**Program 7:**

#include <bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while (t--) {

int n;

cin>>n;

int arr[n];

int comp=0, shift=0;

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

cin>>arr[i];

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

int j= i-1;

int k= arr[i];

while(j>=0 and ar[i]>key) {

comp++;

arr[j + 1] = arr[j];

shifts++;

j--;

}

arr[j+1]=key;

shifts++;

}

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

cout <<arr[i]<<" "<<;

cout << endl<<"comparisons = " << comp;

cout<<endl<<" shifts = "<< shifts;

}

return 0;

}

**//**

**Input:**

3

8

-23 65 -31 76 46 89 45 32

10

54 65 34 76 78 97 46 32 51 21

15

63 42 223 645 652 31 324 22 553 -12 54 65 86 46 325

**//**

**Output:**

-31 -23 32 45 46 65 76 89

Comparisons = 13

Shifts = 20

21 32 34 46 51 54 65 76 78 97

Comparisons = 28

Shifts = 37

-12 22 31 42 46 54 63 65 86 223 324 325 553 645 652

Comparisons = 54

Shifts = 68

**Program 8:**

#include <bits/stdc++.h>

using namespace std;

void selectionsort (int arr[], int n, int&c1, int&c2)

{ int i, j, min\_idx;

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

min\_idx= i;

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

if (arr[j] <<arr[min\_idx])

min\_idx =j;

c1++;

}

swap(arr[min\_idx], arr[i]);

c2++;

}

}

int main() {

int t;

cin>>t;

while (t--) {

int n, c1=0, c2=0;

int arr[n];

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

cin >> arr[i];

selectionsort (arr, n, c1,c2);

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

cout <<arr[i]<<" ";

cout << endl<<" Comparisons = " << c1;

cout << endl<<"Swaps = "<<c2 <<endl;

}

return 0;

}

**//**

**Input:**

3

8

-23 65 -31 76 46 89 45 32

10

54 65 34 76 78 97 46 32 51 21

15

63 42 223 645 652 31 324 22 553 -12 54 65 86 46 325

**//**

**Output:**

-21 -13 12 45 46 65 76 89

Comparisons = 28

Swaps = 7

21 32 34 46 51 54 65 76 78 97

Comparisons = 45

Swaps = 9

12 22 31 42 46 54 63 65 86 223 324 325 553 645

65

Comparisons = 105

swaps = 14

**Program 9:**

#include < bits/stdc++.h>

using namespace std;

void merge (int arr[], int l, int mid, int l) {

int i = l, j = mid+1, k=l;

int b[100];

while (i<=mid and j<=h) {

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

b[k++] = arr[j++];

}

else

b[k++] = arr[j++];

}

for(; i<=mid; i++)

b[k++]=arr[j];

for (; j<=h; j++)

b[k++] = arr[j];

for (int i=l; i<h+1; i++) {

arr[i]=b[j];

}

void Mergesort(int arr[], int l, int h) {

int mid;

if (l>=h) {

return;

}

mid= (l+h) /2;

Mergesort(arr, l, mid);

Mergesort(arr, mid+1, h);

Merge(arr, I, h, mid); }

int main() {

int t;

cin>>t;

while (t--) {

int n;

cin>>n;

int arr[n];

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

cin>>arr[i];

mergesort(arr, 0, n-1);

bool ans = false;

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

if (arr[i] == arr[i+1]) {

ans = true;

break;

}

}

if (ans)

cout <<" Yes" << endl;

else

cout << "NO" << endl;

}

return 0;

}

**//**

**Input:**

3

5

28 52 83 14 73

10

75 65 165 2 6 86 2 75 8

15

73 35 86 57 98 23 73 1 64 8 11 90 61 19 20

**//**

**Output:**

NO

Yes

NO

**Program 10:**

**Given an unsorted array of integers, design an algorithm and implement it using a program to sort the array by dividing the array into two sub arrays and combining these after sorting each of them. Also print the no. of comparisons and inversions during sorting the array.**

**Code:**

#include<bits/stdc++.h>

using namespace std;

int comp=0;

int get Inv count (int arr[ ], int n)

{

int inv\_count = 0;

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

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

if (arr[i]> arr[j])

inv\_count++;

return inv\_ count;

}

void merge (int arr[ ], int l, int h, int mid) {

int i= l, k= l, j = mid + 1;

int b= [100];

while (i<=mid and j<=h) {

if (arr[i] < arr[j])

b[k++] = arr[i++];

else

b[k++] = arr[j+1];

}

for (; i<= mid; i++)

b[k++]= arr[i];

for(; j<= n; j++)

b[k++] = arr[j++];

for (i=l; i<h+1; i++)

arr[i] = b[i];

}

void mergeSort(int arr[ ], int l, int h) {

int mid;

if (l>=h)

return;

mid= (l+h) /2;

mergeSort(arr, l, mid);

mergeSort(arr, mid+l, h);

merge(arr, l, h, mid);

}

int main () {

int t;

cin>> t;

while (t--) {

comp=0;

int n, inv;

cin>>n;

int arr[n];

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

cin>>arr[i];

inv= getInvCount(arr,n);

mergesort(arr, 0, n-1);

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

cout<<arr[i]<<" ";

cout << endl<< " Comparisons = " << comp;

cout << end < " Inversions = "<< inv;

inv=0;

}

return 0;

