(1) Generation of numbers series 1, 2, 3, 4, --- n.

Algorithm:-

Step1 :- Start

Step2 :- Give i=0, j=1, k=0 Fib=0

Step3 :- If i<n ? then.

Step4 :- Yes output

Fibonacci(j) = Fib

if No = end.

Step5 :- if satisfiy i<n ?

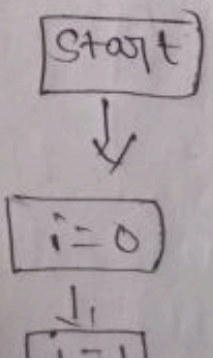
Give Fib = j+k

j=k, k=Fib

Step 6 :- Give i = i+1

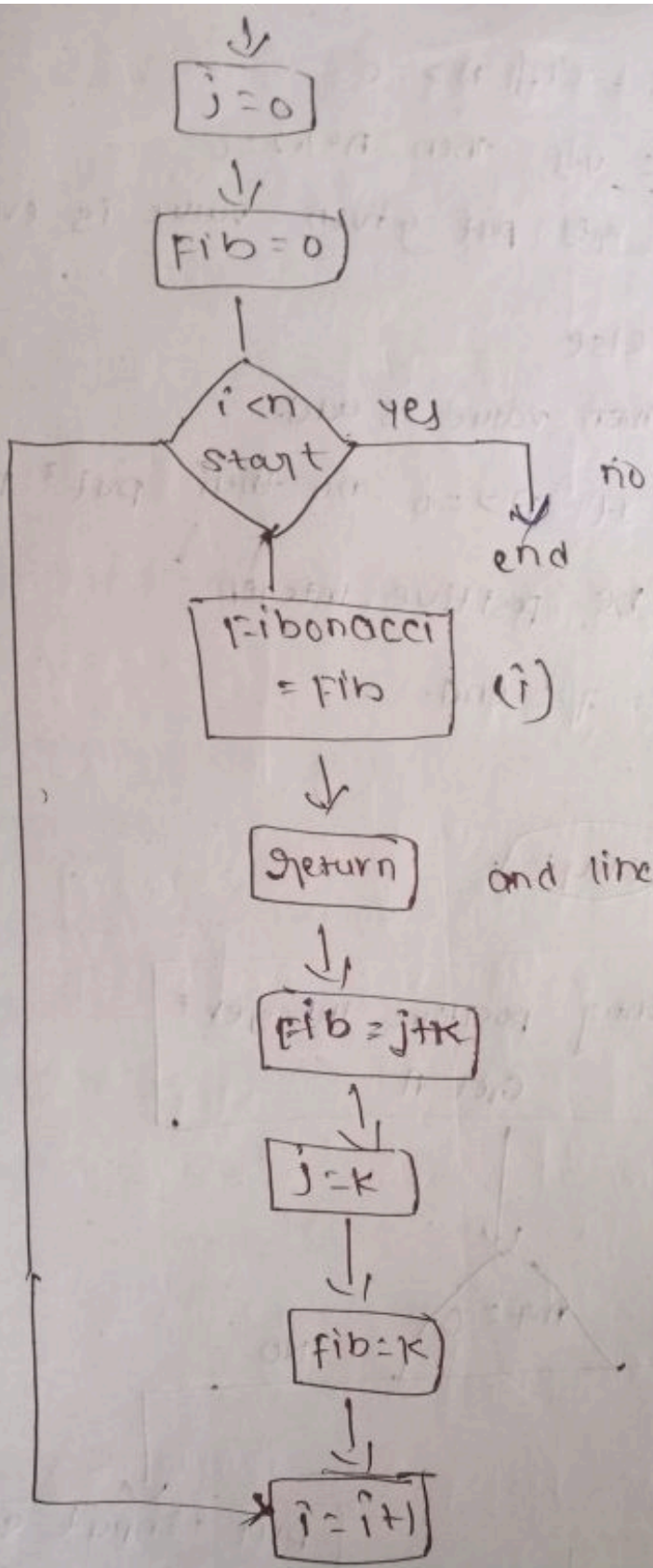
Step7 :- Stop.

Flow Chart



(2) Alg

1, 2, 3.



F ib=0

ib

(2) Algorithm :-

Step: 1:- Start

Step: 2:- "Enter positive integer"

