**Ex.No.1 .1 To Swap two numbers using temporary variable**

Aim: To write a python code to swap two numbers using Temporary variables.

Algorithm:

Step 1: Start the program

Step 2: Read the values num1, num2

Step 3: Assign temp=num1

num1=num2

num2=temp

Step 4: Print the values of num1 and num2

Step 5: Stop the program

Flow Chart:

**Input num1, num2**

temp=num1

num1=num2

num2=temp

**Print num1,num2**

Program:

# To swap two numbers using temporary variable

# <rollno> <name>

num1=input('Enter the First Nmber:')

num2=input('Enter the Second Number:')

print('Before Swapping ')

print('Number1=',num1,' Number2=',num2)

temp=num1

num1=num2

num2=temp

print('After Swapping ')

print('Number1=',num1,' Number2=',num2)

Output:

Enter the First Nmber:12

Enter the Second Number:10

Before Swapping

Number1= 12 Number2= 10

After Swapping

Number1= 10 Number2= 12

Result:

This python program to swap two numbers using temporary variable is executed successfully.

Ex.No.1.2 To Swap two numbers without using temporary variable

Date:

Aim: To write a python code to swap two numbers without using Temporary variables.

Algorithmi:

Step 1: Start the program

Step 2: Read the values num1, num2

Step 3: Assign

num1=num1+num2

num2=num1-num2

num1=num1-num2

Step 4: Print the values of num1 and num2

Step 5: Stop the program

Flow Chart:

**Input num1, num2**

num1=num1+num2

num2=num1-num2

num1=num1- num2

**Print num1,num2**

Program:

#Ex.1.2 Swapping two numbers without using temporary variables

# <rollno> <name>

num1=input('Enter the First Nmber:')

num2=input('Enter the Second Number:')

print('Before Swapping ')

print('Number1=',num1,' Number2=',num2)

num2,num1=num1,num2

print('After Swapping ')

print('Number1=',num1,' Number2=',num2)

Result:

This python program to swap two numbers without using temporary variable is executed successfully.

Ex.No.1.3 To convert Celsius to Fahrenheit and Fahrenheit to Celsius vice versa

Date:

Aim: To write a python code to convert Celsius to Fahrenheit and Fahrenheit to Celsius vice versa

Algorthm:

Step 1: Start the program

Step 2: Input the Celsius value.

Step 3: Calculate Fahrenheit = (Celsius \* 1.8) +32

Step 4: Print the Fahrenheit value.

Step 5: Calculate Celsius = (Fahrenheit – 32)/1.8

Step 6: Print the Celsius value.

Step 7: Stop the Program.

Flow Chart:

**Input Celsius value**

Fahrenheit=(Celsius\*1.8)+32

Celsius=(fahrenheit-32)/1.8

**Print Celsius value**

**Print Fahrenheit value**

Program:

# To convert celsius to fahrenheit and fahrenheit to celsius vice versa

celsius=float(input('Enter the temperature in celsius:\n'))

fahrenheit=(celsius \* 1.8)+32

print('Temperature in fahrenheit is : \t',fahrenheit)

print('Fahrenheit to celsius conversion...')

celsius1=(fahrenheit-32)/1.8

print('Temperature in celsius is:\t',celsius1)

Output:

Enter the temperature in celsius:

32

Temperature in fahrenheit is : 89.6

Fahrenheit to celsius conversion...

Temperature in celsius is: 31.999999999999996

Result: The python program to convert Celsius to Fahrenheit and Fahrenheit to Celsius is executed successfully.

Ex.N0.1.4 To write a Python on code to find the area and circumference of the circle.

Aim: To write a python code to find the area and circumference of circle.

Algorithm:

Step 1: Start the program

Step2: Enter the PI value

Step3:Input the radius value

Step4: Compute the area of the circle πr2

Step5: Compute the circumference of the circle 2 πr

Step6: Print the Circumference of the circle.

Step7: Print the Area of the circle

Step8: Stop the program

**Input radius value**

Pi=3.14

Circumference=2\*pi\*radius

**Print Area, Circumference**

Area=pi\*radius\*radius

Program:

# To find the Area of Circumference of the circle

pi=3.14

radius=float(input('Enter the radius of circle:'))

area=3.14\* radius\* radius

circumference=2\*3.14 \*radius

print('Area of the circle=',area)

print('Circumference of the circle =',circumference)

Output:

Enter the radius of circle:7.0

Area of the circle= 153.86

Circumference of the circle = 43.96

Result: This python program to find the Area and Circumference of the circle is executed successfully.

**Ex.1.5 Integer to Binary, Octal and Hexadecimal and Vice- versa**

Date:

Aim: To write a Python program to convert Integer to Binary, Octal and Hexadecimal and Vice-versa

Algorithms:

Step 1: Start the program

Step 2:Input the value of num1

Step 3: Assign bin1=bin(num1)

Oct1=oct(num1)

Hexadec1=hex(num1)

Step 4: Print the value of integer, Binary, Octal, Hexa decimal equivalent of bin1,oct1,hexadec1.

Step 5: Assign binnum is equal to int(bin1,2)

Step 6: Assign octnum is equal to int(oct1,8)

Step 7: Assign hexadecimal is equal to int(hexadec1,16)

Step 8: Print the integer value

Step 9: Stop the program.

Input num1

bin1=bin(num1)

oct1=oct(num1)

hexadec=hex(num1)

oct

Binnum=int(bin1,2)

Octnum=int(oct1,8)

Hexadecnum=int(hexadec1,16)

Print binnum, octnum

hexadecnum

Print num1, bin1, oct1 and hexadec1

**Program:**

# 1.5 Binary, Octal, HexaDecimal Conversion

num1=int(input('Enter the integer number:'))

bin1=bin(num1)

oct1=oct(num1)

hexadec1=hex(num1)

print('Input Integer value:',num1)

print('Integer to Binary Equivalent:', bin1)

print('Integer to Octal Equivalent:',oct1)

print('Integer to HexaDecimal Equivalent=',hexadec1)

binnum=int(bin1,2)

octnum=int(oct1,8)

hexadecnum=int(hexadec1,16)

print('Binary to Integer Equivalent=', binnum)

print('Octal to Integer Equivalent=',octnum)

print('Hexa Decimal to Integer Equivalent=',hexadecnum)

**Output:**

Enter the integer number:15

Input Integer value: 15

Integer to Binary Equivalent: 0b1111

Integer to Octal Equivalent: 0o17

Integer to HexaDecimal Equivalent= 0xf

Binary to Integer Equivalent= 15

Octal to Integer Equivalent= 15

Hexa Decimal to Integer Equivalent= 15

Result:

This python program created for conversion of Binary, Octal and Hexadecimal conversion in executed successfully.

**Ex.1.6 Find the Distance between two points**

**Aim:** To write a python code to find the distance between two points.

Algorithm:

Step 1: Start the program

Step 2: Using import math;

Step 3: Input the value for x1,y1,x2,y2.

Step 4: Calculate distance1=math.sqrt((x2-x1)\*\*2 + (y2-y1)\*\*2)

Step 5: Print the distance1

Step 6: Stop the program.

Flow Chart:

Import math

Distance=math.sqrt(x2-x1)\*\*2 + (y2-y1)\*\*2

**Print distance**

**Input x1,y1,x2,y2**

Program:

# 1.6 To find Distance between two points

import math

x1=float(input('Enter the First value of Starting point X1:'))

y1=float(input('Enter the Second value of Starting point Y1:'))

x2=float(input('Enter the First value of Ending point X2:'))

y2=float(input('Enter the Second value of Ending point Y2:'))

dist = math.sqrt((x2 - x1)\*\*2 + (y2 - y1)\*\*2)

print ('The distance between the points is:', dist)

# formatted output

print ('The distance between the points is:%.2f'% dist)

**Output:**

Enter the First value of Starting point X1:14

Enter the Second value of Starting point Y1:25

Enter the First value of Ending point X2:16

Enter the Second value of Ending point Y2:29

The distance between the points is: 4.47213595499958

The distance between the points is:4.47

Result:

Thus the python code is written to find the distance between two points and executed.

**Ex.No.2.1 Perform various Arithmetic operations**

Aim:

To write a python code to perform arithmetic operations.

Algorithm:

Step 1: Start the program.

Step 2: Input the values of num1,num2

Step 3: Calculate add1=num1+num2

Sub1=num1-num2

Mul1=num1\*num2

Div1=num1/num2

Mod1=num1%num2

Exp1=num1\*\*num2

floor\_division1=num1//num2

Step 4: Print the value for addition, subtraction, multiplication, division, modulus, exponentiation and floor division.

Step 5: Stop the program.

**Program:**

# Write a python code to perform Arithmetic operation

num1=int(input('Enter the First number:'))

num2=int(input('Enter the Second number:'))

add1=num1+num2

sub1=num1-num2

mul1=num1\*num2

div1=num1\*num2

mod1=num1%num2

expo1=num1\*\*num2

floor\_division1=num1//num2

print('Addition=',add1)

print('Subtraction=',sub1)

print('Multiplication=',mul1)

print('Division=',div1)

print('Modulus=',mod1)

print('Exponent=',expo1)

print('Floor Division=',floor\_division1)

**Output:**

Enter the First number:50

Enter the Second number:10

Addition= 60

Subtraction= 40

Multiplication= 500

Division= 500

Modulus= 0

Exponent= 97656250000000000

Floor Division= 5

Ex.No.2.2 To calculate square, cube, exponent and square root of a number

Aim:

To write a program code to calculate, square, cube, exponent and square root of a number.

