Dhirubhai Ambani University

(Formerly known as DA-IICT)

Topic: Python Flow Control

Course: Programming Lab

Course Code- PC503

Dr. Ankit Vijayvargiya

Assistant Professor

Room No. 4205, Faculty Block 4

Email: ankit_Vijayvargiya[at]dau.ac.in

Phone: 079-68261628(O), 7877709590(M)

Dhirubhai Ambani University
Formerly known as DA-IICT

Python Flow Control

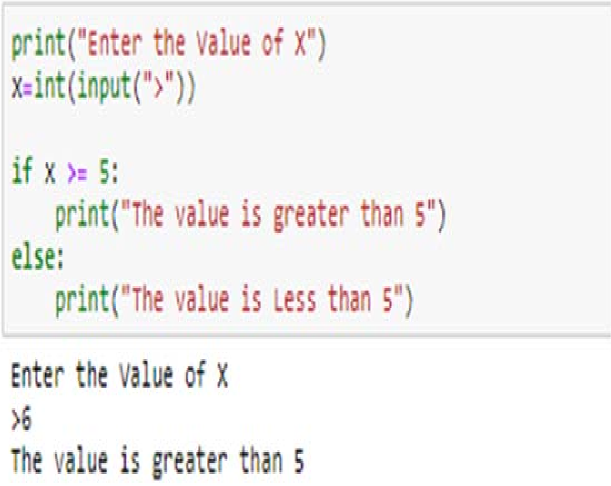
- if….else
- for loop
- while loop
- break and continue

Dhirubhai Ambani University
Formerly known as
DA-IICT

if……else

- The if else statement is used in Python for decision making.

```
if test expression:
        body of if
else:
        body of else
```

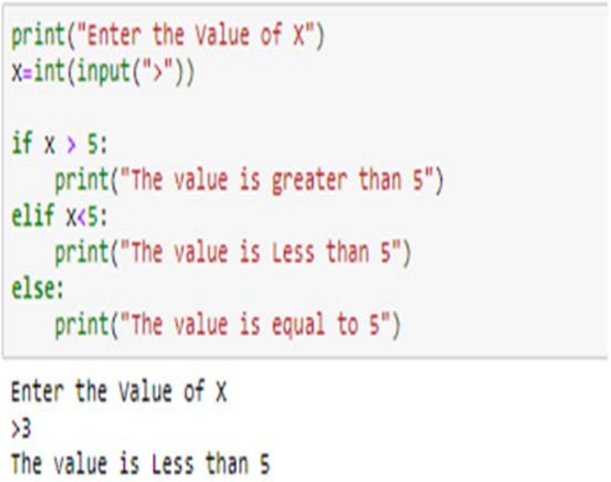


```
print("Enter the Value of X")
x=int(input(">"))

if x >= 5:
    print("The value is greater than 5")
else:
    print("The value is Less than 5")
```

```
Enter the Value of X
>6
The value is greater than 5
```

- To test more than one condition following syntax is used-

```
if test expression:
        body of if
elif test expression:
        body of elif
else:
        body of else
```

