

**Assignment On**

Pseudo Code and Flowchart of For Loop problems

**Department of CIS**

**Course title: Structured Programming**

**Subject code: CIS 115**

**Semester: Fall 2022**

|  |
| --- |
| **Submitted To** |
| Mr. Abdullah Bin Kasem Bhuiyan  Lecturer  Department of CIS |

|  |
| --- |
| **Submitted By** |
| Nabila Sarkar  ID: 222-16-675  Department of CIS |

**Due date: 21/10/2022 Submission date: 21/10/2022**

# **For Loop problems**

**1. Average of 10 numbers:**

1.Declare variables I num, sum=0

2.For i=1 to i<=10.

3. Display a message instructing you to enter the integer.

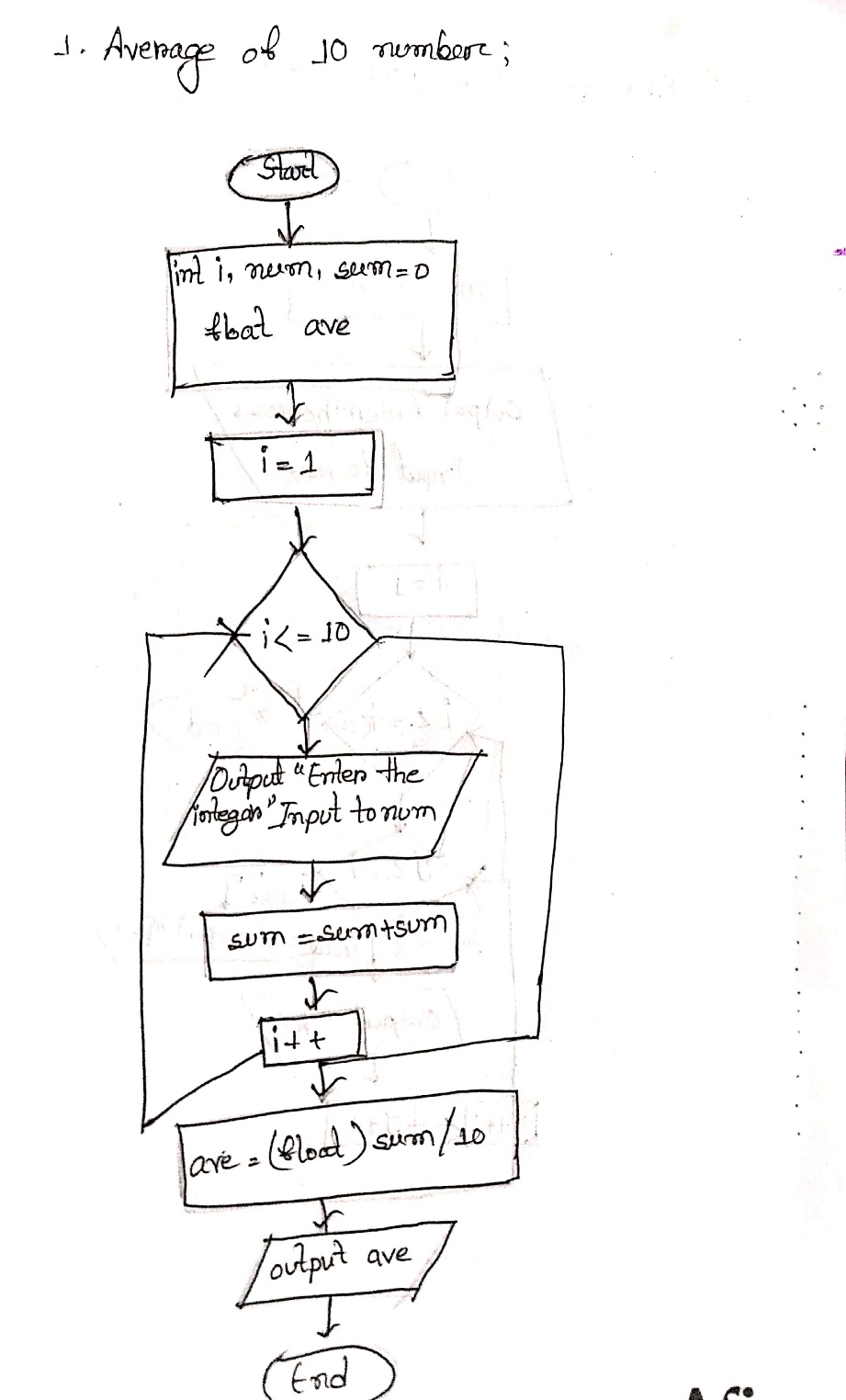
4. Provide input to the num variable.

