

L.T.J.S.S's

LOKMANYA TILAK COLLEGE OF ENGINEERING

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.



FIRST YEAR ENGINEERING



NAME: SINGH SUDHAM DHARMENDRA

ROLL NO./DIV.: AIMLD50 / D

BRANCH: CSE (AI & ML)

COURSE NAME: C Programming

COURSE CODE: FEL204

ACADEMIC YEAR: 2020-21 SEMESTER: II

L.T.J.S.S's

LOKMANYA TILAK COLLEGE OF ENGINEERING

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.



CERTIFICATE

This is to certify that Mr./Ms. Singh Sudham Dharmendra of
Semester II Division D bearing Roll No. AIMLD50 in the **First Year Engineering** has successfully performed 10 Experiments and 02 Assignments in the Subject **C Programming** during the period May2021 to July 2021.

Subject Teacher

Head of the Department

Principal

Date: 24/07/2021

LOKMANYA TILAK COLLEGE OF ENGINEERING

Sector No. 4., Vikas Nagar, Koparkhairane, Navi Mumbai -400 709.



VISION:

To create technically competent and ethically responsible professionals capable of providing efficient solutions to the contemporary world.

MISSION:

We aim to excel in our continual efforts, towards being one of the most recognized institutions by:

- Providing a conducive environment comprising high end infrastructure and state-of-the-art laboratory facilities wherein the students, faculty and staff can collectively enhance their technical potential.*
- Encouraging innovation through research activities for the benefits of society.*
- Developing competent professionals responsive to changing technology.*



Course Outcomes (COs)

Academic Year	Class / Semester	Subject Name	Subject Code
2020-21	F.E/ II	C Programming	FEL204

C01	Formulate simple algorithms for arithmetic, logical problems and translate them to programs in C language
CO2	Implement, test and execute programs comprising of control structures
CO3	Decompose a problem into functions and synthesize a complete program.
C04	Demonstrate the use of arrays, strings and structures in C language.
CO5	Understand the concept of pointers

Student Signature:**Student Name:** SINGH SUDHAM DHARMENDRA

**LABORATORY ASSESSMENT**

Student's Name	SINGH SUDHAM DHARMENDRA	Div & Roll No	D & AIMLD50
Class / Sem.	F.E. - II	Academic Year	2020-21
Name of the Subject	C Programming	Course Code	FEL204

Assessment Parameters for Experiments

Title (Experiments)	Date of performance	Date Of submission
1 Study of basic data types and I/O operations.	01/06/2021	09/06/2021
2 Study of branching statements.	08/06/2021	13/06/2021
3 Study of looping statements.	21/06/2021	04/07/2021
4 Study of functions.	01/07/2021	04/07/2021
5 Study of recursions (function).	02/07/2021	09/07/2021
6 Study of Arrays.	09/07/2021	13/07/2021
7 Study of strings.	12/07/2021	18/07/2021
8 Study of structures.	15/07/2021	21/07/2021
9 Study of pointers.	01/07/2021	21/07/2021
10 Study of sum of series and pattern programs.	23/06/2021	21/07/2021
Assignment no-1		09/07/2021
Assignment no-02		09/07/2021

Student Signature

Subject Teacher Signature

Experiment No: 1

Aim:

Study of basic data types and I/O operations.

Name and roll no of student	Division	Date of performance	Date of submission
Name:Singh Sudham Dharmendra	D	01/06/2021	09/06/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to find the simple interest for the Amount (P), Rate of Interest (R) and Number of years (N).**
- 2. Write a program to convert the given temperature in degree centigrade to Fahrenheit and vice versa.**
- 3. Write a program to obtain roots of second order quadratic equation of the form $ax^2+bx+c=0$ where a, b, c is non-zero.**
- 4. Write a program to swap two numbers without using third variable.**

Program 1 :-Problem :

Write a program to find the simple interest for the Amount(P), Rate of Interest (R) & Number of years (N).

Algorithm :

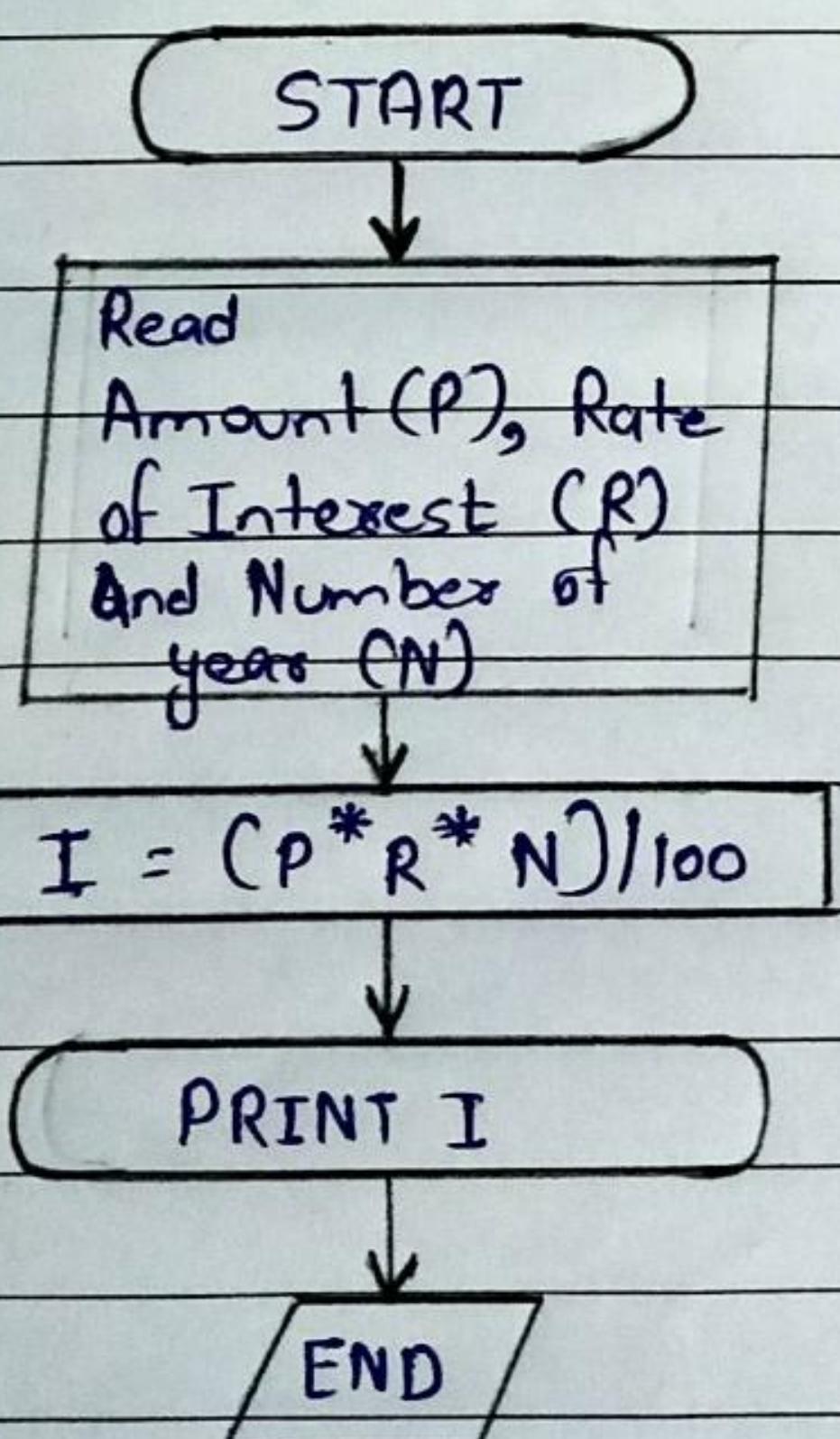
Step 0: START

Step 1: Read Amount (P), Rate of interest (R) and Number of years (N).

Step 2: Convert P, R and N into simple Interest (I) using the formula $I = (P * R * N) / 100$

Step 3: Display the result

Step 4: STOP

Flowchart:

Program :-

```
#include<stdio.h>
main()
{
    float P, R, N, I;
    printf("Enter the amount:\n");
    scanf ("%f", &P);
    printf("Enter the rate:\n");
    scanf ("%f", &R);
    printf("Enter number of years:\n");
    scanf ("%f", &N);
    I = (P * R * N) / 100;
```

```
printf("The simple interest is: %f\n", I);
```

Output:

Enter the amount:

100

Enter the rate:

4

Enter number of years:

2

The simple interest is : 8.000000

Program 2 :-Problem :

Write a program to convert the given temperature in degree centigrade to Fahrenheit and vice versa.

Algorithm :

Step 0: START

Step 1: Read the temperature in degree centigrade.

Step 2: Convert the centigrade to Fahrenheit using the formula
 $F = (9/5 * C) + 32$.

