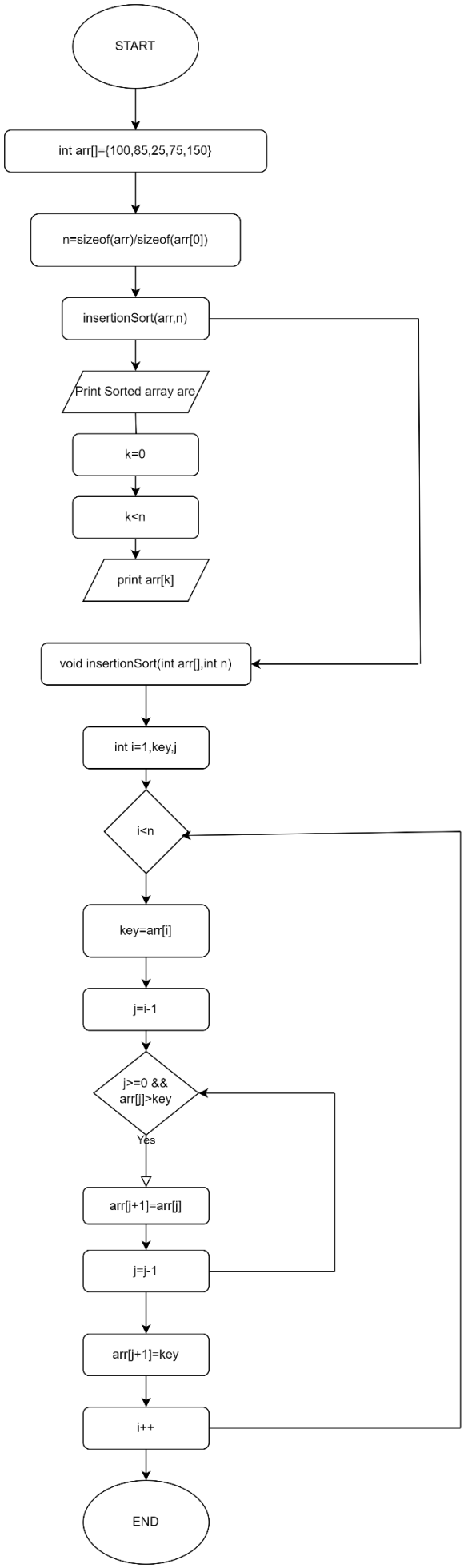
4.Pseudocode and Flowchart for Sorting Algorithm - Write pseudocode and create a flowchart for a insertion sort algorithm. Provide a brief explanation of how the algorithm works and a simple array of integers to demonstrate a dry run of your algorithm.use following "50 ,100, 85, 25, 75, 150"

PSEUDOCODE:

```
BEGIN

Intialize array arr=[100,85,25,75,150]

Find the length of array by using n=sizeof(arr)/sizeof(arr[0])

FUNCTION CALL insertionSort(arr,n)

DISPLAY sorted array are …

FOR k from 0 to less than n  increment by 1

        DISPLAY arr[k]

FUNCTION void insertionSort(int arr[],int n)

DECALRE i,key,j

FOR I from 0 to lessthan n increment by 1

        WHILE j>=0 && arr[j] > key

                arr[j+1]=arr[j]

                j=j-1

        END WHILE

        arr[j+1]=key

END FOR

END
```

INSERTION SORT:

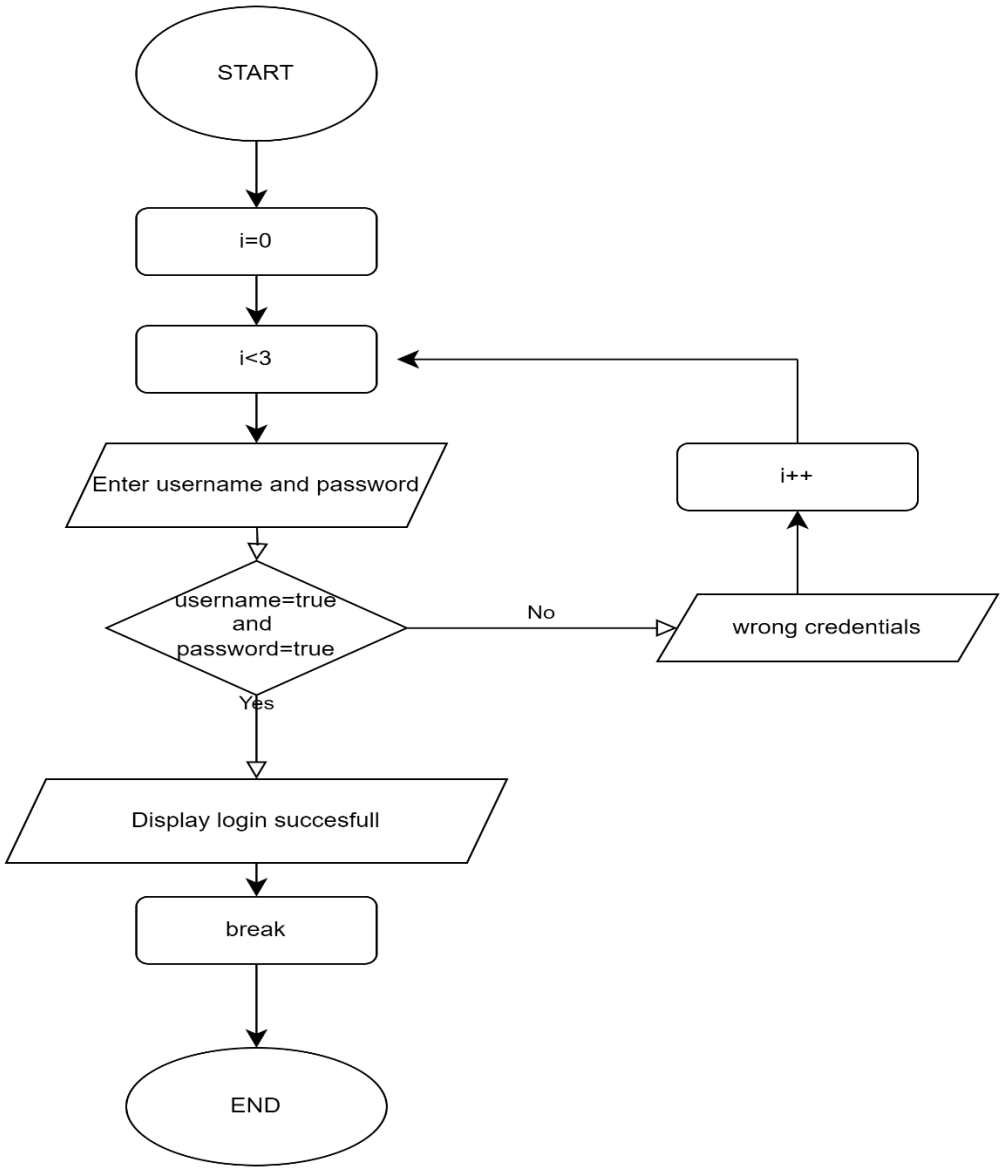


ASSIGNMENT

1. Assignment 1: Pseudocode Development - Task: Write a detailed pseudocode for a simple program that takes a number as input, calculates the square if it's even or the cube if it's odd, and then outputs the result. Incorporate conditional and looping constructs.

```
BEGIN
GET  num
IF num%2==0
      Result=num*num
ELSE
      Result=num*num*num
ENDIF
DISPLAY Result
```

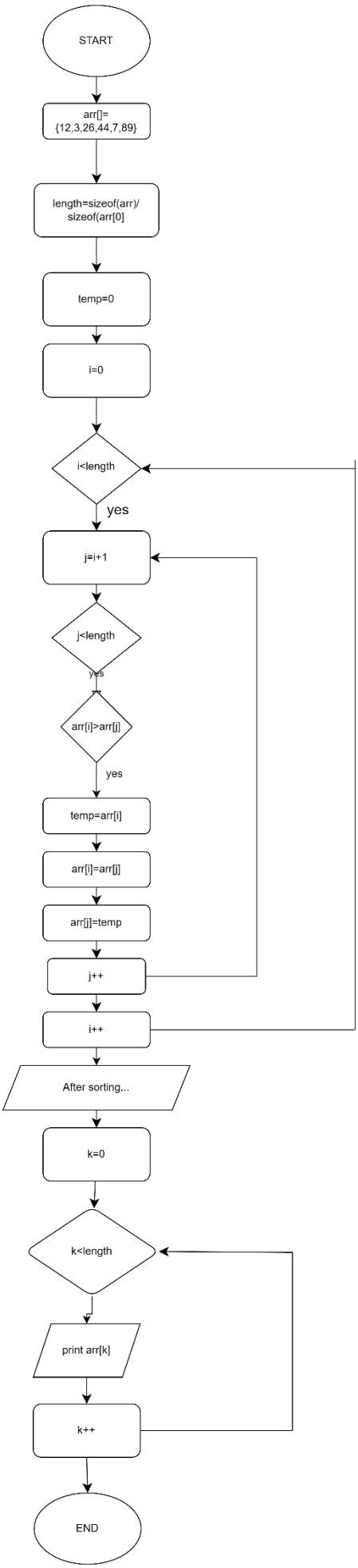
2. Assignment 2: Flowchart Creation - Design a flowchart that outlines the logic for a user login process. It should include conditional paths for successful and unsuccessful login attempts, and a loop that allows a user three attempts before locking the account.



