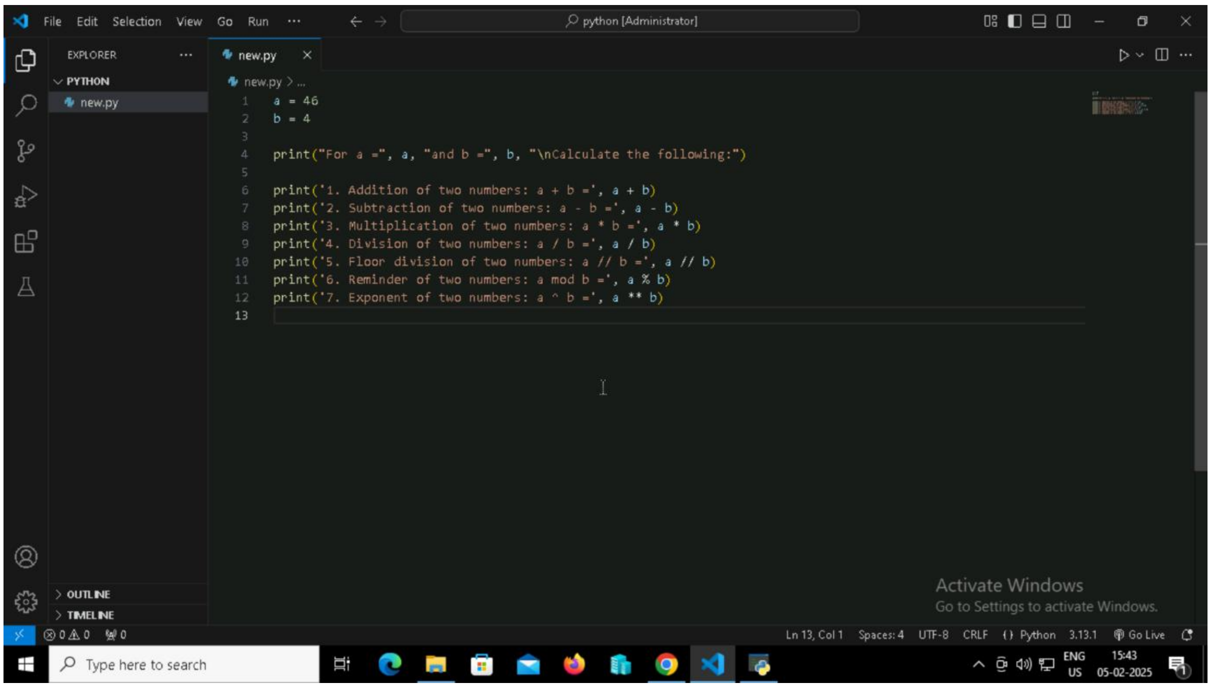
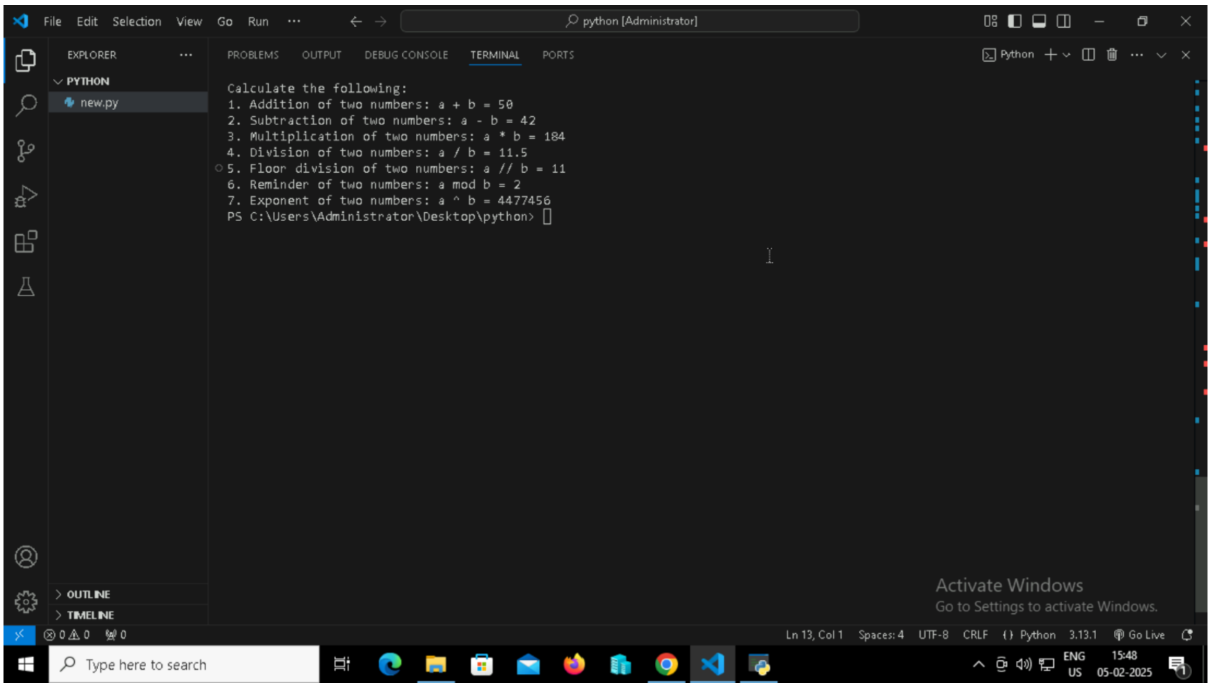
PYTHON OPERATORS

The **Operators** are the symbols used to perform a specific operation on different values and variables.

1. These code examples of arithmetic operators in Python:

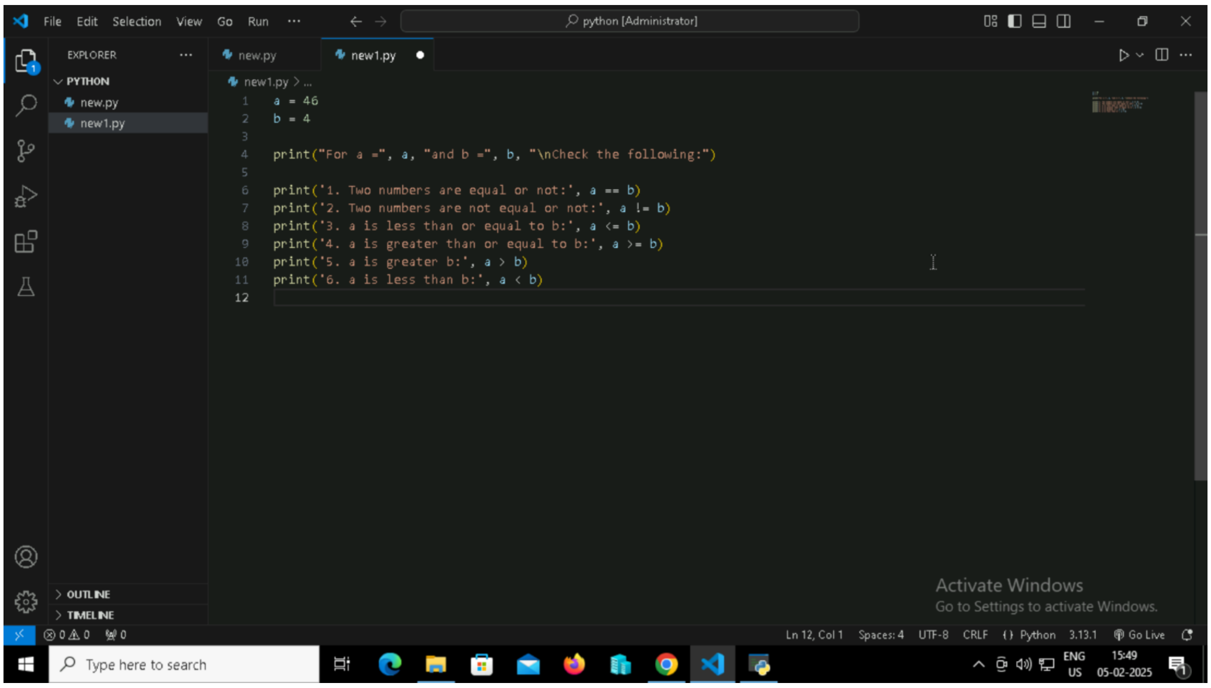


```python
a = 46
b = 4

print("For a =", a, "and b =", b, "\nCalculate the following:")

print('1. Addition of two numbers: a + b =', a + b)
print('2. Subtraction of two numbers: a - b =', a - b)
print('3. Multiplication of two numbers: a * b =', a * b)
print('4. Division of two numbers: a / b =', a / b)
print('5. Floor division of two numbers: a // b =', a // b)
print('6. Reminder of two numbers: a mod b =', a % b)
print('7. Exponent of two numbers: a ^ b =', a ** b)
```

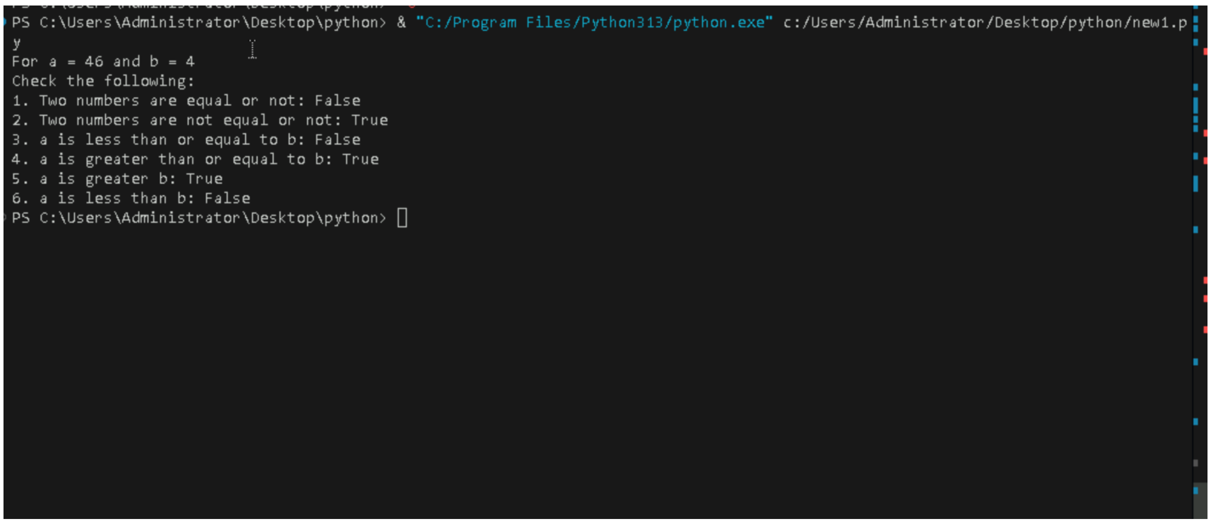


```
Calculate the following:
1. Addition of two numbers: a + b = 50
2. Subtraction of two numbers: a - b = 42
3. Multiplication of two numbers: a * b = 184
4. Division of two numbers: a / b = 11.5
5. Floor division of two numbers: a // b = 11
6. Reminder of two numbers: a mod b = 2
7. Exponent of two numbers: a ^ b = 4477456
PS C:\Users\Administrator\Desktop\python>
```

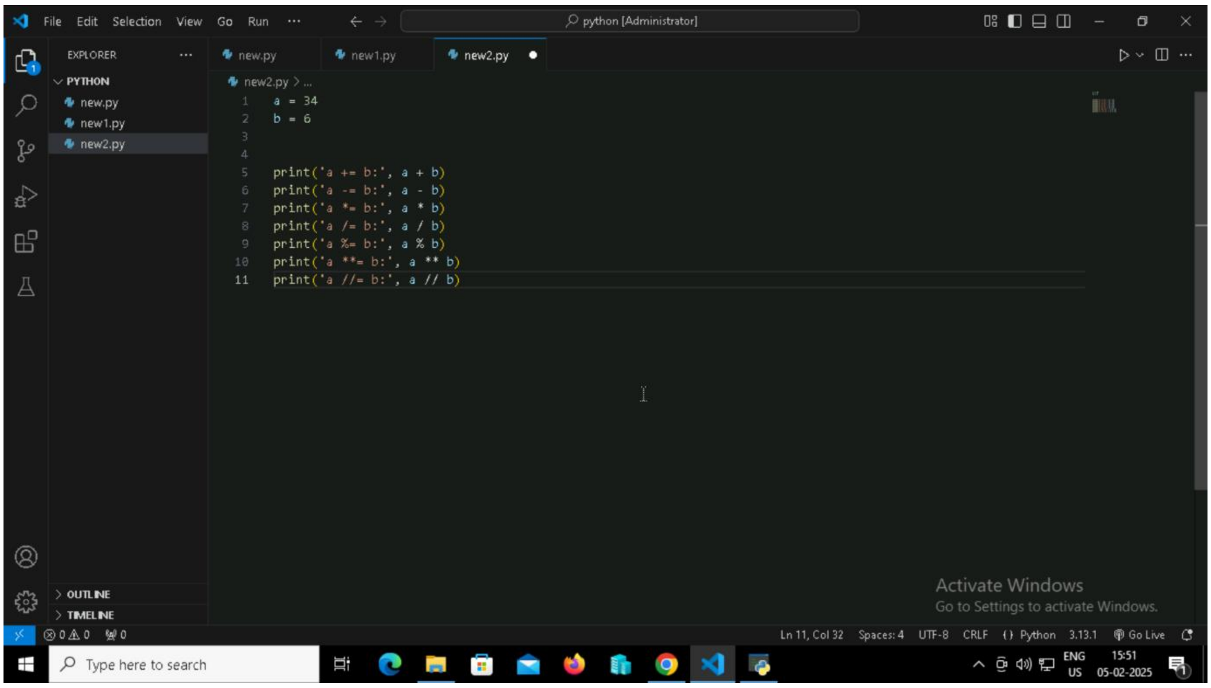
2. code examples of Comparison operators in Python:



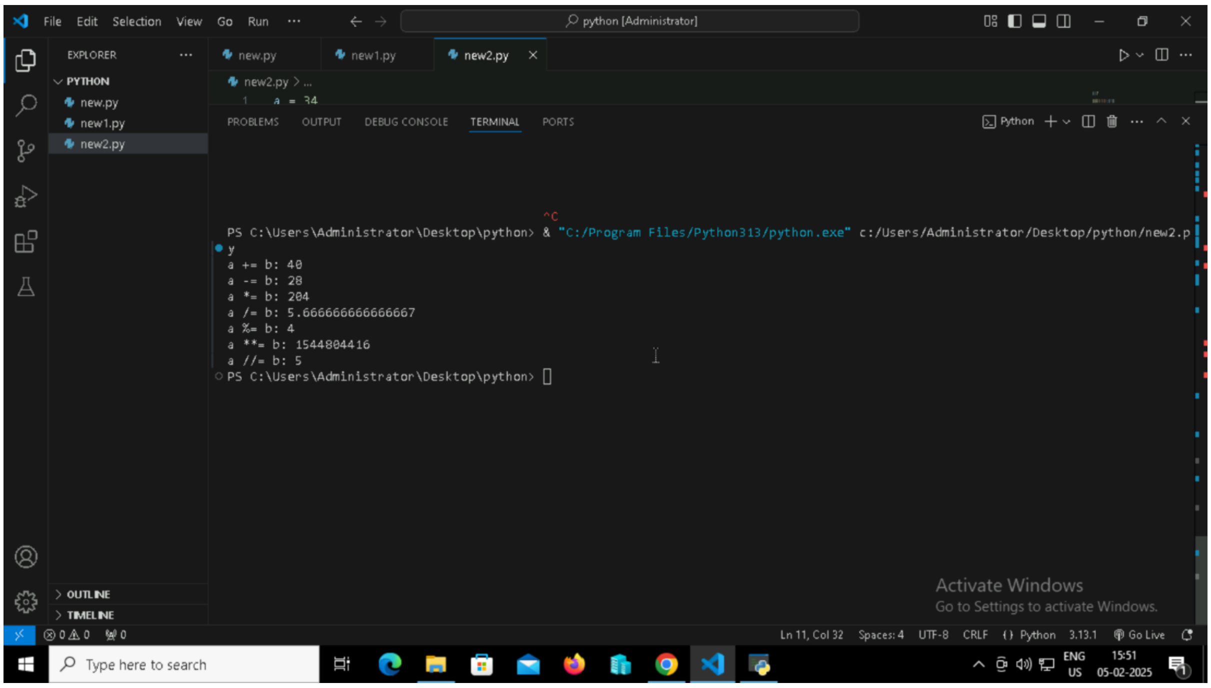
```python
a = 46
b = 4

print("For a =", a, "and b =", b, "\nCheck the following:")

print('1. Two numbers are equal or not:', a == b)
print('2. Two numbers are not equal or not:', a != b)
print('3. a is less than or equal to b:', a <= b)
print('4. a is greater than or equal to b:', a >= b)
print('5. a is greater b:', a > b)
print('6. a is less than b:', a < b)
```



```
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python313/python.exe" c:/Users/Administrator/Desktop/python/new1.p
y
For a = 46 and b = 4
Check the following:
1. Two numbers are equal or not: False
2. Two numbers are not equal or not: True
3. a is less than or equal to b: False
4. a is greater than or equal to b: True
5. a is greater b: True
6. a is less than b: False
PS C:\Users\Administrator\Desktop\python>
```

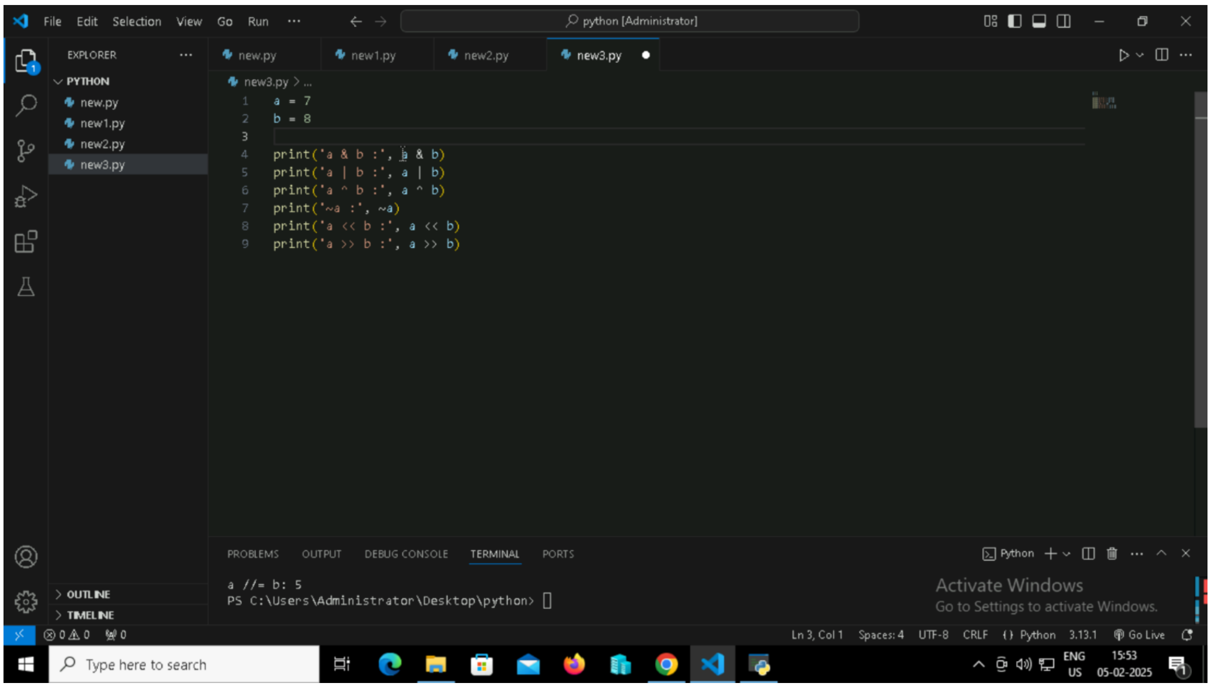

File Edit Selection View Go Run
python [Administrator]
EXPLORER
PYTHON
new.py
new1.py
new2.py
OUTLINE
TIMELINE
Activate Windows
Go to Settings to activate Windows.
Ln 11, Col 32 Spaces: 4 UTF-8 CRLF Python 3.13.1 Go Live
Type here to search
15:51
05-02-2025

```python
a = 34
b = 6


print('a += b:', a + b)
print('a -= b:', a - b)
print('a *= b:', a * b)
print('a /= b:', a / b)
print('a %= b:', a % b)
print('a **= b:', a ** b)
print('a //= b:', a // b)
```


File Edit Selection View Go Run
python [Administrator]
EXPLORER
PYTHON
new.py
new1.py
new2.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
OUTLINE
TIMELINE
Activate Windows
Go to Settings to activate Windows.
Ln 11, Col 32 Spaces: 4 UTF-8 CRLF Python 3.13.1 Go Live
Type here to search
15:51
05-02-2025

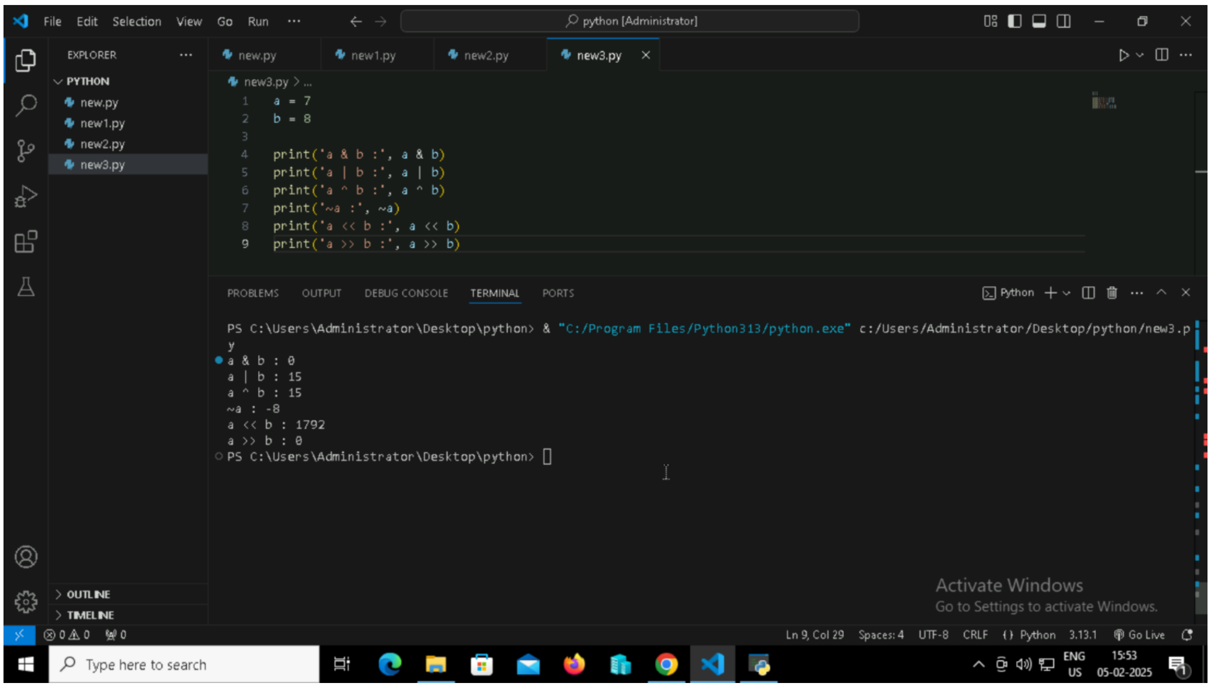
```
^C
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python313/python.exe" c:/Users/Administrator/Desktop/python/new2.p
y
a += b: 40
a -= b: 28
a *= b: 204
a /= b: 5.666666666666667
a %= b: 4
a **= b: 1544804416
a //= b: 5
PS C:\Users\Administrator\Desktop\python>
```

4. code examples of Logical Operators in python:



```python
a = 7
b = 8

print('a & b :', a & b)
print('a | b :', a | b)
print('a ^ b :', a ^ b)
print('~a :', ~a)
print('a << b :', a << b)
print('a >> b :', a >> b)
```

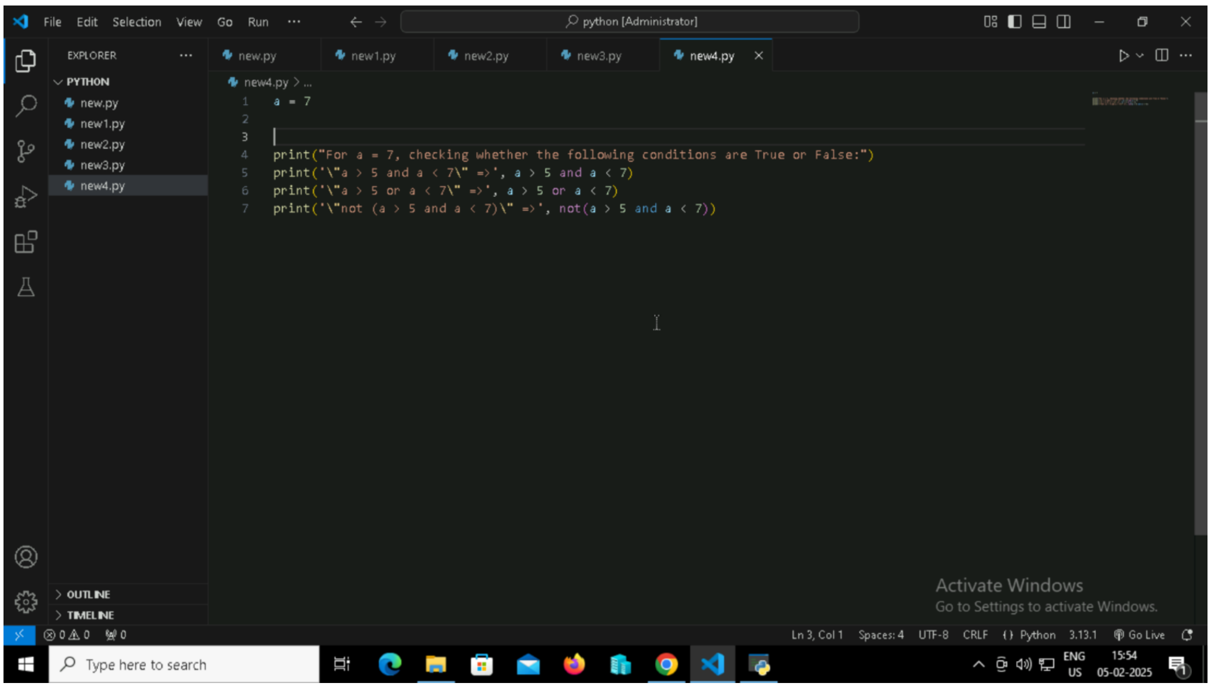
```
a //= b: 5
PS C:\Users\Administrator\Desktop\python> 
```



