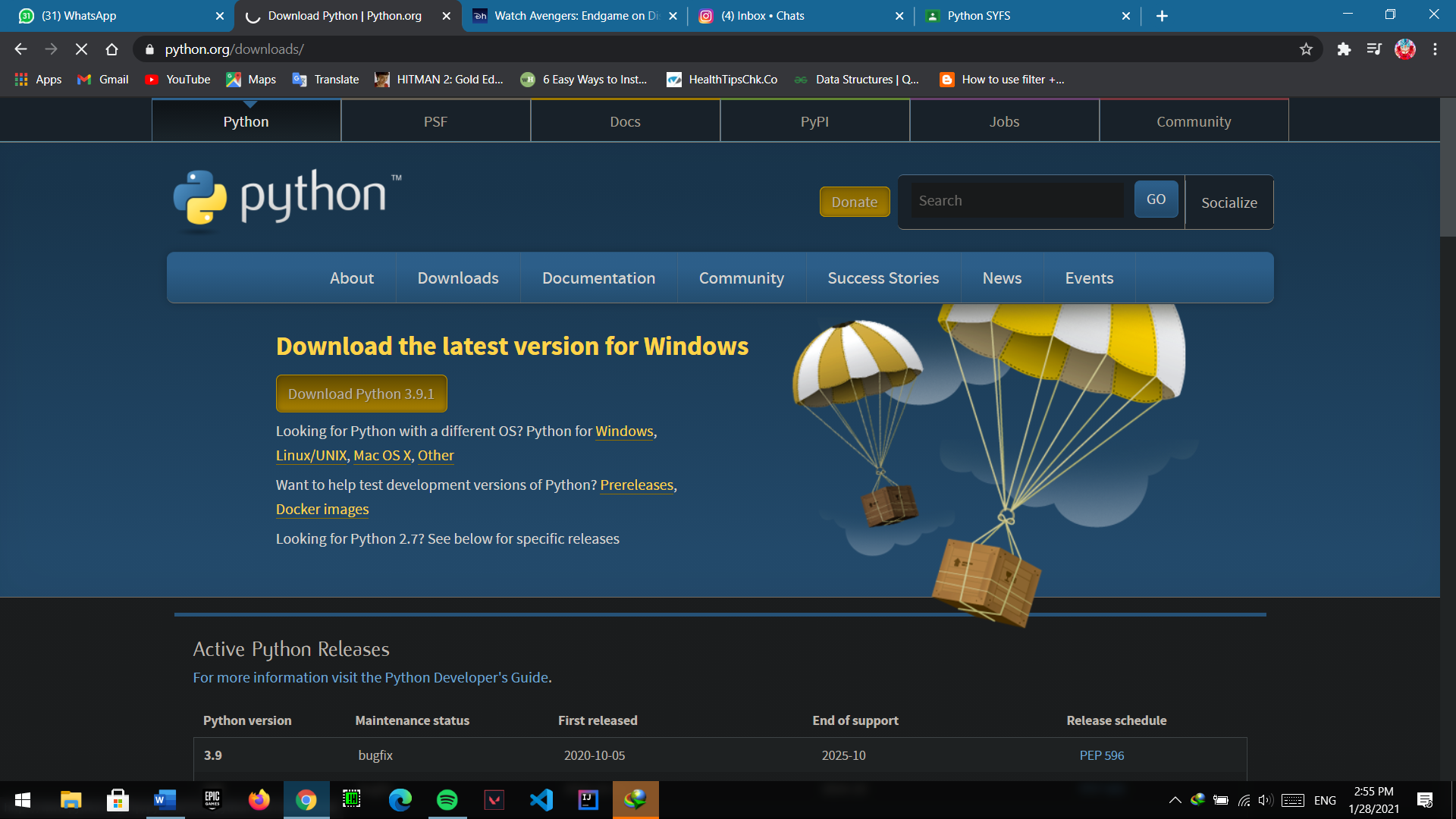
**PYTHON PROGRAMMNG**

1.**INSTALL&CONFIGURE PYTHON IDLE**

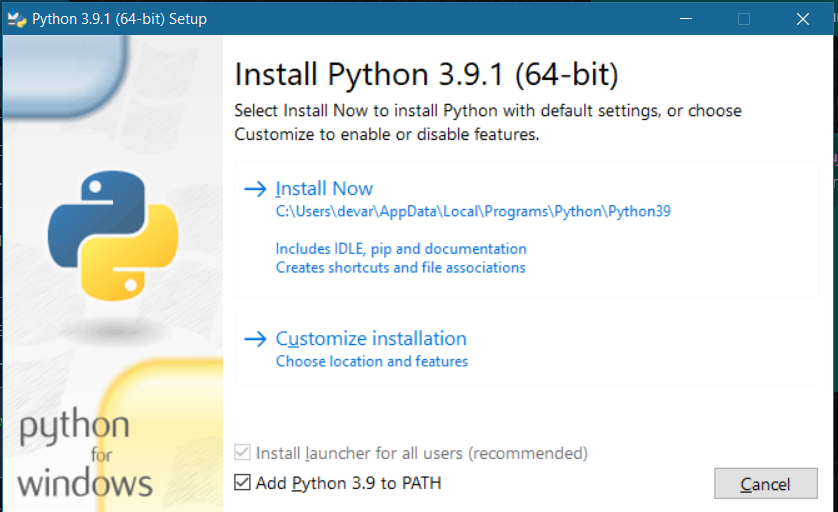
* STEP 1: Go TO <https://www.python.org/>

STEP 2: CLICK ON DOWNLOADS

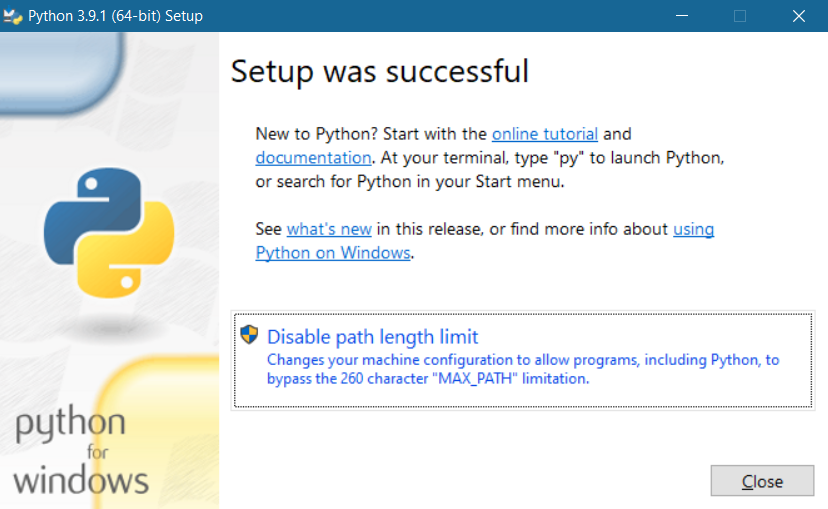


STEP 3: DOWNLOAD THE LATEST PYTHON AVAILABLE

STEP 4: AFTER COMPLETING THE DOWNLOAD, OPEN THE FILE



STEP 5: CLICK ON INSTALL NOW, BEFORE THIS SELECT **ADD PYHTON 3.9 TO PATH**



**HENCE PYTHON IS INSTALLED IN OUR SYSTEM .**

**2.** **Write a simple Python program using Operator,** **Illustrate use of various operators.**

Input:

x = 100

y = 300

print("Value of x :")

print(x)

print("\nValue of y :")

print(y)

print("\n\*\* ARITHMETIC OPERATORS \*\*")

print("\nAddition : x + y ")

print(x + y)

print("Subtraction : x-y ")

print(x-y)

print("\n\*\* Comparison Operators \*\*")

print("\ncheck whether y is greater than x:")

print(x<y)

print("\n\*\* Logical Operators \*\*")