3.

Pseudocode and Flowchart for Sorting Algorithm - Write pseudocode and create a flowchart for a bubble sort algorithm. Provide a brief explanation of how the algorithm works and a simple array of integers to demonstrate a dry run of your algorithm

```
PSEUDOCODE
BEGIN
GET THE arr={12,3,26,44,7,89)
Length=size of arr /size of arr[0]
Intialize temp=0,i=0
For i to length increment by 1
                J=i+1
                For j to length increment by 1
                   IF arr[i]>arr[j]
                            temp = arr[i]
                            arr[i]=arr[j]
                            arr[j]=temp
                   END IF
                   END FOR
ENF FOR
DISPLAY After sorting ….
FOR k to length increment by 1
                DISPLAY arr[k]
END FOR
END
```


START
arr[]= {12,3,26,44,7,89}
length=sizeof(arr)/ sizeof(arr[0]
temp=0
i=0
i<length
yes
j=i+1
j<length
yes
arr[i]>arr[j]
yes
temp=arr[i]
arr[i]=arr[j]
arr[j]=temp
j++
i++
After sorting...
k=0
k<length
print arr[k]
k++
END

4. Write a program that takes two integers name num1 and num2 as input from the user andpasses them to a function named as doublenumber(). The function should double the values of the integers passed to it and print the modified values inside the function. After the function call, print the original values of the integers in the main function

rps@rps-virtual-machine: ~/Desktop/charan/day4

```c
#include<stdio.h>
void doublenumber(int* a,int* b);
int main()
{
        int num1,num2;
        printf("\nEnter num1\n");
        scanf("%d",&num1);
        printf("\nEnter num2\n");
        scanf("%d",&num2);
        printf("Before doubled the numbers are %d and %d\n",num1,num2);
        doublenumber(&num1,&num2);
        printf("After doubled the numbers are %d and %d\n",num1,num2);
}
void doublenumber(int* a,int* b)
{
        *a=2*(*a);
        *b=2*(*b);

}
```

"doublethenum.c" 19L, 397B 10,58-65 Al

OUTPUT

rps@rps-virtual-machine: ~/Desktop/charan/day4

```
rps@rps-virtual-machine:~/Desktop/charan/day4$ gcc doublethenum.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ ./a.out

Enter num1
2

Enter num2
4
Before doubled the numbers are 2 and 4
After doubled the numbers are 4 and 8
rps@rps-virtual-machine:~/Desktop/charan/day4$
```

5.Write a program that takes an integer as input from the user and passes it to a function by reference using a pointer. The function should square the value of the integer passed to it and modify the original value using the pointer. Print the modified value inside the function and also print the modified value in the main function after the function call.

rps@rps-virtual-machine: ~/Desktop/charan/day4/pointers

```c
#include<stdio.h>
int add(int* fa)
{
        int s=(*fa)*(*fa);
        printf("\n a value in add method is : %d",s);
        return s;

}
void main()
{
        int a;
        scanf("\n%d",&a);

        printf("\n value of a in main is %d\n",add(&a));
}

```

OUTPUT

rps@rps-virtual-machine: ~/Desktop/charan/day4/pointers

```
rps@rps-virtual-machine:~/Desktop/charan/day4/pointers$ gcc first.c
rps@rps-virtual-machine:~/Desktop/charan/day4/pointers$ ./a.out
4

 a value in add method is : 16
 value of a in main is 16
rps@rps-virtual-machine:~/Desktop/charan/day4/pointers$
```

6.Write a program to add 10 elements of an array using pointers

rps@rps-virtual-machine: ~/Desktop/charan/day4

```c
#include<stdio.h>
int main()
{
        int arr[10];
        int* ptr=arr;
        int sum=0;
        printf("Enter 10 elements of the array:\n");
        for(int i=0;i<10;i++)
        {
                scanf("%d",ptr+i);
        }
        for(int j=0;j<10;j++)
        {
                sum=sum+(*ptr+j);
        }
        printf("sum :%d",sum);
}
```

"addArrayusingPointer.c" 17L, 243B

OUTPUT

```
rps@rps-virtual-machine: ~/Desktop/charan/day4
rps@rps-virtual-machine:~/Desktop$ cd charan
rps@rps-virtual-machine:~/Desktop/charan$ cd day4
rps@rps-virtual-machine:~/Desktop/charan/day4$ vim addArrayusingPointer.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ gcc  addArrayusingPointer.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ ./a.out
Enter 10 elements 0f the array:
1
2
3
4
5
6
7
8
9
10
rps@rps-virtual-machine:~/Desktop/charan/day4$
```

7.a) In the main function, create an array of integers, pass it to the function, .return anything and print the reversed array

```
rps@rps-virtual-machine: ~/Desktop/charan/day4
#include<stdio.h>
void reverseArray(int* a,int l);
int main()
{
        int arr[]={1,3,5,7};
        int length=sizeof(arr)/sizeof(arr[0]);
        printf("Before sorting...");
        for(int i=0;i<length;i++)
        {
                printf("\n%d ",arr[i]);
        }
        reverseArray(arr,length);
}
void reverseArray(int* a,int l)
{
        printf("\nAfter sorting...");
        for(int i=(l)-1;i>=0;i--)
        {
                printf("\n%d ",a[i]);
        }

}

"arrayIntegers.c" 23L, 371B
```

OUTPUT

```
rps@rps-virtual-machine:~/Desktop$ cd charan
rps@rps-virtual-machine:~/Desktop/charan$ cd day4
rps@rps-virtual-machine:~/Desktop/charan/day4$ vim arrayIntegers.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ gcc  arrayIntegers.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ ./a.out
Before sorting...
1
3
5
7
After sorting...
7
5
3
1 rps@rps-virtual-machine:~/Desktop/charan/day4$
```

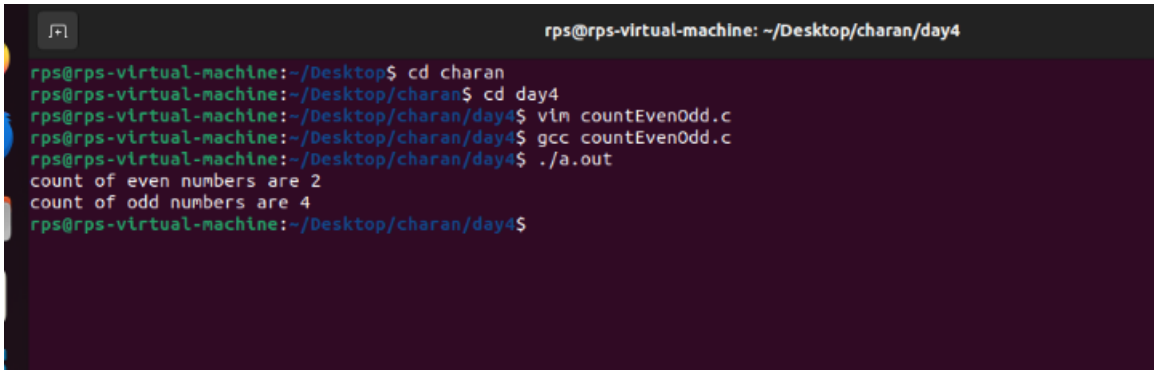
7b)Count Even and Odd Numbers:

Write a program to count the number of even and odd numbers in an array. Implement a function that takes an integer array, its size, and two pointers to integers for storing the counts of even and odd numbers. The function calculates the counts and updates the pointers accordingly. In the main function, create an array of integers, pass it to the function along with the pointers, and print the counts of even and odd numbers

rps@rps-virtual-machine: ~/Desktop/charan/day4

```c
#include<stdio.h>
void countEvenOdd(int* p,int l ,int *x,int *y);
void main()
{
        int arr[]={1,13,21,5,34,62};
        int length=sizeof(arr)/sizeof(arr[0]);
        int m=0,n=0;
        countEvenOdd(arr,length,&m,&n);
        printf("count of even numbers are %d\n",m);
        printf("count of odd numbers are %d\n",n);
}
void countEvenOdd(int* p,int l,int *x,int *y)
{
        for(int i=0;i<l;i++)
        {
                if(*(p+i)%2==0)
                {
                        (*x)++;

                }
                else
                {
                        (*y)++;

                }
        }

}
```

"countEvenOdd.c" 28L, 435B


rps@rps-virtual-machine: ~/Desktop/charan/day4
rps@rps-virtual-machine:~/Desktop$ cd charan
rps@rps-virtual-machine:~/Desktop/charan$ cd day4
rps@rps-virtual-machine:~/Desktop/charan/day4$ vim countEvenOdd.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ gcc countEvenOdd.c
rps@rps-virtual-machine:~/Desktop/charan/day4$ ./a.out
count of even numbers are 2
count of odd numbers are 4
rps@rps-virtual-machine:~/Desktop/charan/day4$