CSE246

Section 4

Fall 2023

**Lab Task - 01**

**Topic: Sorting and Binary Search**

Submitted By

|  |  |
| --- | --- |
| **Name** | **ID** |
| B M Shahria Alam | 2021-3-60-016 |
| Golam Kibria | 2021-3-60-215 |
| Sidratul Moontaha | 2021-3-60-048 |
| MD Imran Khan | 2021-3-60-206 |

**Problem 1**

**Solution:**

#include<bits/stdc++.h>

using namespace std;

int part(double a[], int l, int r)

{

int pivot=a[r];

int i=(l-1);

for (int j=l; j<=r-1; j++)

{

if (a[j]<pivot)

{

i++;

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

}

}

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

return (i+1);

}

void quickSort(double a[], int l,int r)

{

if(l<r)

{

int pivot = part(a, l, r);

quickSort(a, l, pivot-1);

quickSort(a, pivot+1, r);

}

}

int main()

{

int n, k;

cin>>n;

double a[n], B[n];

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

{

cin>>a[i];

B[i] = a[i];

}

cin>>k;

quickSort(a, 0, n-1);

int val = a[k-1];

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

{

if(val==B[i])

{

cout<<i+1<<"\n";

break;

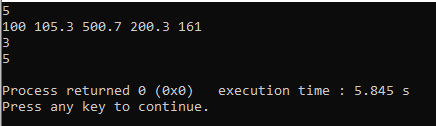
}

}

return 0;

}

**OUTPUT:**



**Problem 2**

**Solution:**

#include<bits/stdc++.h>

using namespace std;

int part(int a[], int l, int r)

{

int pivot = a[r];

int i = (l - 1);

for (int j = l; j <= r - 1; j++)

{

if (a[j] < pivot)

{

i++;

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

}

}

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

return (i + 1);

}

void quickSort(int a[], int l, int r)

{

if(l<r)

{

int pivot = part(a, l, r);

quickSort(a, l, pivot-1);

quickSort(a, pivot+1, r);

}

}

int main()

{

int n;

cin>>n;

int a[n];

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

{

cin>>a[i];

}

quickSort(a, 0, n-1);

int ans = abs(a[0]);

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

{

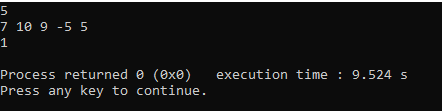
ans = min(ans, abs(a[i]-a[i-1]));

}

cout<<ans<<endl;

}

**OUTPUT**

****

**Problem 3**

**Solution:**

#include<bits/stdc++.h>

using namespace std;

int part(int a[], int l, int r)

{

int pivot=a[r];

int i=(l-1);

for (int j=l; j<=r-1; j++)

{

if (a[j] < pivot)

{

i++;

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

}

}

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

return (i+1);

}

void quickSort(int a[],int l,int r)

{

if(l<r)

{

int pivot = part(a,l,r);

quickSort(a,l,pivot-1);

quickSort(a,pivot+1,r);

}

}

int main()

{

int n;

cin>>n;

int a[n];

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

{

cin>>a[i];

}

quickSort(a, 0, n-1);

int ans=-1, cont=0;

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

{

int j=i+1, cnt=1;

while(j<n && a[j]==a[i])

{

j++;

cnt++;

}

if(cnt>=cont)

{

cont=cnt;

ans=a[i];

}

i=j-1;

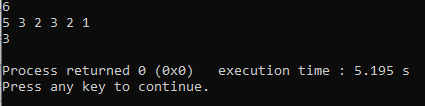
}

cout<<ans<<endl;

return 0;

}

**OUTPUT:**



**Problem 4**

**Solution:**

#include <stdio.h>

#include<bits/stdc++.h>

using namespace std;

int lowerBound (vector <int>vec, int n, int t)

{

int low = 0;

int high = n -1;

int ans = 0;

while (low<=high)

{

int mid =low+(high-low)/2;

if (vec[mid]>t)

{

high=mid-1;

}

else

{

ans=mid;

low=mid+1;

}

}

int i=0;

while(vec[ans]==vec[ans-i] && ans-i>0)

{

i++;

}

if(i!=0)i--;

return ans-i;

}

int upperBound (vector < int >vec, int n, int t)

{

int low = 0;

int high = n - 1;

int ans = n;

while (low <= high)

{

int mid = low + (high - low) / 2;

if (vec[mid] <= t)

{

low=mid+1;

}

else

{

ans=mid;

high = mid - 1;

}

}

int i=0;

while(vec[ans]==vec[ans+i] && ans+i<n)

{

i++;

}

if(i!=0)i--;

return ans+i;

}

int main ()

{

int n,x;

cin >>n;

cin >>x;

vector <int>v;

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

{

int a;

cin >>a;

v.push\_back(a);

}

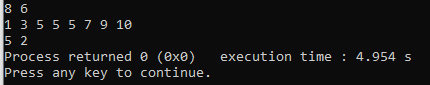
int ans=upperBound (v, n, x);

int ans1=lowerBound (v, n, x);

cout<<ans<<" "<<ans1;

}

**OUTPUT**



**Problem 5**

**Solution:**

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

int main()

{

double n;

double ans;

cin>>n;

double low=0;

double high=n;

while(low<=high)

{

double mid=low+(high-low)/2;

if(mid\*mid == n)

{

cout<<setprecision(3)<<mid;

return 0;

}

else if(mid\*mid>=n)

{

high=mid-1;

}

else

{

low=mid+1;

}

}

double increment =0.1;

for (int i =0; i<3;i++)

{

while (ans \*ans<=n)

{

ans +=increment;

}

ans=ans-increment;

increment =increment/10;

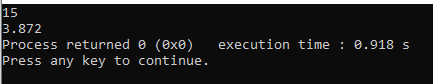
}

cout<<ans;

return 0;

}

**OUTPUT**

****