NAME: Roshan Vijey K C

DEPT: EEE

1.Buy and Sell stocks

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

class Solution{
public:
   vector<vector<int> > stockBuySell(vector<int> A, int n){
      int b = 0;
      vector<vector<int>>ans;
      for(int i = 1;i<n;i++){
         if(A[i] <= A[i-1]){
            if(b!=i-1){
               ans.push_back({b,i-1});
            }
            b = i;
         }
      }
      if(A[n-1] > A[b]){
         ans.push_back({b,n-1});
      }
      return ans;
   }
};

int check(vector<vector<int>> ans, vector<int> A, int p)
{
   int c = 0;
   for(int i=0; i<ans.size(); i++)
      c += A[ans[i][1]]-A[ans[i][0]];
```

```
    return (c==p) ? 1 : 0;
}

int main()
{
    int t;
    cin>>t;
    while(t--){
        int n;
        cin>>n;
        vector<int> A(n);
        for (int i=0; i<n; ++i){
            cin>>A[i];
        }
        Solution ob;
        vector<vector<int> > ans = ob.stockBuySell(A, n);
        int p = 0;
        for(int i=0; i<n-1; i++)
        {
            int x = A[i+1]-A[i];
            if(x>0)
                p += x;
        }
        if(ans.size()==0)
            cout<<"No Profit";
        else{
            cout<<check(ans,A,p);
        }cout<<endl;
    }
    return 0;
}
```

OUTPUT:

865

Complexity:

Time: O(m+n)

Space:O(m+n)

2.Coin Change

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

class Solution {
  public:
   int dfs(int i,int sum,vector<int>&coins,vector<vector<int>>&dp){
      if(sum == 0){
         return 1;
      }
      if(i>=coins.size()){
         return 0;
      }
      if(dp[i][sum] != -1){
         return dp[i][sum];
      }
      if(coins[i] <= sum){
         return dp[i][sum] = dfs(i,sum-coins[i],coins,dp)+dfs(i+1,sum,coins,dp);
      }
      else{
         return dp[i][sum] = dfs(i+1,sum,coins,dp);
      }
   }
   int count(vector<int>& coins, int sum) {
      // code here.
      vector<vector<int>>dp(coins.size(),vector<int>(sum+1,-1));
      return dfs(0,sum,coins,dp);
```

```
    }
};



int main() {

    int t;
    cin >> t;
    cin.ignore();
    while (t--) {
        vector<int> arr;
        string input;
        getline(cin, input);
        stringstream ss(input);
        int number;
        while (ss >> number) {
            arr.push_back(number);
        }
        int sum;
        cin >> sum;
        cin.ignore();
        Solution ob;
        cout << ob.count(arr, sum) << endl;
    }

    return 0;
}
```

OUTPUT

3

Complexity:

Time:O(n*m)

Space:O(n)

3.First and Last occurrence of element

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


class Solution {
  public:
    vector<int> find(vector<int>& arr, int x) {
        // code here
        int l = 0,r=arr.size()-1;
        int ans1 = -1,ans2 = -1;
        while(l<=r){
            int m = (l+r)/2;
            if(arr[m] == x){
                ans1 = m;
                r = m-1;
            }
            else if(arr[m] > x){
                r = m-1;
            }
            else{
                l = m+1;
            }
        }
        if(ans1 == -1){
            return {-1,-1};
        }
        l = 0,r=arr.size()-1;
        while(l<=r){
            int m = (l+r)/2;
```

```
            if(arr[m] == x){
                ans2 = m;
                l=m+1;
            }
            else if(arr[m] < x){
                l=m+1;
            }
            else{
                r = m-1;
            }
        }
        return {ans1,ans2};
    }
};

int main() {

    int t;
    cin >> t;
    cin.ignore();
    while (t--) {
        vector<int> arr;
        string input;
        getline(cin, input);
        stringstream ss(input);
        int number;
        while (ss >> number) {
            arr.push_back(number);
        }
        int x;
        cin >> x;
```

```
        cin.ignore();
        vector<int> ans;
        Solution ob;
        ans = ob.find(arr, x);
        cout << ans[0] << " " << ans[1] << endl;
    }
    return 0;
}
```

OUTPUT:

[0,9]

Complexity;

Time:O(n)

Space:O(n)

4.First transition Point

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

class Solution {
  public:
    int transitionPoint(vector<int>& arr) {
        int l = 0,r = arr.size()-1;
        while(l<=r){
            int m = (l+r)/2;
            if(arr[m] == 0){
                l = m+1;
            }
            else{
                r = m-1;
            }
        }
        return (l>=arr.size()) ? -1 : l;
```

```
    }
};



int main() {


    int t;
    cin >> t;
    cin.ignore();
    while (t--) {
        vector<int> arr;
        string input;
        getline(cin, input);
        stringstream ss(input);
        int number;
        while (ss >> number) {
            arr.push_back(number);
        }
        Solution ob;
        cout << ob.transitionPoint(arr) << endl;
    }
    return 0;
}
```

OUTPUT

First transition point: 3

Complexity:

Time: O(nlogn)

Space: O(nlogn)

5.Wave Array

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

```
class Solution {
  public:
    void convertToWave(vector<int>& arr) {
        // code here
        for(int i = 1;i<arr.size();i+=2){
            swap(arr[i],arr[i-1]);
        }
    }
};



int main() {

    int t;
    cin >> t;
    cin.ignore();
    while (t--)
    {
        vector<int> a;
        string input;
        getline(cin, input);
        stringstream ss(input);
        int number;
        while (ss >> number) {
            a.push_back(number);
        }

        sort(a.begin(), a.end());
        Solution ob;
```

```
        ob.convertToWave(a);

        for (int i = 0; i < a.size(); i++)
            cout << a[i] << " ";

        cout << endl;
    }
}
```
OUTPUT:

[6, 3, 10, 5, 20, 7]

Complexity:

Time: O(nlogn)

Space: O(1)

6.Remove duplicate from sorted Array

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

class Solution {
  public:
    int remove_duplicate(vector<int> &arr) {
        int i = 0,j = 1;
        while(j<arr.size()){
            if(arr[i] != arr[j]){
                arr[i+1] = arr[j];
                i++;
                j++;
            }
            else{
                j++;
            }
        }
        return i+1;
```

```
    }
};

int main() {
    int t;
    cin >> t;
    cin.ignore();
    while (t--) {
        vector<int> arr;
        string input;
        getline(cin, input);
        stringstream ss(input);
        int number;
        while (ss >> number) {
            arr.push_back(number);
        }
        Solution ob;
        int ans = ob.remove_duplicate(arr);
        for (int i = 0; i < ans; i++) {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

OUTPUT:

1 2 3 4 5

Complexity:

Time: O(n)

Space: O(1)