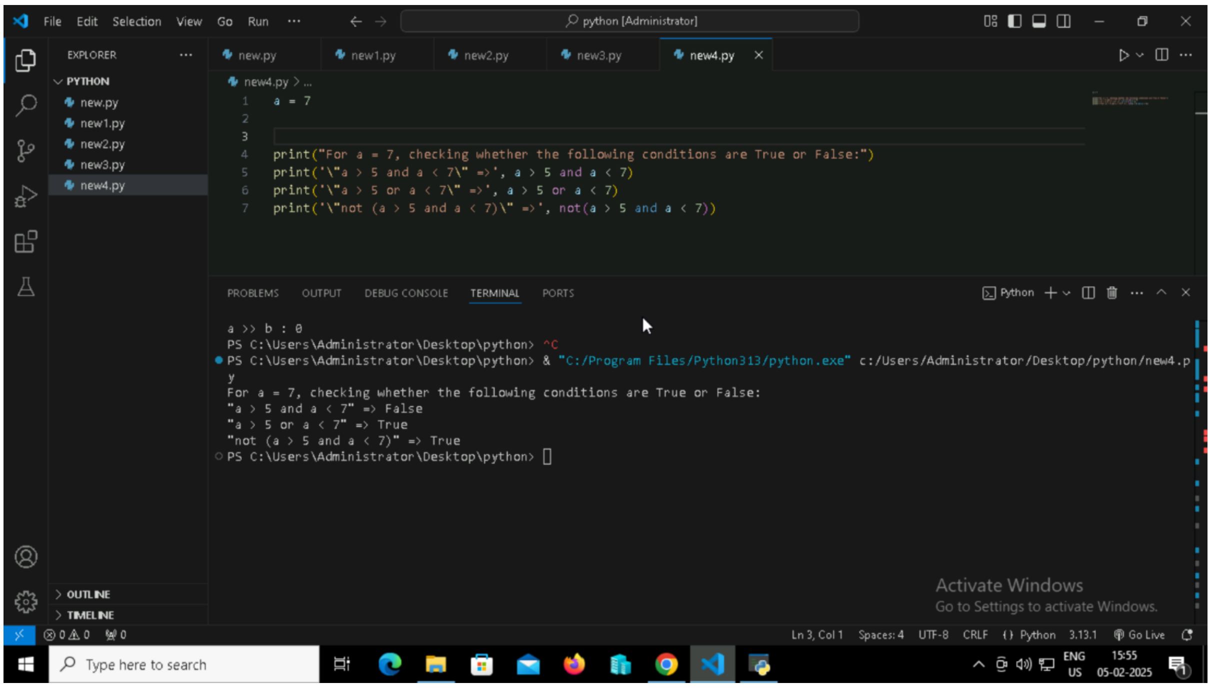
```python
a = 7
b = 8

print('a & b :', a & b)
print('a | b :', a | b)
print('a ^ b :', a ^ b)
print('~a :', ~a)
print('a << b :', a << b)
print('a >> b :', a >> b)
```

```
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python313/python.exe" c:/Users/Administrator/Desktop/python/new3.p
y
a & b : 0
a | b : 15
a ^ b : 15
~a : -8
a << b : 1792
a >> b : 0
PS C:\Users\Administrator\Desktop\python> 
```

5. code examples of **Bitwise** Operators in python:



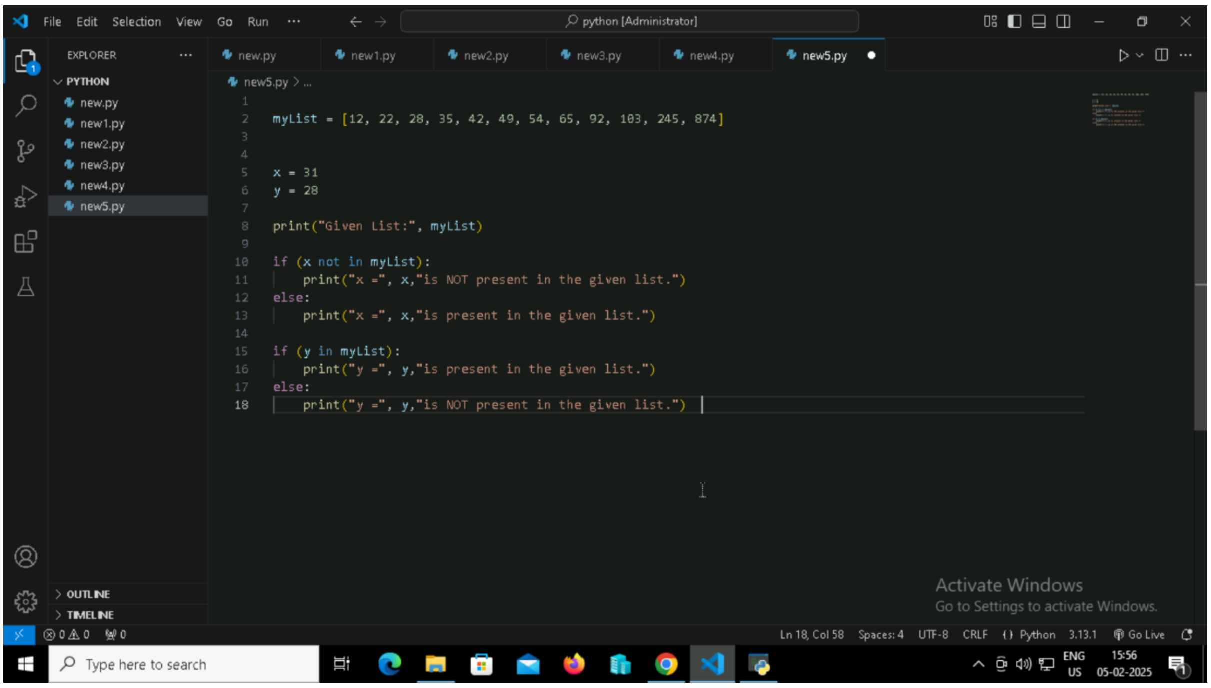
```python
a = 7


print("For a = 7, checking whether the following conditions are True or False:")
print('\"a > 5 and a < 7\" =>', a > 5 and a < 7)
print('\"a > 5 or a < 7\" =>', a > 5 or a < 7)
print('\"not (a > 5 and a < 7)\" =>', not(a > 5 and a < 7))
```



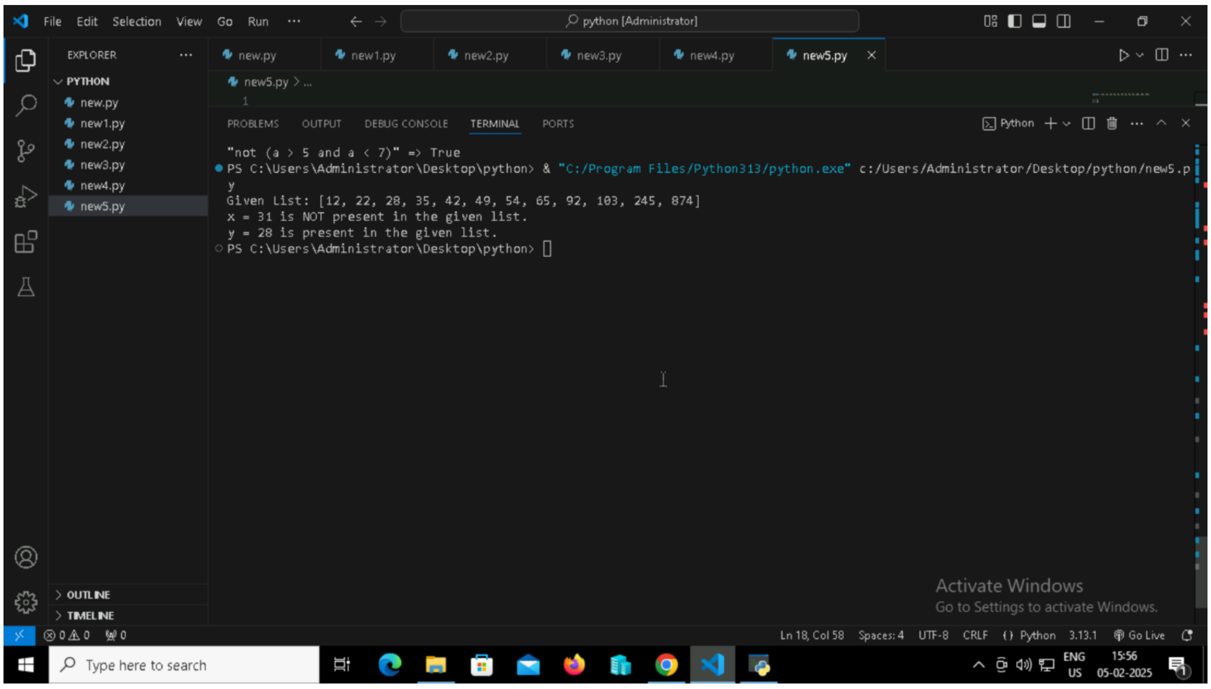
```python
a = 7


print("For a = 7, checking whether the following conditions are True or False:")
print('\"a > 5 and a < 7\" =>', a > 5 and a < 7)
print('\"a > 5 or a < 7\" =>', a > 5 or a < 7)
print('\"not (a > 5 and a < 7)\" =>', not(a > 5 and a < 7))
```

```
a >> b : 0
PS C:\Users\Administrator\Desktop\python> ^C
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python313/python.exe" c:/Users/Administrator/Desktop/python/new4.p
y
For a = 7, checking whether the following conditions are True or False:
"a > 5 and a < 7" => False
"a > 5 or a < 7" => True
"not (a > 5 and a < 7)" => True
PS C:\Users\Administrator\Desktop\python> 
```

6. code examples of Membership Operators in python:

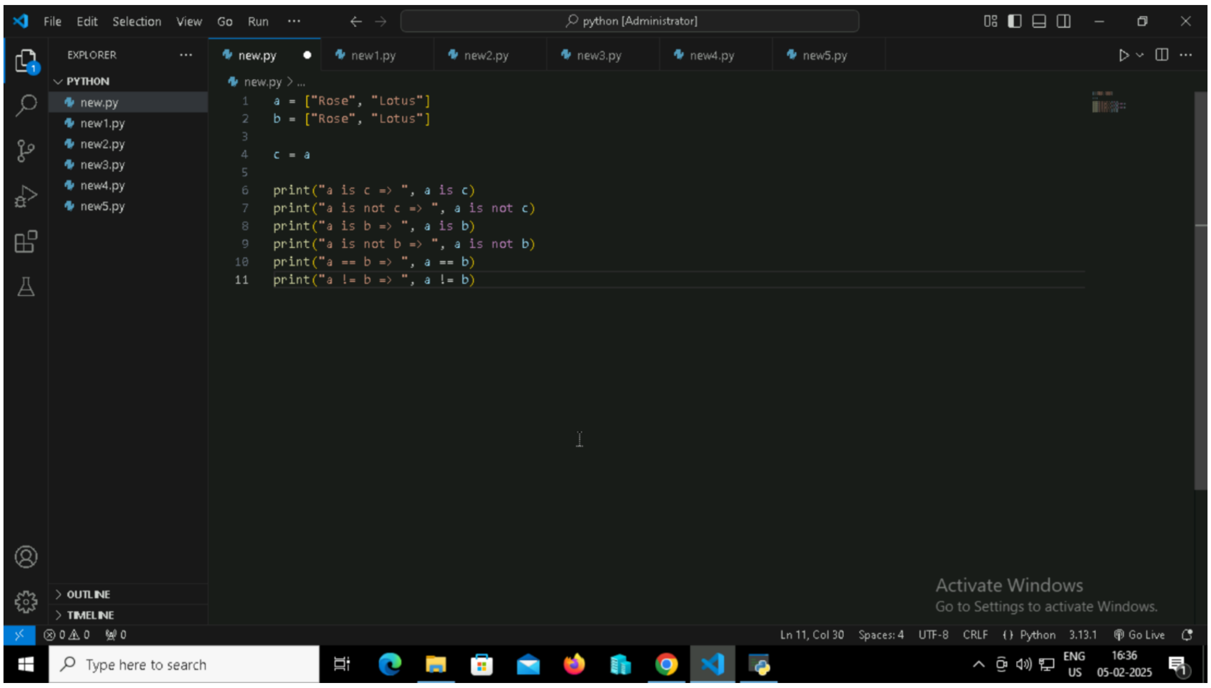


```python

myList = [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]


x = 31
y = 28

print("Given List:", myList)

if (x not in myList):
    print("x =", x,"is NOT present in the given list.")
else:
    print("x =", x,"is present in the given list.")

if (y in myList):
    print("y =", y,"is present in the given list.")
else:
    print("y =", y,"is NOT present in the given list.")
```

