1) What is an algorithm?

Answer:

A step-by-step solution to a problem is called an algorithm.

2) Explain need of an algorithm?

Answer:

On a number of occasions, we humans need to do repetitive tasks. Manually performing these tasks will require a lot of man power and resources which is inefficient. To perform such tasks we take help from computers by writing computer programs which utilize algorithms to solve and provide a step-by-step solution to such problems.

3) Write an algorithm to find average age of a group of 10 players?

Answer:

```
1. Start
2. Initialize “Sum of ages”
3. Initialize “count=0”
4. Input “age”
5. Count = count + 1
6. Sum of ages = sum of ages + age
7. If “count ≤ 9”
       [Go to line no.04]
   Else
       [(sum of ages)/10=average age of players]
8. Print “Average age of players”
9. End
```

4) Write algorithm to this problem:

Explain steps involve in drawing of a flowchart.

“Ramshewak goes to market for buying some fruits and vegetables. He is having a currency of Rs 500 with him for marketing. From a shop he purchases 2.0 kg Apple priced Rs. 50.0 per kg, 1.5 kg Mango priced Rs.35.0 per kg, 2.5 kg Potato priced Rs.10.0 per kg, and 1.0 kg Tomato priced Rs.15 per kg. He gives the currency of Rs. 500 to the shopkeeper. Find out the amount shopkeeper will return to Ramshewak. And also tell the total item purchased.”

Explain steps involve in drawing of a flowchart.
Explain uses of Flowchart.

Answer:

Algorithm:

1. Start
2. Go to Market
3. Input Total amount available (Total amount available = Rs.500/-)
4. Total Cost = 0
5. Count = 0
6. Input price of article
7. Total Cost = Total Cost + price of article
8. Count = count +1
9. While count ≤ Total no. of article (Goto step 7)
10. Print Amount left = Total amount – Total cost
11. Print "No. of articles purchased = count"

Steps and thought process involved in drawing the following flowchart:

The given problem requires the calculation of the amount left after shopping and the number of items bought by the customer.

(1) The first step will be to take in the input of the total amount from the user.

(2) After that a counter has been put into place so that the amount of all the articles can be summed together for the calculation of the total cost. The counter will also determine the number of items bought which will be equal to the number of times the loop ran.

(3) The loop will continue to add the price of each item into the total cost of all the previous items entered.

(4) When the prices of all the articles has been entered the user will instruct the program to calculate the total cost, the amount left as well as the number of articles bought (which will be equal to the number of times the counter ran).

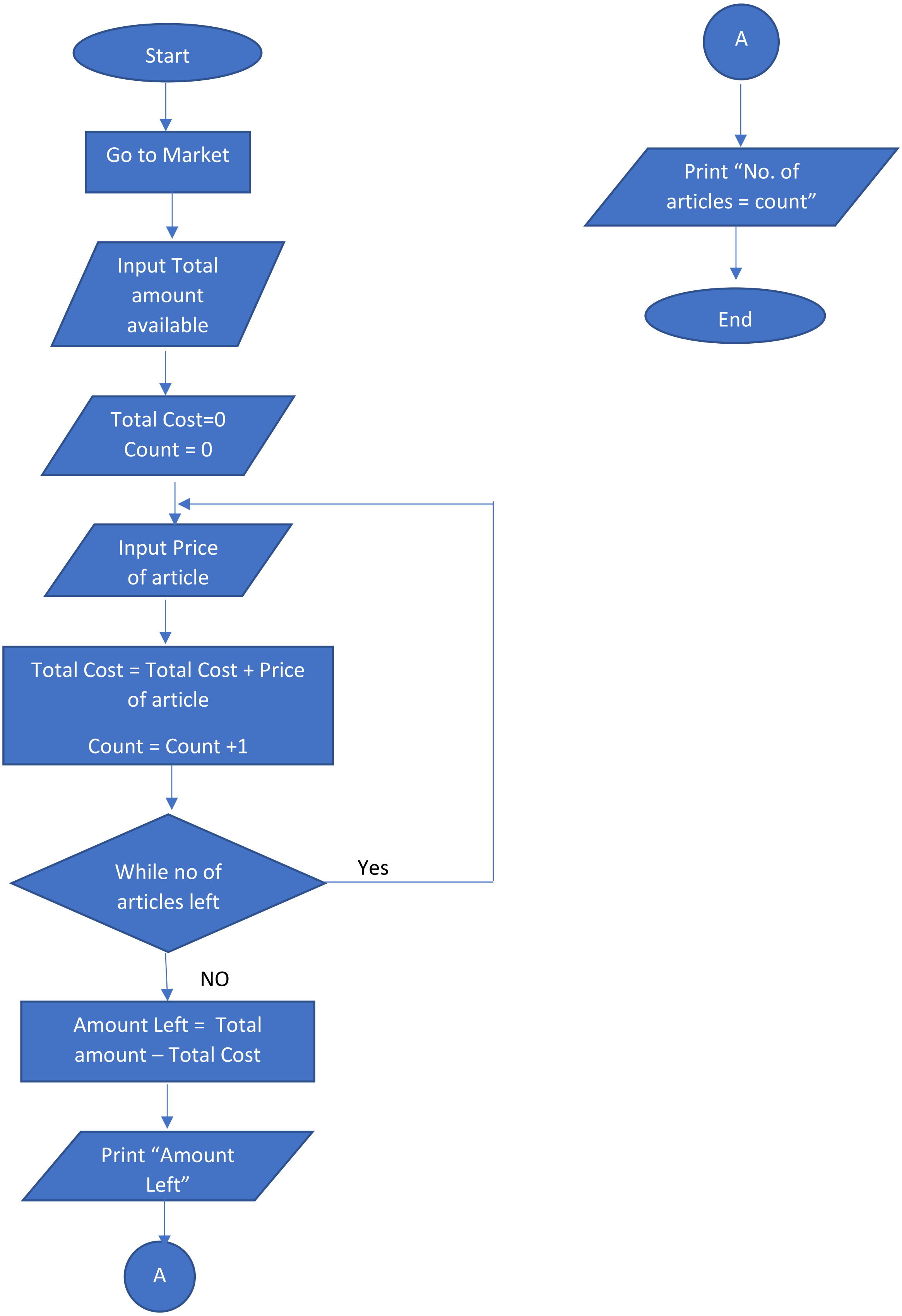
(5) Finally, the total cost, amount left and number of items bought will be printed on the screen.

Uses of a Flowchart:

A flowchart plays a key role in determining the functionality of any program. It is basically a blueprint of the program which states the functionality of a program in a precise step-by-step manner. There are specific shapes which we use while drawing the flowchart diagram of any program.

