

Assignment 1

1. To check whether a number is even or odd:

Algorithm :→

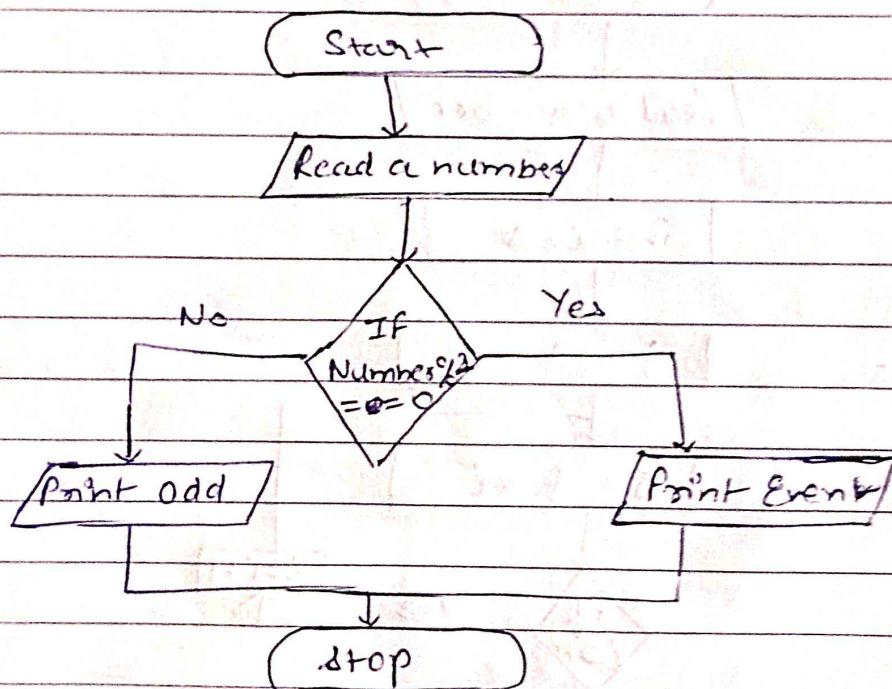
Step 1: Read a number

Step 2: If $n \% 2 == 0$

Step 3: If Yes then Print Even else odd

Step 4: Stop

flowchart



2. Java programme to find factorial of a number:

Algorithm :→

Step 1: Read a number whose factorial is to be calculated.

Step 2: Loop Let $j = 1$, Initialize

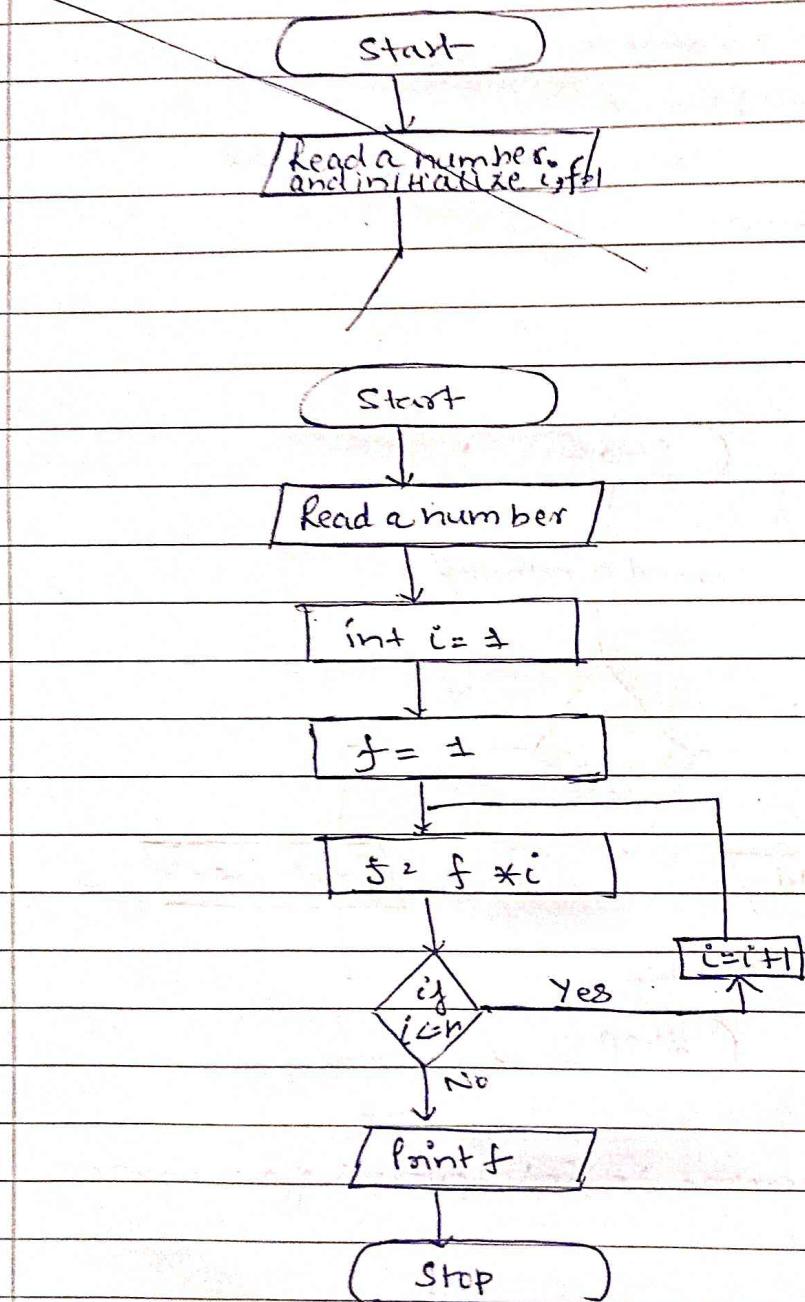
Step 3: Loop from ~~1 to n~~ to n .

Step 4: In the loop - $j = j * i$;

Step 5: Factorial is printed by printing j after the loop has executed

Step1: Stop

~~flowchart:~~



3

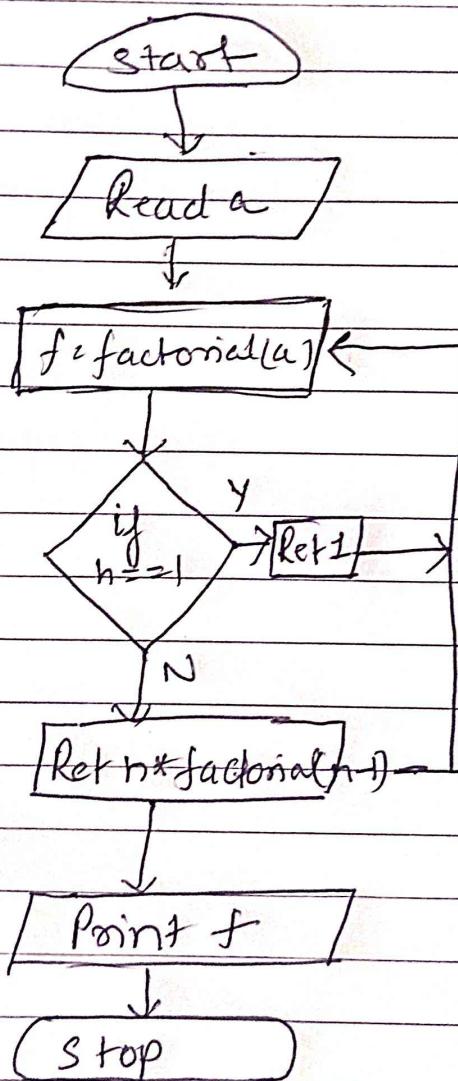
Factorial using recursion:

Step1: Read a number

Step2: Define a function or method factorial

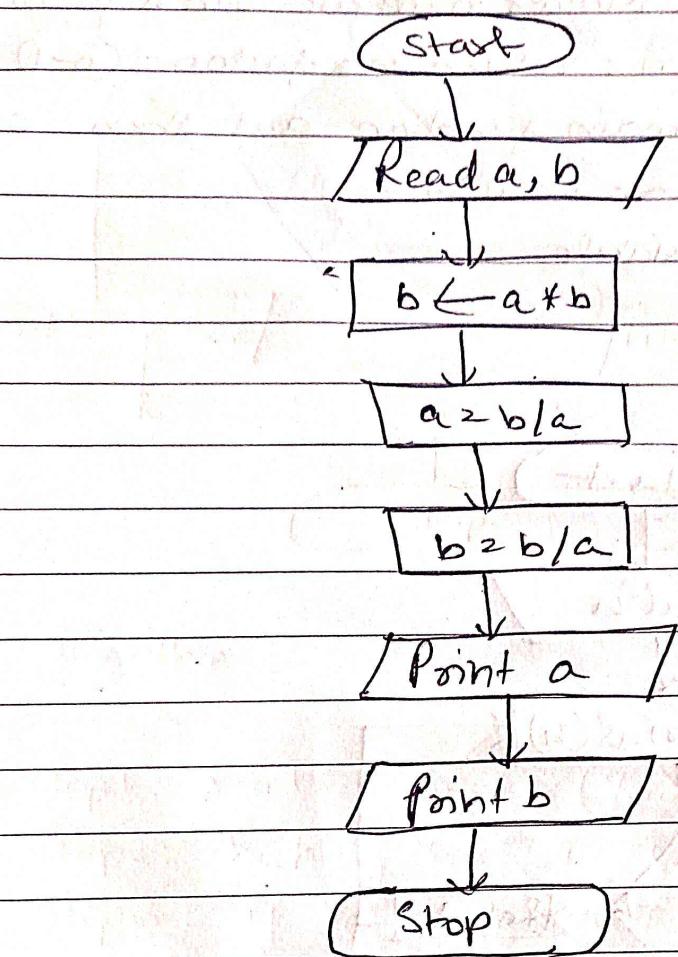
Step3: Give a base condition like $n=1$ / $n=0$ returns 1

- Step 4: Else return number multiple with a factorial value
one less than value n : return $n * \text{factorial}(n-1)$
- Step 5: Define the main function and then call the function store it in a variable
- Step 6: Print the variable
- Step 7: Exit.



4

Flowchart for swapping two numbers



Algorithm:

Step 1 : Read a, b

Step 2 : $b \leftarrow a * b$

Step 3 : $a \leftarrow a / b$

Step 4 : $b \leftarrow b / a$

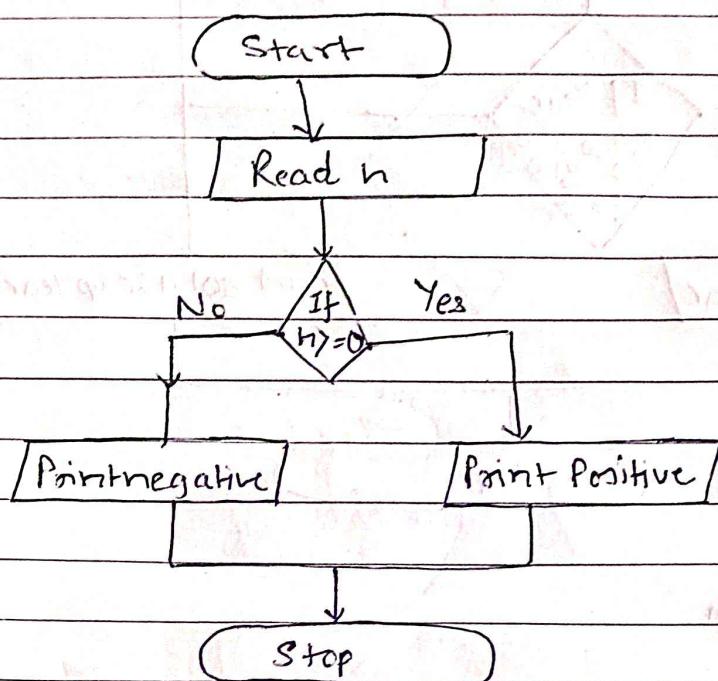
Step 5 : Point a

Step 6 : Point b

Step 7 : Exit

5. Flowchart to find whether the given number is positive or negative

-ve



Algorithm :

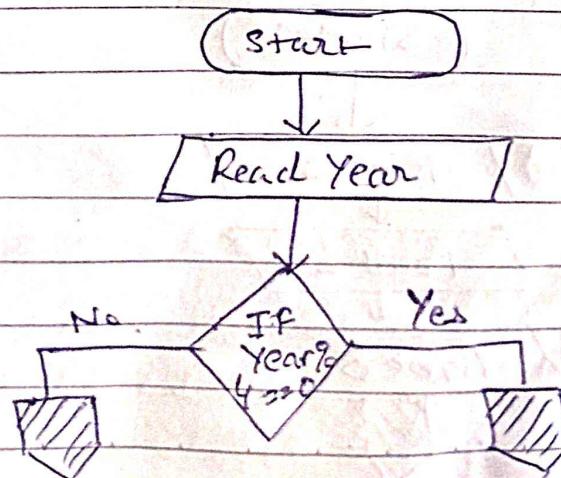
Step1: Read a number

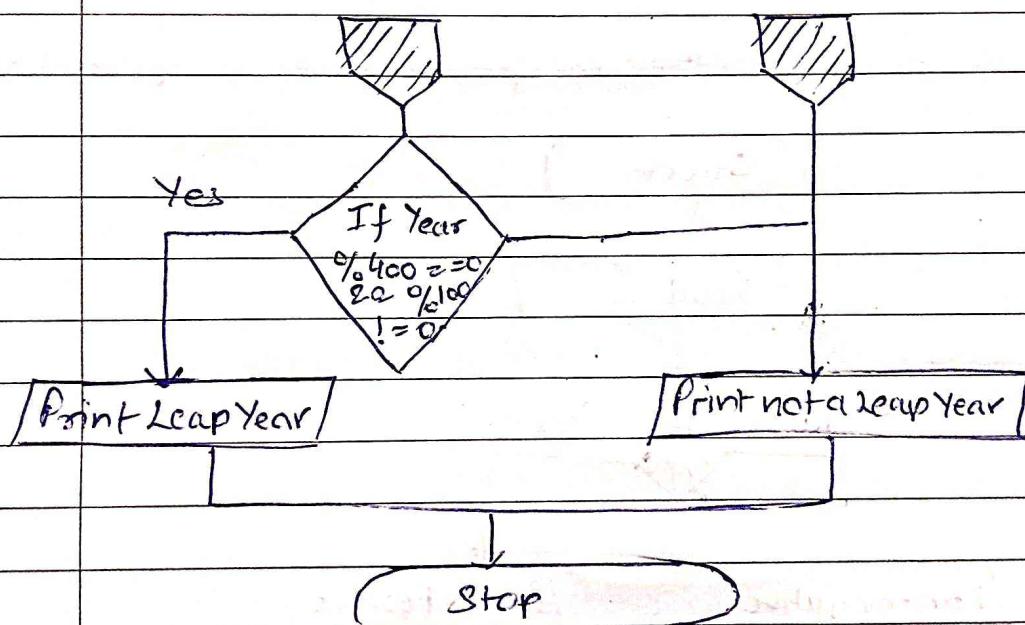
Step2: Check whether the number ≥ 0

Step3: If Yes then print positive else print negative

Step4: Stop

6 Flowchart to display whether the year is leap year or not.





Algorithm

Step1: Read a number (Year)