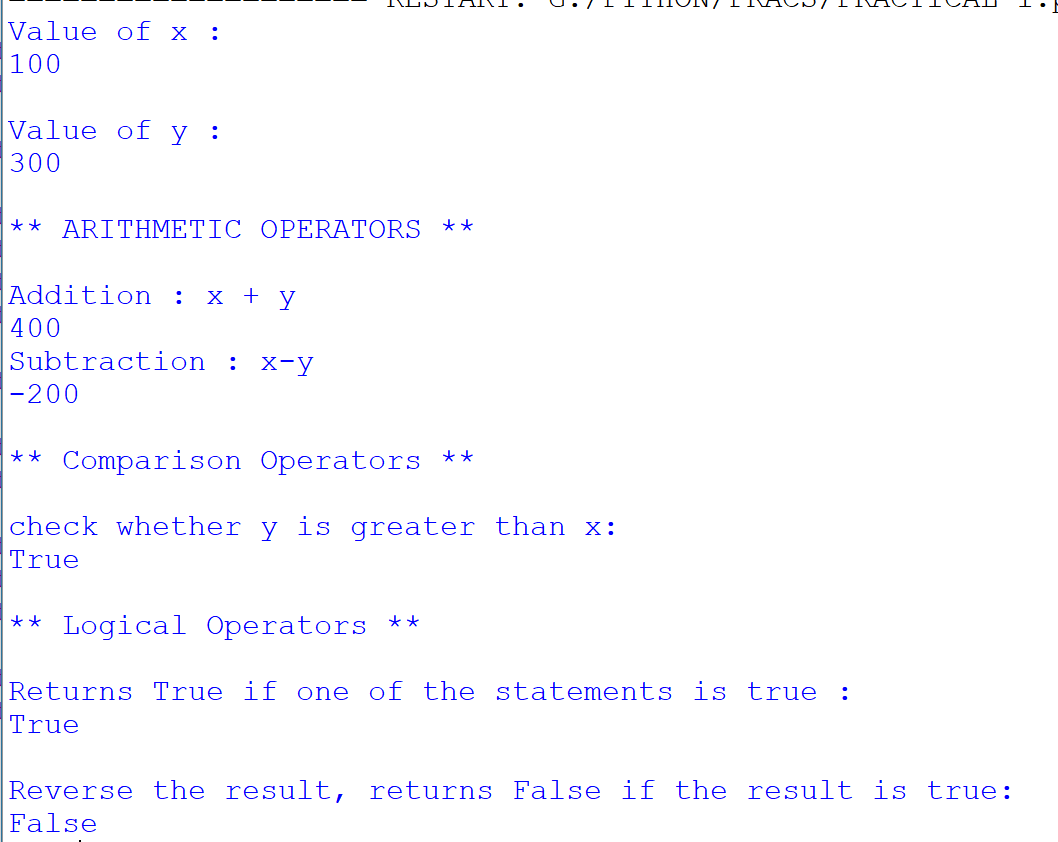
print("\nReturns True if one of the statements is true :")

print(x < 200 or x < 500)

print("\nReverse the result, returns False if the result is true:")

print(x < 5 or x < 4)

output:



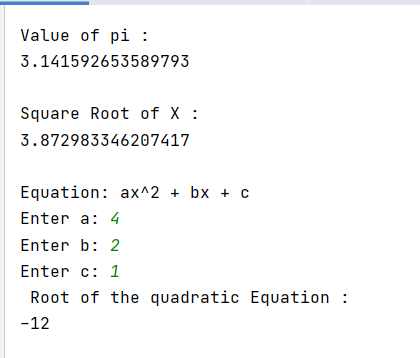
3. Write a Python program to demonstrate math built-in functions

(any two)

Input:

*# Import math Library***import** math  
  
*# Print the value of pi*print(**"\nValue of pi : "**)  
print(math.pi)  
  
*# Print the square root of x*x = 15  
print(**"\nSquare Root of X : "**)  
print(math.sqrt(x))  
  
*# Find Roots of Quadratic Equation*print(**"\nEquation: ax^2 + bx + c "**)  
  
a = int(input(**"Enter a: "**))  
b = int(input(**"Enter b: "**))  
c = int(input(**"Enter c: "**))  
d = b\*\*2-4\*a\*c  
  
print(**" Root of the quadratic Equation : "**)  
print(d)

