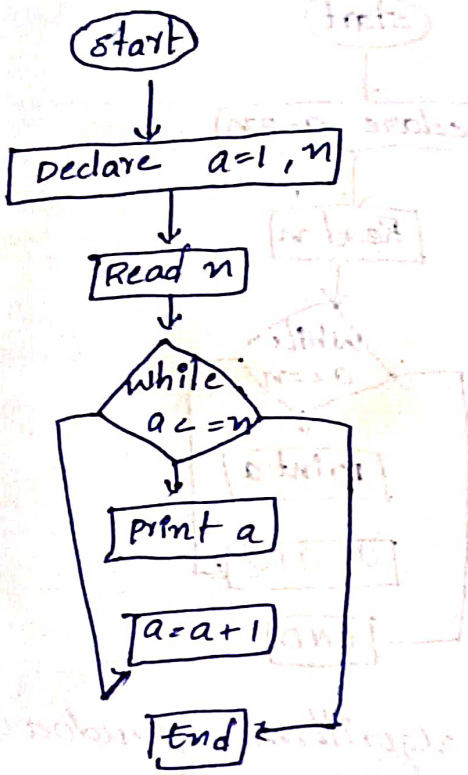
EXP.NO.1 Generation of number series.

Flow chart :-



Aim :- To write algorithm, pseudocode, Flowchart for the Generation of the number series.

Algorithm :- step 1 = Initialize a with 1
step 2 = Read n
step 3 = Check a is less than or equal to n
step 4 = print a
step 5 - Increment with 1.

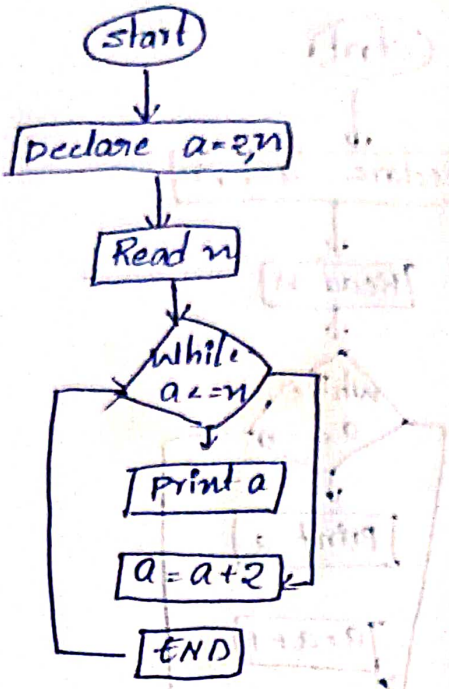
Pseudocode :- Declare a-1, n
Read n
while a<=n
print a
a=a+1
end while.

Result :- Successfully written a algorithm, Pseudocode, Flowchart for generation of number series

(1, 2, 3, 4, 5, ..., n)

EXP NO-2 Generation of even number series

Flow chart.



Aim :- To write a algorithm, pseudocode, Flowchart for generation of even numbers.

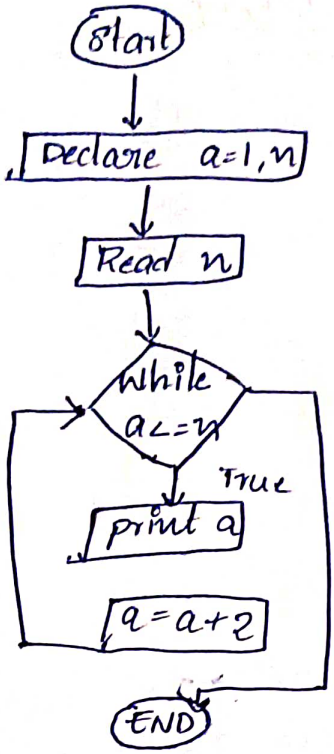
Algorithm :- step 1 - Declare a=2,n
step 2 - Read n
step 3 - while check a is less or equal to n
step 4 - print a
step 5 - increment a with 2.

Pseudocode :- Declare a=2n
Read n
While a<=n
Print a
a=a+2
While end

Result :- successfully written algorithm, Flow chart, Pseudocode for Generation of even number series.

EXP-NO-3 Generation of odd series

Flowchart -



Aim:- To write flow chart, algorithm, Pseudocode, for Generation of odd number series.

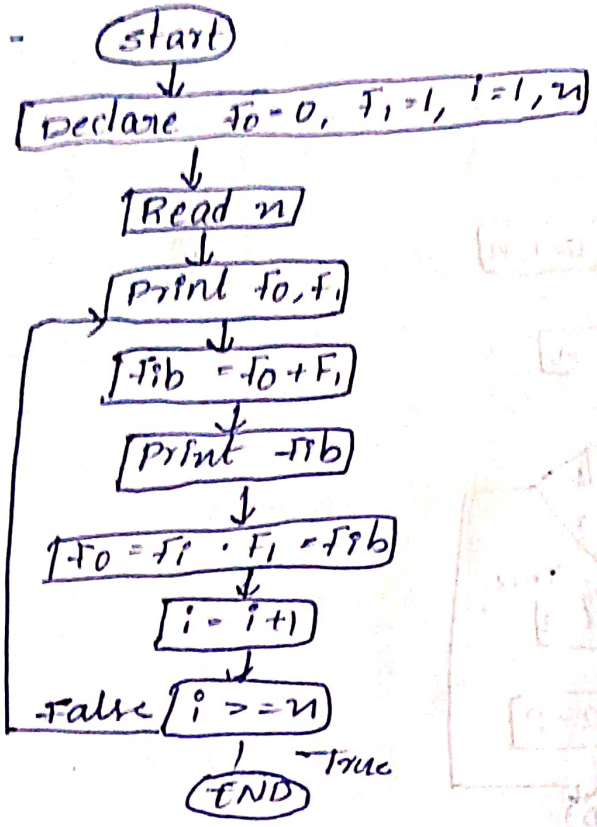
Algorithm - step 1 - Declare a=1,n
step 2 - Read n
step 3 - While a<=n
step 4 - if is the condition is true, print a
step 5 - Increment a with a

Pseudocode - Declare a=1,n
Read n
while a<=n
print a
a=a+2
While - END.