Algorithm:

Step 1: Start the program.

Step 2: Import math and Enter the first number.

Step 3: Enter the Second number.

Step 4: Calculate square1=num1\*\*2

Cube1=num1\*\*2

Exponent1=num1\*\*3

Assign squareroot1=math.sqrt(num1)

Step 5: Print the value for squre1, cub1, exponent1, and squreroot1.

Step 6: Stop the program.

Program:

# 2.2 Calcuate Square, Cube, Exponent of a Number and Squareroot.

import math

number1=int(input('Enter the First Number:'))

number2=int(input('Enter the Second Number:'))

square1=number1\*\*2

cube1=number1\*\*3

exponent1=number1\*\*number2

squareroot1=math.sqrt(number1)

print(num1)

print('Square of ',number1,'is',square1)

print('Cube of ',number1,'is',cube1)

print('Exponent of ',number1,'and',num2,'is',exponent1)

print('Squareroot of ',number1,'is',squareroot1)

**Output:**

Enter the First Number:10

Enter the Second Number:2

50

Square of 10 is 100

Cube of 10 is 1000

Exponent of 10 and 2 is 100

Squareroot of 10 is 3.1622776601683795

**Result:**

This python code to calculate square, cube, exponent and square root of a number is executed successfully.

Ex.No.3. To display the current python version, keywords and calendar

Aim:

To write a python code to display the current python version, keywords and calendar.

**Algorithm:**

Step 1: Start the program.

Step 2: Import sys.math, keyword, and calendar.r

Step 3: Print the sys.version , keyword, calendar.

Step 4: Input value of month in two digit.

Step 5: Input value of year in Four digit.

Step 6: Print the output values

print('Square of ',number1,'is',square1)

print('Cube of ',number1,'is',cube1)

print('Exponent of ',number1,'and',num2,'is',exponent1)

print('Squareroot of ',number1,'is',squareroot1)

Step 7: Stop the program.

**Program:**

# Write a python program to display the current Python version, Keywords and Calendar

import sys

import keyword

import calendar

print('Current System Version=',sys.version)

print('Keywords List=',keyword.kwlist)

month1=int(input('Enter the Month in Two Digits:'))

year1=int(input('Enter theYear in Four Digits:'))

print('\n Calendar of Month',month1,'and Year', year1)

print(calendar.month(year1,month1))

print('-----------------------------')

print('\n Calendar of Year',year1)

print(calendar.calendar(year1))

**Output:**

Current System Version= 3.6.6 |Anaconda, Inc.| (default, Jun 28 2018, 17:14:51)

[GCC 7.2.0]

Keywords List= ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Enter the Month in Two Digits:05

Enter theYear in Four Digits:2019

Calendar of Month 5 and Year 2019

May 2019

Mo Tu We Th Fr Sa Su

1 2 3 4 5

6 7 8 9 10 11 12

13 14 15 16 17 18 19

20 21 22 23 24 25 26

27 28 29 30 31

-----------------------------

Calendar of Year 2019

2019

January February March

Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su

1 2 3 4 5 6 1 2 3 1 2 3

7 8 9 10 11 12 13 4 5 6 7 8 9 10 4 5 6 7 8 9 10

14 15 16 17 18 19 20 11 12 13 14 15 16 17 11 12 13 14 15 16 17

21 22 23 24 25 26 27 18 19 20 21 22 23 24 18 19 20 21 22 23 24

28 29 30 31 25 26 27 28 25 26 27 28 29 30 31

April May June

Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su

1 2 3 4 5 6 7 1 2 3 4 5 1 2

8 9 10 11 12 13 14 6 7 8 9 10 11 12 3 4 5 6 7 8 9

15 16 17 18 19 20 21 13 14 15 16 17 18 19 10 11 12 13 14 15 16

22 23 24 25 26 27 28 20 21 22 23 24 25 26 17 18 19 20 21 22 23

29 30 27 28 29 30 31 24 25 26 27 28 29 30

July August September

Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su

1 2 3 4 5 6 7 1 2 3 4 1

8 9 10 11 12 13 14 5 6 7 8 9 10 11 2 3 4 5 6 7 8

15 16 17 18 19 20 21 12 13 14 15 16 17 18 9 10 11 12 13 14 15

22 23 24 25 26 27 28 19 20 21 22 23 24 25 16 17 18 19 20 21 22

29 30 31 26 27 28 29 30 31 23 24 25 26 27 28 29

30

October November December

Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su

1 2 3 4 5 6 1 2 3 1

7 8 9 10 11 12 13 4 5 6 7 8 9 10 2 3 4 5 6 7 8

14 15 16 17 18 19 20 11 12 13 14 15 16 17 9 10 11 12 13 14 15

21 22 23 24 25 26 27 18 19 20 21 22 23 24 16 17 18 19 20 21 22

28 29 30 31 25 26 27 28 29 30 23 24 25 26 27 28 29

30 31

**Result:** This python code to display the current python version, keywords and calendar is executed successfully.

Ex.No.2.4 To check whether the person is eligible to vote or not.

Aim: To write a python code to check whether the person is eligible to vote or not.

Algorithm:

Step 1: Start the program.

Step 2: Input the value for age.

Step 3: Check the condition if age is greater than and equal to 18.

Step 4: Print (‘You are eligible to cast a vote’)

Step 5: Else, Print(‘You are not eligible to cast a vote’)

Step 6: Stop the program.

**Program:**

# Write a python program to check the person is elgible to cast the vote or not..

age=int(input('Enter your age:'))

if(age>=18):

print('Congrats! You are Elgible to cast the Vote')

else:

print('Sorry! You are not Elgible to vote')

**Output:**

Enter your age:47

Congrats! You are Eligible to cast the Vote

Enter your age:16

Sorry! You are not Eligible to vote

Result:

This python program to check whether a person is eligible to put the vote or not is executed successfully.

Ex.No.2.5 To check the given number is ODD or EVEN

Aim: To write a python code to check the given number is ODD or EVEN.

**Algorithm:**

Step 1: Start the program.

Step 2: Enter a number for num1.

Step 3: Check the condition if(num1 % 2 ==0)

Step 4: print(‘Num1 is ODD’)

Step 5: print(‘Num1 is EVEN’)

Step 6: Stop the program.

**Program:**

# 2.5 Write a python code to check the given number is odd or even..

num1=int(input('Enter a number:'))

if(num1%2 ==0):

print('Num1 is EVEN')

else:

print('Num1 is ODD')

**Output:**

Enter a number:59

Num1 is ODD

Enter a number:90

Num1 is EVEN

**Result:**

This python code to check the given number is ODD or EVEN is executed successfully.

Ex.No.3.1 To find the Biggest among two numbers

Aim: To write a python code to find the biggest between two numbers

Algorithm:

Step 1: Start the program.

Step 2: Read the values for num1,num2.

Step 3: Check the condition if num1 is greater than num2.

Step 4: Print (‘Number1 is the biggest number..’)

Step 5: else Print(‘Number2 is the biggest number..’)

Step 6: Stop the program.

**Program:**

# 2.5 Write a python code to find the biggest among two numbers

num1=float(input('Enter the first number:'))

num2=float(input('Enter the second number:'))

if(num1>num2):

print('Number1 is the biggest number',num1)

else:

print('Number2 is the biggest number',num2)

**Output:**

Enter the first number:45

Enter the second number:25

Number1 is the biggest number 45.0

Result: This python code to find the biggest among two numbers is executed successfully.

**Ex.No.3.2 To find the biggest among three numbers**

**Aim:**

To write a python code to find biggest among three numbers.

**Algorithm:**

Step 1: Start the program.

Step 2: Read the values for num1,num2 and num3.

Step 3: Check the condition if num1 is greater than num2 and num1 is greater than num3.

Step 4: print (num1,’ is greater than ‘,num2 ,’ and ‘,num3)

Step 5: Using elif to check num2 is greater than num3

Step 6: print(num2,’ is greater than ‘, num2,’ and ‘ , num3)

Step 7: else print(num3,’ is greater than’, num1, ‘ and ‘ , num3)

Step 8: Stop the program.

**Program:**

# 3.2 Write a python code to find the biggest number among three numbers..

num1=int(input('Enter the first number'))

num2=int(input('Enter the Second number'))

num3=int(input('Enter the Third number'))

if(num1>num2)and (num1>num3):

print(num1,' is greater than ',num2 ,' and ', num3)

elif(num2>num3):

print(num2,' is greater than ',num1 ,' and ', num3)

else:

print(num3,' is greater than ',num2 ,' and ', num1)

**Output:**

Enter the first number15

Enter the Second number46

Enter the Third number89

89 is greater than 46 and 15

**Result:**

This python code to find biggest among three numbers is executed successfully.

**Ex.No.3.3 To check the given number is positive , Negative or Zero**

**Aim:**

To write a python code to check the given number is positive, negative or zero.

**Algorithm:**

Step 1: Start the program.

Step 2: Read the value for num1.

Step 3: Check the condition if num1 is equal to 0.

Step 4: print(‘The given number’, num1,’ is zero’)

Step 5: Elif condition to check num1>0 and print(‘The given number’ ,num1,’ is negative’)

Step 6: If both the conditions are failed using else statement to print (‘ The given number ‘,num1,’ is poistive’)

Step 7: Stop the program.

