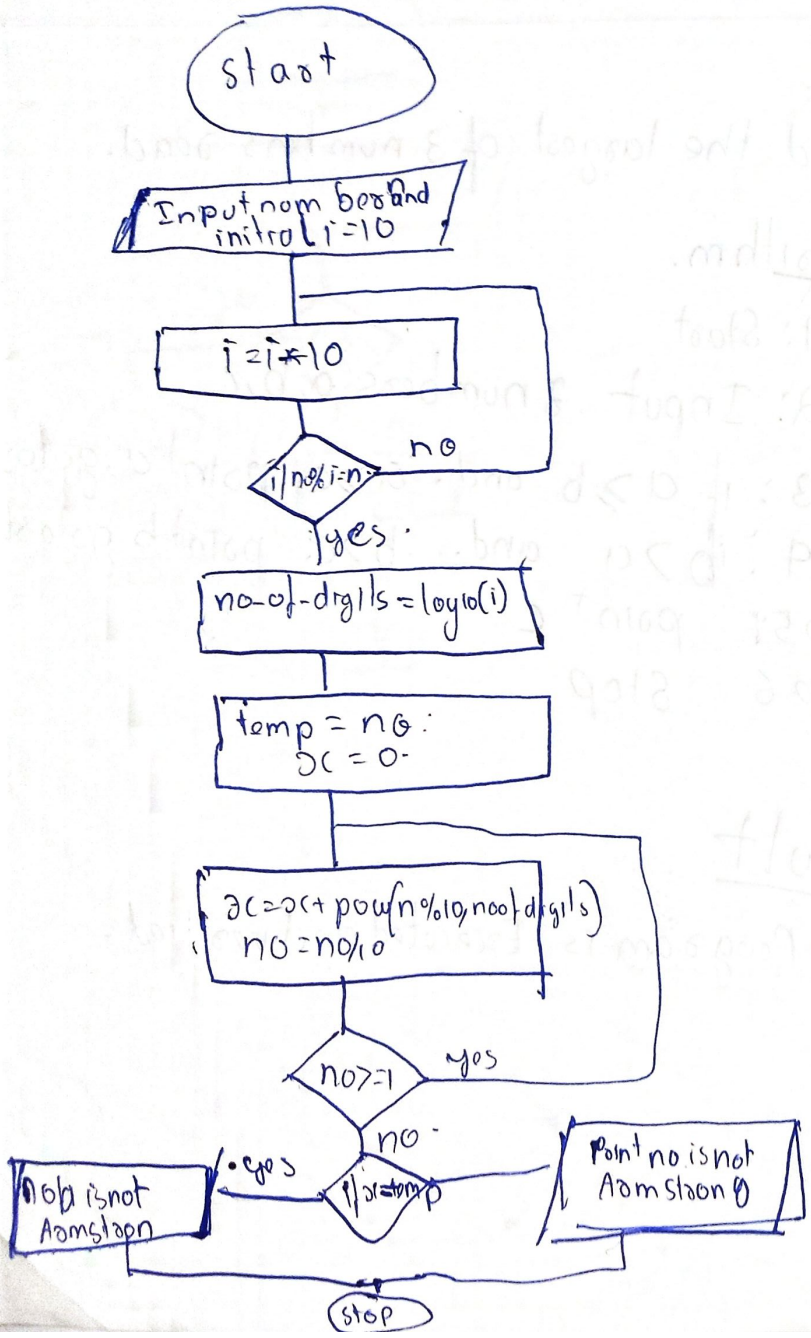
Flow Chart



Armstrong Number.

