**EX NO 1(A) PERFORM ARITHMETIC OPERATIONS USING TWO NUMBERS**

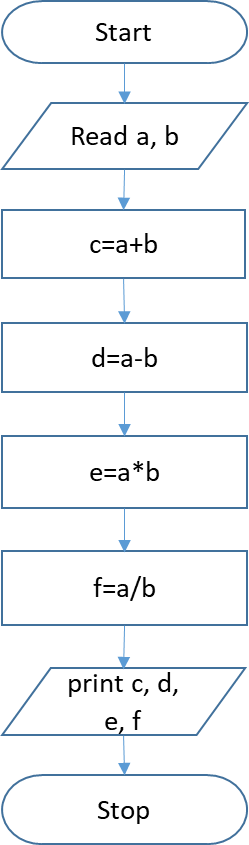
**AIM**

To write a program in Python to perform arithmetic operations using two numbers.

**ALGORITHM**

1. Start
2. Get the first number and second number using input function and convert into integer.
3. Perform addition, subtraction, multiplication and division using above two numbers.
4. Store the above operation results in separate variables.
5. Print the values of the variables.
6. Stop.

**FLOWCHART**

****

**PROGRAM**

a=int(input("Enter 1st number"))

b=int(input("Enter 2nd number"))

c=a+b

d=a-b

e=a\*b

f=a/b

print("Addition",c)

print("Subtraction",d)

print("Multiplication",e)

print("Division",f)

**OUTPUT**

Enter 1st number6

Enter 2nd number3

Addition 9

Subtraction 3

Multiplication 18

Division 2.0

**RESULT**

The above program in Python to perform arithmetic operations using two numbers has been executed successfully and the output was verified.

**EX NO 1(B) FIND THE GREATEST OF THREE NUMBERS**

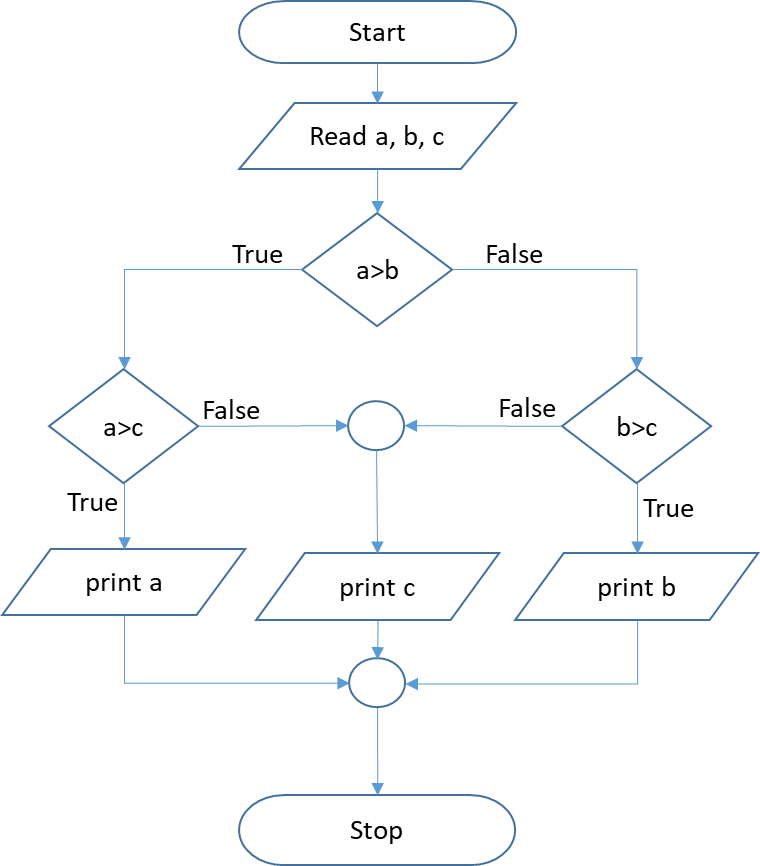
**AIM**

To write a program in Python to find greatest of three numbers.

**ALGORITHM**

1. Start
2. Get the three numbers using input function.
3. Check if a is greater than b and c.
4. If a is greater, print a is greatest.
5. Otherwise check if b is greater than c.
6. If b is greater, print b is greater.
7. Otherwise print c is greater.
8. Stop.

**FLOWCHART**

****

**PROGRAM**

a=input("Enter 1st number")

b=input("Enter 2nd number")

c=input("Enter 2nd number")

if a>b:

if a>c:

print(a)

else:

print(c)

elif b>c:

print(b)

else:

print(c)

**OUTPUT**

Enter 1st number9

Enter 2nd number5

Enter 2nd number10

10

**RESULT**

The above program in Python to find the greatest of three numbers has been executed successfully and the output was verified.

**EX NO 1(C) FIND THE SUM OF FIRST N NUMBERS**

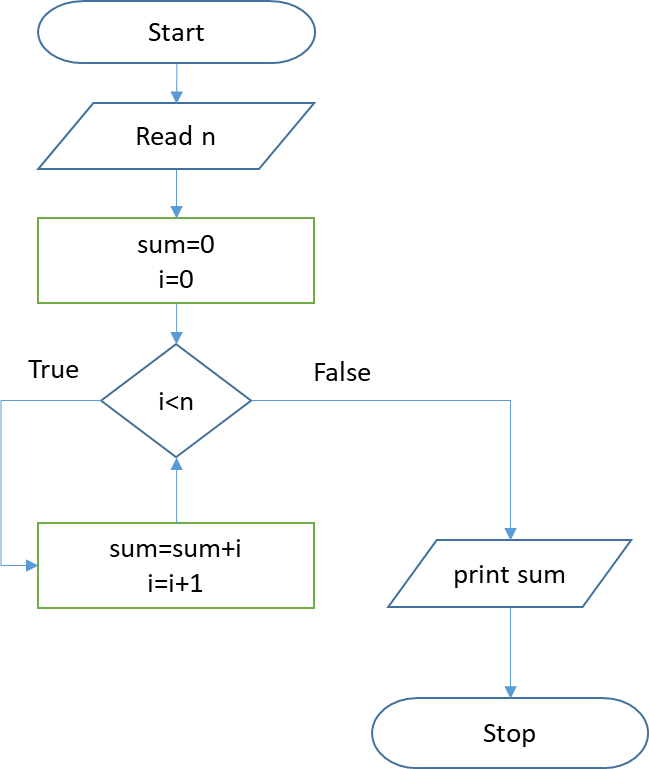
**AIM**

To write a program in Python to find the sum of first N numbers.

**ALGORITHM**

1. Start
2. Get the limit of the series using input function and convert into integer.
3. Initialize the sum variable to zero.
4. Set a for loop from 0 to limit.
5. Inside for loop, add the sum & iteration variable (i) and store it in sum.
6. After termination of the loop, print the sum value.
7. Stop

**FLOWCHART**

****

**PROGRAM**

n=int(input("Enter the Limit"))

sum=0

for i in range(0,n):

sum=sum+i

print("Sum",sum)

**OUTPUT**

Enter the Limit6

Sum 15

**RESULT**

The above program in Python to find the sum of first N numbers has been executed successfully and the output was verified.

**EX NO 2(A) FIND THE GCD OF TWO NUMBERS**

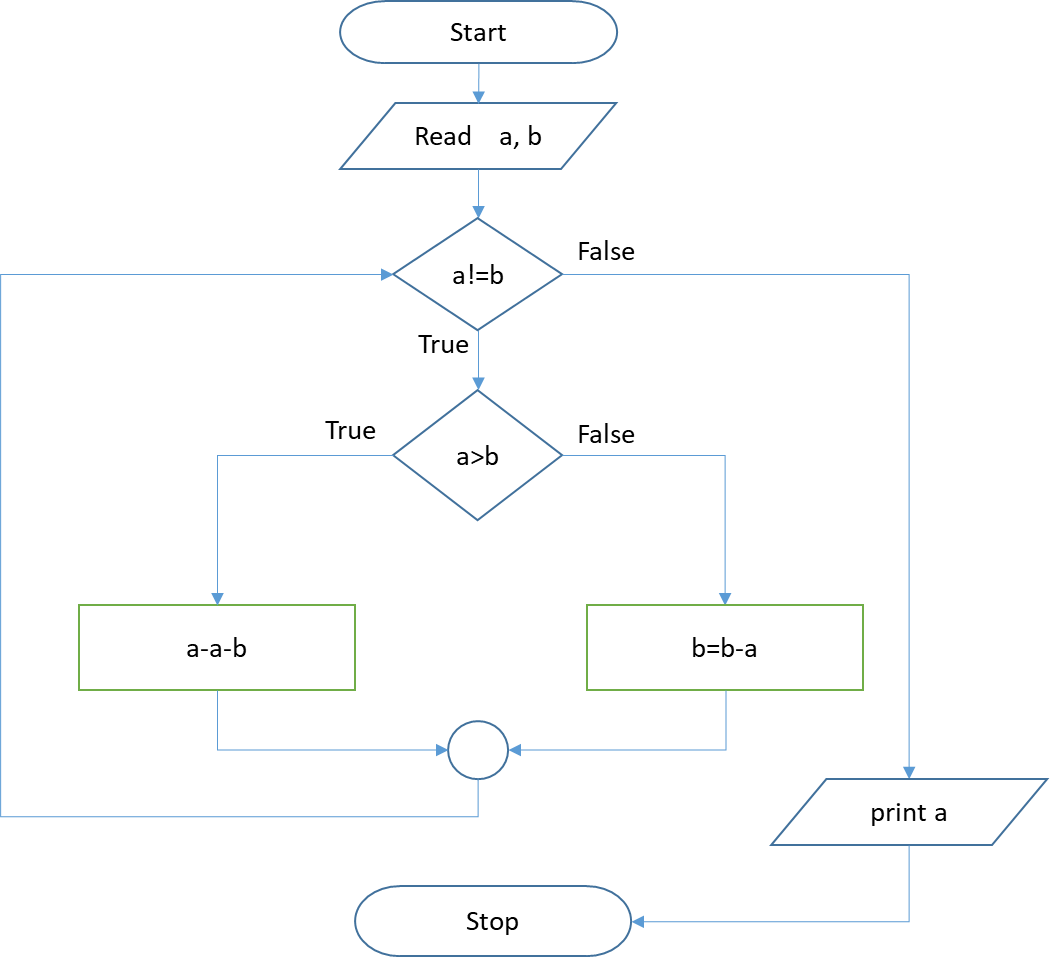
**AIM**

To write a program in Python to find the gcd of two numbers.

