Day-42 of the #101 days of coding challenge-------

Problem:- Write a C++ program to separate 0s and 1s from a given array of values 0 and 1.

1st Solution:- By sorting elements

```
#include<iostream>

using namespace std;

void swapping(int *a, int *b)
{
	int temp;
	temp = *a;
	*a = *b;
	*b = temp;
}

void sorting(int *arr, int n)
{
	int i, j;
	for(i = 0; i<n; i++)
	{
		for(j = 0; j<n-i-1; j++)
		{
			if(arr[j]>arr[j+1])
			{
```

```
                    swapping(&arr[j], &arr[j+1]);
                }
            }
        }
}
void arrangingZeroOnes(int *arr, int n)
{
        sorting(arr, n);
}
int main()
{
        int n,i;
        cout<<"Enter the Size array"<<endl;
        cin>>n;
        int arr[n];

        cout<<"Enter the elements"<<endl;
    for(i = 0; i<n; i++)
    {
        cin>>arr[i];
        }
        arrangingZeroOnes(arr,n);

        cout<<"Solved"<<endl;
        for(i = 0; i<n; i++)
        {
```

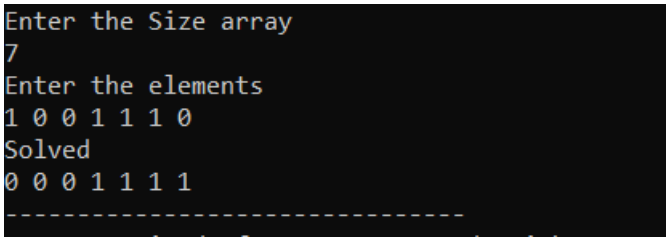
```
            cout<<arr[i]<<" ";
        }

        return 0;
}
```

2nd Solution:- Without Sorting-

```
#include<iostream>
using namespace std;

void arrangingZeroOnes(int *arr, int n)
{
        int i, countZero = 0;

        // counting the number of zeroes
        for(i = 0; i<n; i++)
        {
                if(arr[i] == 0)
                  countZero++;
        }

        // after counting, looping it till numberofzero's available giving value 0
        for(i = 0; i<countZero; i++)
           arr[i] = 0;

        // in remaining setting the value 1 (n - currentZero = remaining space)
        for(i = countZero; i<n; i++)
```

```
            arr[i] = 1;
}

int main()
{
        int n,i;
        cout<<"Enter the Size array"<<endl;
        cin>>n;
        int arr[n];

        cout<<"Enter the elements"<<endl;
    for(i = 0; i<n; i++)
    {
        cin>>arr[i];
        }
        arrangingZeroOnes(arr,n);

        cout<<"Solved"<<endl;
        for(i = 0; i<n; i++)
        {
                cout<<arr[i]<<" ";
        }

        return 0;
}
```

Output:-



2-Problem:-

Write a C++ program to rearrange a given sorted array of positive integers. Note: In final array, first element should be maximum value, second minimum value, third second maximum value, fourth second minimum value, fifth third maximum and so on.

Code:-

```
//Original array: 0 1 3 4 5 6 7 8 10

//Array elements after rearranging: 10 0 8 1 7 3 6 4 5

#include<iostream>

using namespace std;


void arrangingMaxMin(int *arr, int n)

{

	int i, j = 0, val = 1;

	int *arrCollection = new int[n]; // allocating the memory for the array size of n


	// in this loop shifting the last elements into the begining after one elements gap

	for(i = 0; i<n / 2; i++)

	{
```

```
            arrCollection[j] = arr[n-val]; // last element to the first
            j++;
            arrCollection[j] = arr[i]; // then firstElement after last element of the
array
            j++;
            val++;
        }

        if(n%2!=0)
          arrCollection[j] = arr[n / 2];

    // coping the elements from one array to another array
    for(i = 0; i<n; i++)
    {
        arr[i] = arrCollection[i]; // if odd then middle element will be remaining so
this is setting into the last position
        }

        delete[] arrCollection; // after successfully completion of the code
deleting the created memeory
}

int main()
{
        int n,i;
        cout<<"Enter the Size array"<<endl;
        cin>>n;
```

```
        int arr[n];


        cout<<"Enter the elements"<<endl;
    for(i = 0; i<n; i++)
    {
        cin>>arr[i];
        }
        arrangingMaxMin(arr,n);


        cout<<"Solved"<<endl;
        for(i = 0; i<n; i++)
        {
                cout<<arr[i]<<" ";
        }
        return 0;
}
```

Output:-