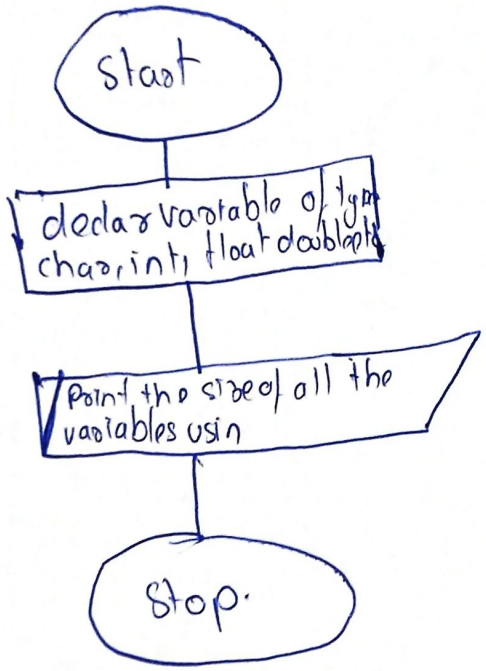
Aim

To check whether then digit number is Armstrong or not.

Algorithm.

step 1: Start
step 2: Input the number and declare i = 10.
step 3: check if. number % i = number.
step 4: Repeate step 3 and each time assign i=i* until number % i = number.
step 5: set no_of_digits to. log10(i)
step 6: Assign no temp to be equal to the number.
step 7: sc = sc + pow(n%10, no_of_digits)
no = no/10.
step 8: Repeat step 7 while number >= 1.
step 9: check if. sc == temp then point. the n the point true else point false.
step 10: Stop.

Flow Chart



Display size of variables using sizeof.

Aim

Write a Program to display the size of all variables using.

Algorithm

Step 1: Start

Step 2: Declare all the variables of differ datatypes

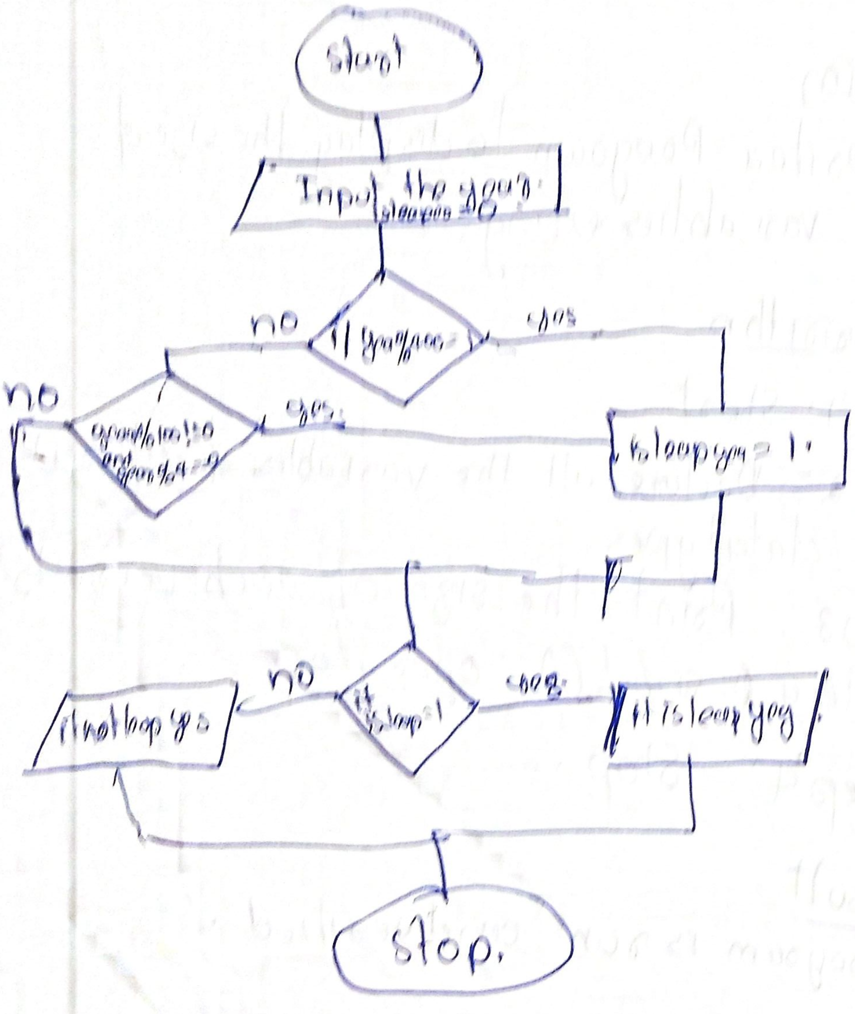
step 3: Print the size of each variable using sizeof() operator.

Step 4: Stop.

Result

Program is run and verified.

Flow Chart



Leap year check.

Aim

To check whether a year is leapyear or not.

Algorithm

Step1: Start

Step2: Input number, and declare is_leap year.

Step 3: If year is divisible by 400 is_leap_year=1.

Step 4: If year not divisible by ~~100~~ and divisible by ~~100~~ 4 then is_leap_year =1.

Step 5: ~~is leap~~ if is_leap_year =1 print.
the year is leap year.

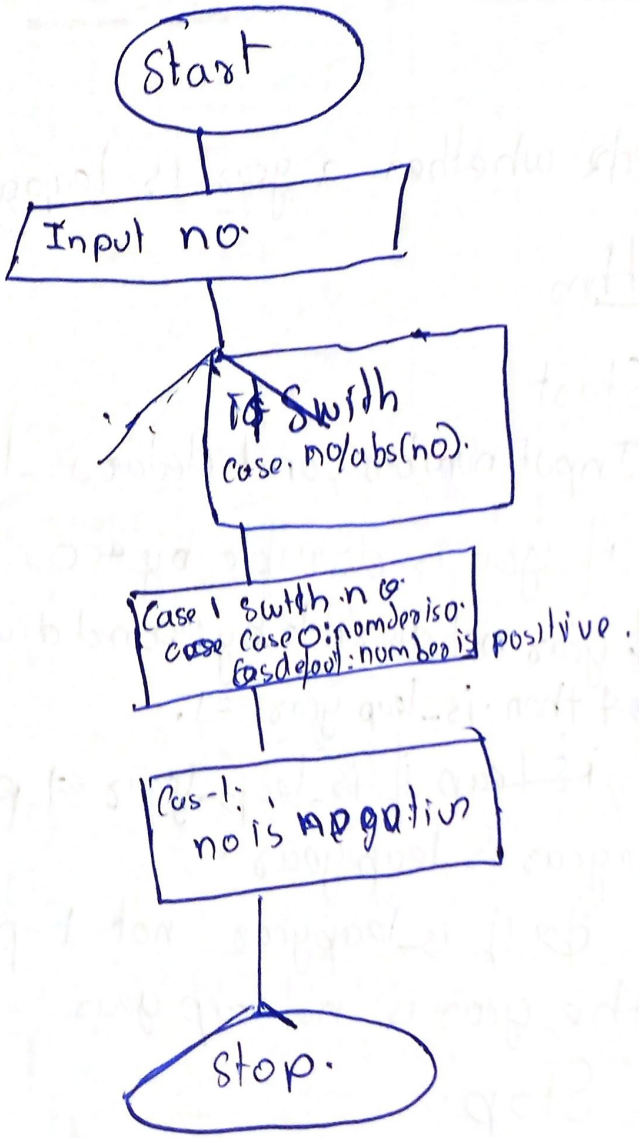
Step6: ~~Else~~ if is_leapyear not 1 print
the year is not leap year.

Step7: Stop.

Result

Program is run an verified

Flow Chart



Determine the number is positive or negative

Aim

Write a program to chek whether a num is positive negtive or zero using Switch ca

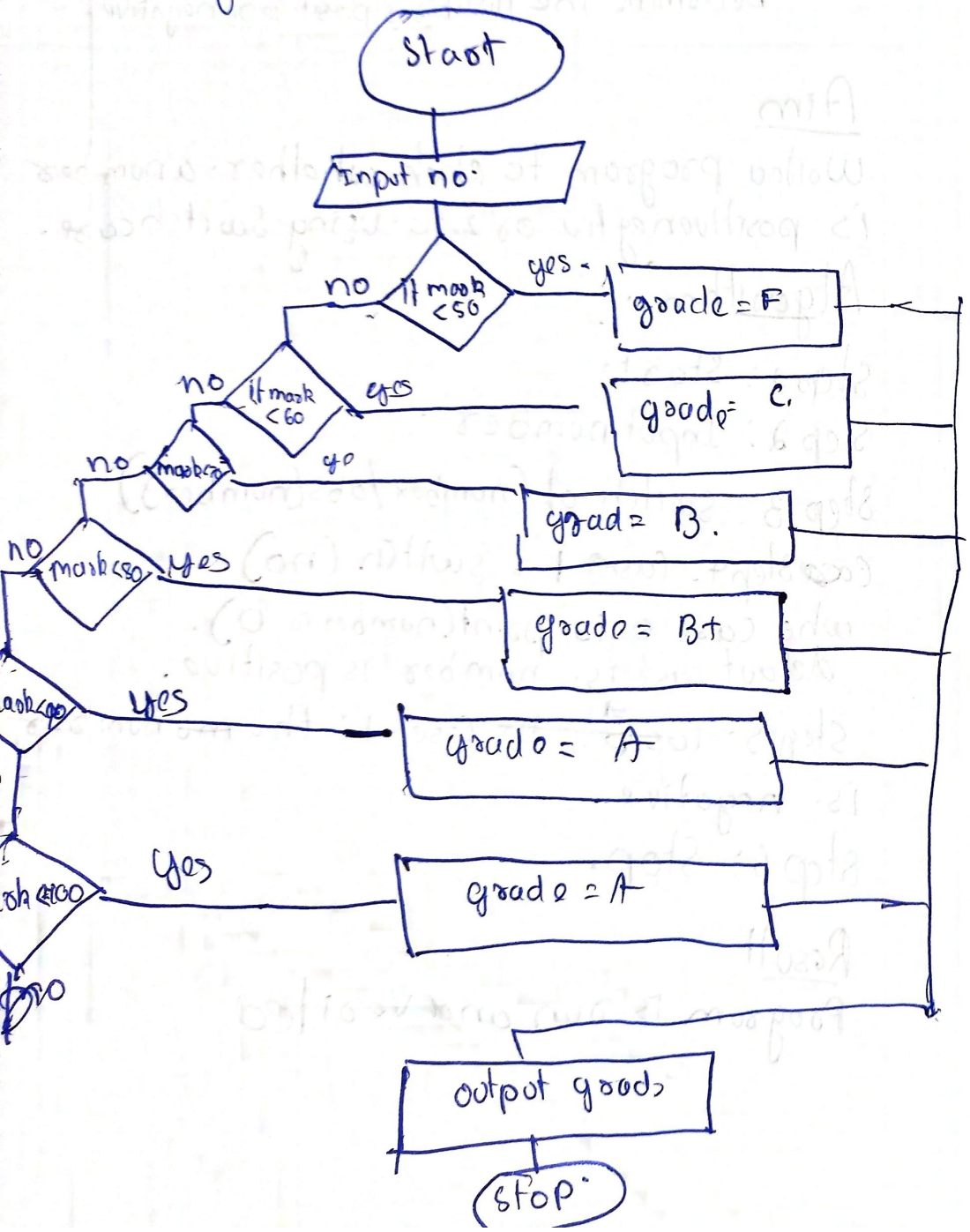
Algorithm

step 1: Start.
step 2: Input number.
step 3: switch of (number/abs(number)).
step 4: case 1 : swith (no).
whe case 0 is print(number is 0).
defaut case is: number is positive.
step 5: case -1: the num is negative.
step 6: Stop.

Result

Program is run and verified

Algorith.