**Program:**

# 3.2 Write a python code to check the given number is Positive or Negative or Zero..

num1=int(input('Enter the number:'))

if(num1==0):

print('The given number ',num1,' is Zero')

elif(num1>0):

print('The given number ',num1,' is Positive')

else:

print('The given number ',num1,' is Negative')

**Output:**

Enter the number:5

The given number 5 is Positive

Enter the number:0

The given number 0 is Zero

Enter the number:-5

The given number -5 is Positive

Ex.No.3.4 Grade for the given mark

Aim:

To write a python code to grade for a given mark.

Algorithm:

Step 1: Start the program.

Step 2: Input the value for mark1.

Step 3: Check the condition if(mark1>90 and mark1<=100)

Print(‘S Grade’)

Step 4: Use elif to check (mark1>80 and mark1 <=90

Print(‘A+ Grade’)

Step 5: use elif to check (mark1>70 and mark1<=80)

Print(‘B+ Grade’)

Step 6: Also check(mark1>=50 and mark1<=60) and print(‘B Grade’).

Step 7: Use else statements, if both are fails, print (‘RA’) as reappear.

Step 8: Stop the program.

**Program:**

# 3.2 Write a python code to find the grade for a given mark

mark1=int(input('Enter the mark1:'))

if(mark1>90 and mark1<=100):

print('S Grade')

elif(mark1>80 and mark1<=90):

print('A+ Grade')

elif(mark1>70 and mark1<=80):

print('A Grade')

elif(mark1>60 and mark1<=70):

print('B+ Grade')

elif(mark1>=50 and mark1<=60):

print('B Grade')

else:

print('RA')

**Output:**

Enter the mark1:45

RA

Enter the mark1:75

A Grade

Result: This python code to grade for a given mark is executed successfully.

Ex.No.3.5 Given year is leap year or not

**Aim:**

To write a python program to find the given year is leap year or not.

**Algorithm:**

Step 1: Start the program

Step 2: Read the value for the year1.

Step 3: Check the condition if((year1 %4 ==0) and (year1%100 !=0)) and print(‘Leap Year..’)

Step 4: Else print(‘Given year is not a Leap Year’)

Step 5: Stop the program.

**Program:**

# 3.5 Write a python code to find the given year is Leap or NOT..

year1=int(input('Enter the year:'))

if((year1%4 ==0) and (year1%100 !=0)):

print('Leap Year')

else:

print('Not a Leap Year')

**Output:**

Enter the year:2005

Not a Leap Year

Enter the year:2004

Leap Year

Result: This python code to find the given year is LEAP or NOT.

**Ex.No. 4.1 To print the first 10 natural numbers.**

Aim:

To write a python code to print the first 10 natural numbers.

Algorithm:

Step 1: Start the program.

Step 2: Input the value of num1 as range value 10.

Step 3: Initialize i=1.

Step 4: Print the first 10 natural number.

Step 5: Use while loop (i<=num1)

Print I and increment value of I as i=i+1

Step 6: Stop the program.

**Program:**

# 4.1 Write a python code to print the first 10 numbers

num1=int(input('Enter the range value 10:'))

i=1

print('The first 10 Natural Number:')

while(i<=num1):

print(i)

i=i+1

print('End of the program..')

**Output:**

Enter the range value 10:10

The first 10 Natural Number:

1

2

3

4

5

6

7

8

9

10

End of the program..

Result:

This python code to print the first 10 natural numbers is executed successfully.

Ex.No.4.2 The sum and average of first N

Aim:

To find the python code to the sum and average of first N numbers.

Algorithm:

Step 1: Start the program

Step 2: Input the value for num1.

Step 3: Initialize sum1=0,average=0.0

Step 4: Use for loop in range (1,num1+1,1) calculate sum1=sum1+i and average1=sum/sum1

Stop 5: Print the value for sum1 and average1.

Step 6: Stop the program.

**Program:**

# 4.2 Write a python code to find the sum and average of first n numbers.

num1=int(input('Enter the value for n:'))

sum1=0

average1=0.0

for i in range(1,num1+1,1):

sum1=sum1+i

average1=sum1/num1

print('Sum of first n numbers=',sum1)

print('Average of first n numbers=',average1)

**Output:**

Enter the value for n:10

Sum of first n numbers= 55

Average of first n numbers= 5.5

Ex.No.4.3 Check the given number is prime or not

Aim:

To write a python code to check the given number is prime or not.

Algorithm:

Step 1: Start the program

Step 2: Read the value for num1.

Step 3: Initialize i=2, prime=0 and use while loop(i<=num//2) and if condition to check(num%i ==0)

Step 4: Assign prime=1 and break the loop.

Step 5: Assign i=i+1 and check if(prime==1)

Step 6: Print(‘Num1 is prime number’), else print(‘Num1 is not a prime number’)

Step 7: Stop the program.

**Program:**

# 4.2 Write a python code to checka given number is PRIME or NOT

num1=int(input('Enter the number greater than or equal to 2:'))

i=2

prime=0

while(i<= num1//2):

if(num1%i == 0):

prime=1

break

i=i+1

if(prime==0):

print(num1,' is Prime Number')

else:

print(num1,' is NOT a Prime Number')

**Output:**

Enter the number greater than or equal to 2: 86

86 is NOT a Prime Number

Enter the number greater than or equal to 2: 7

7 is Prime Number

Result:

This python code created for find the given number is PRIME or NOT and executed successfully.

Ex.No.4.4 To find the GCD of two numbers using Euclidian algorithm.

Aim:

To write a python code to find the GCD of two numbers using Euclidian algorithm.

Alogorithm:

Step 1: Start the program.

Step 2: Input the values for num1 and num2.

Step 3: num1 and num2 is the temporary variable1 and temporary variable2.

Step 4: Check the condition, if(num2>num1)

Assign num1,num2=num2,num1

Step 5: While loop used to check (num2>70) and assign num1,num2=num2,num1.

Step 6 Print GCD of temp1 and temp2 is num1.

Step 7: Stop the program.

**Program**

# 4.4 Write a python code to find the GCD of two numbes using eucledian algorithm

num1=int(input('Enter the First Number:'))

num2=int(input('Enter the Second Number:'))

temp1=num1

temp2=num2

if(num2>num1):

num1,num2=num2,num1

while(num2>0):

num1,num2=num2,num1%num2

print('GCD of ',temp1,' and ',temp2,' is ',num1)

**Output:**

Enter the First Number:116

Enter the Second Number:72

GCD of 116 and 72 is 4

**Result:**This python code to find the GCD of two number sing Euclidian algorithm is executed successfully.

**Ex.No.4.5 To find the square root of a given number using Newton method.**

**Aim:**

To write a python code to find the square root of a given number using newton method.

**Algorithm:**

Step 1: Start the program.

Step 2: Read the value for First number a.

Step 3: Read the value for Second number b.

Step 4: Calculate guess=0.5\*a

Step 5: Generate a loop for I from 0 to b

Step 6: Calculate the value of guess by using the formula guess-(guess+a%guess)/2

Step 7: Print guess

Step 8: Stop the program.

**Program:**

# 4.5 Write a python code to find the squareoot of a given number using NEWTONs method.

a=int(input('Enter the First number:'))

b=int(input('Enter the Second number:'))

guess=0.5\*a

for i in range(b+1):

guess=(guess+a/guess)/2

print('Square root of ',a,'= ',guess)

**Output:**

Enter the First number:4

Enter the Second number:5

Square root of 4 = 2.0

**Result:**

This python code to find the square root of a given number using newton’s method is executed successfully.

**Ex.No.4.6 To Circulate the value of N numbers**

**Aim:**

To write a python code to calculate the value of N numbers.

**Algorithm:**

Step 1: Start the program.

Step 2: Create an empty list for list1.

Step 3: Input the total number of elements.

Step 4: Using loop statement for I in range of num and input the value for num1

Step 5: Append the element in num1 to list1.

Step 6: Print the entered the element in list1.

Step 7: Print the circulating the element in list1.

Step 8: Using another loop to store the element in temporary.

Step 9: Assign temp=list1(pop(0) and list1.append(temp)

Step 10: print(list1)

**Program:**

# 4.6 Write a python code to Circulate the value of N Numbers

list1=[]

num=int(input('Enter the Total Number of Elements:'))

for i in range(num):

num1=int(input())

list1.append(num1)

print('Entered Elements =',list1)

print('\n Circulating the Elements in the list:\n')

for i in range(num):

temp=list1.pop(0)

list1.append(temp)

print(list1)

**Ouput:**

Enter the Total Number of Elements:5

45

56

82

14

60

Entered Elements = [45, 56, 82, 14, 60]

Circulating the Elements in the list:

[56, 82, 14, 60, 45]

[82, 14, 60, 45, 56]

[14, 60, 45, 56, 82]

[60, 45, 56, 82, 14]

[45, 56, 82, 14, 60]

**Result:**

​This python code to circulate value of N numbers is executed successfully.

**Ex.No. 5.1 To check whether the given number is palindrome or Not.**

**Aim:**

To write a python code to check whether the given number is palindrome or Not.

**Algorithm:**

Step 1: Start the program.

Step 2: Assign palind = num1

Step 3: Assign rev=0 and while loop to check (num>0):

Step 4: Calculate rem=num%10

Step 5: Assign rev=rev \* 10 + num

Step 6: num1=num//10

Step 7: Check the condition if(palind==rev) : and print(‘The given number is palindrowm’)

Step 8: Else, print the given number is not a palindrome.

Step 9: Stop the program.