- A rectangle is used to show a processing step.
- A diamond is used when a decision-making step is involved in the flowchart.
- A rhombus is used to show the input or output function.
- Arrows are used to show the direction in which the program will proceed after the completion of a particular step

Flowchart:


Start
Go to Market
Input Total amount available
Total Cost=0
Count = 0
Input Price of article
Total Cost = Total Cost + Price of article
Count = Count +1
While no of articles left
Yes
NO
Amount Left = Total amount – Total Cost
Print “Amount Left”
A
A
Print “No. of articles = count”
End

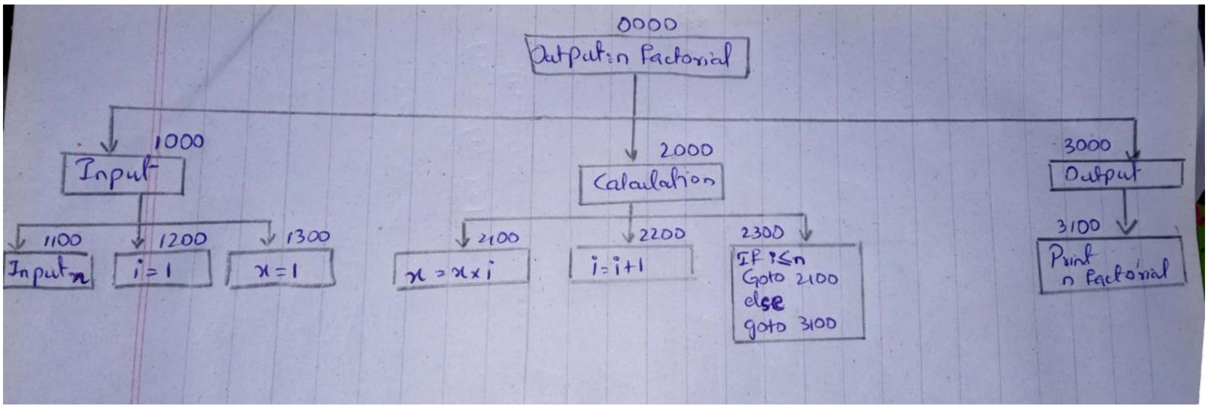
5) Draw the PAC, HIPO, IPO, and flowchart and write the pseudo code for the following problems

- **Find factorial of N?**

PAC:

Data	Process	Output
1. Input "n"	1. Input n	Output = Factorial of n
	2. i=1, x=1	
	3. x=x*i	
	4. i=i+1	
	5. If i ≤ n (Goto step 3)	
	6. Else (print "x")	
	7. End	

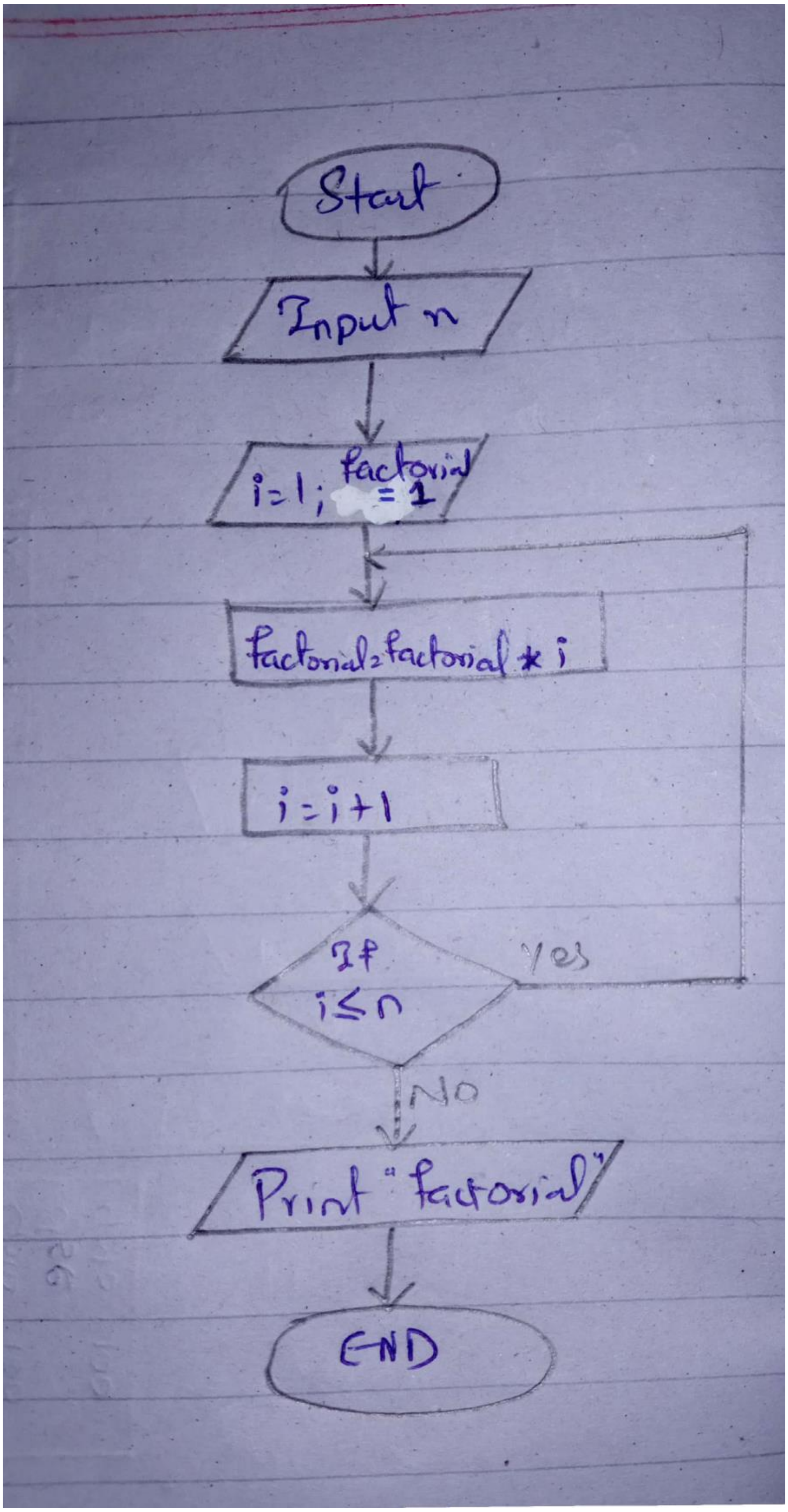
HIPO:



IPO:

Data	Process	Module	Output
1. Input n	1. Input n	1100	Output: n factorial
2. Declare x=1, i =1	2. Declare i=1	1200	
	3. Declare x=1	1300	
	4. x=x*i	2100	
	5. i=i+1	2200	
	6. If i≤n Goto 2100 7. Else, Goto 3100	2300	
	8. Output: factorial n	3100	
	9. Display Output	0000	

Flowchart:

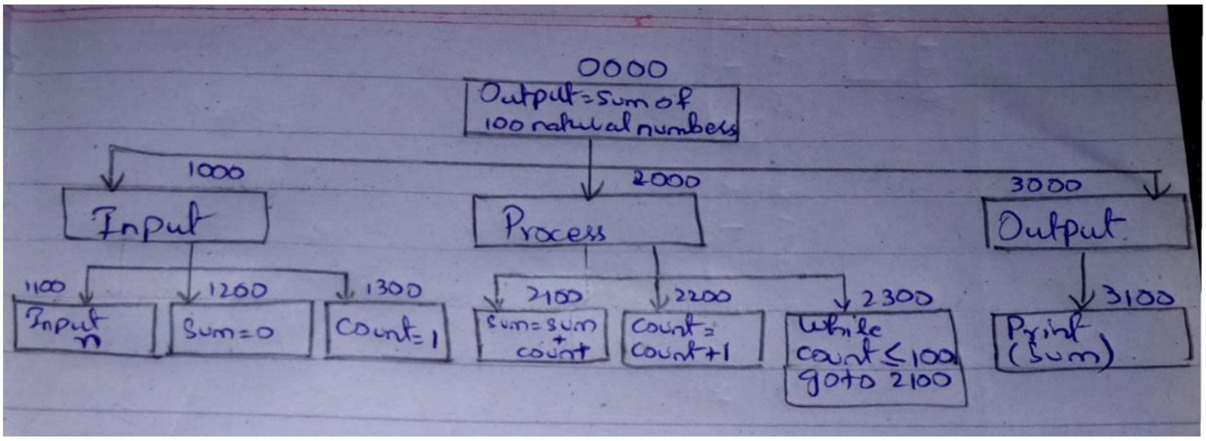

Start
Input n
i=1; factorial = 1
factorial=factorial * i
i=i+1
If
i≤n
Yes
NO
Print "factorial"
END

- **Find the sum of first 100 natural numbers.**

PAC:

Data	Process	Output
1. Input “n”	1. Input n	Output = Sum of first 100 natural numbers
	2. Sum = 0	
	3. Count = 1	
	4. Sum = sum + count	
	5. Count = count +1	
	6. While count ≤ 100 Goto step 4	
	7. Print sum	
	8. End	

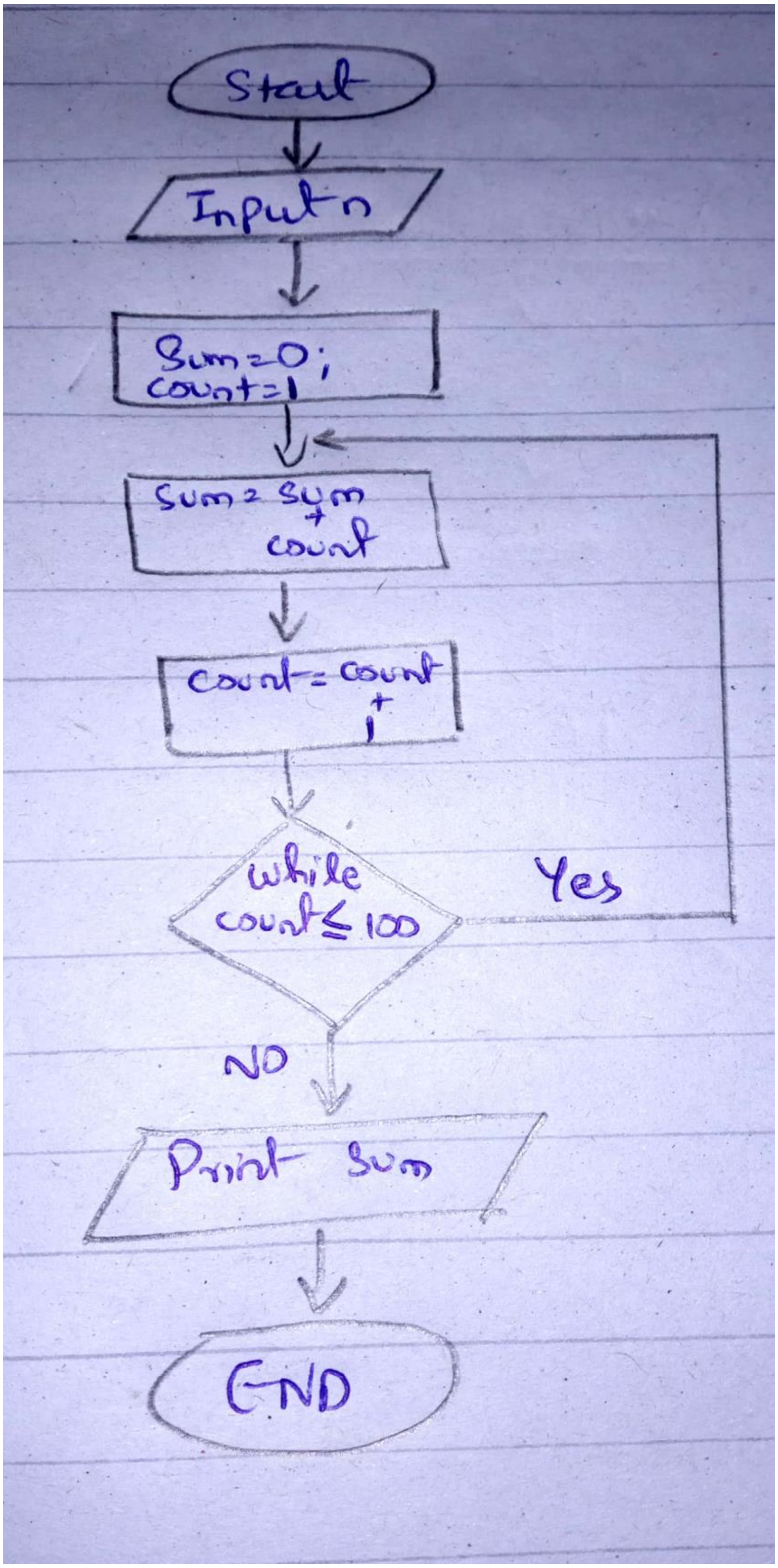
HIPO:



IPO:

Data	Process	Module	Output
1. Input n	1. Input n	1100	Sum of first 100 natural numbers
2. Declare sum =0	2. Declare sum=0	1200	
3. Declare count =1	3. Declare count =1	1300	
	4. Sum = sum + count	2100	
	5. Count = count +1	2200	
	6. While count ≤ 100	2300	
	7. Output: Sum of first 100 natural numbers	3100	
	8. Display Output	0000	

Flowchart:

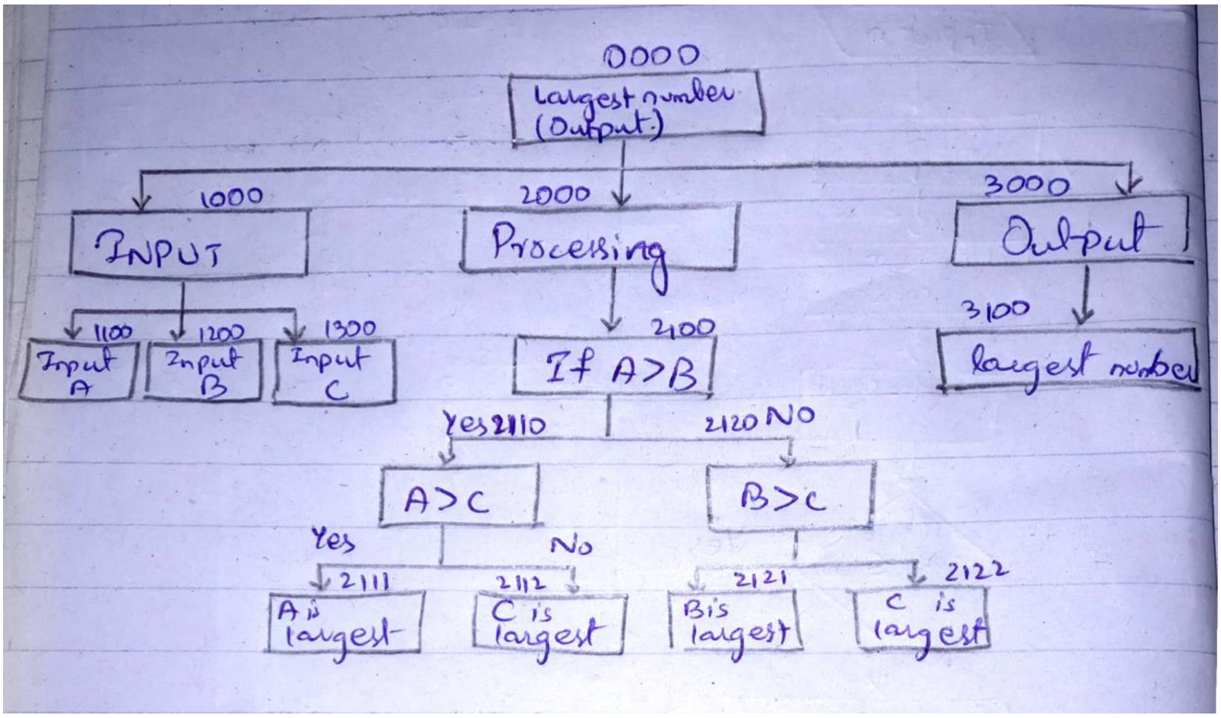

Start
Input n
Sum=0;
count=1
sum = sum + count
count = count + 1
while count ≤ 100
Yes
NO
Print sum
END

- **Find the largest of three numbers x, y and z.**

PAC:

Data	**Process**	**Output**
1. Input A, B, C	1. Input A, B, C	Output = Largest nubmer
	2. If A > B [Then, if A>C (A is the largest number) Else, (C is the largest number)] Else if B>C (B is the largest number) Else (C is the largest number)	
	3. End if	
	4. End	

HIPO:



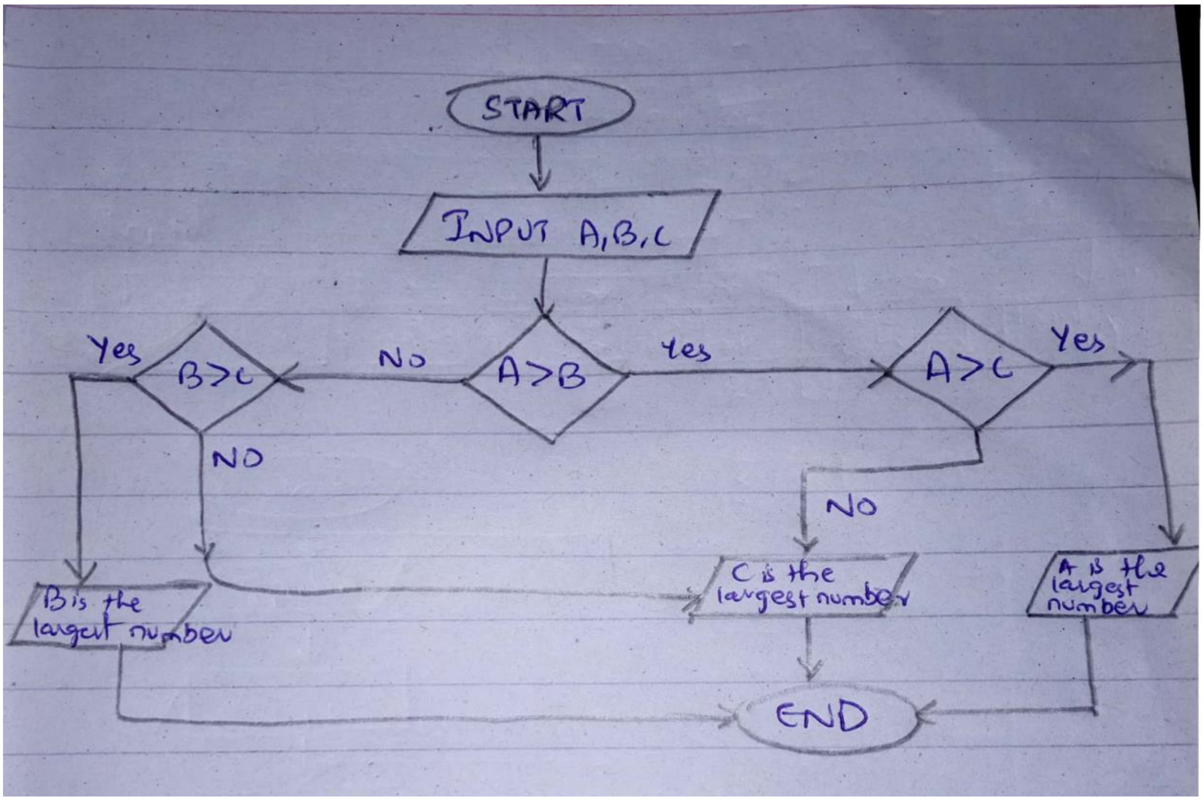
IPO:

Table is on next page

Data	**Process**	**Module**	**Output**
1. Input A	1. Input A	1100	Print “The largest number”
2. Input B	2. Input B	1200	
3. Input C	3. Input C	1300	
	4. If A>B Then, [If A>C (A is the largest number) Else (C is the largest number)] Elseif, B>C (B is the largest number) Else, C is the largest number.	2100 2110 2111 2112 2120 2121 2122	
	5. Print “The largest number”	3000 3100	
	6. Display Output	0000	

Flowchart:

Flowchart is on next page

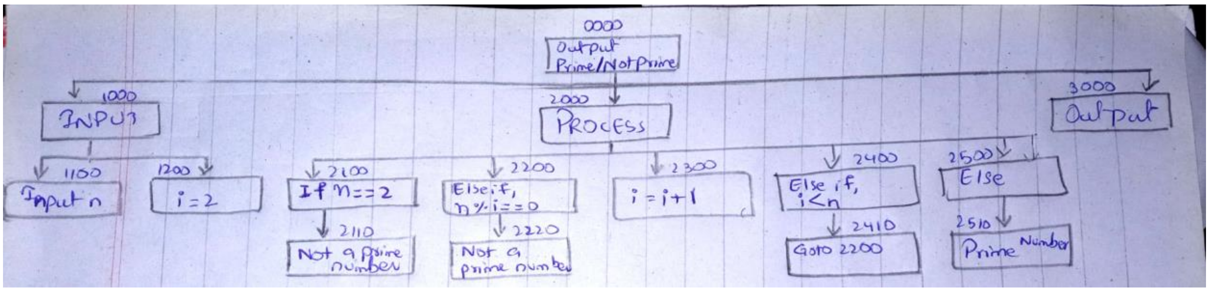


- **Determining prime number?**

PAC:

Data	**Process**	**Output**
Input n	Input n	Output “The number you entered is /is not a prime number”
	Declare i=2	
	If i = = 2 (Print “The number you entered is not a prime number).	
	Else if, [If, n % i = = 0 (Print “The number you entered is not a prime number”).	
	Else if, (i = i+1 If, (i < n) Goto Step 03 Else, Print (“The number you entered is a prime number.”). End if	
	Display Output	

HIPO:

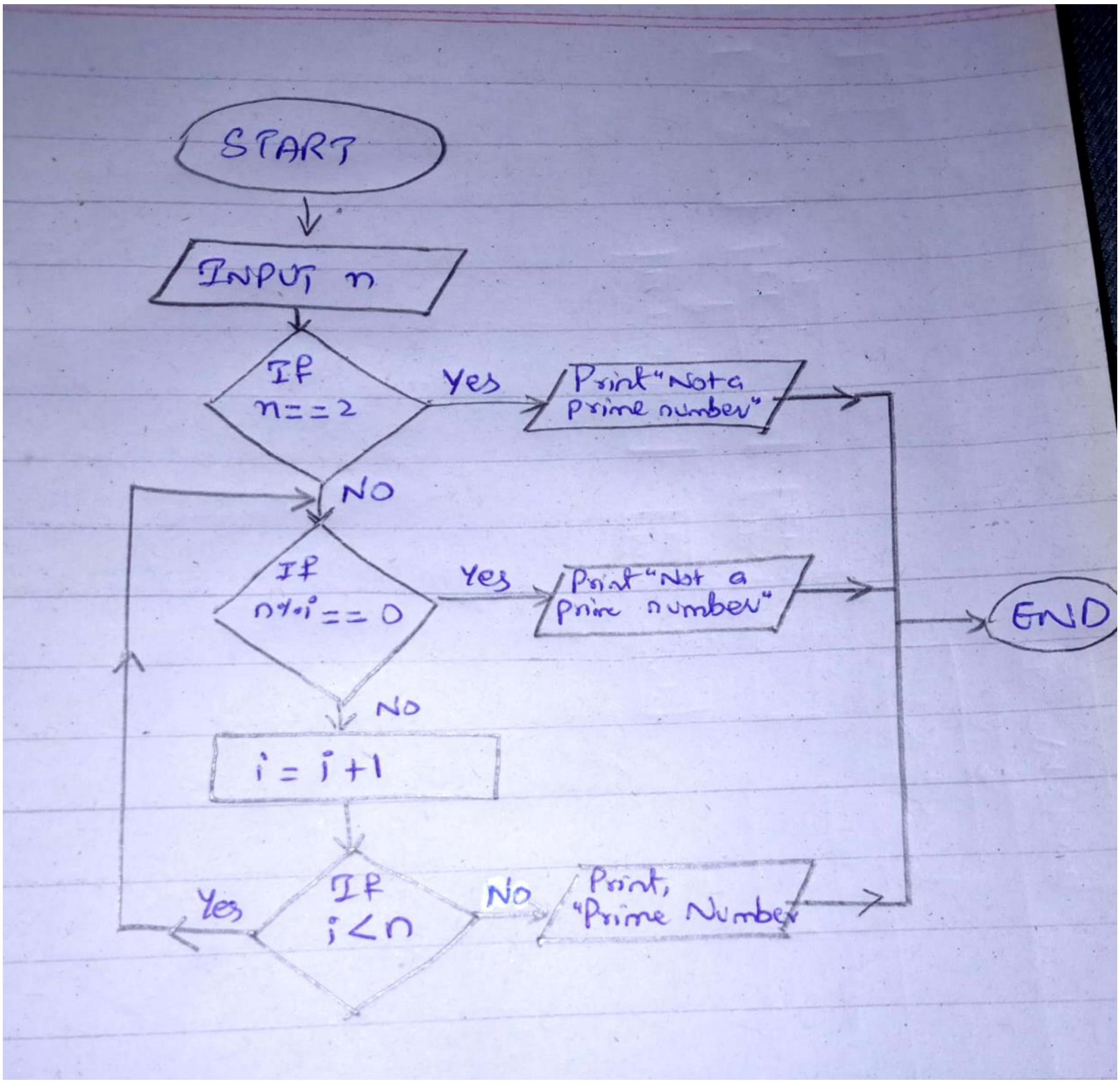


IPO:

Data	Process	Module	Output
Input n	Input n	1100	
i = 2	Declare i=2	1200	
	If (n==i) Print (“Not a prime number”)	2100 2110	
	Else if, (n%i = = 0) Print (Not a prime number)	2200 2210	
	i=i+1	2300	
	Else if, (i>n) Goto Module 2200	2400	
	Else, Print (“The number you entered is a prime number”)	2500	
	Output		

Flowchart:

Flowchart on next page

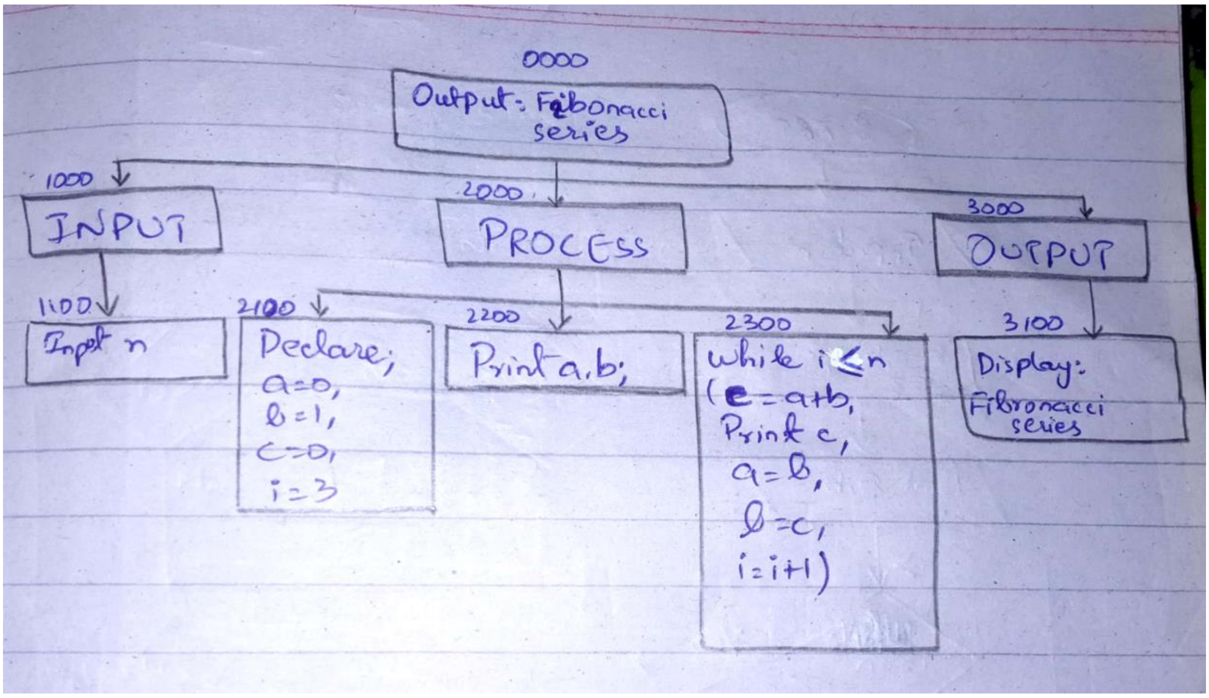

START
INPUT n
If n==2
Yes
Print "Not a prime number"
NO
If n%i==0
Yes
Print "Not a prime number"
END
NO
i = i+1
If i<n
Yes
No
Print, "Prime Number

- **A program which generates first 50 items of the Fibonacci series: 1, 1, 2, 3, 5, 8,**

PAC:

Data	**Process**	**Output**
1. Input n	1. Input n	Print (Fibonacci list)
	2. Declare a=0, b=1, c=0, i=3	
	3. Print a, b;	
	4. While i ≤ n 5. (c= a+b, 6. Print c, 7. a=b, 8. b=c, 9. i=i+1)	
	10. End	

HIPO:



0000
Output: Fibonacci series
1000
INPUT
2000
PROCESS
3000
OUTPUT
1100
Input n
2100
Declare; a=0, b=1, c=0, i=3
2200
Print a,b;
2300
while i≤n (c=a+b, Print c, a=b, b=c, i=i+1)
3100
Display: Fibronacci series

IPO:

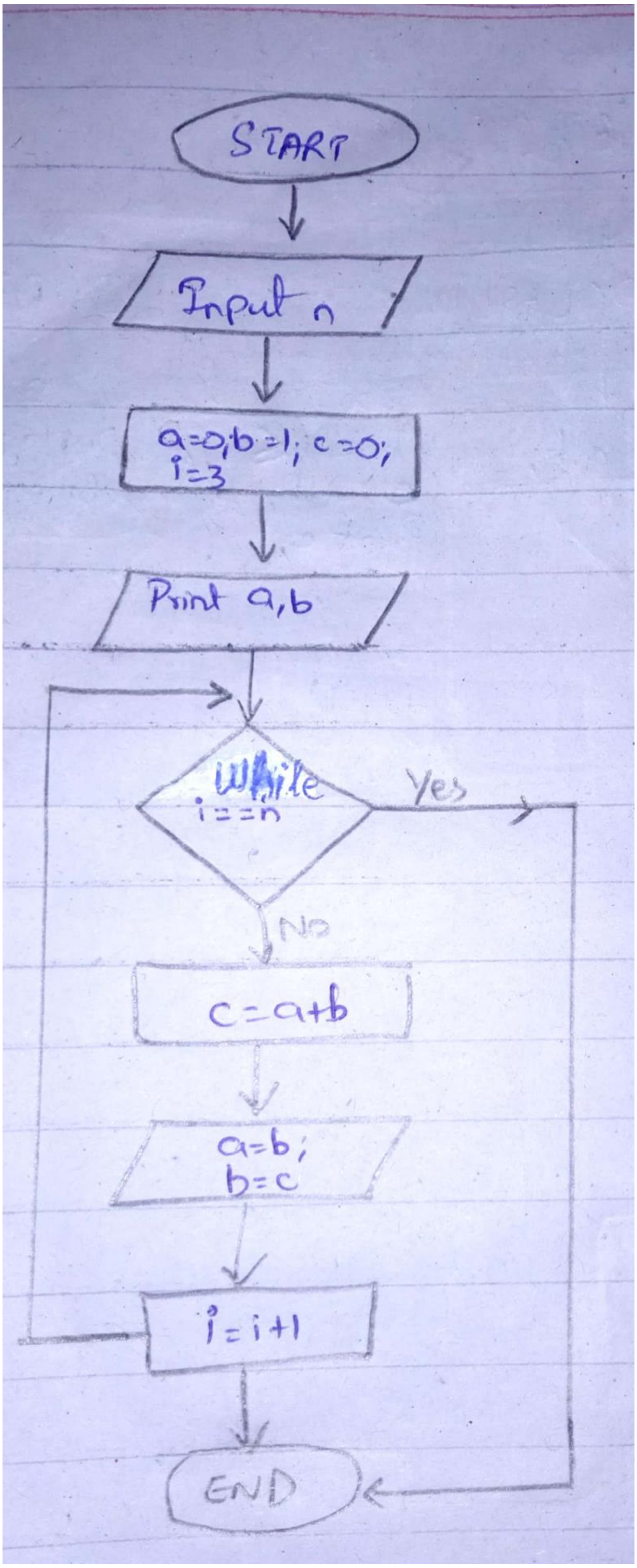
Data	**Process**	**Module**	**Output**
Input n	Input n	1100	Output: Fibonacci Series

	Declare
	a=0, b=1,
	c=0, i=3
		2100	

	Print a, b	2200	
	While i ≤ n (c=a + b, Print c, a=b, b=c, i=i+1)	2300	
	Display Fibonacci Series	3100	