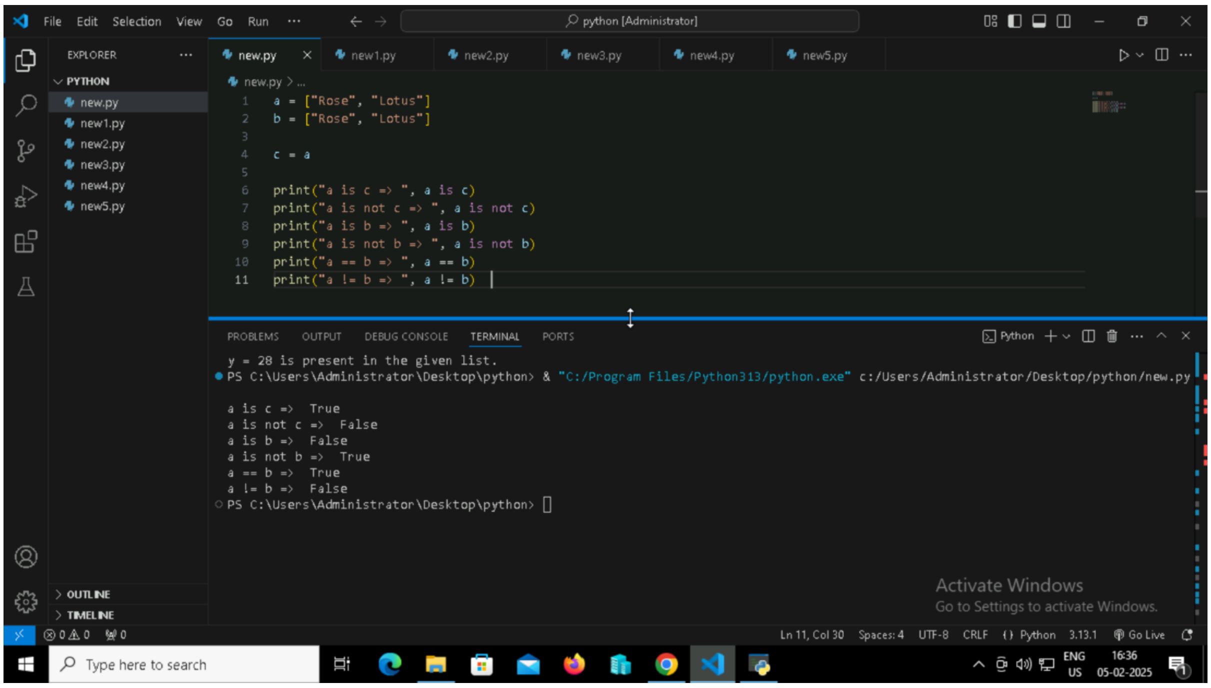


```
"not (a > 5 and a < 7)" => True
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python313/python.exe" c:/Users/Administrator/Desktop/python/new5.p
y
Given List: [12, 22, 28, 35, 42, 49, 54, 65, 92, 103, 245, 874]
x = 31 is NOT present in the given list.
y = 28 is present in the given list.
PS C:\Users\Administrator\Desktop\python>
```

7.. code examples of Identity Operators in python:



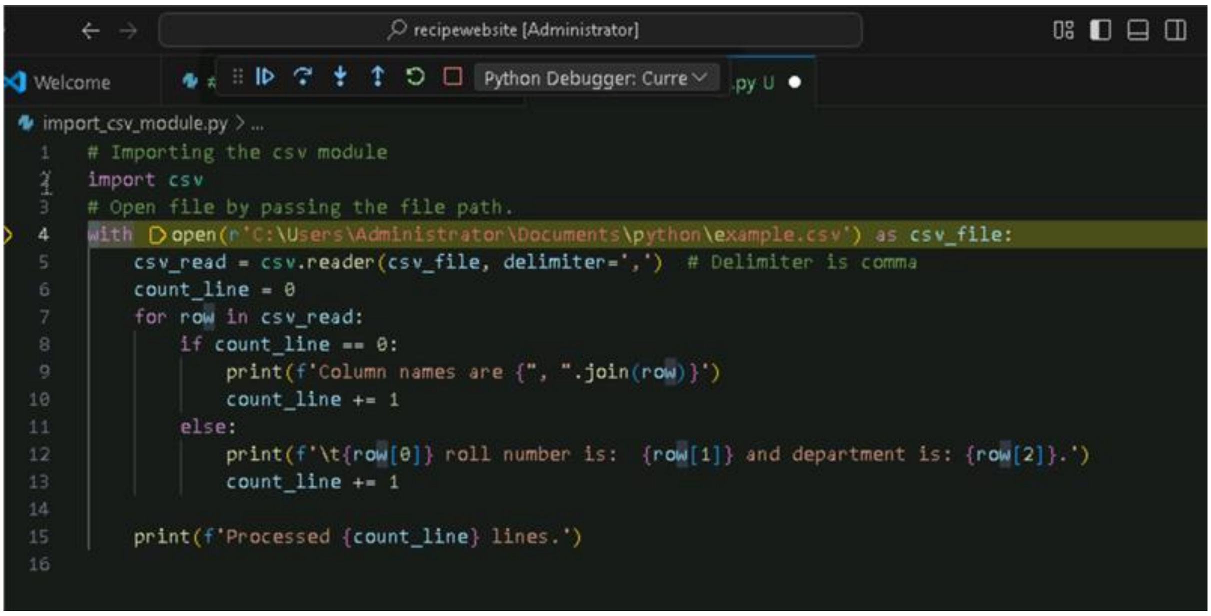
```python
a = ["Rose", "Lotus"]
b = ["Rose", "Lotus"]

c = a

print("a is c => ", a is c)
print("a is not c => ", a is not c)
print("a is b => ", a is b)
print("a is not b => ", a is not b)
print("a == b => ", a == b)
print("a != b => ", a != b)
```



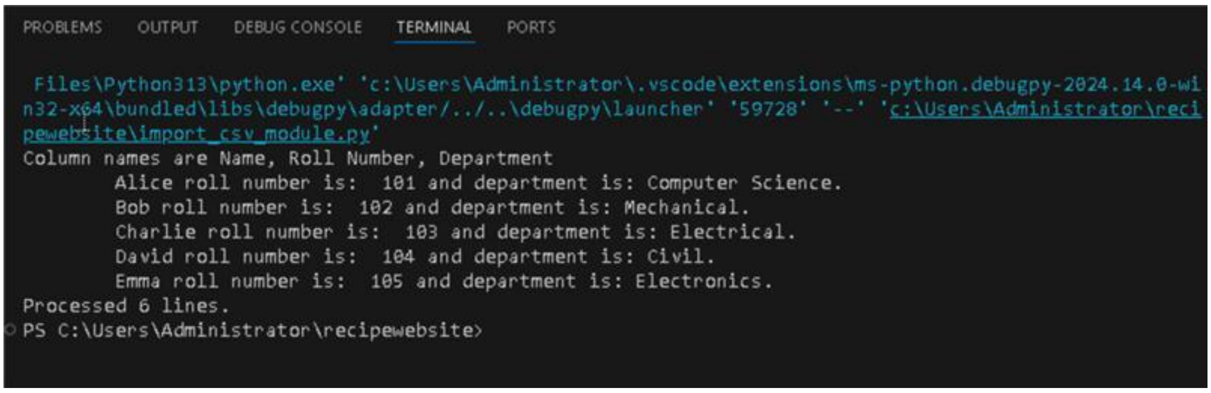
```python
a = ["Rose", "Lotus"]
b = ["Rose", "Lotus"]

c = a

print("a is c => ", a is c)
print("a is not c => ", a is not c)
print("a is b => ", a is b)
print("a is not b => ", a is not b)
print("a == b => ", a == b)
print("a != b => ", a != b)
```

```
y = 28 is present in the given list.
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python313/python.exe" c:/Users/Administrator/Desktop/python/new.py

a is c =>  True
a is not c =>  False
a is b =>  False
a is not b =>  True
a == b =>  True
a != b =>  False
PS C:\Users\Administrator\Desktop\python>
```

8. **To Read CSV file in Python**



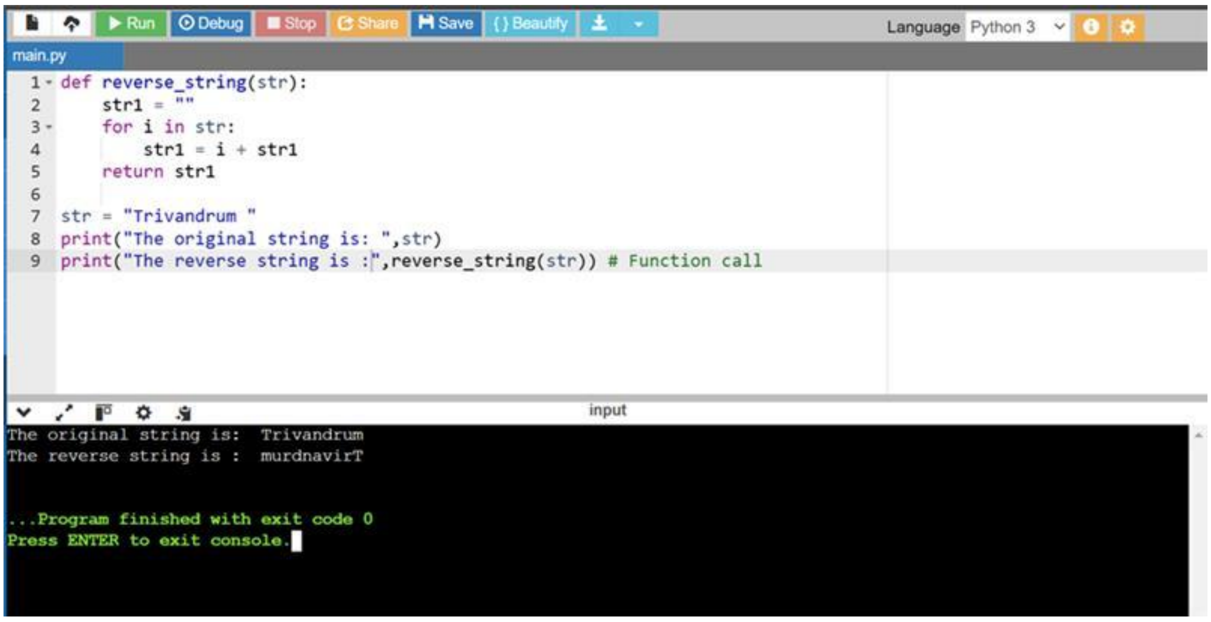
```python
# Importing the csv module
import csv
# Open file by passing the file path.
with open(r'C:\Users\Administrator\Documents\python\example.csv') as csv_file:
    csv_read = csv.reader(csv_file, delimiter=',')  # Delimiter is comma
    count_line = 0
    for row in csv_read:
        if count_line == 0:
            print(f'Column names are {", ".join(row)}')
            count_line += 1
        else:
            print(f'\t{row[0]} roll number is:  {row[1]} and department is: {row[2]}.')
            count_line += 1

    print(f'Processed {count_line} lines.')
```



```
 Files\Python313\python.exe' 'c:\Users\Administrator\.vscode\extensions\ms-python.debugpy-2024.14.0-wi
n32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '59728' '--' 'c:\Users\Administrator\reci
pewebsite\import_csv_module.py'
Column names are Name, Roll Number, Department
        Alice roll number is:  101 and department is: Computer Science.
        Bob roll number is:  102 and department is: Mechanical.
        Charlie roll number is:  103 and department is: Electrical.
        David roll number is:  104 and department is: Civil.
        Emma roll number is:  105 and department is: Electronics.
Processed 6 lines.
PS C:\Users\Administrator\recipewebsite>
```