**Program:**

# 5.1 Write a python code to check whether a given number is palindrome or NOT.

num1=int(input('Enter a Number:'))

palind=num1

rev=0

while num1>0:

rem=num1%10

num1=num1//10

rev=rev\*10 + rem

if(palind==rev):

print('The given number is palindrome..')

else:

print('The given number is not a palindrome..')

**Output:**

Enter a Number:1221

The given number is palindrome..

Enter a Number:2546

The given number is not a palindrome..

**Result:**

This python code to check whether a given number is palindrome or not is executed successfully.

Ex.No. 5.2 To find the given number is Armstrong or not

Aim:

To write a python code to find the given number is Armstrong or Not

Algorithm:

Step 1: Start the program.

Step 2: Enter the number.

Step 3: Initialize s=0, sum=n.

Step 4: Use while loop (n>0)

Step 5: Calculate remainder value as r=n%10, s=s+(r\*\*3), n=n//10

Step 6: Check the condition if(s==num):

Step 7: Print the number is armstrong

Step 8: else, print number is not a Armstrong.

Step 9: Stop the program

**Program:**

# 5.2 Write a python code to find the given number is amstrong or Not.

n=int(input('Enter the number:'))

s=0

num=n

while(n>0):

r=n%10

s=s+(r\*\*3)

n=n//10

if(s==num):

print('The number is Amstrong')

else:

print('The number is not Amstrong')

**Output:**

Enter the number:45

The number is not Amstrong

Enter the number:153

The number is Amstrong

**Result:**

This python code to find the given number is amstrong or Not is executed successfully.

**Ex.No. 5.3 To print first N Prime Numbers**

**Aim:**

To write a python code to print first N Prime Numbers.

Algorithm:

Step 1: Start the program.

Step 2: Read the value for limit.

Step 3: Use for loop I in range (2, limit +1):

Step 4: Assign k=0

Step 5: Use another loop as j in range(2, i//2 + 1):

Step 6: Condition to check if(i%j) ==0)

Step 7: Increase the ‘k’ value as k=k+1 and break and check if(k==0) & print(i)

Step 8: Stop the program.

**Program:**

# 5.3 Write a python code to print first n prime number

limit=int(input('Enter a number:'))

for i in range(2,limit+1):

k=0

for j in range(2,i//2+1):

if(i%j==0):

k=k+1

break

if(k==0):

print(i)

**Output:**

Enter a number:27

2

3

5

7

11

13

17

19

23

**Result:**

This python code to print first n prime numbers is executed successfully.

**Ex.No.5.4 To print the numbers in Triangle pattern**

**Aim:**

To write a python code to print the numbers in Triangle pattern.

**Algorithm:**

Step 1: Start the program

Step 2: Read the value for num1

Step 3: Use for loop I in range of (1,num1+1):

Step 4: To add a new link.

Step 5: Use for loop j in range of (1,i+1) and print j.

Step 6: Stop the program.

**Program:**

# 5.4 Write a python code to print the numbers in Triangle Pattern.

num1=int(input('Enter the limit for pattern:'))

for i in range(1,num1+1):

print()

for j in range(1,i+1):

print(j,end=" ")

Enter the limit for pattern:6

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

Result:

This python code created for print the given range of numbers in Triangular format and executed successfully.

**Ex.No.6.1 To calculate factorial of a number using looping**

**Aim:**

Step 1: Start the program

Step 2: Assign factorial =1

Step 3: Condition to check if (num1<0):

Step 4: Print factorial doesn’t exist for negative value.

Step 5: Use elif to check (num==0): and print the factorial for 0 is 1

Step 6: Else, use for I range (1,num+1):

Assign factorial = factorial \*i

Step 7: Print (‘The factorial of a num1 is ‘ , factorial1)

Step 8: Stop the Program.

**Program:**

# 6.1 Write a python code to calculate factorial of a number using looping.

num=int(input('Enter a number:'))

factorial=1

if(num<0):

print('Sorry.. Factorial does not exists for negative value')

elif (num==0):

print('The factorial of 0 is 1')

else:

for i in range(1,num+1):

factorial=factorial\*i

print('The factorial of ',num, ' is ',factorial )

**Output:**

Enter a number:5

The factorial of 5 is 120

**Result:**

This python code to calculate factorial of a number using looping is executed successfully.

**Ex.No.6.2 Factorial of a given number using recursion.**

**Aim:**

To write a python code to factorial of a given number using recursion.

**Algorithm:**

Step 1: Start the program.

Step 2: Use and call def function of factorial(n)

Step 3: Check the condition if((n==1) or (n==0)):

Step 4: Return 1

Step 5: Else, return the value (n \* factorial (n-1))

Step 6: Read the value for num1

Step 7: Print the factorial value as factorial of n-1.

Step 8: Stop the program.

**Program:**

# 6.1 Write a python code to calculate the factorial of a given number sing recursion

def factorial(n):

if(n==1) or (n==0):

return 1

else:

return(n\*factorial(n-1))

n=int(input('Enter the number:'))

print('Factorial of ',n,' is',factorial(n))

**Output:**

Enter the number:5

Factorial of 5 is 120

**Result:** This python code to calculate factorial value of a given number using recursion function and executed successfully.

**Ex.No.6.3 To print the Number divisible by 2 and not divisible by 3 and 5.**

**Aim:**

To write a python code to print the number divisible by 2 and not by 3 and 5.

**Algorithm:**

Step 1: Start the program.

Step 2: Read the values for num1.

Step 3: Assign count=0

Step 4: Use for loop that I in range(I,num+1):

Step 5: Use if statement to check((i%2==0) and (i%3!=0 and i%5!=0)):

Step 6: Print i

Step 7: print (‘Total numbers between 1 to 100 which is divisible by 2 not by 3 and 5’

Step 8: Stop the Program.

**Program:**

# 6.2 Write a python code to list the numbers between 0 to 100 divisible by 2 and not by 3,5

num1=int(input('\n Enter the Maximum range as 100='))

count=0

for i in range(1,num1+1):

if (i%2==0 and (i%3!=0 and i%5 !=0)):

print(i,' is divisible by 2 not by 3 and 5')

count=count+1

print('Total numbers between 1 to 100 divisible by 2 and not by 3 and 5',count)

Output:

Enter the Maximum range as 100=55

2 is divisible by 2 not by 3 and 5

4 is divisible by 2 not by 3 and 5

8 is divisible by 2 not by 3 and 5

14 is divisible by 2 not by 3 and 5

16 is divisible by 2 not by 3 and 5

22 is divisible by 2 not by 3 and 5

26 is divisible by 2 not by 3 and 5

28 is divisible by 2 not by 3 and 5

32 is divisible by 2 not by 3 and 5

34 is divisible by 2 not by 3 and 5

38 is divisible by 2 not by 3 and 5

44 is divisible by 2 not by 3 and 5

46 is divisible by 2 not by 3 and 5

52 is divisible by 2 not by 3 and 5

Total numbers between 1 to 100 divisible by 2 and not by 3 and 5 14

**Result:** This python code to print the numbers divisible by 2 and not by 3 and 5 is executed successfully.

Ex.No.6.4 To find the square root and cube of a number using function

Aim:

To write a python code to find the square root and cube of a number using function.

Algorithm:

Step 1: Start the program.

Step 2: Enter the value for num1.

Step 3: Call def function square2(num1):

Square1=num1\*\*2

Return(square1)

Step 4: Call def function for cube2(num1) and cube1=num1\*\*3

Step 5: Return the value for cube1

Step 6: Call square1=Square2(num1), cube1=cube2(num1)

Step 7: print the values for square1 and cube1.

Step 8: Stop the program.

**Program:**

# 6.3 Write a python code to find square and cube of a number using function.

def square\_func(num1):

square1=num1\*\*2

return(square1)

def cube\_func(num1):

cube1=num1\*\*3

return(cube1)

num1=int(input('Enter any Number:'))

square1=square\_func(num1)

cube1=cube\_func(num1)

print('\nThe square root of given number ', num1,' is ', square1)

print('\n The cube of given number ',num1,' is ',cube1)

**Output:**

Enter any Number:5

The square root of given number 5 is 25

The cube of given number 5 is 125

**Result:**

This python code to find square and cube of a number using function is executed successfully.

Ex.No.7.1 To perform various built in string methods and functions.

Aim:

To write a python code to perform various built in string methods and functions.

Algorithm:

Step 1: Start the program.

Step 2: Read the string for string1

Step 3: Print the result by using built in function and methods. max(str1), min(str1)

Step 4: length(str1), str1.upper(),str1.lower(),str1.isupper(),str1.islower(),str1.capitialize()

Step 5: Stop the program.

**Program:**

# 7.1 Write a python code to perform various built in string methods and function

str1=input('Enter the String:')

print('\nMaximum =',max(str1))

print('\nMinimum =',min(str1))

print('\nLength =',len(str1))

print('\nUpper = ',str1.upper())

print('\nLower = ',str1.lower())

print('\n isupper=',str1.isupper())

print('\n islower=',str1.islower())

print('\n capitalize=',str1.capitalize())

**Output:**

Enter the String:Kamaraj

Maximum = r

Minimum = K

Length = 7

Upper = KAMARAJ

Lower = kamaraj

isupper= False

islower= False

capitalize= Kamaraj

**Result:**

This python code to perform various built in string methods and function is executed successfully.

**Ex.No.7.2 To reverse the string without using built-in functions.**

**Aim:**

To write a python code to reverse the string without using built-in functions.

**Algorithm:**

Step 1: Start the program.