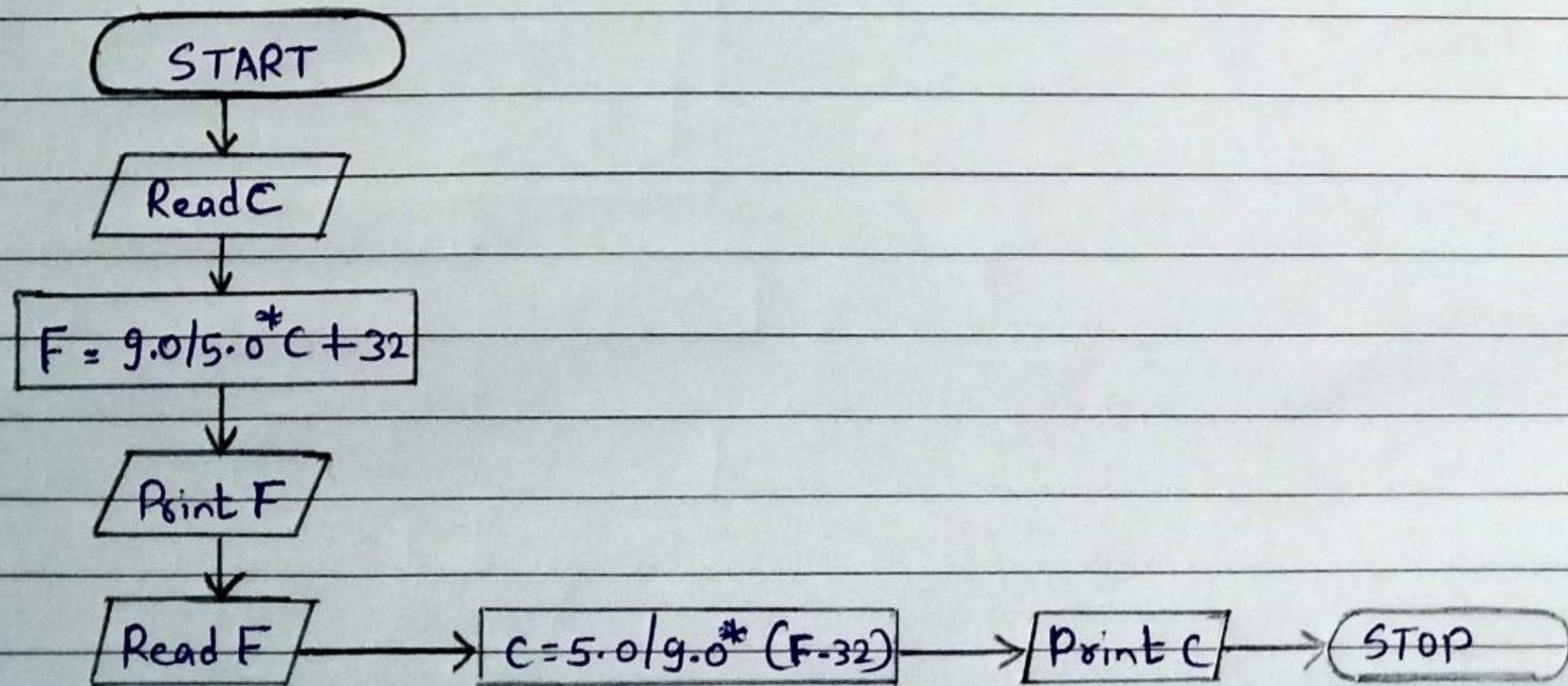
Step 3: Print F.

Step 4: Read the temperature in degree Fahrenheit.

Step 5: Convert the Fahrenheit to centigrade using the formula
 $C = 5/9 * (F - 32)$.

Step 6: Print C.

Step 7: STOP

Flowchart :

Program :

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{
```

```
    float c,f;
```

```
    printf("Enter the temperature in centigrade :");
```

```
    scanf("%f", &c);
```

```
    f = (9.0 / 5.0 * c) + 32;
```

```
    printf("Temperature in Fahrenheit is =>%f\n", f);
```

```
    printf("\nEnter the temperature in Fahrenheit :");
```

```
    scanf("%f", &f);
```

```
    printf("Temperature in centigrade is =>%f\n", c);
```

```
    getch();
```

```
}
```

Output :

Enter the temperature in centigrade : 35

Temperature in Fahrenheit is => 95.000000

Enter the temperature in Fahrenheit : 95

Temperature in centigrade is => 35.000000

Program 3:

Problem:

Write a program to obtain roots of second order quadratic equation of the form $ax^2 + bx + c = 0$ where a, b, c is non-zero

Algorithm:

Step 0: START

Step 1: Read coefficients a, b, c of the equation $ax^2 + bx + c = 0$.

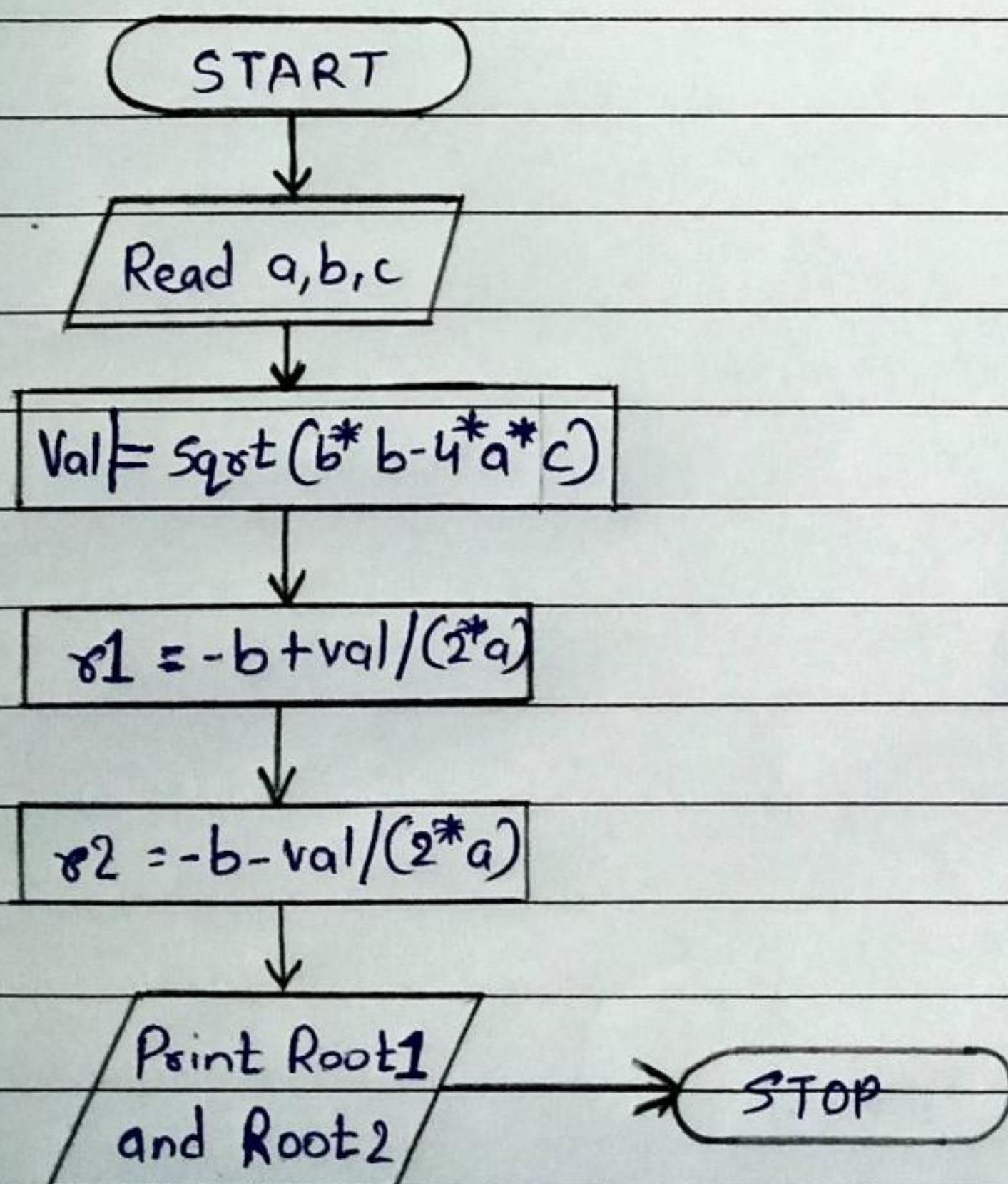
Step 2: Calculate the roots of quadratic equation by formula

$$\text{root}_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Step 3: Display the results.

Step 4: STOP

Flowchart:



Program :

```
#include <stdio.h>
#include <math.h>
main()
{
    int a, b, c;
    float r1, r2, val;

    printf("Enter a, b and c\n");
    scanf ("%d %d %d", &a, &b, &c);

    val = sqrt ((b*b) - (4*a*c));
    r1 = ((-b) + val) / (2*a);
    printf ("root1 is => %.f\n", r1);
    r2 = ((-b) - val) / (2*a);
    printf ("root2 is => %.f", r2);
}
```

Output :

```
Enter a, b and c
1
4
4
root1 is => -2.000000
root2 is => -2.000000
```

Program 4:

Problem:

Write a program to swap two numbers without using third variable.

Algorithm:

Step 0 : START

Step 1 : Read two numbers a and b

Step 2 : Swapping without third variable can be done by applying the following logic:

$$a = a+b$$

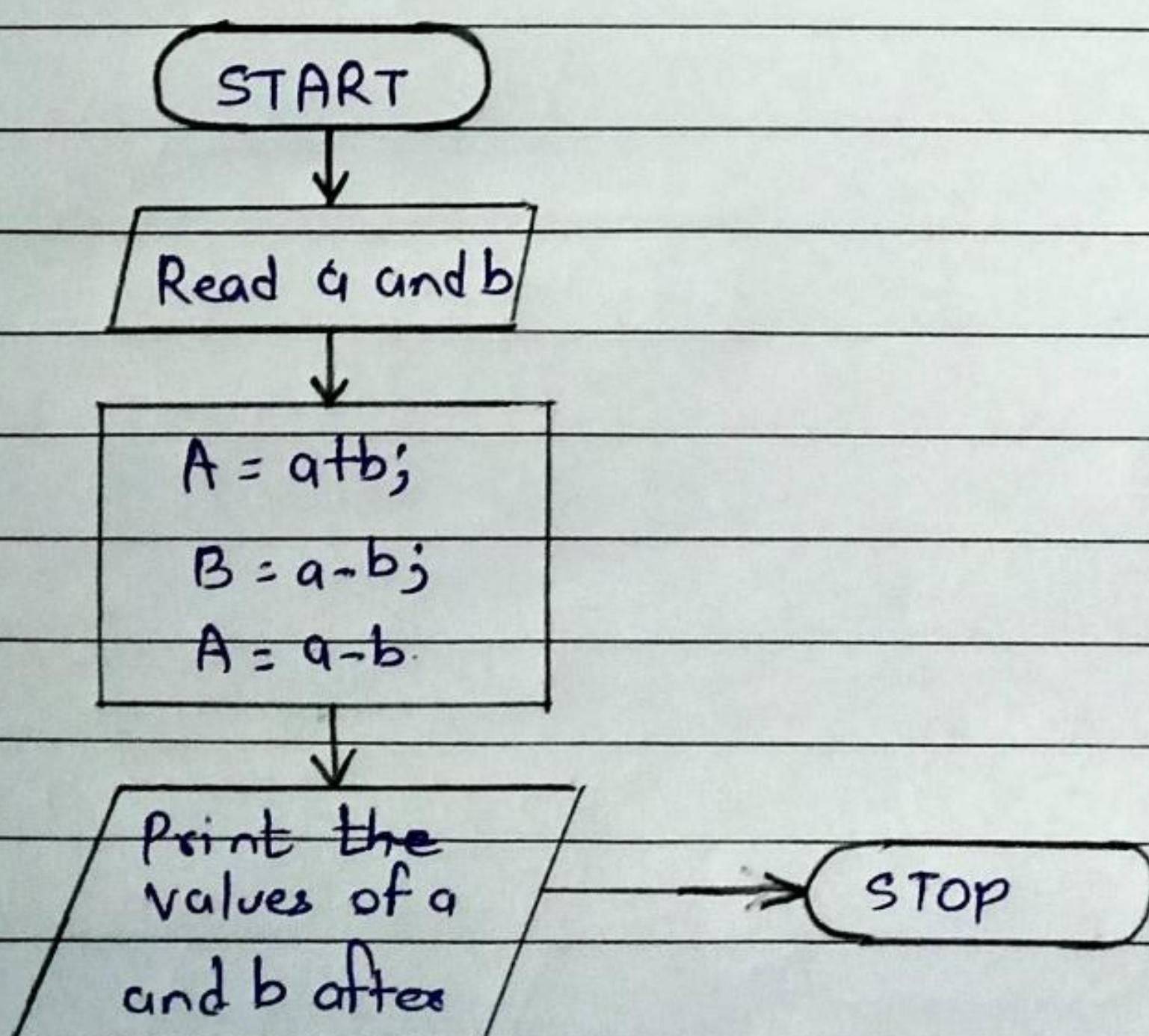
$$b = a-b$$

$$a = a-b$$

Step 3 : Display the result after swapping.

Step 4 : STOP.

Flowchart:



Program :

