

## ASSIGNMENT 4

NAME : ANJALI PATHAK

ROLL NO:- 06

BRANCH :- COMPUTER SCIENCE AND TECHNOLOGY

DIVISION :- C

**Question 1: Write a program to calculate power of a number.**

**Algorithm:**

**Step 1: Start**

**Step 2: Declare variable pow and i**

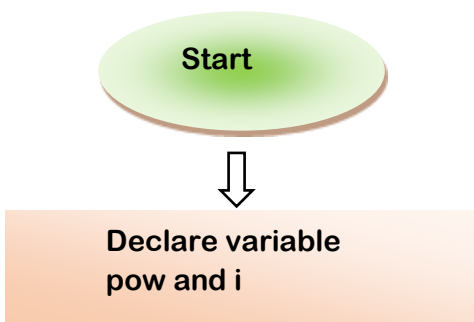
**Step 3: Initialize pow =1 and i=1**

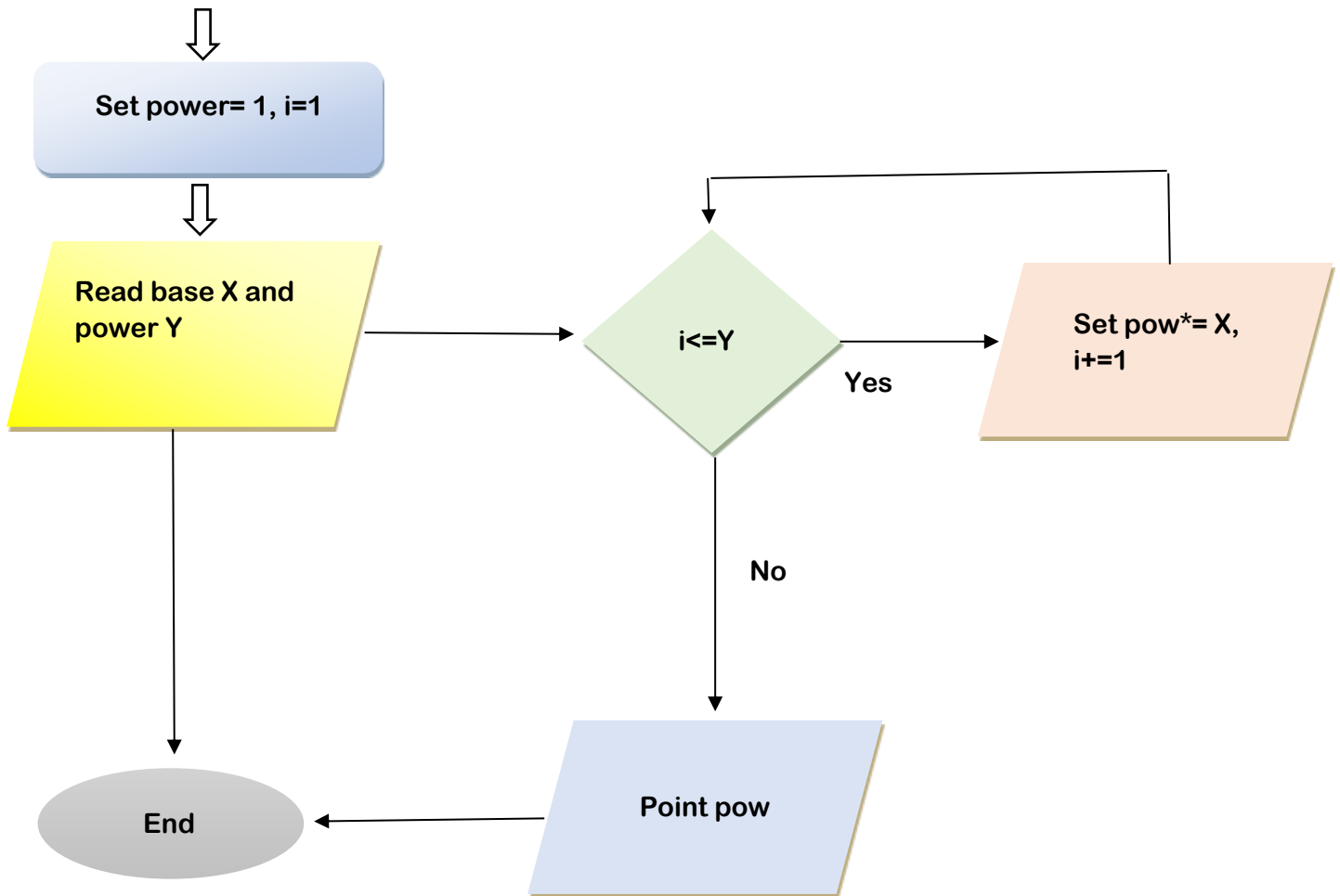
**Step 4: Read base X and power Y from user**