REVERSE A STRING

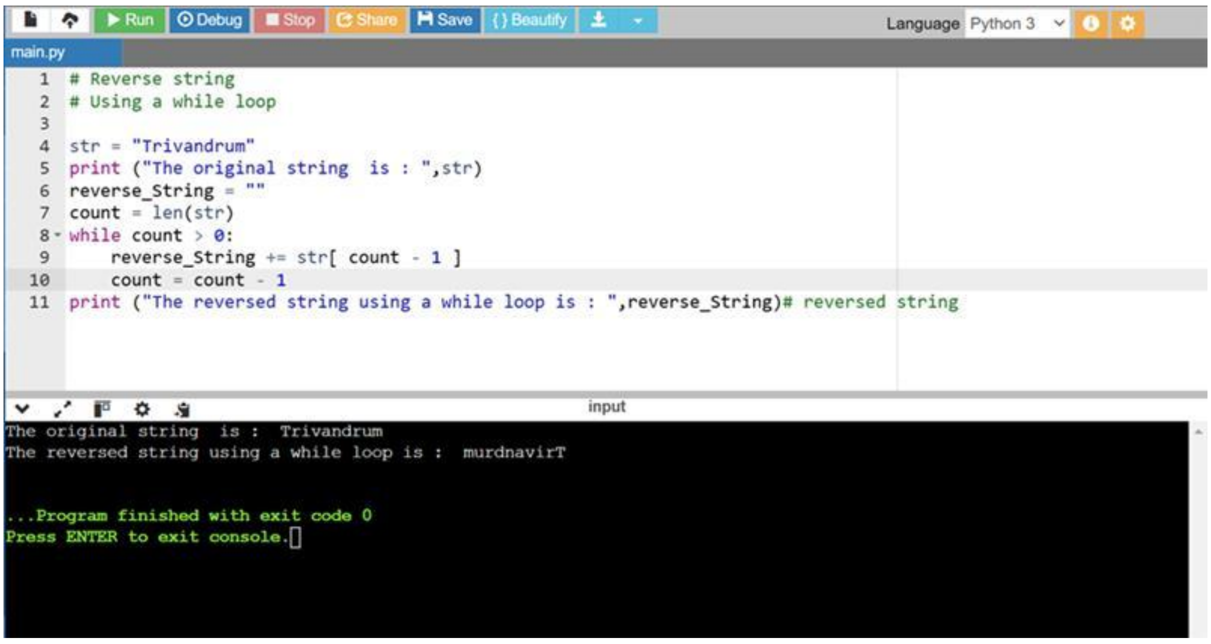
1. **Using FOR Loop**



```python
def reverse_string(str):
    str1 = ""
    for i in str:
        str1 = i + str1
    return str1

str = "Trivandrum "
print("The original string is: ",str)
print("The reverse string is :",reverse_string(str)) # Function call
```

```
The original string is:  Trivandrum
The reverse string is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

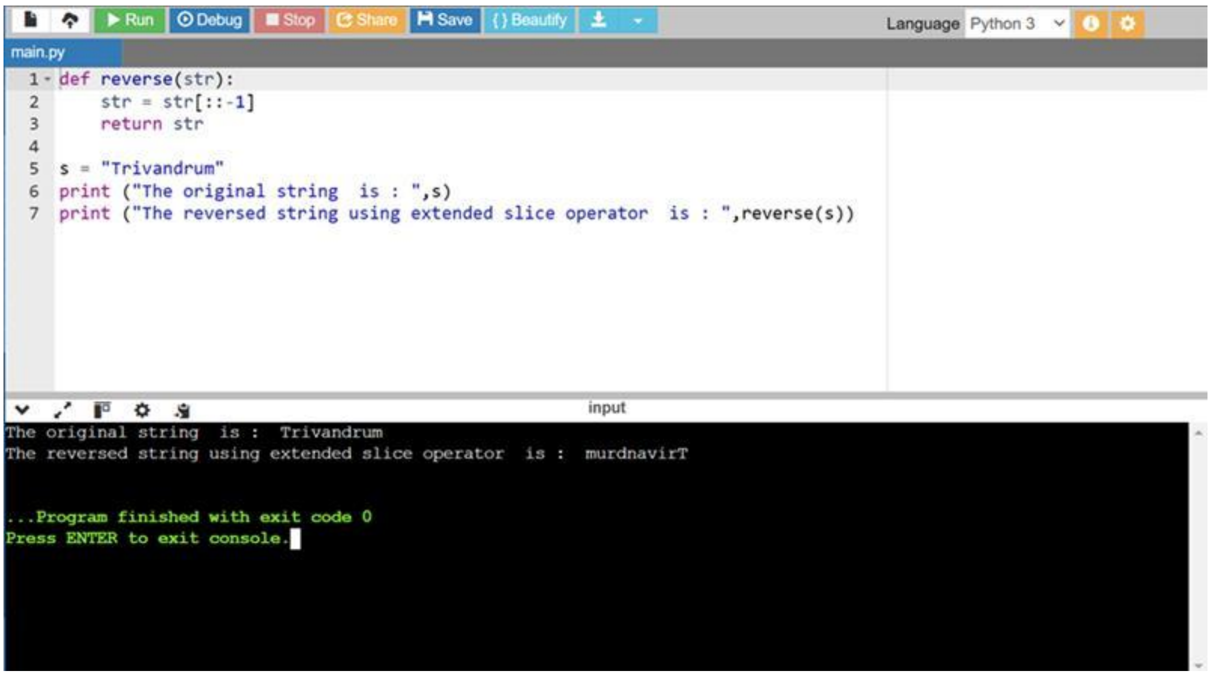
2. **Using WHILE Loop**



```python
# Reverse string
# Using a while loop

str = "Trivandrum"
print ("The original string  is : ",str)
reverse_String = ""
count = len(str)
while count > 0:
    reverse_String += str[ count - 1 ]
    count = count - 1
print ("The reversed string using a while loop is : ",reverse_String)# reversed string
```

```
The original string  is :  Trivandrum
The reversed string using a while loop is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

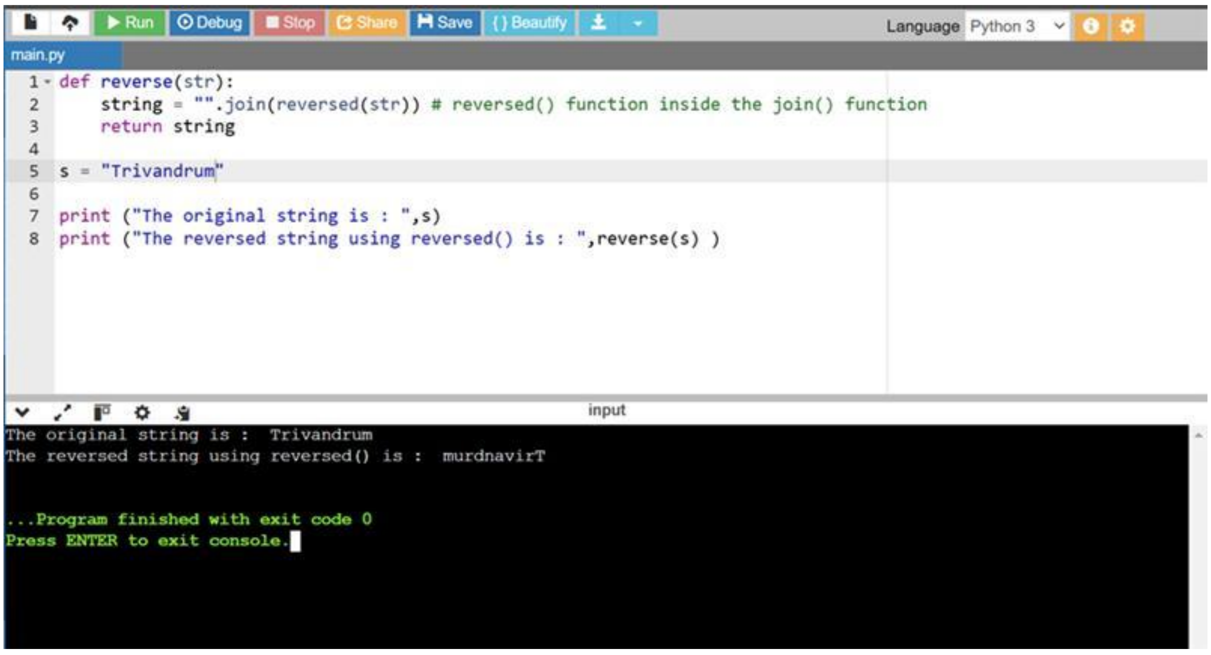
3. **Using the slice operator**



```python
def reverse(str):
    str = str[::-1]
    return str

s = "Trivandrum"
print ("The original string  is : ",s)
print ("The reversed string using extended slice operator  is : ",reverse(s))
```

```
The original string  is :  Trivandrum
The reversed string using extended slice operator  is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

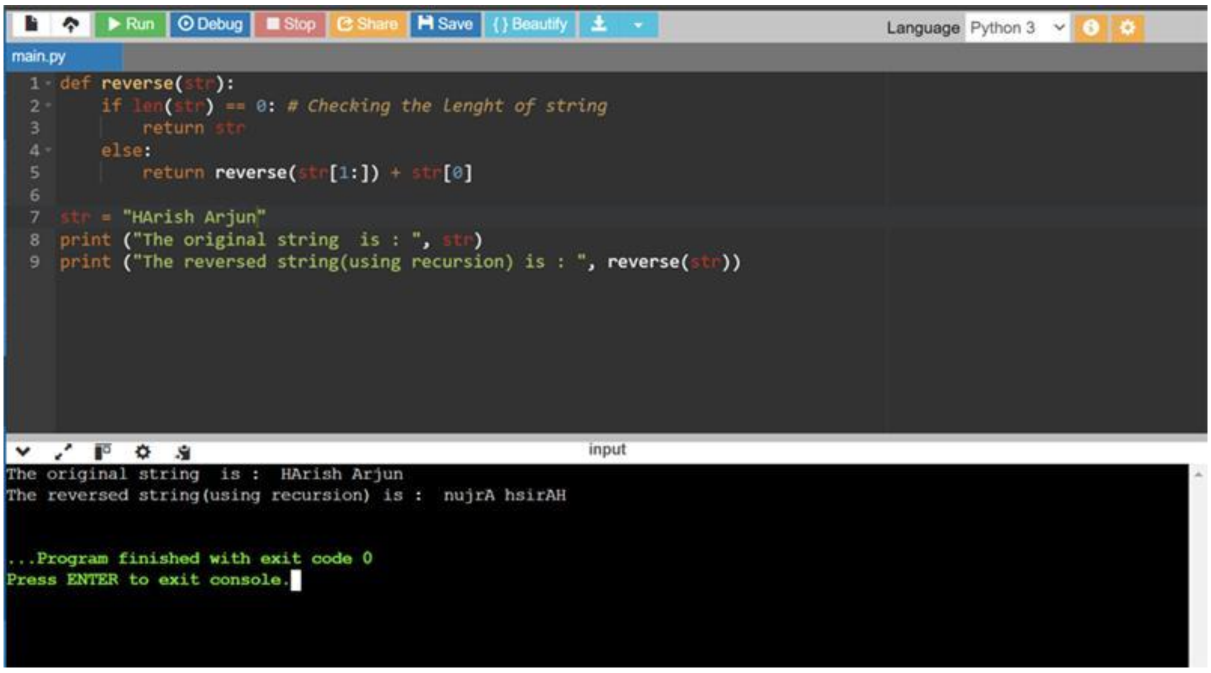
4. **Using the reverse () function**



```python
def reverse(str):
    string = "".join(reversed(str)) # reversed() function inside the join() function
    return string

s = "Trivandrum"

print ("The original string is : ",s)
print ("The reversed string using reversed() is : ",reverse(s) )
```

```
The original string is :  Trivandrum
The reversed string using reversed() is :  murdnavirT