Output:



4. Write a Python program to perform

following operations on list:

a) Create

b) Access

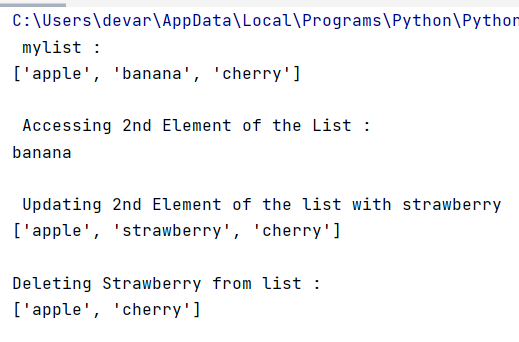
c) Update

d) Delete

Input:

*#perform following operations on list:  
# a) Create  
#b) Access  
#c) Update  
#d) Delete  
  
""""CREATE LIST"""*print(**" mylist :"**)  
mylist = [**"apple"**, **"banana"**, **"cherry"**]  
print(mylist)  
  
**""""ACCESS LIST"""**print(**"\n Accessing 2nd Element of the List : "**)  
print(mylist[1])  
  
**""""UPDATE LIST"""**print(**"\n Updating 2nd Element of the list with strawberry "**)  
mylist[1] = **"strawberry"**print(mylist)  
  
**""""DELETE LIST"""**print(**"\nDeleting Strawberry from list : "**)  
mylist.remove(**"strawberry"**)  
print(mylist)

Output:

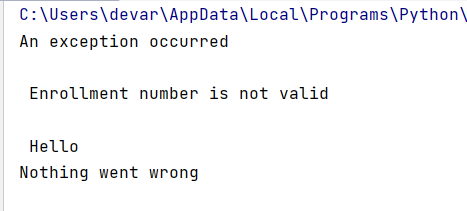


5. Write a program in Python to use try except statement.

Input:

*#The try block will generate an error, because x is not defined:***try**:  
 print(x/0)  
**except**:  
 print(**"An exception occurred"**)  
  
*#Print one message if the try block raises a NameError and another for other errors:***try**:  
 print(x)  
**except** NameError:  
 print(**"\n Enrollment number is not valid"**)  
**except**:  
 print(**"Something else went wrong"**)  
  
**"""You can use the else keyword to define a block of code to be executed if no errors were raised """  
try**:  
 print(**"\n Hello"**)  
**except**:  
 print(**"Something went wrong"**)  
**else**:  
 print(**"Nothing went wrong"**)

Output:



6. Study various application domains of Python along with its features.

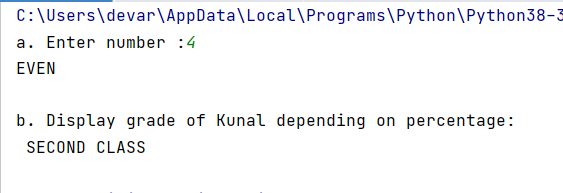
Input:

7. Write a Python program to demonstrate the use of if and if else.

Input:

*# Python supports the usual logical conditions from mathematics:  
# Equals: a == b  
# Not Equals: a != b  
# Less than: a < b  
# Less than or equal to: a <= b  
# Greater than: a > b  
# Greater than or equal to: a >= b  
  
"""""IF-ELSE STATEMENT"""*number = int(input(**"a.Enter number :"**))  
**if** number % 2 == 0:  
 print(**"EVEN"**)  
**else**:  
 print(**"ODD"**)  
  
**"""Display grade of student depending on percentage.  
percentage>=90: Excellent  
percentage>=80 and <90: First class  
percentage>=60 and <80: Second class  
percentage>=40 and <60: Pass Class  
percentage< 40: Fail"""**print(**"b. Display grade of Kunal depending on percentage."**)  
Kunal = 75  
**if** Kunal >= 90:  
 print(**"EXCELLENT "**)  
**else**:  
 **if** Kunal >80 **and** Kunal<90:  
 print(**" FIRST CLASS "**)  
 **else**:  
 **if** Kunal >=60 **and** Kunal<80:  
 print(**" SECOND CLASS "**)  
 **else** :  
 **if** Kunal>=40 **and** Kunal<60:  
 print(**" PASS CLASS "**)  
 **else**:  
 print(**"Fail"**)

