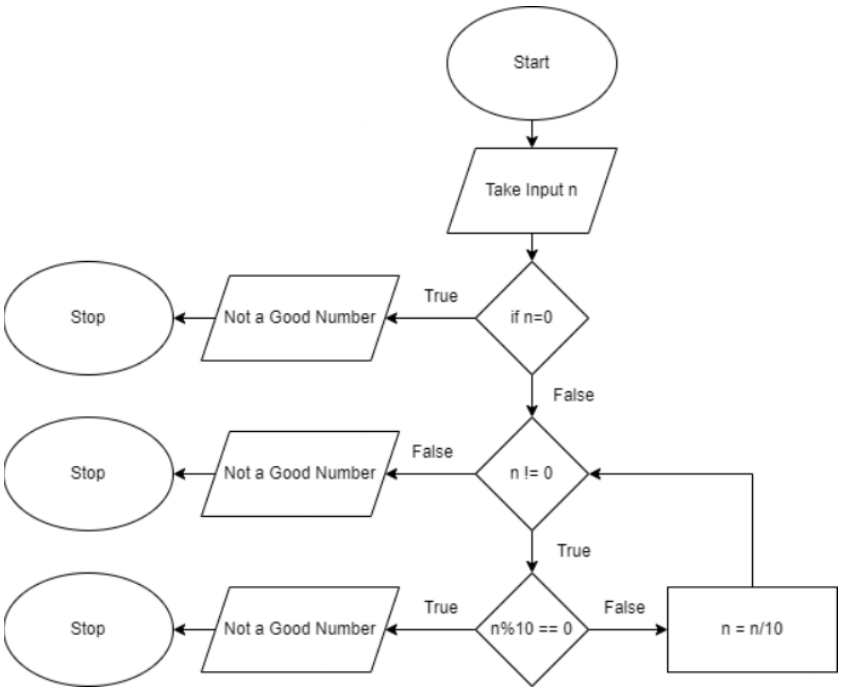
Unit 2

Q2: Good Number

Algorithm:

Step 1: Start the program
Step 2: Take input from user and store in variable n
Step 3: Check if n is zero, if not then run a loop to separate its digit by dividing it through 10.
Step 4: If remainder of n/10 is 0 then its not a Good number, break that loop.
Step 5: If the loop reaches its termination condition then it's a Good Number.
Step 6: Stop

Flowchart:



Code:

```c
#include <stdio.h>
#include <string.h>
int main()
{
    int n;
    printf("Enter the number: ");
    scanf("%d",&n);
    if(n){
        for(;n!=0;n/=10){
            if(!(n%10)){
                printf("Not a GOOD Number");
                break;
            }
        }
        if(!n) printf("GOOD Number");
    }
    else printf("Not a GOOD Number");
    return 0;
}
```

Sample Input and Output:

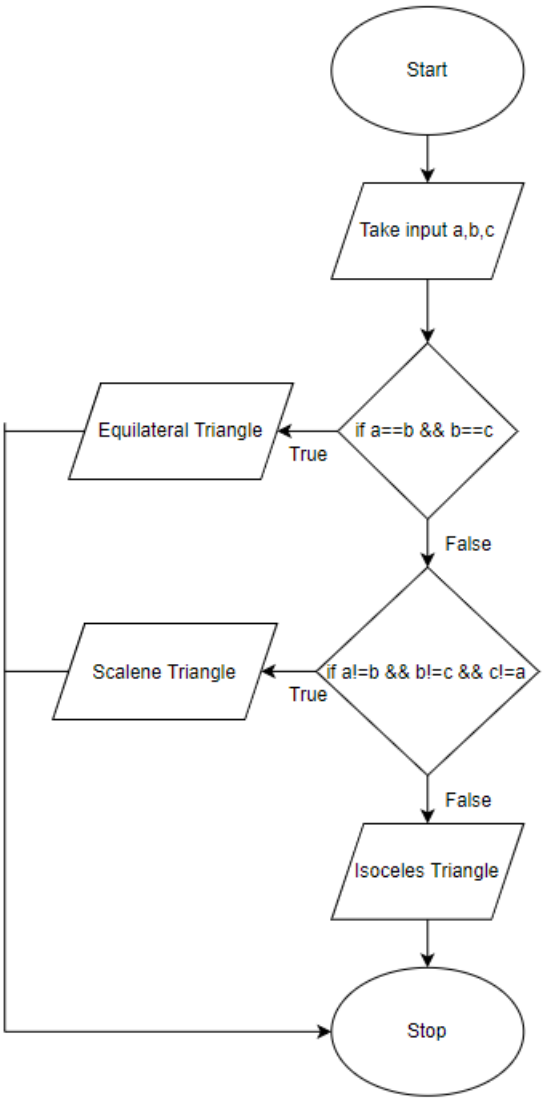
```
D:\Codes\C & C++>cd "d:\Codes\C & C
Enter the number: 3468023
Not a GOOD Number
d:\Codes\C & C++\Learn\Practice>cd
Enter the number: 33132
GOOD Number
d:\Codes\C & C++\Learn\Practice>
```