Get n

Step 3 :- If A>=0

Step 4 :- yes then A%2=0

if its yes put given value is even

Step 5:- else

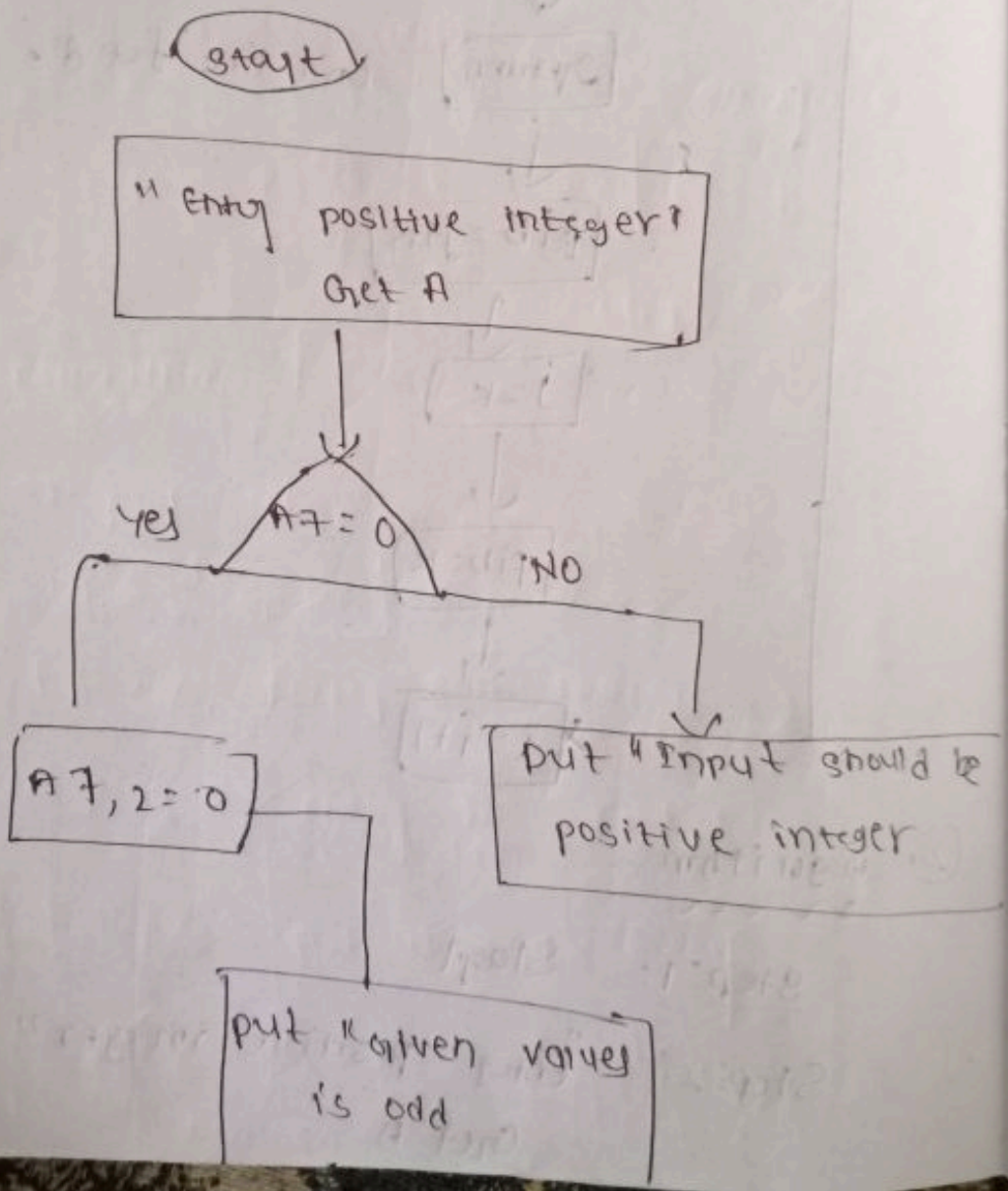
given value is odd.

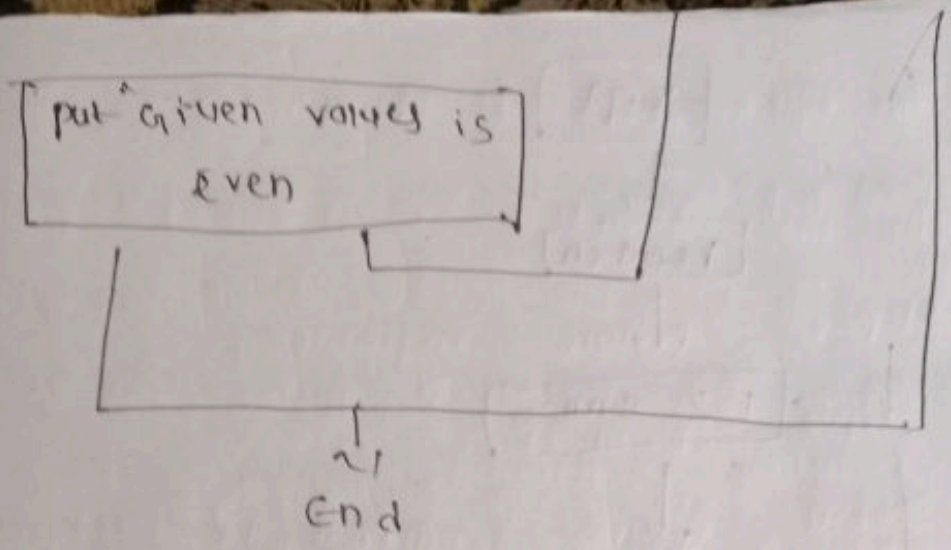
Step:- 6 :- It A>=0 no then put " Input

→ should be positive integer

step :- 7 :- end.

Flow chart:-





(3) Generation of ODD number series 1, 3, 5. - n.