Result:- Successfully written algorithm, Flow chart, Pseudocode for generation of odd number.

Exp-4 Generation of fibonacci series

Flowchart -



Aim :- To write algorithm, Flowchart, Pseudocode for Generation of fibonacci series (0,1,1,2 ..... N)

Algorithm :- step 1 - Declare $f_0=0$, $f_1=1$, $i=1,n$
step 2 - Read n
step 3 - Assign $fib=f_0+f_1$
step 4 - Write fib
step 5 - Assign $f_0=f_1$, $f_1=fib$
step 6 - Increment i with 1. check until $i>=n$

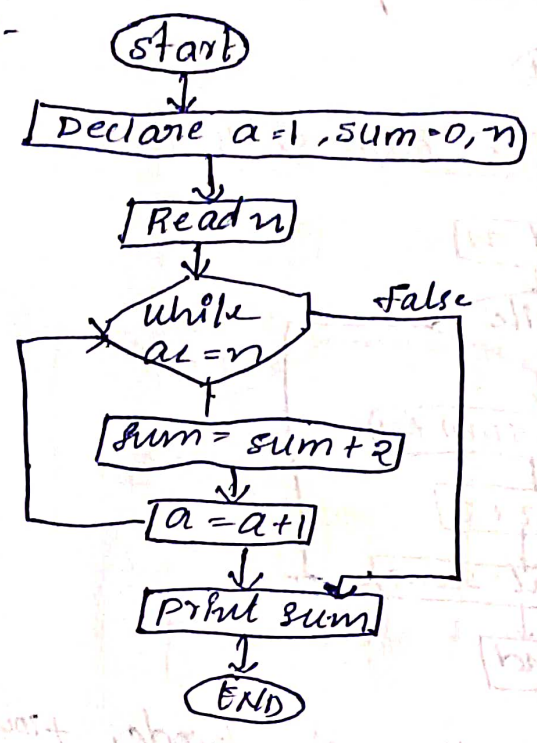
Pseudocode - Declare $f_0=0$, $f_1=1$, $i=1,n$
Read n - print f0f,
Repeat
fib = f0 + f1
Print fib
f0 = f1 , f1 = fib
i = i+1

Result: Successfully written Flowchart, Pseudocode algorithm for Generation of fibonacci series.

Exp-5

summing up numbers.

Flowchart :-



Aim :- To write algorithm, pseudocode, flowchart, for swimming up series.

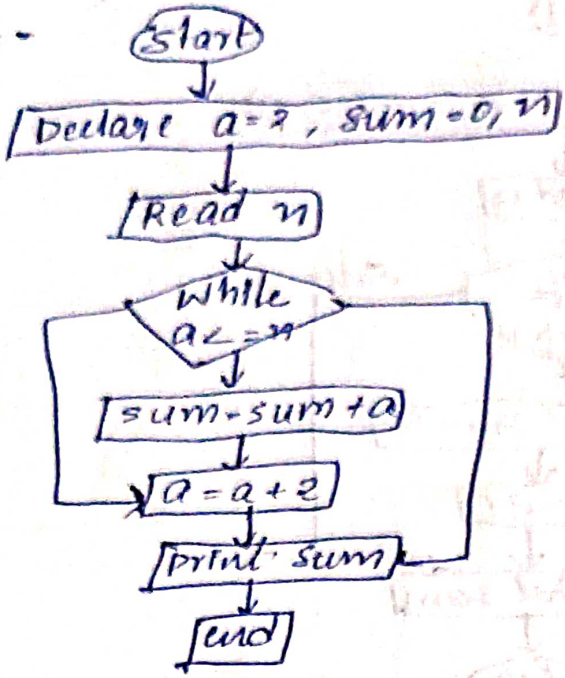
Algorithm - step 1 - Declare a=1, n, sum=0
step 2 - Read n
step 3 - While a<=n
sum = sum + a
a = a + 1
While - End
step 4 - print sum.

Pseudocode - Delare a=1, sum=0, n
Read n
while (a<=n)
sum = sum + a
while - END
print sum

Result :- Successfully written flowchart, pseudocode, flowchart for swimming up series (0, 1, 1, 2, 3, 5 ..... n)

Exp-6 Summing up series

Flowchart -



Aim - To write algorithm, pseudocode, flowchart for summing up series.

Algorithm - step 1 - Declare a=2, sum=0, n
step 2 - Read n
step 3 - while a<=n
step 4 - Add sum and a and assign as sum
step 5 - Increment a with a
step 6 - print sum.

Pseudocode -

```
Declare a=2, sum=0, n
Read n
while a<=n
Sum = sum + a
  a = a+2
while end
print sum.
```

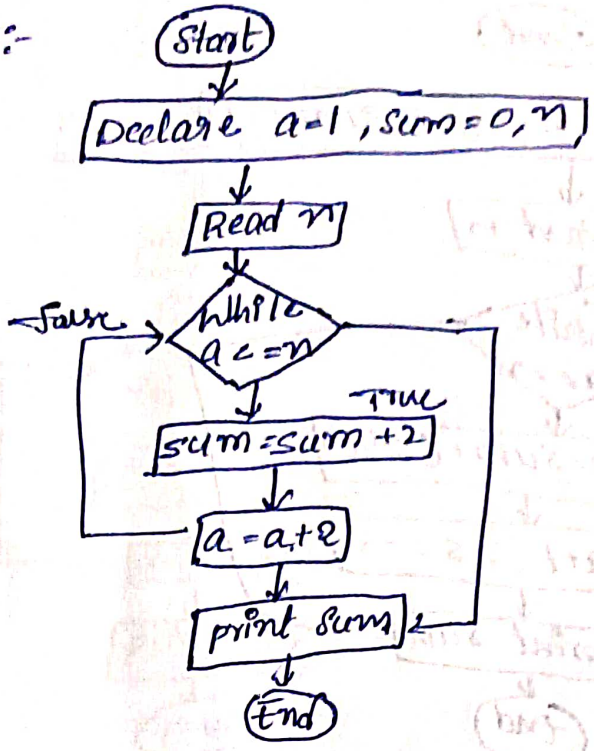
Result :- Successfully written a algorithm, flowchart, Pseudocode for summing up series

(2+4+6+.....+n)

EXP-07

Summing up series.