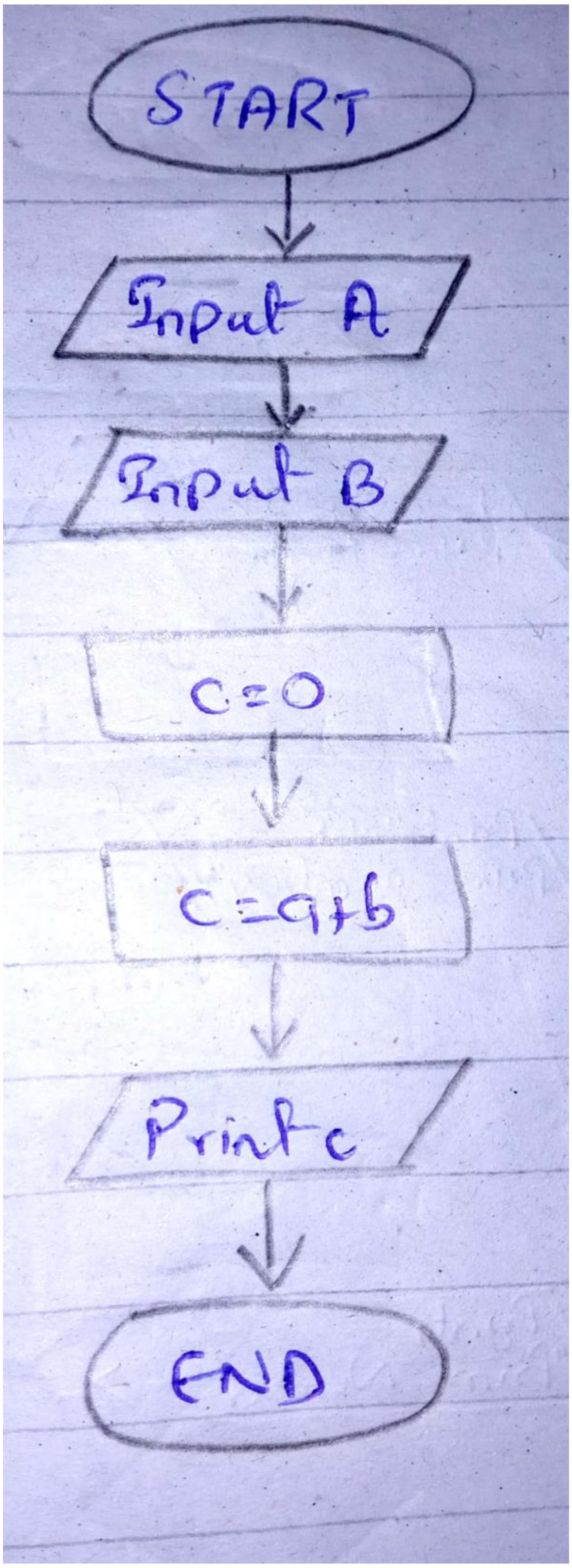
Flowchart:

Flowchart is on next page

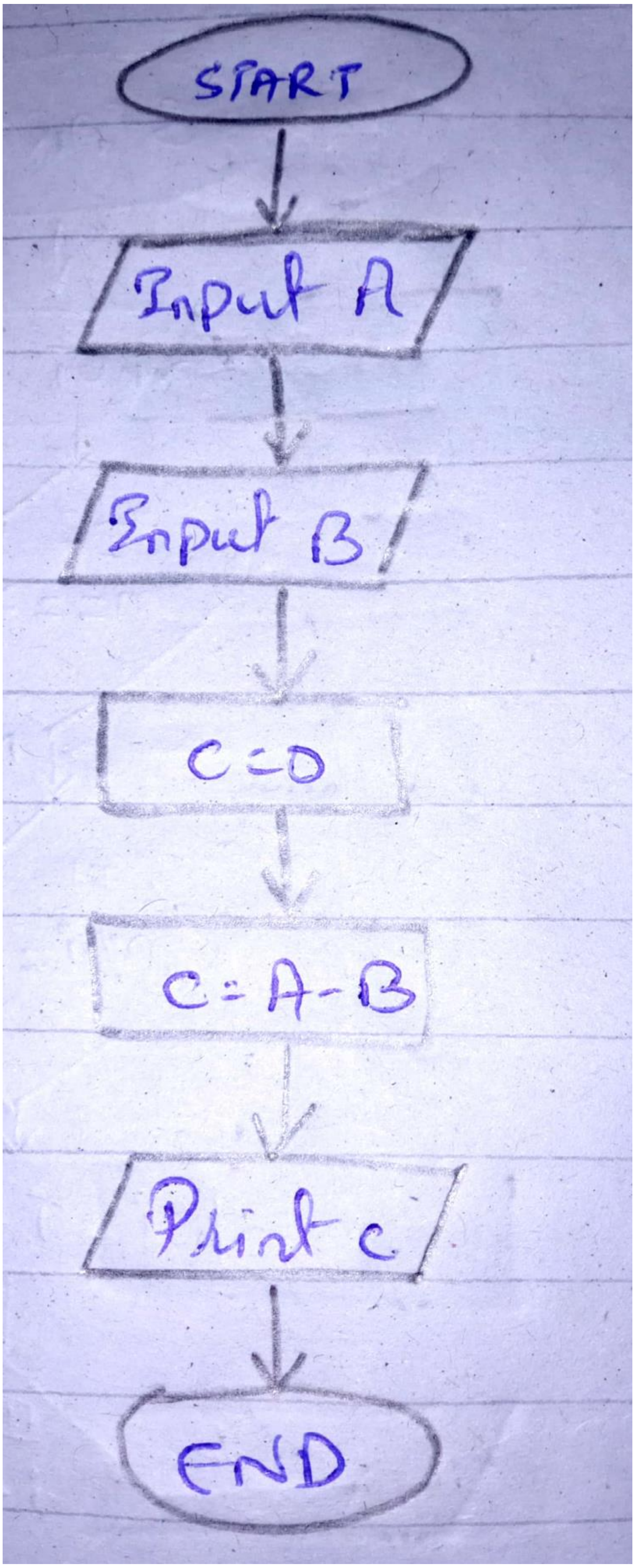

START
Input n
a=0,b=1; c=0;
i=3
Print a,b
While
i==n
Yes
No
c=a+b
a=b;
b=c
i=i+1
END

6. Draw Flowchart of 5 problems of your own choice. But people should come up with unique problems.

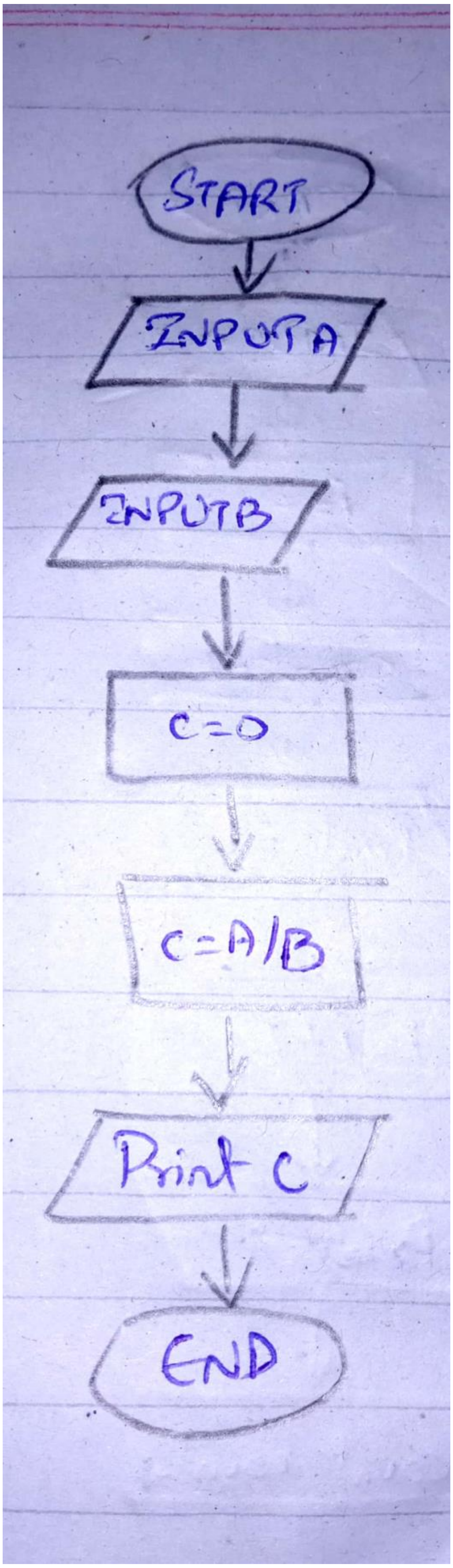
Problem 01) Calculate the sum of two numbers input by the user