Step 2: Call def function rev\_str1(str1):

Step 3: str2=” “

Step 4: Initialize i=len(str1)-1

Step 5: Use while loop(i>=0)

Step 6: str2=str2+str1[i] is concatenation the str2+str1[i]

Step 7: Increase the value of i=i+1

Step 8: Print the string2

Step 9: Print the string1

Step 10: Stop the program.

**Program:**

# 7.2 Write a python code to reverse the string without using built-in function

def rev\_str(str1):

str2=" "

i=len(str1)-1

while(i>=0):

str2=str2+str1[i]

i=i-1

print('The reverse string:',str2)

str1=input("Enter the string:")

rev\_str(str1)

**Output:**

Enter the string:KAMARAJ

The reverse string: JARAMAK

Ex.No.7.3 To print n terms of Fibonacci series

Aim:

To write a python code to print first n terms of Fibonacci series

Algorithm:

Step 1: Start the program.

Step 2: Assign num1=0

Step 3: Assign num2=1

Step 4: Print the Fibonacci series

Step 5: Check the condition, if(limit1==1), print(num1) and else print num1-num2

Step 6: Assign count=2

Step 7: Use while loop(count<unit1):

Increase num3=num1+num2 and assign num1=num2

Num2=num3 and count=count+1 to increase.

Step 8: Get the value for limit1, if(limit1==0), print the value.

Step 9: Else print fib(limit1)

Step 10: Stop the program.

Program:

# 7.3 Write a python code to print first n terms of fibonacci series

# Functon declaration

def fibo1(limit1):

num1=0

num2=1

print('The fibonacci series')

if(limit1==1):

prit(num1)

else:

print(num1)

print(num2)

count=2

while(count<limit1):

num3=num1+num2

print(num3)

num1=num2

num2=num3

count=count+1

# Main program

limit1=int(input('Enter the limit value:'))

if(limit1==0):

print('Enter the positive value:')

else:

fibo1(limit1)

**Output:**

Enter the limit value:10

The fibonacci series

0

1

1

2

3

5

8

13

21

34

**Result:** This python code to print first n terms of Fibonacci series is executed successfully.

Ex.No.7.4 To add and multiply two numbers using lambda function

Aim: To write a python code to add and multiply two number using lambda function.

Algorithm:

Step 1: Start the program.

Step 2: Assign num1=lambda num1, num2=num1+num2 and mult1=lambda num1, num2=num1 \* num2.

Step 3: Read the values for num1 and num2

Step 4: Print the value for print ( ‘addition of two numbers using lambda function’, sum1(num1,num2))

Step 5: Print the integer value.

Step 6: Stop the program.

**Program:**

# 7.3 Write a python code to add and multiply two Numbers using Lambda function.

sum1=lambda num1,num2:num1+num2

mult1=lambda num1,num2:num1\*num2

num1=int(input('Enter the First number:'))

num2=int(input('Enter the Second number'))

print('Addition of two numbers using lamda function:',sum1(num1,num2))

print('Multiplication of two numbers using lambda function:',mult1(num1,num2))

**Output:**

Enter the First number:10

Enter the Second number20

Addition of two numbers using lamda function: 30

Multiplication of two numbers using lambda function: 200

**Result:**

This python code to add and multiply two numbers using lambda function is executed successfully.

Ex.No.8.1 To find the maximum and minimum number in the list

Aim:

To write a python code to find the maximum and minimum number in the list.

Algorithm:

Step 1: Start the program.

Step 2: Input the values in list1.

Step 3: Use for loop I in range (0,num1)

Step 4: Input the value for a.

Step 5: list1.append(a). Append the value of ‘a’ in list1.

Step 6: Assign max1=list1[0], min1=list1[0].

Step 7: Use for loop I in range(1,num1): if list1[i] is greater max1, max1=list1[i] and check if list1[i] is less than min1, min1 is equal to list1[i]

Step 8: Print the value for maximum and minimum number

Step 9: Stop the program.

**Program:**

# 8.1 Write a python code to find the maximum and the minimum number in the list

list1=[]

num1=int(input('Enter the total number of elements:'))

for i in range(0,num1):

a=int(input())

list1.append(a)

max1=list1[0]

min1=list1[0]

for i in range(1,num1):

if (list1[i]>max1):

max1=list1[i]

if (list1[i]<min1):

min1=list1[i]

print('Maxmimum Number of the list:',max1)

print('Minimum Number of the list:',min1)

**Output:**

Enter the total number of elements:5

10

50

48097

2

5

Maxmimum Number of the list: 48097

Minimum Number of the list: 2

Result:

This python code to find the maximum and minimum in the list is executed successfully.

Ex.No. 8.2 To perform Linear search using function

Aim:

To write a python code to perform the linear search using function.

Algorithm:

Step 1: Start the program. Create a new list as empty/

Step 2: Input the values for num1.

Step 3: Print the element one by one and for 2 in range (0,num1)

Step 4: Input the value for num2.

Step 5: Append the value of num2 in list1 and input the value for search.

Step 6: Use call def function in linear search (list1,search)

And initialize flag=0. Use for loop I in range len(list1)

Step 7: Check if(list[i]== search)

Step 8: print(‘The search element’, search, ‘ is found in to locatin’)

Step 9: Assign flag=1 and break theloop.

Step 10: Check if(flag==0)

Step 11: Print the search element is not found.

Step 12: Stop the program.

**Program:**

# 8.2 Write a python code to perform the linear serach

def linear\_search(list1,search):

flag=0

for i in range(len(list1)):

if(list1[i]==search):

print('The Element ',search,' is found in ',i,' Location.')

flag=1

break

if(flag==0):

print('The Element ', search,' is not found..')

# main program starting here..

list1=[]

num1=int(input('\nEnter the total number of elements:'))

print('\n Enter the elements one by one.')

for i in range(0,num1):

num2=int(input())

list1.append(num2)

search=int(input('Enter the number to search:'))

linear\_search(list1,search)

**Output:**

Enter the total number of elements:6

Enter the elements one by one.

87

98

54

65

21

12

Enter the number to search:54

The Element 54 is found in 2 Location.

**Result:**

This python code to perform linear search is executed successfully.

**Ex.No. 8.3 To perform binary search using function.**

**Aim:**

To write a python code to perform binary search using function.

**Algorithm:**

Step 1: Start the program.

Step 2: Create a empty list list1=[]

Step 3: Initialize flag=0

Step 4: Input the total input elements.

Step 5: Enter the elements one by one.

Step 6: For I in range(num1)

Step 7: Input the value of num2 and append it to the list1.

Step 8: Initialize low=0, high=len(list1)-1

Step 9: Enter the element to search.

Step 10: While(low<=high> and mid=(low+hig)//2

If(list[mid]==search): flag=1

Step 11: Print (“The search element”, element of search is found in the mid position)

Step 12: Break the loop elif (search<list1[mid] and high=mid+1, else: low=mid+1

Step 13: If (flag==0) and print the search is not found.

Step 14: Stop the program.

**Program:**

# 8.3 Write a python code to perform binary search..

list1=[]

flag=0

num1=int(input('Enter the total input elements='))

print('Enter the element one by one')

for i in range(num1):

num2=int(input())

list1.append(num2)

low=0

high=len(list1)-1

search=int(input('Enter the element to search:'))

while(low<=high):

mid=(low+high)//2

if(list1[mid]==search):

flag=1

print('The search Element ',search,' is found in ',mid,' position')

break

elif(search<list1[mid]):

high=mid-1

else:

low=mid+1

if(flag==0):

print('The search element ',search,' is not found')

Output:

Enter the total input elements=6

Enter the element one by one

5

4

7

8

9

3

Enter the element to search:7

The search Element 7 is found in 2 position

**Result:**

This python code to perform Binary search is executed successfully.

**Ex.No.8.4 To perform matrix addition**

**Aim:**

To write a python code to perform matrix addition.

**Algorithm:**

Step 1: Start the program.

Step 2: Create the empty lists 🡺 X[][],Y[][] and result[][]

Step 3: Get the input for m and n from user for no. of row and column.

Step 4: Get input for matrix – A[][] and matrix – B[][]

Step 5: Print the value of First and Second matrix.

Step 6: Calculate the value for result[i][j]=A[i][j]+B[i][j].

Step 7: Print the value of Result matrix.

Step 8: Stop the program.