5. Calculate sum=sum+num.

6. Increase the value by one.

7. Perform ave=sum/10 and convert the result to float data type.

8. Print "Average of ten integers," ave.

****

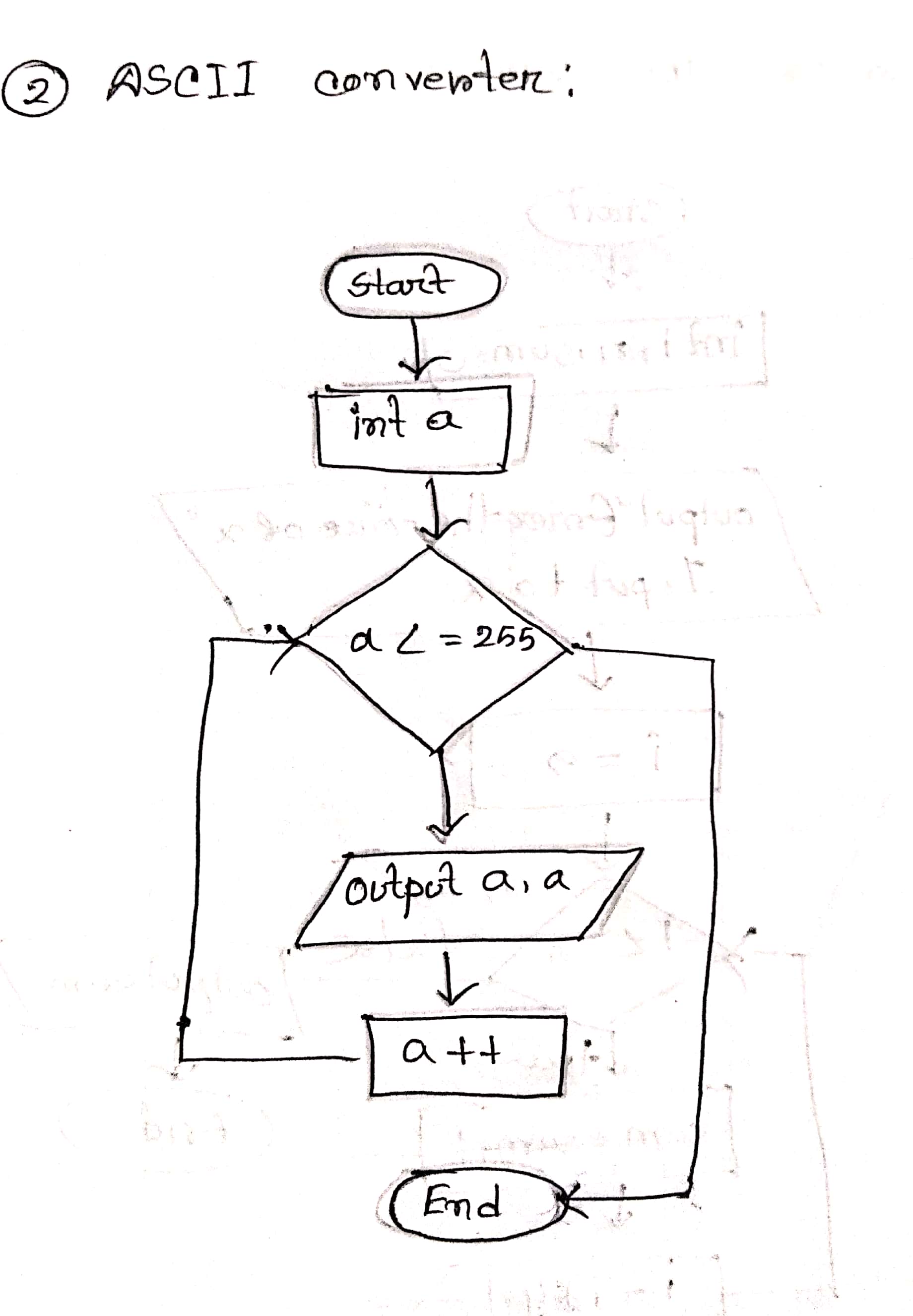
**2.ASCII converter:**

1. Create variable (a) to hold a numeric value.

2. For a=0 to a<=255.

3. Print the "ASCII value and characters," a, a.

4. Decrease a value by one.2.



**3.Overtime Calculator:**

1. Declare variables (num,hour) and (pay) to store numeric values and fractional values, respectively.

2. For num=1 to num<=10.

3. Display a message to enter an employee's working hours.

4. Set the hour variable.

5. Calculate pay as (hour-36)\*120.

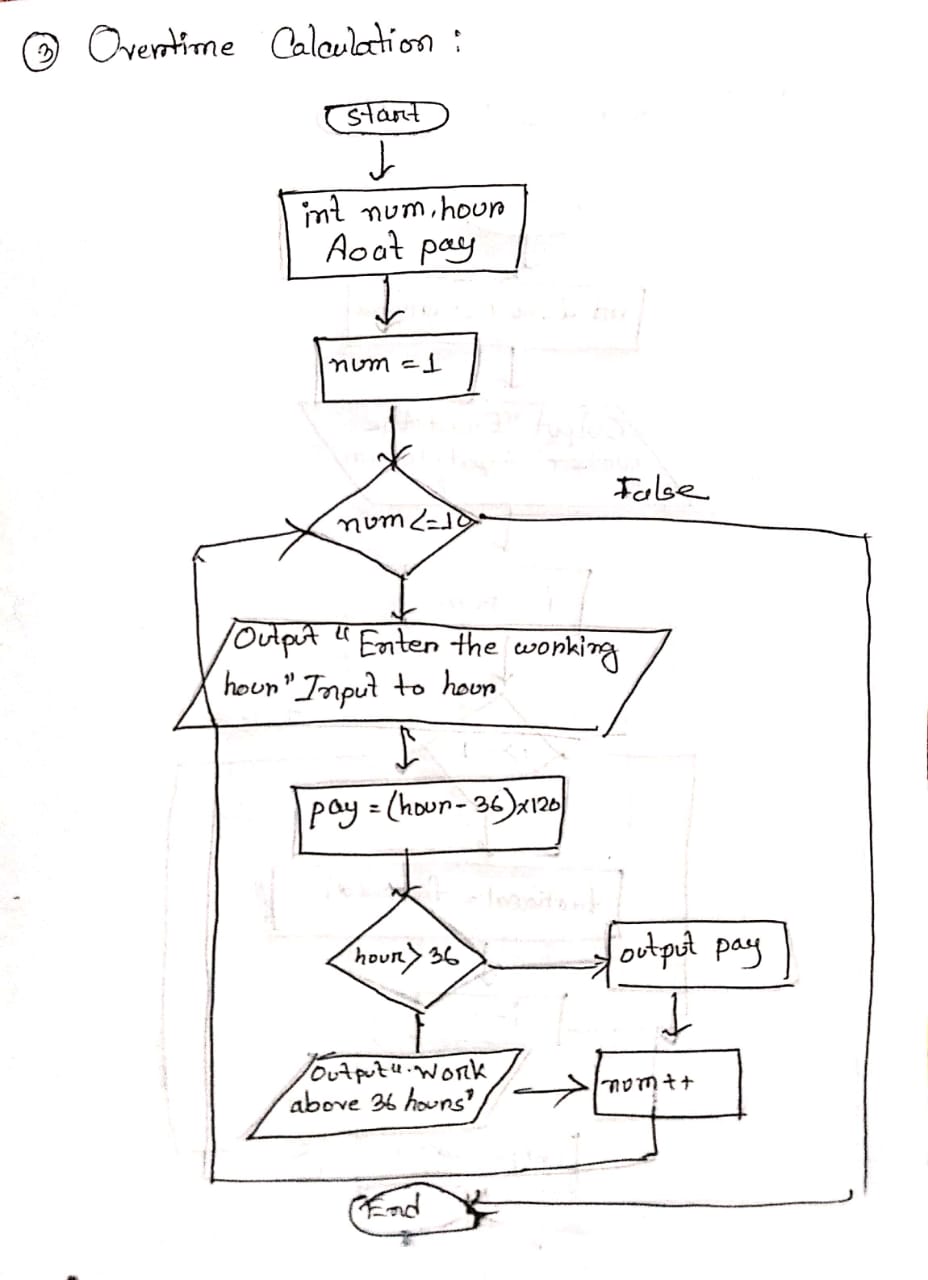
If the hour exceeds 36,

7. Print "Employee number and overpayment", num, pay.

