**Link=**[**https://www.youtube.com/watch?v=1qrlDngStmU&list=PLoa\_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=2**](https://www.youtube.com/watch?v=1qrlDngStmU&list=PLoa_roVVsxA1CJu4DsOljb9d7FJkMCynM&index=2)

**[L1]Class - 2 ( Discussion on Practice problems of STL Vector )[Bangla]**

**Class Video link :**

<https://youtu.be/1qrlDngStmU>

**Discussed topics :**

1) Some built in functions

2) A. Sereja and Dima ( Codeforces )

Link : <https://codeforces.com/problemset/problem/381/A>

My Code : <https://codeforces.com/contest/381/submission/86489536>

3) 852. Peak Index in a Mountain Array ( Leet Code )

Link : <https://leetcode.com/problems/peak-index-in-a-mountain-array/>

My Code : <https://paste.ubuntu.com/p/9QRYtkx43z/>

4) 1502. Can Make Arithmetic Progression From Sequence ( Leet Code )

Link : <https://leetcode.com/problems/can-make-arithmetic-progression-from-sequence/>

My Code : <https://paste.ubuntu.com/p/ZxD9CqRxvc/>

5) Distinct Count ( Hacker erath )

LInk : <https://www.hackerearth.com/practice/data-structures/trees/binary-search-tree/practice-problems/algorithm/distinct-count/submissions/>

My Code : <https://paste.ubuntu.com/p/qGyDR48ScK/>

**Class tasks :**

1. A. Sereja and Dima ( Codeforces )

Link: <https://classroom.google.com/u/0/c/MTE2NTUzNTIxNDIx/a/MTE2ODYzMDM1MTEy/details>

1. 852. Peak Index in a Mountain Array ( Leet code )

Link= <https://classroom.google.com/u/0/c/MTE2NTUzNTIxNDIx/a/MTE2ODYzMDM1MTI2/details>

1. 1502. Can Make Arithmetic Progression From Sequence ( Leet code )

Link= <https://classroom.google.com/u/0/c/MTE2NTUzNTIxNDIx/a/MTE2OTE3NDgxNTE4/details>

1. Distinct Count ( Hackerearth )

Link= <https://classroom.google.com/u/0/c/MTE2NTUzNTIxNDIx/a/MTE2OTE3NDgxNTM1/details>