Step2: check if $\text{Year} \% 4 == 0$

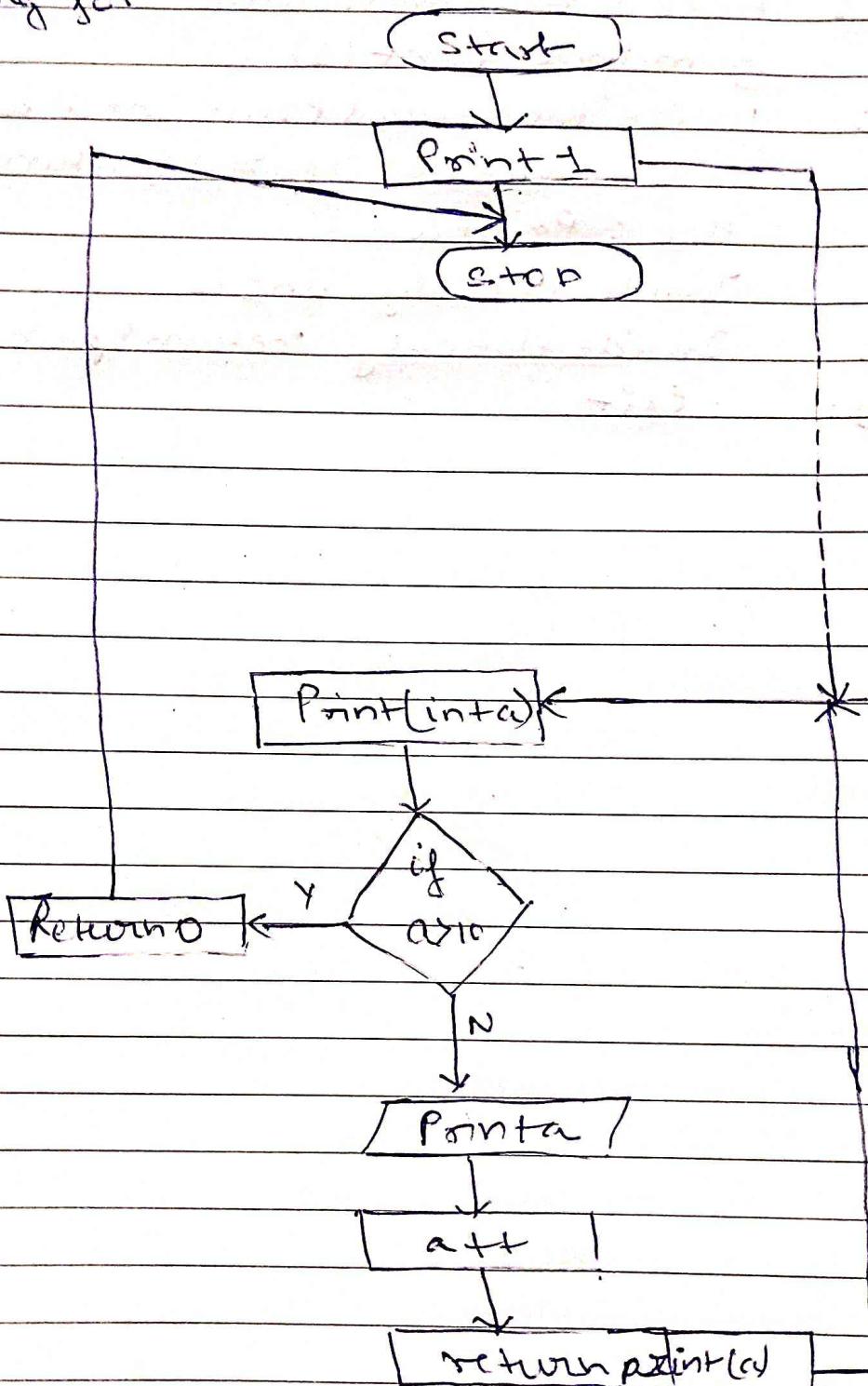
Step3 : If Yes check $\text{Year} \% 400 == 0$ || $\text{Year} \% 100 != 0$ else.
Print not a leap Year

Step4 : check Step3 condition If Yes Print leap Year else
not a leap Year

Step5: Stop

7

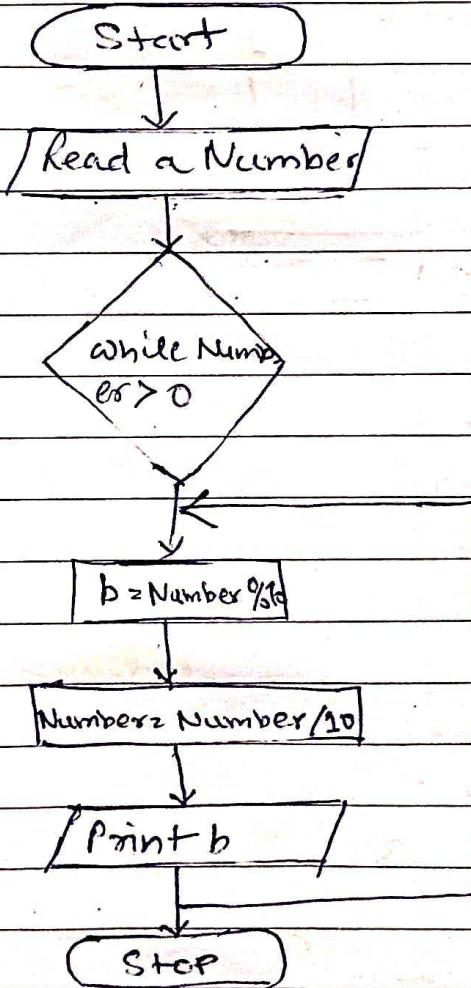
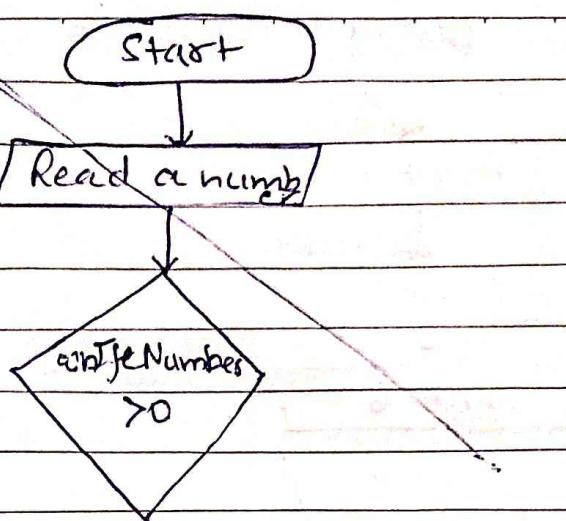
flowchart for printing 10 numbers from 1 to 10 without using for