8. Otherwise

9. Output "You have to work more than 36 hours to get overpay".

10. Increase the value of the count by one.

****

**4. De factorial:**

1. Declare variable (num,i,factorial=1) to store numeric value.

2. Show a message to enter the number.

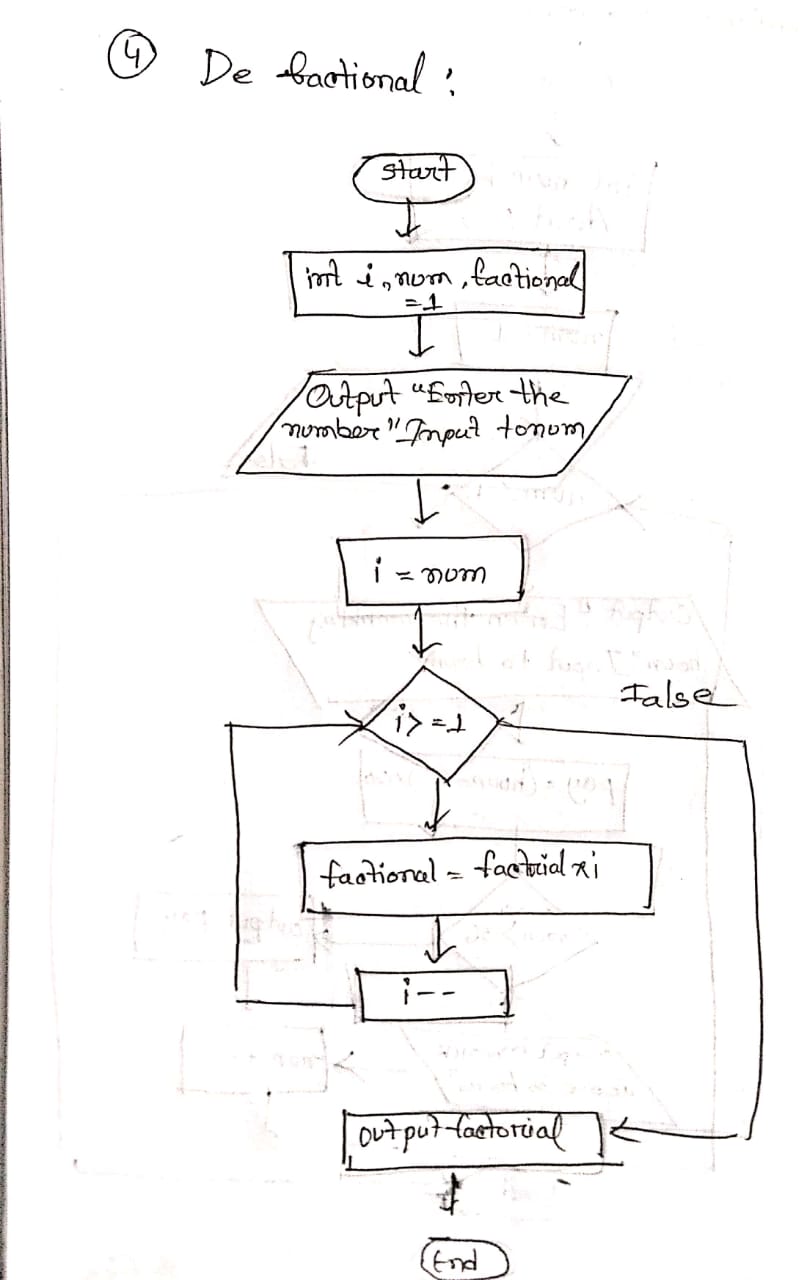
3. Input to num variable.

4. For i=num to i>=1.

5. Compute factorial=factorial\*i.

6. Decrement i value by one.

7. Output “Factorial of the number”, factorial.



**5) Prime time:**

1. Declare variable (i,num,isPrime=1) to store numeric value.

2. Show a message to enter the number.

3. Input to num variable.

4. If num>0

5. For i=2 to i<num.

6. If num%i==0.

7. isPrime = 0.

8. Break statement.

9. If isPrime==0.

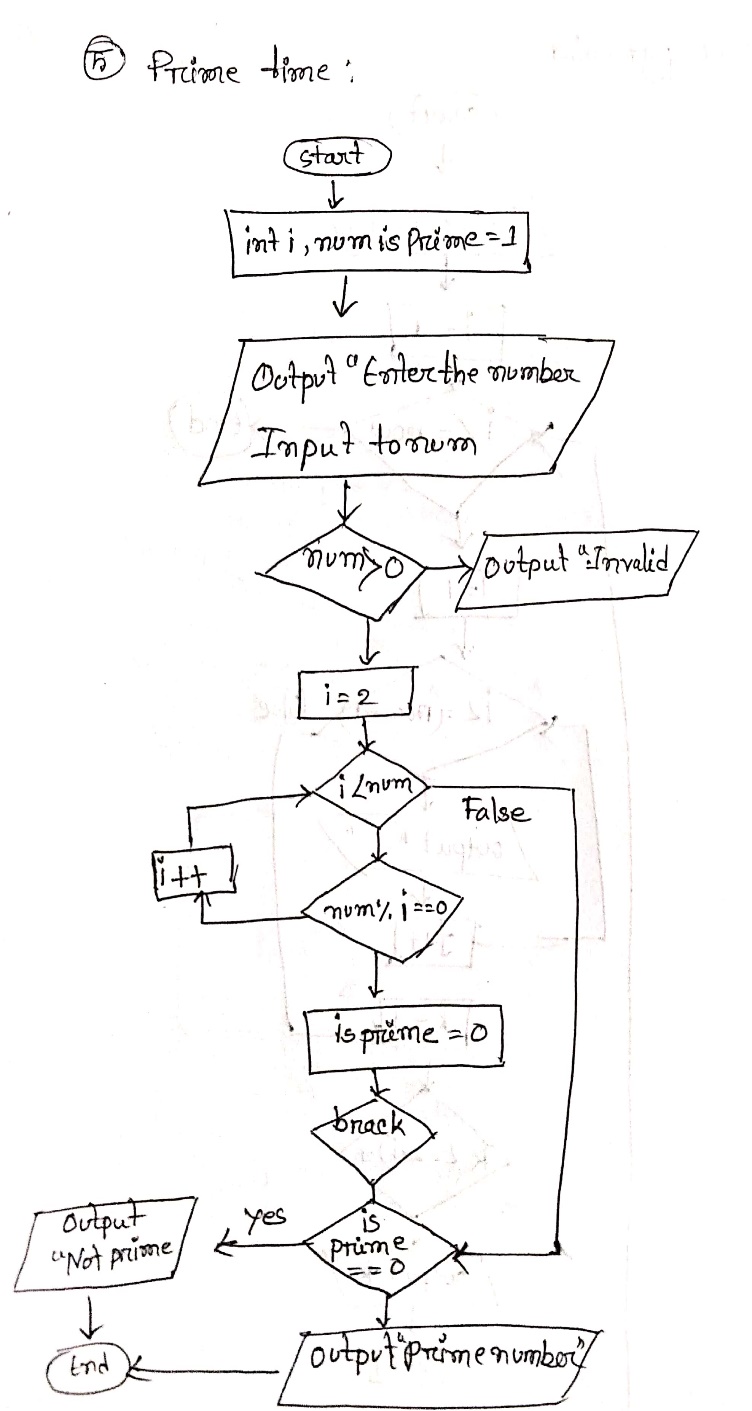
10. Output “Not prime number”.