(4) Summing up series 1+2+3+4 - - - n:

Algorithm:

step 1 - start

step 2 :- read (n)

step 3 :- Given i=2 sum=0

step 4 :- i=2, sum=0

step 5 = then i<=n yes

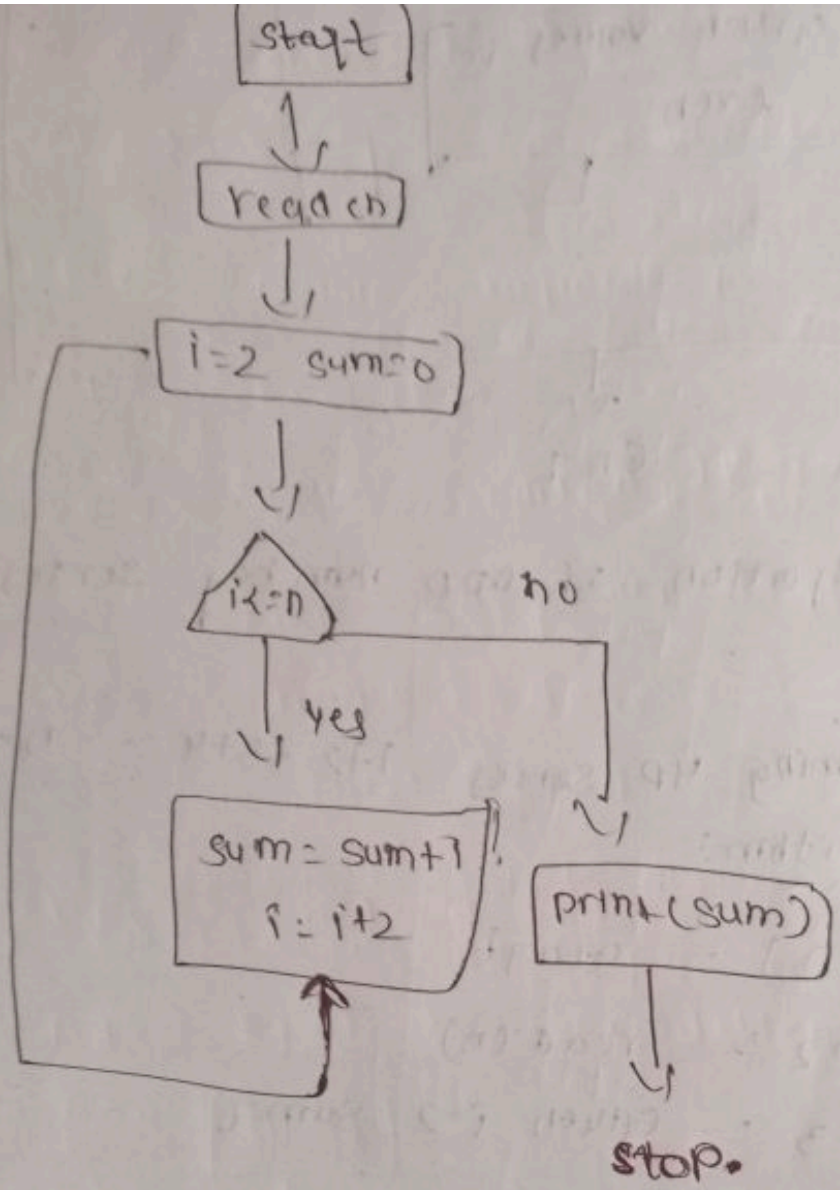
sum = sum + i

i = i+2

Step 6 :- No

print (sum)

Step 7 - Stop.



⑤ Algorithm:-

Step:1 - Start

Step 2 :- Given Sum ← 0

Step 3 :- "Enter even number ← 2

Step: 4 :- Enter N:" Get N

Step 5 :- put N = +N

Step 6 :- given the condition even number

Step 7:- If it is use

put " sum of even number from 1 to N :

u sum.