Flow chart :-



Aim :- To write a algorithm, pseudocode, and flowchart for summing up series (1+3+5.....n)

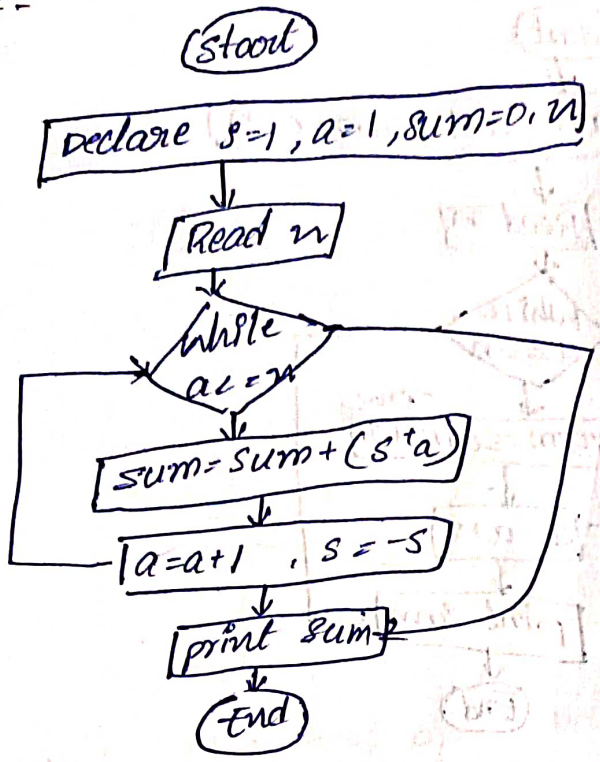
Algorithm :- step 1 - Declare a=1, sum=0, n
step 2 - Read n
step 3 - check a<=n
step 4 - add sum and a and assign it as sum.
step 5 - Increment a with 1
step 6 - print sum.

Pseudocode :- Declare a=1, sum=0, n
Read n
While (a<=n)
sum=sum+a
a=a+2
print sum.

Result :- successfully written flowchart, algorithm, pseudocode for summing up series
(1+3+5+.....+n)

EXP.NO-8 summing up series (1-2+3-4+-----+n)

Flow chart -



Aim:- To write algorithm, flowchart, pseudocode for the summing up series. (1-2+3-4+5-----+n)

Algorithm:- Step 1 - Declare s=1, a=1, sum=0, n

Step 2 - Read n

Step 3 - While a<=n

Step 4 - product sum and a and add with sum assign as sum.

Step 5 - Increment a with 1 and multiply -1 with s.

Step 6 - print sum.

Pseudocode:- Declare s=1, a=1, sum=0, n.

Read n

While a<=n

sum=sum+(s*a).

a=a+1, s=-s

print end sum

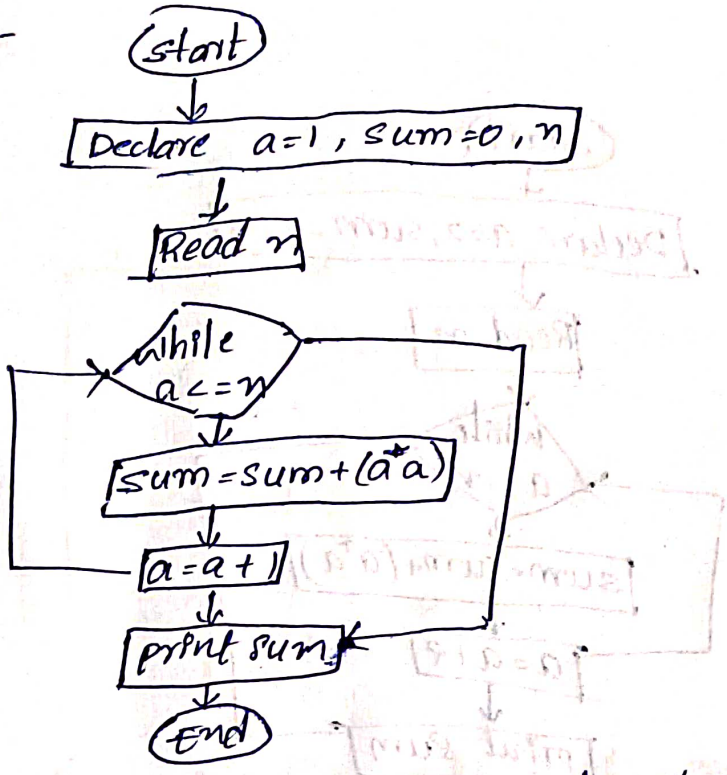
Result:- successfully written algorithm, flowchart, pseudocode for summing up series.

(1-2+3-4+5-----+n)

Exp-9

summing up series $(1^0+2^0+3^0+\ldots\ldots+n^0)$

Flowchart :-



Aim - To write algorithm, pseudocode, flowchart for summing up the series $(1^0+2^0+3^0+4^0\ldots\ldots+n^0)$

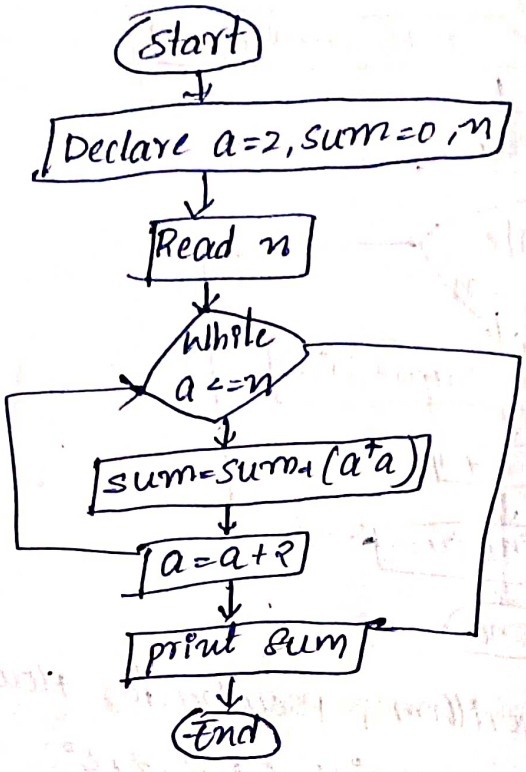
algorithm :- step 1 - Declare a=1, sum=0, n
step 2 - Read n
step 3 - check while a<=n
step 4 - add sum and square of 'a' and assign sum
step 5 - Increment a with 1
step 6 - print sum.

pseudocode - Declare a=1, sum=0, n
Read n
while a<=n
sum= sum+(a*a)
a=a+1
while-end, print sum.