**Step 5: Repeat step until  $i \leq Y$  set,  $\text{pow} = \text{pow} * x$  increment the value of i by 1**

**Step 6: The value stored in pow is the required value**

**Step 7: stop**





**Program:**

```
#include <stdio.h>

int main() {
    int base, exp;
    long double result = 1.0;
    printf("Enter a base number: ");
    scanf("%d", &base);
    printf("Enter an exponent: ");
    scanf("%d", &exp);
```

```

while (exp != 0) {
    result *= base;
    --exp;
}

printf("Answer = %.0Lf", result);

return 0;
}

```

### **Output:**

Enter a base value: 2

Enter a exponent value: 3

Result - 8

**Question 2: Write a program to check whether a number is Armstrong or not.**

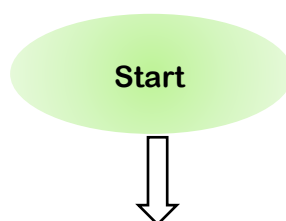
### **Algorithm:**

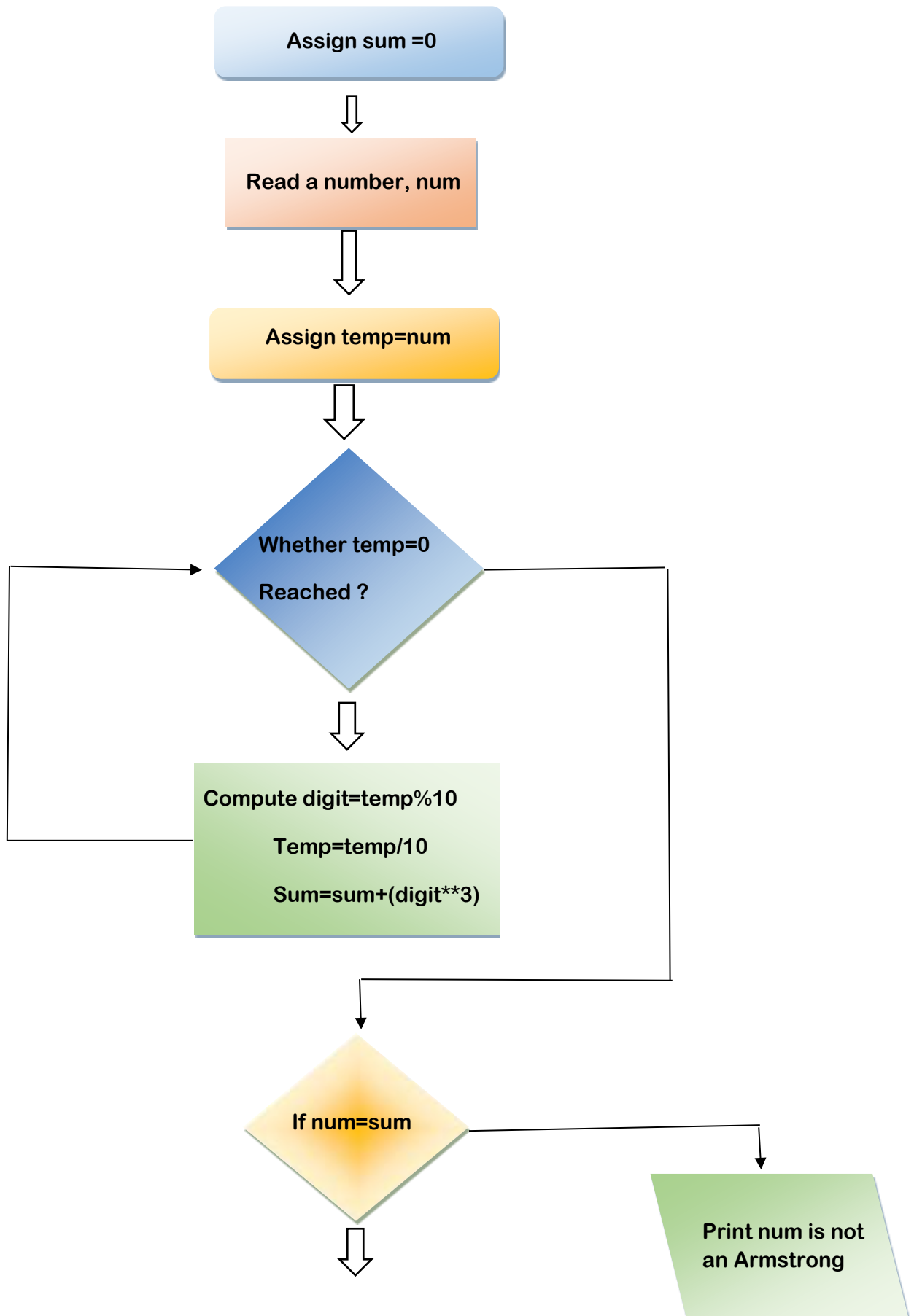
```

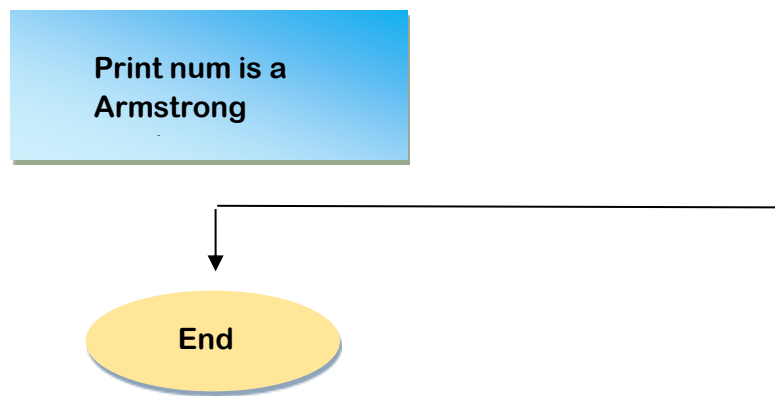
Step 1: Start
Step 2: Declare Variable sum, temp, num
Step 3: Read num from User
Step 4: Initialize Variable sum=0 and temp=num
Step 5: Repeat Until num>=0
    5.1 sum=sum + cube of last digit i.e
    [ (num%10)*(num%10)*(num%10) ]
    5.2 num=num/10
Step 6: IF sum==temp
    Print "Armstrong Number"
    ELSE
    Print "Not Armstrong Number"
Step 7: Stop