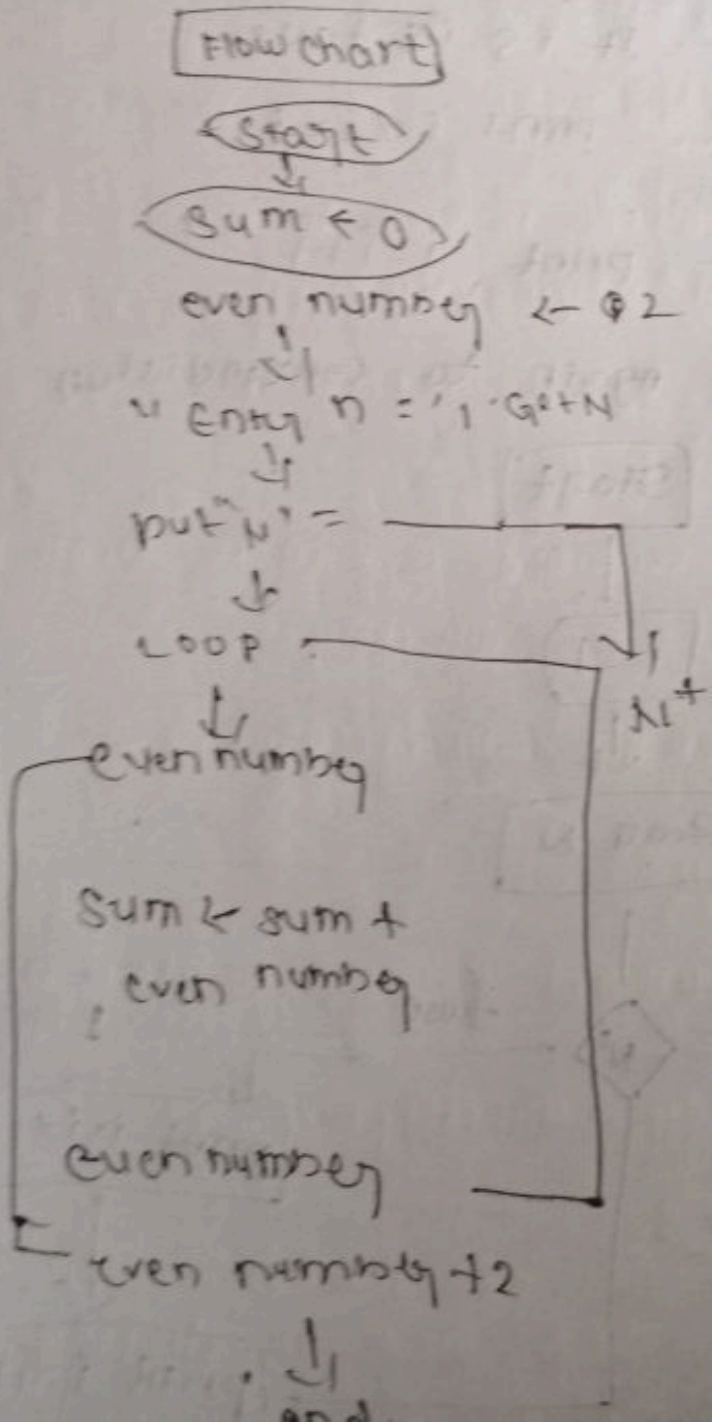
Step 8:- It is an not sum ← sum +

even number.

Step 9:- even number ←

even number 2

Step 10:- end.