11. Otherwise

12. Output “Prime number”.

13. Otherwise

14. Output “Invalid”

****

**6) Star pyramid:**

1. Declare variable (row,i,j,k) to store numeric value.

2. Show a message to enter the number of rows.

3. Input to row variable.

4. For i=1 to i<=row.

5. For j=1 to j<=(row-i).

6. Output “ ”.

7. Increment j value by one.

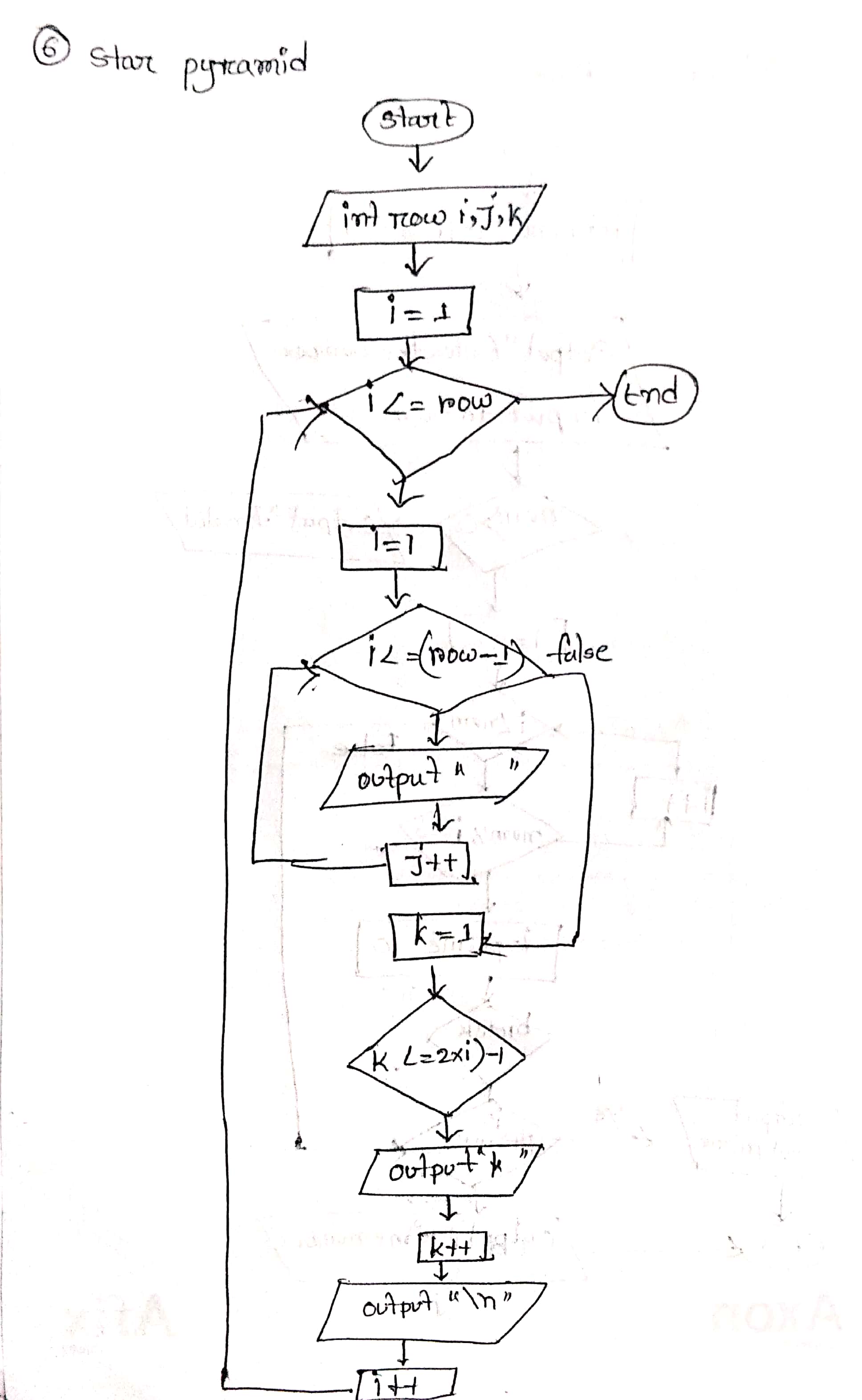
8. For k=1 to k<=(2\*i)-1.