Computer the grades of student

Aim

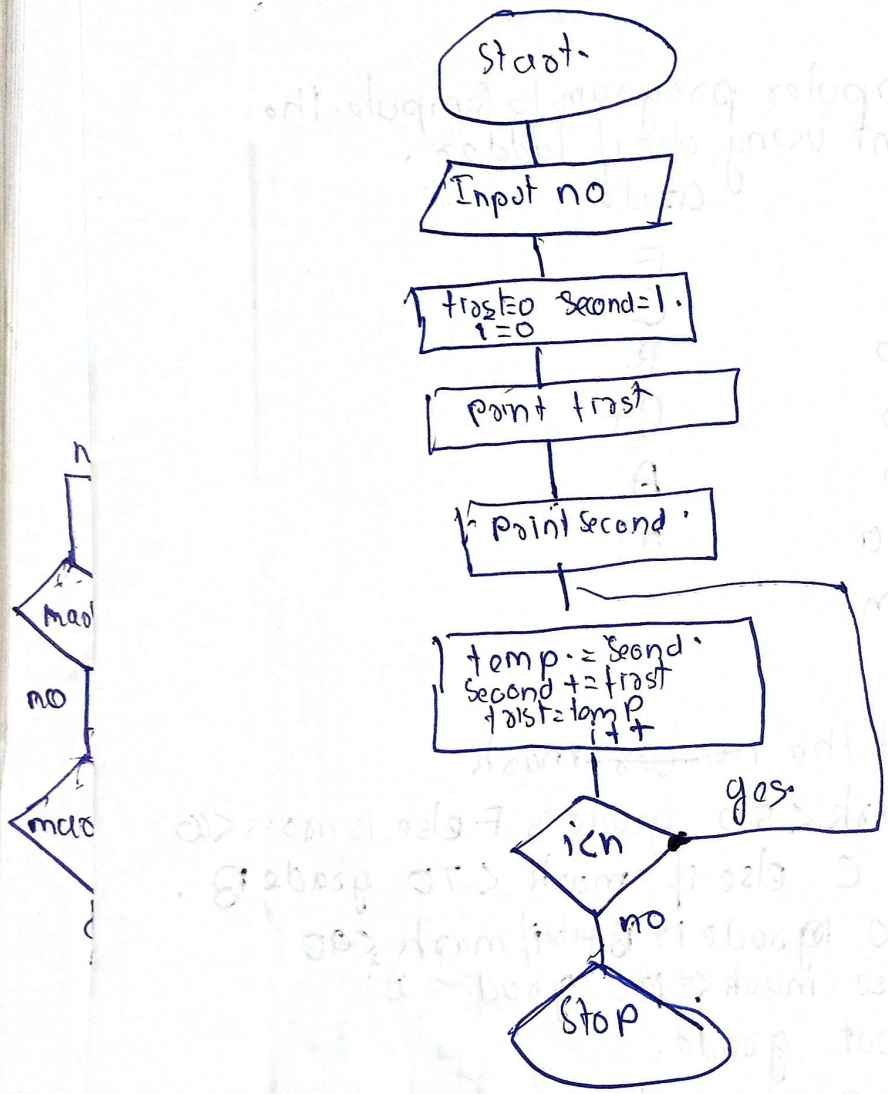
Write a computer program to compute the grad of student using else if ladder.

marks	Grade.
mark < 50	F
50 ≤ mark < 60	C
60 ≤ mark < 70	B.
20 ≤ mark < 80	B+
80 ≤ mark < 90	A
90 ≤ mark < 10	A+.

Algorithm

Step 1: Start
step 2: Input the ~~number~~ mark
Step 3: if mark < 50 grade is F else is mark ~~mark~~ grade is C else if mark < 70 grade is B else if mark < 80 grade is B+ else if mark ≤ 90 grade is A else mark <= 100 grade = A+
step 4: output grade.
Step 5: Stop.
Result: Program is run and verified

Flow Chart



Fibonacci Series

Aim:

Generat Fibonacci Series. upto given no

Algorithm

Step 1: Start

Step 2: Enter no and decl. delare first ≥0 and second=1
int i=0.

Step 3: Print first.

Step 4: Print Second.

Step 5: temp = second.
second = second + first
first = temp.
i++

Step 6: Steps repeat 4) and step(5) till i<n.

Step 7: Stop.

Result

Program is run and verified