**SORTING:**

[A - Gravity Flip](https://vjudge.net/problem/CodeForces-405A)

#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];

}

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

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

cout<<v[i]<<" ";

}

return 0;

}

[B - Bubble Sort Adhoc](https://vjudge.net/problem/HackerRank-si-bubble-sort)

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n,t,count;

cin>>t;

while(t--){

cin>>n;

vector<int> v(n);

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

cin>>v[i];

}

count=0;

for(int i=0;i<n-1;i++){

for(int j=n-1;j>i;j--){

if(v[j] < v[j-1]){

swap(v[j],v[j-1]);

count++;

}

}

}

cout<<count<<endl;

}

return 0;

}

[C - Insertion Sort Adhoc](https://vjudge.net/problem/HackerRank-si-insertion-sort)

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n,t,temp,j;

cin>>t;

while(t--){

cin>>n;

vector<int> v(n);

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

cin>>v[i];

}

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

temp=v[i];

j=i-1;

while(j>=0 && temp<v[j]){

v[j+1]=v[j];

j--;

}

cout<<j+1<<" ";

v[j+1]=temp;

}

cout<<endl;

}

return 0;

}

//

[D - Selection Sort Adhoc](https://vjudge.net/problem/HackerRank-si-selection-sort)

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n,t,temp;

cin>>t;

while(t--){

cin>>n;

vector<int> v(n);

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

cin>>v[i];

}

for(int i=n-1;i>0;i--){

temp=i;

for(int j=i-1;j>=0;j--){

if(v[j] >= v[temp])

temp=j;

}

cout<<temp<<" ";

swap(v[i],v[temp]);

}

cout<<endl;

}

return 0;

}

//

[E - Frequency Sort](https://vjudge.net/problem/HackerRank-si-frequency-sort)

 #include <bits/stdc++.h>

using namespace std;

vector<int> c(2001,0);

bool comp(int a,int b){

if(c[a+1000] != c[b+1000]){

return c[a+1000]<c[b+1000];

}

else

return a<b;

}

int main()

{

int t,n;

cin>>t;

while(t--){

cin>>n;

vector<int> v(n);

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

cin>>v[i];

c[v[i]+1000]++;

}

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

for(int a:v)

cout<<a<<" ";

cout<<endl;

vector<int> temp(2001,0);

c=temp;

}

return 0;

## }[F - Missing Numbers](https://vjudge.net/problem/HackerRank-missing-numbers)

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n1,n2,mx,mn;

cin>>n1;

vector<int> v(n1);

mx=0; mn=INT\_MAX;

for(int i=0;i<n1;i++){

cin>>v[i];

mx=max(v[i],mx);

mn=min(v[i],mn);

}

cin>>n2;

vector<int> v2(n2);

for(int i=0;i<n2;i++){

cin>>v2[i];

mx=max(v2[i],mx);

mn=min(v2[i],mn);

}

vector<int> c(mx-mn+1);

for(int a: v){

c[a-mn]++;

}

for(int a:v2){

c[a-mn]--;

}

for(int i=0;i<mx-mn+1;i++){

if(c[i]<0)

cout<<(i+mn)<<" ";

}

cout<<endl;

return 0;

}

[G - Sort 0s and 1s](https://vjudge.net/problem/HackerRank-si-sort-0s-and-1s)

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t,n,start,end;

cin>>t;

while(t--){

cin>>n;

vector<int> v(n);

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

cin>>v[i];

}

start=0; end=n-1;

while(start<end){

if(!v[start]){

start++;

}

else if(v[start] && !v[end]){

swap(v[start],v[end]);

start++;

end--;

}

else

end--;

}

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

cout<<v[i]<<" ";

}

cout<<endl;

}

return 0;

}

[H - Inversion Count](https://vjudge.net/problem/SPOJ-INVCNT)

Time limit exceeded

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t,n,count;

cin>>t;

cin.ignore();

while(t--){

cin>>n;

vector<int> v(n);

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

cin>>v[i];

}

count=0;

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

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

if(i<j && v[i]>v[j])

count++;

}

}

cout<<count<<endl;

cin.ignore();

}

return 0;

}

[I - Mergesort](https://vjudge.net/problem/SPOJ-MERGSORT)

#include <bits/stdc++.h>

using namespace std;

void merge(vector<int>& v,int l,int mid,int r){

int i=l,j=mid+1;

vector<int> temp;

while(i<=mid && j<=r){

if(v[i] < v[j]){

temp.push\_back(v[i]);

i++;

}

else{

temp.push\_back(v[j]);

j++;

}

}

while(i<=mid){

temp.push\_back(v[i]);

i++;

}

while(j<=r){

temp.push\_back(v[j]);

j++;

}

for(int i=l;i<=r;i++){

v[i]=temp[i-l];

}

}

void merge\_sort(vector<int>& v,int l,int r){

if(l<r){

int mid=l+(r-l)/2;

merge\_sort(v,l,mid);

merge\_sort(v,mid+1,r);

merge(v,l,mid,r);

}

}

int main()

{

int i,j,a;

vector<int> v;

while(cin>>a){

v.push\_back(a);

}

merge\_sort(v,0,v.size()-1);

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

cout<<v[i]<<" ";

}

return 0;

}