Result :- Successfully written an algorithm, flowchart, pseudocode for summing up series..
$(1^0+2^0+3^0+4^0\ldots\ldots n^0)$

EXP-10 summing up series $(2^2+4^2+6^2+\ldots\ldots+n^2)$

Flow chart:-



Aim:- To write a algorithm, pseudocode, Flow chart for series $(2^2+4^2+6^2+\ldots\ldots+n^2)$

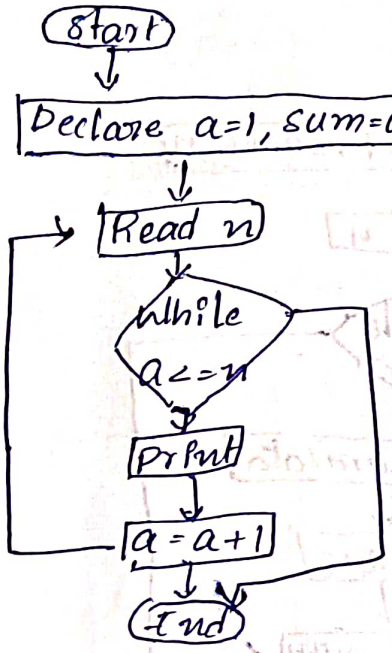
Algorithm:- step 1 - Declare a=2, sum=0, n
step 2 - Read n
step 3 - check while a<=n
step 4 - Add sum and square of a assign as sum.
step 5 - Increment a with 2.
step 6 - print sum

Pseudocode:- Declare a=2, sum=0, n
Read n
while a<=n
sum= sum (a*a)
a=a+2
print sum.

Result:- successfully write a algorithm, Pseudocode, Flowchart for summing up series.

EXP-11 summing up $(1^1+2^2+3^3+4^4+\cdots+n^n)$

Flow chart :-



Aim :- To write a algorithm, Flowchart, Pseudocode for summing up series.

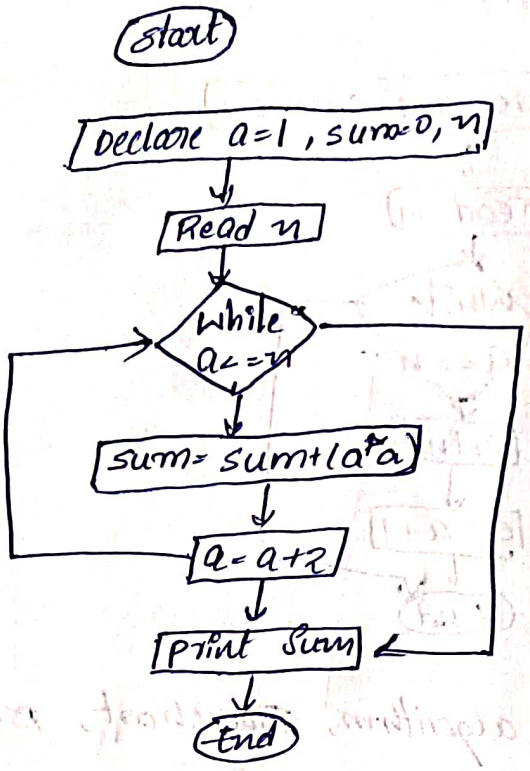
Algorithm :- step 1 - Initialize a with 1
step 2 = Read n
step 3 = check a is less than or equal to n
step 4 = print a
step 5 = Increment

Pseudocode :- Declare a=1, n
Read n
While a<=n
Print a
a=a+1
End-while.

Result :- successfully write a algorithm, Flowchart, Pseudocode of number series.
$(1^1+2^2+3^3+\cdots n^n)$.

EXP-12 summing up of square of odd numbers

Flow chart =



Aim :- To write a algorithm, Flowchart, Pseudocode for summing up for square of odd number.

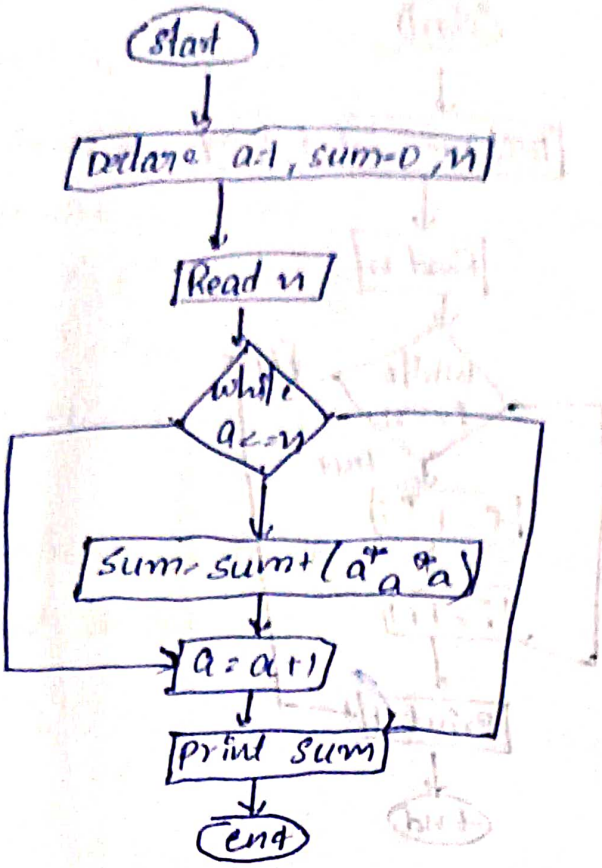
Algorithm :-
step 1 - Declares a=1, sum=0, n
step 2 - Read n
step 3 - Check while a<=n
step 4 - Add sum and a power of a and assign as sum.
step 5 - Increment a with 1
step 6 - print sum.

