	Problem Statement

1.	**Easy Level-Find the Duplicate Number.**
	
	Code:
	#include <iostream>
	#include <bits/stdc++.h>
	using namespace std;
	
	int dupl(vector<int>&num)
	{
	int n=num.size();
	unordered_map<int,int>m;
	for(int i=0;i<n;i++)
	{
	m[num[i]]++;
	if(m[num[i]]>1)
	return num[i];
	}
	return 0;
	
	}
	int main()
	{
	vector<int>num={1,3,4,2,2};
	
	cout<<dupl(num)<<endl;
	return 0;
	}

122.	**1.Easy level- Sort an array of 0s, 1s and 2s** **Code:** #include <bits/stdc++.h> using namespace std; void sort(int a[], int n)

```
{
   int lo = 0;
   int hi = n - 1;
   int mid = 0;


   while (mid <= hi) {
      switch (a[mid]) {


      case 0:
         swap(a[lo++], a[mid++]);
         break;


      case 1:
         mid++;
         break;

      case 2:
         swap(a[mid], a[hi--]);
         break;
      }
   }
}


void printArray(int arr[], int n)
{

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


int main()
{
   int arr[] = { 0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1 };
   int n = sizeof(arr) / sizeof(arr[0]);
```

```
    sort(arr, n);



    printArray(arr, n);

    return 0;
}
```

3. **Easy level-3 Remove Duplicates from Sorted Array.**

Code:

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int removeDuplicates(vector<int>& nums)
{
    set<int>s;
    for(int i=0;i<nums.size();i++)
    {
        s.insert(nums[i]);
    }
    int k=0;
    int p=s.size();
    for(auto it:s)
    {
        nums[k]=it;
        k++;
    }
    return p;
}
int main()
{
    vector<int>nums={0,0,1,1,1,2,2,3,3,4};
    cout<<removeDuplicates(nums)<<endl;



    return 0;
```

	}

4.	**Easy level-4 Set Matrix Zeroes**
	Code:
	#include <iostream>
	#include <bits/stdc++.h>
	using namespace std;
	
	void setZeroes(vector<vector<int>>& matrix) {
	int m=matrix.size(), n=matrix[0].size();
	
	bool col=true, row=true;
	for(int i=0; i<m; i++)
	for(int j=0; j<n; j++)
	if(matrix[i][j]==0){
	if(i==0)
	row = false;
	if(j==0)
	col = false;
	matrix[0][j]=0;
	matrix[i][0]=0;
	}
	
	for(int i=1; i<m; i++)
	for(int j=1; j<n; j++)
	if(matrix[0][j]==0 || matrix[i][0]==0)
	matrix[i][j]=0;
	
	if(col==false)
	for(int i=0; i<m; i++)
	matrix[i][0]=0;
	if(row==false)
	for(int j=0; j<n; j++)
	matrix[0][j]=0;
	}
	int main()
	{
	vector<vector<int>>matrix={{1,1,1},{1,0,1},{1,1,1}};
	setZeroes(matrix);

```
    for (int i = 0; i < matrix.size(); i++) {
    for (int j = 0; j < matrix[0].size(); j++) {
      cout << matrix[i][j] << " ";
    }
    cout<<"\n";

}
    return 0;
}
```

5. **Easy level-5 Move Zeroes**

Code:

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

 void reorder(int A[], int n)
{

    int k = 0;


    for (int i = 0; i < n; i++)
    {

        if (A[i] != 0) {
            A[k++] = A[i];
        }
    }


    for (int i = k; i < n; i++) {
        A[i] = 0;
    }
}

int main(void)
{
    int A[] = { 6, 0, 8, 2, 3, 0, 4, 0, 1 };
    int n = sizeof(A) / sizeof(A[0]);
```

```
    reorder(A, n);

    for (int i = 0; i < n; i++) {
        printf("%d ", A[i]);
    }

    return 0;
}
```

6. **Best Time to Buy and Sell Stock**

Code:

```
#include <bits/stdc++.h>
#include <iostream>

using namespace std;
int maxprofit(int a[],int n)
{
    int pro=0;
    for(int i=0;i<n-1;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            int profit=a[j]-a[i];
            if(profit>pro)
            pro=profit;
        }
    }
    return pro;
}
int main()
{
    int a[]={7,1,5,3,6,4};
    int n=sizeof(a)/sizeof(a[0]);
    cout<<maxprofit(a,n);
    return 0;
}
```

7.	**Chocolate Distribution Problem**
	Code:
	#include <bits/stdc++.h>
	#include <iostream>
	
	using namespace std;
	int minimumdistribution(int a[],int n,int m)
	{
	if(m==0 || n==0)
	return 0;
	sort(a,a+n);
	if(n<m)
	return -1;
	int mini=INT_MAX;
	for(int i=0;i+m-1<n;i++)
	{
	int diff=a[i+m-1]-a[i];
	if(diff<mini)
	mini=diff;
	}
	return mini;
	}
	int main()
	{
	int a[]={7, 3, 2, 4, 9, 12, 56};
	
	int n=sizeof(a)/sizeof(a[0]);
	int m=3;
	cout<<minimumdistribution(a,n,m);
	return 0;
	}

8. **Two Sum**

Code:

```
#include <bits/stdc++.h>
#include <iostream>


using namespace std;
int sumoftwo(int a[],int n,int target)
{
   for(int i=0;i<n;i++)
   {
      for(int j=i+1;j<n;j++)
      {
         if(a[i]+a[j]==target)
         cout<<"a[i]= "<<i<<" "<<"a[j]= "<<j<<endl;
      }
   }
   return 0;
}
int main()
{
   int a[]={2,7,11,15};
   int n=sizeof(a)/sizeof(a[0]);
```

```
    int target=9;
    cout<<sumoftwo(a,n,target);
    return 0;
}
```

9. Best Time to Buy and Sell Stock II

Code:

```
#include <bits/stdc++.h>
#include <iostream>

using namespace std;
 int maxProfit(int prices[],int n) {
        int diff=0;
        for(int i=1;i<n;i++)
        {

           if(prices[i]>prices[i-1])
           {
              diff=diff+prices[i]-prices[i-1];
           }
        }
```

```
        return diff;
    }
int main()
{
    int prices[]={7,1,5,3,6,4};
    int n=sizeof(prices)/sizeof(prices[0]);


    cout<<maxProfit(prices,n);
    return 0;
}
```