**Program:**

m=int(input('ENTER MARTIX ROW SIZE m : '))

n=int(input('ENTER MARTIX COLUMN SIZE n : '))

#initializing matrix elements as 0

X = [[0]\*n for j in range(m)]

Y = [[0]\*n for j in range(m)]

result = [[0]\*n for j in range(m)]

print ('INPUT-FIRST MATRIX : ')

#getting input to matrix X

for i in range (m):

for j in range (n):

print ('entry in row: ',i+1,' column: ',j+1)

X[i][j] = int(input())

print ('INPUT-SECOND MATRIX : ')

#getting input to matrix X

for i in range (m):

for j in range (n):

print ('entry in row: ',i+1,' column: ',j+1)

Y[i][j] = int(input())

#printing first matrix X

print ('PRINT-FIRST MATRIX : ')

for i in range (m):

for j in range (n):

print (X[i][j],end='\t')

print('\n')

print ('PRINT-SECOND MATRIX : ')

#printing second matrix Y

for i in range (m):

for j in range (n):

print (Y[i][j],end='\t')

print ('\n')

#adding X and Y to result

for i in range(len(X)):

for j in range(len(X[0])):

result[i][j] = X[i][j] + Y[i][j]

#displaying result

print ('SUM OF MATRICES IS : ')

for i in range (m):

for j in range (n):

print (result[i][j],end='\t')

print ('\n')

**Output:**

ENTER MARTIX ROW SIZE m : 3

ENTER MARTIX COLUMN SIZE n : 3

INPUT-FIRST MATRIX :

entry in row: 1 column: 1

1

entry in row: 1 column: 2

2

entry in row: 1 column: 3

3

entry in row: 2 column: 1

6

entry in row: 2 column: 2

5

entry in row: 2 column: 3

4

entry in row: 3 column: 1

7

entry in row: 3 column: 2

8

entry in row: 3 column: 3

9

INPUT-SECOND MATRIX :

entry in row: 1 column: 1

9

entry in row: 1 column: 2

6

entry in row: 1 column: 3

3

entry in row: 2 column: 1

2

entry in row: 2 column: 2

5

entry in row: 2 column: 3

8

entry in row: 3 column: 1

7

entry in row: 3 column: 2

4

entry in row: 3 column: 3

1

PRINT-FIRST MATRIX :

1 2 3

6 5 4

7 8 9

PRINT-SECOND MATRIX :

9 6 3

2 5 8

7 4 1

SUM OF MATRICES IS :

10 8 6

8 10 12

14 12 10

**Result:**

This python code to perform matrix addition is executed successfully.

**Ex.No.8.5 To perform matrix Multiplication**

**Aim:**

To write a python code to perform matrix multiplication.

**Algorithm:**

Step 1: Start the program.

Step 2: Create the empty lists 🡺 X[][],Y[][] and result[][]

Step 3: Get the input for m and n from user for no. of row and column.

Step 4: Get input for matrix – A[][] and matrix – B[][]

Step 5: Print the value of First and Second matrix.

Step 6: Calculate the value using three for loops.

for i in range(m):

for j in range(n):

result[i][j]=0

for k in range(m):

result[i][j] = result[i][j]+X[i][k] + Y[k][j]

Step 7: Print the value of Result matrix.

Step 8: Stop the program.

**Program:**

m=int(input('ENTER MARTIX ROW SIZE m : '))

n=int(input('ENTER MARTIX COLUMN SIZE n : '))

#initializing matrix elements as 0

X = [[0]\*n for j in range(m)]

Y = [[0]\*n for j in range(m)]

result = [[0]\*n for j in range(m)]

print ('INPUT-FIRST MATRIX : ')

#getting input to matrix X

for i in range (m):

for j in range (n):

print ('entry in row: ',i+1,' column: ',j+1)

X[i][j] = int(input())

print ('INPUT-SECOND MATRIX : ')

#getting input to matrix X

for i in range (m):

for j in range (n):

print ('entry in row: ',i+1,' column: ',j+1)

Y[i][j] = int(input())

#printing first matrix X

print ('PRINT-FIRST MATRIX : ')

for i in range (m):

for j in range (n):

print (X[i][j],end='\t')

print('\n')

print ('PRINT-SECOND MATRIX : ')

#printing second matrix Y

for i in range (m):

for j in range (n):

print (Y[i][j],end='\t')

print ('\n')

#Multiply X with Y to result

for i in range(m):

for j in range(n):

result[i][j]=0

for k in range(m):

result[i][j] = result[i][j]+X[i][k] + Y[k][j]

#displaying result

print ('MULTIPLY - MATRICES IS : ')

for i in range (m):

for j in range (n):

print (result[i][j],end='\t')

print ('\n')

**Output:**

ENTER MARTIX ROW SIZE m : 3

ENTER MARTIX COLUMN SIZE n : 3

INPUT-FIRST MATRIX :

entry in row: 1 column: 1

1

entry in row: 1 column: 2

2

entry in row: 1 column: 3

0

entry in row: 2 column: 1

0

entry in row: 2 column: 2

1

entry in row: 2 column: 3

1

entry in row: 3 column: 1

2

entry in row: 3 column: 2

0

entry in row: 3 column: 3

1

INPUT-SECOND MATRIX :

entry in row: 1 column: 1

1

entry in row: 1 column: 2

1

entry in row: 1 column: 3

2

entry in row: 2 column: 1

2

entry in row: 2 column: 2

1

entry in row: 2 column: 3

1

entry in row: 3 column: 1

1

entry in row: 3 column: 2

2

entry in row: 3 column: 3

1

PRINT-FIRST MATRIX :

1 2 0

0 1 1

2 0 1

PRINT-SECOND MATRIX :

1 1 2

2 1 1

1 2 1

MULTIPLY - MATRICES:

7 7 7

6 6 6

7 7 7

**Result:**

This python code to perform matrix multiplication is executed successfully.

**Ex.No. 9.1 To perform Built-in function with list**

**Aim:**

To write a python code to perform functionality of Built-in functions with list.

**Algorithm:**Step 1: Input the values in list1 and list2.

Step 2: Print the values for max(list1), min(list1), len(list1), sum(list1) and concatenation of list1 and list2.

Step 3: Print the repetition of list1, list1\*\*2.

Step 4: Print True/False for membership of 30 in lists.

Step 5: Print True/False for membership of 90 in list1, 90 not in list1.

Step 6: Print True/False for membership of 30 in list1, 30 not in list1.

Step 7: Print the counting the input element in list2, list2.count(300).

Step 8: Print the value for sorted of list1, and list2 and print sorting, appending, extending, reversing, count of list1.

Step 9: Stop the program.

**Program:**

# Ex.No. 9.1 Built-In function with list

list1=[10,30,70,80,66,90]

list2=[10,500,400,300,300,700]

print('Maximum in the list1=',max(list1),' list2=',max(list2))

print('Minimum in the list1=',min(list1),' list2=', min(list2))

print('Length of the list=', len(list1), 'and list2=',len(list2))

print('Sum of list1=', sum(list1),' and list2=',sum(list2))

print('Concatenation of the list1 and list2=',list1+list2)

#print('Repetition of the list1=',list1\*\*2)

print('Membership 30 in list1=',30 in list1)

print('Membership 30 not in list=', 30 not in list1)

print('Membership 400 in list2=',400 in list2)

print('Membership 400 not in list2=',400 not in list2)

print('Counting the input number in the list2=', list2.count(300))

print('Sorted of list1=',sorted(list1))

print('Sorted of list2=', sorted(list2))

list1.sort()

print('Sorting of list1=', list1)

list1.append(978)

print('After Append of list1=',list1)

list1.reverse()

print('Reversing of list1=',list1)

#list1.extend(70,30,80)

#print('Extending of list1=',list1)

cnt=list2.count(700)

print('Count 700 in list2=',cnt)

**Output:**

Maximum in the list1= 90 list2= 700

Minimum in the list1= 10 list2= 10

Length of the list= 6 and list2= 6

Sum of list1= 346 and list2= 2210

Concatenation of the list1 and list2= [10, 30, 70, 80, 66, 90, 10, 500, 400, 300, 300, 700]

Membership 30 in list1= True

Membership 30 not in list= False

Membership 400 in list2= True

Membership 400 not in list2= False

Counting the input number in the list2= 2

Sorted of list1= [10, 30, 66, 70, 80, 90]

Sorted of list2= [10, 300, 300, 400, 500, 700]

Sorting of list1= [10, 30, 66, 70, 80, 90]

After Append of list1= [10, 30, 66, 70, 80, 90, 978]

Reversing of list1= [978, 90, 80, 70, 66, 30, 10]

Count 700 in list2= 1

**Result:** This python code to perform various built-in functions with list is executed successfully.

Ex.No. 9.2 To search the element in the dictionary

Aim:

To write a python code to search the element in the dictionary

Algorithm:

Step 1: Start the program.

Step 2: Input the keys and values in students.

Step 3:Assign flag=0

Step 4: Print the Roll number and Name.

Step 5: Input the key to search.

Step 6: For I in student and the student of [i] and check the condition if(i==search):

Step 7: Enter the value of I and initialize flag=1

Step 8: print(“The search element is in the dictionary’, i,’is found’)

Step 9: Break keyword to stop the loop. Check if (flag==0).

Step 10: Print the search element is not found in the dictionary.

Step 11: Stop the program.

**Program:**

# Ex.No. 9.2 To search theelement in the dictionary

students={'19uit01':'XXXXX','19uit02':'YYYYY','19uite03':'ZZZZZ'}

flag=0

print('Roll Number =',students.keys())

print('Name=',students.values())

search=input('Enter the Roll number to search:')

for i in students:

students[i]

if(i==search):

flag=1

print('The search element ', search,' Name :',students[i],' is found')

break

if(flag==0):

print('The search element ', search,' is not found')

**Output:**

Roll Number = dict\_keys(['19uit01', '19uit02', '19uite03'])

Name= dict\_values(['XXXXX', 'YYYYY', 'ZZZZZ'])

Enter the Roll number to search:19uit02

The search element 19uit02 Name : YYYYY is found

Roll Number = dict\_keys(['19uit01', '19uit02', '19uite03'])

Name= dict\_values(['XXXXX', 'YYYYY', 'ZZZZZ'])

Enter the Roll number to search:19uit06

The search element 19uit06 is not found

**Result:**

This python code to search the element in the dictionary is executed successfully.

Ex.No.9.3 To perform command line argument for word count.

Aim:

To write a python code to perform command line argument for word count.

Algorithm:

Step 1: Start the program.

Step 2: import sys

Step 3: Input the string for program name.

Step 4: Enter the arguments.

Step 5: Assign count=len(argument)

Step 6: Print the file name, count value.