6. Sum up Cubes of N Number

Step 1 :- Start

Step 2 :- Declare variable = 1

Step 3 :- Read N

Step 4 :- Given Condition If i > N

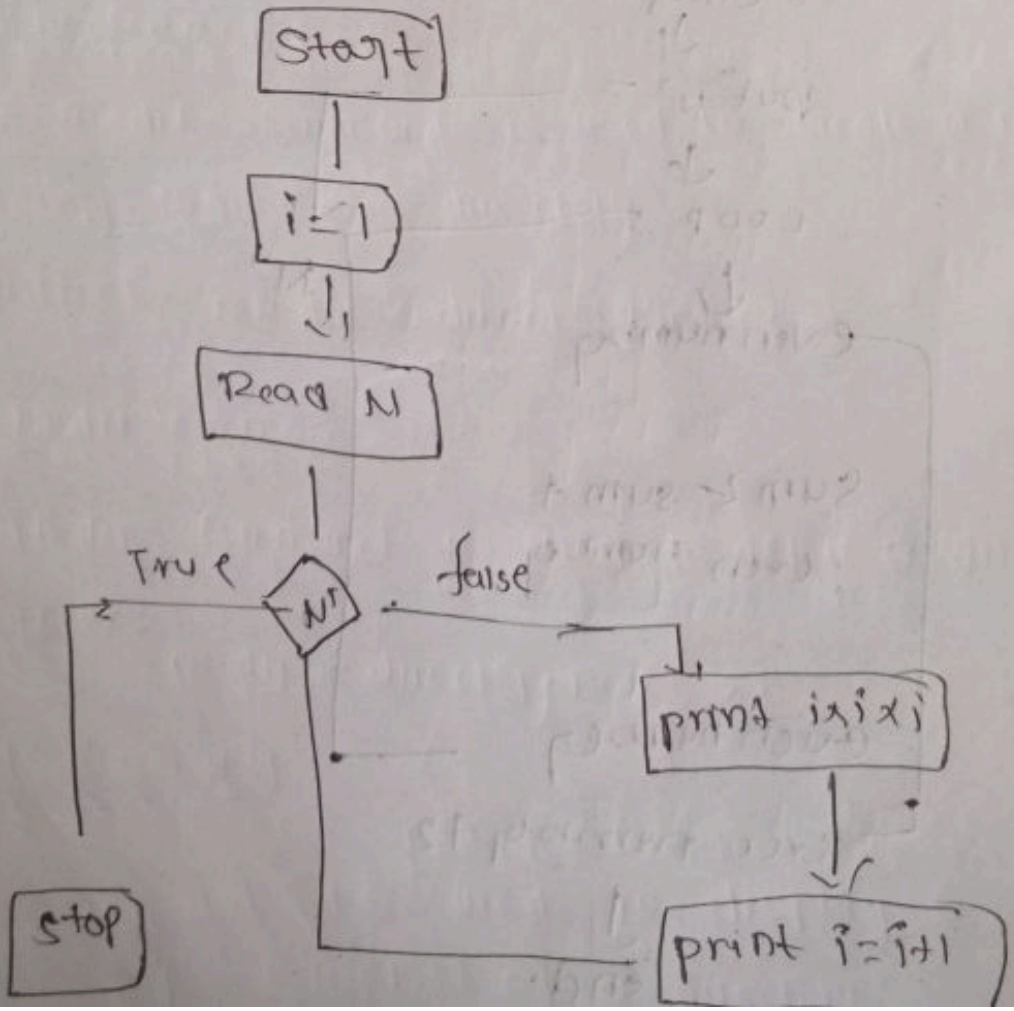
true → Stop

Step 5 :- If is is false

print i x i x i

Step 6 :- print i = i + 1

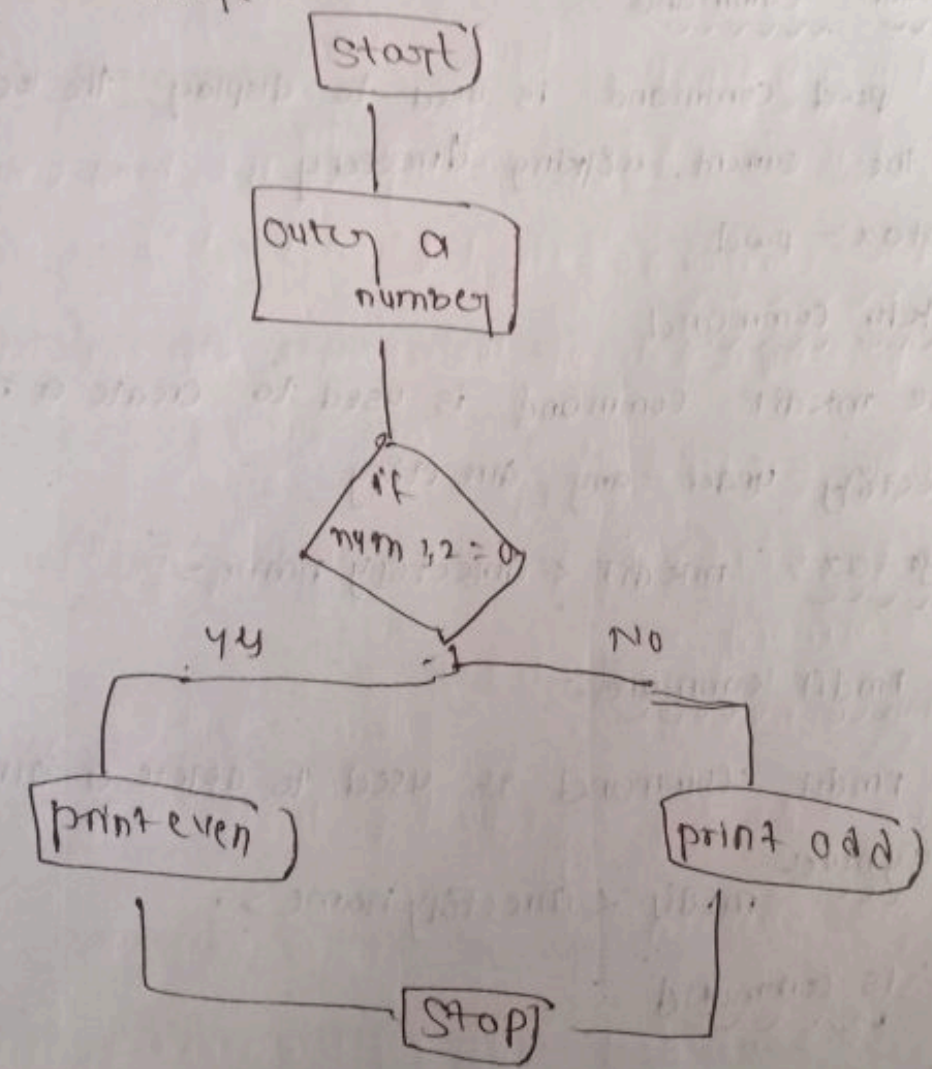
Step 7 :- Again to concondition



(8) Finding the whether given integer its odd & even

Step 1 :- Start
Step 2 :- Read number 1
Step 3 :- remainder = number % 2
Step 4 :- remainder == 0
Step 5 :- write "even number"
else
Step 6 :- write odd number.

Flow chart



(10) Strong & not

Algorithm:-

Step :- start

Step 2 :- " base input.

Give design sum = 0

Step 3 :- Read the number of, num

Step 4 :- whether the temp=0

then

Compute digit = temp % 10

temp = temp / 10

sum = sum + (digit + 3)