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<int>v={2,3,5,5,7,7,1};**

**unique(v.begin(),v.end());**

**for(auto u:v)cout<<u<<" ";**

**cout<<endl;**

**return 0;**

**}**

**Output:**

**2 3 5 7 1 7 1**

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<int>v={2,3,5,5,7,7,1};**

**sort(v.begin(),v.end());**

**unique(v.begin(),v.end());**

**for(auto u:v)cout<<u<<" ";**

**cout<<endl;**

**return 0;**

**}**

**Output:**

**1 2 3 5 7 7 7**

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<int>v={2,3,5,5,7,7,1};**

**sort(v.begin(),v.end());**

**unique(v.begin(),v.end());**

**for(auto u:v)cout<<u<<" ";**

**cout<<endl;**

**return 0;**

**}**

**Output:**

**1 2 3 5 7 7 7**

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<int>v={2,3,5,5,7,7,1};**

**sort(v.begin(),v.end());//O(nlog2(n))**

**int size=unique(v.begin(),v.end())-v.begin();///O(n)**

**cout<<"Unique Size:"<<size<<endl;**

**for(int i=0;i<size;i++){**

**cout<<v[i]<<' ';**

**}**

**cout<<endl;**

**for(auto u:v)cout<<u<<" ";**

**cout<<endl;**

**return 0;**

**}**

**Output:**

**Unique Size:5**

**1 2 3 5 7**

**1 2 3 5 7 7 7**

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<int>v={2,3,5,5,7,7,1};**

**cout<<"Maximum value print::"<<endl;**

**cout<<\*max\_element(v.begin(),v.end())<<endl;**

**//or**

**vector<int>::iterator it=max\_element(v.begin(),v.end());**

**cout<<\*it<<endl;**

**vector<int>::iterator it1=max\_element(v.begin()+1,v.begin()+4);**

**cout<<\*it1<<endl;**

**cout<<"Index=";**

**int n=max\_element(v.begin(),v.end())-v.begin();**

**cout<<n<<endl;**

**return 0;**

**}**

**Output:**

**Maximum value print::**

**7**

**7**

**5**

**Index=4**

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<int>v={2,3,5,5,7,7,1};**

**cout<<"Minimum value print::"<<endl;**

**cout<<\*min\_element(v.begin(),v.end())<<endl;**

**//or**

**vector<int>::iterator it=min\_element(v.begin(),v.end());**

**cout<<\*it<<endl;**

**vector<int>::iterator it1=min\_element(v.begin()+1,v.begin()+4);**

**cout<<\*it1<<endl;**

**cout<<"Index=";**

**int n=min\_element(v.begin(),v.end())-v.begin();**

**cout<<n<<endl;**

**return 0;**

**}**

**Output:**

**Minimum value print::**

**1**

**1**

**3**

**Index=6**

***#include <bits/stdc++.h>***

**using namespace std;**

**int main()**

**{**

**vector<vector<int>>v;**

**vector<int> a1={1,2,3,4};**

**vector<int> a2={5,6,7,8,9,10};**

**v.push\_back(a1);**

**v.push\_back(a2);**

**for(auto u:v){**

**for(auto g:u){**

**cout<<g<<" ";**

**}**

**cout<<endl;**

**}**

**return 0;**

**}**

**Output:**

**1 2 3 4**

**5 6 7 8 9 10**

**Problem link=**[**https://codeforces.com/problemset/problem/381/A**](https://codeforces.com/problemset/problem/381/A)

**Code Link=**[**https://codeforces.com/contest/381/submission/86489536**](https://codeforces.com/contest/381/submission/86489536)

*#include <bits/stdc++.h>*

using namespace std;

int main()

{

int n;

cin>>n;

vector<int>v(n);

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

cin>>v[i];

}

int s=0,d=0,f=1;

**while** (!v.empty()){

**if**(f==1){

**if**(\*v.begin()>v.back()){

s+=\*v.begin();

v.erase(v.begin());

}**else**{

s+=v.back();

v.pop\_back();

}

f=2;

}**else**{

**if**(\*v.begin()>v.back()){

d+=\*v.begin();

v.erase(v.begin());

}**else**{

d+=v.back();

v.pop\_back();

}

f=1;

}

}

cout<<s<<" "<<d<<endl;

**return** 0;

}

**Output:**

42

15 29 37 22 16 5 26 31 6 32 19 3 45 36 33 14 25 20 48 7 42 11 24 28 9 18 8 21 47 17 38 40 44 4 35 1 43 39 41 27 12 13

613 418

**problem link=**[**https://leetcode.com/problems/peak-index-in-a-mountain-array/**](https://leetcode.com/problems/peak-index-in-a-mountain-array/)

**code =**[**https://paste.ubuntu.com/p/9QRYtkx43z/**](https://paste.ubuntu.com/p/9QRYtkx43z/)

**class** **Solution** {

public:

int peakIndexInMountainArray(vector<int>& arr)

{

**return** max\_element(arr.begin(),arr.end())-arr.begin();

}

};

In [ ]:

**class** **Solution** {

public:

int peakIndexInMountainArray(vector<int>& arr)

{

**return** max\_element(arr.begin(),arr.end())-arr.begin();

}

};

**Output:**

Input: arr = [24,69,100,99,79,78,67,36,26,19]

Output: 2

**problem link=**[**https://leetcode.com/problems/can-make-arithmetic-progression-from-sequence/**](https://leetcode.com/problems/can-make-arithmetic-progression-from-sequence/)

**code =**[**https://paste.ubuntu.com/p/ZxD9CqRxvc/**](https://paste.ubuntu.com/p/ZxD9CqRxvc/)

**class** **Solution** {

public:

bool canMakeArithmeticProgression(vector<int>& arr) {

sort(arr.begin(),arr.end());

**if**(arr.size()<=2) **return** true;

int d=arr[1]-arr[0];

**for**(int i=2;i<arr.size();i++){

**if**(arr[i]-arr[i-1]!=d){

**return** false;

}

}

**return** true;

}

};

**Output:**

Input: arr = [3,5,1]

Output: true

Explanation: We can reorder the elements **as** [1,3,5] **or** [5,3,1] **with** differences 2 **and** -2 respectively, between each consecutive elements.

**Output:**

Input: arr = [1,2,4]

Output: false

Explanation: There **is** no way to reorder the elements to obtain an arithmetic progression.

**Problem=**[**https://www.hackerearth.com/practice/data-structures/trees/binary-search-tree/practice-problems/algorithm/distinct-count/description/**](https://www.hackerearth.com/practice/data-structures/trees/binary-search-tree/practice-problems/algorithm/distinct-count/description/)

**code=**[**https://paste.ubuntu.com/p/qGyDR48ScK/**](https://paste.ubuntu.com/p/qGyDR48ScK/)

*#include <bits/stdc++.h>*

using namespace std;

int main()

{

int t;

cin>>t;

**while** (t--){

int n,x;

cin>>n>>x;

vector<int>v(n);

**for**(int i=0;i<v.size();i++){

cin>>v[i];

}

sort(v.begin(),v.end());

int z=unique(v.begin(),v.end())-v.begin();

**if**(z==x) cout<<"Good"<<endl;

**else** **if**(z>x) cout<<"Average"<<endl;

**else** cout<<"Bad"<<endl;

}

}

**Output:**

4

4 1

1 4 2 5

4 2

4 2 1 5

4 3

5 2 4 1

4 4

1 2 4 5

Average

Average

Average

Good