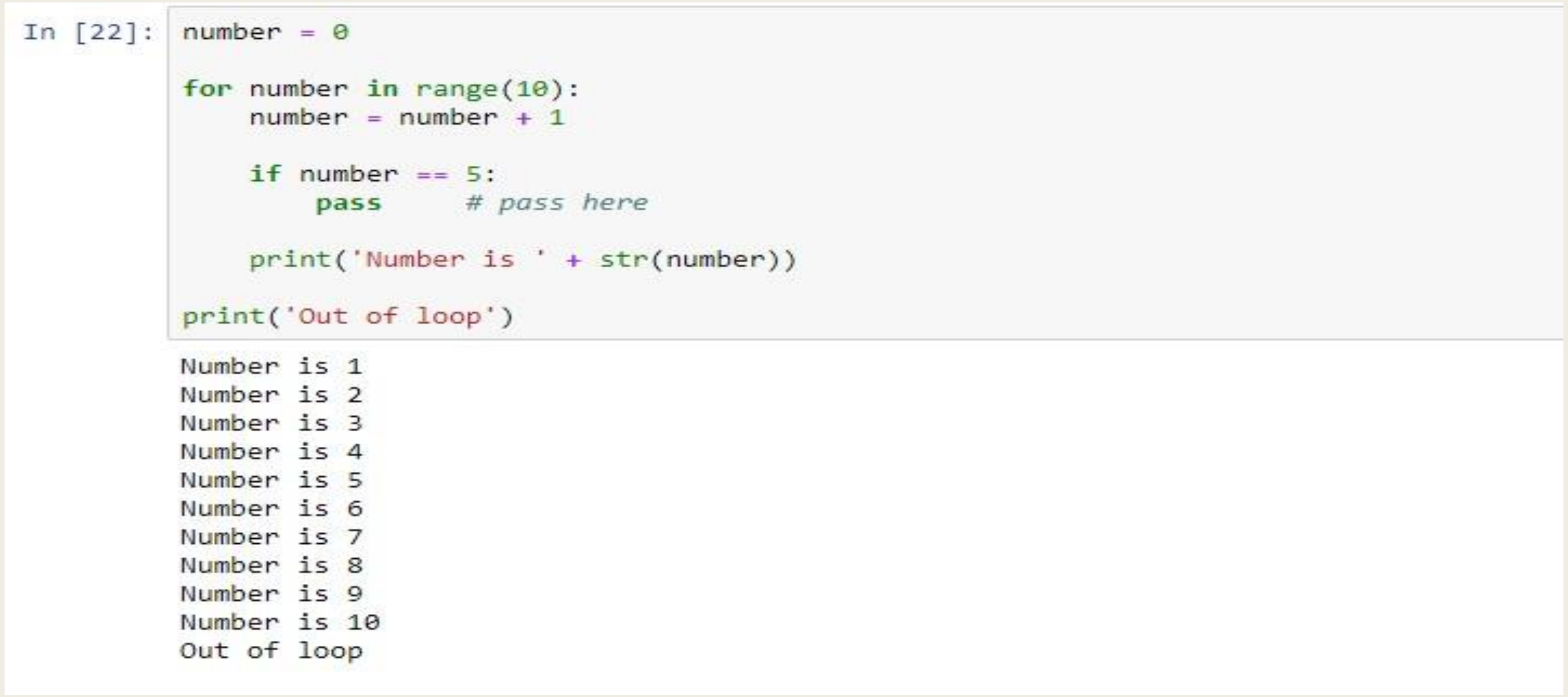
```
In [23]: number = 0

         for number in range(10):
             number = number + 1

             if number == 5:

                 continue    # continue here

             print('Number is ' + str(number))

         print('Out of loop')
```

```
Number is 1
Number is 2
Number is 3
Number is 4
Number is 6
Number is 7
Number is 8
Number is 9
Number is 10
Out of loop
```

Abdelrahman A. Mohamed

Pass Statement

- When an external condition is triggered, the pass statement allows you to handle the condition without the loop being impacted in any way
- usually be after a conditional if statement.



```
In [22]: number = 0

         for number in range(10):
             number = number + 1

             if number == 5:
                 pass     # pass here

             print('Number is ' + str(number))

         print('Out of loop')
```

```
Number is 1
Number is 2
Number is 3
Number is 4
Number is 5
Number is 6
Number is 7
Number is 8
Number is 9
Number is 10
Out of loop
```