Output:



8. Develop a user defined Python function to demonstrate the use of parameterized function & value return functions.

Input:

9.Write a Python program to perform following operations on

Tuples:

a) Create

b) Access

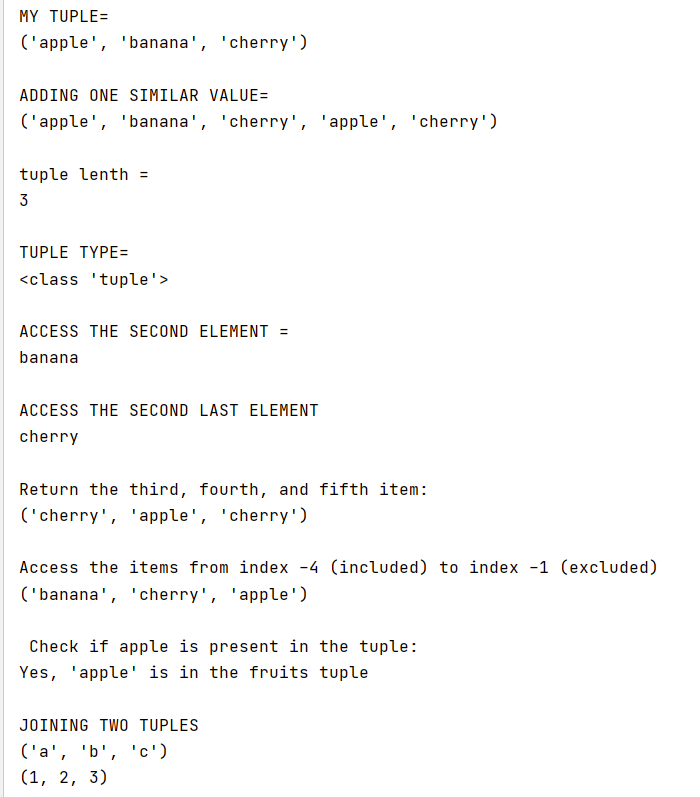
c) Update

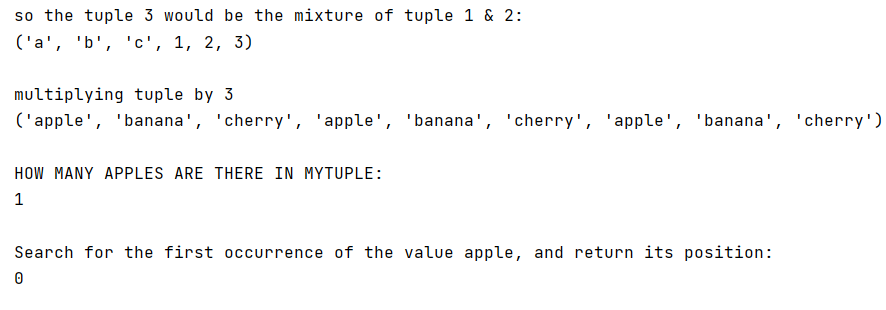
d) Delete

input:  
**"Create a Tuple:"**mytuple = (**"apple"**, **"banana"**, **"cherry"**)  
print(**"MY TUPLE="**)  
print( mytuple)  
  
**"""Tuple items are ordered, unchangeable, and allow duplicate values.  
Tuple items are indexed, the first item has index [0], the second item has index [1] etc"""""**print(**"\nADDING ONE SIMILAR VALUE="**)  
thistuple = (**"apple"**, **"banana"**, **"cherry"**, **"apple"**, **"cherry"**) *#it allows same/duplicate values*print(thistuple)  
  
*# Tuple Length*tuple = (**"apple"**, **"banana"**, **"cherry"**)  
print(**"\ntuple lenth ="**)  
print( len(tuple))  
  