**ALGORITHM**

1. Start
2. Get the first number and second number using input function and convert into integer.
3. Set a while loop where first and second number must not be equal.
4. Inside while loop, check if first number is greater than second number.
5. If first is greater, subtract second from first and store it in first.
6. Otherwise, subtract first from second and store it in second.
7. Repeat step-3 to 6 till the condition in the loop satisfied.
8. After the termination of loop, print the first number which is the gcd of two numbers.
9. Stop.

**FLOWCHART**

****

**PROGRAM**

a = int(input("Enter first number: "))  
b = int(input("Enter second number: "))

while a != b:  
 if a > b:  
 a -= b  
 elif b > a:  
 b -= a

print("The GCD of the two numbers is", a)

**OUTPUT**

Enter first number: 6

Enter second number: 3

The GCD of the two numbers is 3

**RESULT**

The above program in Python to find the gcd of two numbers has been executed successfully and the output was verified.

**EX NO 2(B) FIND THE LCM OF TWO NUMBERS**

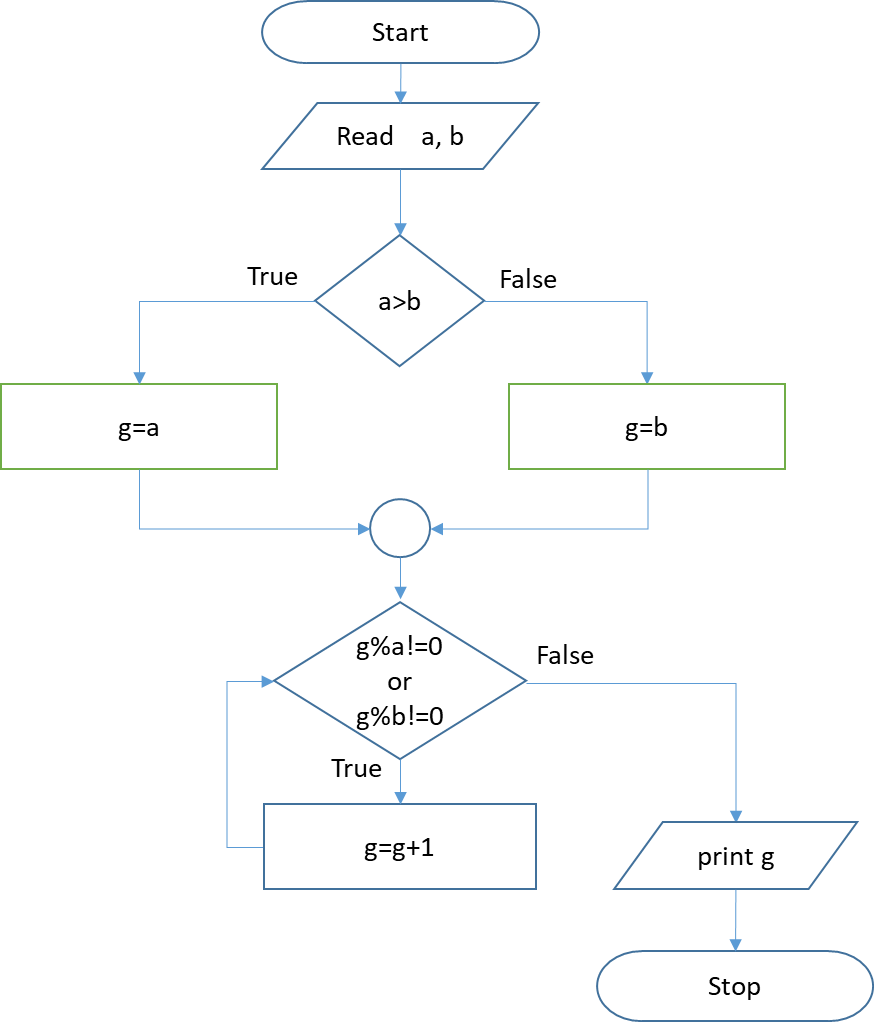
**AIM**

To write a program in Python to find the lcm of two numbers.

**ALGORITHM**

1. Start
2. Get the first number and second number using input function and convert into integer.
3. Check if first is greater than second, then assign first to greatest. Otherwise assign second to greatest.
4. Set a while loop with the condition greater no modulus a should not equal to 0 or greater no modulus b should not equal to 0
5. If Step-4 is True, increment the value of greater no by 1.
6. Repeat step-4 to 5 till the condition in the loop satisfied.
7. After the termination of loop, print the greatert number which is the lcm of two numbers.
8. Stop.

**FLOWCHART**

****

**PROGRAM**

a=int(input("Enter first number"))

b=int(input("Enter second number"))

if a>b:

g=a

else:

g=b

while g%a!=0 or g%b!=0:

g=g+1

print("LCM",g)

**OUTPUT**

Enter first number: 6

Enter second number: 3

LCM 6

**RESULT**

The above program in Python to find the lcm of two numbers has been executed successfully and the output was verified.

**EX NO 3 FIND THE SQUARE ROOT OF A NUMBER (NEWTON‘S METHOD)**

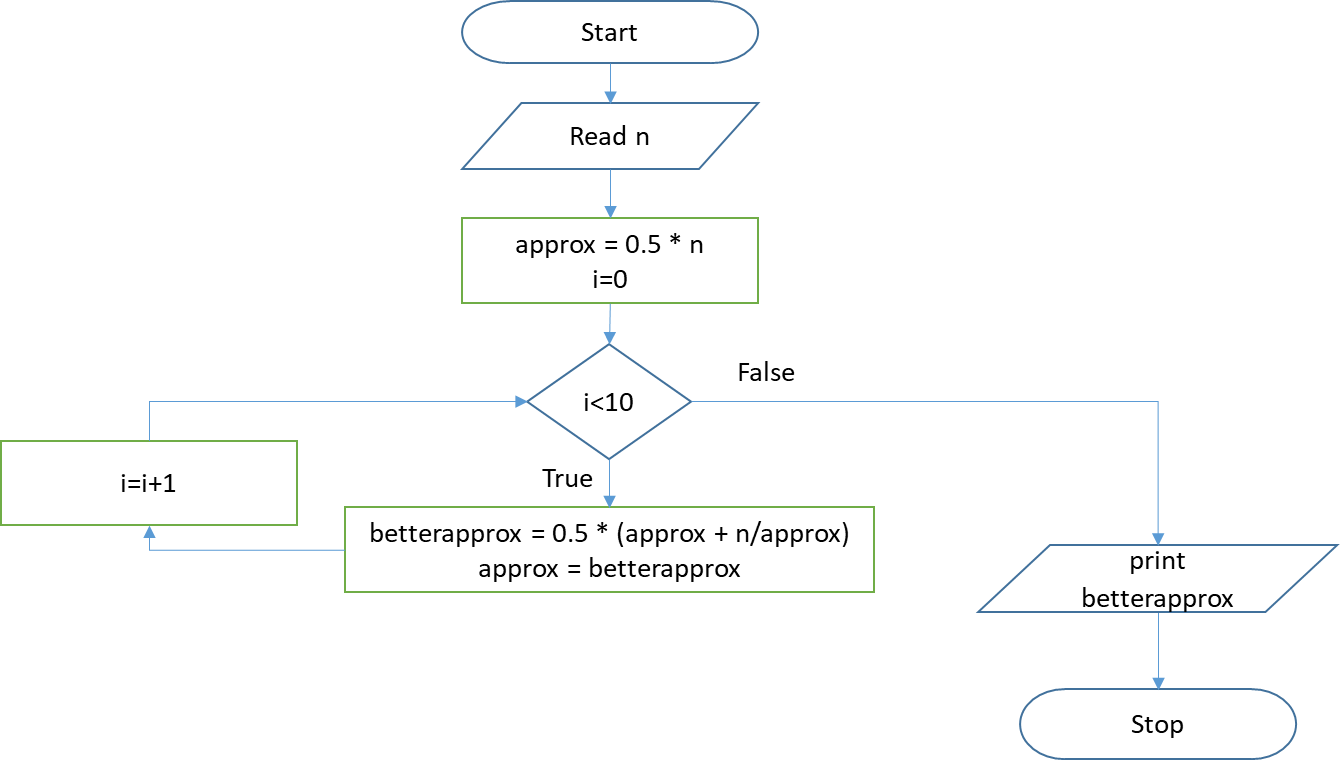
**AIM**

To write a program in Python to find the square root of a given number using Newton’s method.

**ALGORITHM**

1. Start
2. Get the number using input function and convert into integer which stored in a variable (n).
3. Multiply the number with 0.5 and store the result in approx. variable.
4. Set for loop with a range upto 10.
5. Inside for loop, multiply 0.5 with sum of approx and divide approx by n.
6. Store the above result in betterapprox and assign the betterapprox value to approx.
7. Repeat step-5 & 6 till it reaches the end of the sequence.
8. After termination of loop, print the value of betterapprox as a square root of a given number.
9. Stop

**FLOWCHART**

****

**PROGRAM**

n=int(input("Enter number"))

approx = 0.5 \* n

for i in range(10):

betterapprox = 0.5 \* (approx + n/approx)

approx = betterapprox

print(betterapprox)

**OUTPUT**

Enter number169

13.0

**RESULT**

The above program to find the square root of a given number using newton’s method is executed successfully and the output was verified.

**EX NO 4 FIND THE EXPONENTIATION OF A NUMBER**

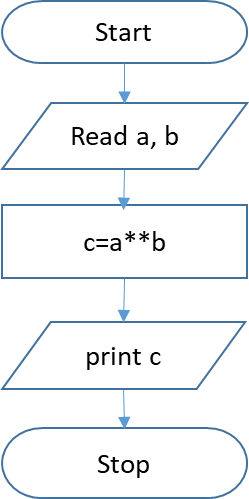
**AIM**

To write a program in Python to find the exponentiation of a number.

**ALGORITHM**

1. Start
2. Get the number and the exponent using input function and convert into integer.
3. Calculate the exponent of that number using exponent operator.
4. Print the result value.
5. Stop

**FLOWCHART**

****

**PROGRAM**

a=int(input("Enter Number"))

b=int(input("Enter Power"))

c=a\*\*b

print(c)

**OUTPUT**

Enter Number 6

Enter Power 3

216

**RESULT**

The above program to find the exponentiation of a number has been executed successfully and the output was verified.

**EX NO 5 FIND THE MAXIMUM NUMBER IN LIST**

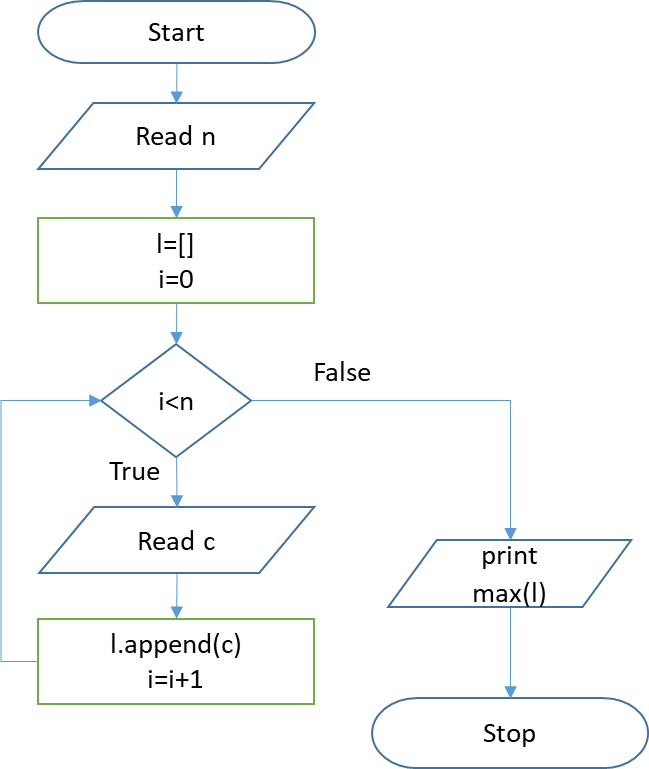
**AIM**

To write a program in Python to find the maximum number in the list of numbers.

**ALGORITHM**

1. Start
2. Create an empty list to store the numbers.
3. Get the number of elements using input function and convert into integer which stored in a variable(n).
4. Set a for loop ranges from zero to n.
5. Get the numbers using input function and append the numbers to the list.
6. Print the maximum value in the list using max function.
7. Stop

**FLOWCHART**

****

**PROGRAM**

n=int(input("enter no of elements"))

l=[]

for i in range(0,n):

c=int(input("enter number"))

l.append(c)

print("Max Value is",max(l))

**OUTPUT**

enter no of elements5

enter number1

enter number0

enter number3

enter number6

enter number9

Max Value is 9

**RESULT**

The above program to find the maximum number in the list has been executed successfully and the output was verified.

**EX NO 6(A) IMPLEMENTATION OF LINEAR SEARCH**

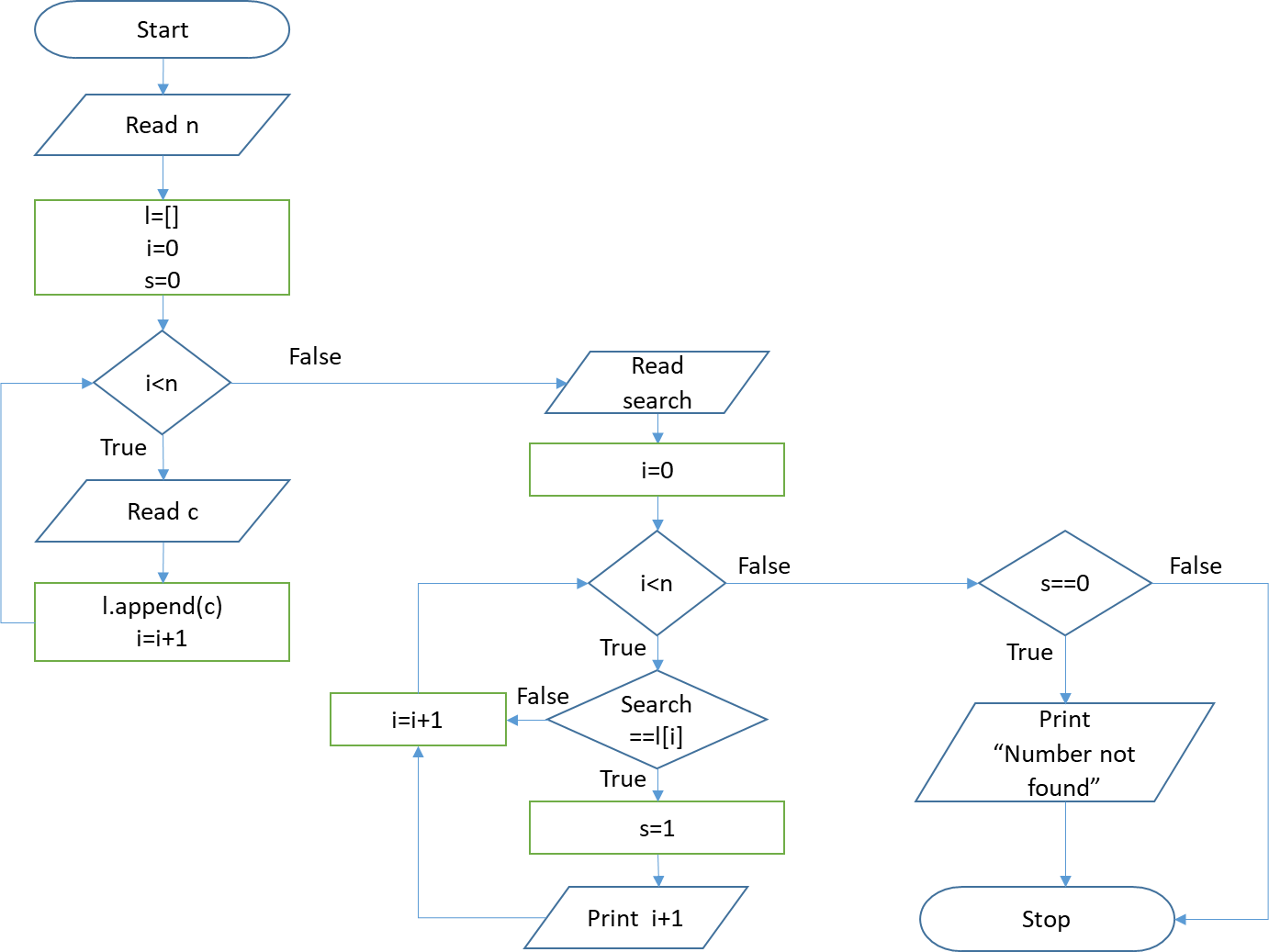
**AIM**

To write a program in Python to find a number in the list using linear search.

**ALGORITHM**

1. Start
2. Create an empty list to store the numbers.
3. Get the number of elements using input function and convert into integer which stored in a variable (n).
4. Set a for loop ranges from zero to n.
5. Get the numbers using input function and append the numbers to the list.
6. Get the number to be find and initialize a status variable equal to zero.
7. Set a for loop ranges from zero to n.
8. If the number to be find is equal to ith value in the list, print i as position and set the status to 1.
9. Repeat step-8 till the for loop reaches the end of sequence.
10. After termination of for loop, Check if the status value equal to zero.
11. If True, print value not found.
12. Stop

**FLOWCHART**

****

**PROGRAM**

n = int(input("Enter the size of the list: "))

l = []

s=0

for i in range(0,n):

c=int(input("Enter elements"))

l.append(c)

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

for i in range(0,n):

if search==l[i]:

s=1

print("The position is",i+1)

if s==0:

print("Element Not Found")

**OUTPUT**

Enter the size of the list: 5

Enter elements3

Enter elements6

Enter elements8

Enter elements4

Enter elements1

Enter the number to search: 8

The position is 3

**RESULT**

The above program to find the number in the list using linear search has been executed successfully and the output was verified.

**EX NO 6(B) IMPLEMENTATION OF BINARY SEARCH**

**AIM**

To write a program in Python to find the number in the list using binary search.

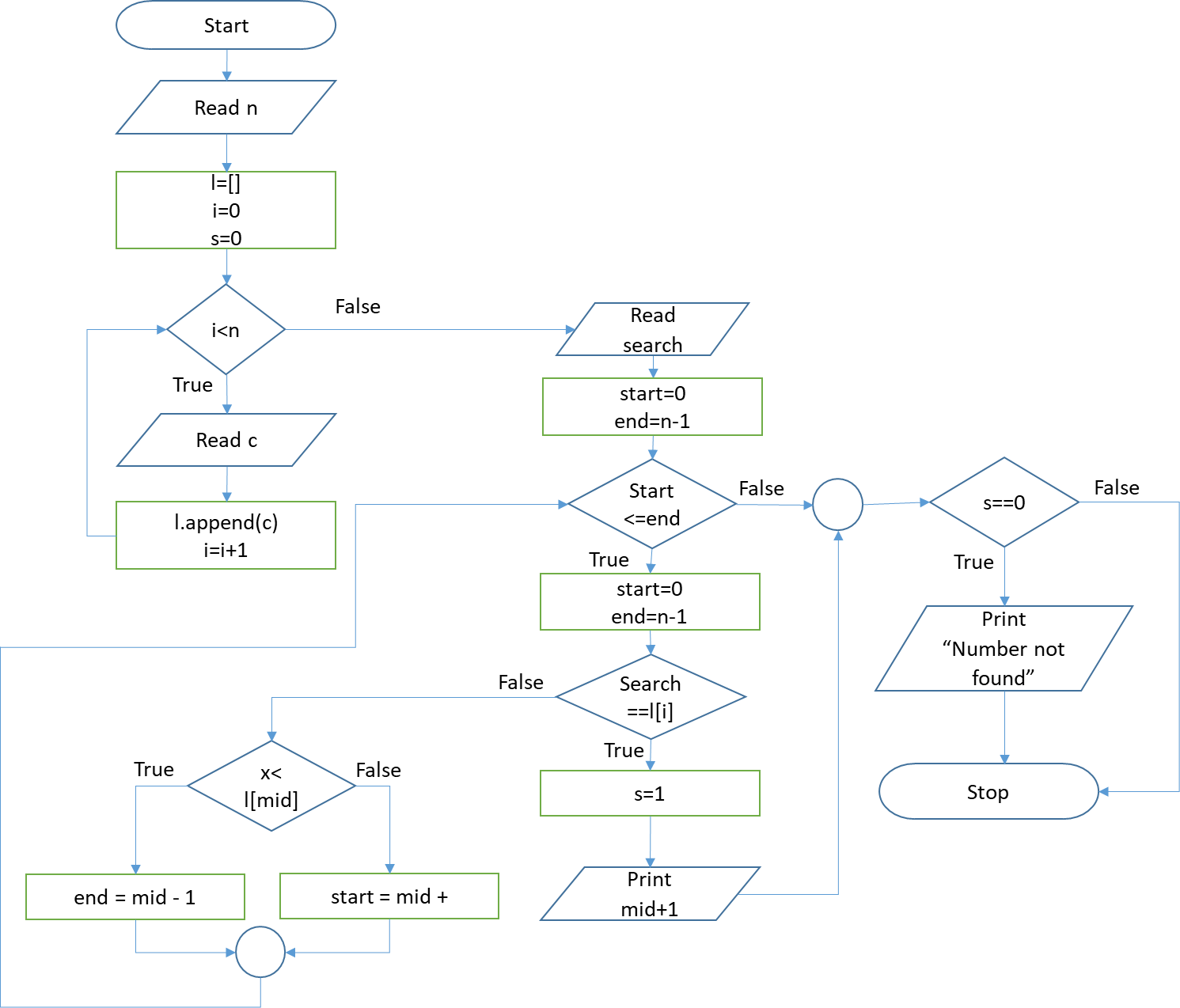
**ALGORITHM**

1. Start.
2. Create an empty list to store the numbers.
3. Get the number of elements using input function and cast into integer which stored in a

variable (n).

1. Set a for loop ranges from zero to n.
2. Get the numbers using input function and append the numbers to the list.
3. Initialize the start value to 0, end value to n-1 and stat value to 0.
4. Set a while loop till start value is less than end value.
5. Calculate the mid value by adding start and end where the result is floor divided by 2.\
6. If the number to find is equal to the number whose position is mid in the list, print the mid as position of the element and terminate the loop using break statement.
7. If the number to find is less than the number whose position is mid in the list, set the end value to mid-1.
8. Otherwise set the start value to mid+1.
9. Repeat step 9 to 11 till the condition in the while loop satisfied.
10. After termination of loop, Check if the stat value equal to zero.
11. If True, print Number not found.
12. Stop.

**FLOWCHART**

****

**PROGRAM**

n = int(input("Enter the size of the list: "))

l = []

for i in range(0,n):

c=int(input("Enter element: "))

l.append(c)

x = int(input("Enter the number to search: "))

start = 0

end = n-1

stat=0

while start <= end:

mid = (start + end) // 2

if x == l[mid]:

stat=1

print("position is", mid + 1)

break;

elif x < l[mid]:

end = mid - 1

else:

start = mid + 1

if stat==0:

print("Value not found")

**OUTPUT**

Enter the size of the list: 5

Enter element: 2

Enter element: 5

Enter element: 6

Enter element: 8

Enter element: 9

Enter the number to search: 8

position is 4

**RESULT**

The above program to find the number in the list using binary search has been executed successfully and the output was verified.

**EX NO 7(A) IMPLEMENTATION OF SELECTION SORT**

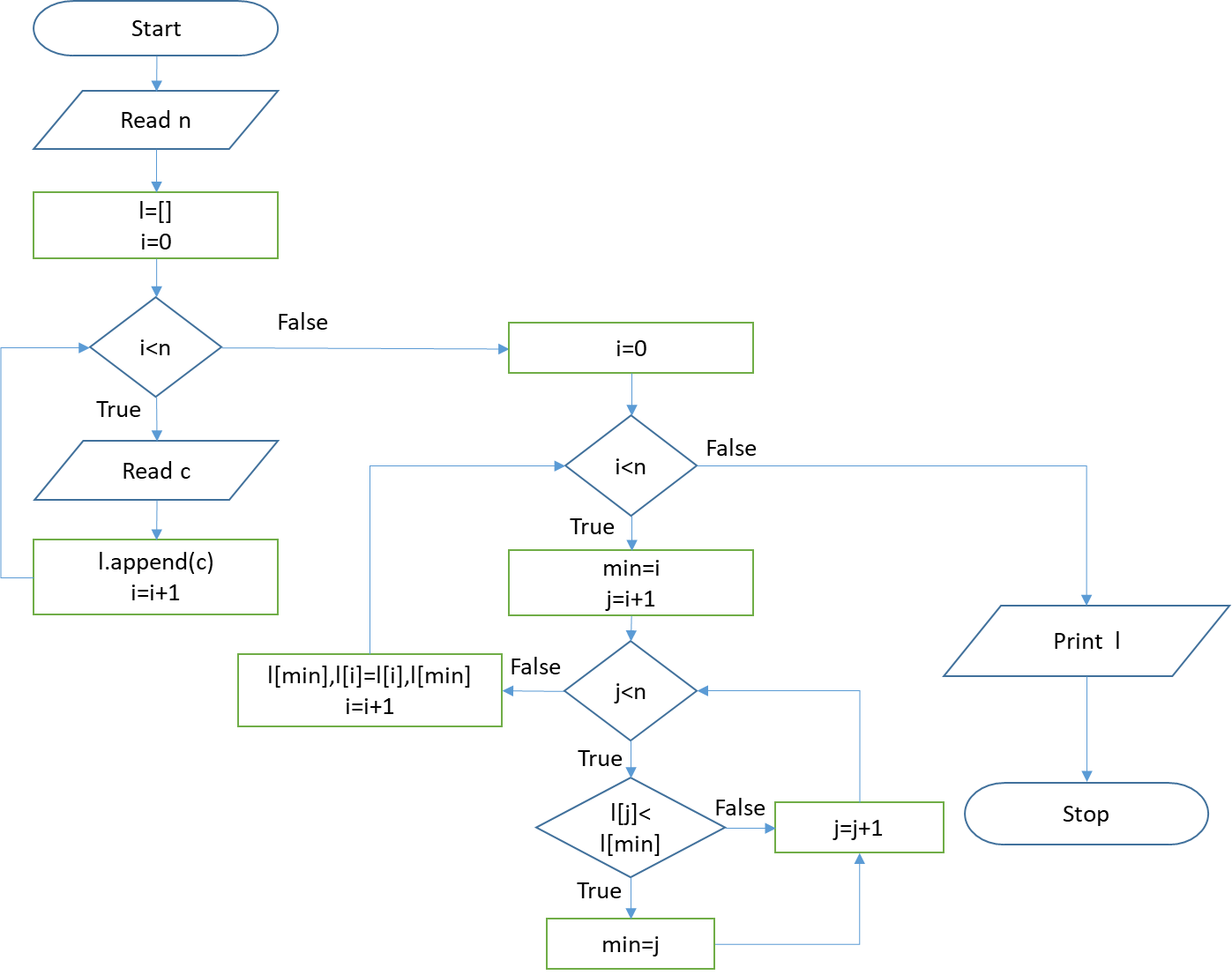
**AIM**

To write a program in Python to sort the numbers using selection sort.

**ALGORITHM**

1. Start.
2. Define selectionSort function which takes list of numbers as parameter.
3. Set a for loop ranges from length of list to 0 and decrement every step by 1.
4. Initialize the max position value as zero. Define another loop inside the above which ranges from 1 to start value of upper for loop+1.
5. If the value in the list whose position is iteration value of inner for loop is greater than the value in the list whose position is max position, update the max position value to the iteration value.
6. Swap the values in the list whose positions are iteration value of outer for loop and max position.
7. Get the count using input function and cast into integer which stored in a variable.
8. Create an empty list to store the numbers.
9. Set a for loop ranges from zero to count.
10. Get the numbers using input function and append the numbers to the list.
11. Call selectionSort function which passes the list and print the sorted list.
12. Stop.

**FLOWCHART**

****

**PROGRAM**

n = int(input("Enter the size of the list: "))

l = []

for i in range(0,n):

c=int(input("Enter element: "))

l.append(c)

for i in range(0,n):

min=i

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

if l[j]<l[min]:

min=j

l[min],l[i]=l[i],l[min]

print(l)

**OUTPUT**

Enter the size of the list: 5

Enter element: 2

Enter element: 1

Enter element: 0

Enter element: 3

Enter element: 6

[0, 1, 2, 3, 6]

**RESULT**

The above program to sort the numbers using selection sort has been executed successfully and the output was verified.

**EX NO 7(B) IMPLEMENTATION OF INSERTION SORT**

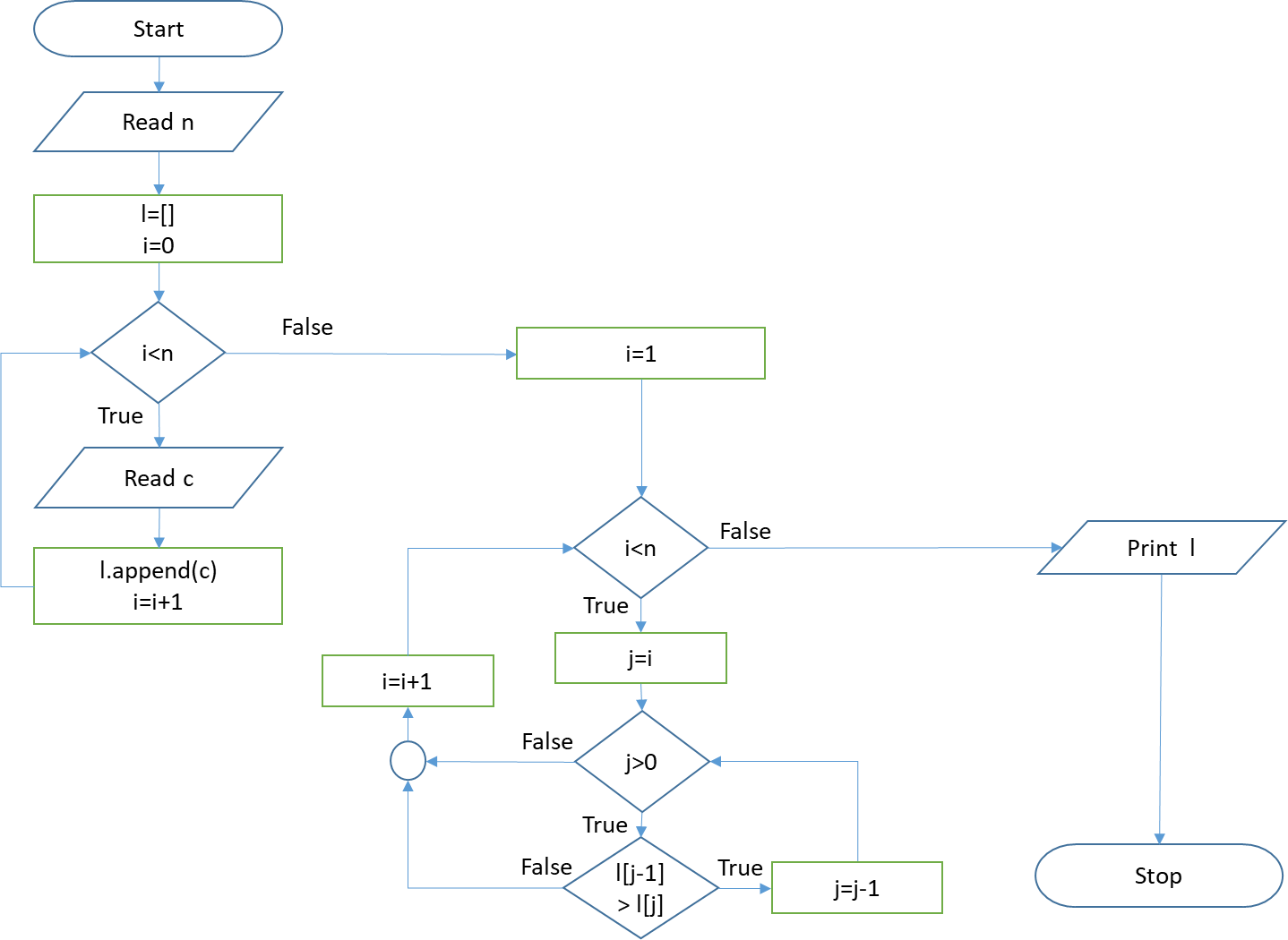
**AIM**

To write a program in Python to sort the numbers using insertion sort.

**ALGORITHM**

1. Start.
2. Set a for loop ranges from 1 to length of list.
3. Set the current value to the value in the list whose position is iteration value of the loop which can act as index.
4. Define another while loop inside the above which checks the position greater than 0 and checks value in the list whose position is one lesser than previous greater than current value.
5. If true, assign the previous position value to the current position value and decrement the position by 1.
6. Assign the current value to the final position value in the list.
7. Get the count using input function and cast into integer which stored in a variable.
8. Create an empty list to store the numbers.
9. Set a for loop ranges from zero to count.
10. Get the numbers using input function and append the numbers to the list.
11. Call insertionSort function which passes the list and print the sorted list.
12. Stop.

**FLOWCHART**

****

**PROGRAM**

n=int(input("Enter no of elements"))

l=[]

for i in range(0,n):

c=int(input("Enter value"))

l.append(c)

for i in range(1,n):

j = i

while j>0:

if l[j-1] > l[j]:

l[j-1],l[j] = l[j],l[j-1]

else:

break

j = j-1

print(l)

**OUTPUT**

Enter no of elements5

Enter value2

Enter value0

Enter value3

Enter value5

Enter value4

[0, 2, 3, 4, 5]

**RESULT**

The above program to sort the numbers using insertion sort has been executed successfully and the output was verified.

**EX NO 8 IMPLEMENTATION OF MERGESORT**

**AIM**

To write a program in Python to sort the numbers using merge sort.

**ALGORITHM**

1. Start.
2. Define the function msort with the argument alist
3. If length of the list is greater than 1 then find the middle value by mid = len(alist)//2
4. Fix the left and right value
5. Call the msort function repeatedly with the parameter lhalf and rhalf
6. Initialize i,j,k as 0
7. Repeat the following steps till i <len(lhalf) and j <len(rhalf)

7.1 if lhalf [i] < rhalft[j]: then alist[k]=lhalf [i] and i=i+1

7.2 otherwise alist[k]=rhalf [j] and j=j+1k=k+1

1. Repeat the following steps till i <len(lhalf)

8.1 alist[k]=lhalf [i]

8.2 i=i+1

8.3 k=k+1

1. Repeat the following steps till j <len(rhalf)

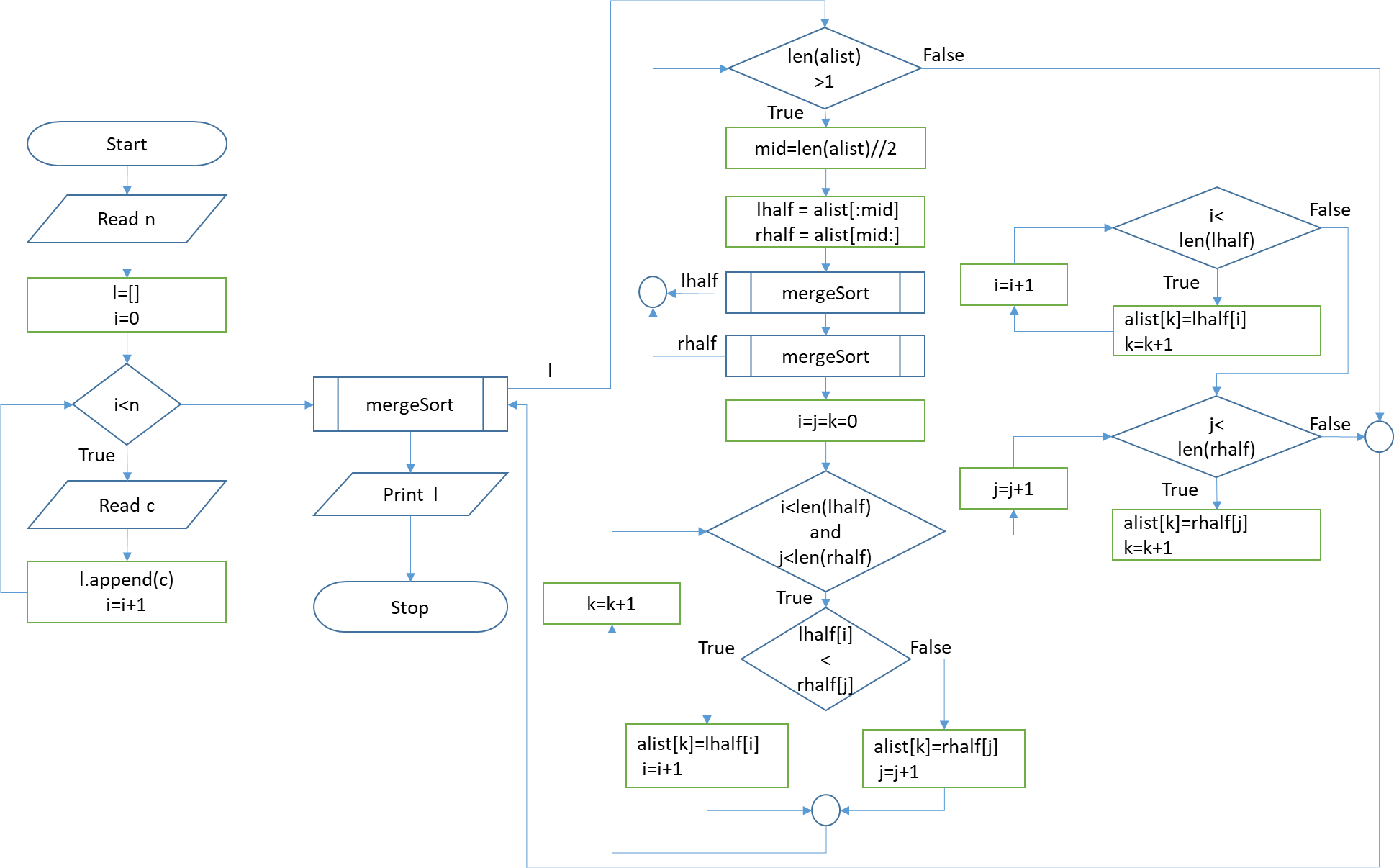
9.1 alist[k]=rhalf [j]

9.2 j=j+1

9.3 k=k+1

1. Create an empty list l
2. Read n
3. Read and append n numbers within for loop
4. Call msort function with the argument l
5. Print the list l.
6. Stop.

**FLOWCHART**

****

**PROGRAM**

def mergeSort(alist):

if len(alist)>1:

mid = len(alist)//2

lhalf = alist[:mid]

rhalf = alist[mid:]

mergeSort(lhalf)

mergeSort(rhalf)

i=j=k=0

while i < len(lhalf) and j < len(rhalf):

if lhalf[i] < rhalf[j]:

alist[k]=lhalf[i]

i=i+1

else:

alist[k]=rhalf[j]

j=j+1

k=k+1

while i < len(lhalf):

alist[k]=lhalf[i]

k=k+1

i=i+1

while j < len(rhalf):

alist[k]=rhalf[j]

k=k+1

j=j+1

l=[]

n=int(input("enter no of elements"))

for i in range(0,n):

c=int(input("enter number"))

l.append(c)

mergeSort(l)

print(l)

**OUTPUT**

enter no of elements5

enter number2

enter number0

enter number-5

enter number4

enter number1

[-5, 0, 1, 2, 4]

**RESULT**

The above program to sort the numbers using merge sort has been executed successfully and the output was verified.

**EX NO 9 FIND THE FIRST N PRIME NUMBERS**

**AIM**

To write a program in Python to find the N prime numbers.

**ALGORITHM**

1. Start.
2. Read l
3. If l>1

3.1 Set i=2 and repeat the following steps till i<l+1

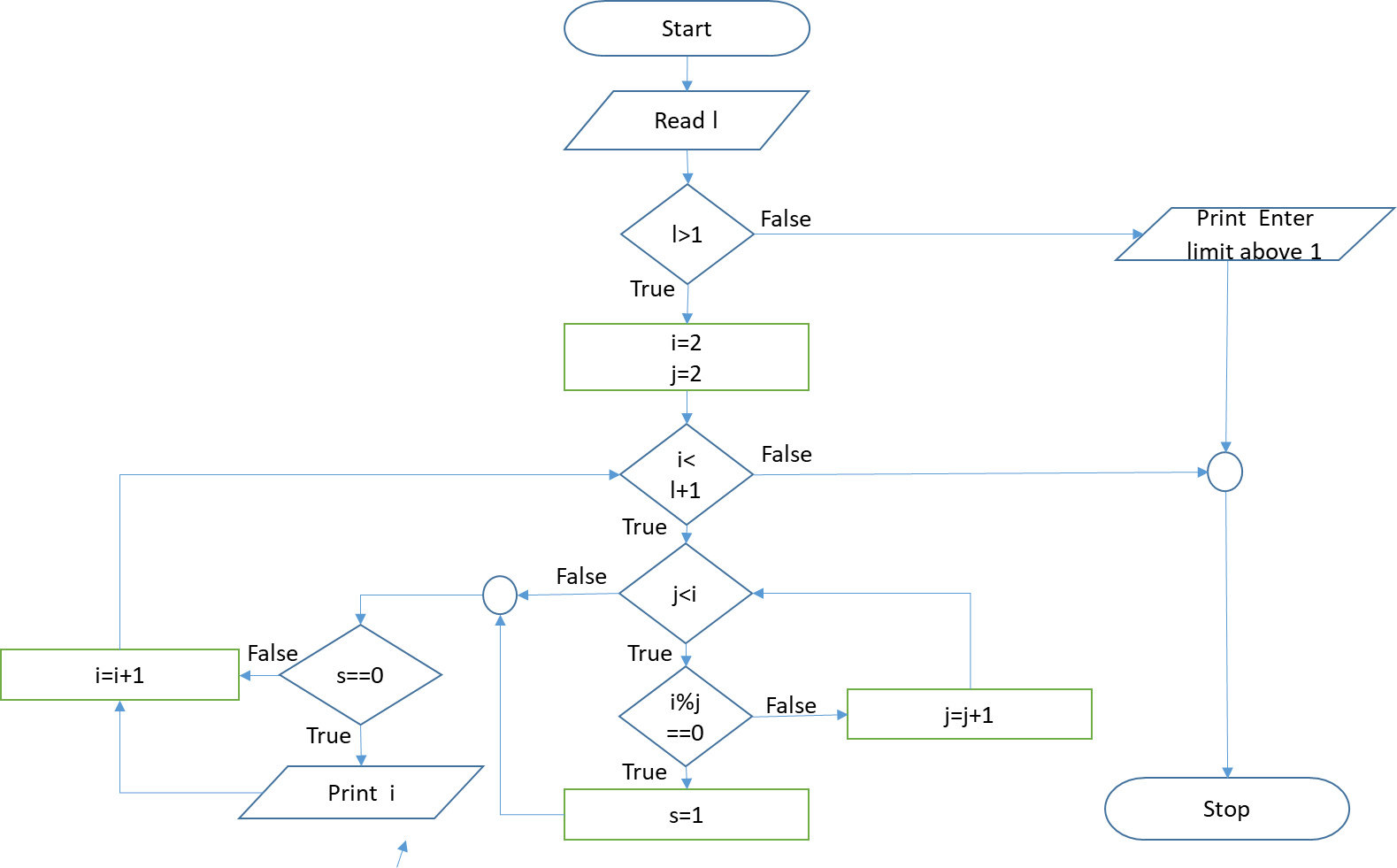
3.2 s=0

3.3 Set j=2 and repeat the following steps till j<i

3.3.1 if i%j==0 then s=1 and break

1. If s=0 then print i
2. Otherwise print “Enter limit above 1”
3. Stop

**FLOWCHART**

****

**PROGRAM**

l=int(input("Enter Limit"))

if l>1:

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

s=0

for j in range(2,i):

if i%j==0:

s=1

break

if s==0:

print(i)

else:

print("Enter Limit Above 1")

**OUTPUT**

Enter Limit20

2

3

5

7

11

13

17

19

**RESULT**

The above program to find N prime numbers has been executed successfully and the output was verified.

**EX NO 10(A) PERFORM ADDITION OF MATRICES**

**AIM**

To write a program in Python to perform matrix addition.

**ALGORITHM**

1. Start.
2. Import numpy package and define matrix function.
3. Set i=0 and repeat the following steps till i< r1

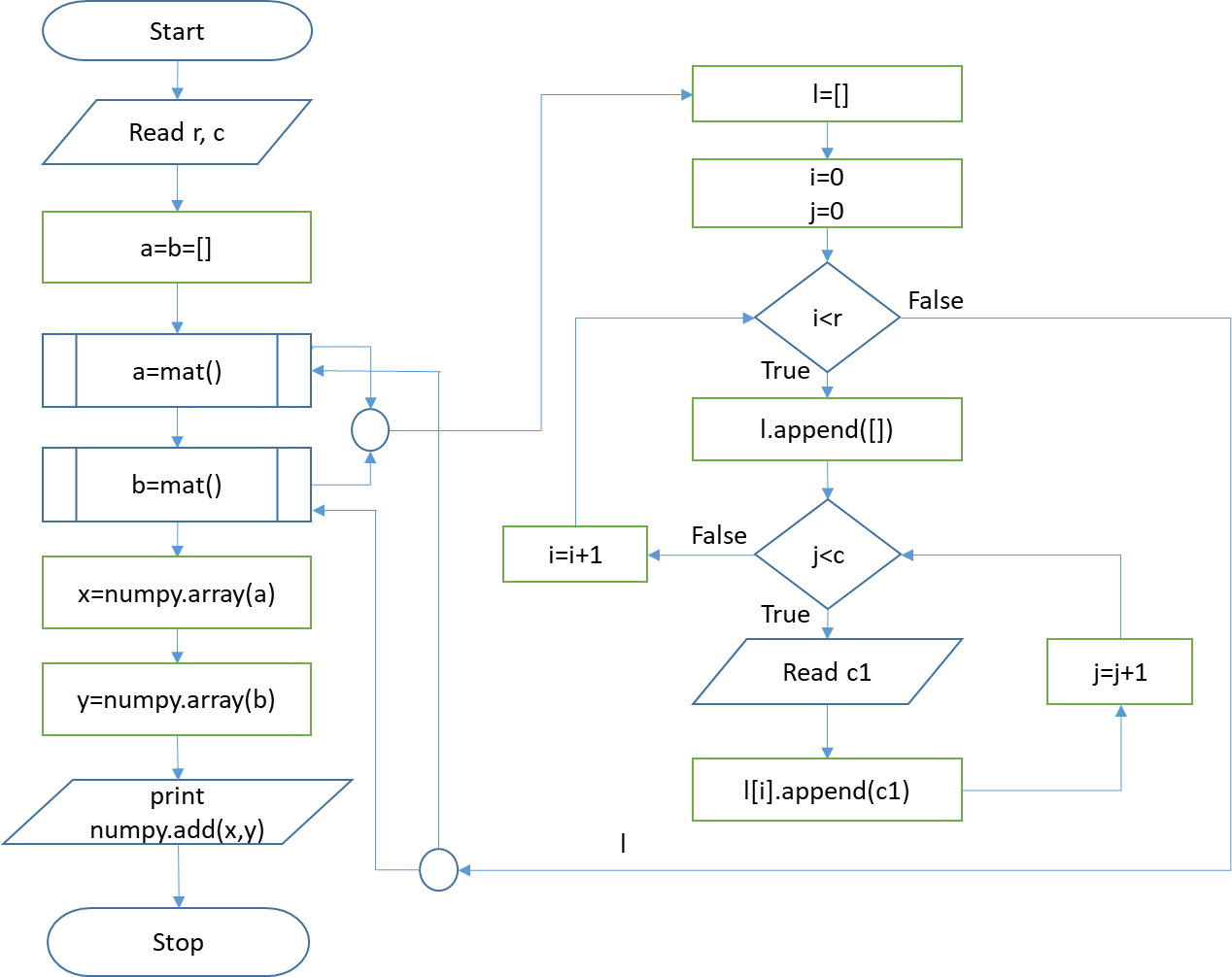
3.1 Append [] with l list

1. Set j=0 and repeat the following steps till j< c1
   1. If s=0 then read t and append t with l[i]
   2. Otherwise append 0 with l[i]
   3. Return l
2. Read r1 and c1
3. Create the empty lists a and b
4. Call the matrix function for a & b and store the return list l.
5. Initialize x & y as matrices with a & b as values using array function.
6. Perform matrix addition using add function with x & y as arguments and print the

value.

1. Stop

**FLOWCHART**

****

**PROGRAM**

import numpy

def mat():

l=[]

print("Enter matrix")

for i in range(0,r):

l.append([])

for j in range(0,c):

c1=int(input("Enter Value"))

l[i].append(c1)

return l

r=int(input("enter row"))

c=int(input("enter column"))

a=b=[]

a=mat()

b=mat()

x=numpy.array(a)

y=numpy.array(b)

print(numpy.add(x,y))

**OUTPUT**

enter row2

enter column2

Enter matrix

Enter Value1

Enter Value2

Enter Value3

Enter Value4

Enter matrix

Enter Value1

Enter Value2

Enter Value3

Enter Value4

[[2 4]

[6 8]]

**RESULT**

The above program to perform addition of two matrices has been executed successfully and the output was verified.

**EX NO 10(B) PERFORM MULTIPLICATION OF MATRICES**

**AIM**

To write a program in Python to perform matrix multiplication.