...Program finished with exit code 0
Press ENTER to exit console.
```

5. Using the Recursion

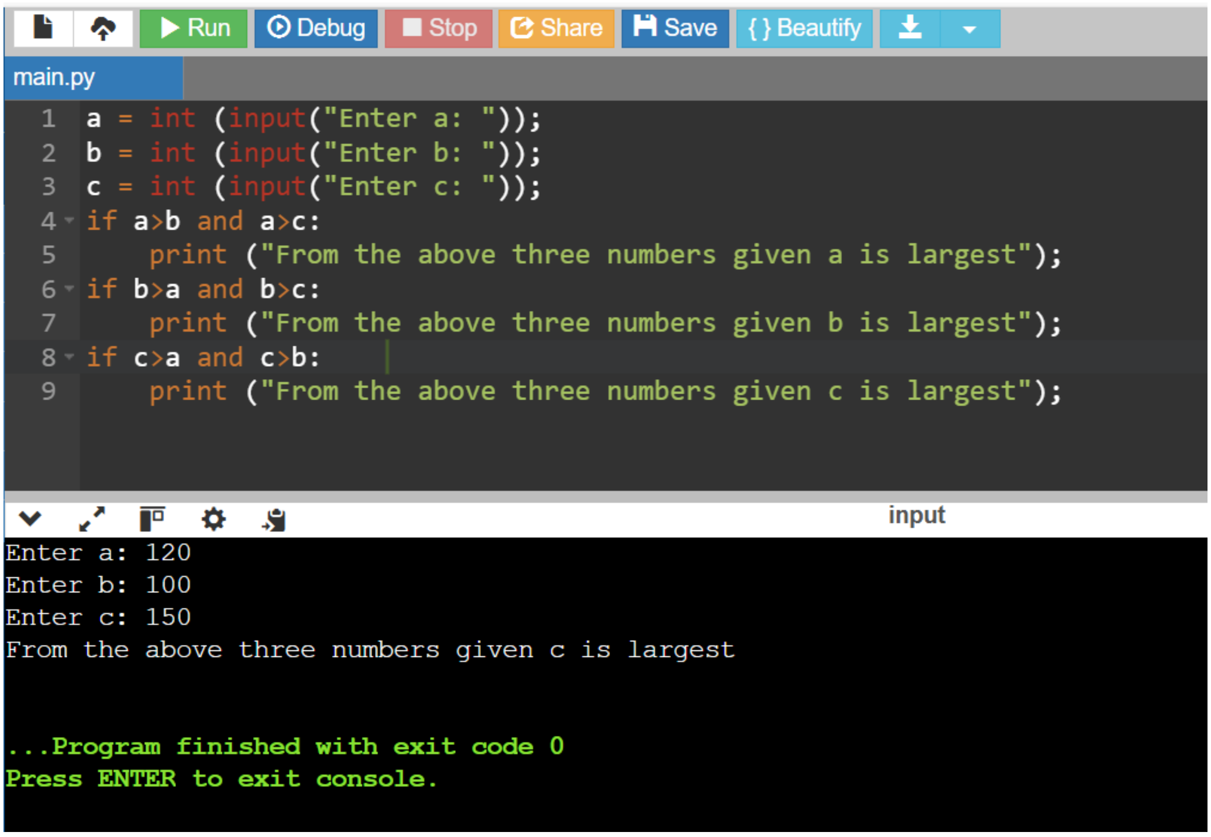


```python
def reverse(str):
    if len(str) == 0: # Checking the Lenght of string
        return str
    else:
        return reverse(str[1:]) + str[0]

str = "HArish Arjun"
print ("The original string  is : ", str)
print ("The reversed string(using recursion) is : ", reverse(str))
```

```
The original string  is :  HArish Arjun
The reversed string(using recursion) is :  nujrA hsirAH


...Program finished with exit code 0
Press ENTER to exit console.
```

If Statement:

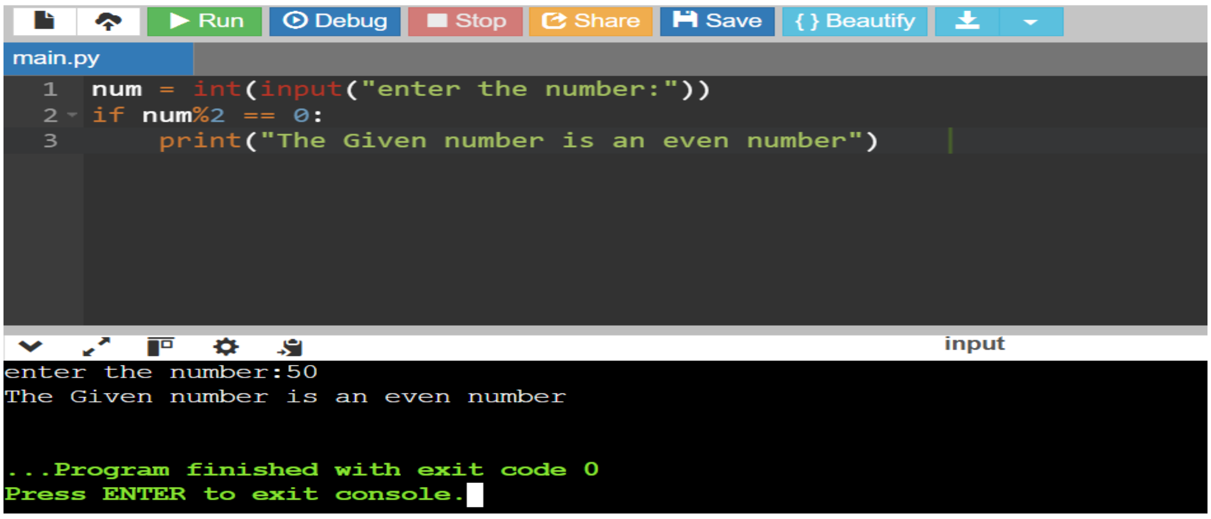
Example 1:



```python
a = int (input("Enter a: "));
b = int (input("Enter b: "));
c = int (input("Enter c: "));
if a>b and a>c:
    print ("From the above three numbers given a is largest");
if b>a and b>c:
    print ("From the above three numbers given b is largest");
if c>a and c>b:
    print ("From the above three numbers given c is largest");
```

```
Enter a: 120
Enter b: 100
Enter c: 150
From the above three numbers given c is largest


...Program finished with exit code 0
Press ENTER to exit console.
```

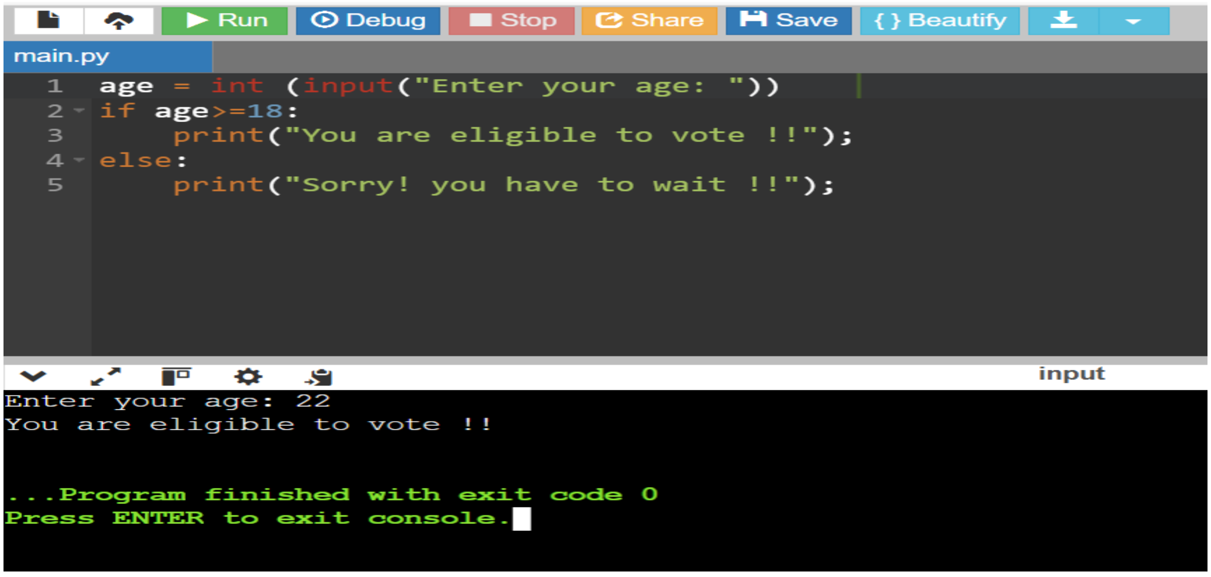
Example 2:



```python
num = int(input("enter the number:"))
if num%2 == 0:
    print("The Given number is an even number")
```

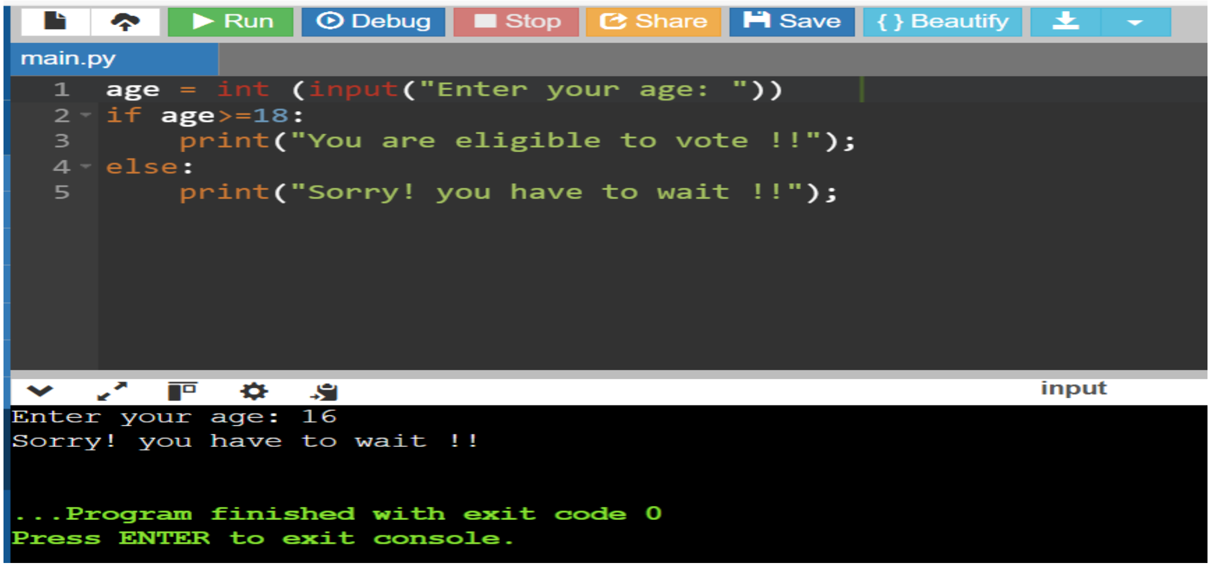
```
enter the number:50
The Given number is an even number

...Program finished with exit code 0
Press ENTER to exit console.
```

If-Else Statement:



```python
age = int (input("Enter your age: "))
if age>=18:
    print("You are eligible to vote !!");
else:
    print("Sorry! you have to wait !!");
```

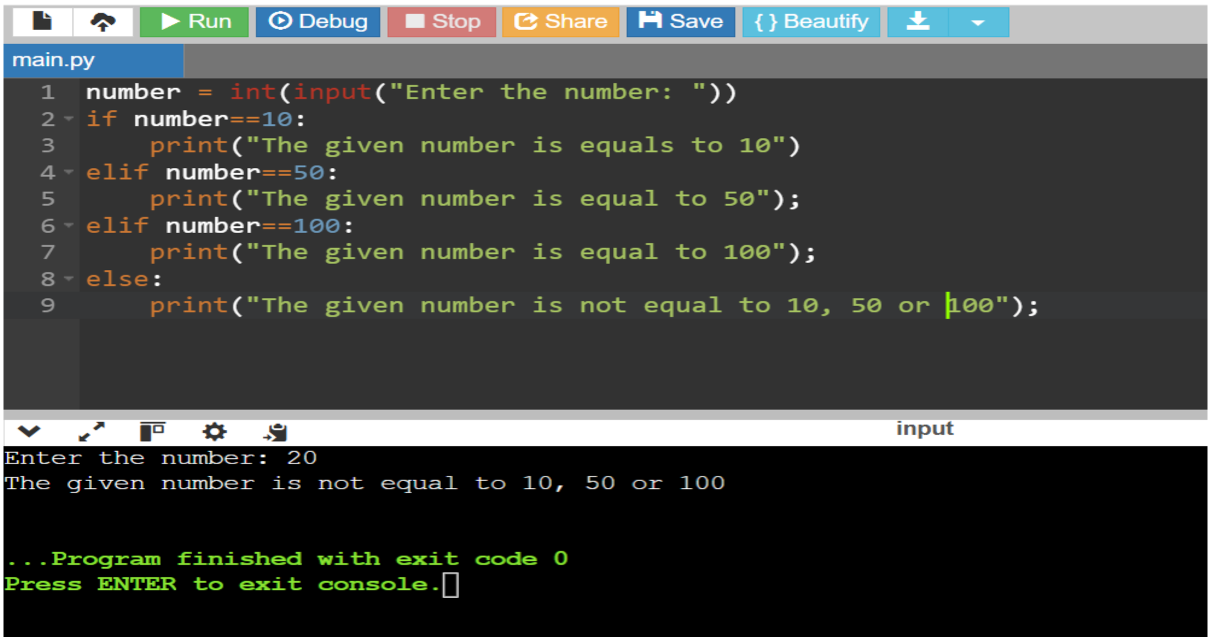
```
Enter your age: 22
You are eligible to vote !!