*# Tuple items can be of any data type:***"TUPLE TYPE"**print(**"\nTUPLE TYPE="**)  
print( type(tuple))  
  
*# ACCESSING TUPLES***"Print the second item in the tuple:"**print(**"\nACCESS THE SECOND ELEMENT ="**)  
print(thistuple[1])  
  
  
**"""Negative indexing means start from the end.  
-1 refers to the last item, -2 refers to the second last item etc"""**print(**"\nACCESS THE SECOND LAST ELEMENT"**)  
print(thistuple[-1])  
  
*#Range of Positive Index*print(**"\nReturn the third, fourth, and fifth item:"**)  
print(thistuple[2:5]) *#ELEMENTS BETWEEN 2 (CHERRY) AND 5(MELON) WILL GET PRINT EXCEPT FOR LAST ELEMENT MELON***"Range of Negative Indexes"**print(**"\nAccess the items from index -4 (included) to index -1 (excluded)"**)  
print(thistuple[-4:-1])  
  
**"Check if apple is present in the tuple:"**print(**"\n Check if apple is present in the tuple:"**)  
**if "apple" in** thistuple:  
 print(**"Yes, 'apple' is in the fruits tuple"**)  
  
*# Note: Once a tuple is created, you cannot add items to it.  
# Note: You cannot remove items in a tuple.***"Join two tuples:"**tuple1 = (**"a"**, **"b"** , **"c"**)  
tuple2 = (1, 2, 3)  
  
tuple3 = tuple1 + tuple2  
print(**"\nJOINING TWO TUPLES"**)  
print(tuple1)  
print( tuple2)  
print(**"so the tuple 3 would be the mixture of tuple 1 & 2:"**)  
print(tuple3)  
  
**"Multiply the fruits tuple by 2:"**MUL = mytuple \* 3  
print(**"\nmultiplying tuple by 3"**)  
print(MUL)  
  
**"The count() method returns the number of times a specified value appears in the tuple."**print(**"\nHOW MANY APPLES ARE THERE IN MYTUPLE:"**)  
print(mytuple.count(**"apple"**))  
  
**"The index() method finds the first occurrence of the specified value."**print(**"\nSearch for the first occurrence of the value apple, and return its position:"**)  
print(thistuple.index(**"apple"**)) *#The index() method raises an exception if the value is not found*

*.*

Output:

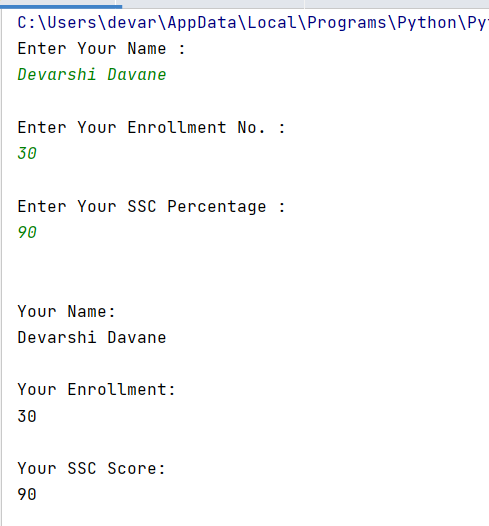




10. Write a Python program to readkeyboard input & print it to the screen.

Input:

print(**"Enter Your Name :"**)  
Name = str(input())  
print(**"\nEnter Your Enrollment No. :"**)  
Enrollment = int(input())  
print(**"\nEnter Your SSC Percentage :"**)  
SSC = int(input())  
  
print(**"\n"**)  
print(**"Your Name:"**)  
print(Name)  
print(**"\nYour Enrollment:"**)  
print(Enrollment)  
print(**"\nYour SSC Score:"**)  
print(SSC)

output:

11.Write a Python program to display a

message on the screen.