9. Output “\*”.

10. Increment k value by one.

11. Output “\n”.

12. Increment i value by one.



**7) Factorial fun:**

1. Declare variable (num1,num2,N1,N2,f1=1,f2=1) to store numeric value.

2. Show a message to enter two numbers.

3. Input to num1 and num2 variable.

4. If (num1>=0&&num1<=20)&&(num2>=0&&num2<=20)

5. For N1=num1 to N1>=1.

6. Compute f1=f1\*N1.

7. Decrement N1 value by one.

8. For N2=num2 to N2>=1.

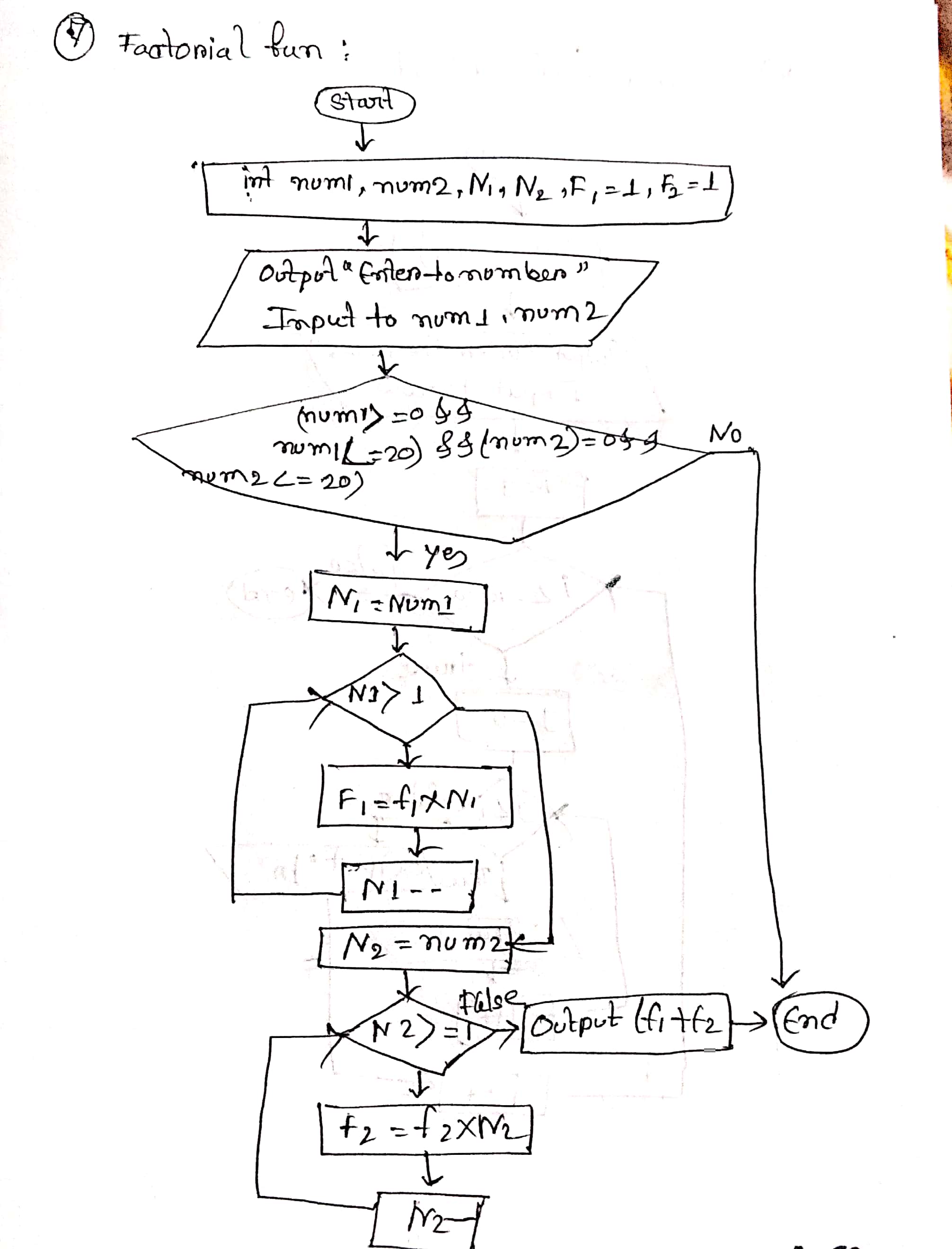
9. Compute f2=f2\*N2.

10. Decrement N2 value by one.

11. Output “Sum of factorials”, f1+f2.

12. Otherwise

13. Output “Invalid”.



**8) Sum of the series:**

1. Declare variable (i,n,sum=0) to store numeric value.

2. Show a message to enter the value of n.

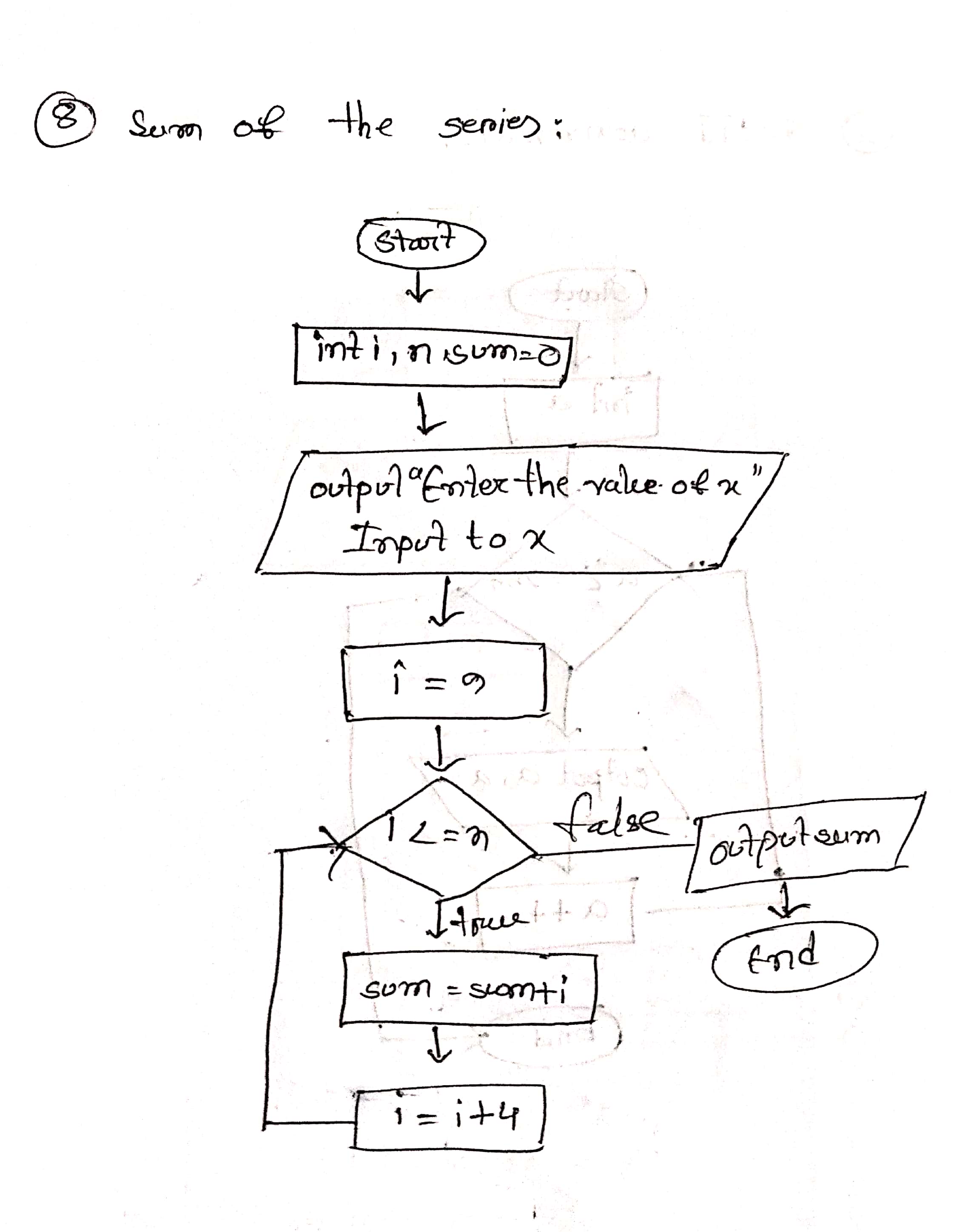
3. Input to n variable.

4. For i=9 to i<=n.

5. Compute sum=sum+i.

6. i=i+4.