...Program finished with exit code 0
Press ENTER to exit console.
```



```python
age = int (input("Enter your age: "))
if age>=18:
    print("You are eligible to vote !!");
else:
    print("Sorry! you have to wait !!");
```

```
Enter your age: 16
Sorry! you have to wait !!


...Program finished with exit code 0
Press ENTER to exit console.
```

Elif Statement:

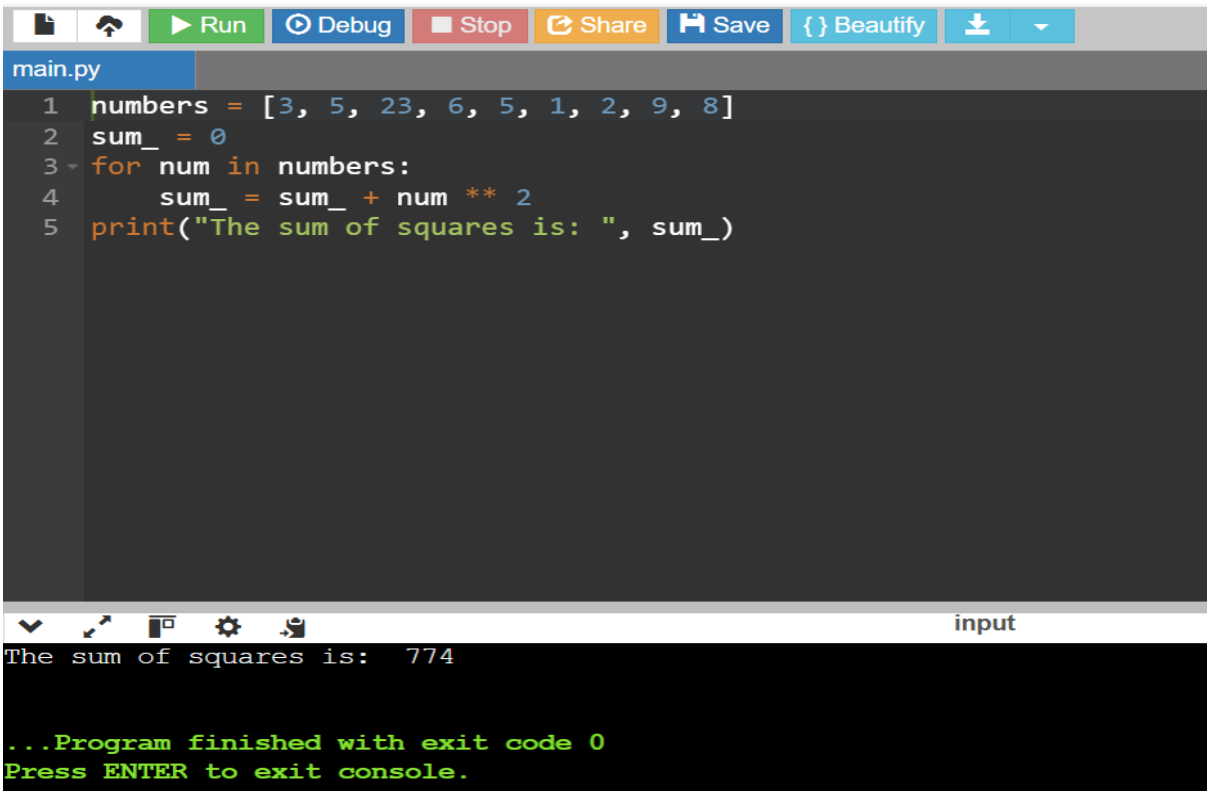


```
number = int(input("Enter the number: "))
if number==10:
    print("The given number is equals to 10")
elif number==50:
    print("The given number is equal to 50");
elif number==100:
    print("The given number is equal to 100");
else:
    print("The given number is not equal to 10, 50 or 100");
```

```
Enter the number: 20
The given number is not equal to 10, 50 or 100


...Program finished with exit code 0
Press ENTER to exit console.
```

FOR Loops:

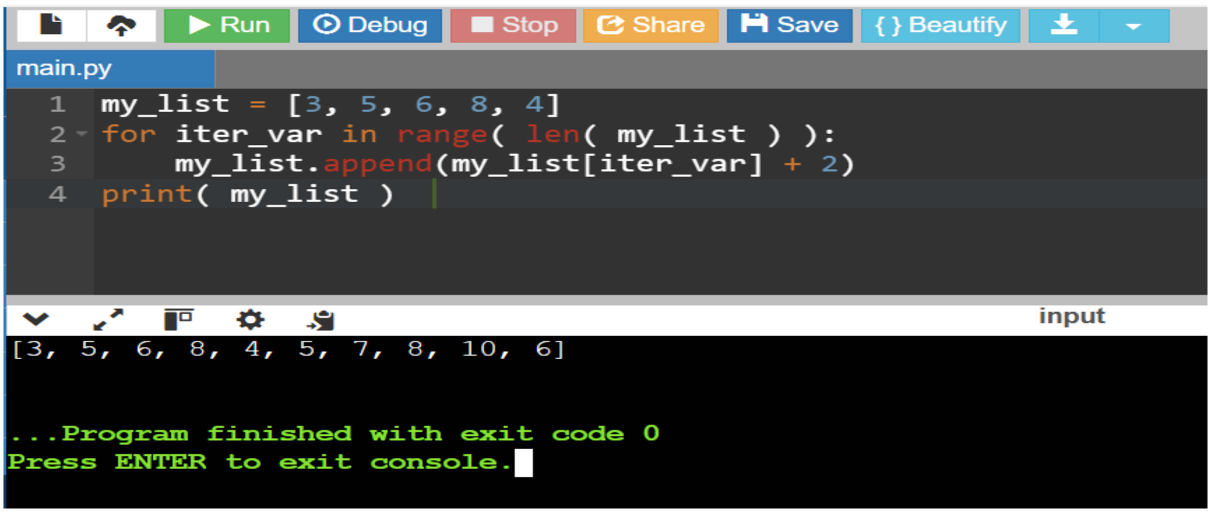
1. **Iterating by using index of sequence**



```
numbers = [3, 5, 23, 6, 5, 1, 2, 9, 8]
sum_ = 0
for num in numbers:
    sum_ = sum_ + num ** 2
print("The sum of squares is: ", sum_)
```

```
The sum of squares is:  774


...Program finished with exit code 0
Press ENTER to exit console.
```

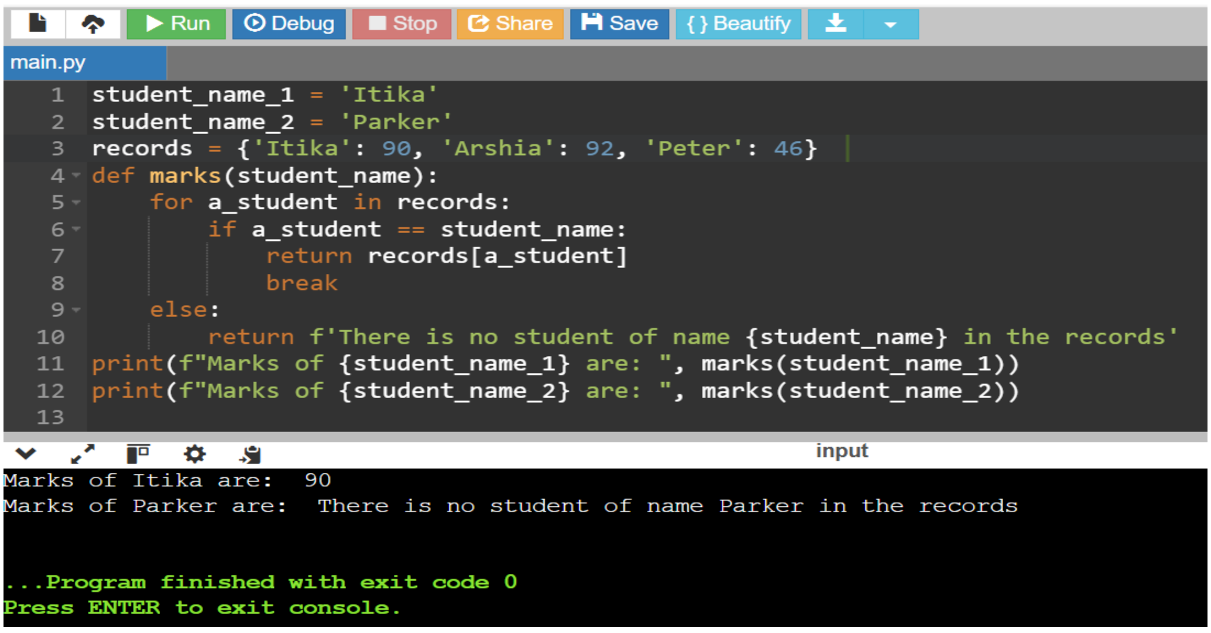
2. Using Range ()



```python
my_list = [3, 5, 6, 8, 4]
for iter_var in range( len( my_list ) ):
    my_list.append(my_list[iter_var] + 2)
print( my_list )
```

```
[3, 5, 6, 8, 4, 5, 7, 8, 10, 6]


...Program finished with exit code 0
Press ENTER to exit console.
```

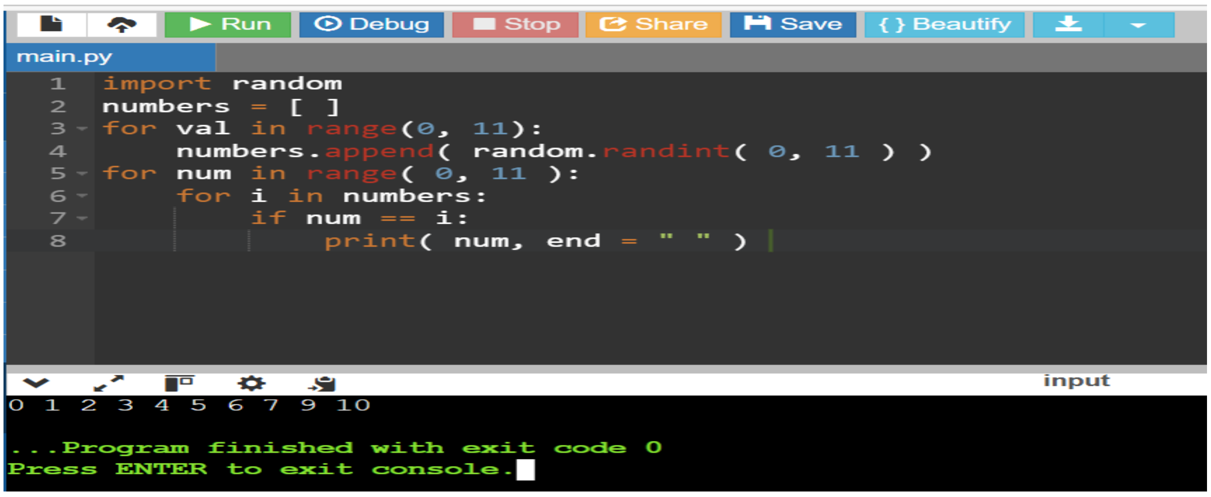
3. Using else statement with loop



```python
student_name_1 = 'Itika'
student_name_2 = 'Parker'
records = {'Itika': 90, 'Arshia': 92, 'Peter': 46}
def marks(student_name):
    for a_student in records:
        if a_student == student_name:
            return records[a_student]
            break
    else:
        return f'There is no student of name {student_name} in the records'
print(f"Marks of {student_name_1} are: ", marks(student_name_1))
print(f"Marks of {student_name_2} are: ", marks(student_name_2))

```

```
Marks of Itika are:  90
Marks of Parker are:  There is no student of name Parker in the records


...Program finished with exit code 0
Press ENTER to exit console.
```

4. Nested loop

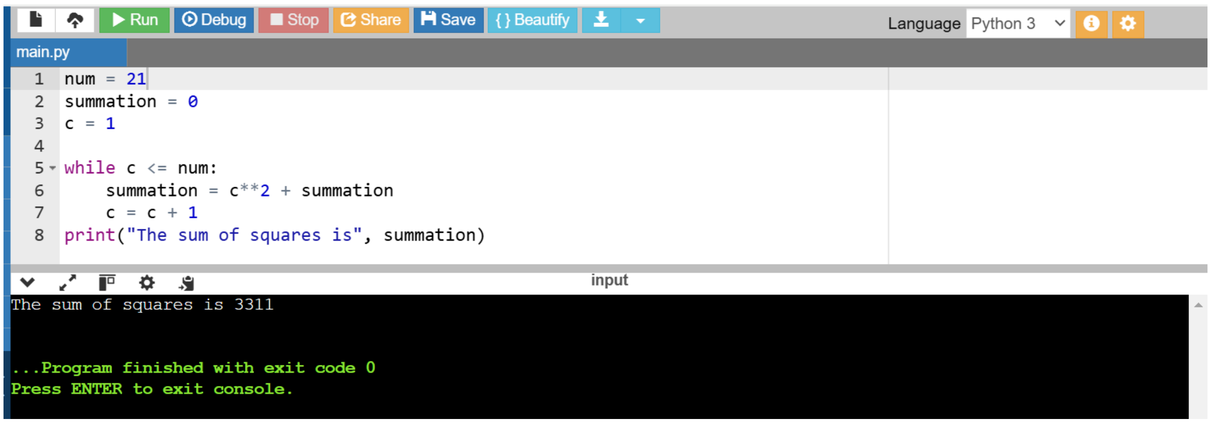