8

flow chart for printing the digits of a number :

Page _____
HITAISHI
Date _____



Algorithm

Step 1 : Read a number

Step 2 : while number ≥ 0 run a while loop

Step 3 : initialize $b \leftarrow \text{number \% 10}$

Step 4 : $\text{number} \leftarrow \text{number} / 10$

Step 5 : Print b , ~~and go to step 2~~ and go to step 2

Algorithm

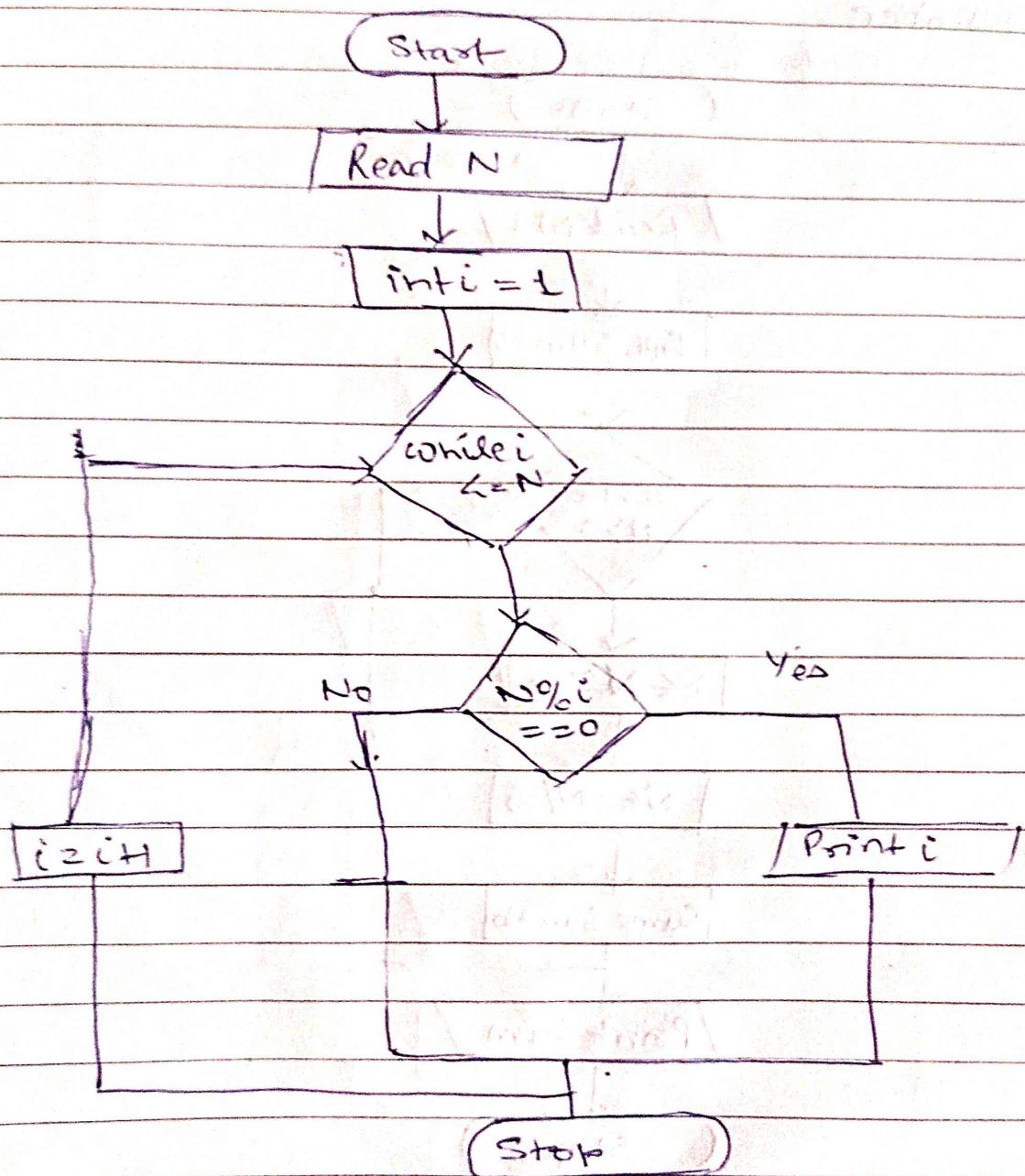
- Step 1: Make a main method and call a
 f method Print(1)
- Step 2: Inside the method Print(int c) write
 base condition if(c>0) {return n3}
- Step 3: else : Print a
- Step 4: Inside else only inc a
- Step 5: Inside else only return Print(a)
- Step 6: Exit.

Step 6 : Exit the ~~exit~~ while loop

Step 7 : Stop

9

Flowchart for Printing all the factors of the given number



Algorithm :

Step 1 : Read number N

Step 2 : initialize i = 1

Step 3 : while i value less than N run loop

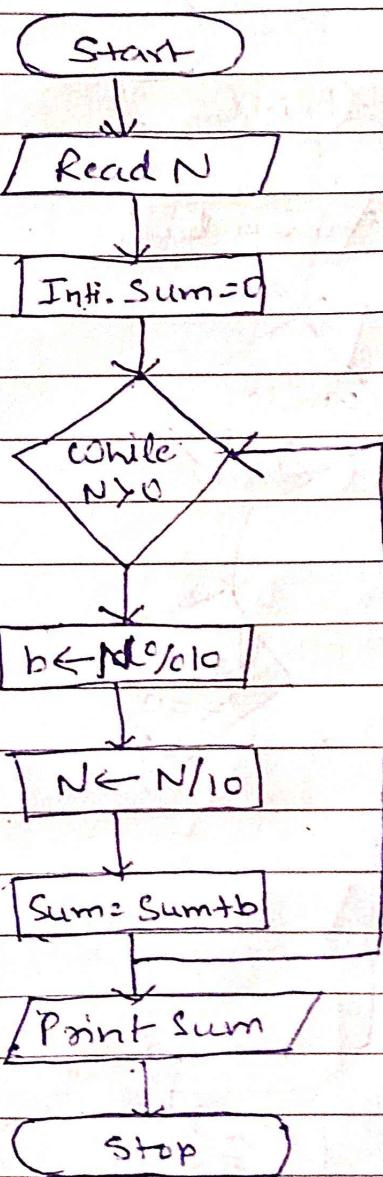
Step 4 : check $N \% i == 0$

Step 5 : If yes point i else nothing

Step 6: increment i, Return to Step 4
 Step 7: Exit.