```

### **Flowchart :**







**Program-**

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

```
int main() {
```

```
    int num, originalNum, remainder, result = 0;
```

```
    printf("Enter a three-digit integer: ");
```

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

```
    originalNum = num;
```

```
    while (originalNum != 0) {
```

```
        remainder = originalNum % 10;
```

```
        result += remainder * remainder * remainder;
```

```
        originalNum /= 10;
```

```
    }
```

```
    if (result == num)
```

```
        printf("%d is an Armstrong number.", num);
```

```
    else
```

```
        printf("%d is not an Armstrong number.", num);
```

```
    return 0;
```

```
}
```

## OUTPUT-

Enter a three-digit integer: 370

370 is an Armstrong number.

**Question 3: Write the program to convert binary to decimal and decimal to binary**

### Algorithm:

Step 1: Start

Step 2: Read the binary number from the user, say 'n'

Step 3: Initialize the decimal number, d=0

Step 4: Initialize i=0

Step 5: Repeat while n != 0:

    Step 5.1: Extract the last digit by: remainder = n % 10

    Step 5.2: n = n/10

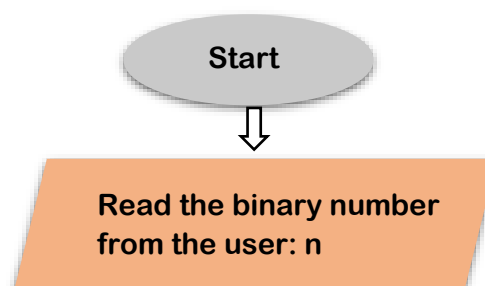
    Step 5.3: d = d + (remainder \* 2<sup>i</sup>)