```
#include<stdio.h>
#include <conio.h>
void main()
{
    int a,b;
    printf("Enter the value for a and b:\n");
    scanf("%d%d", &a, &b);
    printf("Before swapping value of a is : %d\n", a);
    printf("Before swapping value of b is : %d\n", b);
    a = a+b;
    b = a-b;
    a = a-b;
    printf("After swapping value of a : %d\n", a);
    printf("After swapping value of b : %d\n", b);
    getch();
}
```

Output:

Enter value for a and b:

17
11

Before swapping value of a is : 17
Before swapping value of b is : 11

After swapping value of a : 11
After swapping value of b : 17

Experiment No: 2

Aim:

Study of branching statements.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) D	08/06/2021	13/06/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to find whether the given number entered is even or odd.**
- 2. Write a program to read marks of the student and display grade based on the following norms using else if Ladder:**
>=75: Grade A
>=60 and <75: Grade B
>=40 and <60: Grade C
<40: Fail
- 3. Write a program to develop a simple calculator that accepts two floating point numbers from the keyboard. Display a menu to the user and get the user's choice. Perform the operation and display the result using switch statement.**

Program 1:Problem:

Write a program to find whether the given number entered is even or odd.

Algorithm:

Step 0: START

Step 1: Take user input (number)

Step 2: Check if number is equal to zero

Step 3: If true, print number is zero

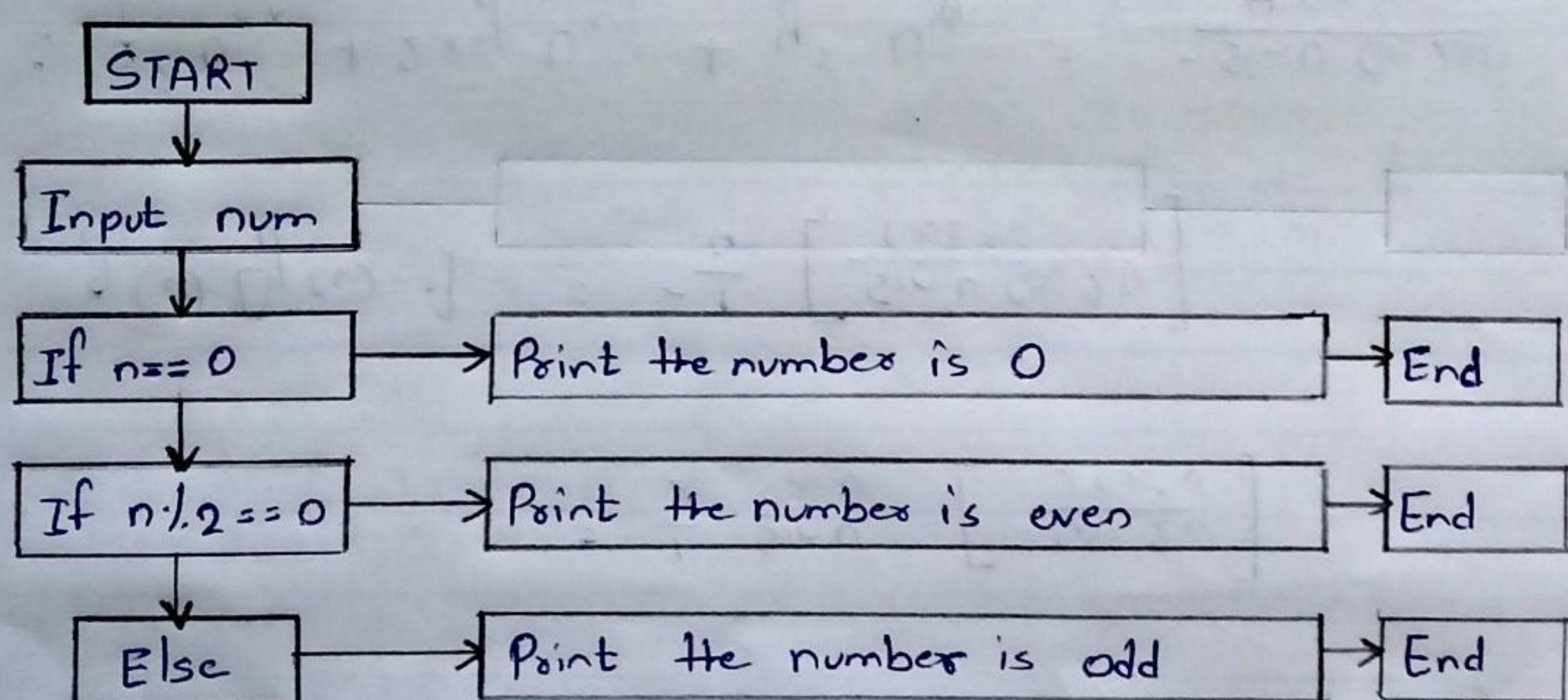
Step 4: If false, check if number is divisible by 2

Step 5: If true, print number is even

Step 6: If none of above case is true, then print number is odd

Step 7: Display the result

Step 8: STOP

Flow chart:

Program :

```
#include <stdio.h>
int main ()
{
    int n;
    printf ("Enter the number:");
    scanf ("%d", &n);

    if (n == 0)
        printf ("The given number is zero");
    else if (n % 2 == 0)
        printf ("The number is even");
    else
        printf ("The number is odd");

    return 0;
}
```

Output :

Enter the number : 7
The number is odd

Program 2 :Problem :

Write a program to read marks of the student and display grade based on the following norms using else if ladder :

≥ 75 : Grade A

≥ 60 and < 75 : Grade B

≥ 40 and < 60 : Grade C

< 40 : Fail

Algorithm :

Step 0 : START

Step 1 : Take marks from the user input

Step 2 : If input is greater than equal to 75 then print Grade A

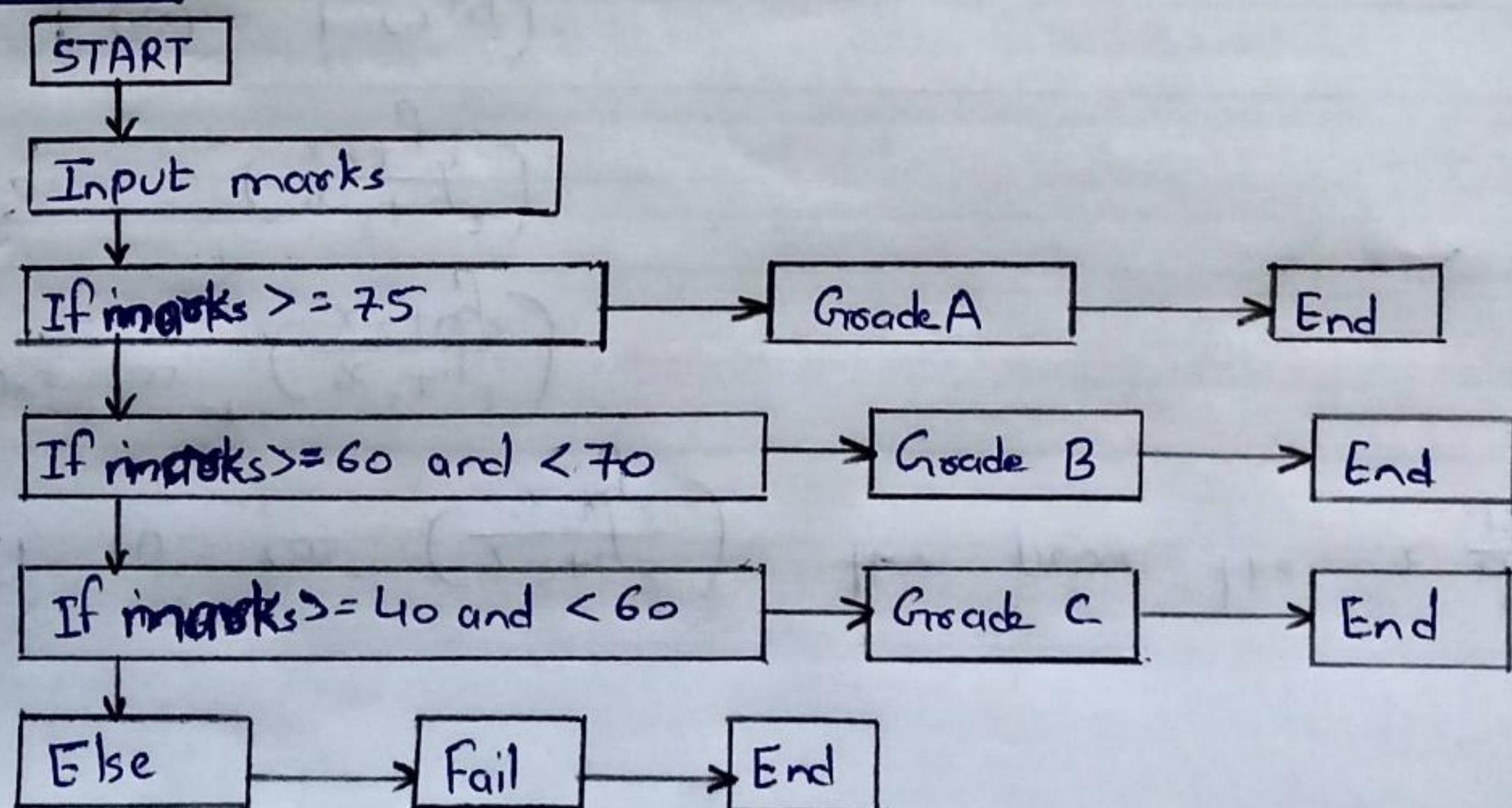
Step 3 : If input is greater than equal to 60 and less than 75 then print Grade B .

Step 4 : If input is greater than equal to 40 and less than 60 then print Grade C

Step 5 : Else print fail (If input is ~~small~~ less than 40)

Step 6 : Display the result .

Step 7 : STOP .

Flowchart :

Program:

```
#include <stdio.h>
int main()
{
    float mark;
    printf("Enter marks of the student from 0 to 100: ");
    scanf("%f", &mark);

    if(mark >= 75)
        printf("Grade A");
    else if (mark >= 60 && mark < 75)
        printf("Grade B");
    else if (mark >= 40 && mark < 60)
        printf("Grade C");
    else
        printf("Fail");

    return 0;
}
```

Output:

Enter marks of the student from 0 to 100 : 78.5

Grade A

Program 3:Problem:

Write a program develops a simple calculator that accepts two floating point numbers from the keyboard. Display a menu to the user and get the user's choice. Perform the operation and display the result using switch statement.

Algorithm :

Step 0 : START

Step 1 : Declare 2 variables in float as n_1 and n_2 . Also declare one variable in char datatype as Ch.

Step 2 : ~~Point the whole menu with choice of operators and ask the users to input it's choice.~~

Step 3 : ~~Declare 1 variable in float datatype as a .~~

Step 4 : ~~Point the whole when with choice of operators and ask the user to input it's choice .~~

Step 5 : In switch case if user input '+' then add the two input

Step 6 : If user input '-' then subtract n_1 and n_2 and print answer

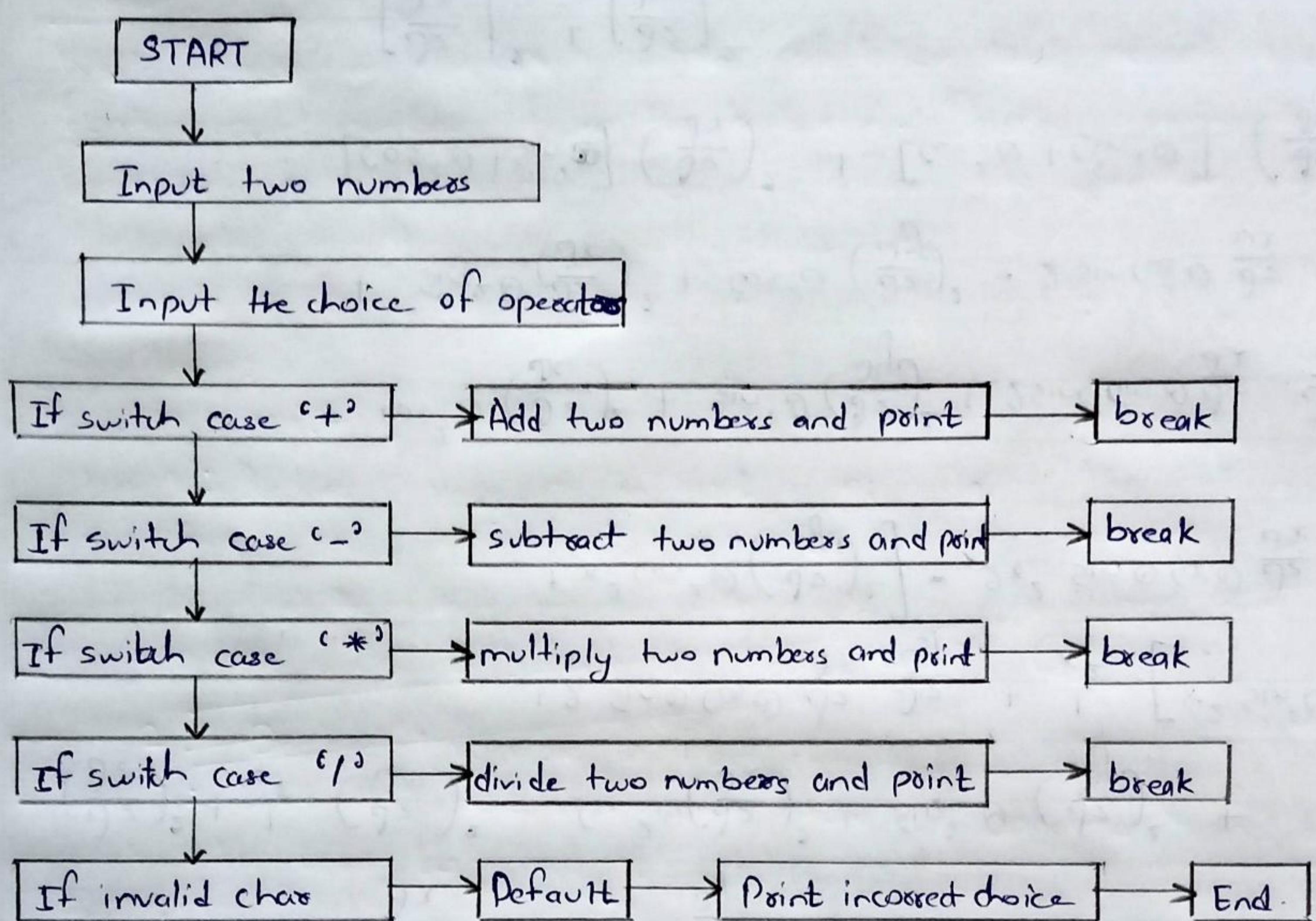
Step 7 : If user input '*' then multiply n_1 and n_2 and print answer.

Step 8 : If user input '/' then divide n_1 and n_2 and print answer

Step 9 : If user input a invalid character then print incorrect choice .

Step 10 : Display the result.

Step 11 : STOP.

Flowchart :

Program :