```python
import random
numbers = [ ]
for val in range(0, 11):
    numbers.append( random.randint( 0, 11 ) )
for num in range( 0, 11 ):
    for i in numbers:
        if num == i:
            print( num, end = " " )
```

```
0 1 2 3 4 5 6 7 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```

WHILE Loops:

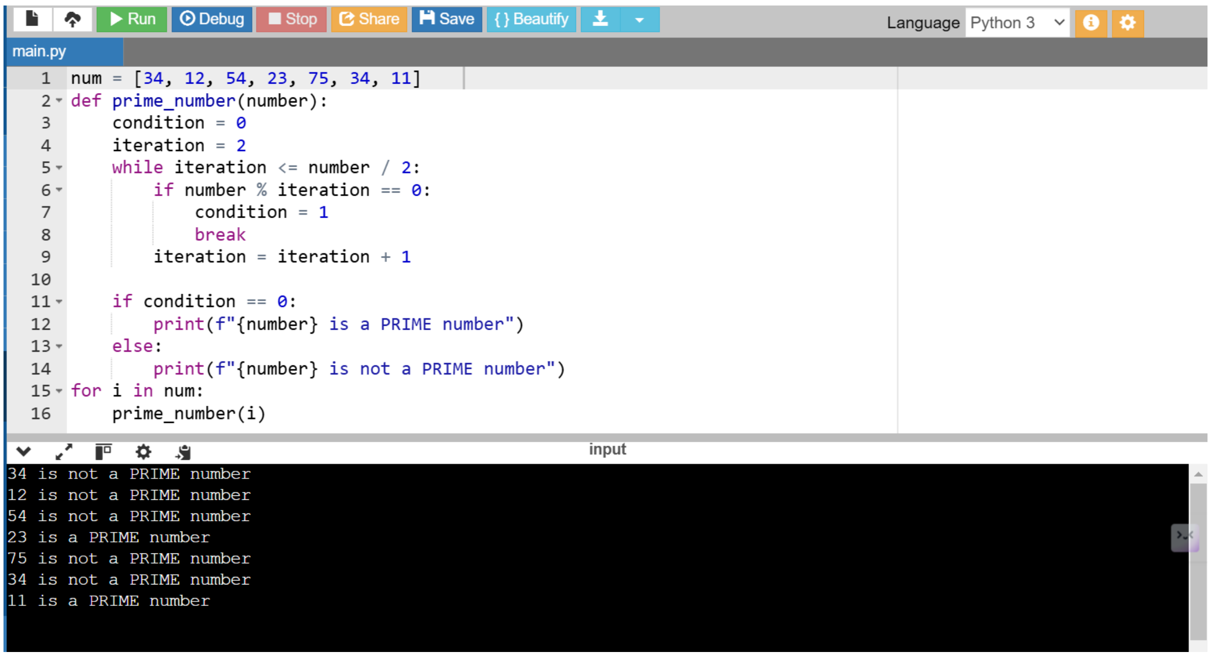
1. Sum of squares



```python
num = 21
summation = 0
c = 1

while c <= num:
    summation = c**2 + summation
    c = c + 1
print("The sum of squares is", summation)
```

```
The sum of squares is 3311

...Program finished with exit code 0
Press ENTER to exit console.
```

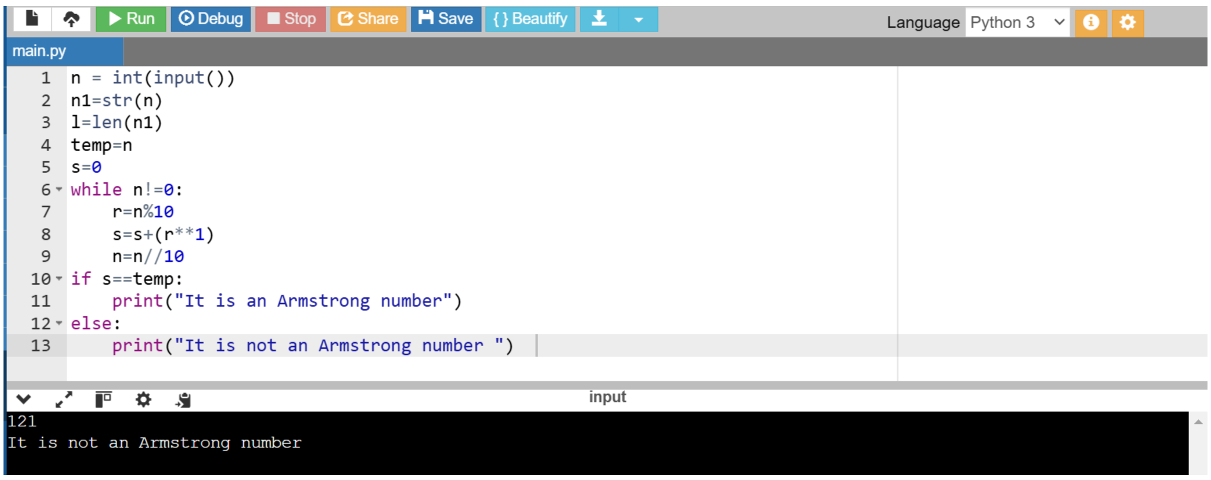
2. To check whether given number is Prime or not



```python
num = [34, 12, 54, 23, 75, 34, 11]
def prime_number(number):
    condition = 0
    iteration = 2
    while iteration <= number / 2:
        if number % iteration == 0:
            condition = 1
            break
        iteration = iteration + 1

    if condition == 0:
        print(f"{number} is a PRIME number")
    else:
        print(f"{number} is not a PRIME number")
for i in num:
    prime_number(i)
```

```
34 is not a PRIME number
12 is not a PRIME number
54 is not a PRIME number
23 is a PRIME number
75 is not a PRIME number
34 is not a PRIME number
11 is a PRIME number
```

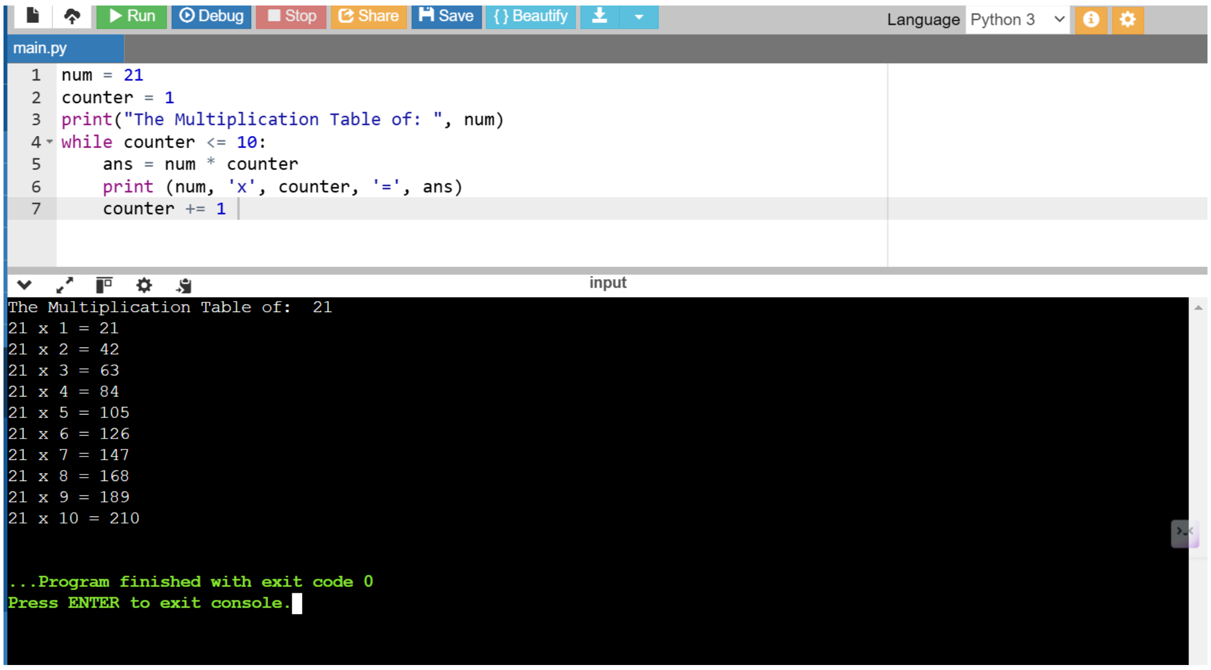
3. Armstrong number



```python
n = int(input())
n1=str(n)
l=len(n1)
temp=n
s=0
while n!=0:
    r=n%10
    s=s+(r**1)
    n=n//10
if s==temp:
    print("It is an Armstrong number")
else:
    print("It is not an Armstrong number ")
```

```
121
It is not an Armstrong number
```

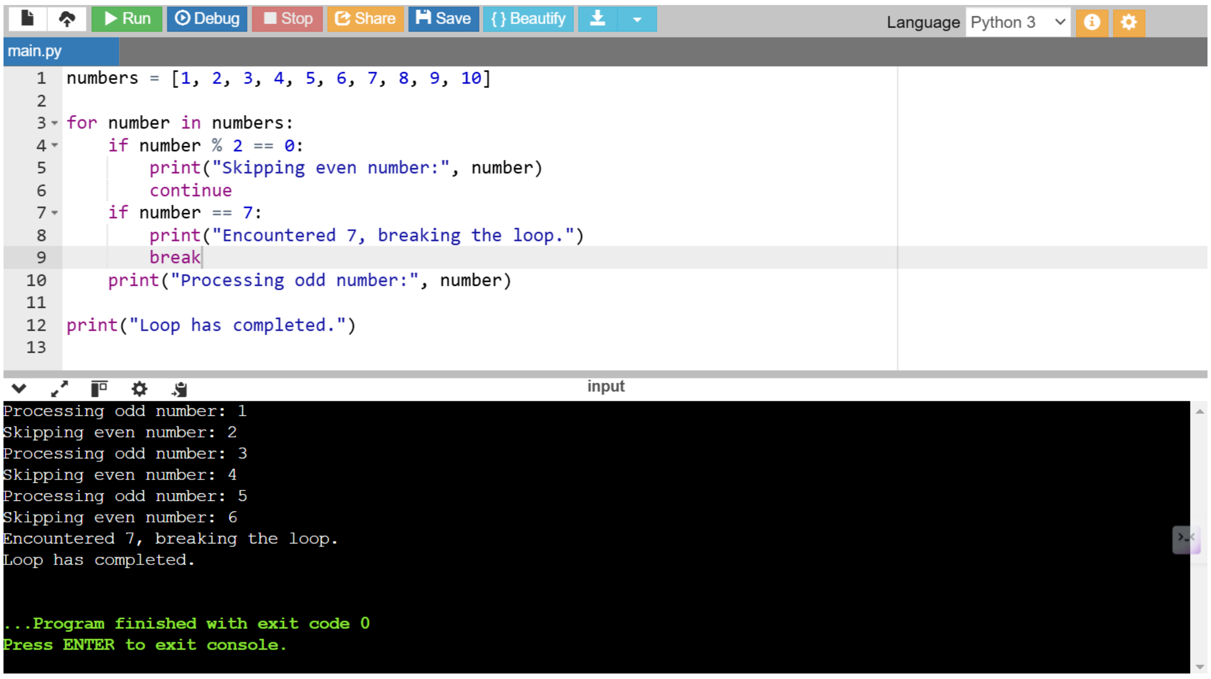
4. Multiplication Table:



```python
num = 21
counter = 1
print("The Multiplication Table of: ", num)
while counter <= 10:
    ans = num * counter
    print (num, 'x', counter, '=', ans)
    counter += 1
```

```
The Multiplication Table of:  21
21 x 1 = 21
21 x 2 = 42
21 x 3 = 63
21 x 4 = 84
21 x 5 = 105
21 x 6 = 126
21 x 7 = 147
21 x 8 = 168
21 x 9 = 189
21 x 10 = 210


...Program finished with exit code 0
Press ENTER to exit console.
```

BREAK Statement:



```python
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for number in numbers:
    if number % 2 == 0:
        print("Skipping even number:", number)
        continue
    if number == 7:
        print("Encountered 7, breaking the loop.")
        break
    print("Processing odd number:", number)

print("Loop has completed.")
```

```
Processing odd number: 1
Skipping even number: 2
Processing odd number: 3
Skipping even number: 4
Processing odd number: 5
Skipping even number: 6
Encountered 7, breaking the loop.
Loop has completed.


...Program finished with exit code 0
Press ENTER to exit console.
```