Pseudocode :- Declare a=1, sum=0, n
Read n
while (a<=n)
sum= sum+(a*a)
a=a+2
Print Sum

Result - Successfully written algorithm, Pseudocode, Flowchart for summing up of square of odd numbers.

EXP-13 summing up cubes of 'n' number

Flowchart-



Aim= To write algorithm, flowchart, pseudocode for summing up cubes of n numbers.

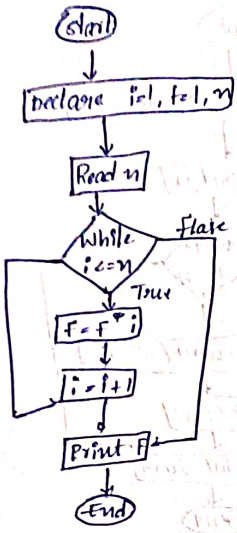
Algorithm :- step 1 - Declare a=1, sum=0, n
step 2 - Read n
step 3 - Check while a<=n
step 4 - Add sum and cubes of a and assign as sum
step 5 - Increment a with 1
step 6 - print sum.

Pseudocode - Declare a=1, sum=0, n
Read n
while (a<=n)
sum=sum+(a*a*a)
a=a+1
print sum.

Result: successfully written algorithm, flowchart, for summing up cubes of n number.

Exp. 11 Product series

Flowchart -



Aim - To write a algorithm, Pseudocode and Flow Chart for product series

Algorithm - step 1 - Declare i= f=1, n

step 2 - Read n

step 3 - check while i<=n

step 4 - multiply f and i and assign as f

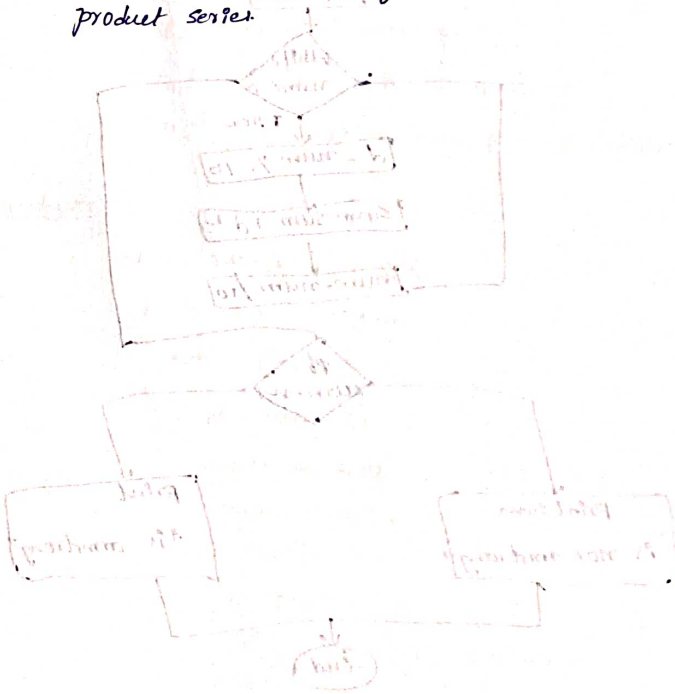
step 5 - Increment i with 1

step 6 - print f.

Pseudocode -

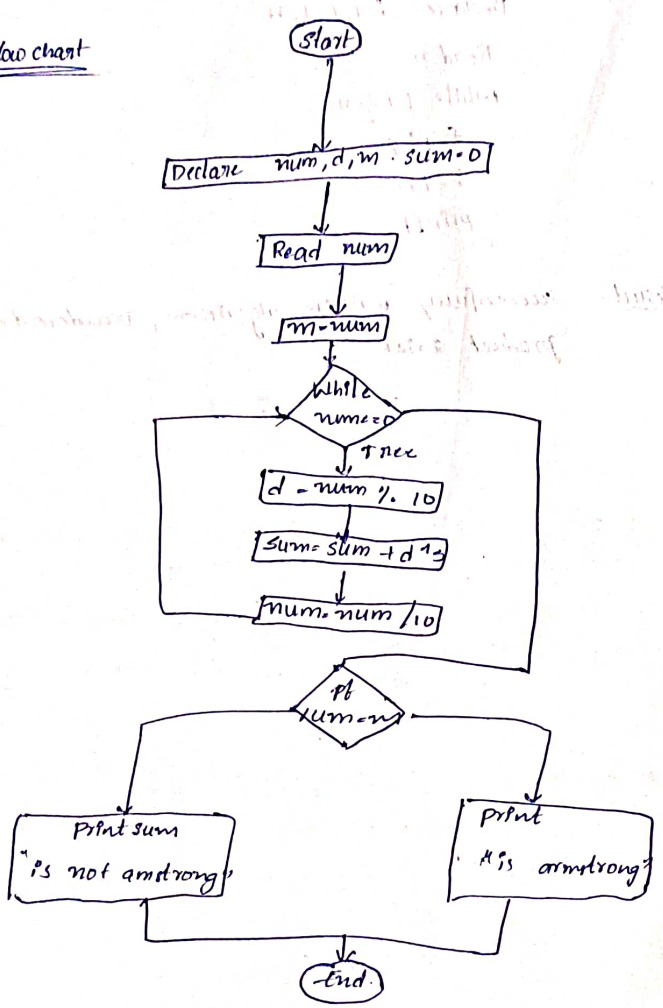
```
Declare  i=1, f=1, n
Read n
While (i<=n)
  f=f*i;
  i=i+1
printf.
```

Result - succesfully written algorithm, Pseudocode f product series.



Exp-15 ARMSTRONG OR NOT

Flow chart



Aim - To write algorithm, pseudocode, and flow chart for armstrong or not.