```

#include <stdio.h>
int main()
{
    char ch;
    float n1, n2, a;
    printf ("Enter the two numbers : ");
    scanf ("%f %f", &n1, &n2);
    printf ("1. addition => + \n 2. subtraction => - \n 3. multiplication => * \n 4. division => / \n Enter your choice : ");
    scanf ("%c", &ch);
    switch (ch)
    {
        case '+':
            a = (n1 + n2);
            printf ("The answer is %f \n", a);
            break;
        case '-':
            a = (n1 - n2);
            printf ("The answer is %f \n", a);
            break;
        case '*':
            a = (n1 * n2);
            printf ("The answer is %f \n", a);
            break;
        case '/':
            a = (n1 / n2);
            printf ("The answer is %f \n", a);
            break;
        default:
            printf ("incorrect choice of operator ");
            break;
    }
    return 0;
}

```

Output:

Enter the two numbers : 45

50

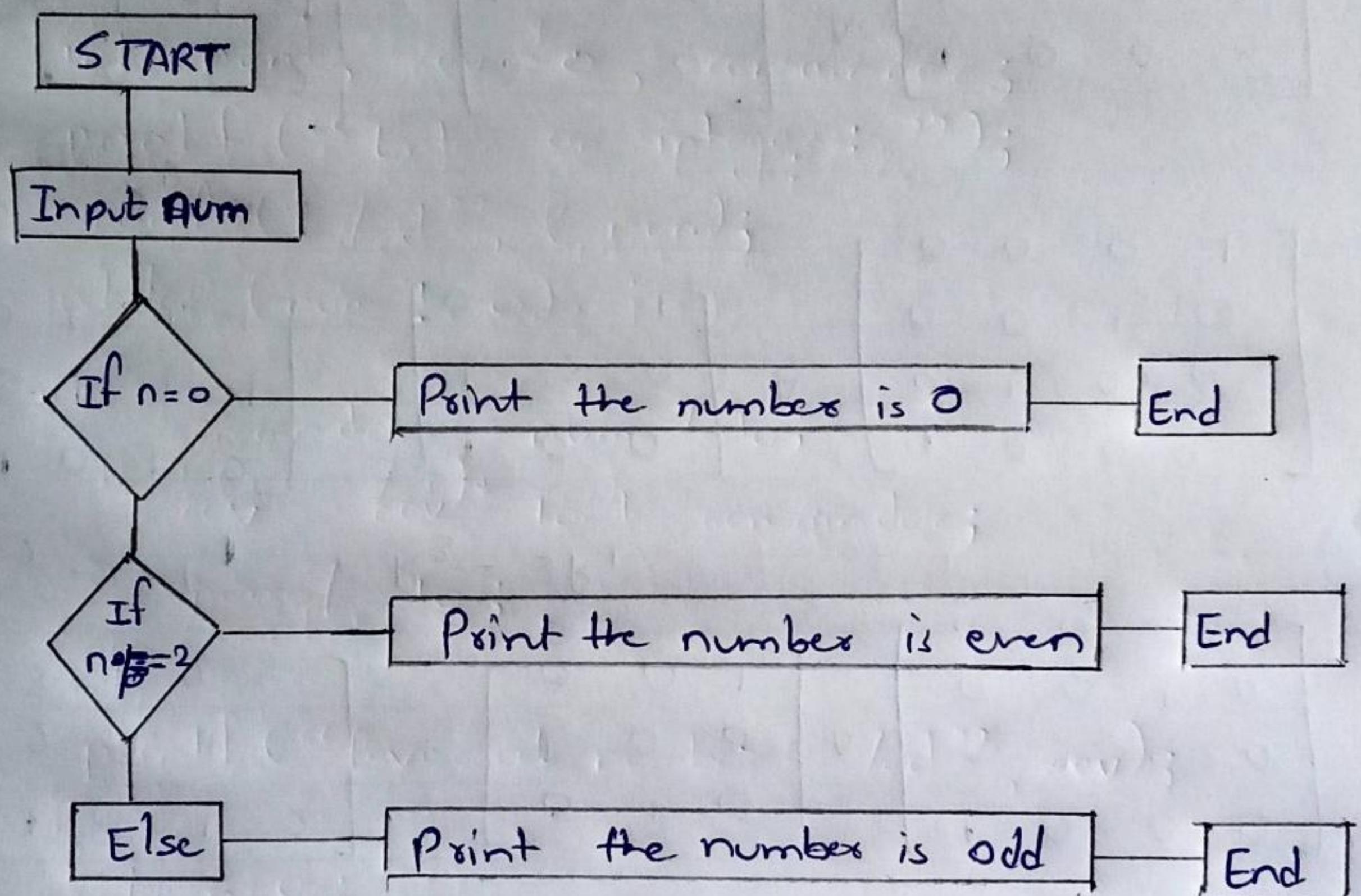
1. addition $\Rightarrow +$
2. subtraction $\Rightarrow -$
3. multiplication $\Rightarrow *$
4. division $\Rightarrow /$

Enter your choice : +

The answer is 95

Program 1:Problem:

Write a program to find whether the given number entered is even or odd

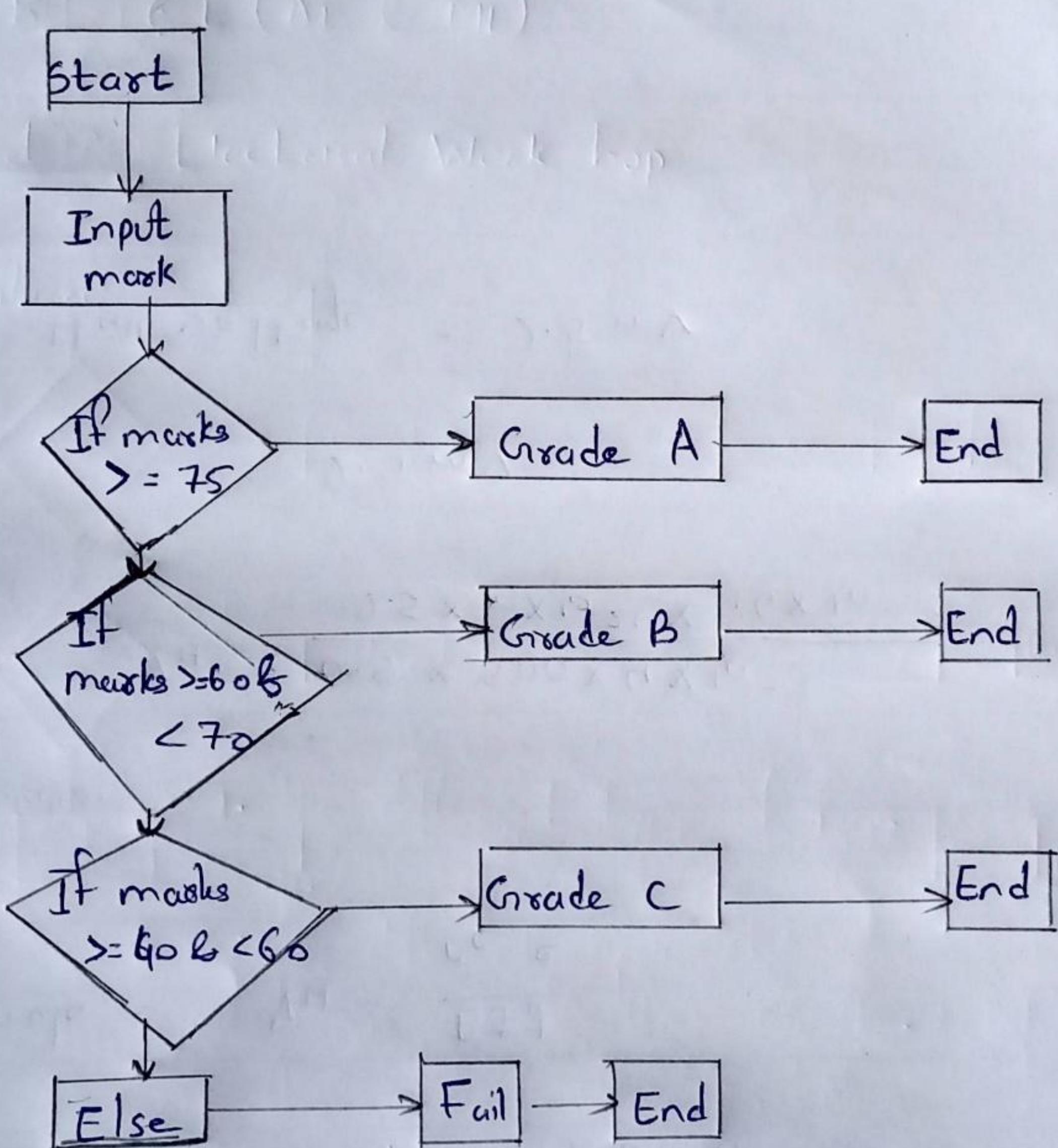
Flowchart:

Program 2:

Problem:

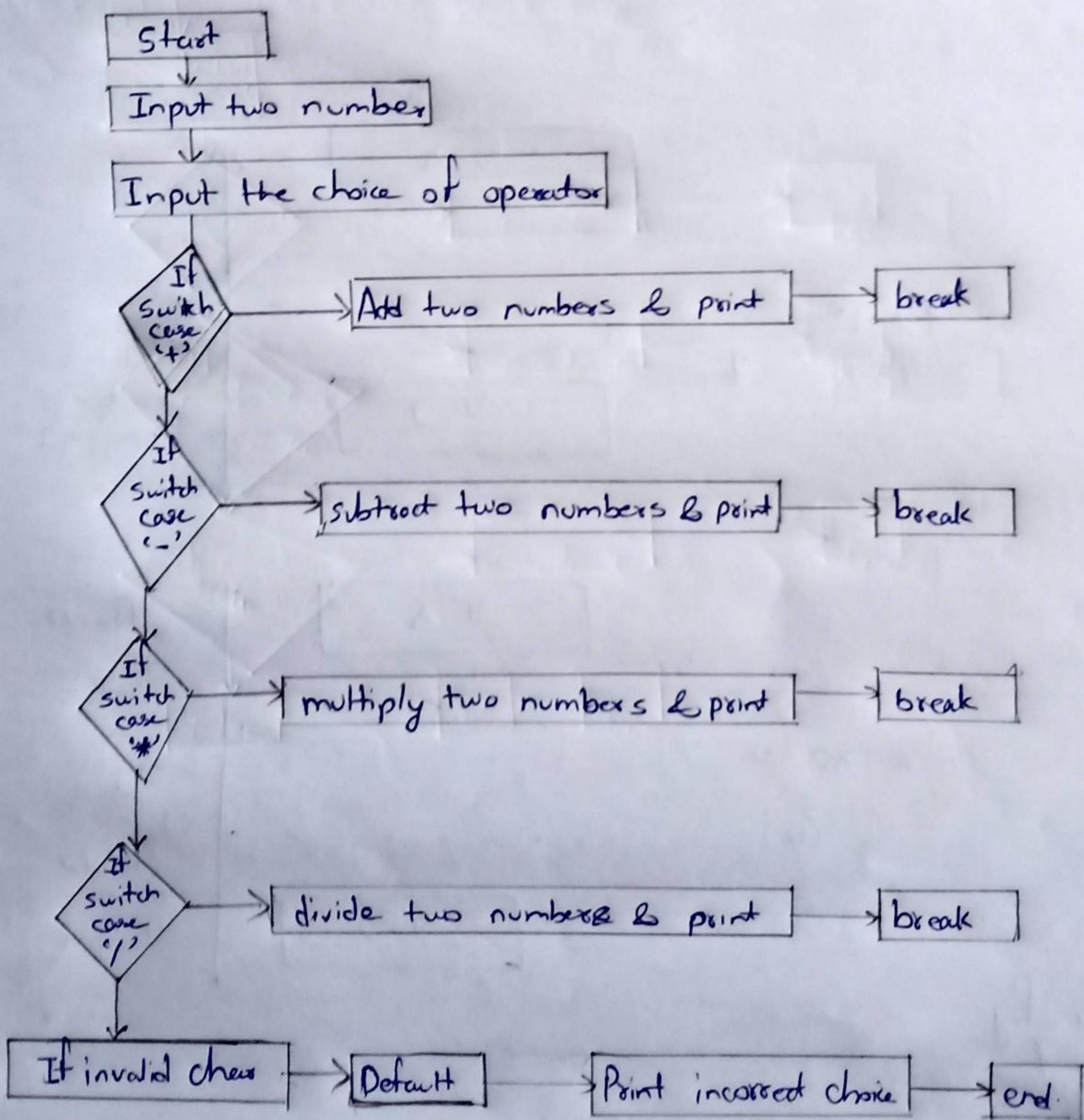
Write a program to read marks of the student and display grade based on the following norms using else if ladder.

Flowchart:



Program 3:Problem:

Write a program develop a simple calculator that accept two floating point numbers from keyboard. Display a menu to the user to get the user's choice. Perform the operation & display the result using switch statement.

Flowchart:

Experiment No: 3

Aim:

Study of looping statements.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) & D	21/06/2021	04/07/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to print the first ten terms of the Fibonacci Sequence assuming the first two terms as 0 and 1.**
- 2. Write a program to check whether given number is prime or not.**
- 3. Write a program to reverse a given number using while loop.**

Program 1:Problem:

Write a program to print the first ten terms of the Fibonacci sequence assuming the first two terms as 0 & 1

Program:

```
#include <stdio.h>
int main ()
{
    int i, n, t1 = 0, t2 = 1;
    int nextTerm = t1 + t2;

    printf ("Enter the number of terms : ");
    scanf ("%d", &n);

    printf ("Fibonacci Series : %d, %d, ", t1, t2);
    for (i = 3, i < n; ++i)
    {
        printf ("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }
    return 0;
}
```

Output:

Enter the number of terms : 10
 Fibonacci Series : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Program 2:Problem:

Write a program to check whether given number is prime or not.

Program:

```
#include <stdio.h>
void main()
{
    int num, i, count = 1;

    printf ("Input a number : ");
    scanf ("%d", &num);
    for (i=2; i<=num; i++)
    {
        if ((num % i) == 0)
        {
            count++;
        }
    }
    if (count == 2)
    {
        printf ("It is prime number");
    }
    else
        printf ("It is not a prime number");
}
```

Output:

Input a number : 5
It is a prime number

Program 3:Problem:

Write a program to reverse a given number using while loop

Program:

```
#include <stdio.h>
int main()
{
    int num, rev = 0, remainder;
    printf ("Enter an integer: ");
    scanf ("%d", &num);
    while (num != 0)
    {
        remainder = num % 10;
        rev = rev * 10 + remainder;
        num /= 10;
    }
    printf ("Reversed number = %d", rev);
    return 0;
}
```

Output:

Enter an integer: 25
 Reversed an integer: 52

Experiment No: 4

Aim:

Study of functions.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) & D	01/07/2021	04/07/2021
Roll no: AIMLD50			

Program list:

- 1. Devise a function called $\min(x, y)$ that returns the smaller of two double values. Test the function with sample data.**
- 2. Write a program to find sum of digits of a number using function.**

Program 1 :Problem :

Devise a function called $\min(x, y)$ that returns the smaller of two double values. Test the function with sample data.

Program :

```
#include <stdio.h>
double min(double, double);
int main()
{
    double num1, num2;
    printf("Enter the first number : ");
    scanf("%f", &num1);
    printf("Enter the second number : ");
    scanf("%f", &num2);
    double minimum = min(num1, num2);
    printf("\nsmallest of the two numbers is : %f", minimum);
    return 0;
}
```

```
double min (double num1, double num2)
```

```
{
    if (num1 > num2)
        return num2;
    else
        return num1;
}
```

Output:

Enter the first number : 234

Enter the second number: 573

smallest of the two numbers is : 234

Program 2:Problem:

Write a program to find sum of digits of a number using function.

Program:

```
#include <stdio.h>
int sum(int);
int main()
{
    int sum1, t, n, rem;
    printf("Enter the number : ");
    scanf("%d", &n);
    t = n;
    while (n != 0)
    {
        rem = n % 10;
        sum1 = sum1 + rem;
        n = n / 10;
    }
    printf("Sum of digits of %d is %d\n", t, sum1);
}

int sum(int a)
{
    return 0;
}
```

Output:

Enter number : 123
 Sum of digits of 123 is 6

Experiment No: 5

Aim:

Study of recursion(function).

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) & D	02/07/2021	09/07/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to calculate the factorial of a given number using recursive function.**
- 2. Write a program to compute a^b using recursion.**

Program 1 :Problem :

Write a program to calculate the factorial of a given number using recursive function.

Program :

```
#include <stdio.h>
void main()
{
    int factorial (int);
    int n, f;
    printf ("Enter the number : ");
    scanf ("%d", &n);
    f = factorial (n);
    printf ("Factorial of the number is %d", f);
}

int factorial (int n)
{
    int f;
    if (n == 1)
        return 1;
    else
        f = n * factorial (n-1);
    return f;
}
```

Output:

Enter the number : 5
 Factorial of the Number is 120

Program 2:Problem:

Write a program to compute a^b using recursion

Program:

```
#include <stdio.h>
int power (int n1, int n2);
int main()
{
    int base, a, result;
    printf ("Enter base number: ");
    scanf ("%d", &base);
    printf ("Enter power number (positive integer): ");
    scanf ("%d", &a);
    result = power (base, a);
    printf ("%d ^ %d = %d", base, a, result);
    return 0;
}
int power (int base, int a)
{
    if (a == 0)
        return (base * power (base, a - 1));
    else
        return 1;
}
```

Output:

Enter base number : 5

Enter power number (positive integer) : 2

$$5^2 = 25$$

Experiment No: 6

Aim:

Study of Arrays.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) & D	09/07/2021	13/07/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to sort elements of an array (Bubble sort).**
- 2. Write a program to add two matrices using multi-dimensional arrays.**

Program 1 :-Problem:

Write a program to sort elements of an array
(Bubble sort).

Program:

```
#include <stdio.h>
int main()
{
    int array [100], n, c, d, swap;
    printf ("Enter number of elements\n");
    scanf ("%d", &n);
    printf ("Enter %d integers\n", n);
    for (c=0; c<n; c++)
        scanf ("%d", &array [c]);
    for (c=0; c<n-1; c++)
    {
        for (d=0; d<n-c-1; d++)
        {
            if (array [d] > array [d+1])
            {
                Swap = array [d];
                array [d] = array [d+1];
                array [d+1] = Swap;
            }
        }
    }
    printf ("sorted list in ascending order:\n");
    for (c=0; c<n; c++)
        printf ("%d\n", array [c]);
    return 0;
}
```

Output:

Enter the number of elements

5

Enter 5 integers

2

9

6

4

3

Sorted list in ascending order:

2

3

4

6

9

Program 2 :Problem :

Write a program to add two matrix using multi-dimensional arrays.

Program :

```
#include <stdio.h>
int main()
{
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;
    printf("Enter the number of rows: ");
    scanf("%d", &r);
    printf("Enter the number of columns: ");
    scanf("%d", &c);

    printf("\nEnter elements of 1st matrix:\n");
    for (i=0; i<r; ++i)
        for (j=0; j<c; ++j)
    {
        printf("Enter element a%d%d : ", i+1, j+1);
        scanf("%d" &a[i][j]);
    }

    printf("Enter elements of 2nd matrix:\n");
    for (i=0; i<r; ++i)
        for (j=0; j<c; ++j)
    {
        printf("Enter element b%d%d : ", i+1, j+1);
        scanf("%d" &b[i][j]);
    }

    for (i=0; i<r; ++i)
        for (j=0; j<c; ++j)
    {
        sum[i][j] = a[i][j] + b[i][j];
    }
}
```

```

printf("Sum of two matrices:\n");
for(i=0; i<r; ++i)
printf for(j=0; j<c; ++j)
{
    printf("%d ", sum[i][j]);
    if(j == c-1)
        printf("\n\n");
}
return 0;
}

```

Output:

Enter the number of rows: 2
 Enter the number of columns: 2

Enter elements of 1st matrix:

Enter element a11: 2
 Enter element a12: 5
 Enter element a21: 6
 Enter element a22: 3

Enter elements of 2nd matrix:

Enter element b11: 1
 Enter element b12: 7
 Enter element b21: 4
 Enter element b22: 5

Sum of two matrices :

3	12
10	8

Experiment No: 7

Aim:

Study of strings.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) & D	12/07/2021	18/07/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to find whether a given string is palindrome or not.**
- 2. Write a program to demonstrate strrev () function.**
- 3. Write a program to find frequency of characters in a string.**

Program 1:Problem:

Write a program to find whether a given string is palindrome or not.

Program:

```
#include <stdio.h>
#include <string.h>
int main ()
{
    char str1 [20], str2 [20];
    printf ("Enter string to check palindrome:\n");
    gets (str1);
    strcpy (str2, str1);
    strrev (str2);
    if (strcmp (str1, str2) == 0)
    {
        printf ("Given string is a palindrome\n");
    }
    else
    {
        printf ("not palindrome");
    }
    return 0;
}
```

Output:

Enter string to check palindrome:
level
Given string is a palindrome.

Program 2:Problem:

Write a program to demonstrate `strrev()` function.

Program:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s[100];
    printf ("Enter a string to reverse \n");
    gets(s);
    strrev(s);
    printf ("Reverse of the string : %s\n", s);
    return 0;
}
```

Output:

Enter a string to reverse
sucham

Reverse of the string : ~~suham~~ mahdus

Program 3:Problem:

Write a program to find frequency of characters in a string.

Program:

```
#include<stdio.h>
int main()
{
    char str[100], ch;
    int count = 0;
    printf("Enter the string:");
    fgets(str, 99, stdin);
    printf("Enter a character to find its frequency:");
    scanf("%c", &ch);
    for (int i=0; str[i] != '\0'; i++)
    {
        if (ch == str[i])
        {
            count++;
        }
    }
    printf("Frequency of %c: %d\n", ch, count);
    return 0;
}
```

Output:

Enter the string: Sudham

Enter a character to find its frequency: S

frequency of S: 1

Experiment No: 8

Aim:

Study of structures.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI & ML) & D	15/07/2021	21/07/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to read and print an employee's detail using structure.**

Program 1:Problem :

Write a program to read and print an employee's detail using structure.

Program:

```
#include <stdio.h>
struct employee
{
    int id, age, salary;
    char name [30];
}
emp [100];
int main ()
{
    int i, n;
    printf ("Enter the number of employees\n");
    scanf ("%d", &n);
    printf ("Enter employee info as id, name, age, salary\n");
    for (i = 0; i < n; i++)
    {
        scanf ("%d %s %d %d", &emp[i].id, emp[i].name, &emp[i].age, &emp[i].salary);
    }
    printf ("\nEMP_ID\tEMP_NAME\tEMP_AGE\t\tEMP_SALARY\n");
    for (i = 0; i < n; i++)
    {
        printf ("%d\t%s\t%d\t\t%d\n", emp[i].id, emp[i].name,
               emp[i].age, emp[i].salary);
    }
    return 0;
}
```

Output:

Enter the number of employees

2

Enter ~~the~~ employee info as id, name, age , salary

1

Sudham

19

30000

2

Monu

19

30000

EMP_ID	EMP_NAME	EMP_AGE	EMP_SALARY
1	Sudham	19	30000
2	Monu	19	30000

Experiment No: 9

Aim:

Study of pointers.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI&ML) & D	01/07/2021	21/07/2021
Roll no: AIMLD50			

Program list:

- 1. Write a program to swap two integer numbers using pointers.**

Program:Problem:

Write a program to swap two integer numbers using pointers.