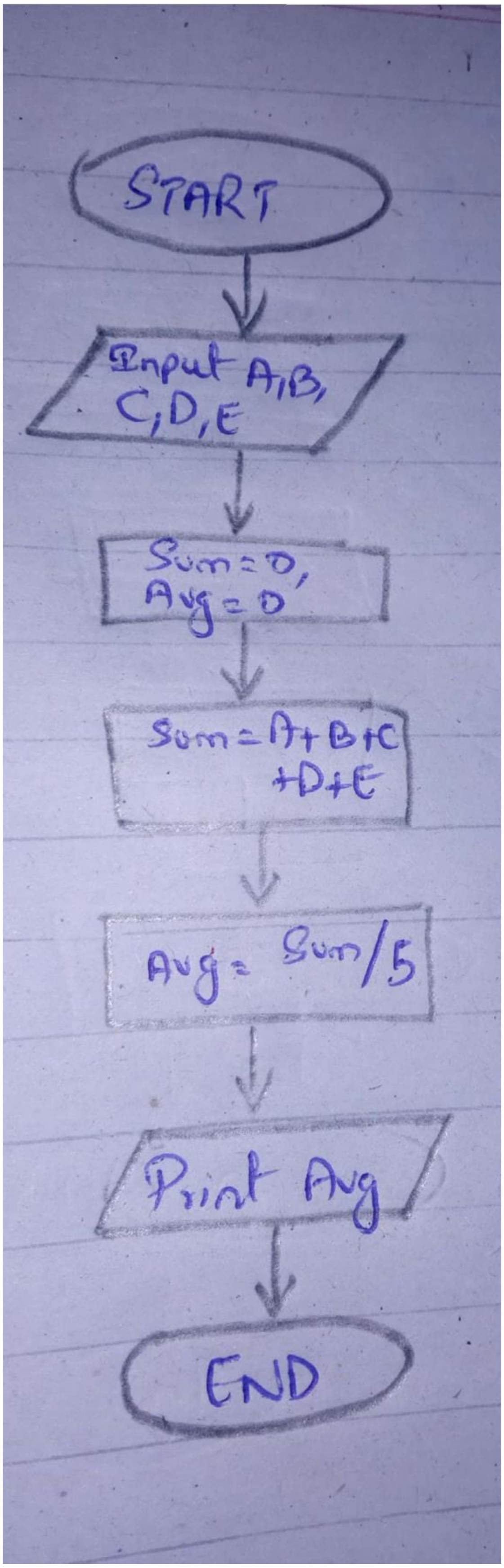
Problem 02) Calculate the difference of two numbers input by the user



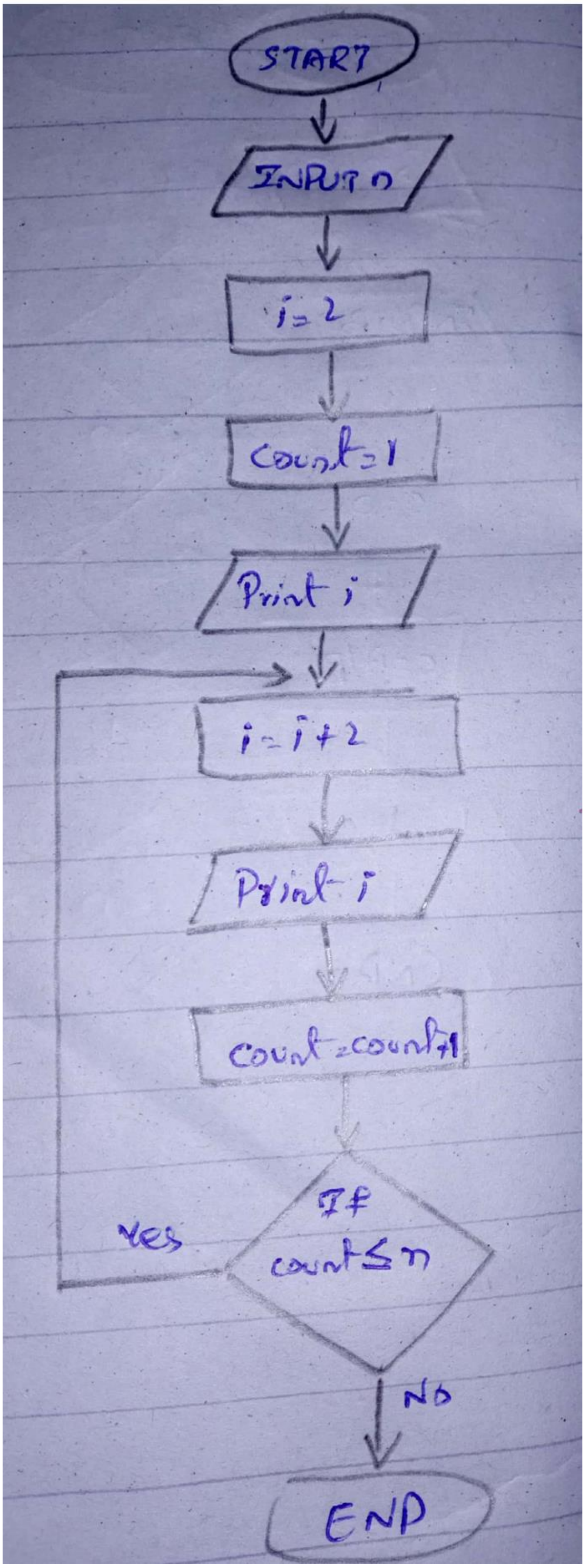
Problem 03) Calculate the mean of two numbers input by the user



Problem 04) Calculate the average of 5 numbers input by the user



Problem 05) Print all the even numbers upto 100


START
INPUT n
i=2
count=1
Print i
i=i+2
Print i
count=count+1
If count≤n
Yes
No
END