**ALGORITHM**

1. Start.
2. Import numpy package and define matrix function with arguments with r1 & c1.
3. Set i=0 and repeat the following steps till i< r1.

3.1 Append [] with l list.

1. Set j=0 and repeat the following steps till j< c1.
   1. If s=0 then read t and append t with l[i].
   2. Otherwise append 0 with l[i].
   3. Return l.
2. Read r1, c1, r2 and c2.
3. Check if r2 is equal to c1. If True, proceed to Step-7. If False, print Incorrect size and

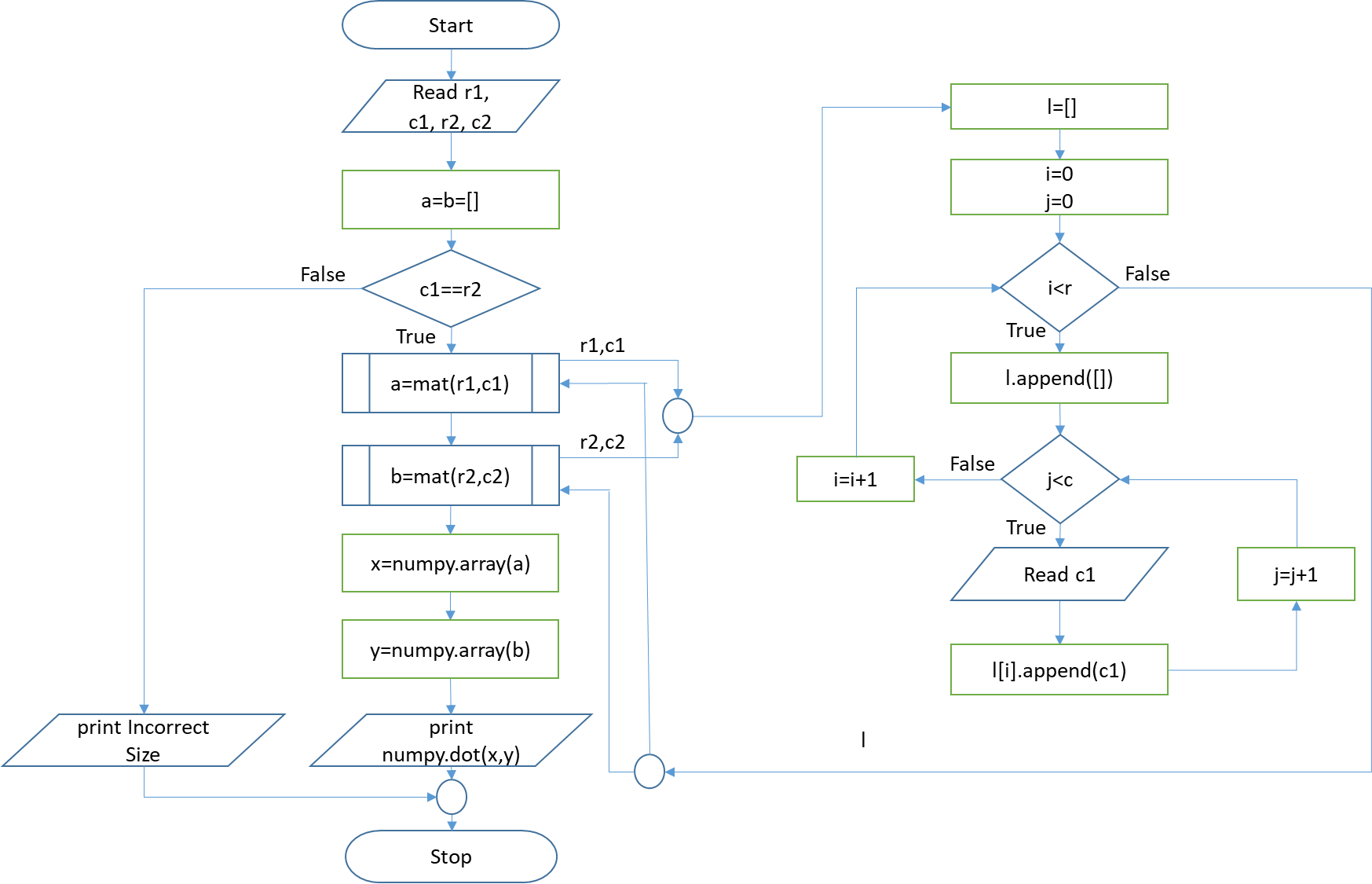
proceed to Step-12.

1. Create the empty lists a and b.
2. Call the matrix function for a with arguments r1 & c1 and store the return list l.
3. Call the matrix function for b with arguments r2 & c2 and store the return list l.
4. Initialize x & y as matrices with a & b as values using array function.
5. Perform matrix multiplication using dot function with x & y as arguments and print the

value.

1. Stop

**FLOWCHART**

****

**PROGRAM**

import numpy

def mat(r,c):

l=[]

print("Enter matrix")

for i in range(0,r):

l.append([])

for j in range(0,c):

c1=int(input("Enter Value"))

l[i].append(c1)

return l

r1=int(input("enter row of matrix A"))

c1=int(input("enter column of matrix A"))

r2=int(input("enter row of matrix B"))

c2=int(input("enter column of matrix B"))

a=b=[]

if c1==r2:

a=mat(r1,c1)

b=mat(r2,c2)

x=numpy.array(a)

y=numpy.array(b)

print(numpy.dot(x,y))

else:

print("Incorrect Size")

**OUTPUT**

enter row of matrix A2

enter column of matrix A2

enter row of matrix B2

enter column of matrix B2

enter matrix

Enter Value1

Enter Value2

Enter Value4

Enter Value5

enter matrix

Enter Value7

Enter Value8

Enter Value9

Enter Value10

enter matrix

[[25 28]

[73 82]]

**RESULT**

The above program to perform multiplication of two matrices has been executed successfully and the output was verified.

**EX NO 11 FIND THE WORD COUNT USING COMMAND LINE ARGUMENTS.**

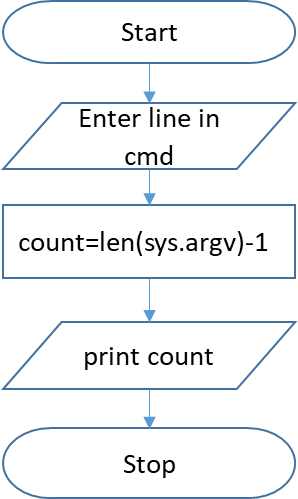
**AIM**

To write a program in Python to find the word count using command line arguments.

**ALGORITHM**

1. Start.
2. Import sys module
3. Set count=len(sys.argv)-1
4. Print count
5. Stop

**FLOWCHART**

****

**PROGRAM**

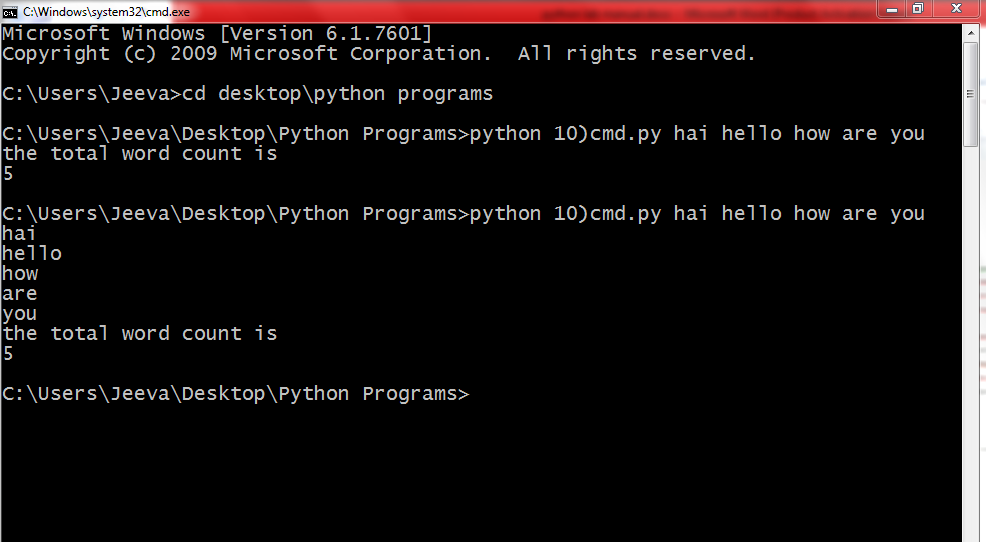
import sys

count=len(sys.argv)-1

print ("the total word count is")

print (count)

**OUTPUT**



**RESULT**

The above program to find the word count using command line arguments has been executed successfully and the output was verified.

**EX NO 12 FIND THE MOST FREQUENT WORDS IN A TEXT READ FROM A FILE**

**AIM**

To write a program in Python to find the most frequent words in a text read from a file.

**ALGORITHM**

1. Start.
2. Import re package and create an empty dictionary d
3. Open a file “words.txt” with “r” mode and store the reference in f
4. Read the content from f using read function and convert into lowercase and store it in

text.

1. Extract the words in text using findall function that separates text & numbers from

symbols and store in words.