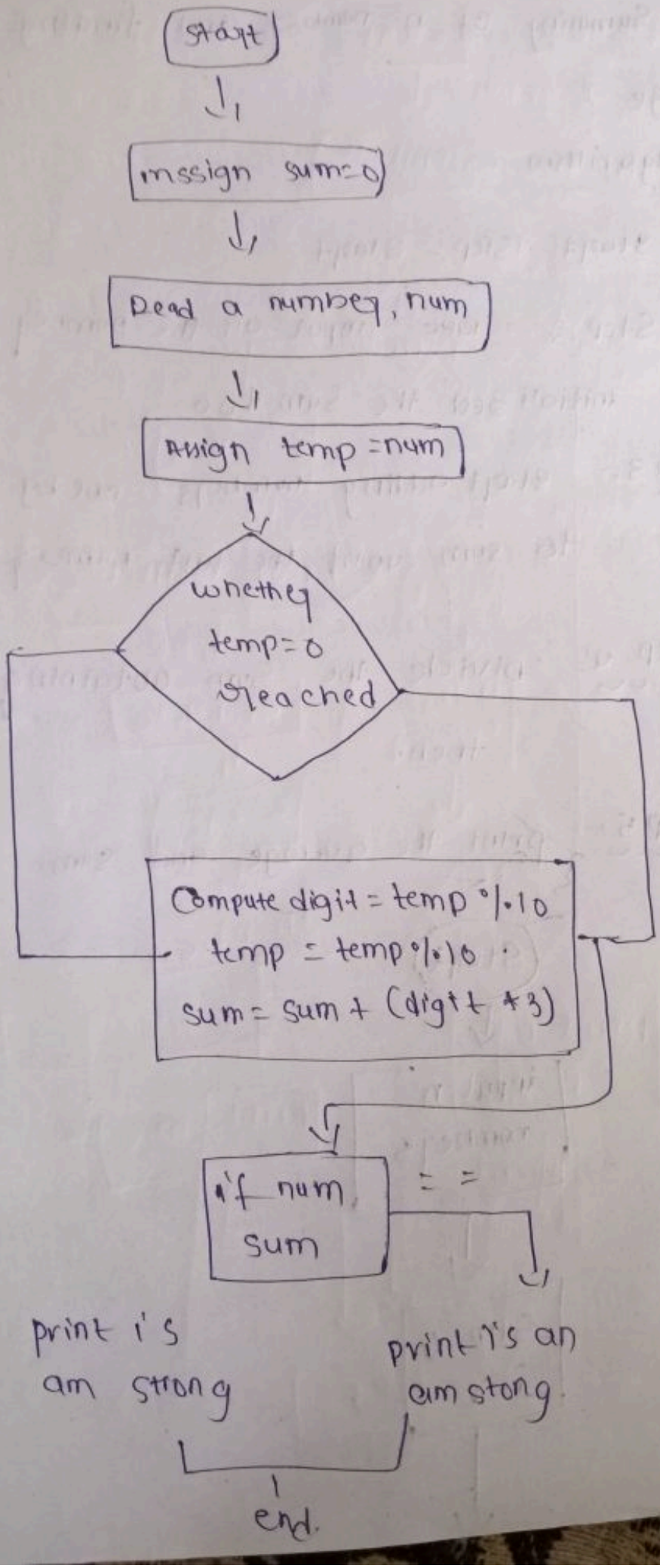
Step 6 :- if num == sum

print it is am strong

else;

not am strong

Step 7 :- End.


start
mssign sum=0
Read a number, num
Assign temp = num
whether temp=0 reached
Compute digit = temp %.10
temp = temp %.10
sum = sum + (digit * 3)
if num sum
= =
print i's am strong
print i's an am stong
end.

(12) Summing of n numbers and finding average ?

Algorithm

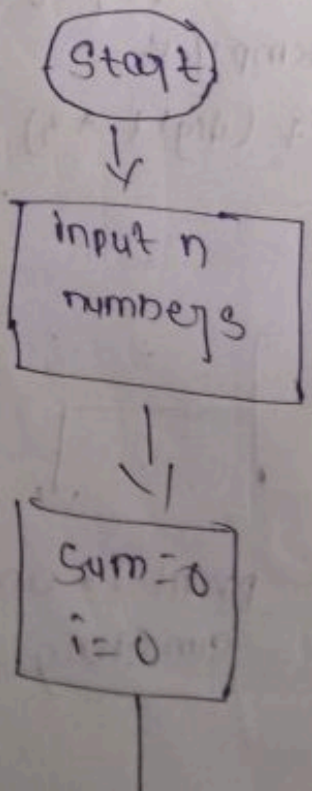
Start : Step :- Start

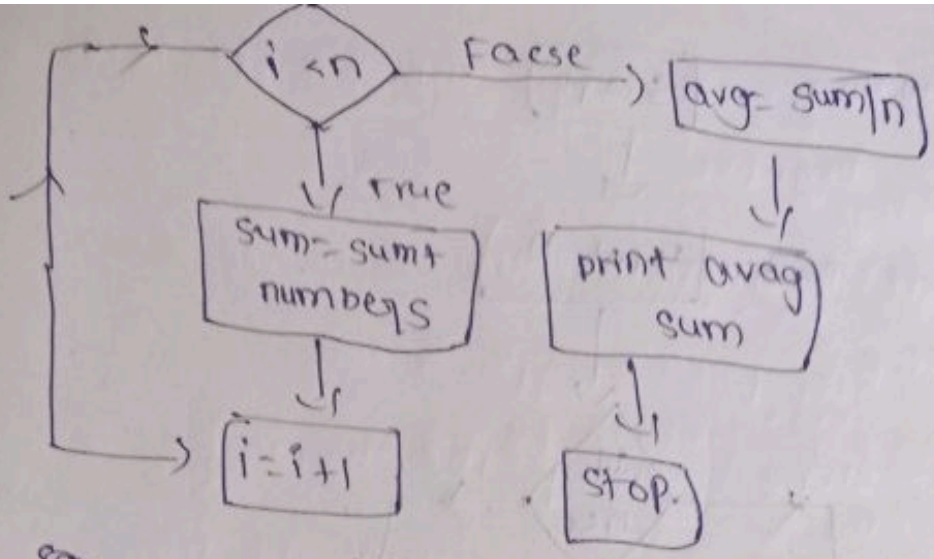
Step :2 :- Take input all the numbers and initialized the Sum to 0

Step 3 :- start adding numbers one by one to sum until the last number.

Step 4 :- Divide the Sum obtaining by then.

Step 5 :- print the average and sum.





(13)

Step 1 :- Start

Step 2 : Input n

Step 3 :- Give condition if n<0 yes then print not possible.