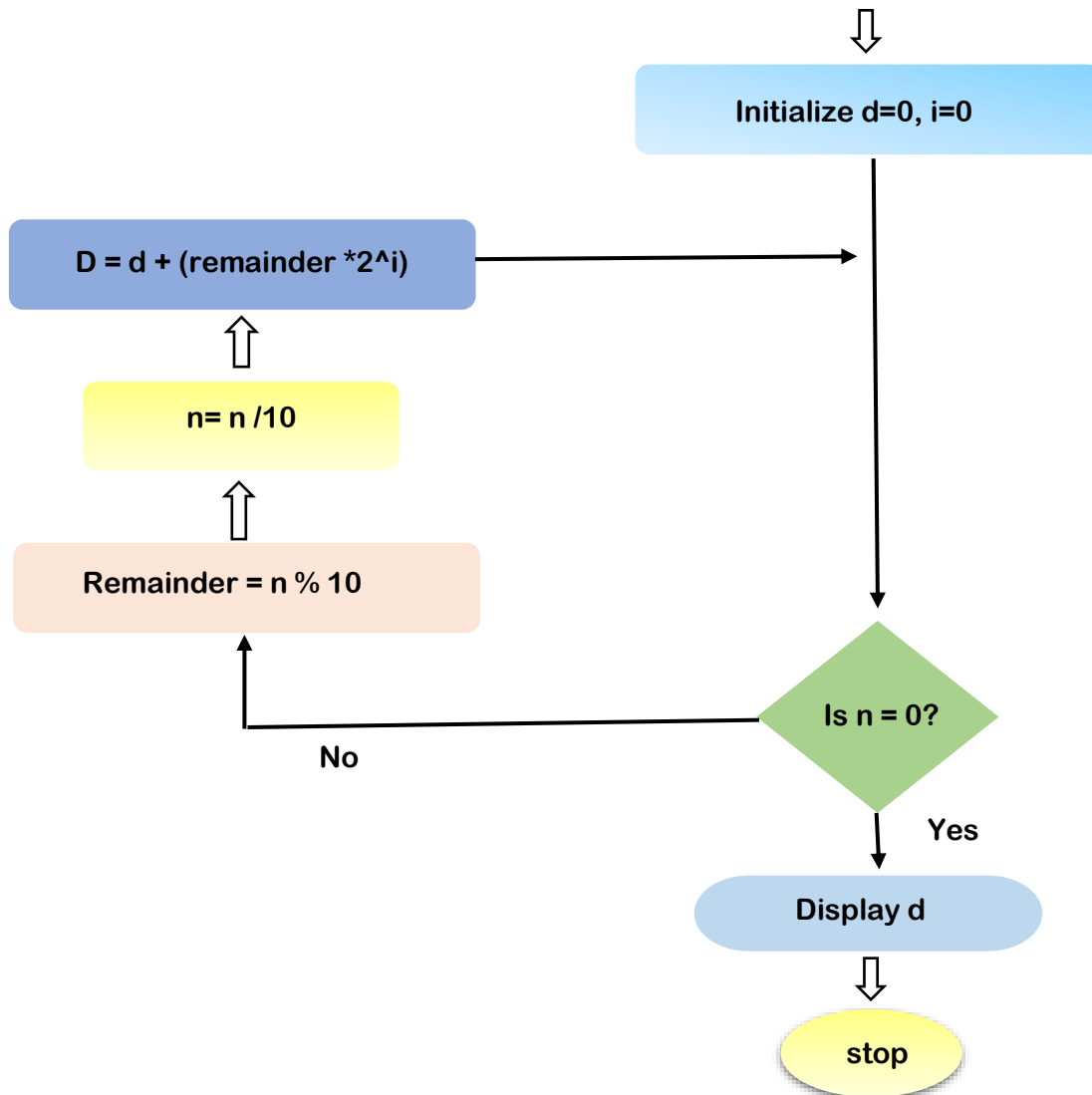
    Step 5.4: Increment i by 1

Step 6: Display the decimal number, d

Step 7: Stop

### FLOWCHART-





### Program

```
#include <math.h>
```

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

```
int convert(long long n);
```

```
int main() {
```

```
    long long n;
```

```
    printf("Enter a binary number: ");
```

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

```

printf("%lld in binary = %d in decimal", n, convert(n));

return 0;
}

```

```

int convert(long long n) {

    int dec = 0, i = 0, rem;

    while (n != 0) {

        rem = n % 10;

        n /= 10;

        dec += rem * pow(2, i);

        ++i;

    }

    return dec;

}

```

### Output

Enter a binary number: 65565

in binary = 173 in decimal

**Question 4: Write a program to make simple calculator using switch statement.**

### Algorithm

**Step 1:** Start

**Step 2:** Input two numbers and a character from user in the given format. Store them in some variable say num1, op and num2.

