

## 1. [Smallest Positive missing number](#)

```
int firstMissingPos(int A[], int n)
{
    bool present[n + 1] = {false};
    for (int i = 0; i < n; i++)
    {
        if (A[i] > 0 && A[i] <= n)
            present[A[i]] = true;
    }
    for (int i = 1; i <= n; i++)
        if (!present[i])
            return i;
    return n + 1;
}
```

## 2. [Move all negative elements to end](#)

```
void segregateElements(int arr[], int n)
{
    int temp[n];

    int j = 0;
    for (int i = 0; i < n; i++)
        if (arr[i] >= 0)
            temp[j++] = arr[i];
    if (j == n || j == 0)
        return;
    for (int i = 0; i < n; i++)
        if (arr[i] < 0)
            temp[j++] = arr[i];
    for (int i = 0; i < n; i++)
    {
        arr[i] = temp[i];
    }
}
```

### 3. [Number of occurrence](#)

```
int count(int arr[], int n, int x)
{
    // code here
    int count = 0;
    for (int j = 0; j < n; j++)
    {
        if (arr[j] == x)
        {
            for (int i = j; i < n; i++)
            {
                if (arr[i] == x)
                    count++;
                else
                {
                    return count;
                }
            }
            return count;
        }
    }
    // if x is not found in array then it will return 0
    return count;
}
```

### 4. [Count number of elements between two given elements in array](#)

```
int getCount(int arr[], int n, int num1, int num2)
{
    int i = 0;
    for (i = 0; i < n; i++)
        if (arr[i] == num1)
            break;
    if (i >= n - 1)
        return 0;
    int j;
    for (j = n - 1; j >= i + 1; j--)
        if (arr[j] == num2)
            break;
    if (j == i)
```

```

        return 0;
    return (j - i - 1);
    return 0;
}

```

## 5. [First Repeating Element](#)

```

int firstRepeatingElement(int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (arr[i] == arr[j])
            {
                return i;
            }
        }
    }
    return -1;
}

```

## 6. [Sum of Unique Elements](#)

```

int sumOfUnique(vector<int>& nums)
{
    int n=nums.size();
    // the element is between 1 to 1000
    int arr[1005]={0};
    for(int i=0; i<n; i++)
    {
        arr[nums[i]]++;
    }
    int sum=0;
    for(int i=0; i<n; i++)
    {
        if(arr[nums[i]]==1) sum+=nums[i];
    }
    return sum;
}

```

Note - actually in some problem i have just shared simple approach where some extra space is used , if someone who wants more optimize solution ..let me know