Step 7: Use for loop and I in sys.argv.

Step 8: print the value of i.

Step 9: Open a new terminal to run the command line argument.

Step 10: Stop the program.

**Program:**

# Ex.No. 9.2 To perform command line argument for word count

import sys

program\_name=sys.argv[0]

arguments=sys.argv[1:]

count=len(arguments)

print('File name:',program\_name)

print('Number of words in command line ',count)

for i in sys.argv:

print('Argument:',i)

**Output:**

$python commandline.py KAMARAJ KCET VIRUDHUNAGAR

(‘file name:’,’commandline.py’)

**Result**: This code to perform command line argument for word count is executed successfully.

**Ex.No. 9.4 To implement histogram to count the frequency of character using dictionary**

**Aim:**

To write a python program to implement histogram to count the frequency of character using dictionary.

**Algorithm:**

Step 1: Start the program.

Step 2: Input the string for str1.

Step 3: Create a empty dictionary as dict1.

Step 4: For i in str1, check the condition if I not in dict1, dict1[i]=1.

Step 5: Else, print dict1[i]=dict1[i]+1

Step 6: Print the keys, values and Frequency count.

Step 7: Use for loop I in dict1.

Step 8: Print the values of i, dict1[i], ‘\*’\*dict1[i]

Stop 9: Stop the program.

**Program:**

# Ex.No. 9.4 To Implement Histogram to count the Frequency of character using DICTIONARY.

str1='Welcome to Python Programming'

dict1={}

for i in str1:

if i not in dict1:

dict1[i]=1

else:

dict1[i]=dict1[i]+1

print('\n\n KEYS\t VALUES \t FREQUENCY COUNT')

for i in dict1:

print(i,'\t',dict1[i],'\t\t','\*'\*dict1[i])

**Output:**

KEYS VALUES FREQUENCY COUNT

W 1 \*

e 2 \*\*

l 1 \*

c 1 \*

o 4 \*\*\*\*

m 3 \*\*\*

3 \*\*\*

t 2 \*\*

P 2 \*\*

y 1 \*

h 1 \*

n 2 \*\*

r 2 \*\*

g 2 \*\*

a 1 \*

i 1 \*

**Result:**

This python code to implement histogram to count the frequency of characters using dictionary is executed successfully.

Ex.No.10.1 To perform selection sort using function.

**Algorithm:**

Step 1: Start the program.

Step 2: Input the value for num1.

Step 3: Enter the element one by one.

Step 4: Else for loop that i in range(num1) end enter the value for num2.

Step 5: Append the value of num2 in list1.

Step 6: Call the function of selection\_sort(list1)

Step 7: Print the input element before sorting.

Step 8: For I in range(num1-1) and min1=I is the initialize and use another for loop of j in range(i+1,num1) and if(list1[min]>list1[i] and min1=j and assign list1[i], list1[min]=list1[min],list[i].

Step 9: Print input elements after sorting of list1.

Step 10: Stop the program.

**Program:**

# Ex.No. 10.1 To do selection sort using function.

def selection\_sort(list1):

print('The input elements before sorting\n',list1)

for i in range(num1-1):

min1=i

for j in range(i+1,num1):

if(list1[min1]>list1[j]):

min1=j

list1[i],list1[min1]=list1[min1],list1[i]

print('The input element after sorting list..',list1)

# Main program starting here..

list1=[]

num1=int(input('Enter the total input elements='))

print('Enter the elements one by one=')

for i in range(num1):

num2=int(input())

list1.append(num2)

selection\_sort(list1)

**Output:**

Enter the total input elements=5

Enter the elements one by one=

12

98

56

34

75

The input elements before sorting

[12, 98, 56, 34, 75]

The input element after sorting list.. [12, 98, 56, 34, 75]

The input element after sorting list.. [12, 34, 56, 98, 75]

The input element after sorting list.. [12, 34, 56, 98, 75]

The input element after sorting list.. [12, 34, 56, 75, 98]

**Result:**

This python code to do selection sort using function is executed successfully.

**Ex.No.10.2 To perform Insertion sort using function**

**Aim:**

To write a python code to perform insertion sort using function.

**Algorithm:**

Step 1: Start the program.

Step 2: Create a empty list for list1.

Step 3: Enter the element one by one.

Step 4: Use for I in range(num1) and input the element of num2.

Step 5: Append the num2 in list1.

Step 6: Call insertion\_sort(list1).

Step 7: Print the input element after before sorting.

Step 8: All the element in the position from 0 to N are in ascending order.

Step 9: At each sequence, we move the element in the position N until the correct position found.

Step 10: Swap the larger number position in the right.

Step 11: Print the input element after sorting.

Step 12: Stop the program.

**Program:**

# Ex.No. 10.2 To do Insertion sort using function.

def insertion\_sort(list1):

print('The input elements before sorting\n',list1)

for i in range(1,num1):

j=i

while(j>0)and(list1[j-1]>list1[j]):

temp=list1[j]

list1[j]=list1[j-1]

list1[j-1]=temp

j=j-1

print("The input elements after sorting\n",list1)

# main program starting here..

list1=[]

num1=int(input('Enter the total input elements='))

print('Enter the elements one by one..')

for i in range(num1):

num2=int(input())

list1.append(num2)

insertion\_sort(list1)

**Output:**

Enter the total input elements=5

Enter the elements one by one..

12

16

89

64

13

The input elements before sorting

[12, 16, 89, 64, 13]

The input elements after sorting

[12, 16, 89, 64, 13]

The input elements after sorting

[12, 16, 89, 64, 13]

The input elements after sorting

[12, 16, 64, 89, 13]

The input elements after sorting

[12, 13, 16, 64, 89]

**Result:**

This python code to perform insertion sort using function is executed successfully.

Ex.No.10.3 To perform merge sort using function

Aim:

To write a python code to perform merge sort using function.

Algorithm:

Step 1: Start the program.

Step 2: Create an empty list for list1

Step 3: Input the value for num1.

Step 4: Enter the element one by one and use for loop I in range (0,num)

Step 5: Input the value for num1 and append it to the list1.

Step 6: Print the element before sorting in list1.

Step 7: This is to merge tow sorted list.

Step 8: The elements are compared and moved to the third list.

Step 9: When one list is compared to and all the remaining element in the next list can be moved through the third list.

Step 10: Print the element before sorting.

Step 11: Stop the program.

Program:

def merge(left,right):

result=[]

i,j=0,0

while len(result)<len(left)+len(right):

if left[i]<right[j]:

result.append(left[i])

i+=1

else:

result.append(right[j])

j+=1

if i==len(left) or j==len(right):

result.extend(left[i:] or right[j:])

break

return result

def mergesort(list):

if len(list)<2:

return list

mid=len(list)//2

left=mergesort(list[:mid])

right=mergesort(list[mid:])

return merge(left,right)

n=int(input('Enter the total element:'))

list1=[]

print('\nEnter the elements one by one..')

for i in range(0,n):

list1.append(int(input()))

print("\nBefort sorting..")

print(list1)

print("\nAfter Sorting..")

r=mergesort(list1)

print(r)

**Output:**

Enter the total element:5

Enter the elements one by one..

45

56

85

2

78

Before sorting..

[45, 56, 85, 2, 78]

After Sorting..

[2, 45, 56, 78, 85]

Result:

This python code to perform merge sort using function is executed successfully.

Ex.No.11.1 To copy file1 contents into file2

Aim:

To write a python code to copy file1 contents into file2

Algorithm:

Step 1: Start the program.

Step 2: Open file1 and file2 in reading writing mode respectively.

Step 3: Use for loop i in fp1

fp2.write(i)

Step 4: Print file1 is copied to file2.

Step 5: fp1 and fp2 is closed.

Step 6: Stop the program.

Program:

# 10.3 To copy file1 contentsinto file2

fp1=open('mergesort.py','r')

fp2=open('copiedfile.py','w')

for i in fp1:

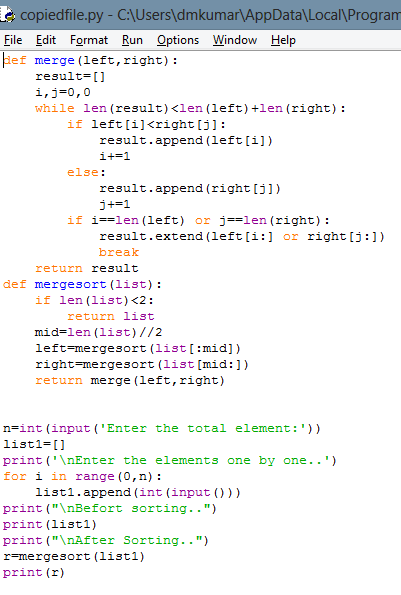
fp2.write(i)

print('\nFile1 is copied into file2')

fp1.close()

fp2.close()

**Output:**File1 is copied into file2

****

**Result:**

This python code to copy file1 contents into file2 is executed successfully.

Ex.No.11.2 To find the most frequent word in the given input file

Aim:

To write a python code to find the most frequent word in the given input file.

Algorithm:

Step 1: Start the program.

Step 2: Open file1 in reading mode.

Step 3: Create an empty dictionary for wordfreq.

Step 4: Use for i in file1.

Step 5: Assign word\_line.space()

Step 6: for i in words and check I in wordfreq

Wordfreq[i] = wordfreq[i]+1

Step 7: Else: wordfreq[1]=1

Step 8: for i in wordfreq

Step 9: print wordfreq[i]

Step 10: wordfreq1=sorted(wordfreq.values())

Step 11: Print wordfreq.