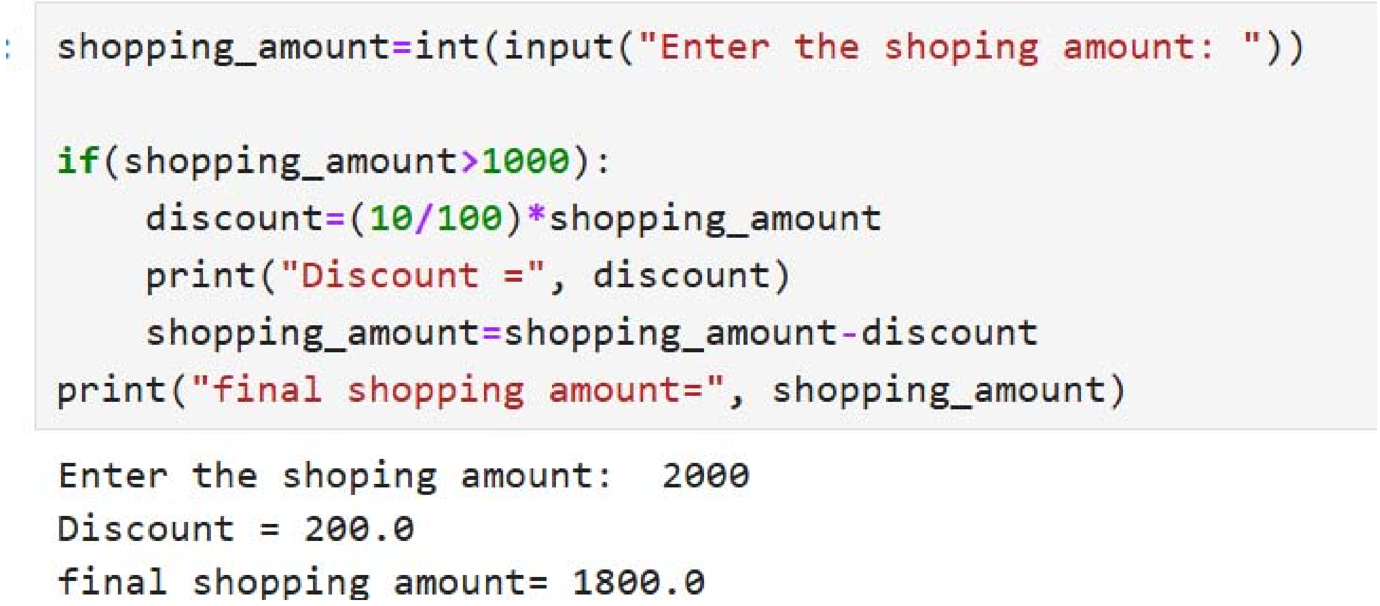


```
print("Enter the Value of X")
x=int(input(">"))

if x > 5:
    print("The value is greater than 5")
elif x<5:
    print("The value is Less than 5")
else:
    print("The value is equal to 5")
```

```
Enter the Value of X
>3
The value is Less than 5
```

Exercise- 4

Write a program to give a discount of 10% if the total bill amount exceeds 1000 Rs.



```
shopping_amount=int(input("Enter the shoping amount: "))

if(shopping_amount>1000):
    discount=(10/100)*shopping_amount
    print("Discount =", discount)
    shopping_amount=shopping_amount-discount
print("final shopping amount=", shopping_amount)
```

```
Enter the shoping amount:  2000
Discount = 200.0
final shopping amount= 1800.0
```

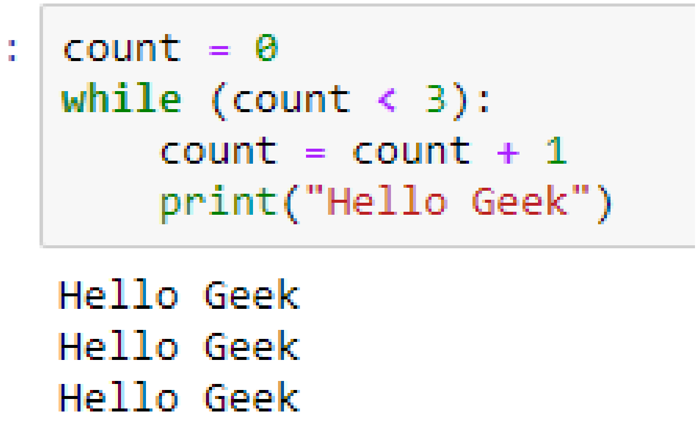
While loop

- A while loop is used to execute 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.
- We generally use this loop when we don't know beforehand the number of times to iterate.

Syntax :

while expression:

Body of while

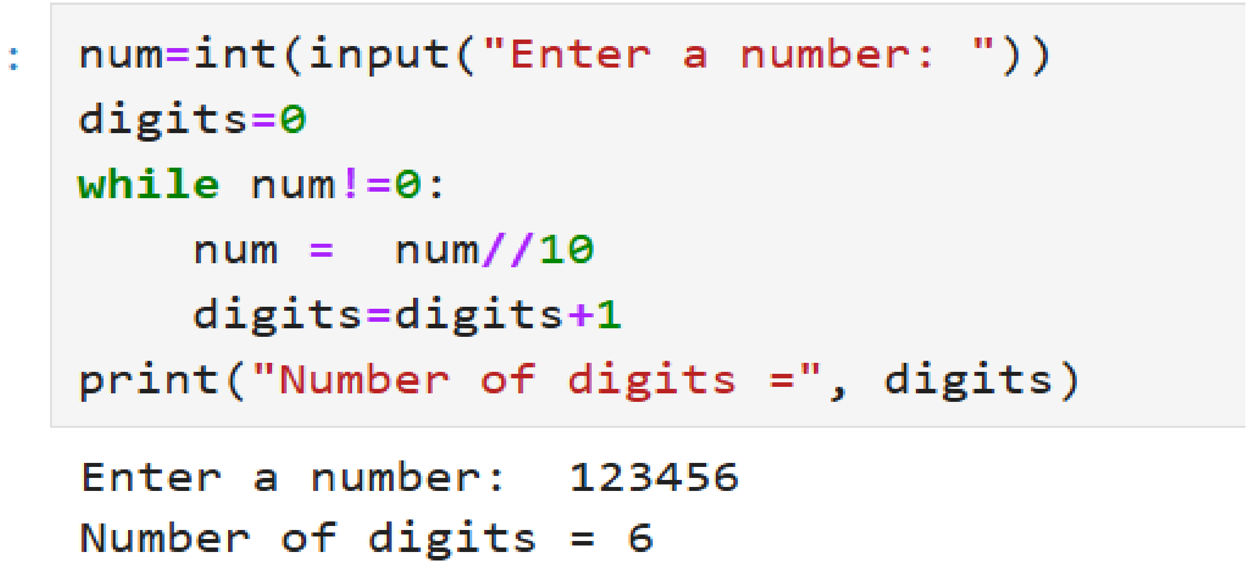


```
count = 0
while (count < 3):
    count = count + 1
    print("Hello Geek")
```

```
Hello Geek
Hello Geek
Hello Geek
```

Exercise- 5

Write a program to print the number of digits of a given number using while loop.



```
num=int(input("Enter a number: "))
digits=0
while num!=0:
    num =  num//10
    digits=digits+1
print("Number of digits =", digits)
```

```
Enter a number:  123456
Number of digits = 6
```

For loop

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

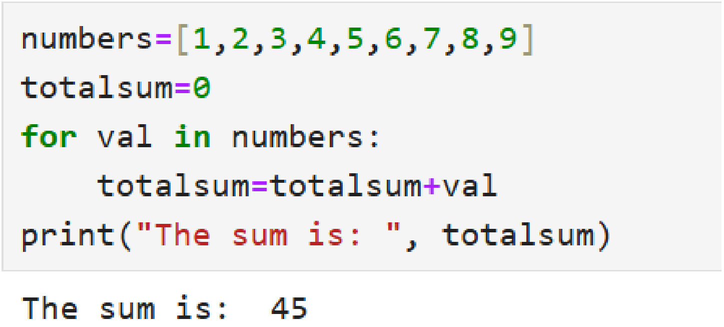
  Syntax :

  for val in sequence:

  Body of for

- Here, val is the variable that takes the value of the item inside the sequence on each iteration.
- Loop continues until we reach the last item in the sequence.

Program to find the sum of all numbers stored in a list.



```
numbers=[1,2,3,4,5,6,7,8,9]
totalsum=0
for val in numbers:
    totalsum=totalsum+val
print("The sum is: ", totalsum)
```

```
The sum is:  45
```

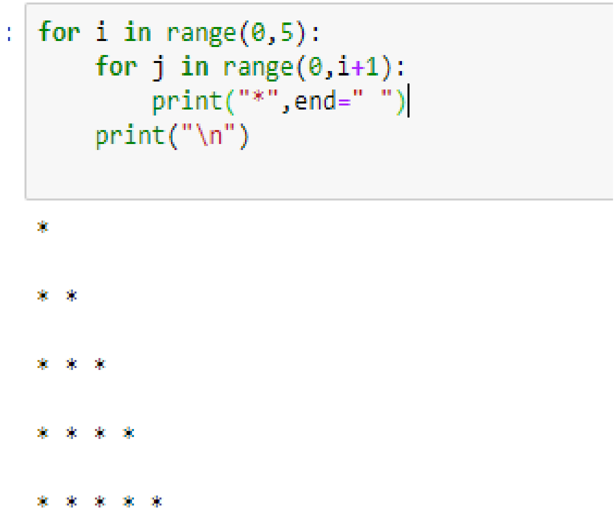
Range Function

- We can generate the sequence of numbers using the range() function.
- Range(10) will generate numbers from 0 to 9.

Exercise- 6

Print the following pattern

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

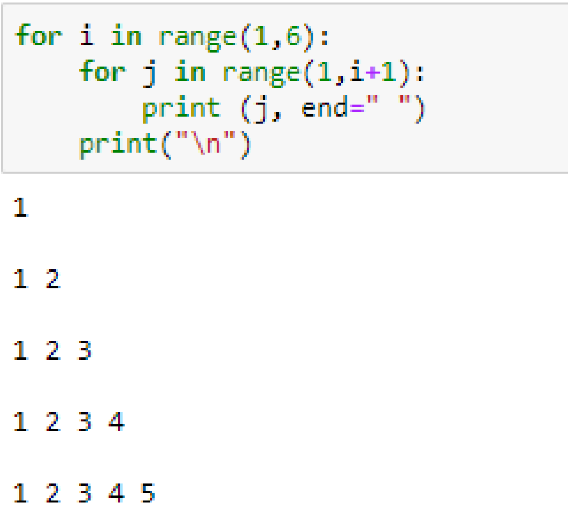


```
for i in range(0,5):
    for j in range(0,i+1):
        print("*",end=" ")
    print("\n")
```

```
*

* *

* * *

* * * *

* * * * *
```

Print the following pattern

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



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

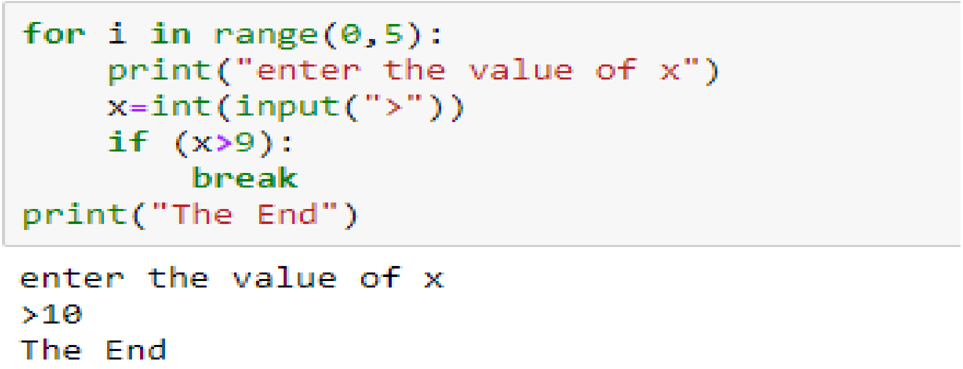
```
1

1 2

1 2 3

1 2 3 4

1 2 3 4 5
```

break and continue

- ❖ Loop iterates over a block of code until the text expression is false, but sometimes we wish to terminate the current iteration or even the whole loop without checking text expression.

Break:

- The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.
- If break statement is inside a nested loop, break will terminate the inner most loop.

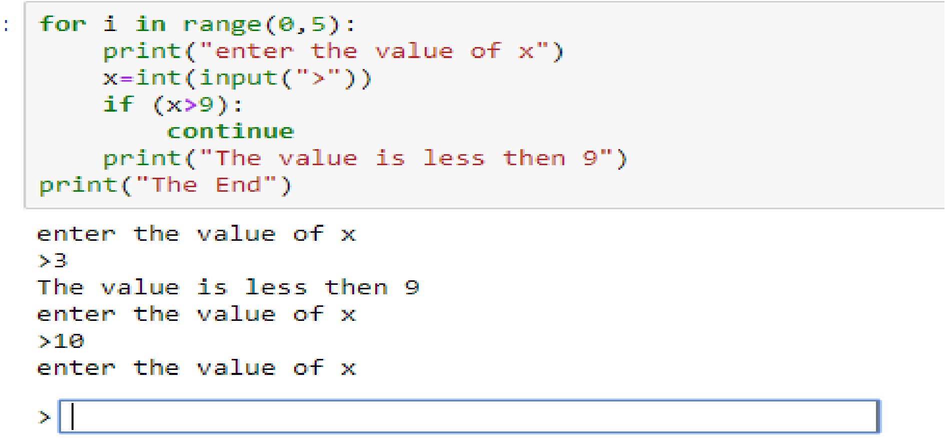


```
for i in range(0,5):
    print("enter the value of x")
    x=int(input(">"))
    if (x>9):
        break
print("The End")
```

```
enter the value of x
>10
The End
```

Continue:

- The continue statement is used to skip the rest of the code inside the loop for the current iteration only.
- Loop does not terminate but continues with the next iteration.



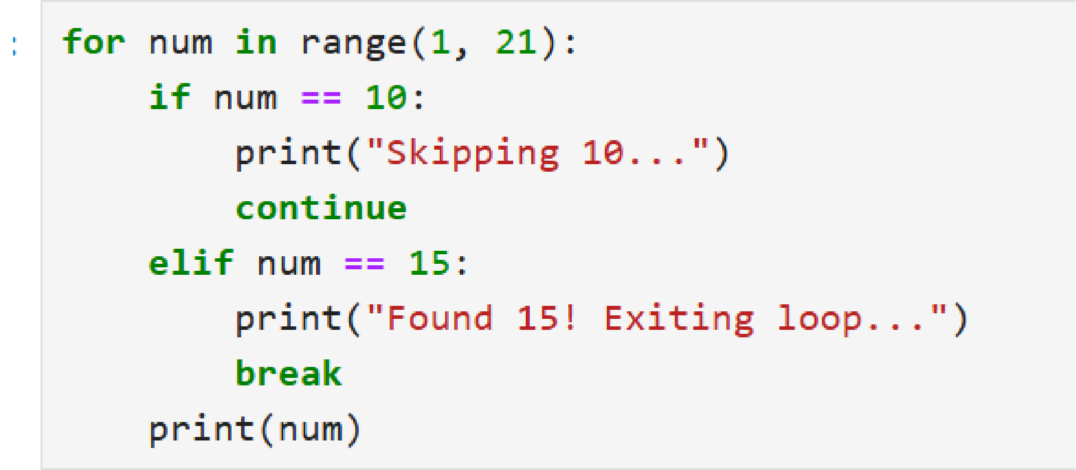
```
for i in range(0,5):
    print("enter the value of x")
    x=int(input(">"))
    if (x>9):
        continue
    print("The value is less then 9")
print("The End")
```

```
enter the value of x
>3
The value is less then 9
enter the value of x
>10
enter the value of x

>
```

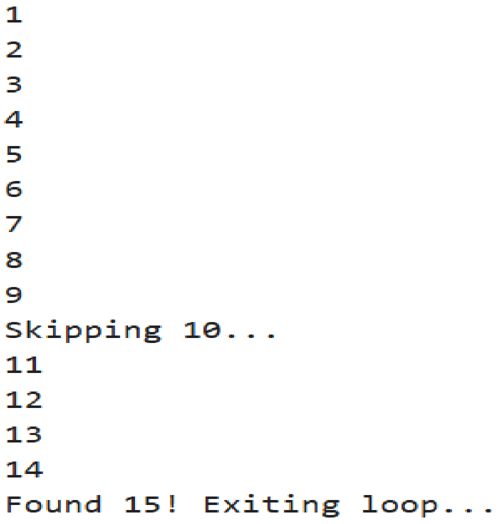
Exercise- 7

Write a program that loops through numbers from 1 to 20. Print each number, but:

- If the number is 10, skip it using continue.
- If the number is 15, exit the loop using break.



```
for num in range(1, 21):
    if num == 10:
        print("Skipping 10...")
        continue
    elif num == 15:
        print("Found 15! Exiting loop...")
        break
    print(num)
```



```
1
2
3
4
5
6
7
8
9
Skipping 10...
11
12
13
14
Found 15! Exiting loop...
```

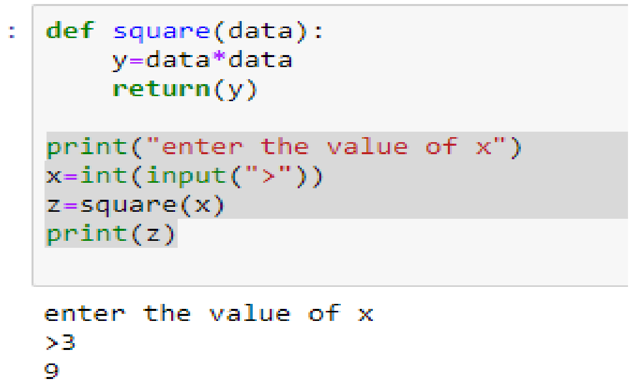
Function

- ➢ In Python, a function is a group of related statements that perform a specific task.

Syntax:

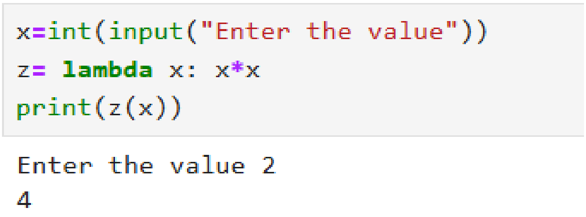
```
def  functionname(parameters):
          statements
```

- ➢ Keyword def marks the start of the function header.
- ➢ Parameters through which we pass a value to a function. They are optional.
- ➢ An optional return.



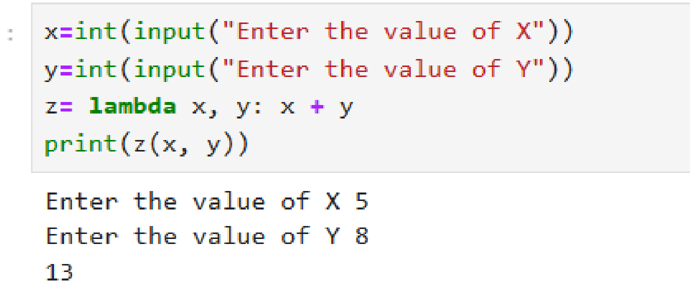
```
def square(data):
    y=data*data
    return(y)

print("enter the value of x")
x=int(input(">"))
z=square(x)
print(z)
```

```
enter the value of x
>3
9
```



```
x=int(input("Enter the value"))
z= lambda x: x*x
print(z(x))
```

```
Enter the value 2
4
```

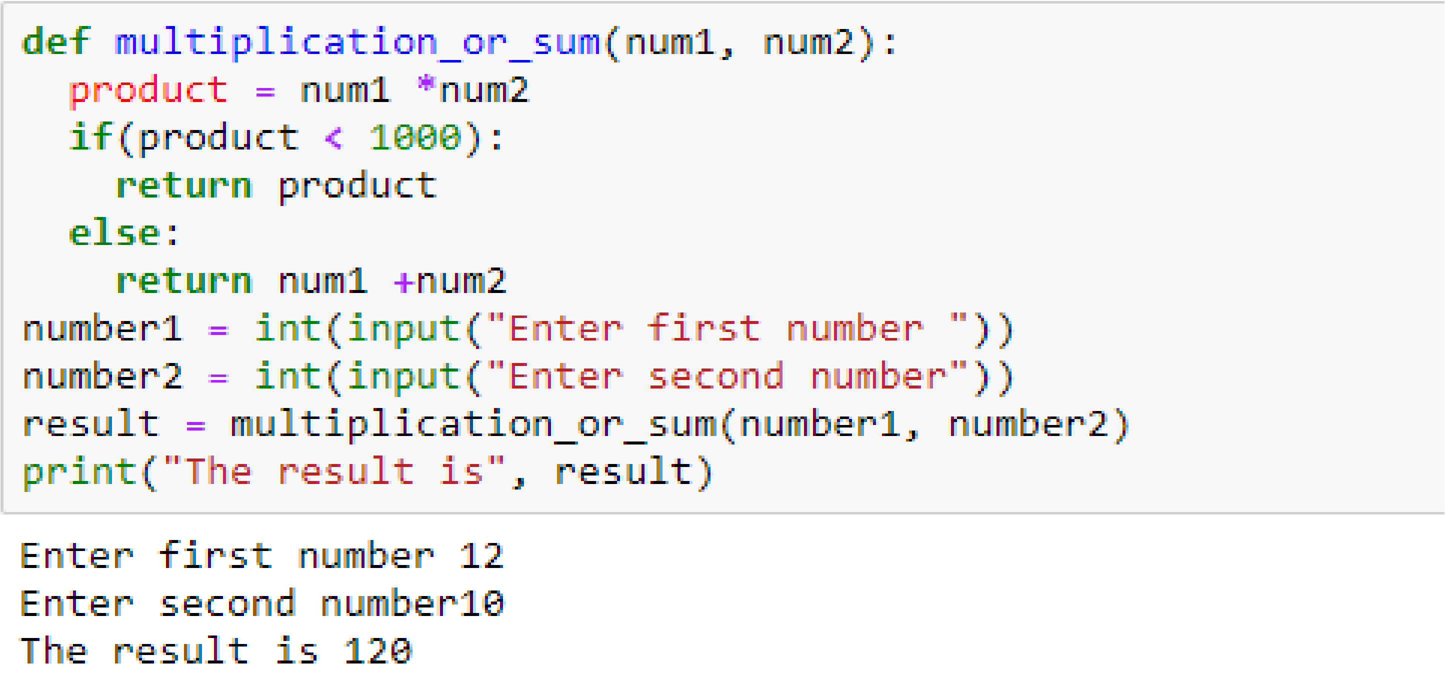


```
x=int(input("Enter the value of X"))
y=int(input("Enter the value of Y"))
z= lambda x, y: x + y
print(z(x, y))
```

```
Enter the value of X 5
Enter the value of Y 8
13
```

Exercise- 8

Write a program to accept two integer values from the user and return their product. If the product is greater than 1000, then return their sum.



```
def multiplication_or_sum(num1, num2):
  product = num1 *num2
  if(product < 1000):
    return product
  else:
    return num1 +num2
number1 = int(input("Enter first number "))
number2 = int(input("Enter second number"))
result = multiplication_or_sum(number1, number2)
print("The result is", result)
```

```
Enter first number 12
Enter second number10
The result is 120
```