Input:

Output:

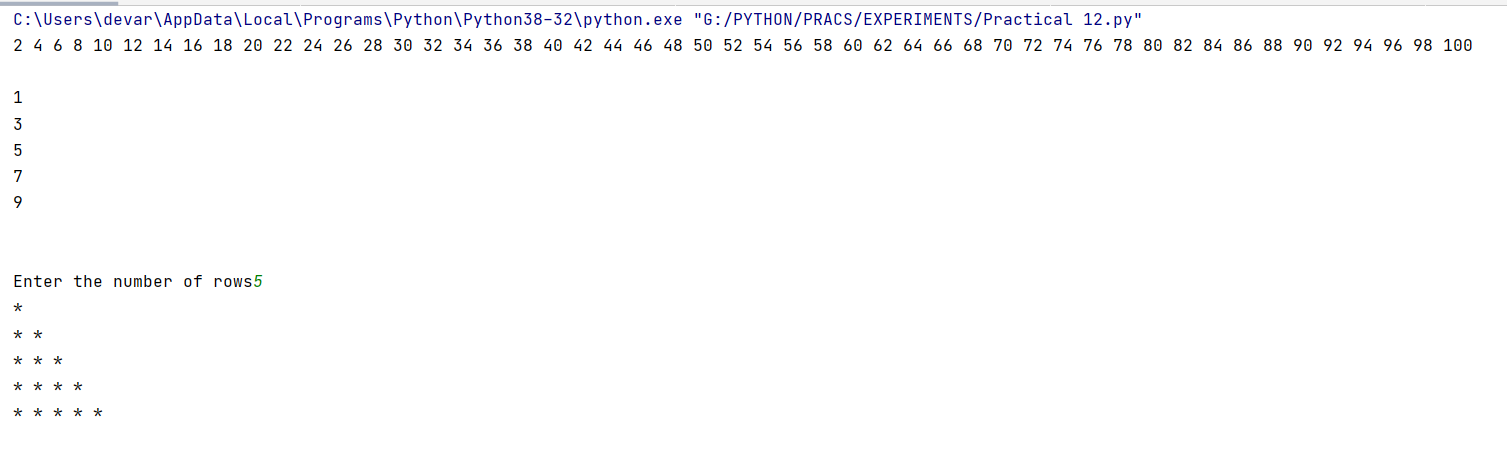
12. Write a Python program to demonstrate the use of looping

statements.

Input:

*# Python program to print Even Numbers in given range*start, end = 1, 100  
  
*# iterating each number in list***for** num **in** range(start, end + 1):  
  
 *# checking condition* **if** num % 2 == 0:  
 print(num, end=**" "**)  
  
print(**"\n"**)  
*# Python3 code to iterate over a list*list = [1, 3, 5, 7, 9]  
  
*# Using for loop***for** i **in** list:  
 print(i)  
  
print(**"\n"**)  
*# This is the example of print simple pyramid pattern*n = int(input(**"Enter the number of rows"**))  
*# outer loop to handle number of rows***for** i **in** range(0, n):  
 *# inner loop to handle number of columns  
 # values is changing according to outer loop* **for** j **in** range(0, i + 1):  
 *# printing stars* print(**"\* "**, end=**""**)  
  
 *# ending line after each row* print()

Output:



13. Write a Python program to demonstrate the use of built-in

packages & user defined packages.

Input:

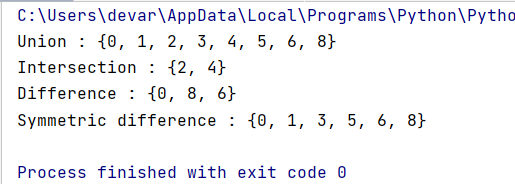
Output:

14. Write a Python program to perform operations on Sets.

Input:

*# Program to perform different set operations  
# as we do in mathematics  
  
# sets are define*A = {0, 2, 4, 6, 8};  
B = {1, 2, 3, 4, 5};  
  
*# union*print(**"Union :"**, A | B)  
  
*# intersection*print(**"Intersection :"**, A & B)  
  
*# difference*print(**"Difference :"**, A - B)  
  
*# symmetric difference*print(**"Symmetric difference :"**, A ^ B)

Output:



15. Write a program in Python to extract data from csv /txt

Input:

Output: