|  |  |
| --- | --- |
| **Ex.No.9**  **13/Apr/2019** | **PROBLEM SOLVING TECHNIQUES** |

**Aim:**

To write programs in Python to

a. To find the number

b. To extract digits from a paragraph

c. To find G.C.,D of two numbers

d. Solve Pascal’s triangle

**1a.To find the number in the list using binary search:**

**Algorithm:**

1. START
2. ENTER THE LIST
3. ENTER THE NUMBER TO BE SEARCHED
4. INITIALISES LAST AND FIRST TO 0
5. MP=FIRST+LAST/2
6. L[MP]==0,THEN RETURN TRUE
7. ITEM<L[MP],THEN LAST=MP-1
8. ITEM>L[MP],THEN FIRST=MP+1
9. RETURN THE POSITION OF THE NUMBER
10. STOP

**FLOWCHART:**

BINARYSEARCH(L,N)

START

L,N=INPUT()

FIRST=0

LAST=0

Binarysearch(l,n)

ITEAM==MP

Print FOUND POSITION

RETURN FOUND

END

ITEAM<L[MP]

LAST=MP-1

ITEAM>L[MP]

FIRST=MP+1

**CODING:**

**def binarySearch (arr, l, r, x):**

**if (r >= l):**

**mid = int(l + (r - l)/2)**

**if (arr[mid] == x):**

**return mid**

**elif (arr[mid] > x):**

**return binarySearch(arr, l, mid-1, x)**

**else:**

**return binarySearch(arr, mid + 1, r, x)**

**else:**

**return -1**

**arr = [int(x)for x in input().split()]**

**d = int(input())**

**result = binarySearch(arr, 0, len(arr)-1, d)**

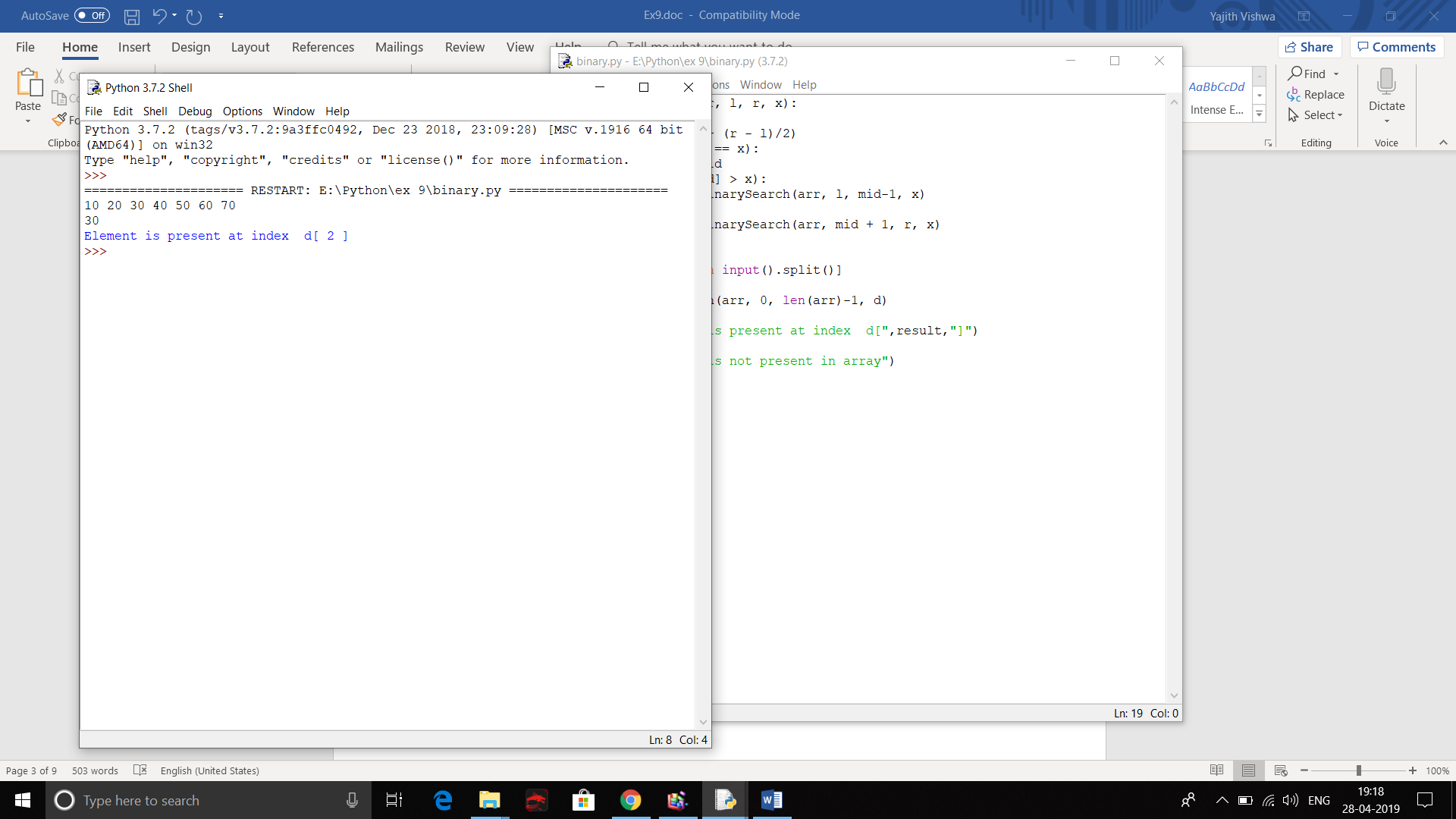
**if (result != -1):**

**print ("Element is present at index d[",result,"]")**

**else:**

**print ("Element is not present in array")**

**SCREENSHOT:**



**2b.To extract numbers from a given paragraph:**

**ALGORITHM:**

STEP 1: Start

STEP 2:Read the paragraph as input from the user

STEP 3:split the paragraph to convert it as a list

STEP 4:Iterate and check every item in the list for digits

STEP 5:Print the item if it is a digit

STEP 6:Stop

**FLOWCHART:**

START

PARAGRAPH INPUT

PARAGRAPH.SORT()

I IN NEW

I.ISDIGIT()

CONTINUE

I

**CODING:**

Paragraph=input("enter pargraph")

new=Paragraph.split()

for i in new:

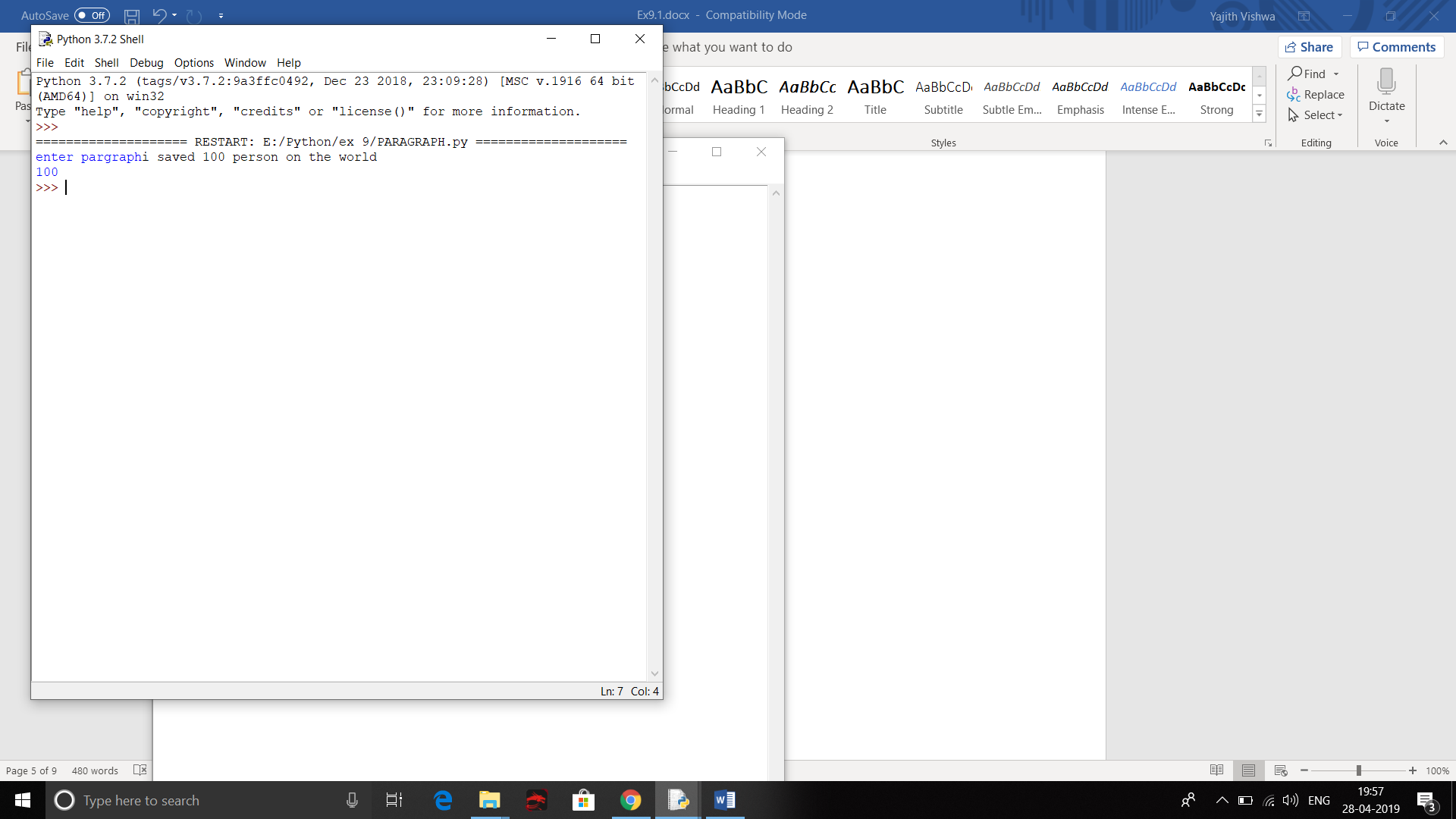
if(i.isdigit()):