7. Output “Sum of the series”, sum.

****

**9) Half of star pyramid:**

1. Declare variable (row,i,j) to store numeric value.

2. Show a message to enter the number of rows.

3. Input to row variable.

4. For i=1 to i<=row.

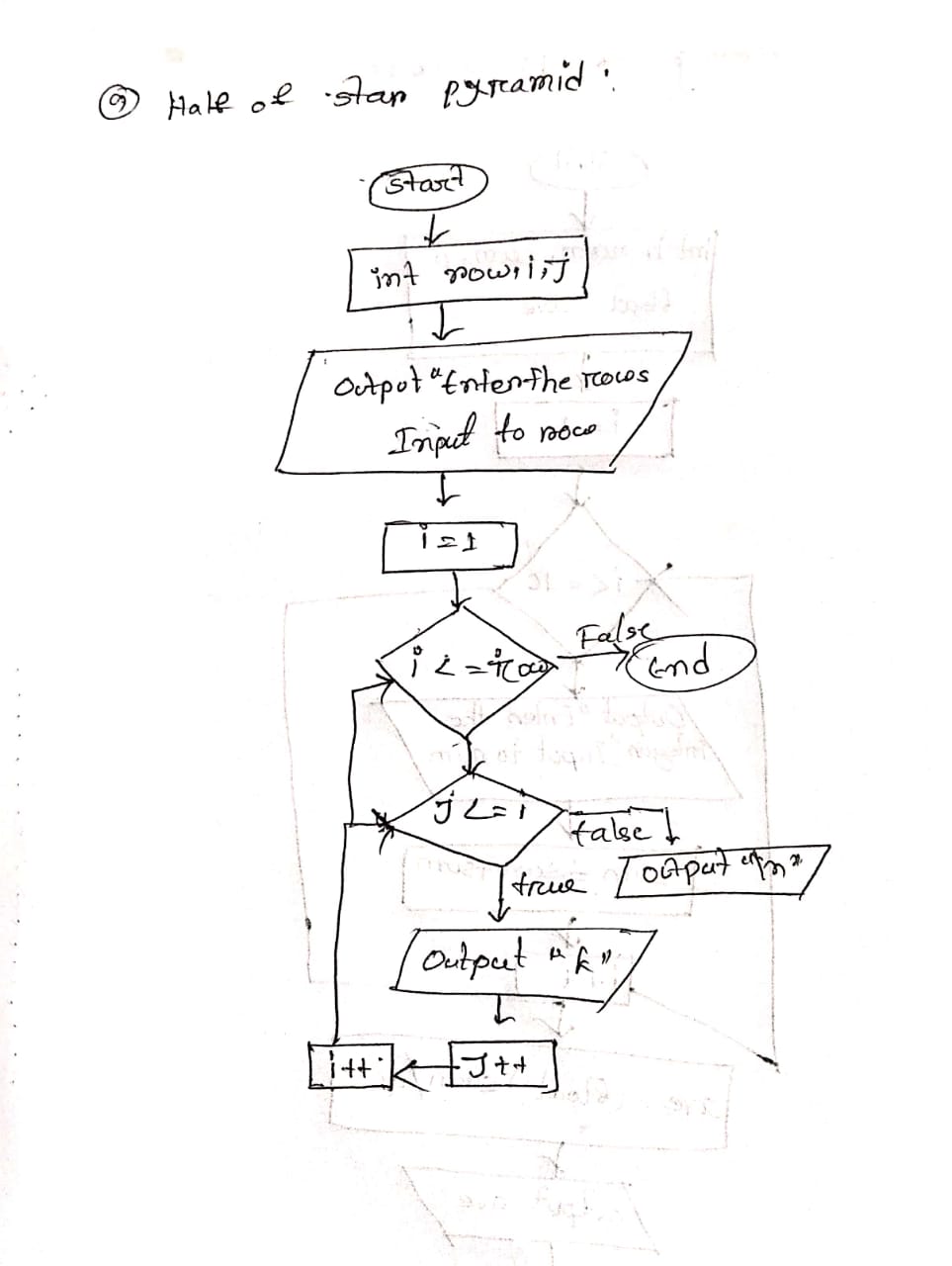
5. For j=1 to j<=i.

6. Output “\*”.

7. Increment j value by one.

8. Output “\n”.

9. Increment i value by one.

****

**10) Binary half pyramid:**

1. Declare variable (row,i,j) to store numeric value.

2. Show a message to enter the number of rows.

3. Input to row variable.

4. For i=1 to i<=row.

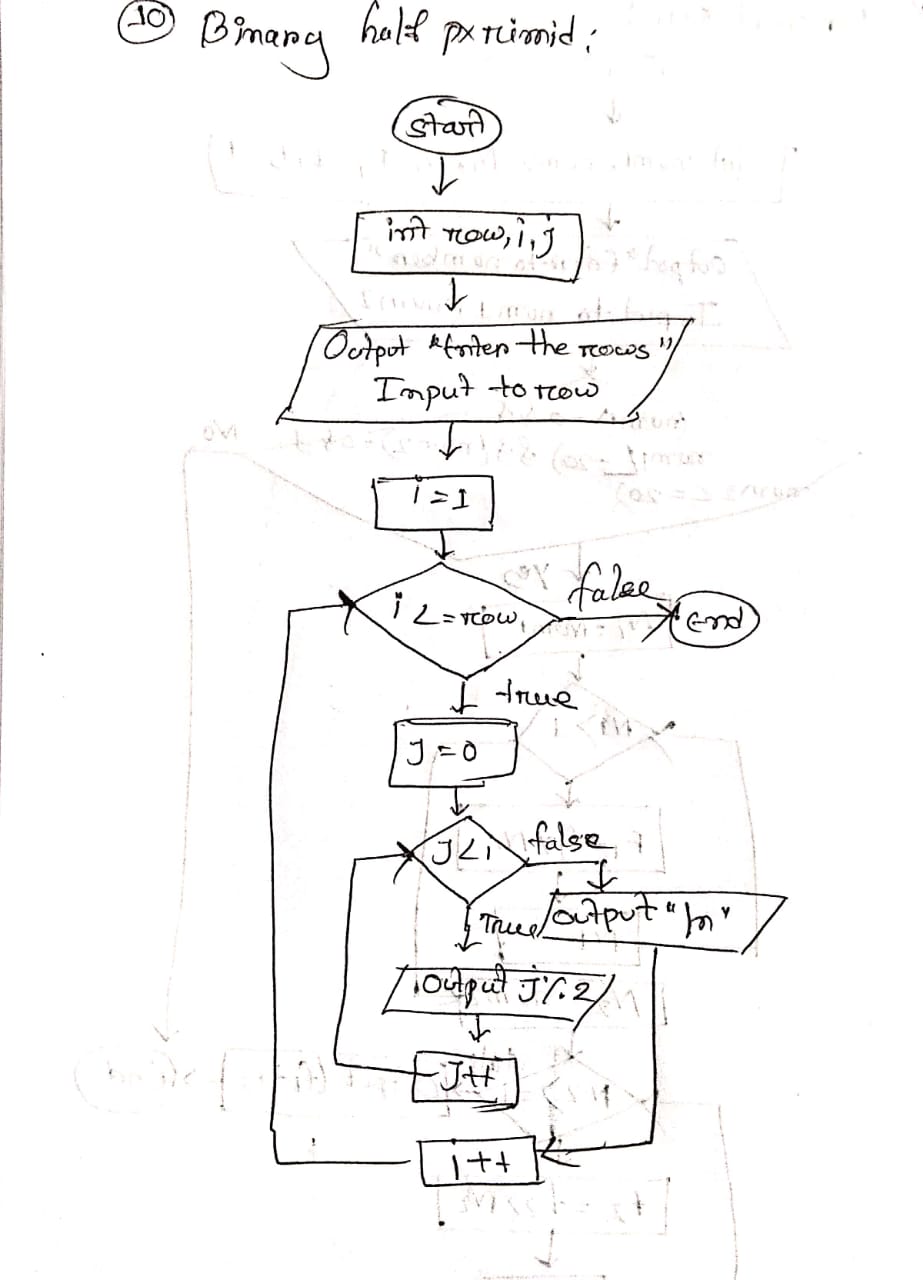
5. For j=0 to j<i.