10

Flowchart to print the sum of digits of a given number



Algorithm

Step 1: Read a Number

Step 2: Initialize variable sum = 0

Step 3: Run a loop until n>0

Step 4: Store b <- N%10

Step 5: Store $N \leftarrow N/10$

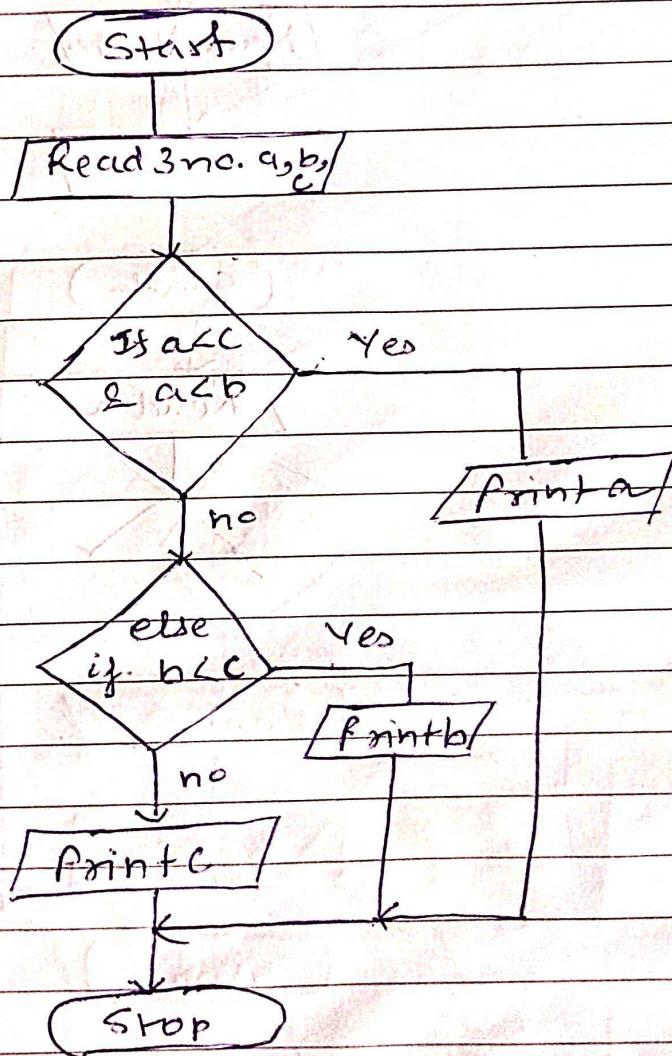
Step 6: Add $\text{Sum} + b$ and store in $\text{Sum} \leftarrow \text{Sum} + b$

Step 7: Go to Step 3 and check condition

Step 8: If out of loop print sum

Step 9: Exit.

II Flowchart for finding ^{smallest} ~~different~~ among 3 numbers



Algorithm

Step 1: Input three numbers a, b, c

Step 2: compare a with b and c : $a \leq b \leq c$

Step 3: If true print a

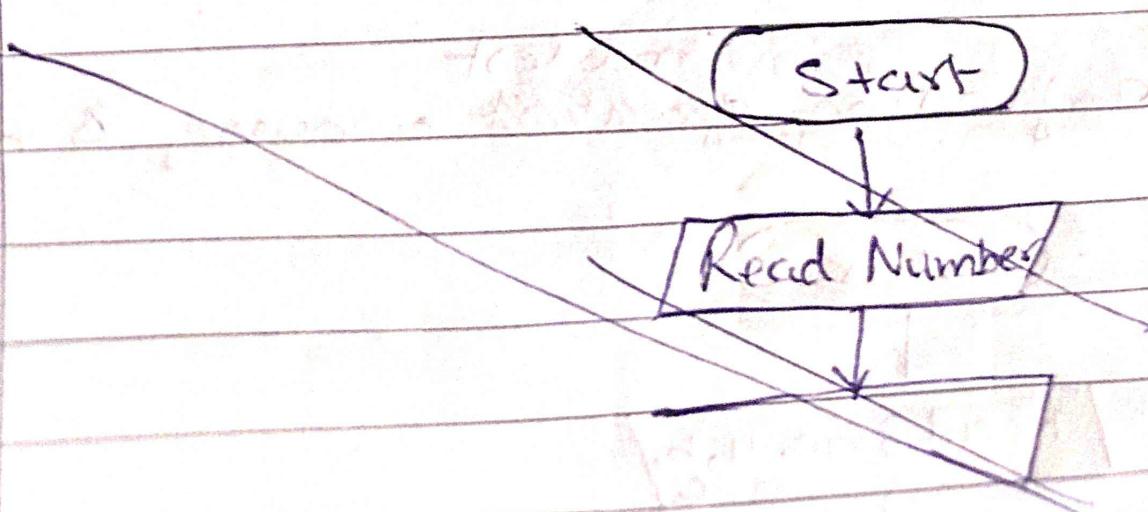
Step 4: else compare b and c : bcc

Step 5: If true Print b

Step 6: Print c if false (step 5)

