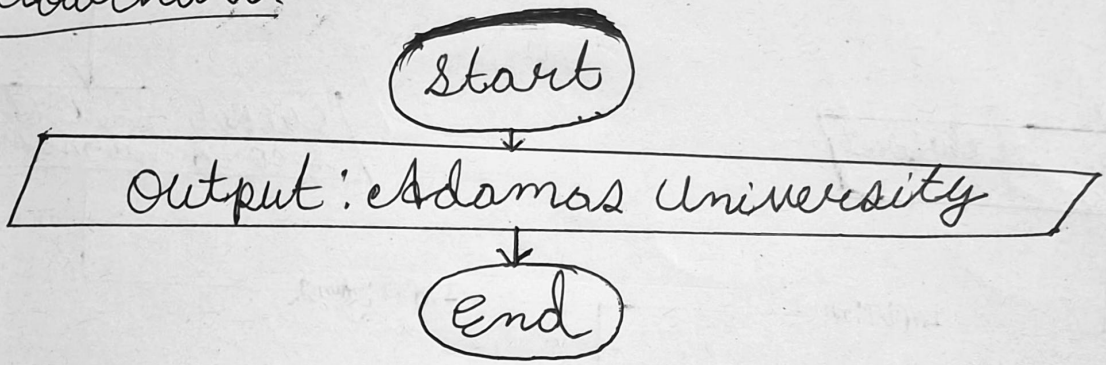
Algoritham And Flowchart :-

1. Write a C program to print 'ADAMAS UNIVERSITY'

Algoritham :-

Step 1: Start
Step 2: Print Adamas University
Step 3: End.

Flowchart:

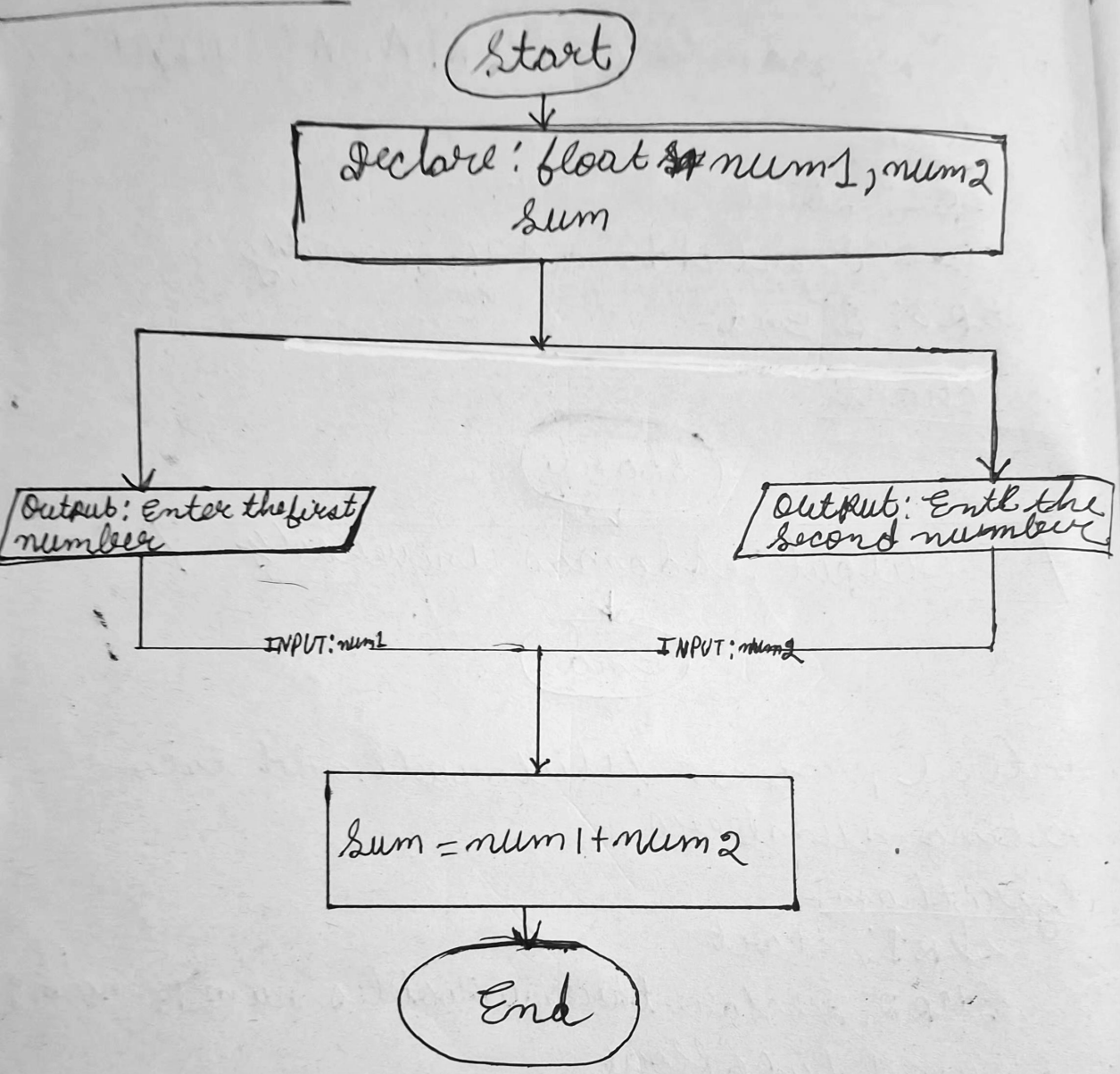


2. Write a C program which will add two floating numbers.

Algoritham :-

Step 1: Start
Step 2: Declare three variables num 1, num 2 and sum of type float.
Step 3: Input the 1st number from user and store it in num 1.
Step 4: Input the 2nd number from the user and store it in num 2.
Step 5: Add the two numbers. ~~Sum = num1+nu~~ Sum = num 1 + num 2

Flowchart:



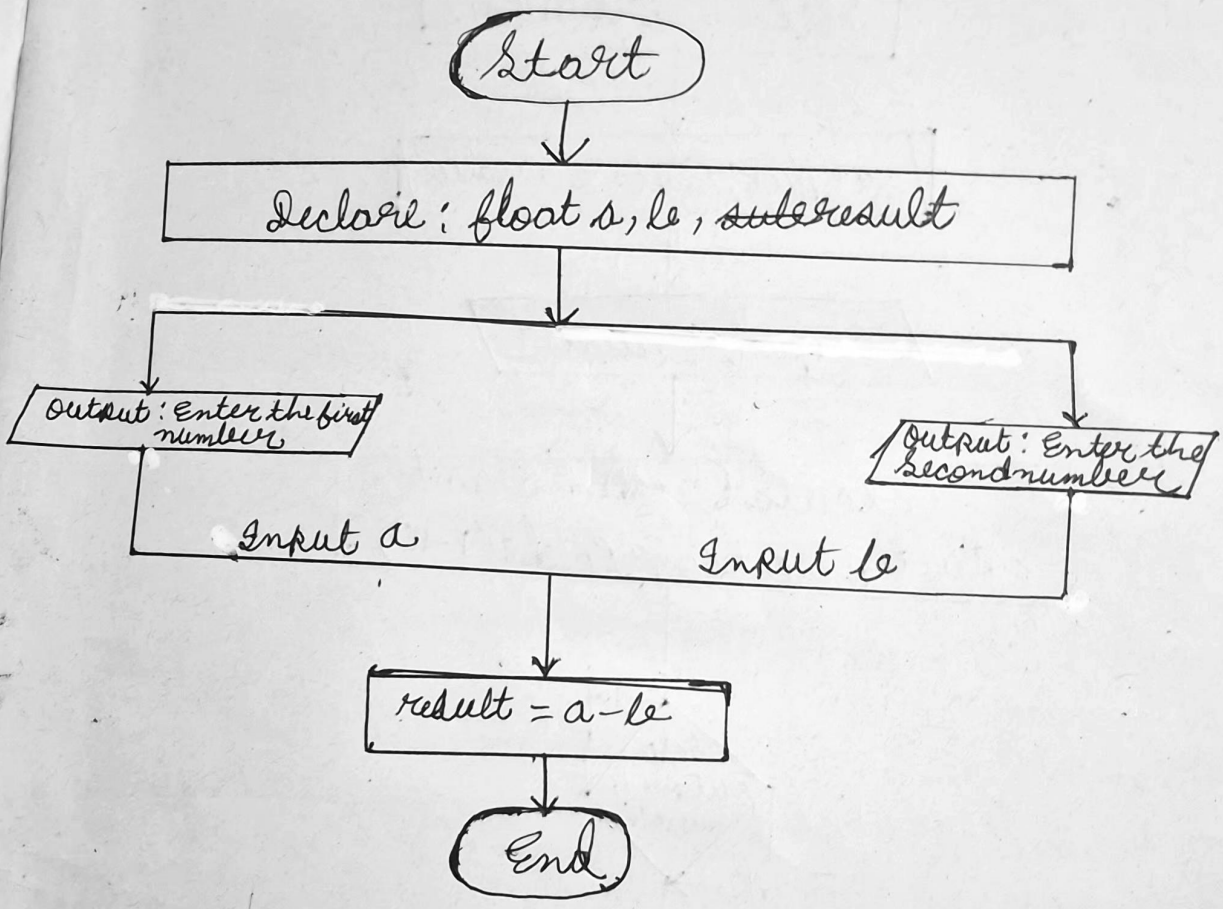
3. Write a C program to subtract two numbers taking from the user.

Algorithm:

Step1: Start
Step 2: Declare three integer variables: a, b and result.
Step 3: Input the first number and store in a.
Step 4: Input the second number and store in b.
Step 5: Subtract the second number from the first:
result = a - b

Step 6: Display the result using printf().
Step 7: Stop.

Flowchart:

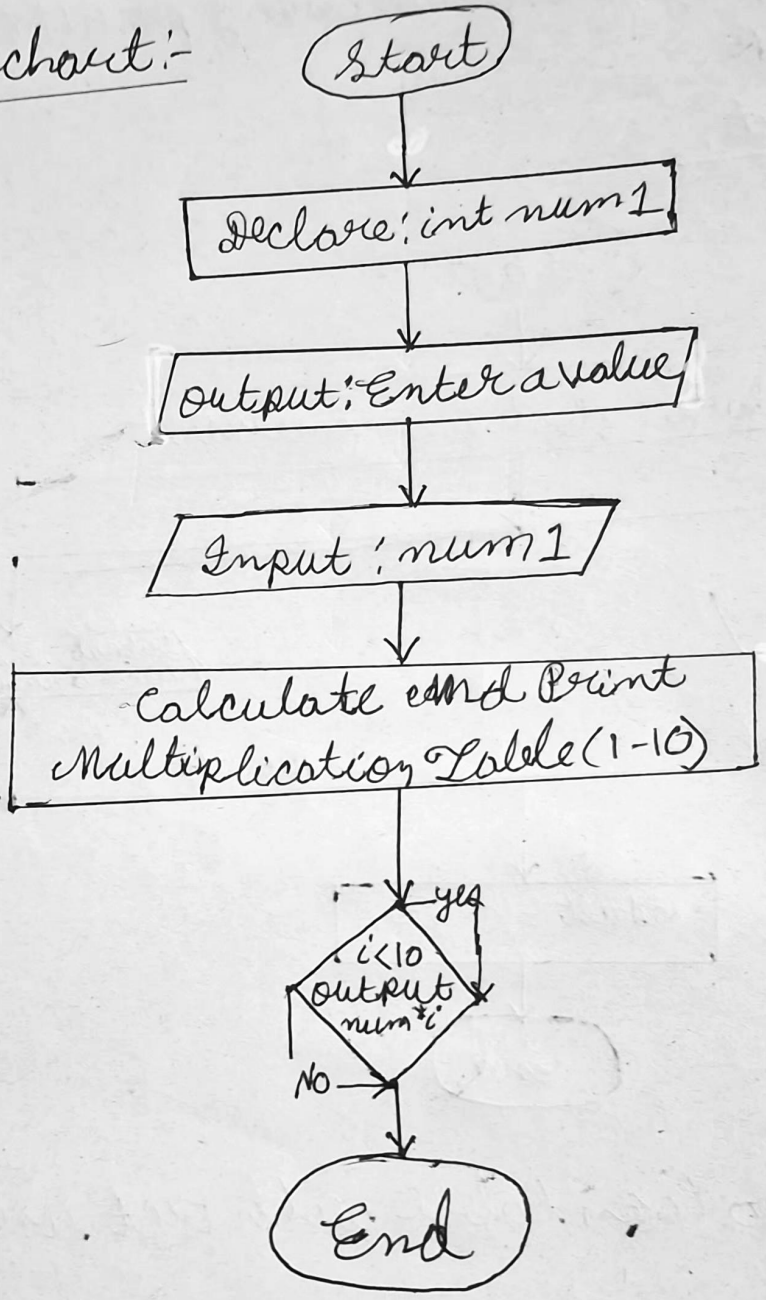


4. Multiplication table with out using loop

Algoritam:-

Step 1: Start
Step 2: Declare an integer variable num.
Step 3: Input a number from the user and store it in num
Step 4: Multiply num by 1, 2, 3, ----, 10.
Step 5: Print each result in the formot: num*i = result.
Step 6: Stop

Flowchart:-



5. ASCII value of the character

Algoritam:-

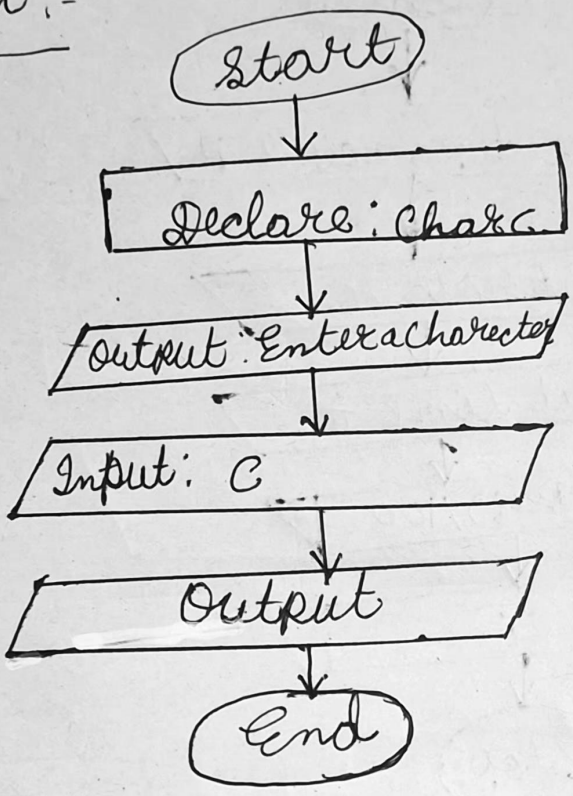
Step 1:- Start

Step 2:- Declare a variable c of type char

Step 3:- Input a character from the use
and store it is c,

Step 4:- Stop.

Flowchart:-

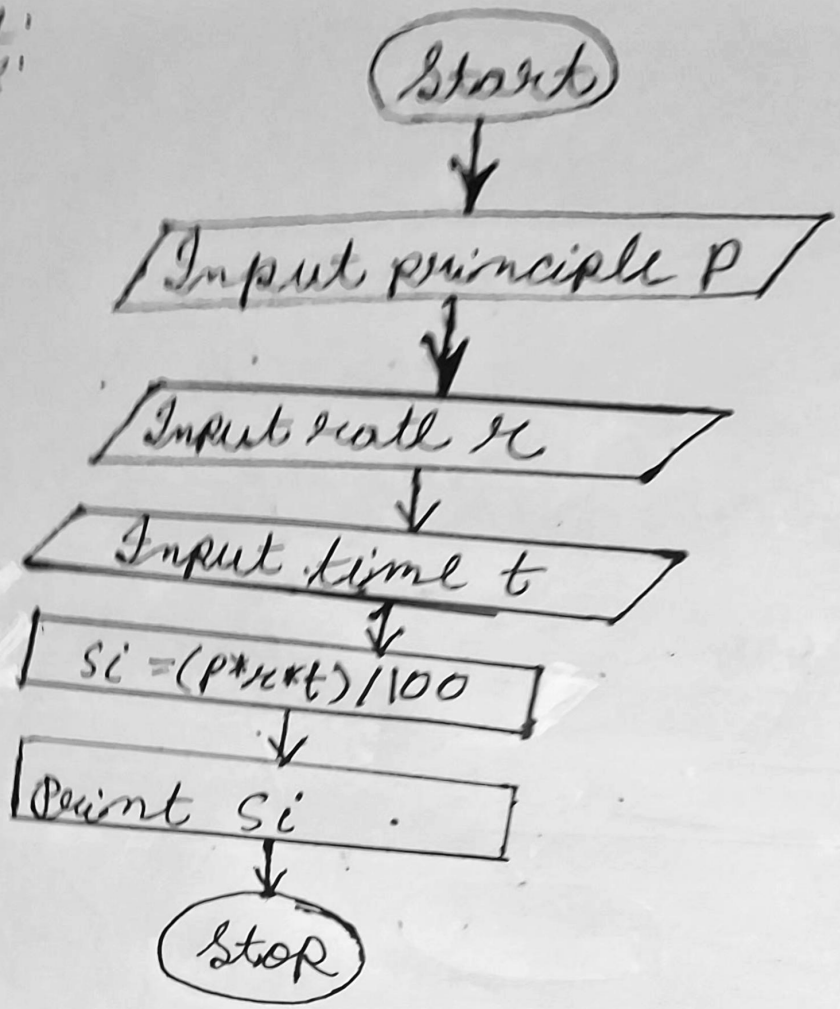


6. Calculate simple interest

Algoritham:-

Step 1: Start
Step 2: Declare variable p (principal), r (rate), t (time), si (simple interest).
Step 3: Input the principal amount p.
Step 4: Input the rate of interest r.
Step 5: Input the time period t.
Step 6: Calculate simple interest: $si = (p * r * t)/100$.
Step 7: Display the value of si.
Step 8: Stop.

Flowchart:



7. Calculate the area of a circle.

Algorithm:-

Step 1: Start

Step 2: Declare variable r (radius) and area as float.

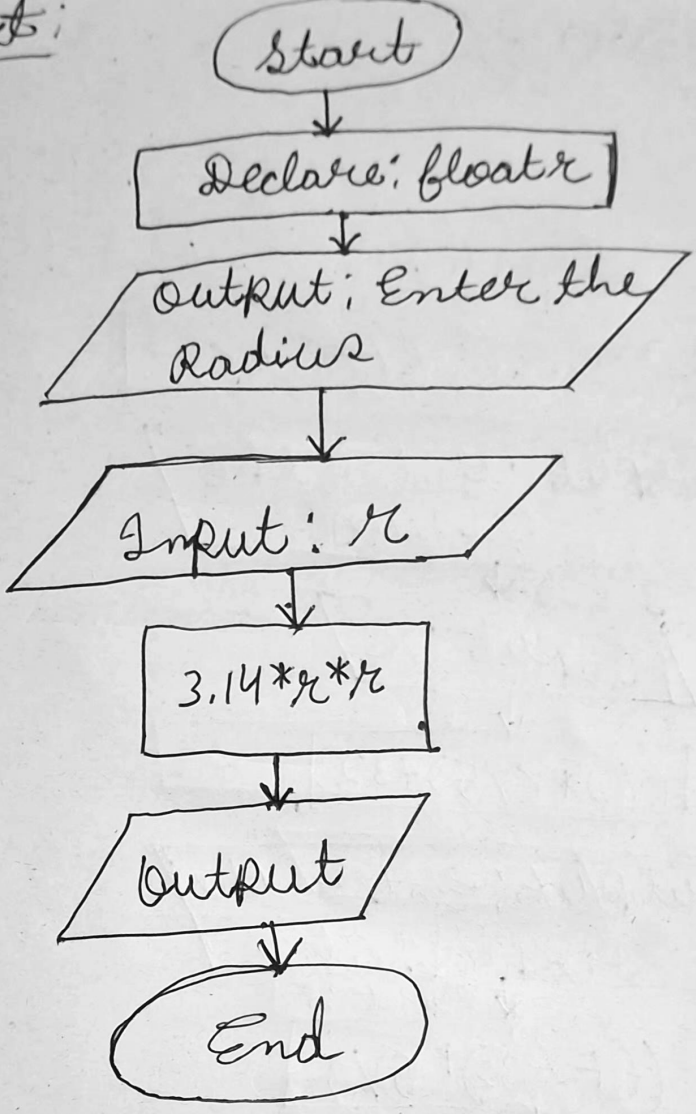
Step 3: ~~Prompt~~ Input the user to enter the radius.

Step 4: Calculate the area using the formula:
area $= 3.14 * r * r$

Step 5: Display the area of the circle on the screen.

Step 6: End.

Flowchart:

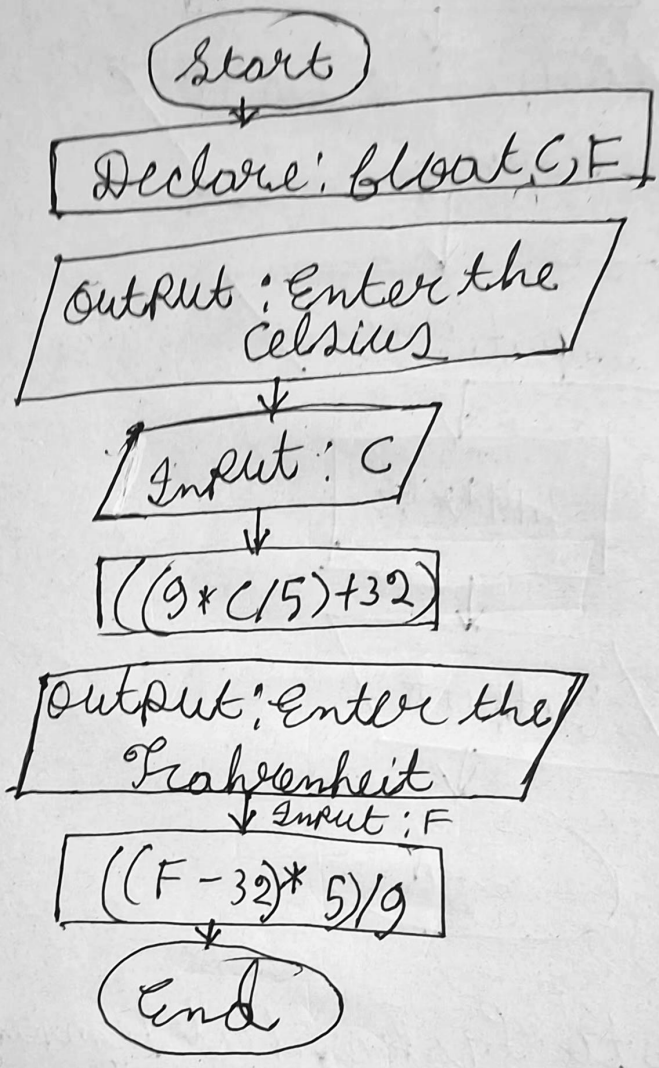


8. Celsius the to Fahrenheit temperature
Fahrenheit to Celsius temperature
algoritham:

Step 1: Start
Step 2: Declare: float C, F
Step 3: Output: Enter the celsius
Step 4: Input C
Step 5: (9* C/5)+32
Step 6: Output: Enter Fahrenheit
Step 7: Input: F

Step8: (CF-32)*5)/9)

Step9: End



9. Swap two number by 3rd variable

Algoritham:

Step 1: Start

Step 2: Declare variable a, b, c.

Step 3: c=a (Store first number in c)

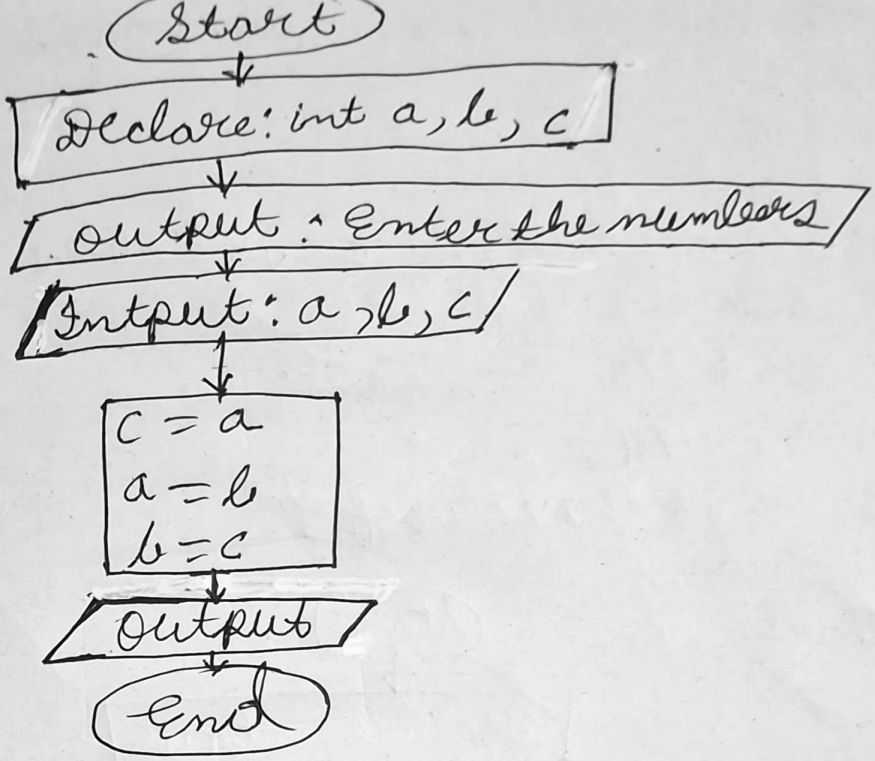
Step 4: a=b (assign second number to first)

Step 5: b=c (assign c to second).

Step 6: Print a and b

Step 7: End

Flowchart:

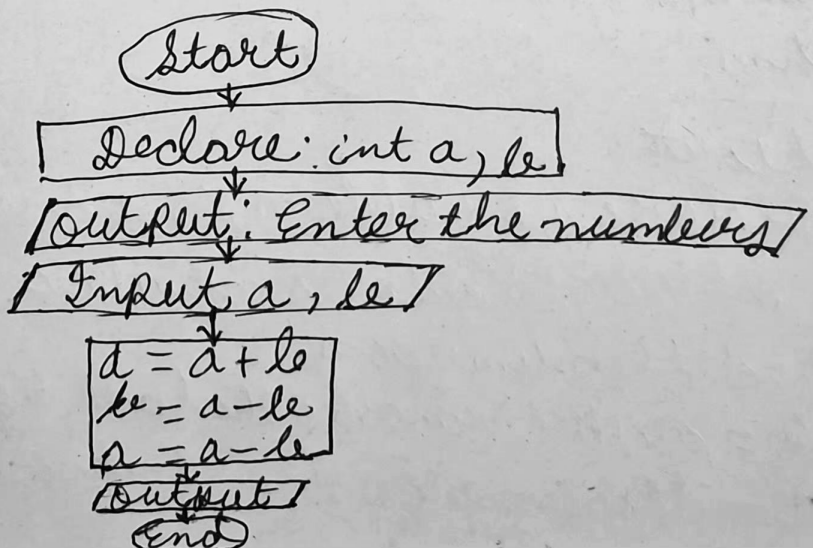


10. Swap two numbers without using a third variable

Algorithm:

Step 1: Start
Step 2: Initialize a and b.
Step 3: a = a+b
Step 4: b = a-b
Step 5: a = a-b
Step 6: Print a and b.
Step 7: End

Flowchart:

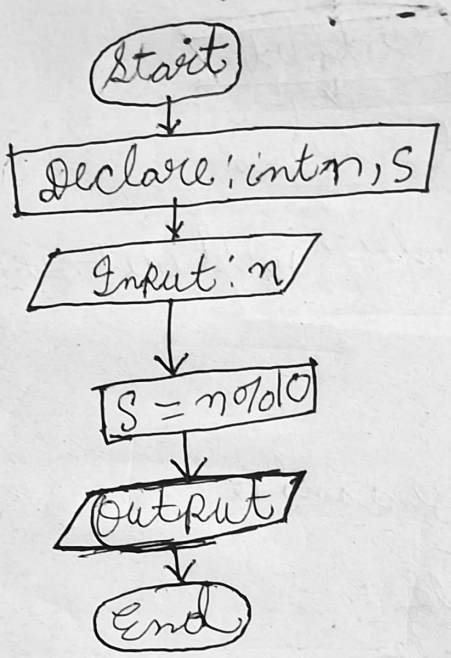


11.(A). The last digit of a integer with modulus operator.

Algorithm:

Step 1: Start
Step 2: Declare two integer variables n and S.
Step 3: Ask the user ~~to two~~ to enter a number.
Step 4: Read the number n.
Step 5: S = n%10
Step 6: Print the value of S
Step 7: End.

Flowchart:



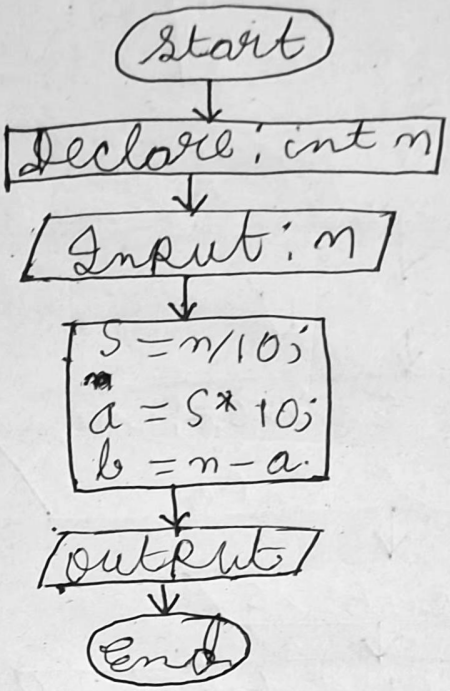
11.(B) Find out the last digit of a int without modulus operator.

Algorithm:

Step 1: Start
Step 2: Declare variable n, S, a and le.
Step 3: Input user to enter a number.
Step 4: Read the number n.
Step 5: S = n/10 (removes the last digit)
Step 6: a = S*10 (results the number without the last digit)

Step 7: b = n - a (this gives the last digit)
Step 8: Print b as the last digit.
Step 9: End.

Flowe chart:



12. Find out compound interest.

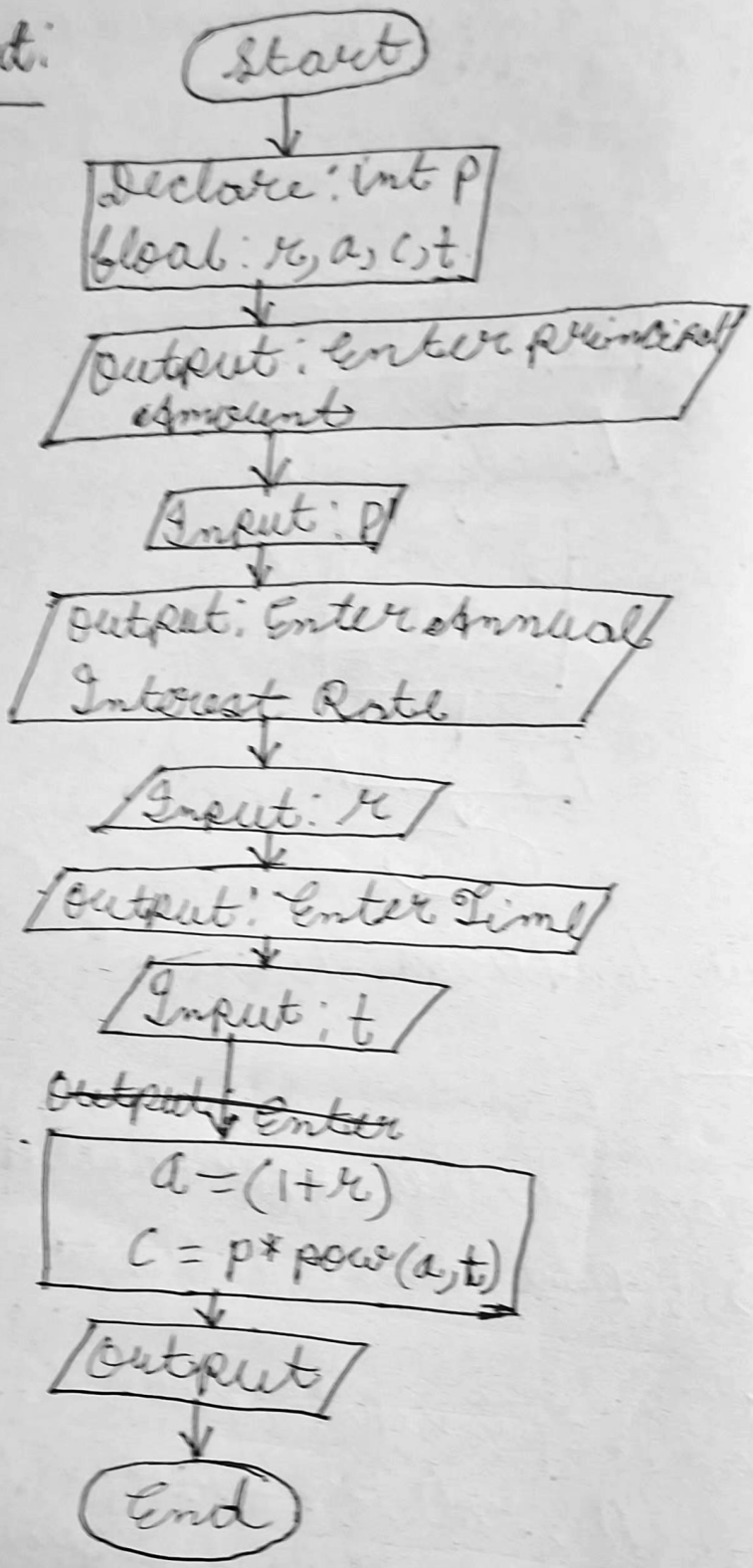
Algorithem:

Step 1: Start
Step 2: Declar variables P (principal amount), r (Rate of interest), t (Time period), a (calculation & C (compound interest).
Step 3: Read the principal amount (P) from the user.
Step 4: Read the rate of interest (r) from the user.
Step 5: Read the thime period (t) from the user.
Step 6: Calculate

$$a = 1 + r$$
$$C = P \times (a)^t$$

Step 7: Display the compound amount C.
Step 8: End

Flow chart:

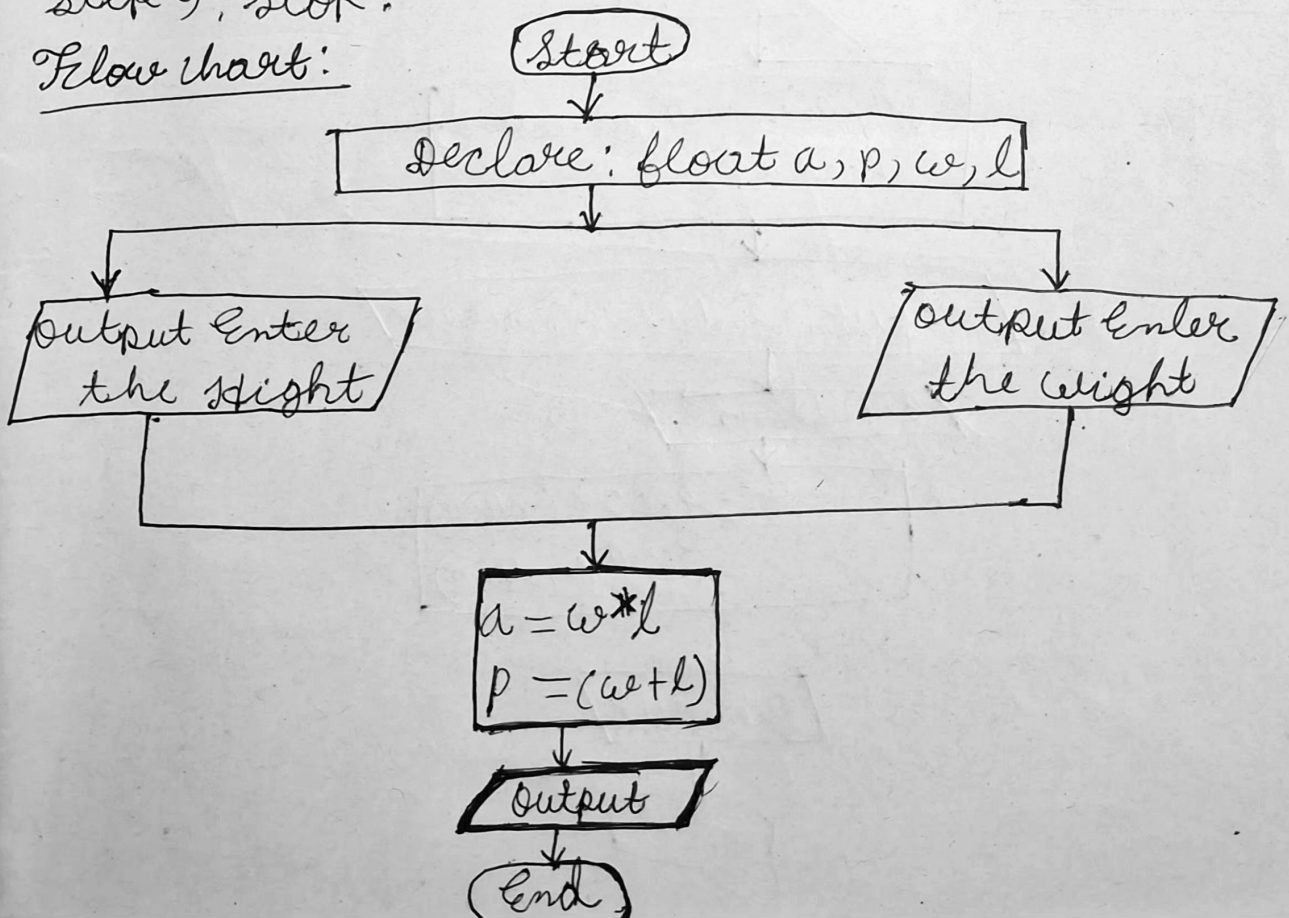

Start
Declare: int P
float: r, a, c, t
Output: Enter Principal amount
Input: P
Output: Enter annual Interest Rate
Input: r
Output: Enter Time
Input: t
Output: Enter
a = (1+r)
c = p * pow(a, t)
Output
End

13. Calculate the area and perimeter of the rectangel.

Algorithm:

Step 1: Start
Step 2: Declare variables of type float: a (area), P (perimeter), w (width) and l (length).
Step 3: Input width Display a message: "Enter the width:" Read the value of w.
Step 4: Input length Display message: "Enter the length:" Read the value of l.
Step 5: calculate area a = w*l; ~~Calculate area a = w*l Display area~~.
Step 6: Print: "The area of the rectangle = a" //Display area
Step 7: Calculate perimeter P = 2*(w+l).
Step 8: Display perimeter/output perimeter.
Step 9: Stop.

Flow chart:



14. Find out floor and ceiling value of a given number which can be positive or negative

Algorithm:

Step1: Start

Step2: Declare three variables of type double num(input number), f(floor value and c (ceiling va

Step3: Input the number.
Read the value of num.

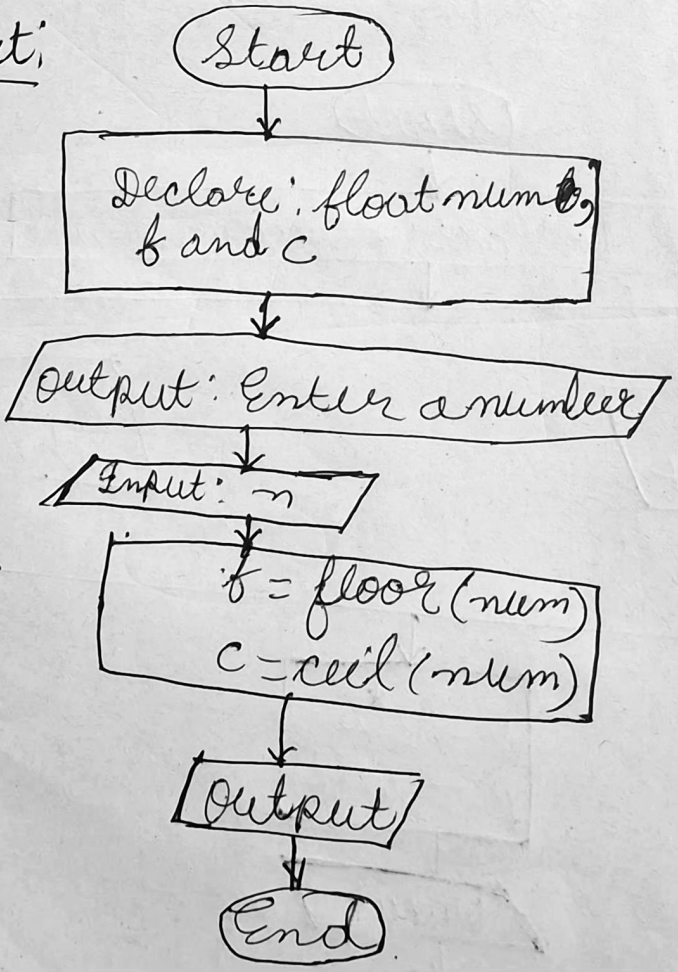
Step4: f = floor(num).

Step5: c = ceil(num).

Step6: Output the result.

Step7: Stop.

Flowchart:



15. Find out the roots of a quadratic equation

algorithm:

Step 1: Start

Step 2: Declare six variables ~~not~~ of type float: a, b, c, d, root1 & root2.

Step 3: Input coefficients.

Step 4: Calculate discriminant d = b*b - 4*a*c

Step 5: Calculate the roots

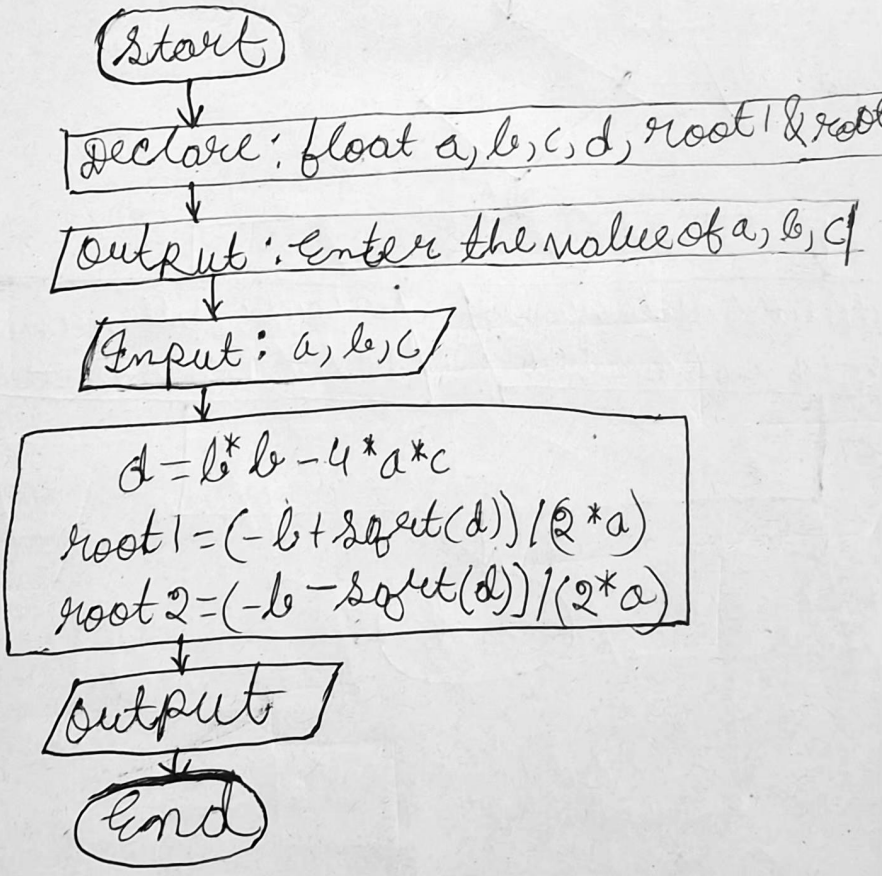
root1 = (-b + sqrt(d)) / (2*a)

root2 = (-b - sqrt(d)) / (2*a)

Step 6: Display the results.

Step 7: Stop.

Flowchart:

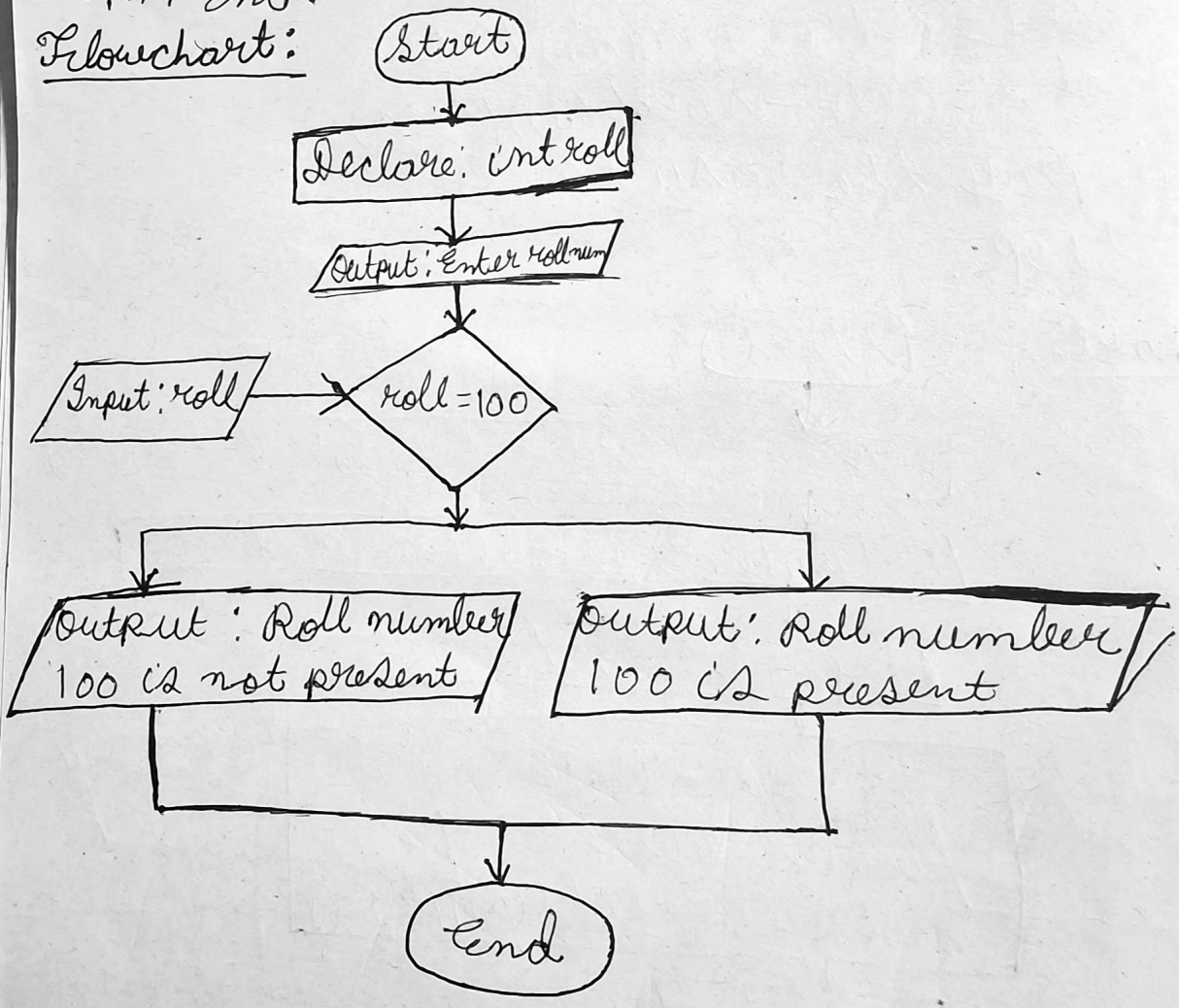


16. Check whether roll 100 is present or not (using if else)

algoritham:

Step 1: Start
Step 2: Declare an integer variable roll.
Step 3: Ask the user to enter the roll number
Step 4: Read the value of roll.
Step 5: If roll == 100, then Display "Roll 100 is present"
Step 6: If roll != 100, then Display "Roll 100 is not present"
Step 7: End.

Flowchart:

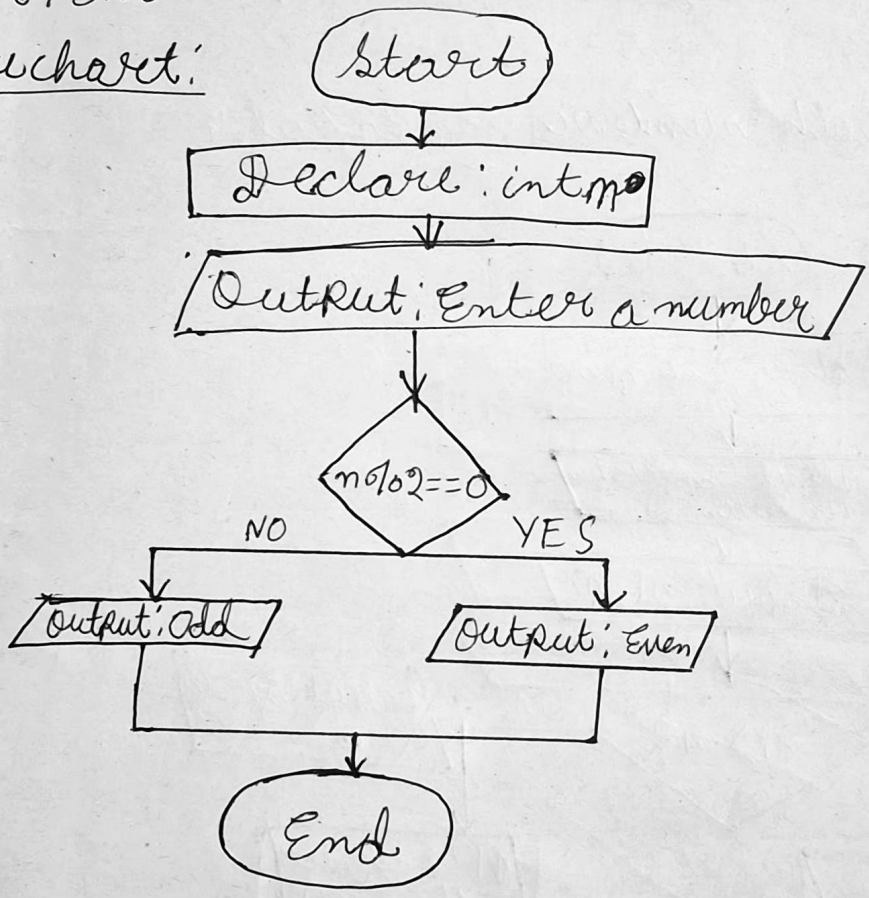


17. Check whether a int is odd or even.

Algorethan:-

Step 1: Start
Step 2: Declare an integer variable n.
Step 3: Ask the user to enter an intheger number.
Step 4: Read the value of n.
Step 5: check if n%02==0
If true, print "n is Even"
Else, print "n is odd"
Step 6: End.

Flowchart:



15. Check the greatest among two number.

Algoritham:

Step 1: Start

Step 2: Declare two integer variables: n1, n2.

Step 3: Ask the user to enter the numbers.

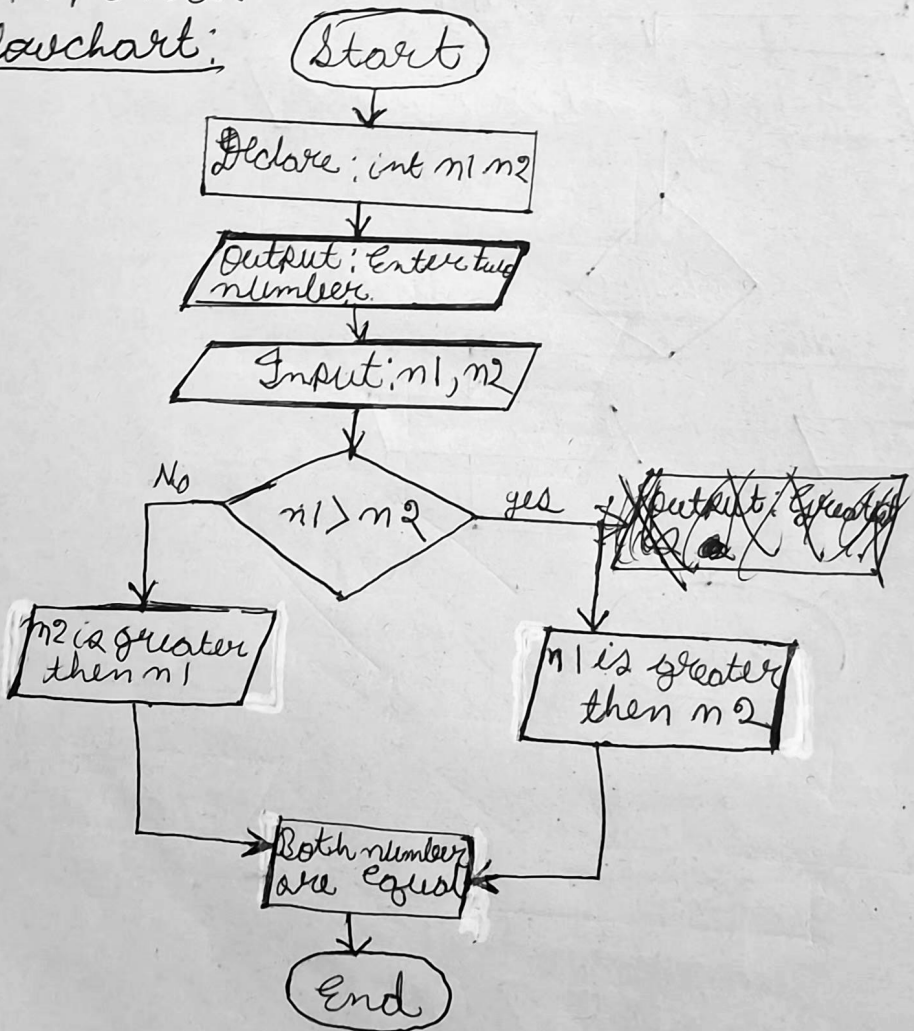
Step 4: If n1 > n2, then
Print "n1 is greater then n2".

Step 5: Else if n2 > n1, then
Print "n2 is greater then n1".

Step 6: Else
Print "Both numbers are equal."

Step 7: End.

Flowchart:

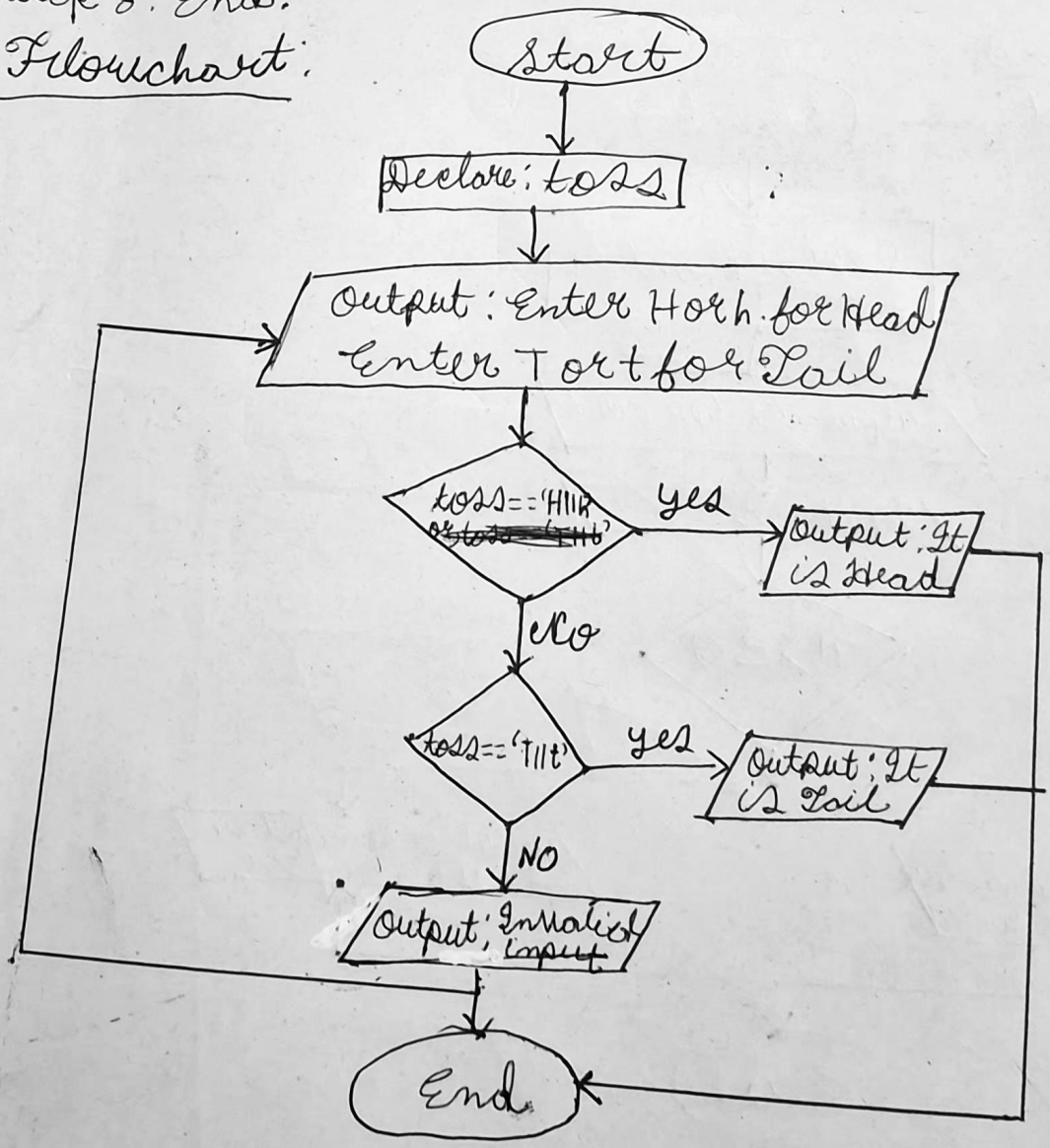


19. A coin has been tossed once. Write a C program to check whether it is Head or Tail.

Algoritham:

Step 1: Start
Step 2: Declare a character variable toss.
Step 3: Ask the user to enter H for Head or T for Tail.
Step 4: Read the input character toss.
Step 5: If toss == 'H' or toss == 'h', Print "It is Head".
Step 6: Else if toss == 'T' or toss == 't', Print "It is Tail".
Step 7: Else Print "Invalid input! Please Enter H or T only".
Step 8: End.

Flowchart:



20. Check wether it a number is whole number or not.

Algoritham :-

Step 1: Start

Step 2: Declare an integer variable n.

Step 3: Ask the user to enter a number

Step 4: Read the number.

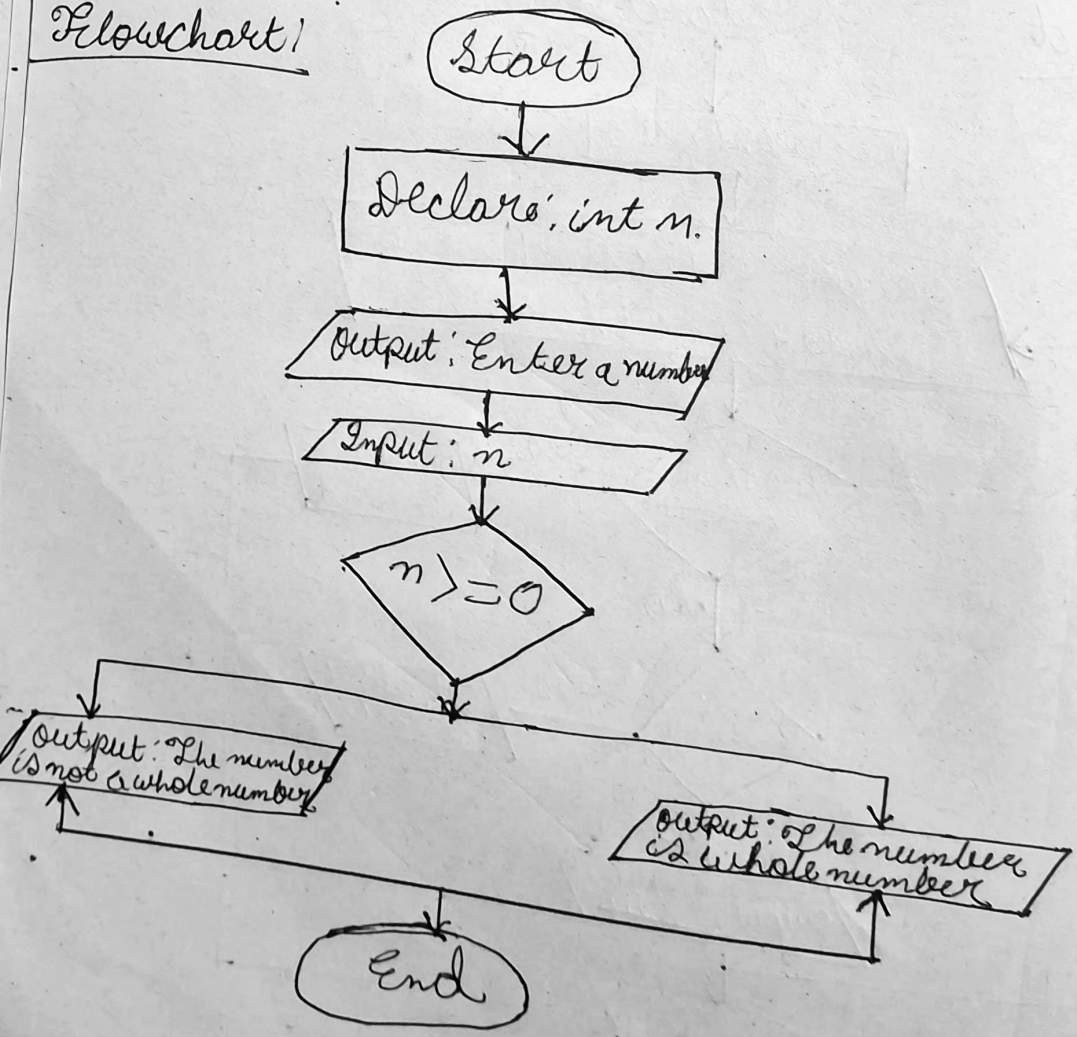
Step 5: check if n >= 0

→ If true, print: "n is a whole Number"

→ ~~If~~ Else, print "n is Not a whole Number"

Step 6: End.

Flowchart:



21. Check whether it is Leap year or not.

Algoritham:

Step 1: Start

Step 2: Declare an integer variable year.

Step 3: Prompt the user to enter a year.

Step 4: Read the value of year.

Step 5: If year %400 == 0, then
Print "year is a Leap Year".

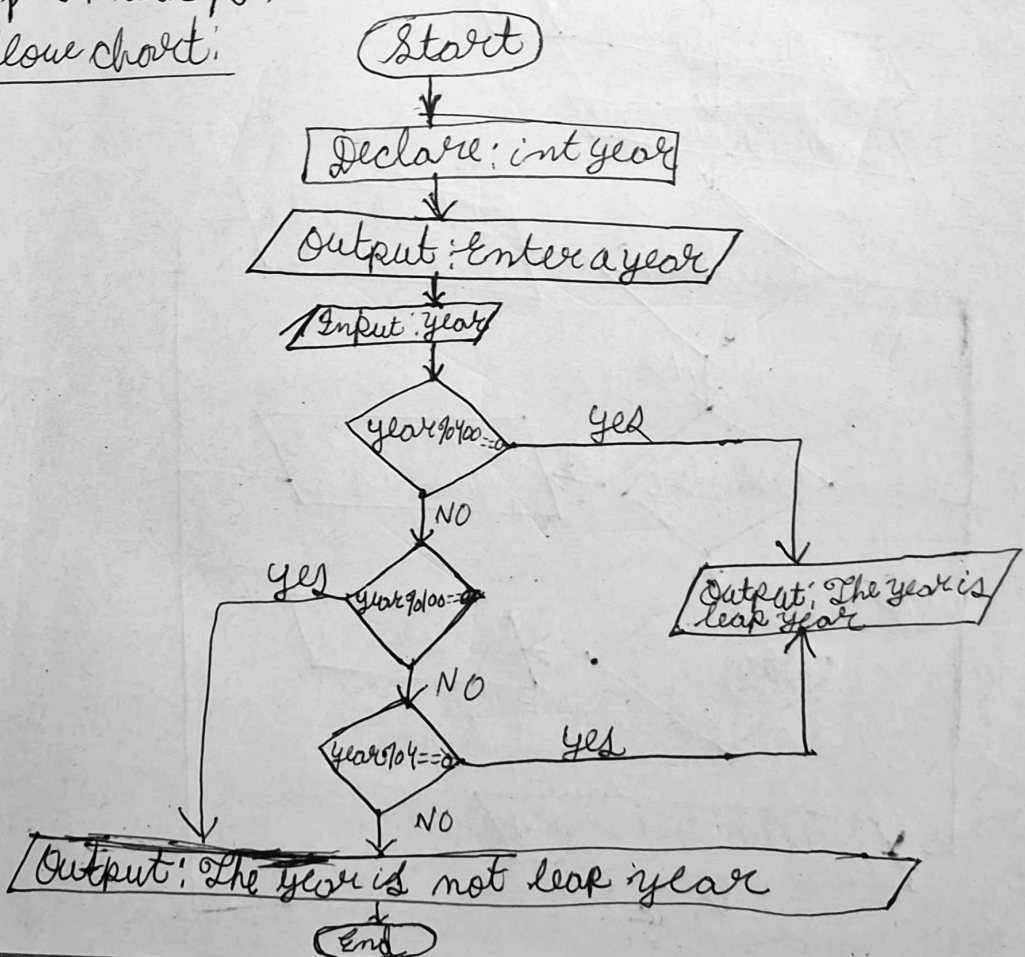
Step 6: Else if year %100 == , then
Print "year is not a Leap year".

Step 7: Else if year %04 == 0, then
Print, "year is a ~~Lep~~ Leap year".

Step 8: Else Print "year is Not a Leap Year".

Step 9: Stop.

Flowchart:

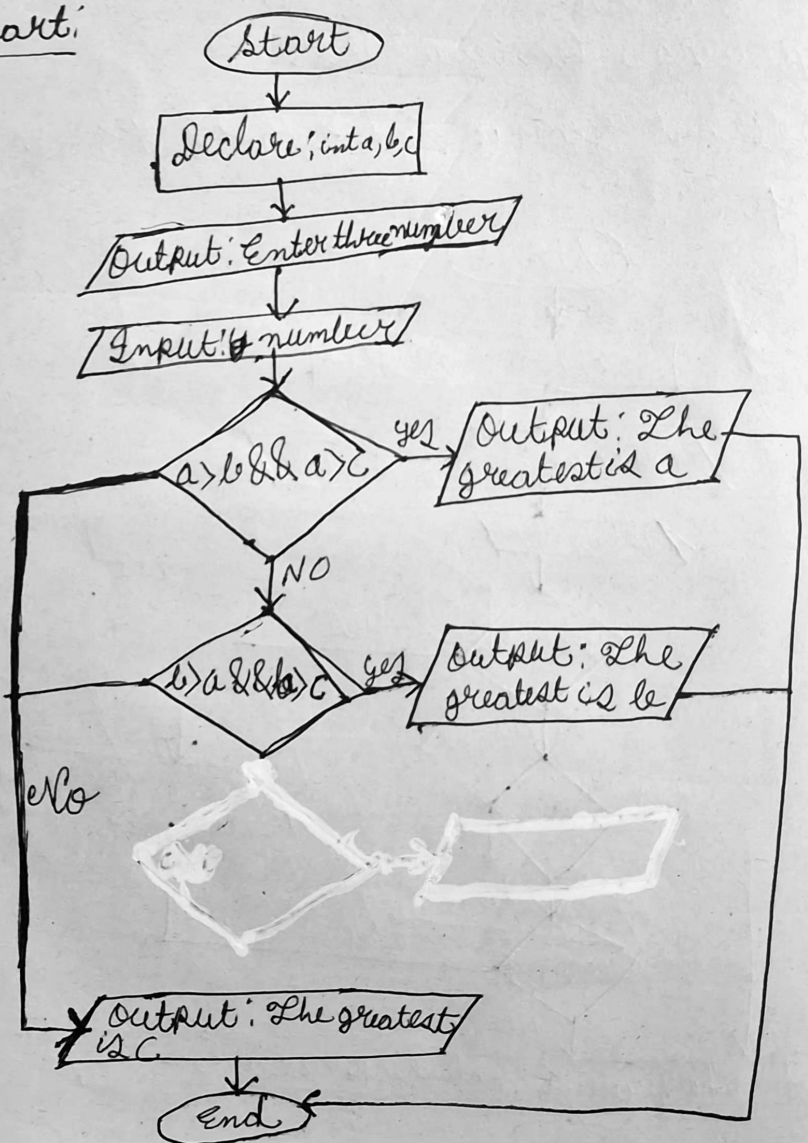


29. Check maximum among three numbers. using ladder if else.

Algorithm:

Step 1: Start
Step 2: Declare three integer variables a, b and c.
Step 3: Read three numbers a, b and c from the user.
Step 4: If (a>=b && a>=c)
Print "a is the greatest".
Step 5: Else if (b>=a && b>=c)
Print "b is the greatest."
Step 6: Else Print "c is the greatest".
Step 7: Stop.

Flowchart:



23. Check if a letter is vowel or consonant using ladder. If the

Algorithm:

Step 1: Start

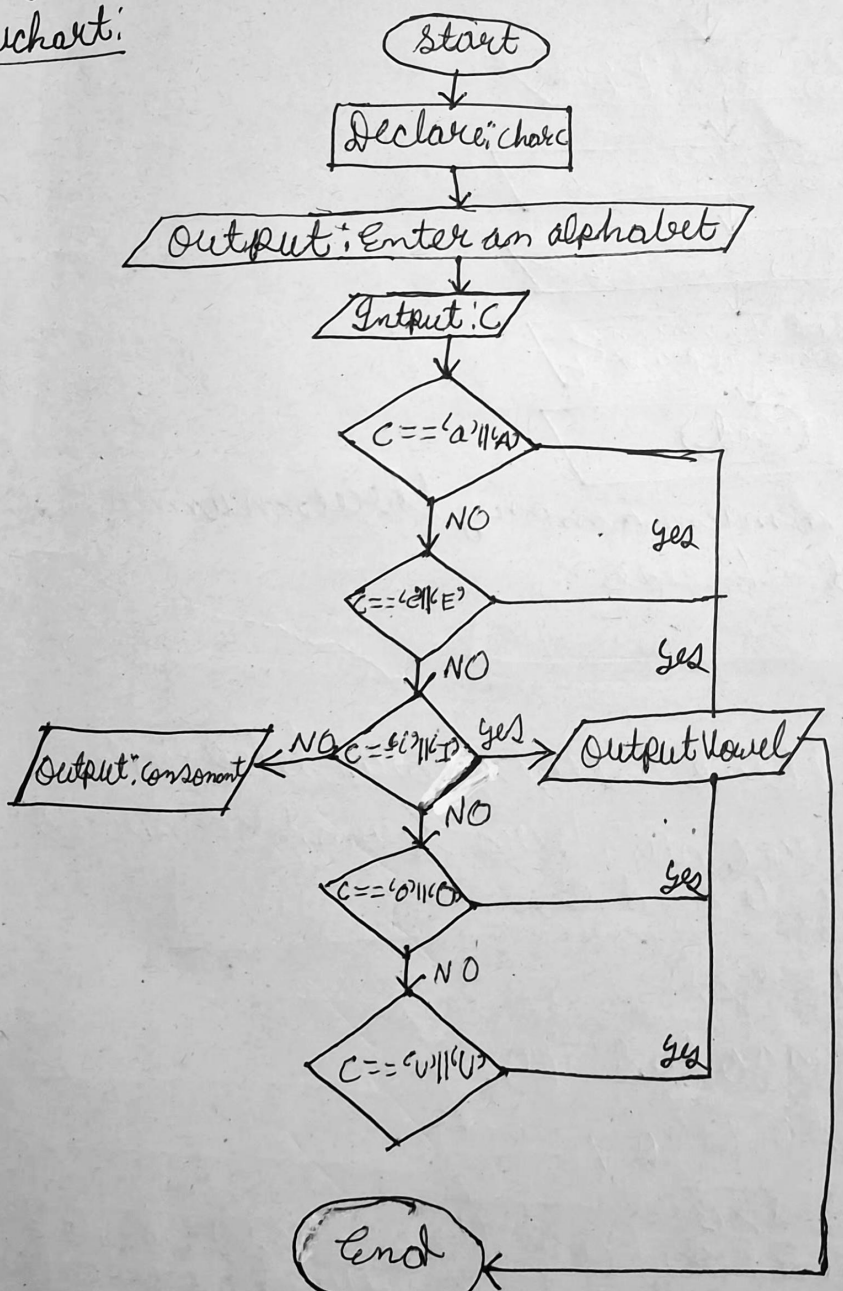
Step 2: Read a character c

Step 3: If C is A, E, I, O or U (uppercase or lowercase) → Print "vowel".

Step 4: Else → Print "Consonant"

Step 5: Stop.

Flowchart:

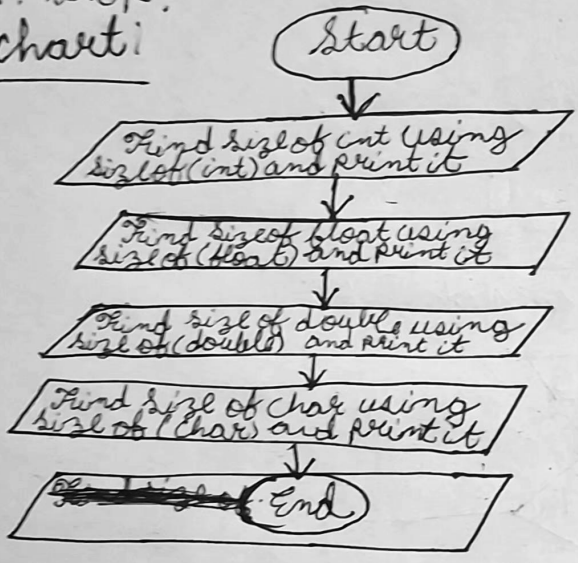


24. Find the size of integer, float, double & char.

Algoritham:

Step 1: Start
Step 2: Use sizeof() to find sizes of int, float, double and char.
Step 3: Print the sizes using printf().
Step 4: Stop.

Flowchart:



25. Check maximum among three numbers using nested-if-else.

Algoritham:

Step 1: Start
Step 2: Declare three integer variables - a, b, c.
Step 3: Read three numbers from the user and store them in a, b and c.
Step 4: Check if a > b
If true, then go to Step 5.
If false, go to Step 7
Step 5: Check if a > c
If true, then print "a is the greatest" and go to Step 9.
If false, then print "c is the greatest" and go to Step 9.

26. Create a simple calculator using switch-case. The program should take number inputs and operators (+, -, *, /) from the user and display the result.

Algorithm:

Step 1: Start

Step 2: Input an operator (+, -, *, /).

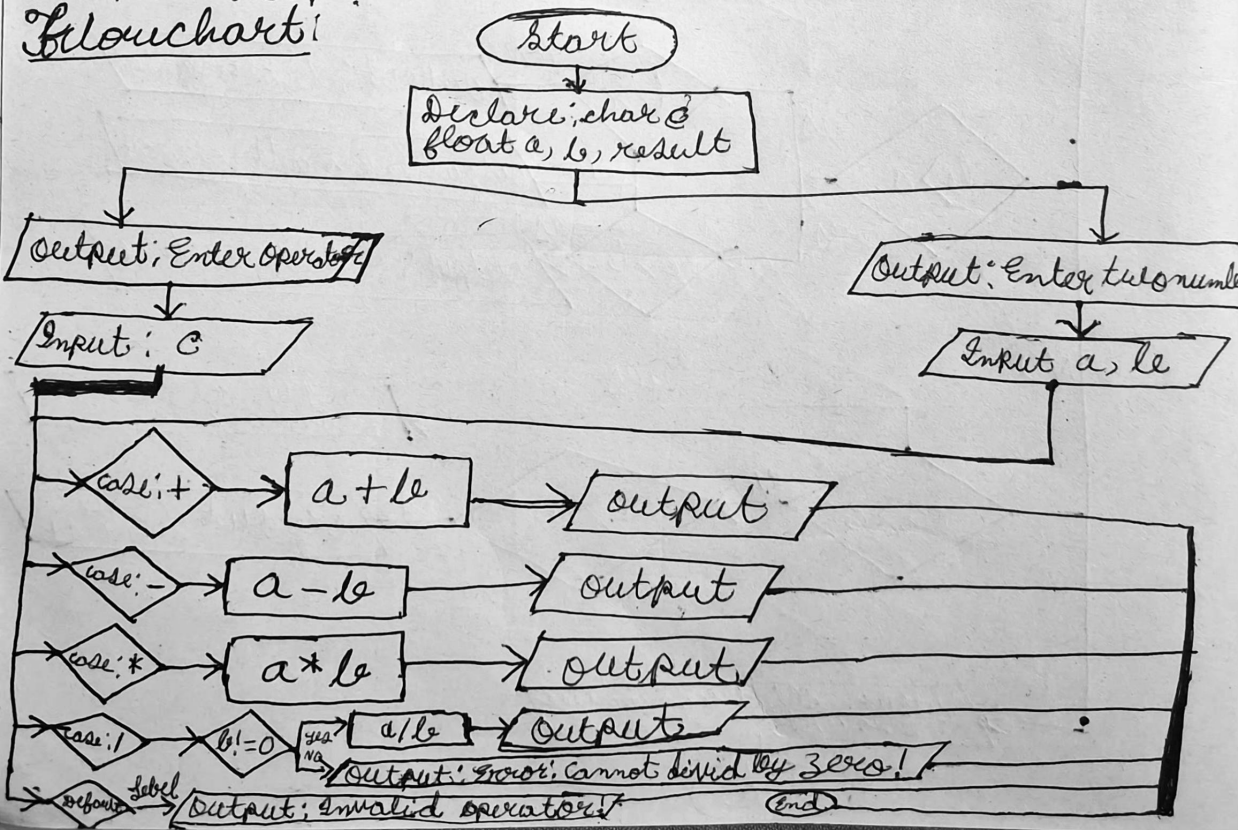
Step 3: Input two numbers — num1 and num2.

Step 4: Use switch statement based on the operator c:

- Case '+' → result = num1 + num2 → Print result.
- Case '-' → result = num1 - num2 → Print result.
- case '*' → result = num1 * num2 → Print result.
- case '/' →
  - If num2 != 0 → result = num1 / num2 → Print r
  - Else → Print "Error! Division by zero".
- Default → Print "Invalid operator!"

Step 5: Stop.

Flowchart:



Step 6: (Skipped if Step 4 is true)

Step 7: check if b>c

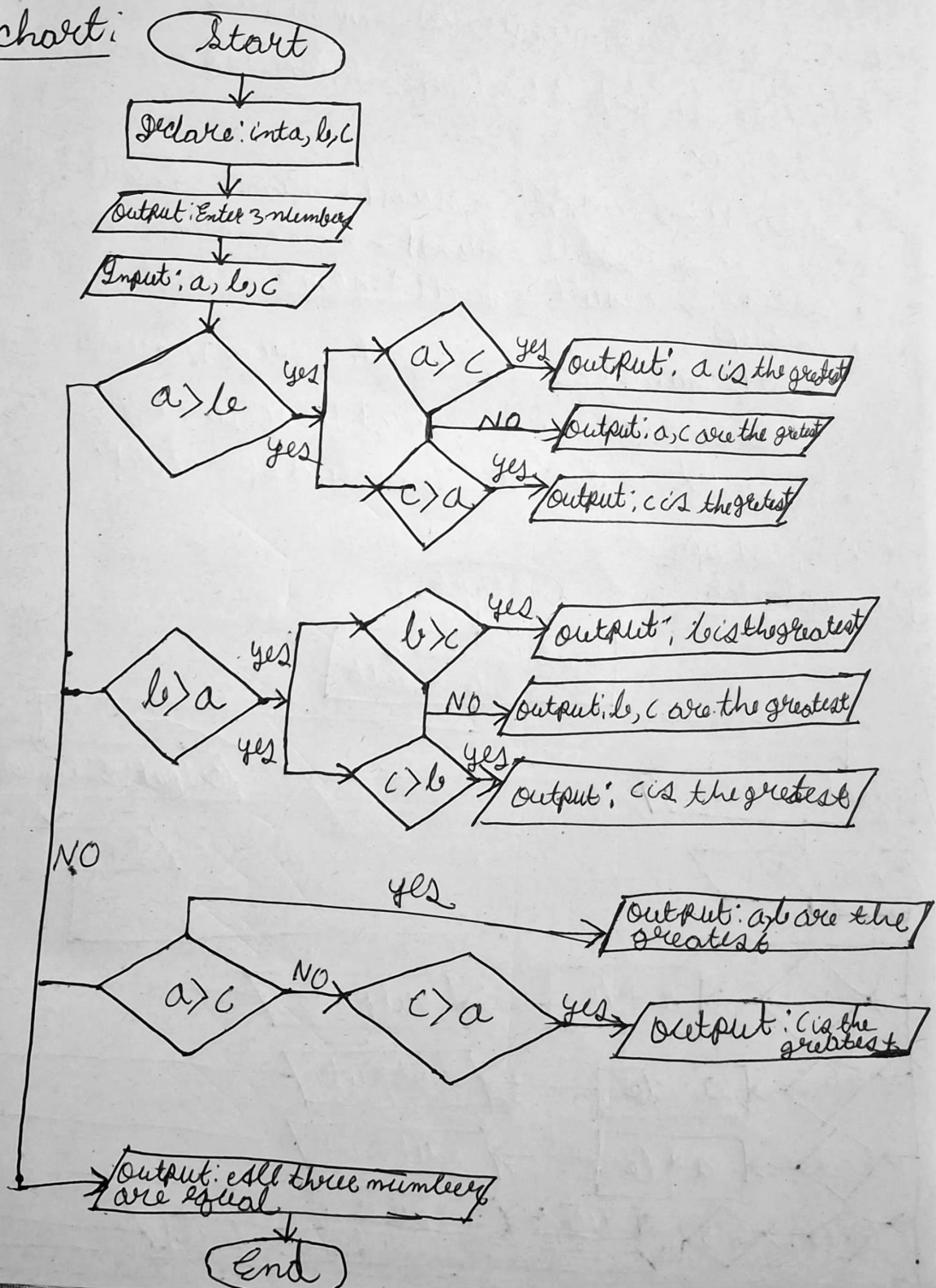
If true, then print "b is the greatest" and go to Step 9

If false, then print "c is the greatest".

Step 8: (All comparisons done.)

Step 9: Stop.

Flowchart:



27. Using Switch-case to check whether a given alphabet is a vowel or consonant.

Algorithm:

Step1:- Start

Step2: Read a character n

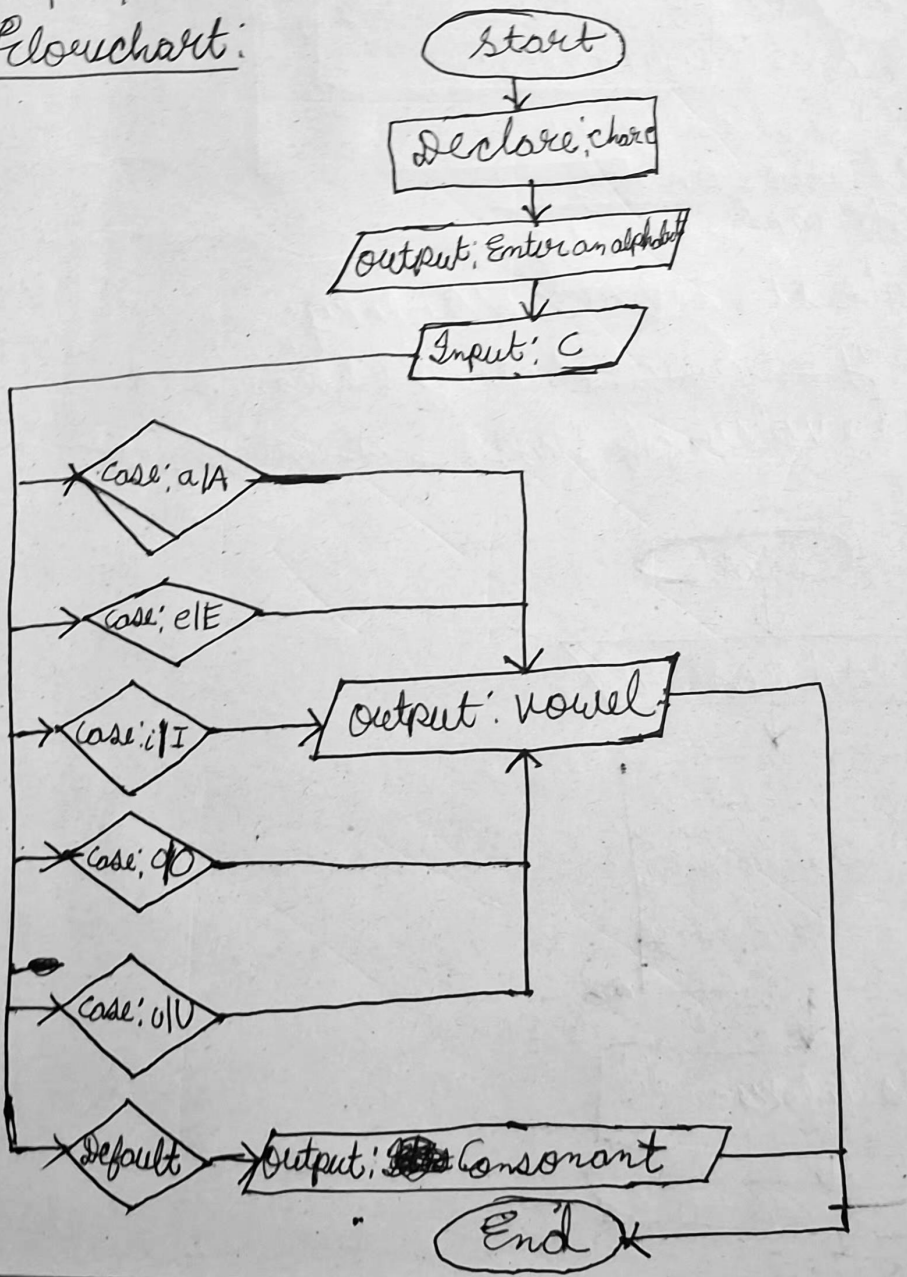
Step3: use switch(n)

Step4: If n = 'A', 'E', 'I', 'O', 'U' or 'a', 'e', 'i', 'o', 'u' -> Print "Vowel".