Step 12:File1 to be close and search –max(wordfreq)

Step 13: For i in wordfeq to check the condition

If(wordfreq[i]==search)

Step 14: Print the value of i and break.

Step 15: Stop the program

**Program:**

# 11.1 To find most frequent word in the given input file.

file1=open('input.py','r')

wordfreq={}

for line in file1:

words=line.split()

for i in words:

if i in wordfreq:

wordfreq[i]=wordfreq[i]+1

else:

wordfreq[i]=1

for i in wordfreq:

print(i,'\t',wordfreq[i])

wordfreq1=sorted(wordfreq.values())

print(wordfreq)

file1.close

search=max(wordfreq)

for i in wordfreq:

if(wordfreq[i]==search):

print('The most frequent words is:',i,search)

break

**Output:**

Hello 1

how 1

are 2

you 2

what 1

doing 1

now 1

{'Hello': 1, 'how': 1, 'are': 2, 'you': 2, 'what': 1, 'doing': 1, 'now': 1}

>>>

Result:

This python code to find most frequent word in given input file is executed successfully

Ex.No.12.1 To simulate Elliptical orbit in pygame

Aim:

To write a python code to simulate elliptical orbit in pygame.

Algorithm:

Step 1: Start the program.

Step 2: Import and initialize the pygame, math and sys.

Step 3: Set the method of pygame.init()

Step 4: Define the caption using set.caption method.

Step 5: Generate the loop for each event and if event type is quit then close the pygame.

Step 6: Assign circles of x radius and y radius with initial values.

Step 7: Generate a loop for range 0 to 360’ degree with ten steps.

Step 8: Calculate the degree of rotation with respect to x and y axis.

Step 9: Draw a circle and ellipse with predefined values.

Step 10: Set time using clock.tick() method.

Step 11: Stop the program.

Program:

#Elliptical orbits

import pygame

import math

import sys

pygame.init()

screen = pygame.display.set\_mode((600, 300))

pygame.display.set\_caption("Elliptical orbit")

clock = pygame.time.Clock()

while(True):

for event in pygame.event.get():

if event.type == pygame.QUIT:

sys.exit()

xRadius = 250

yRadius = 100

for degree in range(0,360,10):

x1 = int(math.cos(degree \* 2 \* math.pi / 360) \* xRadius) + 300

y1 = int(math.sin(degree \* 2 \* math.pi / 360) \* yRadius) + 150

screen.fill((0, 0, 0))

pygame.draw.circle(screen, (255, 0, 0), [300, 150], 35)

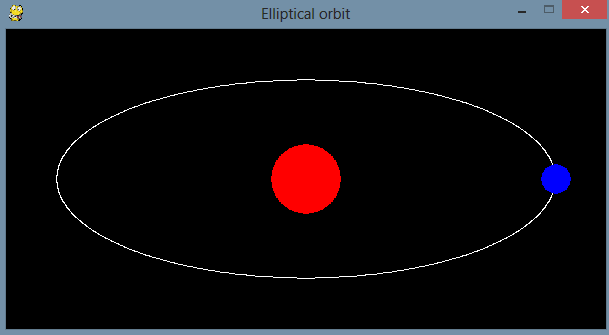
pygame.draw.ellipse(screen, (255, 255, 255), [50, 50, 500, 200], 1)

pygame.draw.circle(screen, (0, 0, 255), [x1, y1], 15)

pygame.display.flip()

clock.tick(5)

**Output:**



**Result:**

This python code to simulate elliptical orbit in pygame is executed successfully.

**Ex.No.12.2 To simulate bouncing ball using pygame.**

**Aim:**

To write a python code to simulate bouncing ball using pygame.

**Algorithm:**

Step 1: Start the program.

Step 2: Import and initialize the modules, pygame and sys.

Step 3: Set the pygame window width, height also set the speed in x,y direction and set background color.

Step 4: Create a graphical window with pygame display call.

Step 5: load call image.

Step 6: Create a rctangel over the call image. Get the pygame event. Repeat set a while the event is pygame.Quite.

Step 7: Erase the screen for next animation. Full the screen with background color. Draw the image is new screen with surface. Update the display calling the pygame display\_flip(). If the event is pygame. Quit().

Step 8: Step the program.

**Program:**

# 12.2 To simulate elliptical orbit in pygame

import pygame

import math

import sys

pygame.init()

size=width, height=1500,750

speed=[1,1]

black=0,0,0

screen=pygame.display.set\_mode(size)

ball=pygame.image.load('ball.jpg')

ballrect=ball.get\_rect()

while 1:

for event in pygame.event.get():

if event.type==pygame.QUIT:

sys.exit()

ballrect=ballrect.move(speed)

if ballrect.left<0 or ballrect.right>width:

speed[0]=speed[0]

if ballrect.top<0 or ballrect.bottom>height:

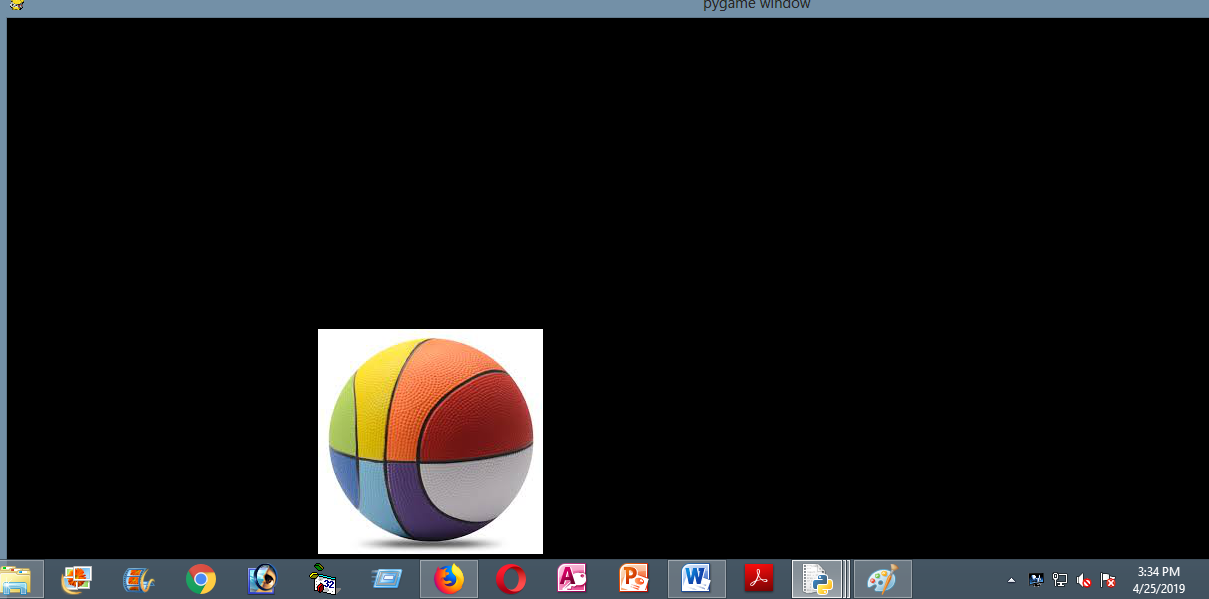
speed[1]=speed[1]

screen.fill(black)

screen.blit(ball,ballrect)

pygame.display.flip()

**Output:**



Result:

This python program to simulate bouncing ball using executed successfully.

Ex.No.13.1 To draw a square shape turtle.

Aim:

To write a python program to draw a square using turtile.

Algorithm:

Step 1: Start the program.

Step 2: Import turtle.

Step 3: Use turtle.forward() to move in straight and use turtle.right() to change the direction.

Step 4: Display the result

Step 5: Stop the program.

**Program:**

# 13.1 To draw a square shpae using turtle

import turtle

turtle.forward(50)

turtle.right(90)

turtle.forward(50)

turtle.right(90)

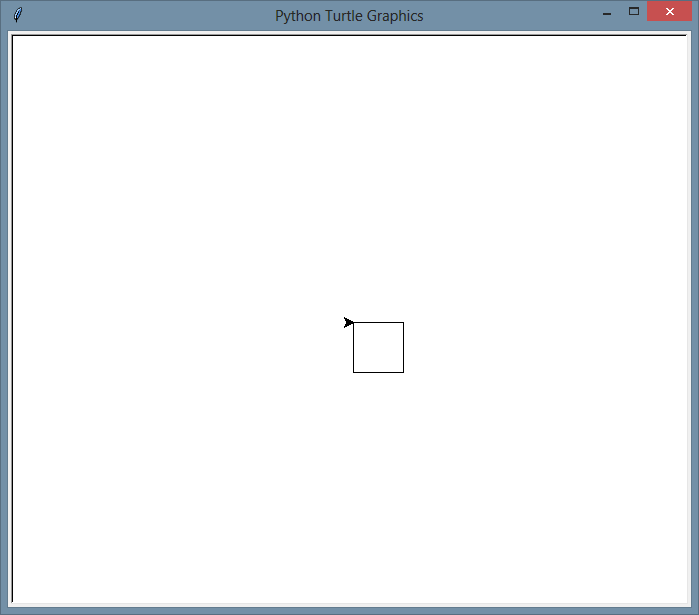
turtle.forward(50)

turtle.right(90)

turtle.forward(50)

turtle.right(90)

**Output:**



Result: Thus, the python program to draw a square shape using turtle is executed successfully.