Abdelrahman A. Mohamed

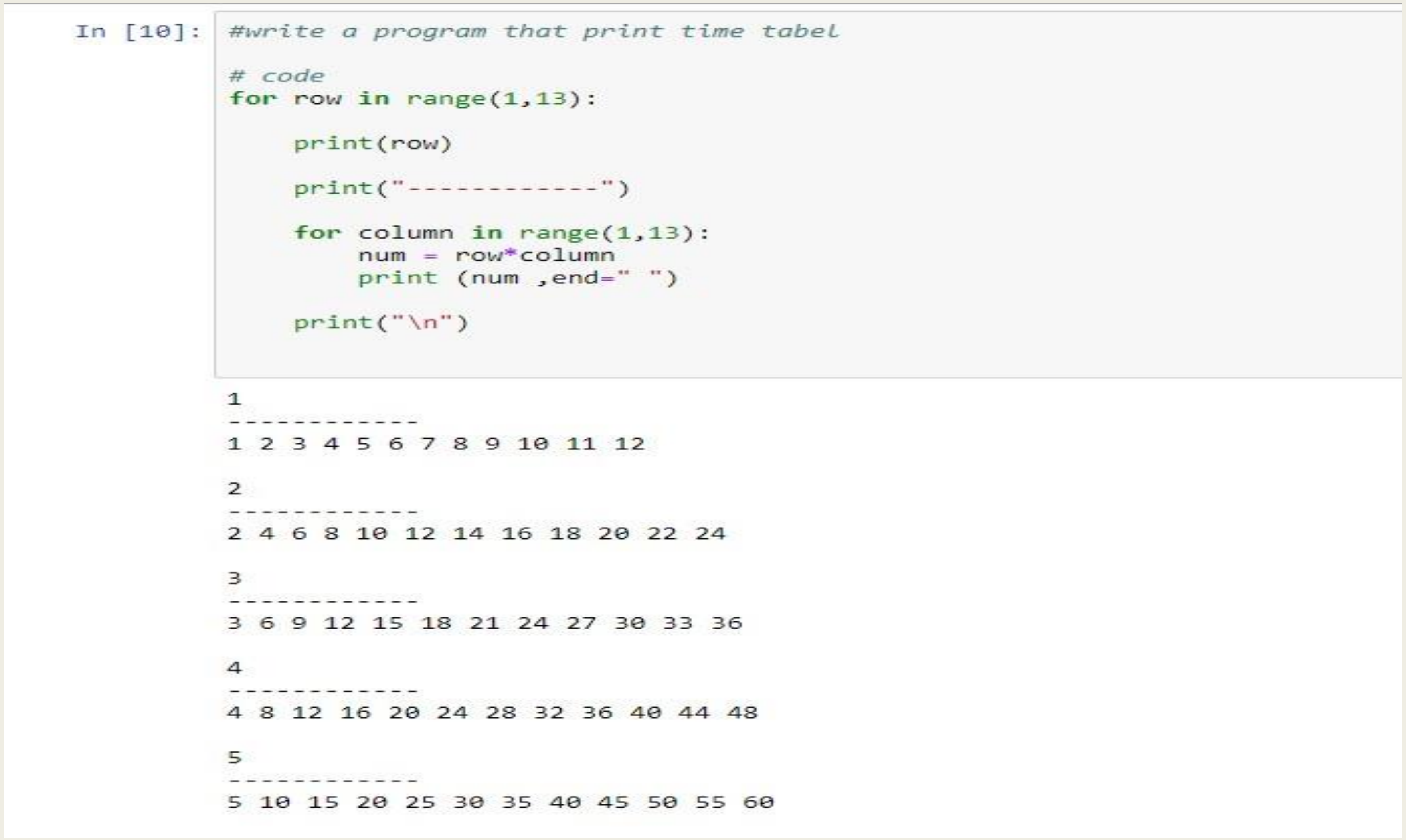
Nested Loops

- It’s meaning using one loop inside another loop.

Nested loop with For loop	Nested loop with While loop

for iterating_var in sequence:
    for iterating_var in sequence:
        statements(s)
    statements(s)
	while expression:
	    while expression:
	        statement(s)
	    statement(s)

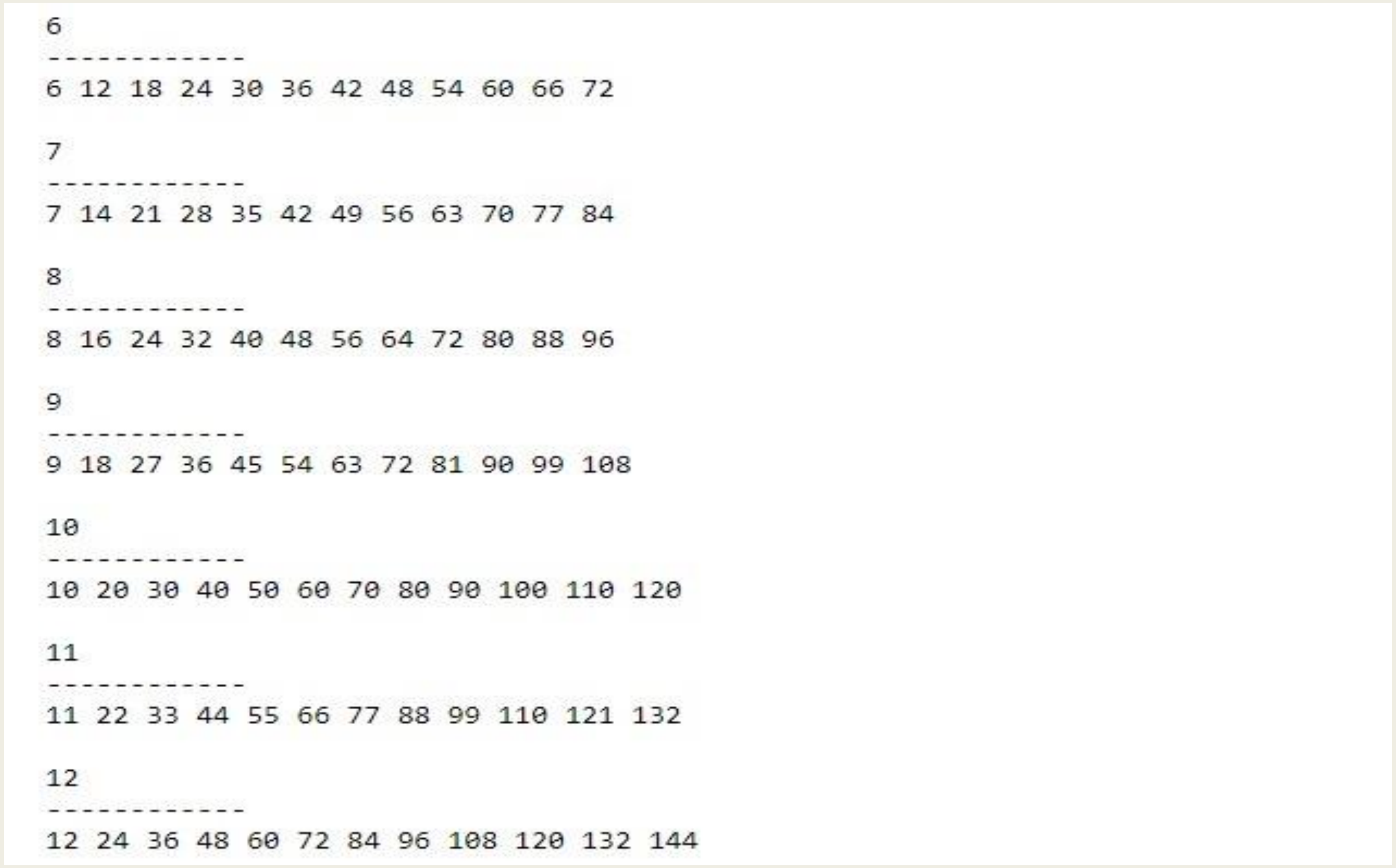
Nested Loops Example



```
In [10]: #write a program that print time tabel

         # code
         for row in range(1,13):

             print(row)

             print("------------")

             for column in range(1,13):
                 num = row*column
                 print (num ,end=" ")

             print("\n")
```

```
1
------------
1 2 3 4 5 6 7 8 9 10 11 12

2
------------
2 4 6 8 10 12 14 16 18 20 22 24

3
------------
3 6 9 12 15 18 21 24 27 30 33 36

4
------------
4 8 12 16 20 24 28 32 36 40 44 48

5
------------
5 10 15 20 25 30 35 40 45 50 55 60
```

Nested Loops Example(continue)



```
6
------------
6 12 18 24 30 36 42 48 54 60 66 72

7
------------
7 14 21 28 35 42 49 56 63 70 77 84

8
------------
8 16 24 32 40 48 56 64 72 80 88 96

9
------------
9 18 27 36 45 54 63 72 81 90 99 108

10
------------
10 20 30 40 50 60 70 80 90 100 110 120

11
------------
11 22 33 44 55 66 77 88 99 110 121 132

12
------------
12 24 36 48 60 72 84 96 108 120 132 144
```

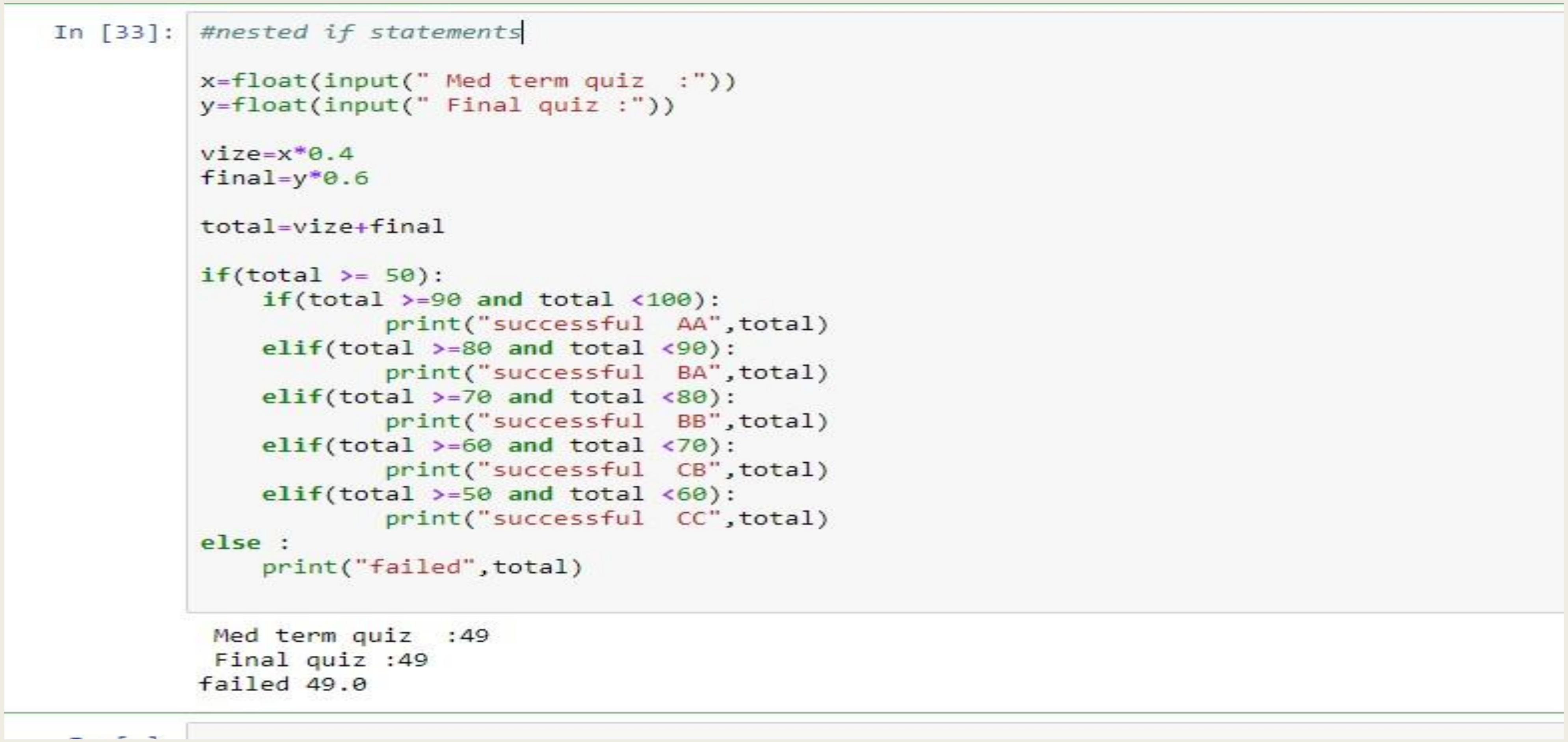
Nested “IF” statements

- Sometimes we have a conditions inside another conditions , In such a situations we can use the nested ”**if”** construct.

The syntax of the nested *if...elif...else* construct may be

```
if expression1:
      statement(s)
      if expression2:
             statement(s)
       elif expression3:
              statement(s)
       elif expression4:
             statement(s)
       else:
             statement(s)
else:
   statement(s)
```

Nested “IF” statements Example



```
In [33]: #nested if statements

         x=float(input(" Med term quiz  :"))
         y=float(input(" Final quiz :"))

         vize=x*0.4
         final=y*0.6

         total=vize+final

         if(total >= 50):
             if(total >=90 and total <100):
                     print("successful  AA",total)
             elif(total >=80 and total <90):
                     print("successful  BA",total)
             elif(total >=70 and total <80):
                     print("successful  BB",total)
             elif(total >=60 and total <70):
                     print("successful  CB",total)
             elif(total >=50 and total <60):
                     print("successful  CC",total)
         else :
             print("failed",total)

          Med term quiz  :49
          Final quiz :49
         failed 49.0
```

Abdelrahman A. Mohamed