Step5: default -> Print "Consonant"

Step6: Stop.

Flowchart:



23. Calculates the total salary of an employee

Input: Basic Salary

Calculate: HRA = 20% of Basic
TA = 10% of Basic
DA = 5% of Basic

Algoritham:

Step1: Start

Step2: Declare variables: All float basic_salary, hra, ta, da and gross_salary.

Step3: Read the value of basic_salary from the user

Step4: Calculate the allowances:

hra: 0.20 * basic_salary
ta: 0.10 * basic_salary
da: 0.05 * basic_salary

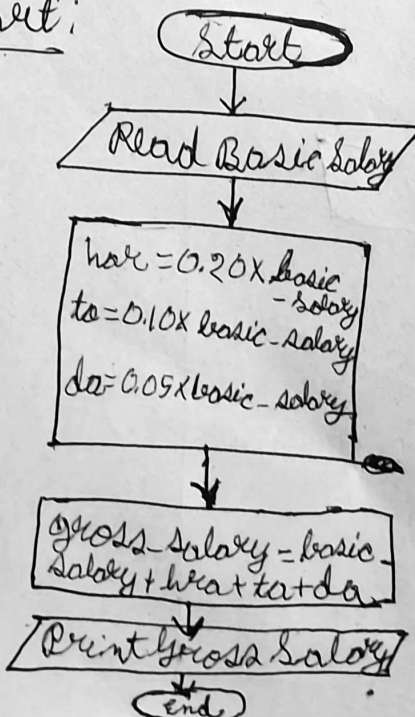
Step5: Calculate the total ~~so~~ salary:

gross_salary = basic_salary + hra + ta + da

Step6: Print the value of gross_salary

Step7: Stop

Flowchart:



29. Switch-case to calculate the electricity bill based on units consumed.

For first 100 units → 2 5/unit

Next 100 units → 2 7/unit

Above 200 units → 2 10/unit

Algorithm:

Step 1: Start

Step 2: Declare variables units (int) and bill (float).

Step 3: Input the number of units consumed.

Step 4: use switch (units <=100)

Case 1: If units <=100

→ bill = unit * 5

Case 0: If units >100

use another switch (units <=200)

Case 1; If unit <=200

→ bill = (100*5) + (units-100)*7)

Case 0: If units >200

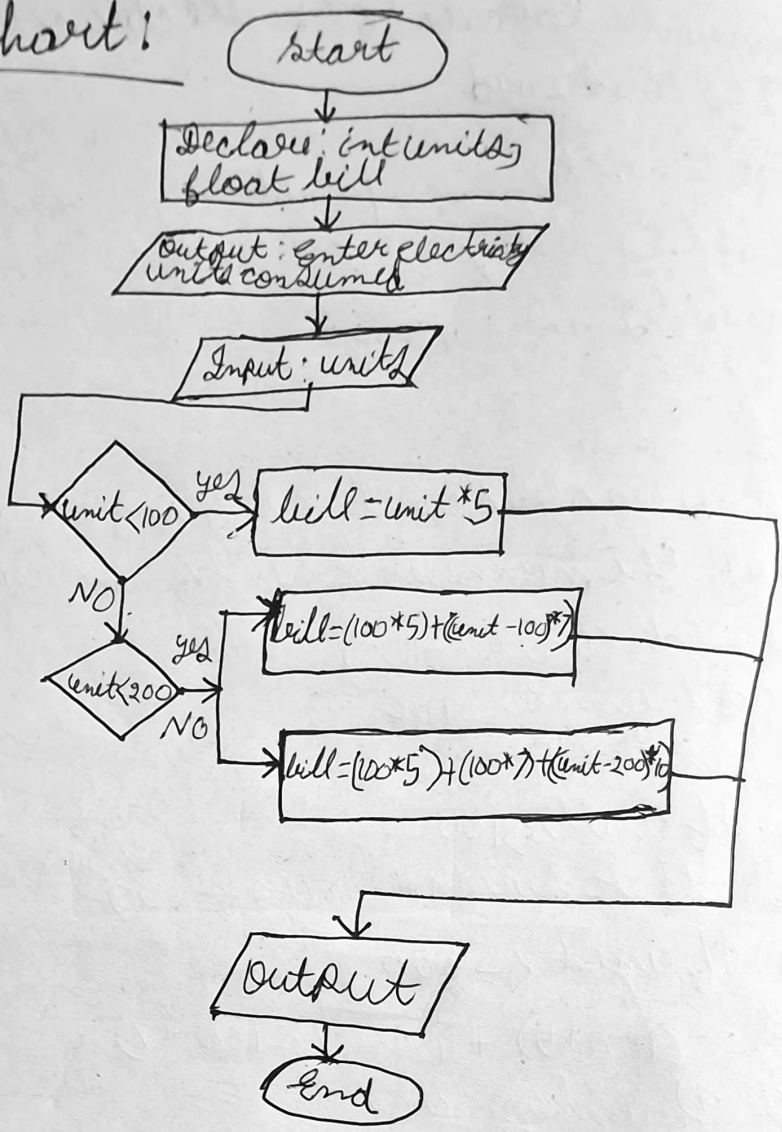
→ bill = (100*5) + (100*7) + ((units-200)*10)

Step 5: Print the total bill amount

Step 6: Stop.

Flowchart:

Flow chart:



30. Calculate the grade of a student based on their marks.

Step 1: Start

Step 2: Declare integer variables m, e, l, a, c and avg.

Step 3: Input marks of five subjects: Maths, English, Life Science, Applied Science and Computer Science.

Step 4: Calculate average:

ave = (m + e + l + a + c) / 5

Step 5: If avg >= 90 → Print "grade O"

Elseif avg >= 0 → Print "grade A+"

Else if avg >= 70 → Print "Grade A"

Else if avg >= 60 → Print "Grade B+"

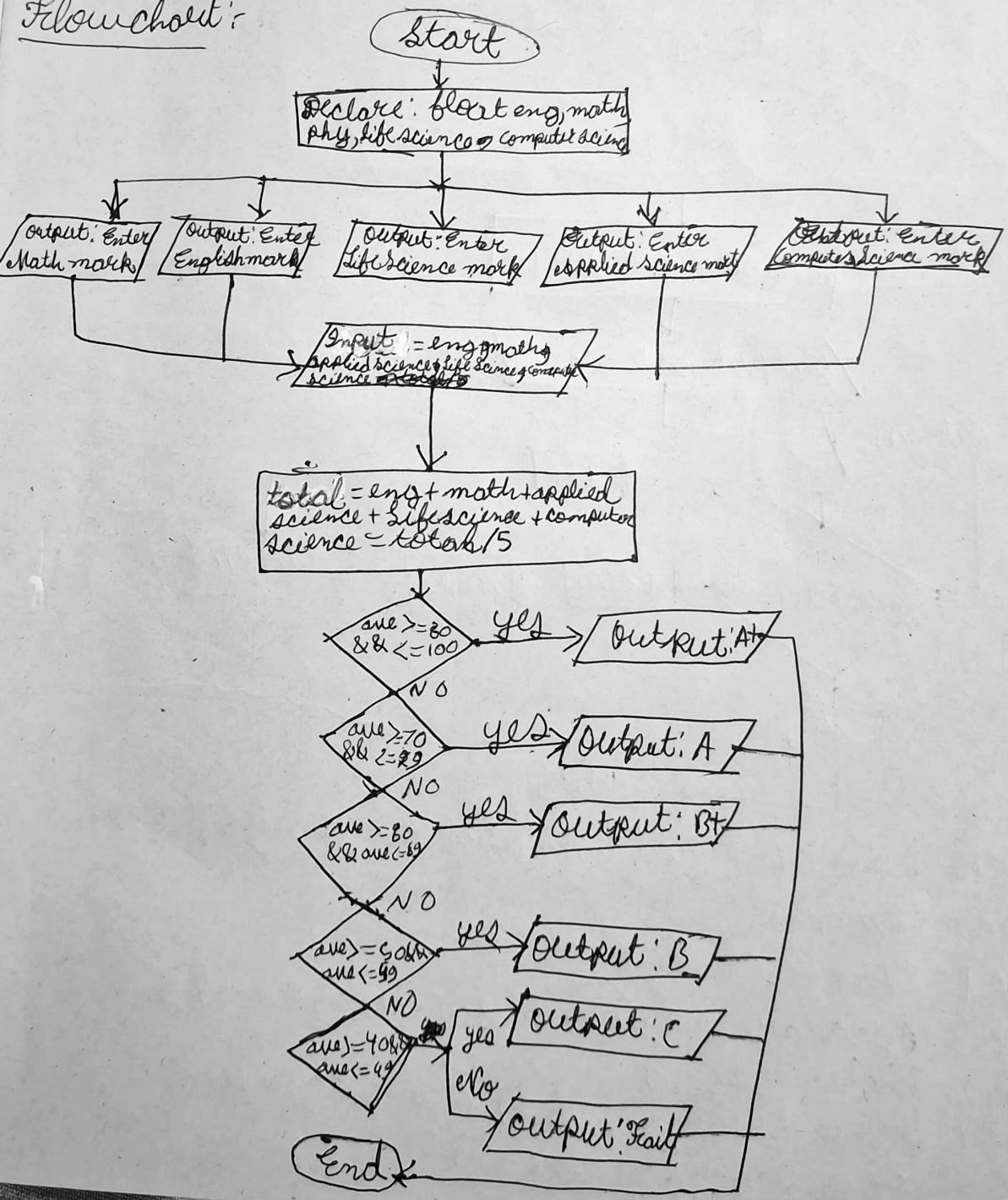
Else if avg >= 50 → Print "Grade B"

Else if avg >= 40 → Print "Grade C"

Else → Print "Fail".

Step 6: Stop.

Flowchart:

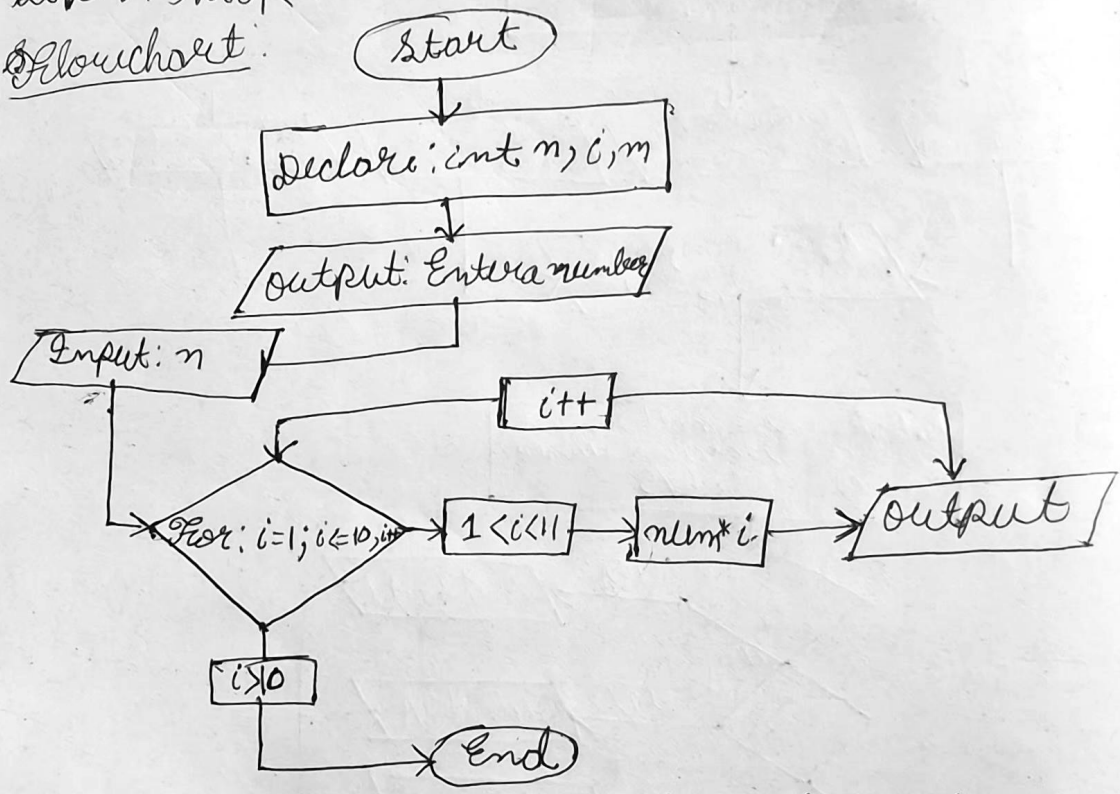


1. Multiplication table with using loop.

Algorithm:-

Step 1: Start
Step 2: Declare variables → a, i and m as integers
Step 3: Display message → "Enter the value:"
Step 4: Read the input number and store it on variable n.
Step 5: Set i = 1
Step 6: Repeat step 7 & 8 while i <= 10
Step 7: m = n × i
Step 8: Print n * i = m
Step 9: Stop.