Step 4 :- If it is no, initialize

S=0 m=1

repeat until n=0

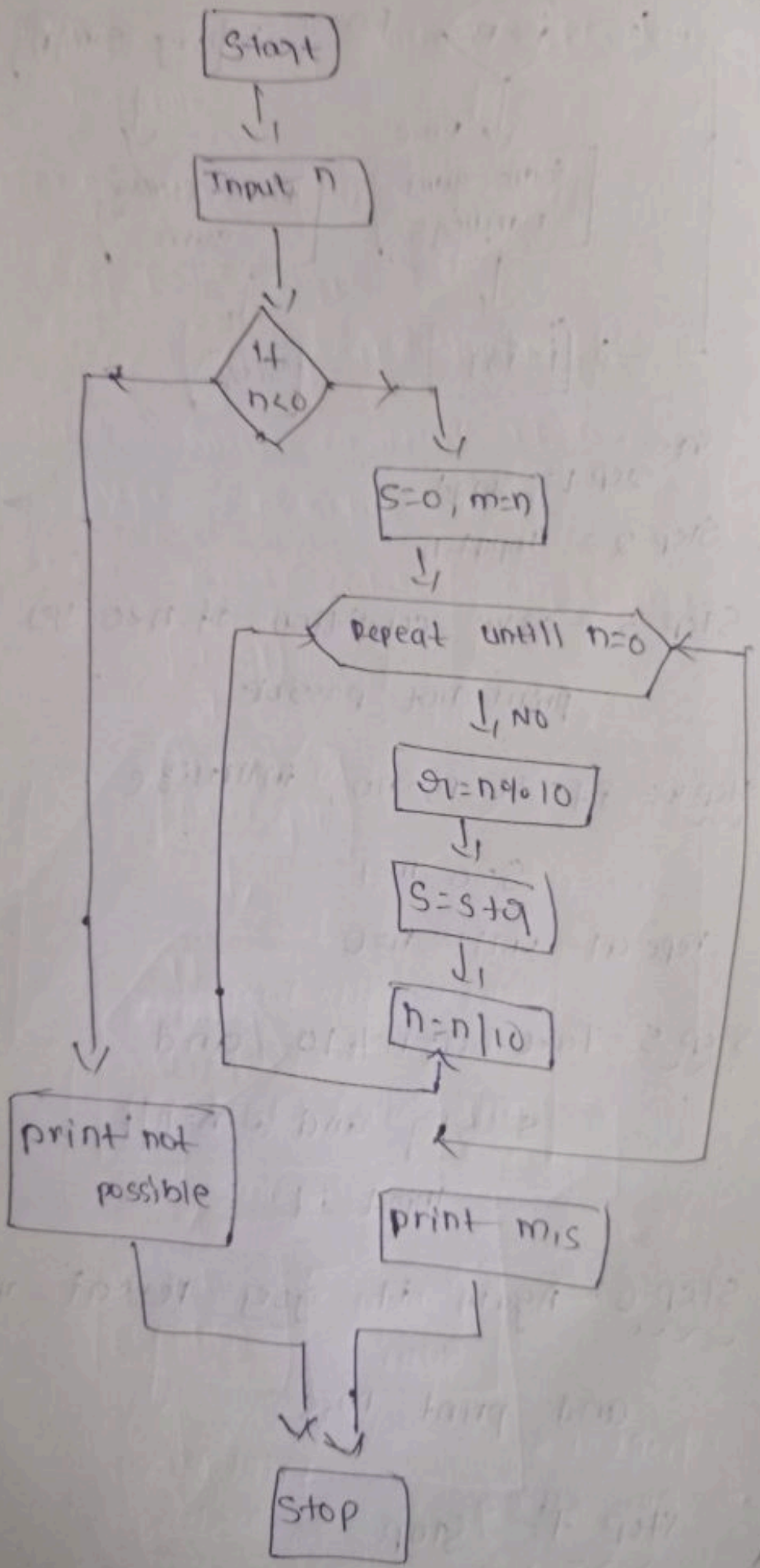
Step 5 :- Take q=n%.10 and

S = S+q and n=n/p

Input it

Step 6 :- Again it goes repeat until 10 and print m,s.

Step 7 :- Stop


Start
Input n
If n<0
S=0, m=n
Repeat untill n=0
NO
r=n%10
S=S+r
n=n/10
print not possible
print m,s
Stop

(iv) Printing the digits of an integer number

Step 1:- start

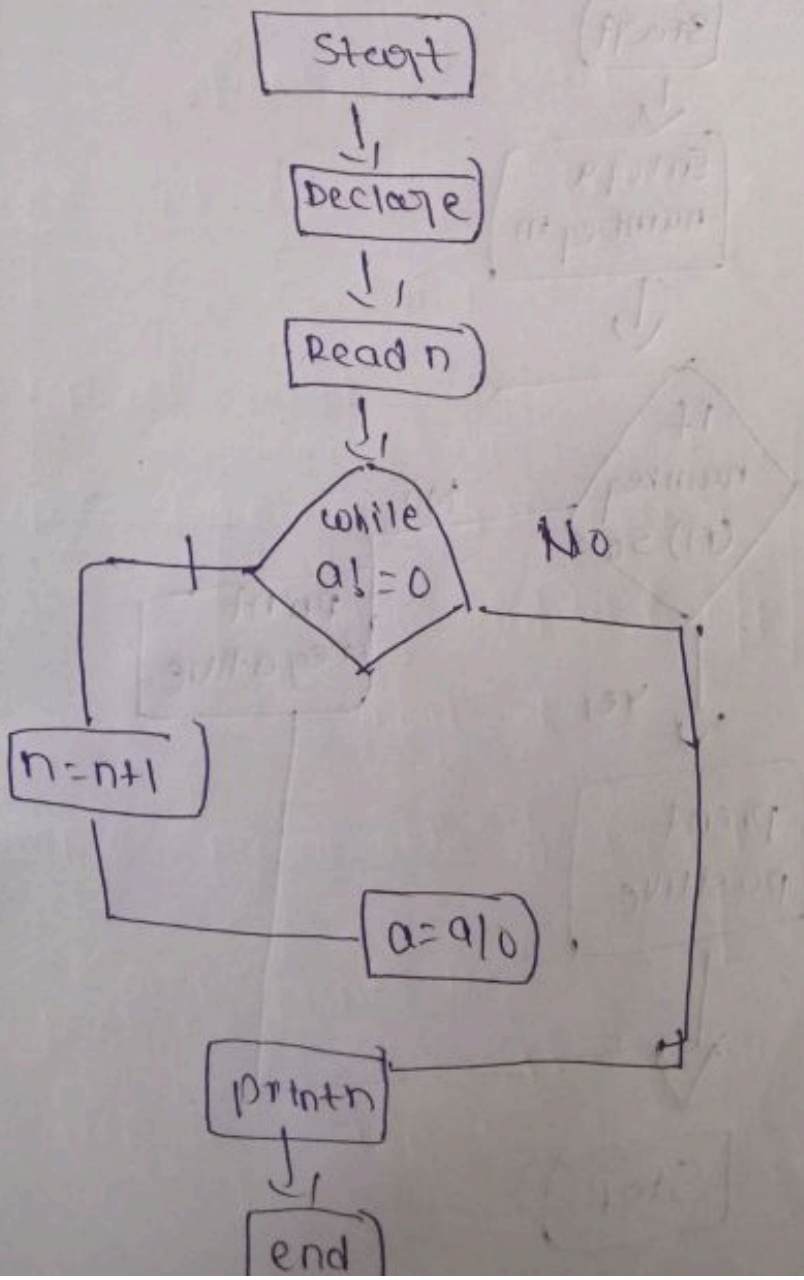
Step 2:- Declay variable a and n
and Read a