**Step 3:** Switch the value of op i.e. switch (op)

**Step 4:** There are four possible values of op i.e. '+', '-', '\*' and '/'



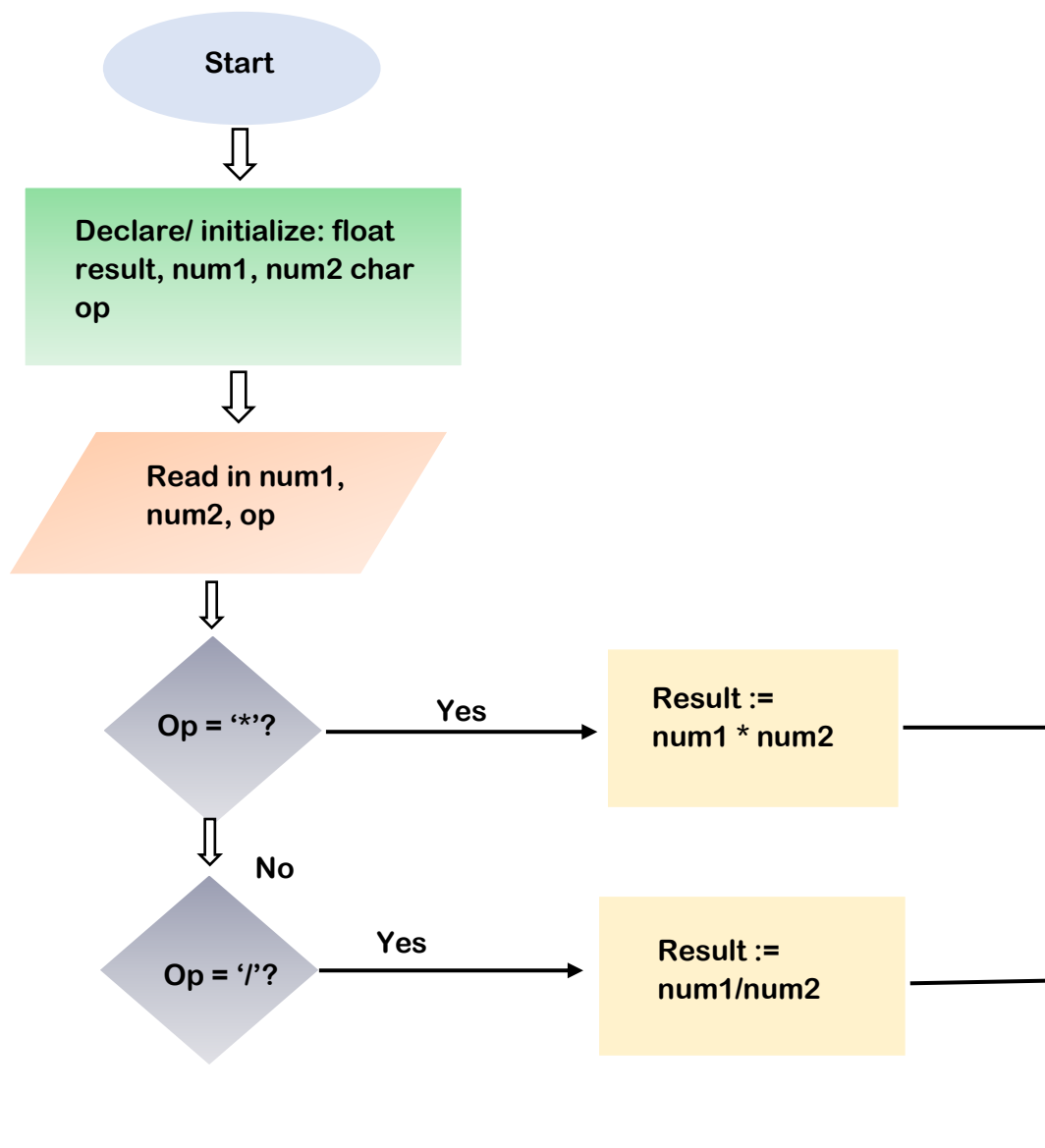
**Step 5:** for case '+' perform addition and store result in some variable i.e.  $\text{result} = \text{num1} + \text{num2}$