Q6: Triangle

Algorithm:

Step 1: Start the program
Step 2: Take the 3 sides from user and store in 3 variables a, b, c
Step 3: If a is equal to b and b is equal to c then its Equilateral
Step 4: If a is not equal to b and b is not equal to c and c is not equal to a then it's a Scalene Triangle
Step 5: If above condition doesn't match then it's an Isosceles Triangle.
Step 6: Stop

Flowchart:



Code:

```c
#include <stdio.h>
#include <string.h>
int main()
{
    int a,b,c;
    printf("Enter the sides off the triangle: ");
    scanf("%d %d %d", &a,&b,&c);
    if(a==b && b==c && c==a) printf("Equilateral Triangle");
    else if(a!=b && b!=c && c!=a) printf("Scalene Triangle");
    else printf("Isoceles Triangle");
    return 0;
}
```

Sample Input and Output:

```
D:\Codes\C & C++>cd "d:\Codes\C & C++\Lear
Enter the sides off the triangle: 2 2 2
Equilateral Triangle
d:\Codes\C & C++\Learn\Practice>cd "d:\Cod
Enter the sides off the triangle: 2 2 4
Isoceles Triangle
d:\Codes\C & C++\Learn\Practice>cd "d:\Cod
Enter the sides off the triangle: 1 2 3
Scalene Triangle
d:\Codes\C & C++\Learn\Practice>
```

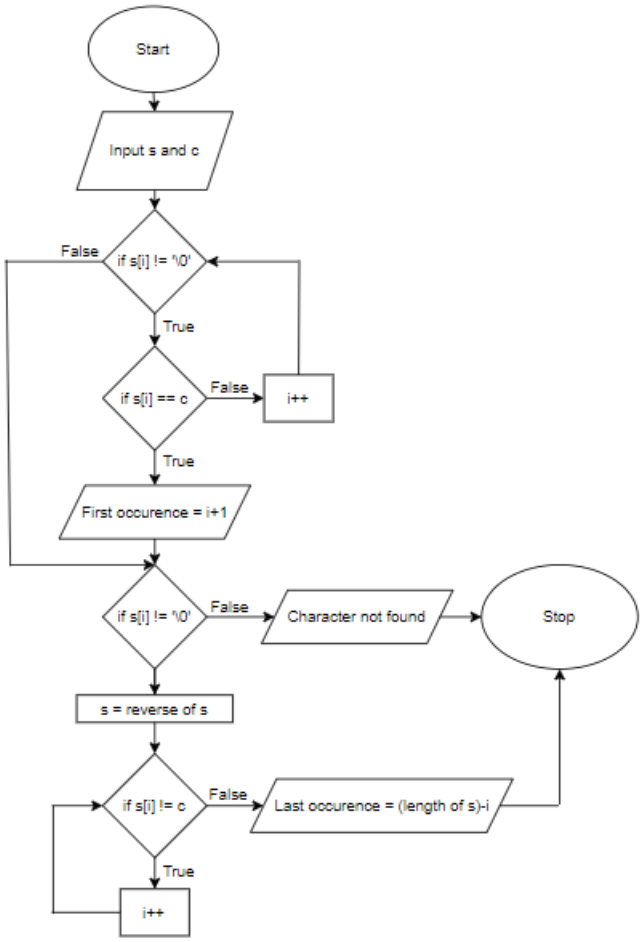
Unit 3

Q1: First and Last Occurrence of character in String

Algorithm:

Step 1: Start the program
Step 2: Scan the word and character and assign it to s and c variables
Step 3: Run a loop to the string end, if in between a character of s matches with c, display its index position and break the loop.
Step 4: Check if the loop ran till the end or not. If it did then character was not present in the word. Display appropriate message and exit the program.
Step 5: If the loop terminated in between then, reverse the string s and again run a loop to find a matching character.
Step 6: The loop terminates if it finds a matching character. Display its index position now subtracting it from length of s.
Step 7: Exit

Flowchart:



Code:

```c
#include <stdio.h>
#include <string.h>
int main(){
    int i;
    char s[10000], c;
    printf("Enter the String: ");
    scanf("%s",s);
    printf("Enter the character: ");
    scanf(" %c",&c);
    for(i=0;s[i]!='\0';i++){
        if(s[i]==c){
            printf("First Occurence at position: %d\n",i+1);
            break;
        }
    }
    if(s[i]!='\0'){
        strrev(s);
        for(i=0;s[i]!=c;i++);
        printf("Last Occurence at position: %d ",strlen(s)-i);
    }
    else printf("Character not found");
    return 0;
}
```

Sample Input and Output:

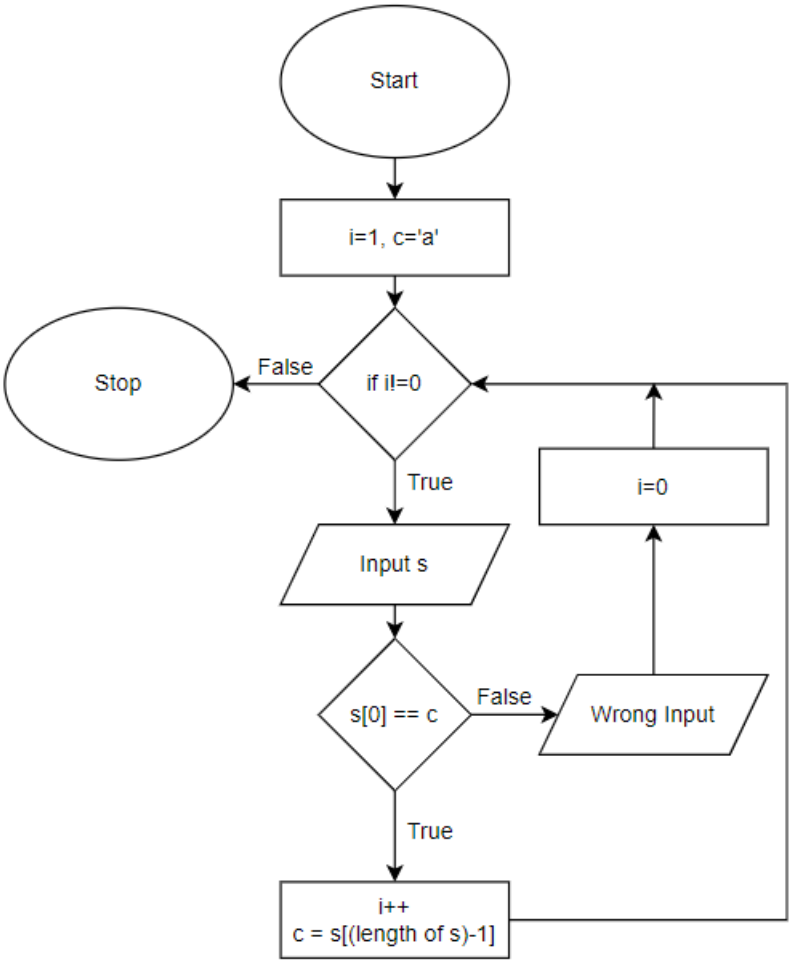
```
D:\Codes\C & C++>cd "d:\Codes\C & C++\Assi
Enter the String: avenger
Enter the character: e
First Occurence at position: 3
Last Occurence at position: 6
d:\Codes\C & C++\Assignments\Unit 3>cd "d:
Enter the String: fish
Enter the character: a
Character not found
d:\Codes\C & C++\Assignments\Unit 3>
```

Q3: John and Peter word game

Algorithm:

Step 1: Start the program
Step 2: Assign variable i to 1 and char c to ‘a’
Step 3: Run a loop based on the condition if i is not zero.
Step 4: Take input from John and Peter turn wise by checking odd and even value of i.
Step 5: If first character of the input word matches with c then assign c the last character of the input word and increment i.
Step 6: If it doesn’t match then the word entered was wrong. Assign i to 0 to falsify the loop condition, to terminate it.
Step 7: Stop

Flowchart:



Code:

```c
#include <stdio.h>
#include <string.h>
int main(){
    char c='a',s[100];
    int i=1;
    while(i){
        printf("%s's turn: ",(i%2)? "John" : "Peter");
        scanf("%s",s);
        if(s[0]==c){
            c=s[strlen(s)-1];
            i++;
        }
        else{
            printf("Wrong word, %s lost",(i%2)? "John" : "Peter");
            i=0;
        }
    }
    return 0;
}
```

Sample Input and Output:

```
D:\Codes\C & C++>cd "d:\Codes\C & C++\A
John's turn: apple
Peter's turn: elephant
John's turn: tomato
Peter's turn: oral
John's turn: bus
Wrong word, John lost
d:\Codes\C & C++\Assignments\Unit 3>
```