print(i)

else:

continue

**SCREENSHOT:**



**3c.To find GCD OF TWO NUMBERS:**

**ALGORITHM:**

1. **START**
2. **READ A,B**
3. **GCD(A,B)**
4. **IF B==0,THEN RETURN A**
5. **ELSE,RECURSIVE FUNCTION GCD(B,A%B)**
6. **END**

**FLOWCHART:**

GCD(A,B)

START

RETURN GCD(B,A%B)

B==0

INPUT A,B

GCD(A,B)

RETURN A

STOP

END

**CODING:**

**def gcd(a,b):**

**if(b==0):**

**return a**

**else:**

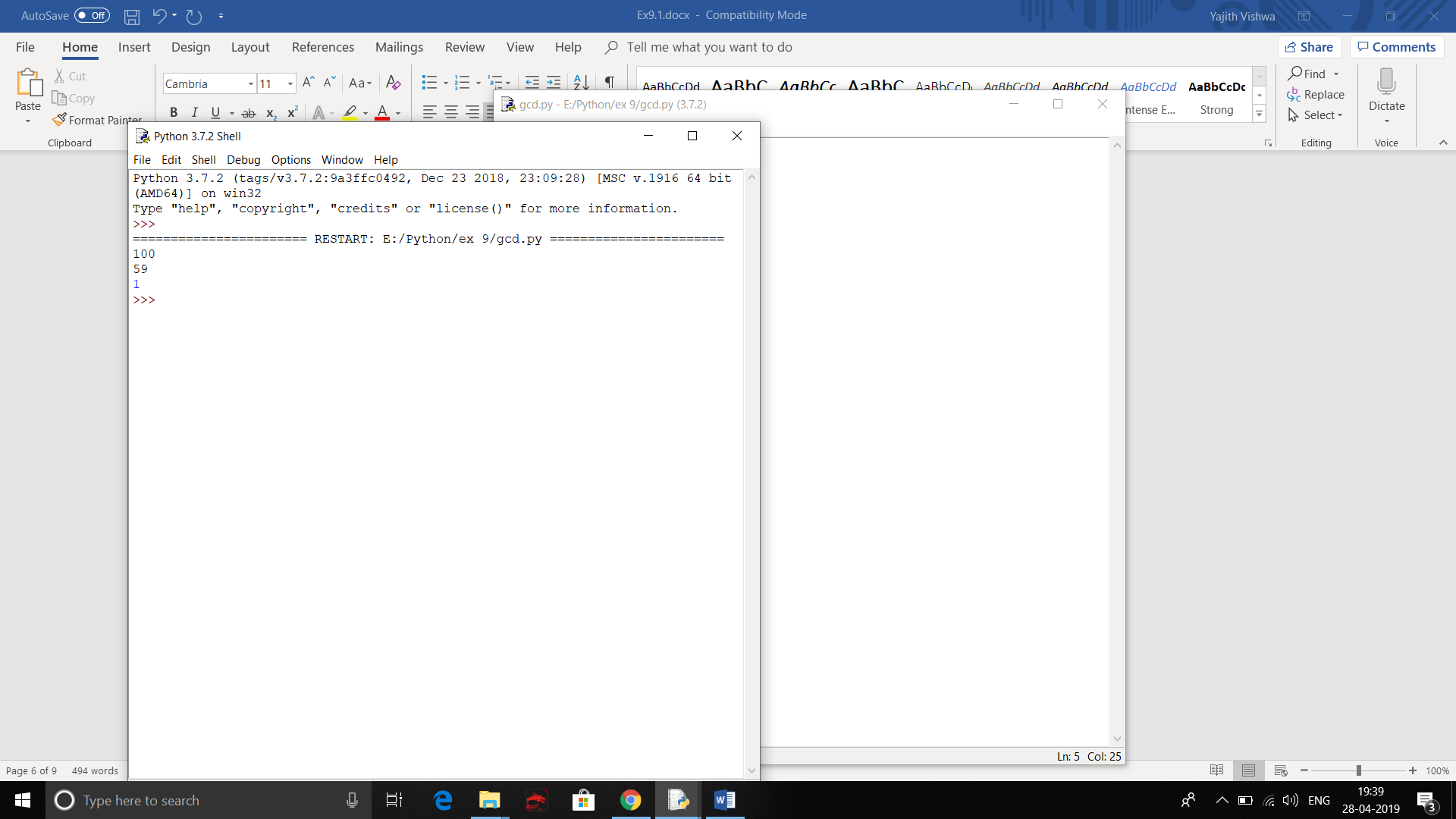
**return gcd(b,a%b)**

**a=int(input())**

**b=int(input())**

**print(gcd(a,b))**

**SCREENSHOT:**

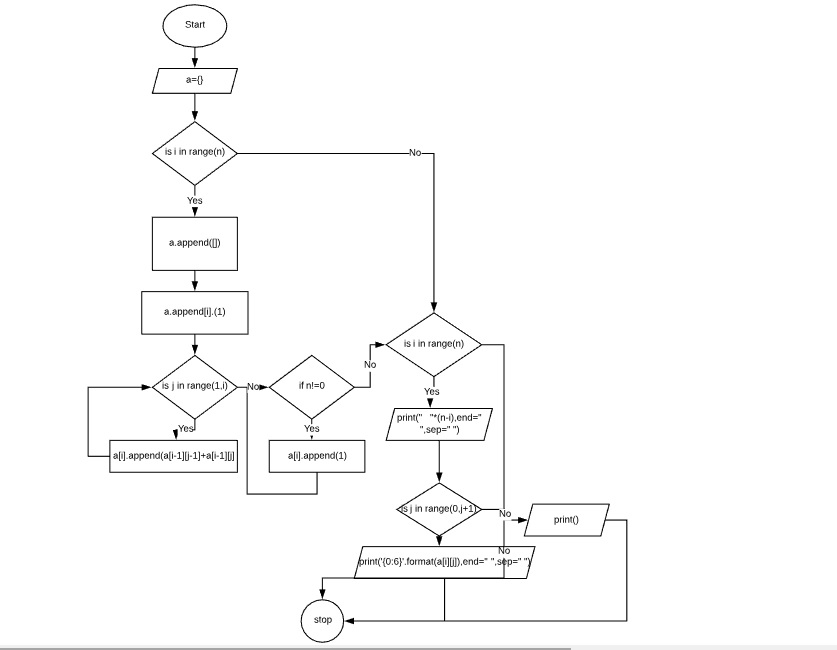


**4d. To print Pascal’s triangle:**

**Algorithm:**

1. start
2. Read number of rows to be printed.
3. Loop untill the range
4. Append the value of i if the statement is true.
5. Create a nested for loop in the range 1 to i