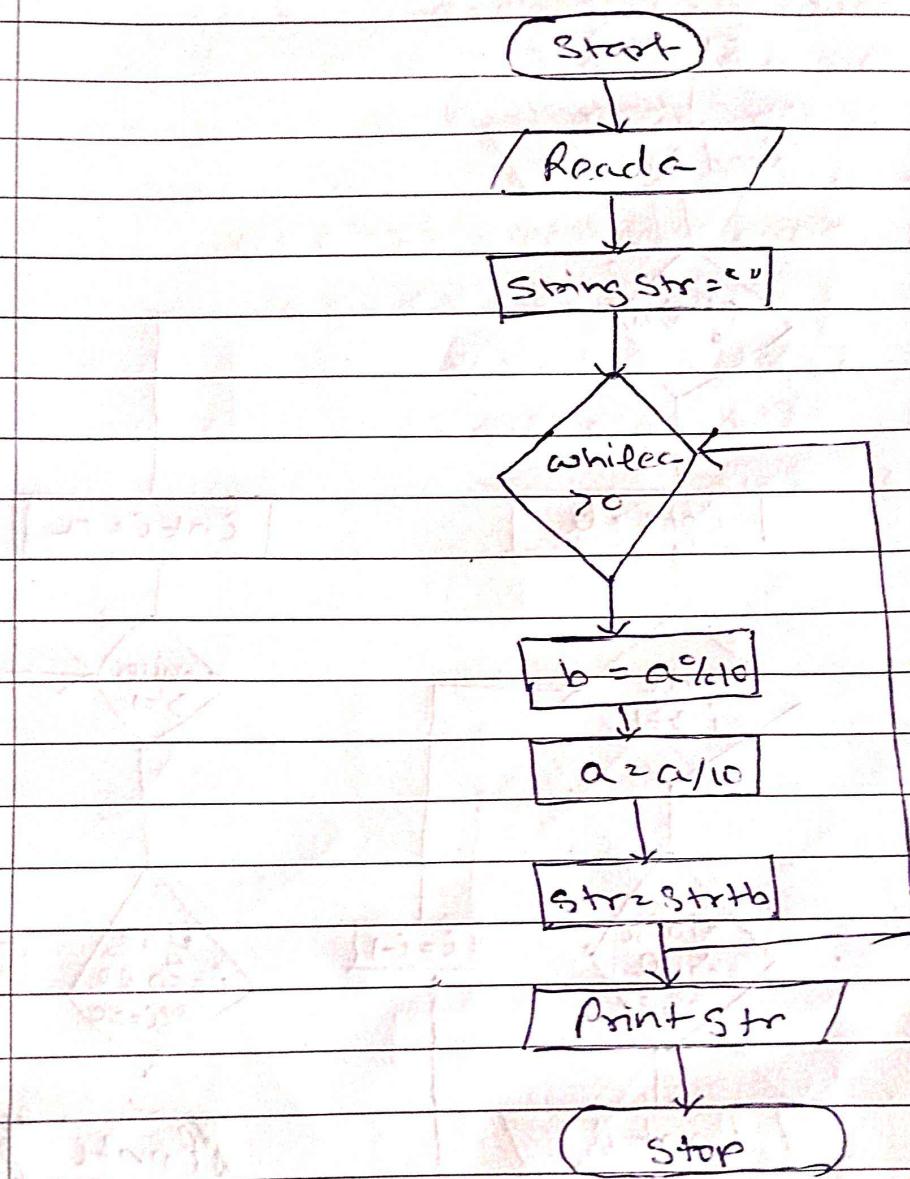
Step 7: Exit

flowchart for reversing a number



13

Flowchart for reversing a number



Algorithm:

Step 1: Read a number a

Step 2: initialize a string str empty

Step 3: Run a loop until a>0

Step 4: Store b $\leftarrow a \% 10$ Step 5: Store a $\leftarrow a / 10$ Step 6: Store str $\leftarrow str + b$

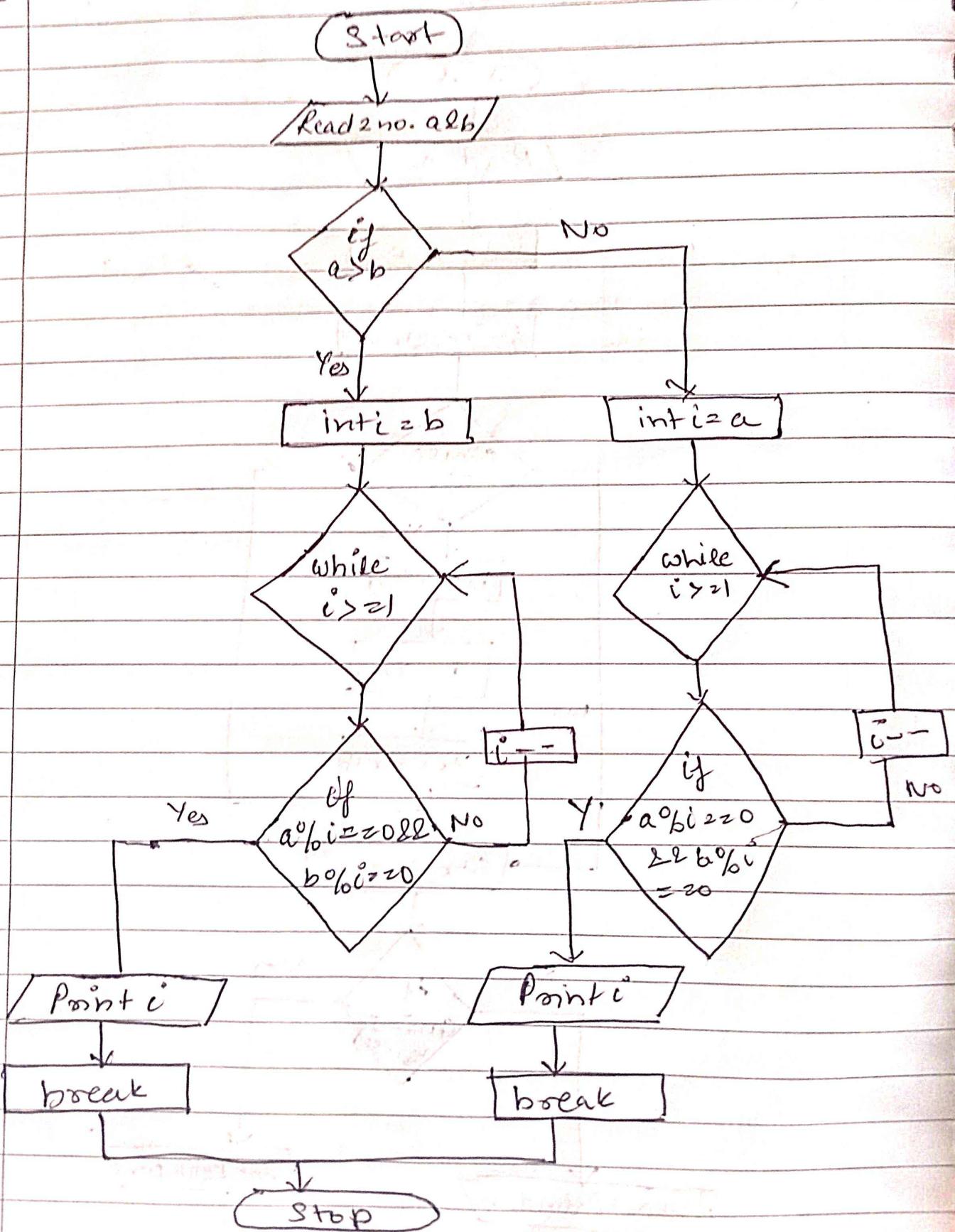
Step 7: Go to Step 3 and check the condition

Step 8: Print str

Step 9: Exit

14.

Flowchart for GCD of two numbers



Algorithm:

Step1: Read two numbers a, b

Step2: if $a > b$ then interchange $i = a$

Step3: otherwise $i = b$

Step4: Run a loop while $i \geq 1$

Step5: if $a \% i == 0$ & $b \% i == 0$

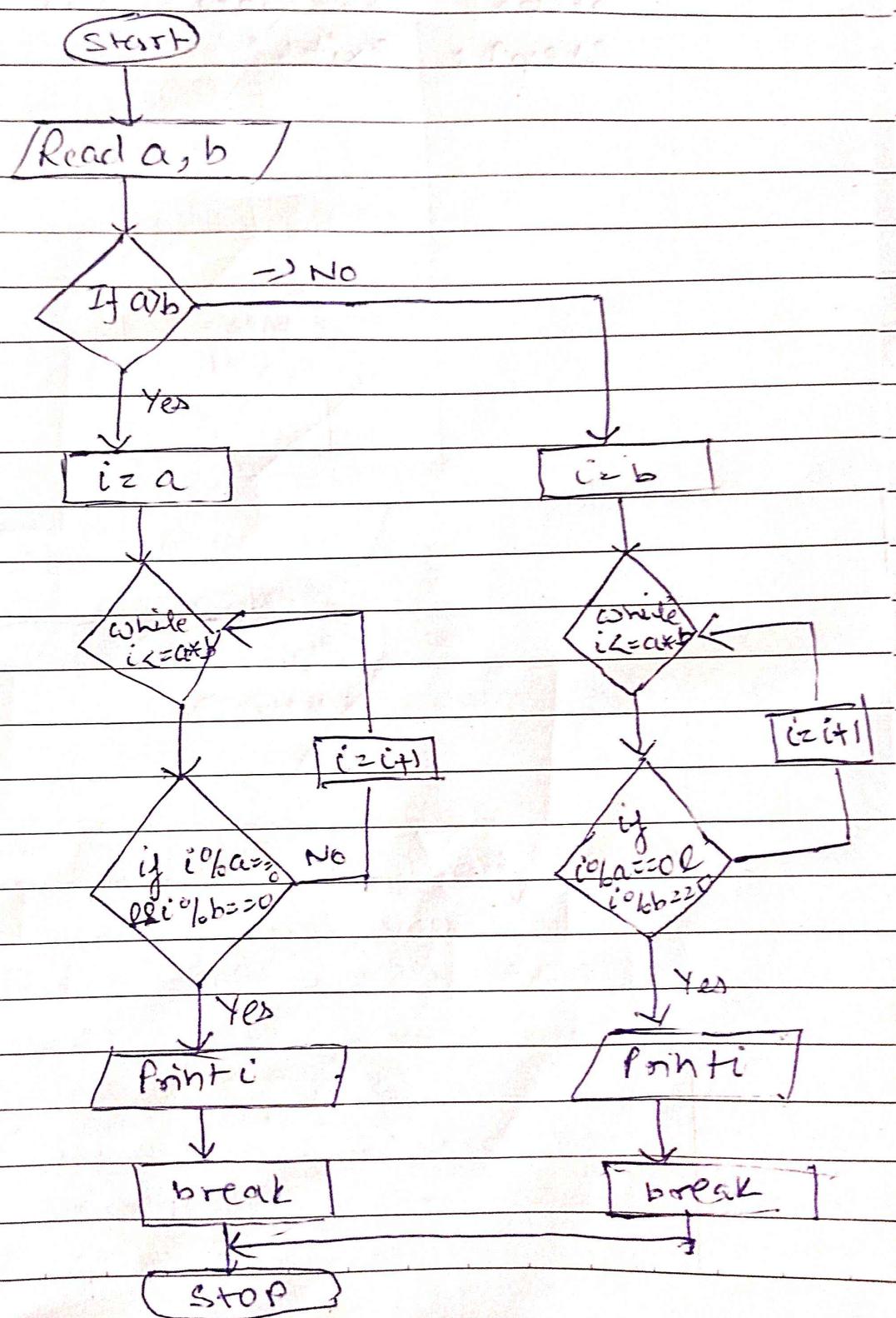
Step6: Print i

Step7: if Break

Step 9: Go to step 5 and check the condition

Step 10: Exit.

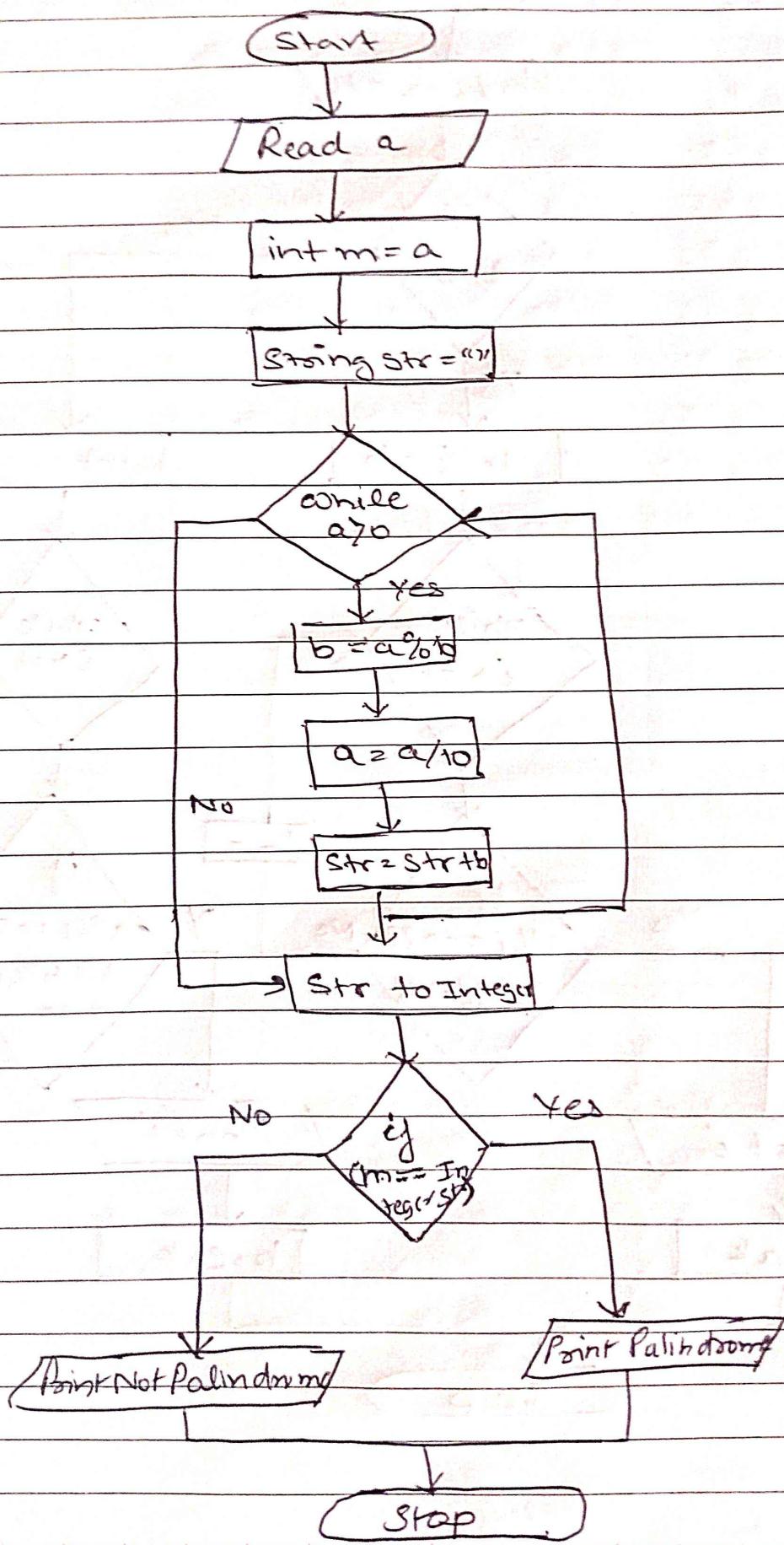
15 Flowchart for finding the LCM of two given numbers



Algorithm

- Step 1: Read two numbers a & b
- Step 2: compare a & b
- Step 3: if $a > b$ $i \leftarrow a$
- Step 4: else $i \leftarrow b$
- Step 5: Run a loop while $i \neq a * b$
- Step 6: if $i \% a == 0$ & $i \% b == 0$
- Step 7: Print i and break
- Step 8: else Go to step 5, before that inc. $i = i + 1$
- Step 9: Exit.

17. flowchart to print whether a number is palindrome or not.