}

**//**

**Input:**

3

8

23 65 21 76 46 89 45 32

21 23 32 45 46 65 76 89

Comparisons = 16

Inversions = 13

10

54 65 34 76 78 97 46 32 51 21

21 32 34 46 51 54 65 76 78 97

Comparisons = 22

Inversions = 28

15

63 42 223 645 652 31 324 22 553 12 54 65 86 46 3 12 22 31 42 46 65 86 223 324 325 553 645 652

Comparisons = 43

Inversions = 54

**Program 11:**

#include<iostream>

using namespace std;

int ccount, scount;

int partition (int arr[ ], int l, int r)

{

int i= l-1;

int pivat = arr [r];

for (int j = l; j<r; j++)

{

ccount++;

if (arr[j]< pivot)

{

scount++;

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i+1], arr[r]);

scount++;

return i+1;

}

void quicksort(int arr[], int I, int r)

{

if ( l<r)

{ int par= partition (arr I, r);

quicksort (arr, l, par-1);

quicksort (arr, par+1,r);

}

}

void solve() {

ccount=0,scount=0;

int n;

cin>>n;

int arr[n];

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

cin>>arr[i];

quicksort(arr, 0, n-1);

for(int i=0, i<n, i++)

cout<<arr[i]<<" ";

cout << endl;

cout<<" comparisons =" <<ccount << endl;

cout<<" swaps = " << scount<<endl;

}

int main()

{ int t;

cin>>t;

while (t--) {

solve(); }

}

**//**

**Input:**

3

8

23 65 21 76 46 89 45 32

21 23 32 45 46 65 76 89

Comparison = 14

Swaps = 10

10

54 65 34 76 78 97 46 32 51 21

21 32 34 46 51 54 65 76 78 97

Comparisons= 29

Swaps = 21

15

63 42 223 645 652 31 324 12 54 65 86 46 325

12 22 31 42 46 54 63 65 86 223 324 325 553 645 652

Comparisons = 45

Swaps = 39

**Program 12:**

**Given an unsorted array of integers, design an algorithm and implement it using a program to find kth smallest or largest element in the array.**

**Code:**

#include<bits/stdc++.h>

using namespace std;

int main()

{

int t;

cin>>t;

while (t--) {

int n;

cin>>n;

set<int>s;

int a;

for (int i=0, i<n, i++)

{

cin>> a;

s.insert(a);

}

priority\_queue <int, vector <int>, greater < int>> pq;

for (auto i:s) {

pq.push(i);

}

int k;

cin>>k;

int res";

while (k--) {

res= pq.top ();

pq.top(); }

cout <<ress <<endl;

}

return 0;

}

**//**

**Input:**

3

10

123 656 54 765 344 514 765 34 765 234

3

--> 123

15

43 64 13 78 864 346 786 456 21 19 8 434 76 270 601

8

78

**Program 13:**

**Given an unsorted array of alphabets containing duplicate elements. Design an algorithm and implement it using a program to find which alphabet has maximum occurences and print it.**

**Code:**

#include<bits/stdc++.h>

using namespace std;

void solve() {

int n;

cin>>n;

char str[n];

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

cin>>str[i];

sort (str, str+n);

int maxx = 0, c=1;

char ans =0;

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

if(str[i]==str[i-1]) {

c++;

}

else {

if(c>maxx) {

maxx=c;

ans = str[i-1];

}

c=1;

}

}

if (c>maxx) {

maxx=c;

ans = str[n-1];

}

else if(maxx==1)

cout <<" No Duplicates present "<< endl;

else

cout<<ans<< "\_" <<maxx <<endl;

}

int main() {

int t;

cin>> t;

while(t--) {

solve();

}

return 0;

}

**//**

**Input:**

3

10

a e d w a d q a f p

a-3

15

r k p g v y u m q a d j c z e

No duplicate found

20

8 t l l t c w a w g l c w d s a a v c l

l-4

**Program 14:**

**Given an unsorted array of integers, design and implement it using a program to find wether two elements exist such that their sum is equal to given key element.**

**Code:**

#include< bits/stdc++.h>

using namespace std;

void Solve() {

int n;

cin>>n;

vector<int>ans;

int a;

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

cin>> a;

ans.push\_back (a);

}

int sum; cin>> sum;

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

int i= 0, j= ans.size()-1;

while (i<j) {

if(ans[i]+ans[j] = = sum) {

cout << ans[i]<<" "<<ans[j] << endl;

return;

}

else if(ans[i]+ans[j]< sum)

i++;

else j--;

}

cout<<" No such element exist" << endl;

}

int main () {

int t;

cin>>t;

while (t--) {

solve ();

}

return 0;

}

**//**

**Input:**

2

10

64 28 97 40 12 72 84 24 38 10

50

10 40

15

56 10 72 91 29 341 45 61 20 11 39 9 12 94

302

No such element exist

**Program 15:**

#include<bits/stdc++.h>

using namespace std;

int main() {

int n, m, i, j;

cin>>n;

arr1[n];

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

cin>>arr1[i];

cin>>m;

int arr2[m];

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

cin>>arr2[i];

sort(arr1, arr1+h);

sort(arr2, arr2+h);

vector<int>ans;

i=0, j=0;

while(i<n and j<m) {

if(arr1[i]== arr2[j]) {

ans.push\_back(arr1[i]);

i++;

j++;

}

else if(arr1[i] < arr2[j])

i++;

else

j++;

}

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

cout<<ans[i]<<" ";

}

cout<<endl;

return 0;

}

**//**

**Input:**

7

34 76 10 39 85 10 55

12

30 55 34 72 10 34 10 89 11 30 69 51

--> 10 10 34 55