Algorithm-

Step 1 - Declare num, d, m, sum=0

Step 2 - Read num

Step 3 - m= num, check, num!=0

Step 4 - assign d=0, num % 10, sum = sum + d^3
num=num/10

Step 5 - check if sum = m

Step 6 - print "sum is armstrong"

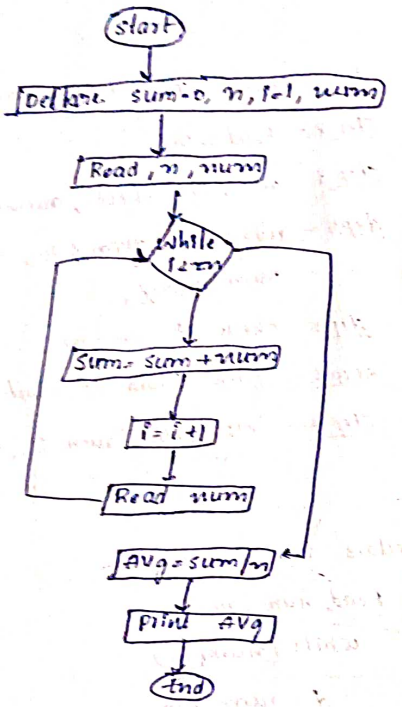
Step 7 - else print "num is not armstrong"

Pseudocode -

```
Declare num, d, m, sum=0
Read num, m=num
while (num!=0)
d = num %10
sum=sum +d^3
num=num/10
while -end
if sum=m
print "num is armstrong"
else
print "num is not armstrong."
```

Exp-11 summing up and N number and finding average.

Flowchart:-



Aim - To write algorithm, Flowchart, summing up N numbers and finding average.

Algorithm-

step 1 - Declare sum=0, n, i=1

step 2 - Read n, num

step 3 - check while (i<=n)

step 4 - Add sum and num and assign

step 5 - Increment i with 1 and Read num

step 6 - calculate Avg = sum/n, print Avg.

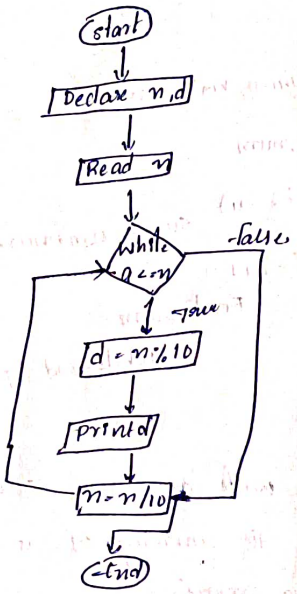
Pseudocode :-

```
Declare sum=0, n, i=1, num
Read n, num
While (i<=n)
        sum= sum+num
        i= i+1
        Read num
        While-end - print Sum.
```

Result :- successfully write algorithm, flowchart, pseudocode for summing up any N number and finding average.

Exp-17 printing the digit of the number.

Flowchart :-



Aim:- To write algorithm, pseudocode, and flowchart for printing the digit of the number.

Algorithm:- step 1 - Declare n,d

Step 2 - Read n

step 3 - while n!=0

step 4 - calculate d=n%10 printd

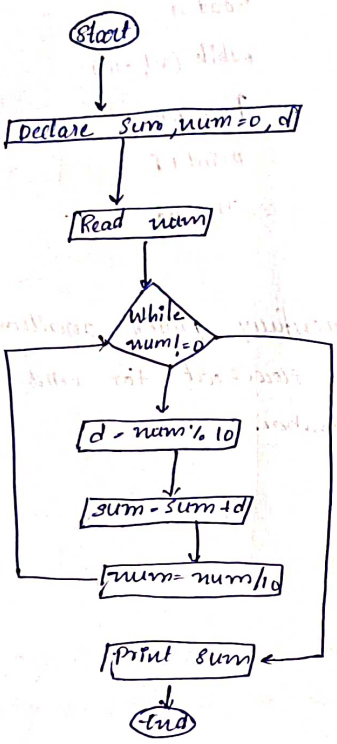
Step 5 - calculate n=n/10

Pseudocode :-

```
Declare n,d
Read n
while (n!=10)
d = n%10
print+d
n=n/10
```

Result :- successfully written algorithm, pseudocode, and flowchart for print and digit of the number.

Exp-18 Summing up of digits of an integer



Aim :- To write algorithm, pseudocode, Flowchart for summing up of digits of an integer.

Algorithm :-

Step1 = Declare num, sum=0, d

Step 2: Read num

step 3 = check while num!=0

step 4 = calculate d=num %.10 n=n/10.

step5 = Add sum and d and assign as sum.

step 6 = print sum.

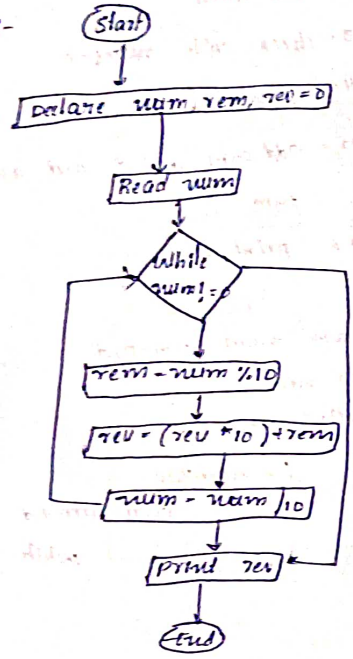
Pseudocode :-

```
Declare num, sum=0,d
Read num
While num!=0
    d = num%10
        sum=sum+d
            While- End
                print sum.
```

Result :- successfully written algorithm, pseudocode, Flowchart for summing up of digits of an integer.

Exp-19 Reversing the digits of a number.

Flowchart :-



Aim :- To write algorithm, pseudocode, Flowchart, for reversing the digit.

Algorithm :- step 1 - Declare num, rem, rev = 0

step 2 - Read num

step 3 - while (num! = 0)

step 4 - calculate rem = num % 10,
num = num / 10.

step 5 - add rem and rev * 10 and assign as rev

step 6 - print rev.