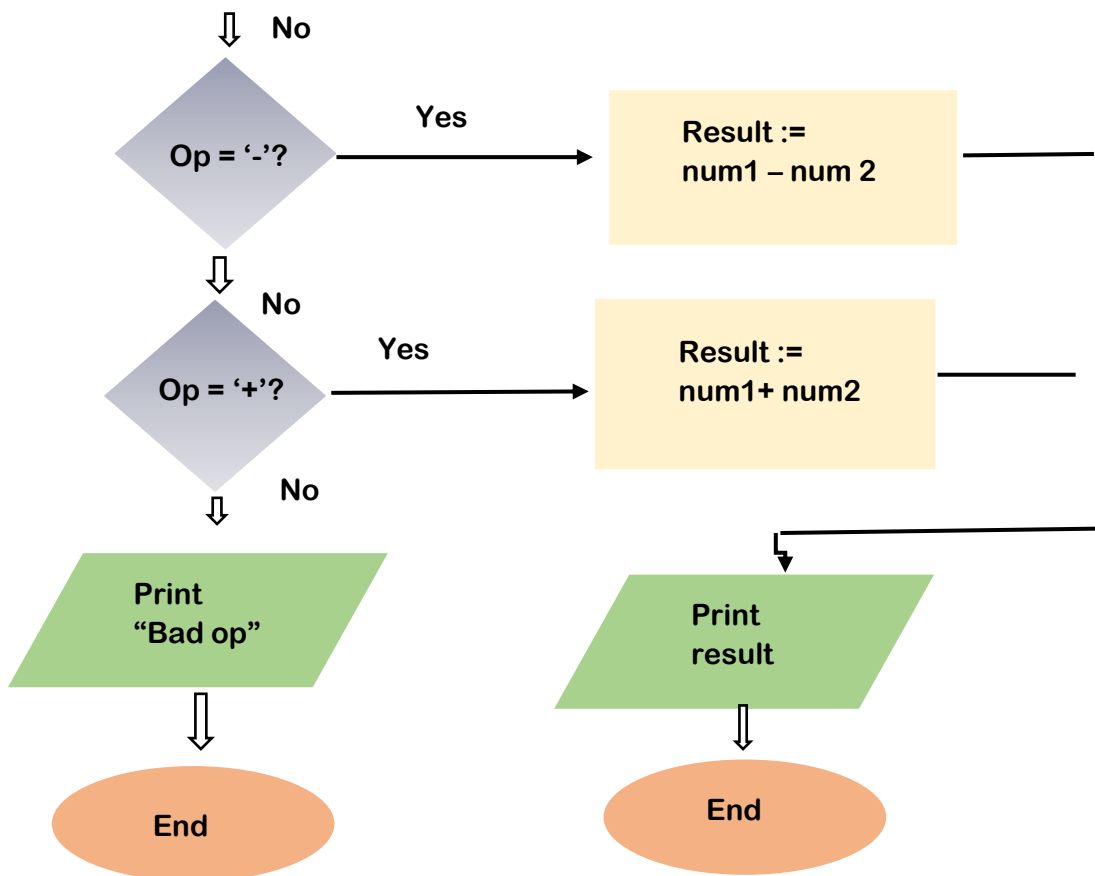
**Step 6:** Similarly for case '-' perform subtraction and store result in some variable i.e.  $\text{result} = \text{num1} - \text{num2}$

**Step 7:** Repeat the process for multiplication and division

**Step 8:** Stop

**Flowchart :**





Program-

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

```
int main() {
```

```
    char op;
```

```
    double first, second;
```

```
    printf("Enter an operator (+, -, *, /): ");
```

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

```
    printf("Enter two operands: ");
```

```
    scanf("%lf %lf", &first, &second);
```

```
    switch (op) {
```

```

case '+':

    printf("%.1lf + %.1lf = %.1lf", first, second, first + second);

    break;

case '-':

    printf("%.1lf - %.1lf = %.1lf", first, second, first - second);

    break;

case '*':

    printf("%.1lf * %.1lf = %.1lf", first, second, first * second);

    break;

case '/':

    printf("%.1lf / %.1lf = %.1lf", first, second, first / second);

    break;

default:

    printf("Error! operator is not correct");

}

return 0;

}

```

OUTPUT-

Enter an operator (+, -, \*, /): \*

Enter two operands: 5 2

5.0 \* 2.0 = 10.0