6. Output j%2.

7. Increment j value by one.

8. Output “\n”.

9. Increment i value by one.



**11) Number sequence pyramid:**

1. Declare variable (row,i,j) to store numeric value.

2. Show a message to enter the number of rows.

3. Input to row variable.

4. For i=1 to i<=row.

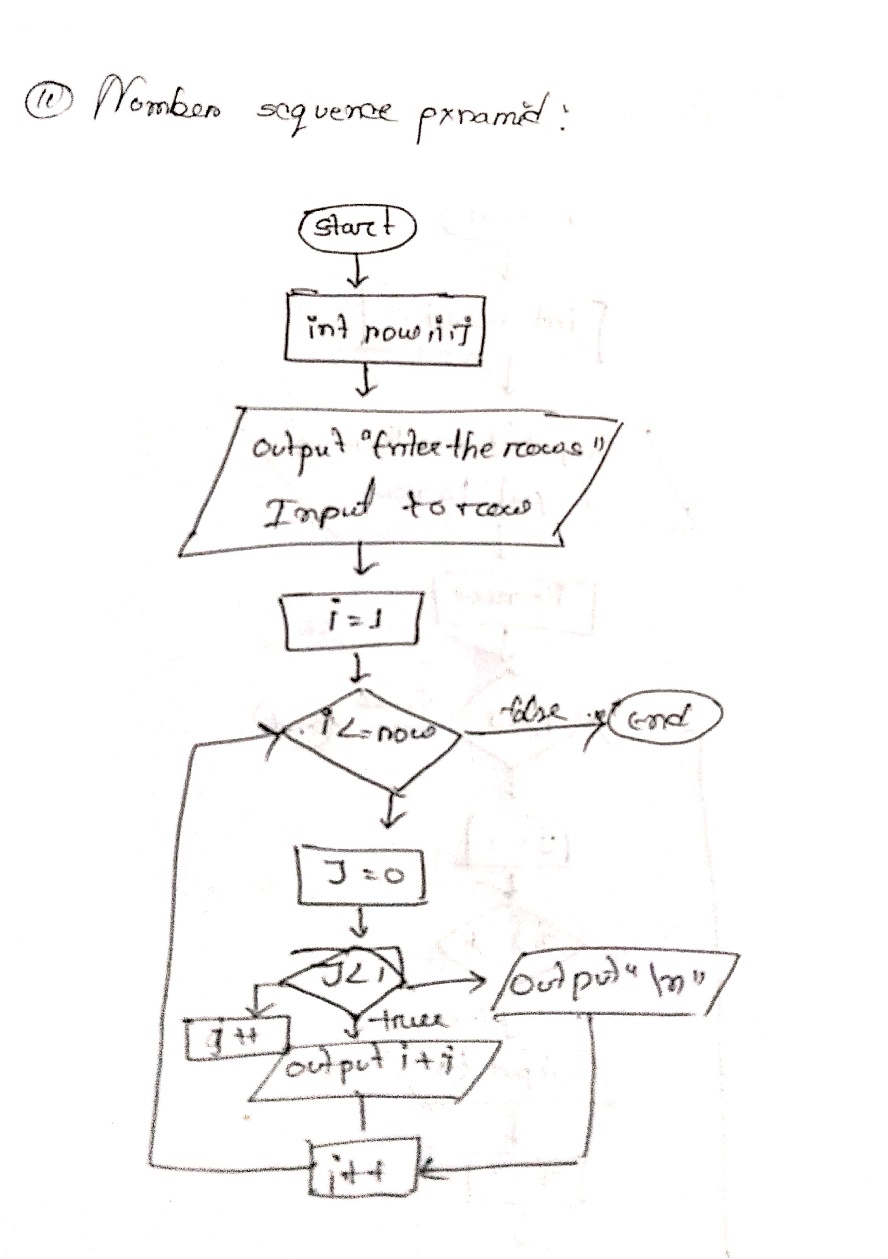
5. For j=0 to j<i.