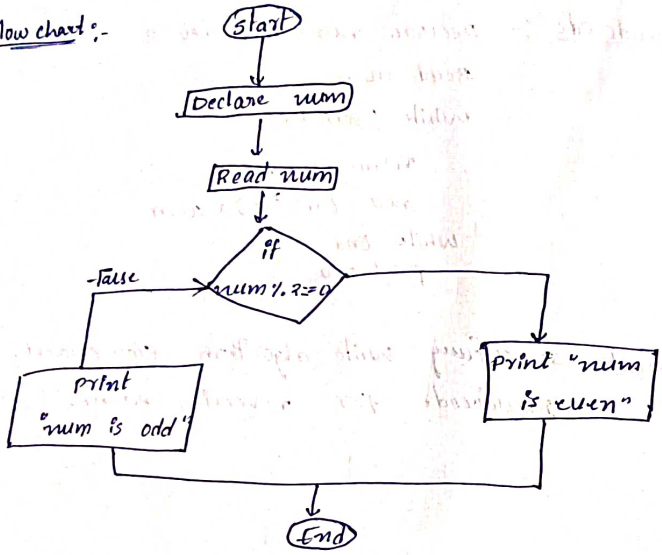
Pseudocode :-

```
Declare num, rem, rev=0
Read num
While (num! =0)
    rem = num % 10
    rev = (rev * 10) + rem
While -End
print rev.
```

Result - Successfully write algorithm, Flowchart, pseudocode for reversing num.

Exp-20 Finding wheather the given number is odd or even

Flow chart :-



Aim = To write algorithm, Pseudocode, Flowchart for given number is odd or even.

Algorithm :

Step 1 : Declare num

Step 2 = Read num

Step 3 = check the condition if num%2=0

Step 4 = condition is True, print num is even.

Step 5 = else print num is odd

Pseudocode =

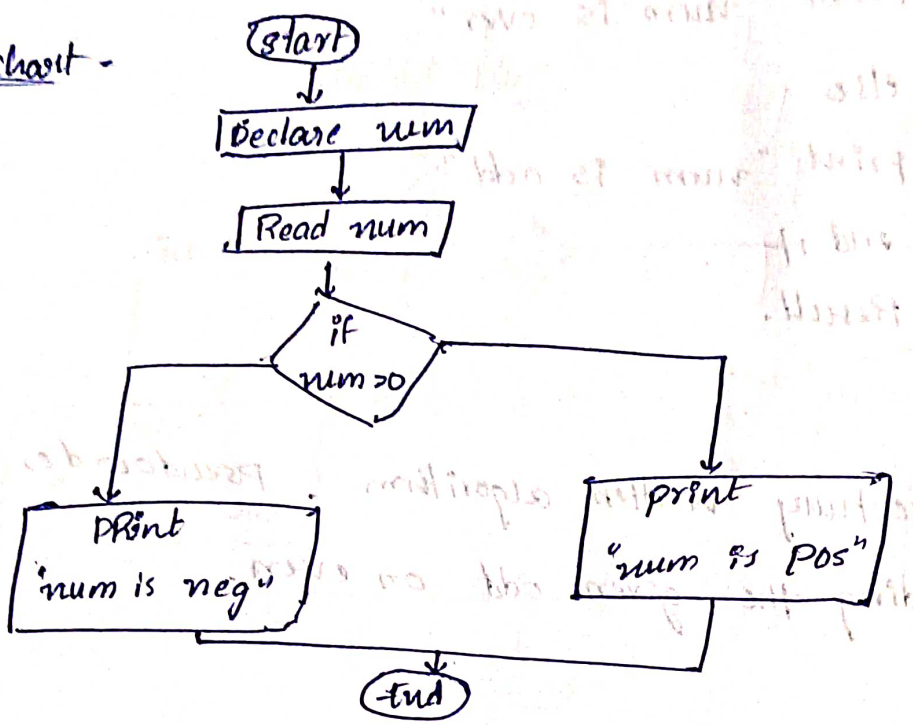
```
Declare num
Read num
if (num%2=0)
Print "num is even"
else
Print "num is odd"
end if
Result.
```

Result = Successfully written algorithm, Pseudocode, For finding the given odd or even.

Exp -> 1 finding whether the given integer is positive or negative.

Aim - To write algorithm, pseudocode, flowchart for finding whether given num is "+" or "-"

Flow chart -

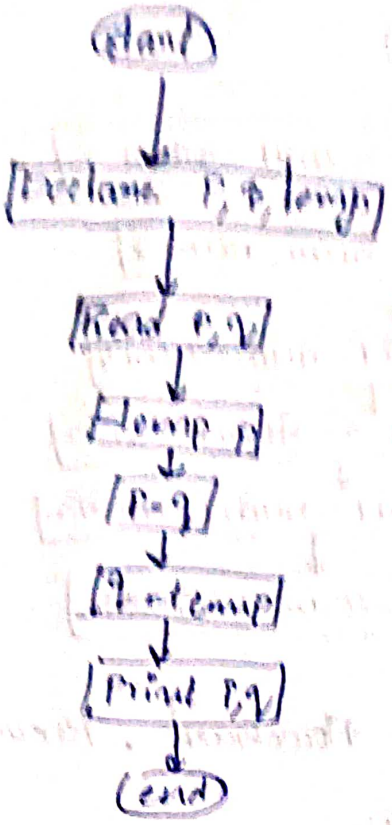


Algorithm - step1 = Declare num
step2 = Read num
step 3 = check the condition if num >0
step 4 = condition is true print positive else neg.

Pseudocode - Declare num
Read num
if num>0
Print "num is pos"
else
Print "num is neg"

Result - Successfully written Algorithm, Pseudocode for finding positive and negative number.

Exp-08 swapping of two numbers with temporary variable.



Aim- To write algorithm, pseudocode for swapping two num.

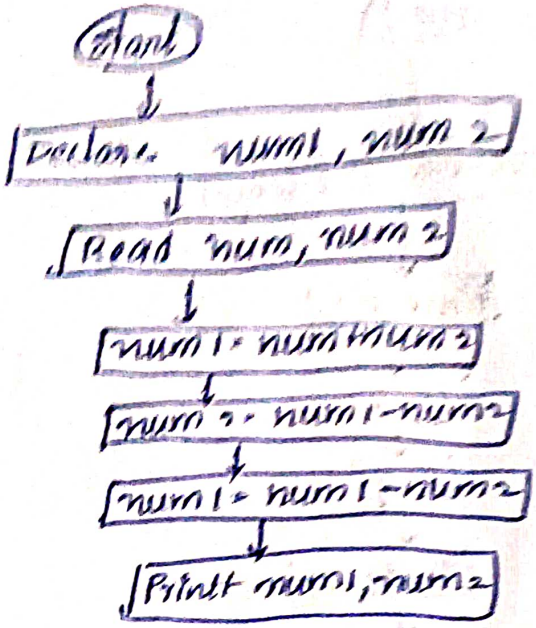
algorithm- step 1 - Declare P, Q, temp P
step 2 - Read P, Q
step 3 - assign temp - P
step 4 - assign P - Q and Q - temp
step 5 - Print P, Q

Pseudocode - Declare P, Q, temp P
Read P, Q
temp = P
P = Q
Q = temp
Print P, Q

Result:- successfully written algorithm, pseudocode for swapping two numbers.