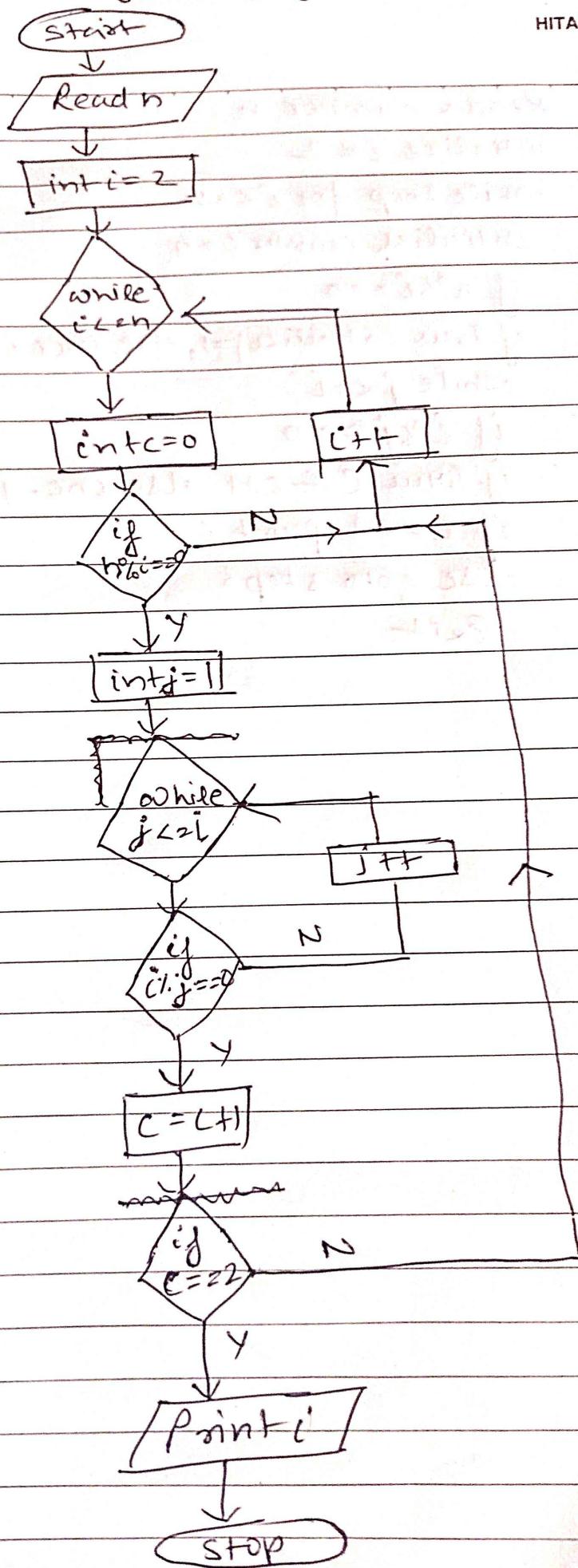
Algorithm :

- Step1: Read a number a
- Step2: Store that number in another var m
- Step3: Start a loop until a > 0 String Str \leftarrow ""
- Step4: Start a loop until a ≥ 0
- Step5: Store b $\leftarrow a \% 10$
- Step6: Store a $\leftarrow a / 10$
- Step7: Store in empty string Str = Str + b
- Step8: Goto Step4 check the cond.
- Step9: convert Str into integer
- Step10: compare Str with m
- Step11: If True print palindrome else not
- Step12: exit

Flowchart for printing prime factors of a number

18

HITAISHI Page _____
Date _____

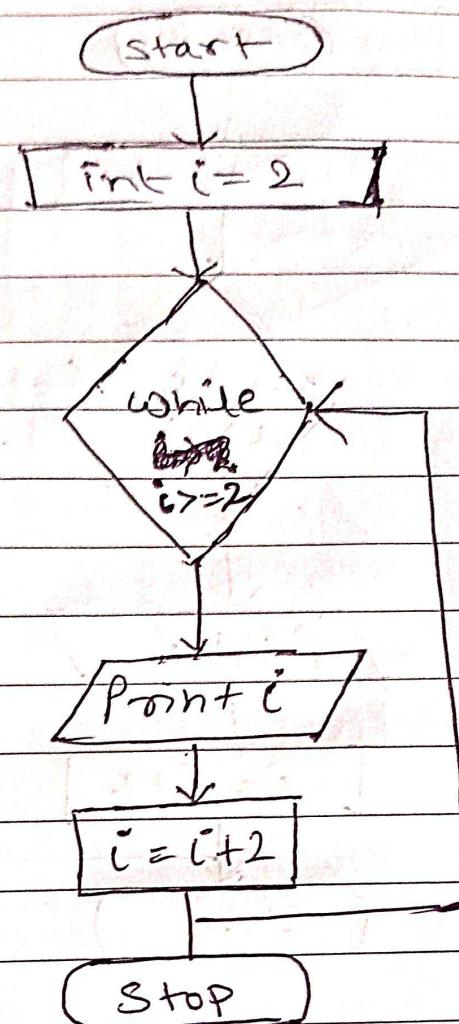


Algorithm:

- Step 1: Read a number n
- Step 2: Initialize $i = 2$
- Step 3: while loop for $i \leq n$
- Step 4: Initialize counter $c = 0$
- Step 5: if $n \% i == 0$
- Step 6: if True initialize $j = 1$, else incr i and goto step 3
- Step 7: while $j \leq i$
- Step 8: if $i \% j == 0$
- Step 9: if True $c \leftarrow c + 1$ else inc. j goto step 7
- Step 10: if $c == 2$ print i
- Step 11: else goto step 3
- Step 12: Exit

19.

Flowchart for printing an Even number series



Algorithm:

Step 1: initialize $i=2$

Step 2: Run a while loop for $i >= 2$

Step 3: Point i

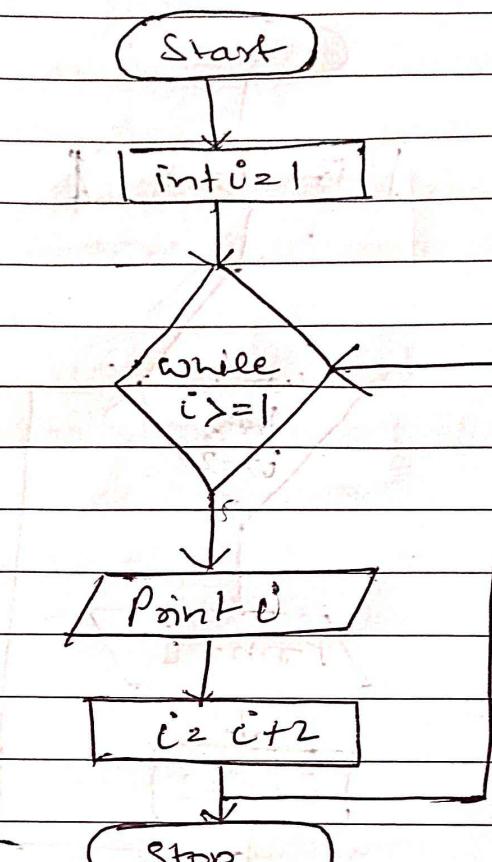
Step 4: Increment i by 1 ($i = i + 2$)

Step 5: Check the condition

Step 6: Stop

20

flowchart for printing an odd series



Algorithm

- Step1 : initialize int i=1
- Step2 : Run a while loop for $i \geq 1$
- Step3 : Print i
- Step4 : increment $i \leftarrow i + 2$
- Step5 : check the condition
- Step6 : Exit