Cho mảng A[] gồm n số nguyên không âm và số k. Hãy tìm số lớn nhất của mỗi dãy con liên tục gồm k phần tử của mảng. Ví dụ với mảng A[] = {1, 2, 3, 1, 4, 5, 2, 3, 6}, K = 3, ta có kết quả *3 3 4 5 5 5 6*.

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[] và số k; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ k < n ≤107; 0≤ A[i] ≤107;

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

#include<iostream>

#include<string>

#include<vector>

#include<map>

#include<queue>

#include<cmath>

#include<set>

#include<algorithm>

#define ll long long

using namespace std;

const int oo = 1e7+5;

int A[oo];

int main(){

cin.tie(0); ios\_base::sync\_with\_stdio(false);

int test; cin >> test;

while(test--){

int n, k; cin >> n >> k;

for(int i=0; i<n; i++) cin >> A[i];

int res = 0;

for(int i=0; i<k; i++) res = max(res, A[i]);

cout << res << " ";

for(int i=1; i<=n-k; i++){

if(A[i-1] == res){

res = 0;

for(int j=i; j<=i+k; j++) res = max(res, A[j]);

}

else if(A[k+i-1] > res) res = A[k+i-1];

cout << res << " ";

}

cout << endl;

}

}

Cho mảng X[] gồm n phần tử và mảng Y[] gồm m phần tử. Hãy đếm số các cặp xy>yx, trong đó x€X[] và y€Y[]. Ví dụ X[] = {2, 1, 6 }, Y[] = {1, 5} ta có kết quả là 3 cặp (2, 1), (2, 5), (6, 1).

Input:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng đầu tiên đưa vào n, m tương ứng với số phần tử của mảng X[] và Y[]; dòng tiếp theo là n số X[i] của mảng X[]; dòng cuối cùng là m số của mảng Y[]; các số được viết cách nhau một vài khoảng trống.
* T, n, m, X[i], Y[j] thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ n, m ≤105; 1≤ X[i], Y[j] ≤103.

Output:

* Đưa ra kết quả mỗi test theo từng dòng.

#include<bits/stdc++.h>

using namespace std;

int tam[5];

int n,m,a[100000],b[100000];

int dem(int x){

if(x==0){

return 0;

}

if(x==1){

return tam[0];

}

int \*pos=upper\_bound(b,b+m,x);

int ans=(b+m)-pos;

ans+=tam[0]+tam[1];

if(x==2){

ans-=tam[3]+tam[4];

}

if(x==3){

ans+=tam[2];

}

return ans;

}

int main(){

int t;

cin>>t;

while(t--){

memset(tam,0,sizeof(tam));

cin>>n>>m;

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

cin>>a[i];

}

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

cin>>b[i];

if(b[i]<=4){

tam[b[i]]++;

}

}

sort(b,b+m);

int ans=0;

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

ans+=dem(a[i]);

}

cout<<ans<<endl;

}

}