Exp-23 Swapping numbers without using temp var

Flowchart -



Aim - To write algorithm, Flowchart, Pseudocode for swap of two numbers.

Algorithm - Step 1 - Declare num_1, num_2

Step 2 - Read num_1, num_2

Step 3 - Assign $num_1 + num_2 = num_1$ and $num_2 = num_1 - num_2$ & $num_1 = num_1 - num_2$

Pseudocode :- Declare num1, num2

Read num1, num2

num1 = num1 + num2

num2 = num1 - num2

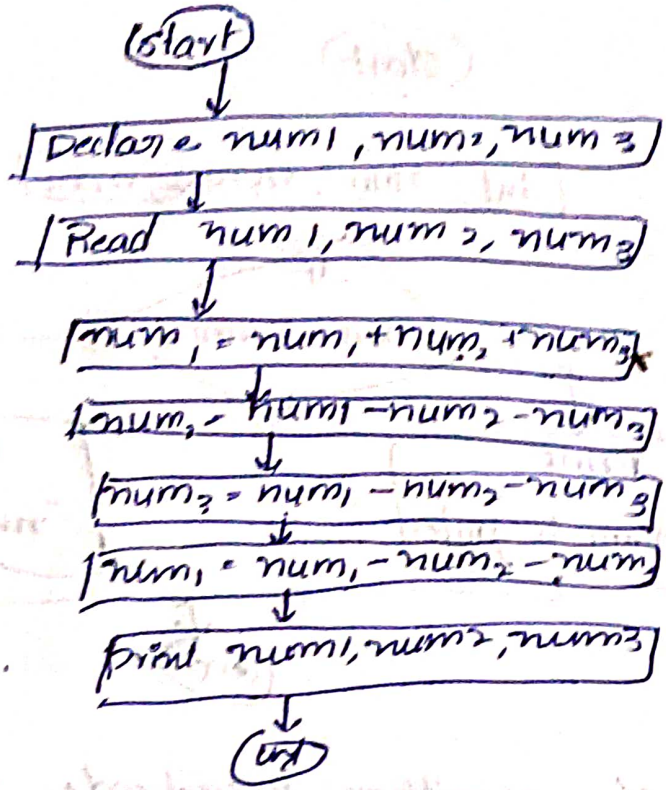
num1 = num1 - num2

Print num1, num2

Result :- Successfully written algorithm, Pseudocode, Flow chart for swapping numbers.

Exp-24 Swapping of three numbers

Flow chart :-



Aim :- To write Pseudocode, Algorithm for swap three numbers.

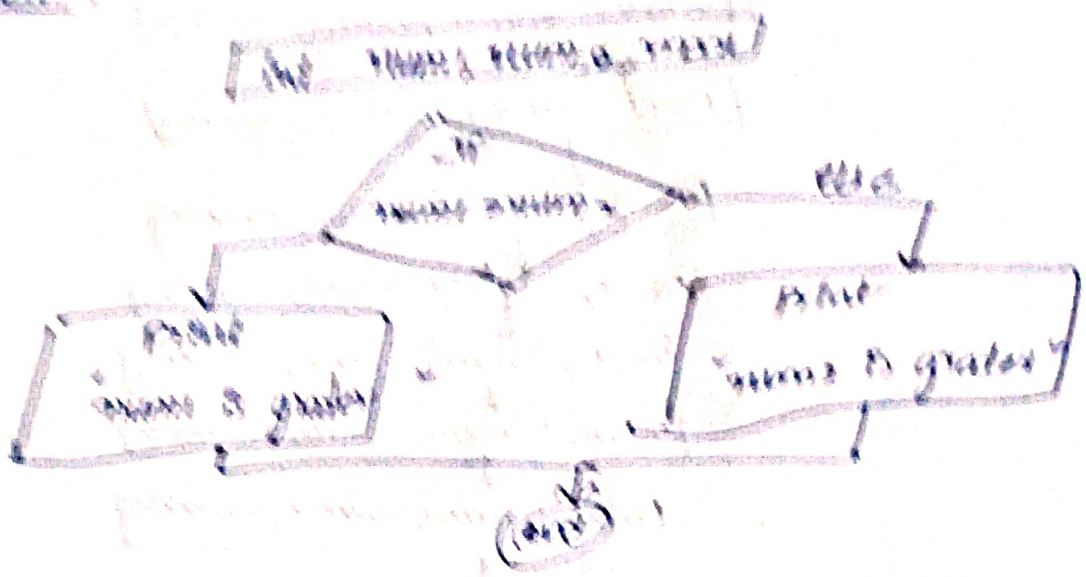
Algorithm :- step1 - Declare num1, num2, num3
step2 - Read num1, num2, num3
step3 - Assign num1 = num1 + num2 + num3
step4 - $num_2 = num_1 - num_2 - num_3$, $num_3 = num_1 - num_2$
step5 - print (num1, num2, num3)

Pseudocode - Declare num1, num2, num3
num1 = num1 + num2 + num3
num2 = num1 - num2 - num3
num3 = num1 - num2 - num3
num1 = num1 - num2 - num3
print num1, num2, num3

Result :- successfully written flow chart, Pseudocode, algorithm for swapping three numbers.

Exp 34 Biggest number

Flowchart :



Aim : To write algorithm, flowchart, and coding for finding biggest number.

Algorithm :
- Declare n_1, n_2, result
- Assign $n_1 = n_2$ or $n_2 = n_1$
- If $n_1 > n_2$ print n_1 is greater
- If $n_2 > n_1$ print n_2 is greater
- Print result.

Pseudocode :

```
number1, number2 declare
if (number1 > number2)
print number1
else
print number2
```

Result - successfully written pseudocode, flowchart, algorithm, for finding biggest number.