6. Output i+j.

7. Increment j value by one.

8. Output “\n”.

9. Increment i value by one.

****

**12) Reverse “A” pyramid:**

1. Declare variable (row,i,j) to store numeric value.

2. Show a message to enter the number of rows.

3. Input to row variable.

4. For i=row to i>=1.

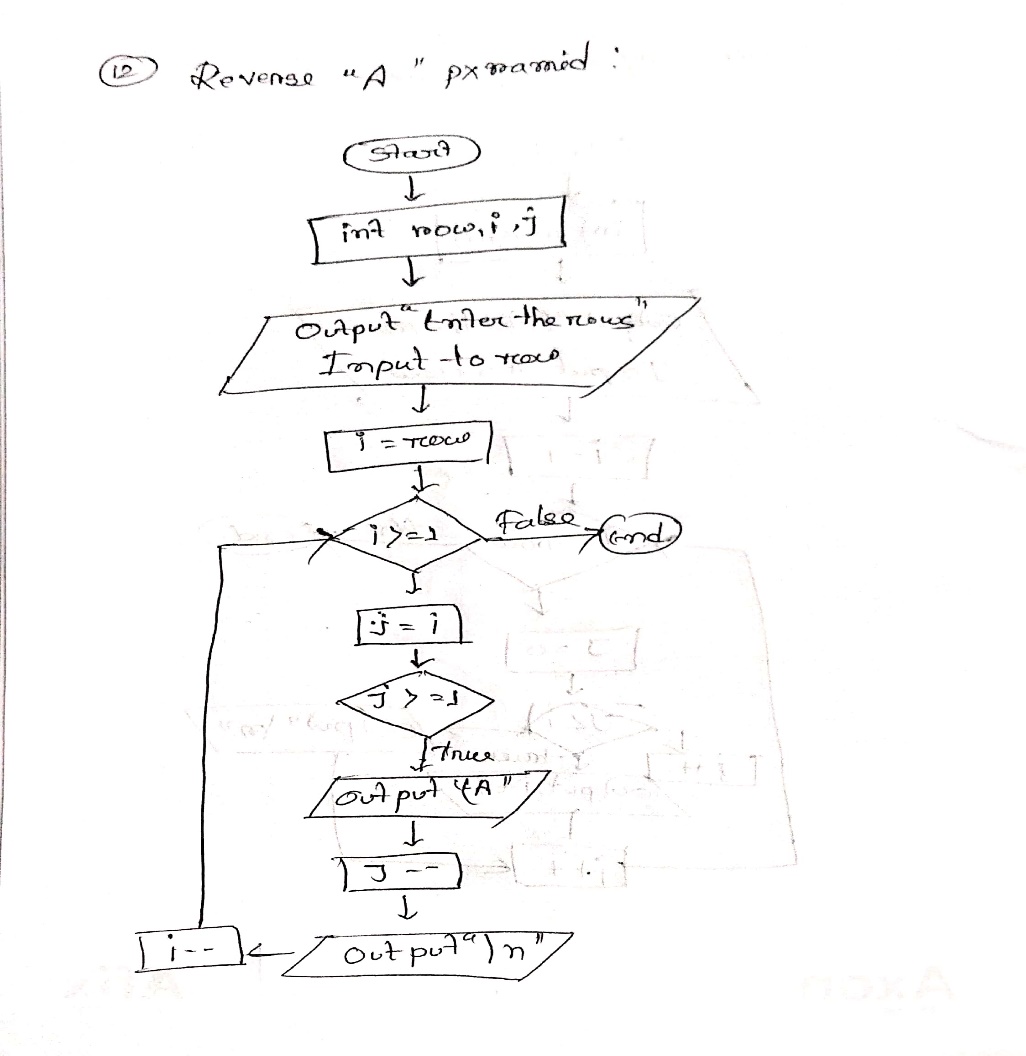
5. For j=i to j>=1.

6. Output “A”.

7. Decrement j value by one.

8. Output “\n”.

9. Decrement i value by one.

****

**13) Reverse number sequence:**

1. Declare variable (row,i,j) to store numeric value.

2. Show a message to enter the number of rows.

3. Input to row variable.

4. For i=row to i>=1.

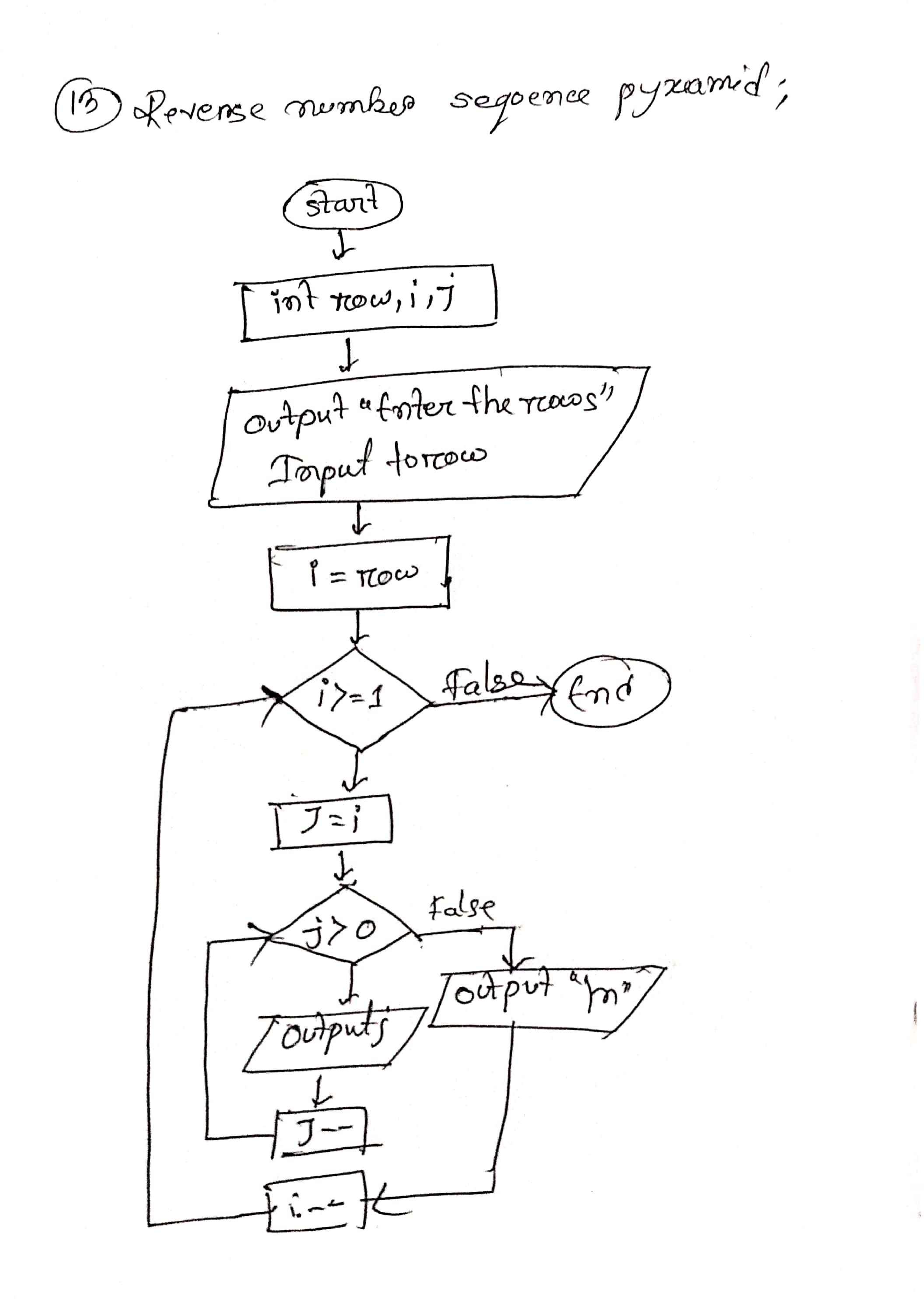
5. For j=i to j>0.

6. Output j.

7. Decrement j value by one.

8. Output “\n”.

9. Decrement i value by one.

****