**First Unique character in string**

#include<bits/stdc++.h>

using namespace std;

int main(){

string s;

map<char,int> m;

cin>>s;

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

m[s[i]]++;

}

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

if(m[s[i]]==1)

{

cout<<s[i];

break;

}

}

return 0;

}

**First Letter to Appear Twice**

#include<bits/stdc++.h>

using namespace std;

int main(){

string s;

map<char,int> m;

cin>>s;

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

if(m[s[i]]==1)

{

cout<<s[i];

break;

}

m[s[i]]++;

}

return 0;

}

**"Given a string s, sort it in decreasing order based on the frequency of the characters**

**If there are multiple answers, return any of them.**

**Input: s = ""tree""**

**Output: ""eert""**

**Explanation: 'e' appears twice while 'r' and 't' both appear once.**

**So 'e' must appear before both 'r' and 't'. Therefore ""eetr"" is also a valid answer."**

#include<bits/stdc++.h>

using namespace std;

bool comp(pair<char,int> a,pair<char,int> b){

return a.second>b.second;

}

int main(){

string s;

map<char,int> m;

cin>>s;

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

m[s[i]]++;

}

vector <pair<char,int>> arr;

for(char i='a';i<'a'+26;i++){

if(m[i]!=0)

arr.push\_back({i,m[i]});

}

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

for(auto it:arr){

for(int i=0;i<it.second;i++)

cout<<it.first;

}

return 0;

}

**Given an array of integers nums, sort the array in increasing order based on the frequency of the values. If multiple values have the same frequency, sort them in decreasing order.**

#include<bits/stdc++.h>

using namespace std;

bool comp(pair<int,int> a,pair<int,int> b){

if(a.second==b.second)

return a.first>b.first;

return a.second>b.second;

}

int main(){

int n;

map<int,int> m;

cin>>n;

vector<int> v(n);

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

cin>>v[i];

m[v[i]]++;

}

vector <pair<int,int>> arr(m.begin(),m.end());

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

for(auto it:arr){

for(int i=0;i<it.second;i++)

cout<<it.first<<" ";

}

return 0;

}

**"You are given an integer array nums consisting of 2 \* n integers.**

**You need to divide nums into n pairs such that:**

**Each element belongs to exactly one pair.**

**The elements present in a pair are equal.**

**Return true if nums can be divided into n pairs, otherwise return false."**

#include<bits/stdc++.h>

using namespace std;

bool check(map<int,int> m){

for(auto it:m){

if(it.second%2!=0)

return false;

}

return true;

}

int main(){

int n;

map<int,int> m;

cin>>n;

vector<int> v(n);

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

cin>>v[i];

m[v[i]]++;

}

if(check(m))

cout<<"true";

else

cout<<"false";

return 0;

}

**"You are given a 0-indexed integer array nums. In one operation, you may do the following:**

**Choose two integers in nums that are equal.**

**Remove both integers from nums, forming a pair.**

**The operation is done on nums as many times as possible.**

**Return a 0-indexed integer array answer of size 2 where answer[0] is the number of pairs that are formed and answer[1] is the number of leftover integers in nums after doing the operation as many times as possible."**

#include<bits/stdc++.h>

using namespace std;

void operation(map<int,int> m){

int prs=0,single=0;

for(auto it:m){

if(it.second%2==1)

single+=1;

prs+=(it.second/2);

}

cout<<prs<<" "<<single;

}

int main(){

int n;

map<int,int> m;

cin>>n;

vector<int> v(n);

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

cin>>v[i];

m[v[i]]++;

}

operation(m);

return 0;

}

**Given an integer array nums and an integer k, return the k most frequent elements. You may return the answer in any order.**

#include<bits/stdc++.h>

using namespace std;

bool comp(pair<int,int> a,pair<int,int> b){

return a.second>b.second;

}

int main(){

int n,k;

map<int,int> m;

cin>>n;

vector<int> v(n);

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

cin>>v[i];

m[v[i]]++;

}

cin>>k;

vector <pair<int,int>> arr(m.begin(),m.end());

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

for(auto it:arr){

cout<<it.first<<" ";

if(--k==0)

break;

}

return 0;

}