6. Append a[i-1][j-1]+a[i-1][j] if the condition is satisfied
7. Create another for loop in the range n and nested loop j in range i+1
8. stop

**Flow Chart:**



**Coding:**

n=int(input("Enter number of rows: "))

a=[]

for i in range(n):

a.append([])

a[i].append(1)

for j in range(1,i):

a[i].append(a[i-1][j-1]+a[i-1][j])

if(n!=0):

a[i].append(1)

for i in range(n):

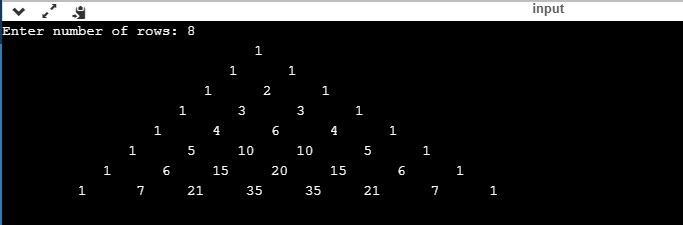
print(" "\*(n-i),end=" ",sep=" ")

for j in range(0,i+1):

print('{0:6}'.format(a[i][j]),end=" ",sep=" ")

print()

**Screen Shots:**



**Result:**

Thus the programs to

a. BINARY SEARCH

b. To extract digits from a paragraph

c. To find G.C.,D of two numbers

d. Solve Pascal’s triangle

are written in Python and the results are verified.