Step 3:- Given the condition while (a!=0)

Step4:- if it is then a=a/10 and it is reverse to n=n+1

Step 5:- if it is no print n.

Step 6:- End.



(15) Finding whether given is positive & negative.

Algorithm:-

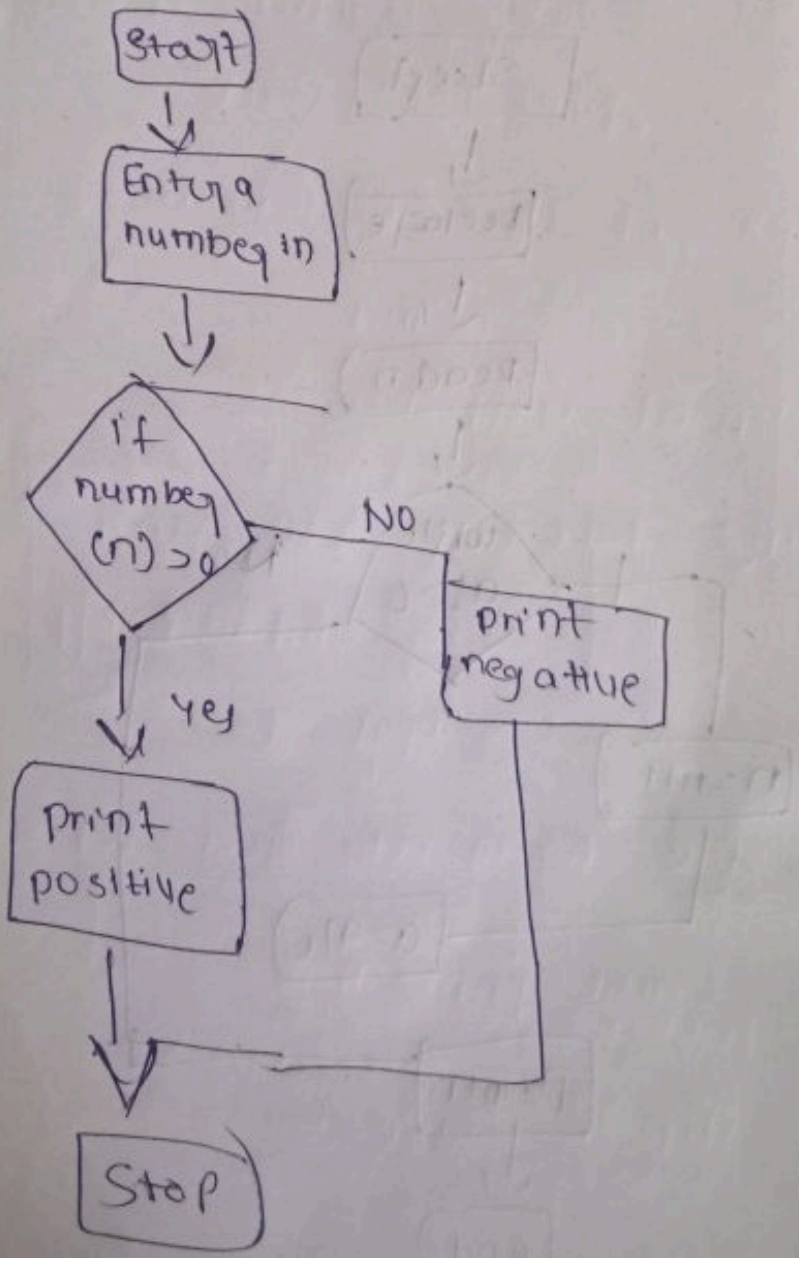
Step 1:- Start

Step 2:- Enter the number

Step 3:- Given condition if n>0

yes → possitive.

Step 4:- NO → negative Step 5:- End.



(16) Swapp
variab
Step 1
Step 2:
Step 3:
Step 4
Step 5

(16) Swapping on numbers with a temporary variable

Algorithm

Step 1 :- Start

Step 2 :- Declare variable a, b, and c.

Step 3 :- Read a and b

Step 4 :- Give c=a, a=b, b=c

Step 5 :- End