1. Set i=0 and repeat the following steps till i<len(words)

6.1 Strip the numbers from words using strip function and store in w.

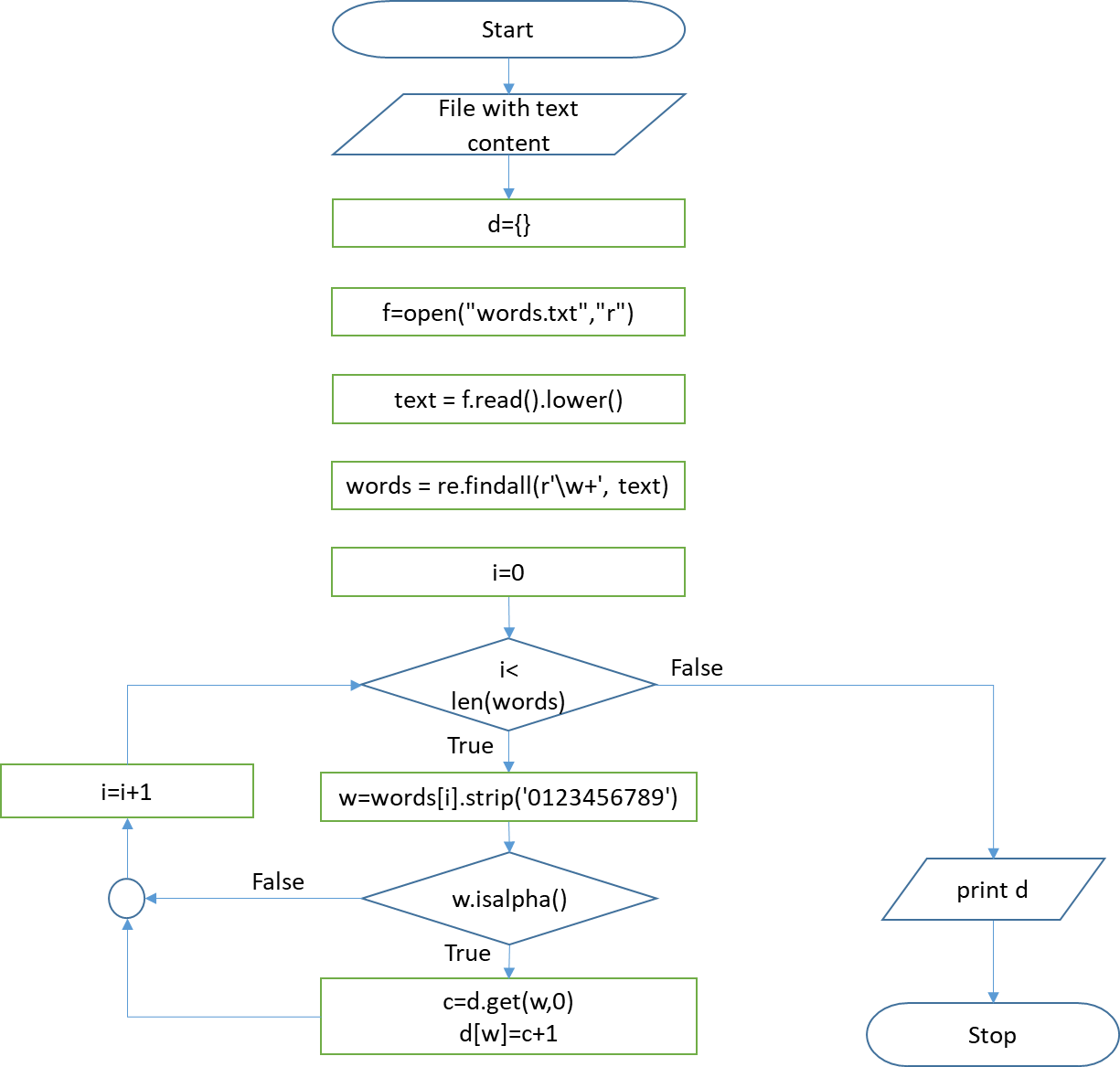
6.2 if w.isalpha() then

6.2.1 c = d.get(w,0)

6.2.2 d[word] = c + 1

1. Print d
2. Stop

**FLOWCHART**

****

**PROGRAM**

import re

d={}

f=open("words.txt","r")

text = f.read().lower()

words = re.findall(r'\w+', text)

for i in range(0,len(words)):

w=words[i].strip('0123456789')

if w.isalpha():

c=d.get(w,0)

d[w]=c+1

print(d)

words.txt

hai hello 3how are 5 you. What about you?

**OUTPUT**

{'hai': 1, 'hello': 1, 'how': 1, 'are': 1, 'you': 2, 'what': 1, 'about': 1}

**RESULT**

The above program to find the most frequent words in a text read from a file has been executed successfully and the output was verified.

**EX NO 13 COPY THE CONTENT OF THE FILE.**

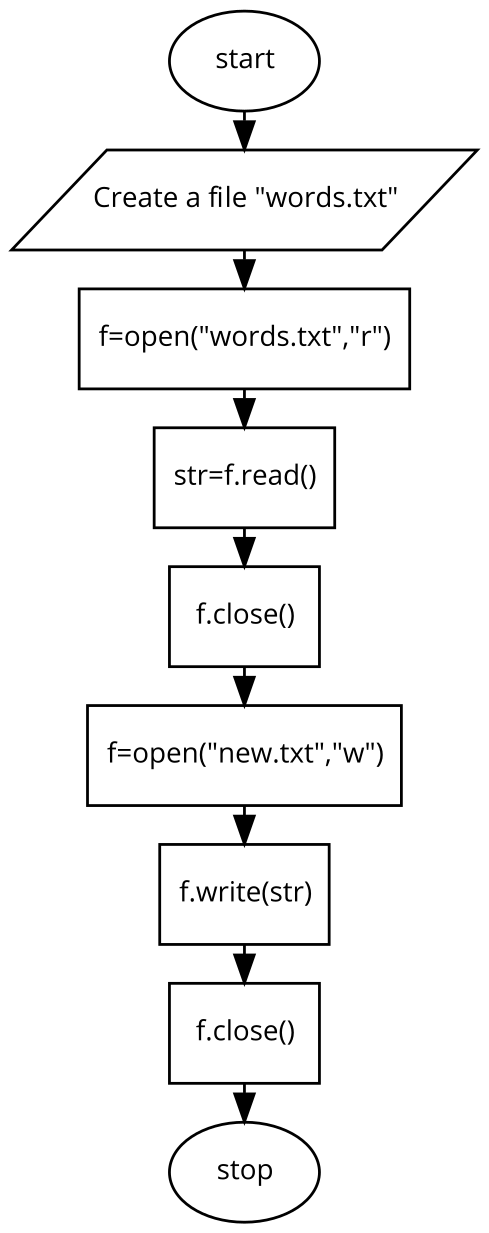
**AIM**

To write a program in Python to copy the content of the file.

**ALGORITHM**

1. Start.
2. Read a file “words.txt” with “r” mode
3. Read the contents from the file and store it in str
4. Close the file
5. Read a file “words.txt” with “w” mode
6. Write str into the file
7. Close the file
8. Stop

**FLOWCHART**

****

**PROGRAM**

f=open("words.txt","r")

str=f.read()

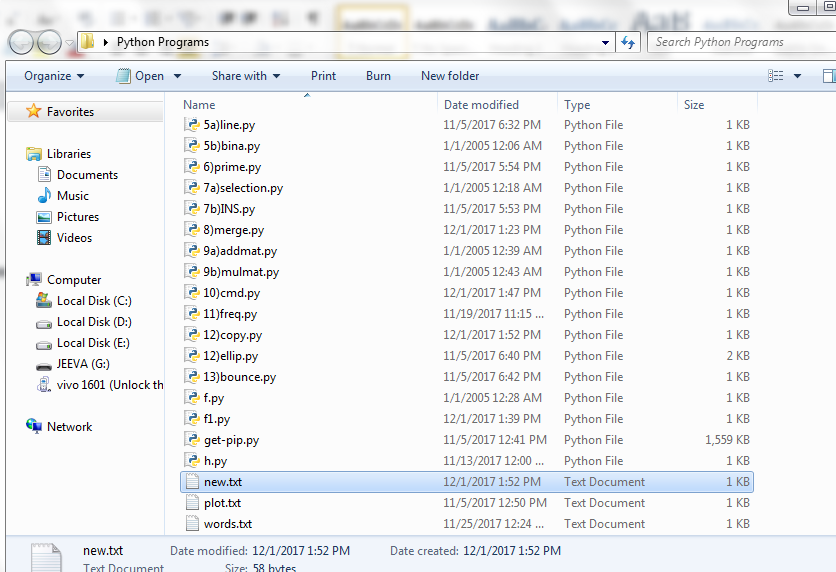
f.close()

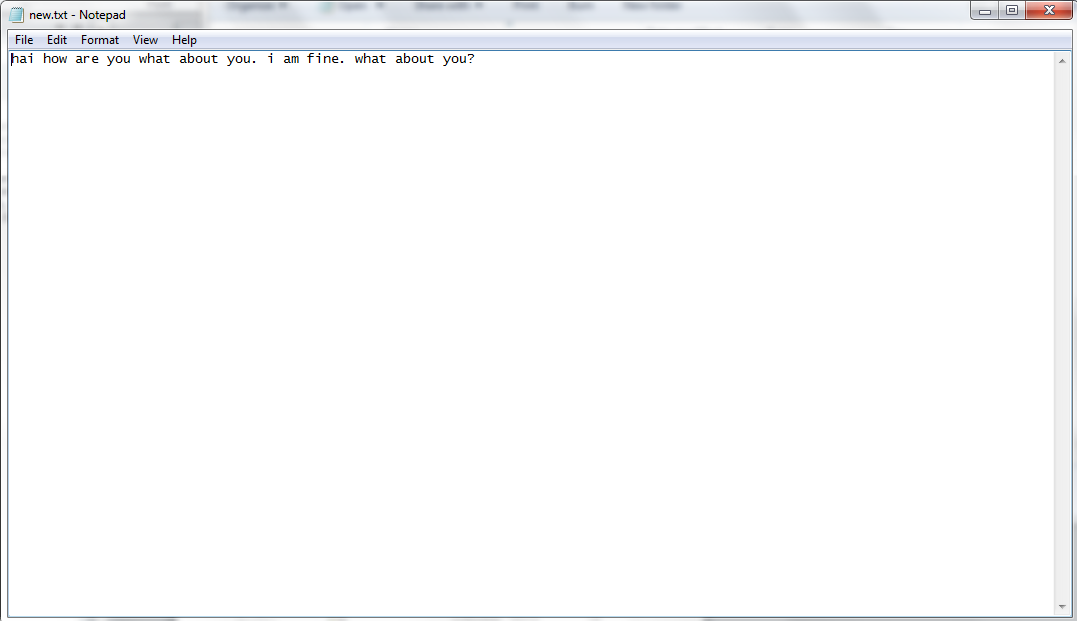
f=open("new.txt","w")

f.write(str)

f.close()

**OUTPUT**





**RESULT**

The above program to copy the content of the file has been executed successfully and the output was verified.