Program:

```
#include <stdio.h>
int main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before swapping\n x = %d\n y = %d\n", x, y);
    a = &x;
    b = &y;
    temp = *b;
    *b = *a;
    *a = temp;
    printf("After swapping\n x = %d\n y = %d\n", x, y);
    return 0;
}
```

Output:

Enter the value of x any y

5

3

Before swapping

x = 5

y = 3

After swapping

x = 3

y = 5

Experiment No: 10

Aim:

Study of sum of series and pattern programs.

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra Roll no: AIMLD50	CSE(AI&ML) & D	23/06/2021	21/07/2021

Program list:

- 1. WAP to calculate sum of series of $1/1! + 2/2! + 3/3! + \dots + N/N!$**
- 2. WAP to display following pattern:**

1			
2	3		
4	5	6	
7	8	9	10

Program 1:Problem:

WAP to calculate sum of series of $1/1! + 2/2! + 3/3! + \dots + N/N!$

Program:

```
#include <stdio.h>
int main()
{
    int num = 1, count;
    float sum = 0.0, fact;
    while (num <= 7)
    {
        fact = 1;
        for (count = 1; count <= num; count++)
        {
            fact = fact * count;
        }
        sum = sum + (num / fact);
        num++;
    }
    printf ("sum of series is %f\n", sum);
    return 0;
}
```

Output:

Sum of Series is 2.718056

Program 2:Problem:

WAP to display Following pattern:-

Program:

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int n;
    printf(" Enter the number of row: ");
```

```
    scanf("%d", &n);
```

```
    int countpoint=1;
```

```
    for (int i=1; i<=n; i++)
```

```
    { for (int j=1; j<=i; ++j)
```

```
        { printf("%d", countpoint); countpoint++  
        countpoint++;
```

```
}
```

```
    printf("\n");
```

```
for
```

```
}
```

```
return 0;
```

```
}
```

Out put:

Enter the number of row: 4

1

2 3

4 5 6

7 8 9 10

1	2	3	
4	5	6	
7	8	9	10

Scanned by TapScanner

Assignment No: 1

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE(AI & ML) & D		09/07/2021
Roll no: AIMLD50			

- 1. Enlist all the data types in C language along with memory requirements, format specifiers and range.**
- 2. Distinguish between**
 - i) while and do while statement ii)
continue and break statement**
- 3. explain storage class.**
- 4. What is computer programming. What do you mean by structured programming? Develop an algorithm and flowchart to find reverse of a number.**
- 5. Enlist bitwise operators in C language. Explain any two with example.**

Q. 1) Enlist all the data types in C language along with memory requirements, format specifiers and range.

-
- char :- stores character
 - int :- variable used to store integer.
 - float :- used to store decimal numbers with single precision
 - double :- used to store decimal numbers with double precision.

Data type	Memory [bytes]	Range	Format specifiers
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	-(2^63) to (2^63)-1	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615	%llu
Signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
float	4	3.4×10^{-38} to 3.4×10^{38}	%f
double	8	1.7×10^{-308} to 1.7×10^{308}	%lf
long double	16	-3.4×10^{-4932} to 1.7×10^{4932}	%lf

Q.2) Distinguish between:-

(i)	<u>While statement</u>	<u>Do while statement</u>
	<ul style="list-style-type: none"> ① Statement is executed zero times, if condition is false ② Test condition is checked at the start of body loop ③ It is a entry controlled loop 	<ul style="list-style-type: none"> ① At least once the statement is executed ② Test condition is checked at the end of body loop. ③ It is a exit controlled loop.
(ii)	<u>Continue statement</u>	<u>Break statement</u>
	<ul style="list-style-type: none"> ① Instead of terminating the while loop control is passed to next statement within the loop. ② The code inside the loop followed by continue is skipped instead of terminating while loop if condition is false 	<ul style="list-style-type: none"> ① Used to terminate the loop, as soon as break is encountered within loop. ② Loop terminates and control is given to the next statement after the loop.

Q.3) Explain storage class

→ Storage classes are used to describe the features of a variable function. These features basically include the scope, visibility and life-time which help us to trace the existence of a particular variable during the runtime of a program. C language uses four storage classes, namely: auto, extern, static & registers.

Storage Specifier	storage	initial value	Scope	life
auto	stack	Garbage	within block	End of block
extern	Data Segment	zero	global multiple files	Till end of program
static	Data Segment	zero	within block	Till end of program
register	CPU Registers	Garbage	within block	End of block.

Q.4) What is computer programming. What do you mean by structured programming? Develop an algorithm & flowchart to find reverse of a number.

→ Computer Programming is the process of designing and building an executable computer program to accomplish a specific computing result or to perform a specific task. Structured programming is a programming paradigm aimed to improve clarity, quality and development time of computer program by making extensive use of functions, decision making statements, block structures and subroutines.

* Write a program to reverse a number using while loop.

Algorithm:-

① Input number

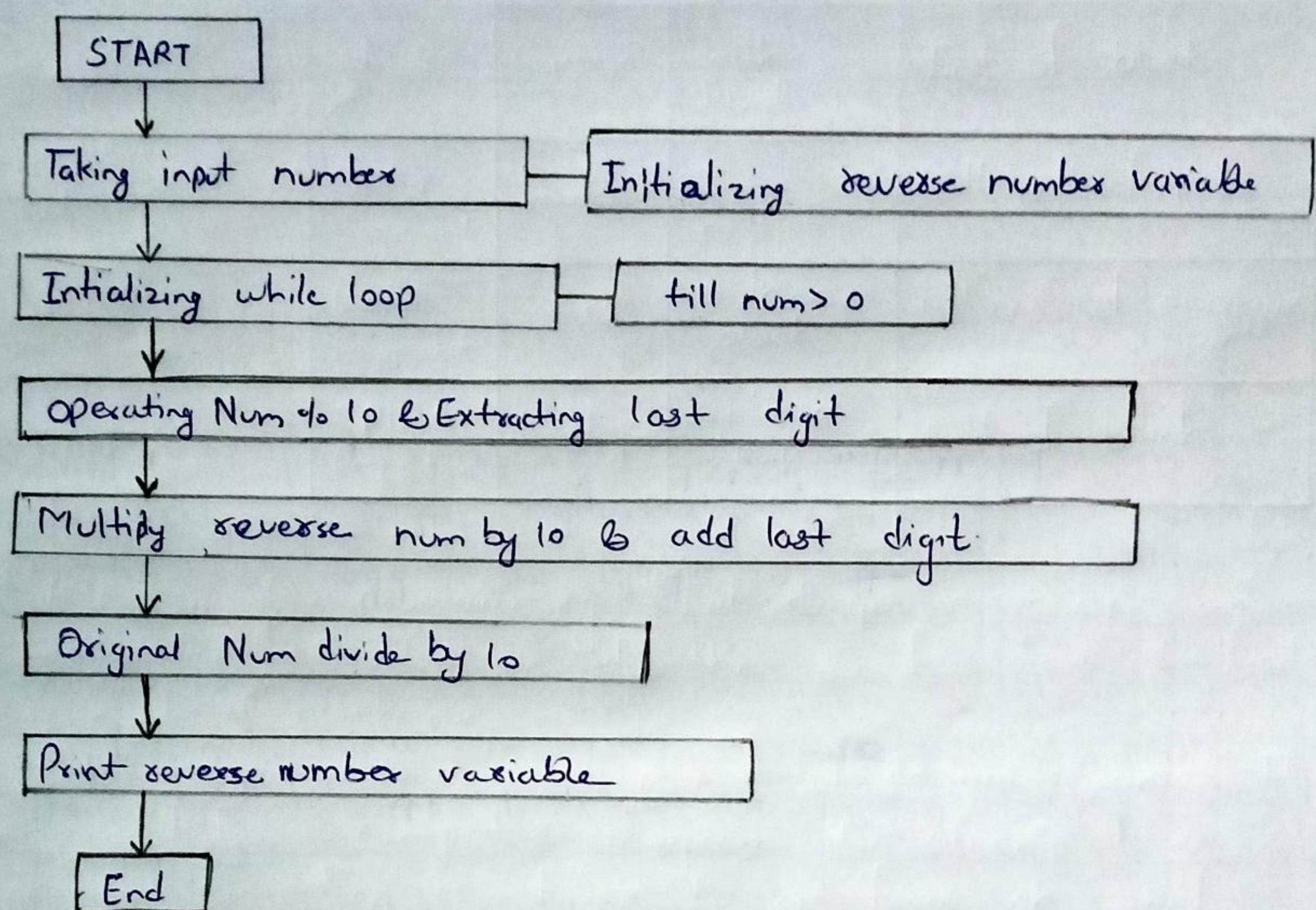
② Initializing loop till number becomes 0

③ In loop extracting the last digit

④ Multiplying existing digit by 10 & adding the extracted digit

⑤ Deleting the last number from original number

⑥ Printing the final reversed number.

Flowchart:-Program:

```

#include <stdio.h>
#include <math.h>
int main ()
{
    int ToRevNum;
    printf ("Enter number to be reversed-\n");
    scanf ("%d", &ToRevNum);
    int RevNum = 0;
    while (ToRevNum > 0)
    {
        int operation = ToRevNum % 10;
        RevNum = RevNum * 10 + operation;
        ToRevNum /= 10;
    }
    printf ("%d", RevNum);
    return 0;
}
  
```

Output :- Enter the number to be reversed - 49375
RevNum - 57394

Q.5) Enlist bitwise operators in C language. Explain any two with example.

$$A = 60 \text{ and } B = 13$$

\rightarrow	Operator	Description	Example.
	&	Binary AND operator copies a bit to the result if it exists in both operands	$(A \& B) = 12 \text{ ie, } 00001100$
		Binary OR operator copies a bit if it exists in either operand	$(A B) = 61, \text{ ie, } 00111101$
	\wedge	Binary XOR operator copies a bit if it is set in one operand but not both	$(A \wedge B) = 49, \text{ ie, } 00110001$
	\sim	Binary One's complement operator is unary and has the effect of flipping bits	$(\sim A) = \sim (60) \text{ ie, } 11000011$
	<<	Binary left shift operator the left operand value is moved left by the number of bits specified by the right operand.	$A << 2 = 24 \text{ o i.e, } 11110000$
	>>	Binary right shift operator the left operand value is moved right by the number of bits specified by the right operand	$A >> 2 = 15 \text{ i.e, } 00001111$

Assignment No: 2

Name and roll no of student	Class and Division	Date of performance	Date of submission
Name: Singh Sudham Dharmendra	CSE (AI & ML) & D		09/07/2021
Roll no: AIMLD50			

- 1. What is an array? What does an array name signify? Can array index be negative? Write a program to arrange the number stored in an array in such a way that the array will have the odd numbers followed by even numbers.**
- 2. Distinguish between structure and union. Explain concept of nested structure. Declare a structure to enter employee Information like name, id, salary, date of joining. Use nested structure to get the address of an employee. Write a program to read 10 records and display them.**
- 3. What is string? Explain the use of gets ()? Write a c program that will read a word and rewrite it in alphabetical order. For ex. If the word is “matrix” the program should print “aimrtx”.**
- 4. What is pointer. Explain how the pointer variable is declared and initialized.**
- 5. Explain string functions with example.**

Q.1) What is an array? What does an array name signify?
 Can array index be negative? Write a program to arrange the numbers stored in array in such a way that the array will have odd numbers followed by even numbers.

- • An array is any programming language^{which} is a collection of similar data items stored at contiguous memory locations and elements can be accessed randomly using indices of an array.
 • Array name represents memory address to array itself.
 • Array index can never be negative in C language as array starts from index 0 to positive natural number index.

Program: //Lomuto's Partition method.

```

// Time Complexity: O(n)
#include<stdio.h>
#include<math.h>
int main ()
{
    int arrsize;
    printf ("Enter array size:\n");
    scanf ("%d", &arrsize);
    int arr [arrsize];
    printf ("Enter array:\n");
    for (int s=0; s<arrsize; s++)
    {
        scanf ("%d", &arr [s]);
    }
    int temElement m = -1
    for (int s=0; s<arrsize; s++)
    {
        if (arr [s] % 2 == 0)
        {
            m++;
            int temElement = arr [s];
            arr [s] = arr [m];
            arr [m] = tem element
        }
    }
    printf ("Array as per condition is: ");
    for (int s=0; s<arrsize, s++) {
        printf ("%d", arr [s]);
    }
    return 0;
}
  
```

Q.2) Distinguish between structure and union. Explain concept of nested structure. Declare a structure to enter employee information like name, id, salary, date of joining. Use nested structure to get address of an employee. Write a program to read 10 records and display them.



Structure	Union
① keyword struct is used to define structure.	① keyword union is used to define union.
② When a variable is associated with structure, the compiler allocates memory for each member. The size of structure is greater than or equal to the sum of sizes of its members.	② When a variable is associated with union, the compiler allocates the memory by considering the size of the largest memory. So size of union is equal to the size of largest number.
③ Each member within a structure is assigned unique storage area of location.	③ Memory allocated is shared by individual members of union.
④ Altering the value of a member will not affect other members of structure.	④ Altering the value of any of the member will alter other member values.
⑤ Individual members can be accessed at a time.	⑤ Only one member can be accessed at a time.
⑥ Several members of a structure can initialize at once.	⑥ Only the first member of a union can be initialized.

Provide us the feature of nesting one structure within another structure by using which, complex data types can be created. For example, we may need to store the employee details in structure, the attribute address may also have subparts like building, pin no, city name. Hence, to store the address of employee in employee data we need store address in different structure and nest them.

Internal StructureProgram:

```

#include <stdio.h>
#include <stdlib.h>
struct data
{
    char name[30];
    int id;
    int salary;
    int joining date;
}
Employee [10];
struct address
{
    char building No[100];
    int pin No;
    char city [50];
}
emp[10];
int main ()
{
    int s;
    printf ("Enter upto 10 employee details :\n\n");
    for (s=0; s<10; s++)
    {
        printf ("Employee %d :- \n", s+1);
        printf ("Name : ");
        scanf ("%s", &employees[s].name);
        printf ("Id : ");
        scanf ("%d", &employees[s].id);
        printf ("Salary : ");
        scanf ("%d", &employees[s].salary);
        printf ("Joining date : ");
        scanf ("%d", &employees[s].joining date);
        printf ("Enter building / flat No : ");
        scanf ("%s", &employees[s].building No);
        printf ("Enter pin no : ");
        scanf ("%d", &employees[s].pin No);
        printf ("Enter City : ");
        scanf ("%s", &employees[s].city);
    }
}

```

```

        printf ("\n");
    }

    printf ("Employees Details are :\n");
    for (s=0; s<10; s++)
    {
        printf ("Name : ");
        scanf ("%s\n", &employees [s]. name);
        printf ("Id : ");
        scanf ("%d\n", &employees [s]. id);
        printf ("Salary : ");
        scanf ("%d\n", &employees [s]. salary);
        printf ("joining date : ");
        scanf ("%d\n", &employees [s]. joining date);
        printf ("Building / Flat No. : ");
        scanf ("%s\n", &employees [s]. building No);
        printf ("Pin No : ");
        scanf ("%d\n", &employees [s]. pin No);
        printf ("City : ");
        scanf ("%s\n", &employees [s]. city);
        printf ("\n");
    }
    return 0;
}

```

Q. 3) What is string? Explain the use of gets ()? Write a C program that will read a word and rewrite it in alphabetical order. For ex. If the word is "matrix" the program should print "aimstx".

→ Strings are defined as an array of characters, the difference between string & character ' \0 '.

There are two types of get functions.

Both get(char*) & get(void) prototype can be used to fetch a character including the blank space, tab and newline characters.

The `get(char*)` version assigns the input character to its argument and the `get(void)` version returns the input character. Since these functions are members of `input/output stream classes`, these must be invoked using appropriate objects.

Program :- // time complexity $O(n^2)$

```
#include <stdio.h>
#include <string.h>
int main ()
{
    int string length;
    printf ("Enter string length:\n");
    scanf ("%d", &string length);
    char unsort string [string length];
    printf ("Enter string to be sorted:\n");
    scanf ("%s", &unsort string);
    char tempchar;
    printf ("Entered string - %s\n", unsort string);
    for (int s=0; s<string length - 1; s++)
    {
        for (int m=s+1; m<string length; m++)
        {
            if (unsort string [s] > unsort string [m])
            {
                temp char = unsort string [s];
                unsort string [s] = unsort string [m];
                unsort string [m] = temp char;
            }
        }
    }
    printf ("String after sorting - %s\n", unsort string);
    return 0;
}
```

Q.4) What is pointer? Explain how the pointer variable is declared and initialized.

→ A pointer is a variable whose value is the address of another variable, i.e; direct address of the memory location like any variable or constant, ~~using~~ you must declare a pointer before using it to store any variable address. The general form of a pointer variable declaration is type * var-name;

Here, type is pointer's base type; it must be a valid C data type and var-name is the name of the pointer variable.

The asterisk ~~*~~ used to declare a pointer is the same asterisk used for multiplication. However, in this statement the asterisk is being used to designate a variable as a pointer. Let's take a look at some valid pointer declaration:-

```
int *ip; // Pointer to an integer
double *dp; // Pointer to a double
float *fp; // Pointer to a float
char *ch; // Pointer to a char
```

The actual data type of the value of all pointers, whether integers or characters is the same. A long hexadecimal number that represents a memory address the only difference between pointers of different data types is the data type of the variable or constant that the pointer points to.

Q.5] Explain string functions with example.

→ Strings: Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character ' \0 '.

Some of the most commonly used string functions are:

- ① strcpy
- ② strcat
- ③ strcmp
- ④ strlen

① strcpy :- strcpy() is a standard library function in C/C++ and is used to copy one string to another. In C it is present in string.h header file and in C++ it is present in cstring header file.

Program:

```

// C program to demonstrate
// strcpy() function in C/C++
#include <stdio.h>
#include <string.h>
int main ()
{
    char str1 [] = "Hello everyone!";
    char str2 [] = "Hello for ";
    char str3 [40];
    char str4 [40];
    char str5 [] = "GFG";
    strcpy (str2, str1);
    strcpy (str3, "copy successful");
    strcpy (str4, str5);
    printf ("str1: %s\nstr2: %s\nstr3: %s\nstr4: %s\n", str1, str2, str3, str4);
    return 0;
}

```

Output:

```

str1: Hello everyone!
str2: Hello everyone!
str3: copy successful
str4: GFG

```

② strcat: The `strcat()` function will append a copy of the source string to end of destination string. The `strcat()` function takes two arguments:

- 1) dest
- 2) src

It will append copy of the source string in the destination string. The terminating character at the end of dest is replaced by the first character of src.

Program:

```
// CPP Program to demonstrate
// strcat
#include <cstring>
#include <iostream>
using namespace std;
int main()
{
    char dest [50] = "This is an";
    char src [50] = "example";
    strcat (dest, src);
    cout << dest;
    return 0;
}
```

Output:

This is an example

③ strcmp: `strcmp()` is a built-in library function and is declared in `<string.h>` header file. This function takes two strings as arguments and compare these two strings lexicographically.

`strcmp()` compare the two strings lexicographically means it starts comparison character by character starting from the first character until the character in both strings are equal or a Null character is encountered.

Program:

```

    // C program to illustrate
    // strcmp() function
    #include <stdio.h>
    #include <string.h>
    int main ()
    {
        char leftStr [] = "g f g";
        char rightStr [] = "g f g";
        // using strcmp()
        int res = strcmp (leftStr, rightStr);
        if (res == 0)
            printf ("Strings are equal");
        else
            printf ("Strings are unequal");
        printf ("Value returned by strcmp() is : %d", res);
        return 0;
    }

```

Output: Strings are equal

Value returned by strcmp() is : 0

④ strlen: The strlen() function calculates the length of given string. The strlen() function is defined in `string.h` header file. It doesn't count null characters ('`\0`').

Program:

```

    // Example of strlen() function
    #include <stdio.h>
    #include <string.h>
    int main ()
    {
        char ch [] = {'g', 'e', 'e', 'k', 's', '\0'};
        printf ("Length of string is : %lu", strlen(ch));
        return 0;
    }

```

Output:

Length of string is : 5