Flowchart:



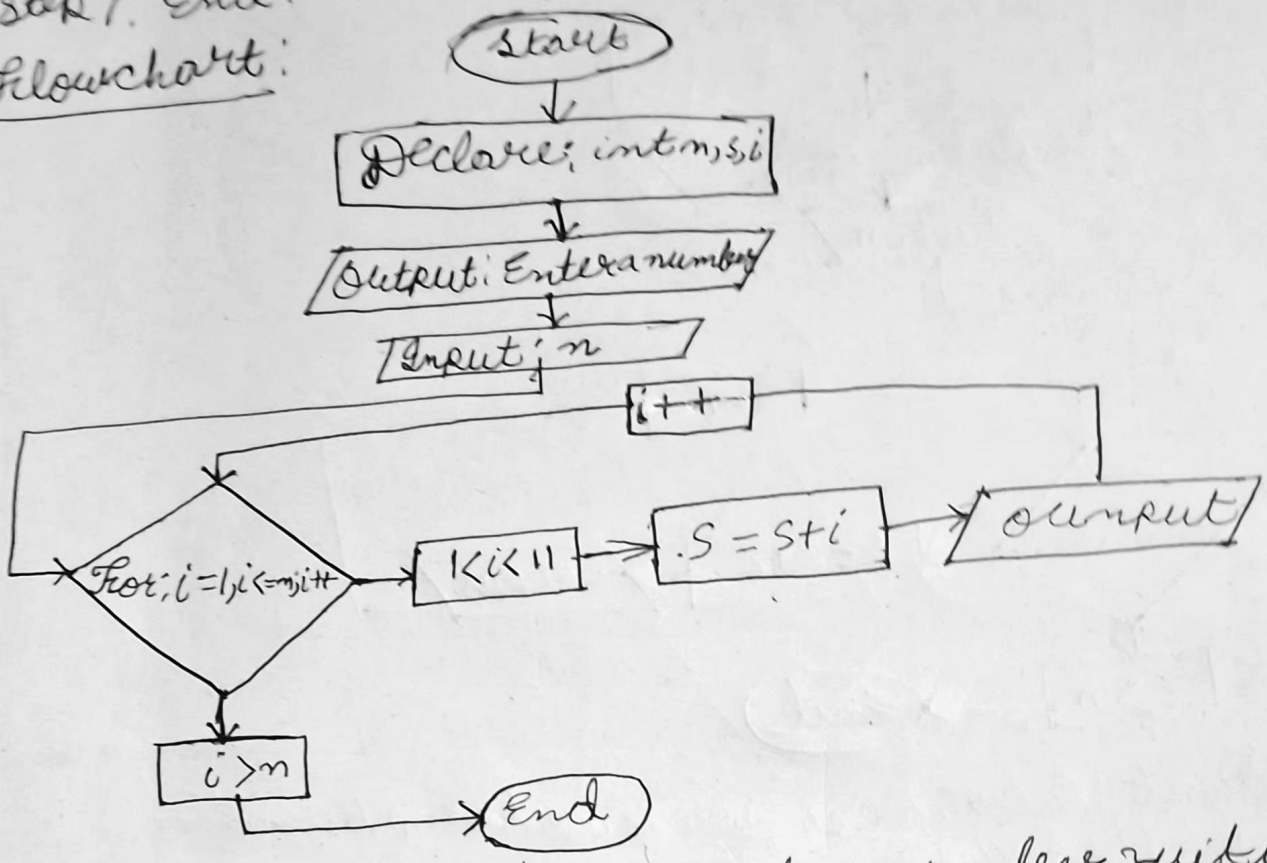
2. The sum of first (n) natural number (using loop)

Algorithm:

Step 1: Start
Step 2: Declare int variables n, i and s.
Step 3: Initialize s = 0
Step 4: Input the value of n from the user.
Step 5: Loop from i = 1 to i = n
- Add i to s (s = s + i).

Step6: After the loop, print the value of S.
Step7: End.

Flowchart:

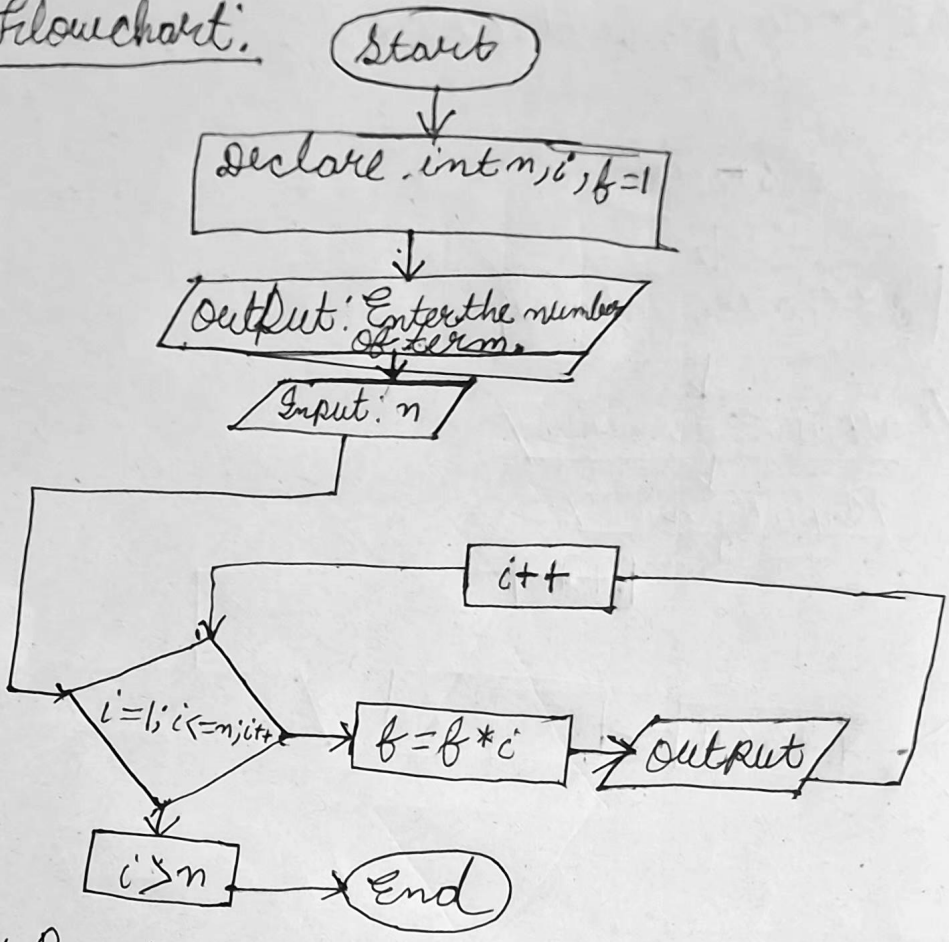


33. Find out factorial value of a number with using loop.

Algorithom:-

Step1: Start
Step2: Declare integer variables: n, i and f=1.
Step3: Read the number n from the user.
Step4: use a for loop from i=1 to i<=n;
multiply f by i (f=f*i);
Step5: Print the factorial f.
Step6: End.

Flowchart.



4. Print Fibonacci series up to (n) term and sum of the series.

Algorithm:

Step1: Start

Step2: Declare integer variables: n, i, a=0, b=1, x, sum=0.

Step3: Input the number of terms n from the user.

Step4: Print a message: "The Fibonacci series up to n terms is:"

Step5: Loop from i=1 to i=n;

print a

add a to sum. (sum = sum + a)

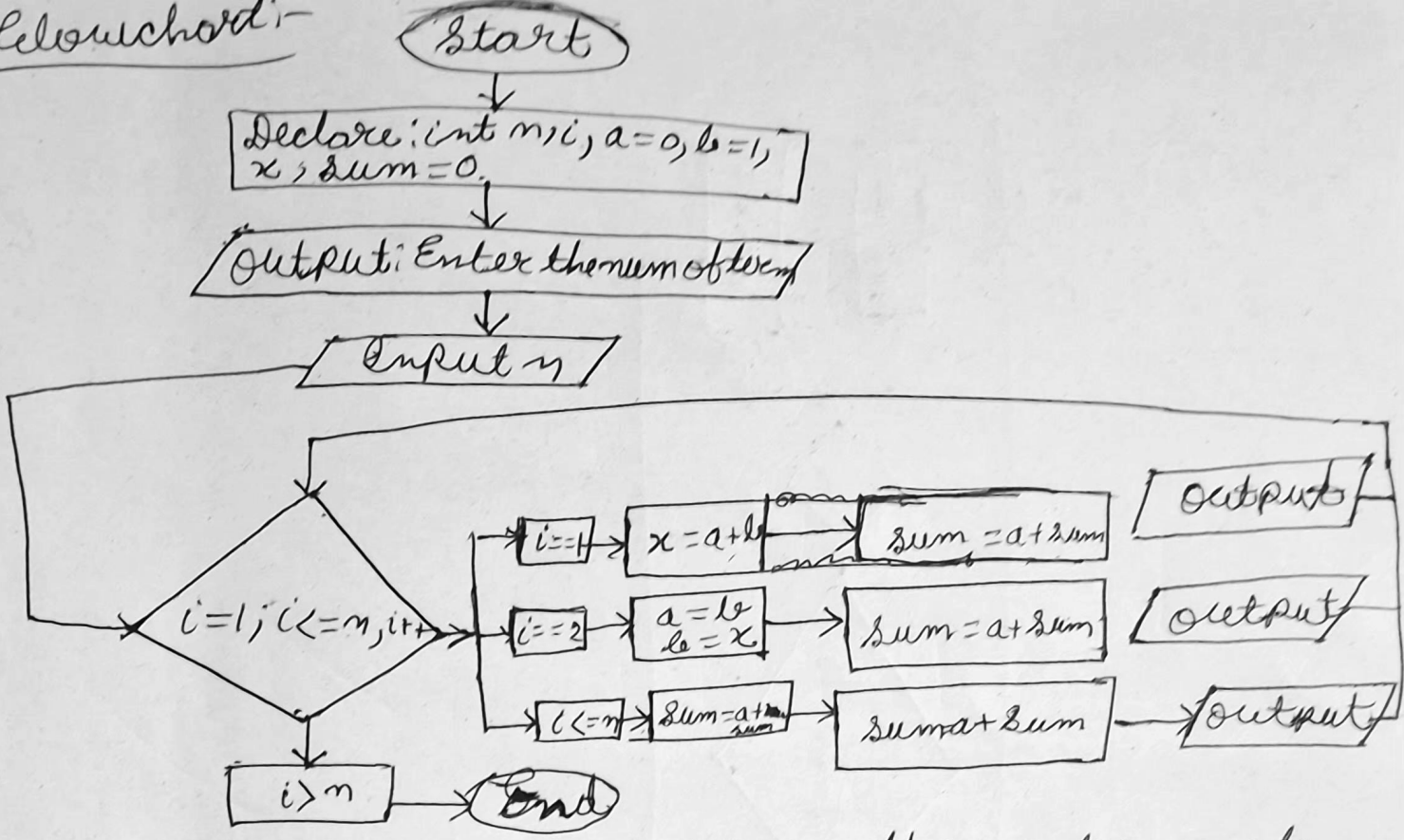
compute next term x = a+b

update a=b and b=x

Step 6: After the loop, print the sum of the series.

Step 7: End.

Flowchart:-



35. Find the prime numbers with in a range of number which give by user.

Algoritham:-

Step 1: Start

Step 2: Input two numbers, start and end

Step 3: If start < 2, set start = 2 (since 2 is the smallest prime)

Step 4: For each number i from start to end

a) Assume is prime (isPrime = 1)

b) For each number j from 2 to i-1

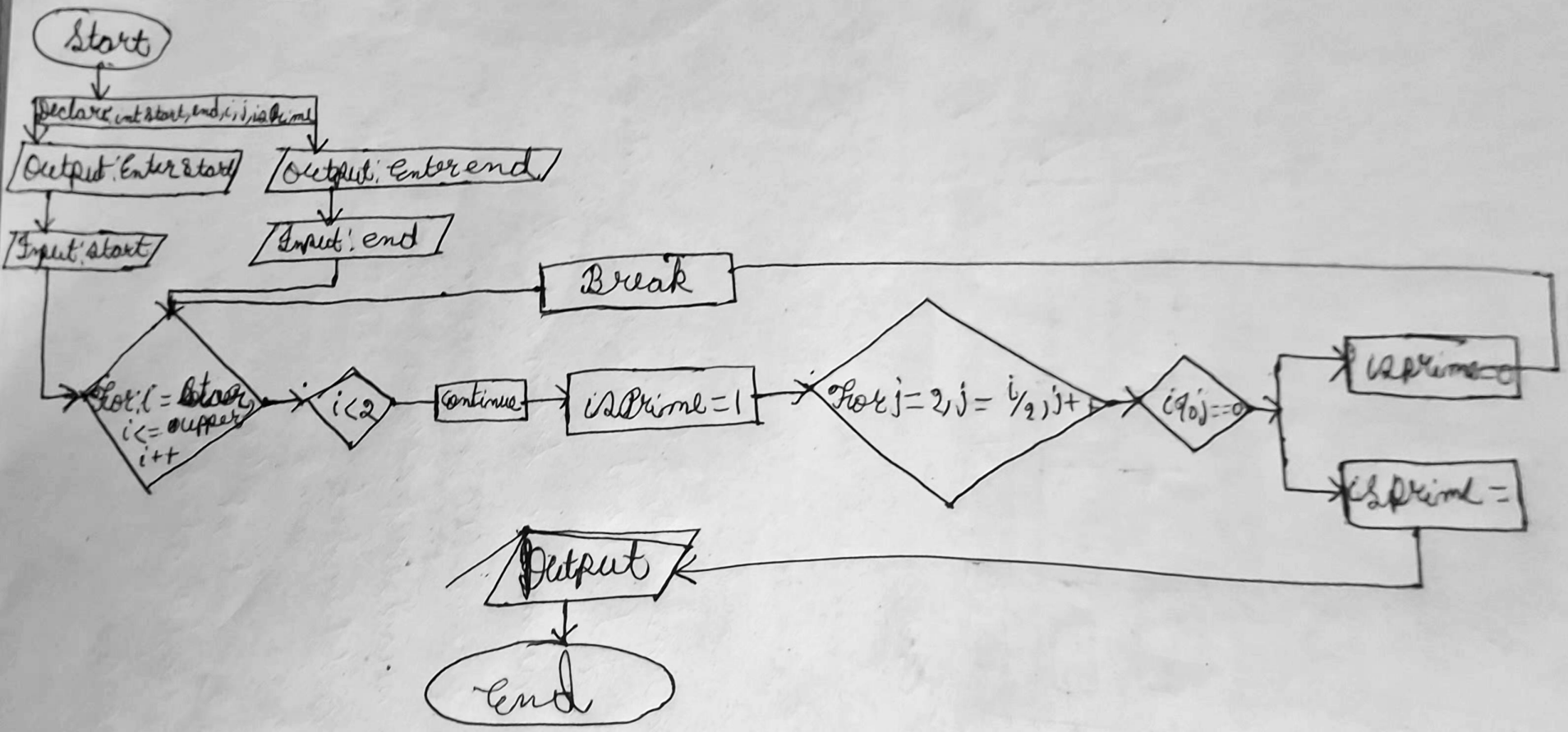
i) If i % j == 0 then set isPrime = 0 and break (not prime)

c) If isPrime == 1, print i (it is prime)

Step 5: Repeat until all numbers from start to end are checked

Step 6: End.

Flowchart:


Start
Declare int start, end, i, j, isPrime
Output: Enter start
Output: Enter end
Input: start
Input: end
Break
for(i = start; i<= upper; i++
i<2
continue
isPrime = 1
for j = 2; j = i/2; j++
i%j == 0
isPrime = 0
isPrime =
Output
End