LAB ASSIGNMENT	
NAME: Aashirwad Kumar	Roll NUMBER: **IMH/10004/18**

Question 1	**Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the lIst to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.**	28/05/2021

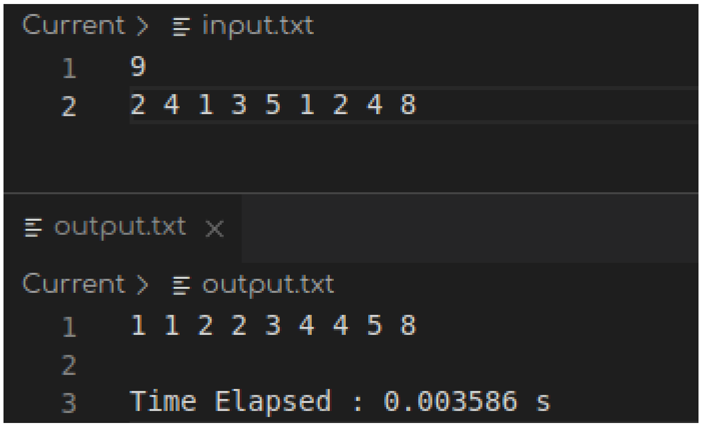
```
#include <bits/stdc++.h>
using namespace std;



#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;





#define F first
#define S second
#define PB push_back
#define MP make_pair





void swap1 (long long int *a , long long int *b)
{
long long int temp=*a;
*a=*b;
*b=temp;
}

long long int partition1(long long int l , long long int h, long long int a[])
{

long long int i=l; long long int j=h;
long long int pivot = a[l];
long long int k;k=l+1;
```

```
for (;i<=h;i++)
{
if(a[i]<pivot)
{
if(i!=k)
{
swap1(&a[k],&a[i]);
}




k++;

}
}
swap1(&a[l],&a[k-1]);
a[k-1]=pivot;

return k-1;

}




void quicksort(long long int l, long long int h , long long int a[])
{
if (l<h){
long long int j= partition1(l,h,a);
quicksort(l,j-1,a);
quicksort(j+1,h,a);

}


}



int main(){


fast;
long long int n;
cin>>n;
long long int a[n];
long long int i;
for ( i=0;i<n;i++)
{
cin>>a[i];
}
```

```
quicksort(0,n-1,a);
for (i = 0; i< n;i++){
cout<<a[i]<<" ";
}
cout<<endl;

#ifndef ONLINE_JUDGE
cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
#endif

return 0;
}
```

Take a screenshot of your output and show here

Quicksort Input and Output



```
9
2 4 1 3 5 1 2 4 8
```

```
1 1 2 2 3 4 4 5 8

Time Elapsed : 0.003586 s
```

Question 2	Write question here **Implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.**	Date : 28/05/2021

Write the code with proper indentation

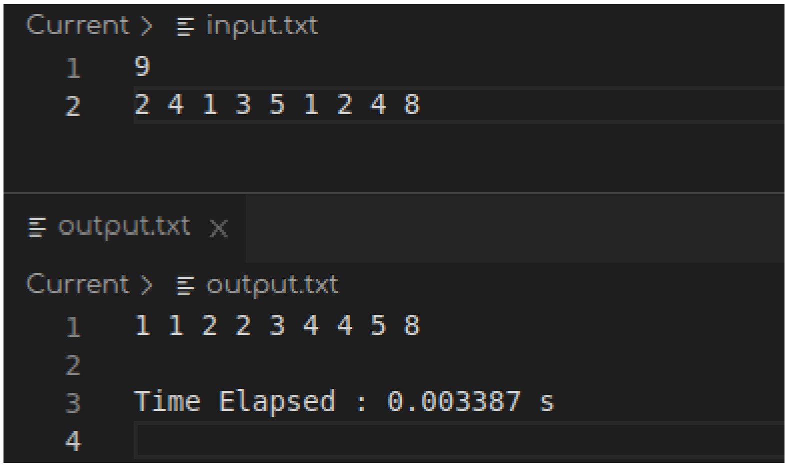
```
#include <bits/stdc++.h>
using namespace std;




#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;
#define F first
#define S second
#define PB push_back
#define MP make_pair




long long int smerge(long long int l,long long int mid,long long int h,long long int a[],long long int n);

void smergesor (long long int l,long long int h,long long int a[],long long int n);
long long int smerge(long long int l,long long int mid,long long int h,long long int a[],long long int n){

long long int inv=0;
long long int i=l;
long long int k=l;

long long int j=mid+1;
long long int b[n];


while(i<=mid && j<=h){
if(a[i]<=a[j]){
```

```
b[k++]=a[i++];
}
else{
b[k++]=a[j++];
}
}


while(j<=h){
b[k++]=a[j++];
}


while(i<=mid){
b[k++]=a[i++];
}

for (k=l;k<=h;k++){
a[k]=b[k];
}
return inv;
}

void smergesor(long long int l,long long int h,long long int a[],long long int
n)

{

long long int mid,inv=0;


if(l<h){
mid=(l+h)/2;
smergesor(l,mid,a,n);
smergesor(mid+1,h,a,n);
smerge(l,mid,h,a,n);


}
}
int main()

{
fast;
long long int n;
long long int t = 1;
cin >> n;
long long int a[n];
for(long long int i=0;i<n;i++){
cin>>a[i];
}
```

```
while(t--){
smergesor(0,n-1,a,n);
}



for(long long int i=0;i<n;i++){
cout<<a[i]<<" ";
}


cout<<endl;


#ifndef ONLINE_JUDGE
cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
#endif
return 0;

}
```

Take a **screenshot** of your output and show here

Mergesort Input and Output



```
9
2 4 1 3 5 1 2 4 8
```

```
1 1 2 2 3 4 4 5 8

Time Elapsed : 0.003387 s
```

Question 3	Write question here **3. Divide and Conquer** **i.** **Given a sorted array of integers, find index of first and last occurrence of a given number. If the** **element is not found in an array, report that as well.** **ii.** **Given a sorted array, find a given element in O (Log n) time.** **iii.** **Given an Array, find peack element in it. A Peack elements is the element that is greater than its** **neighbours.** **iv.** **Given an Array, find the number of inversions of it. If (i and A (i) >A (j)) then the pair (i, j) is called the** **inversion of the array. We need to count all such pairs of inversions of Array.** **v.** **Given an array of integers, find the minimum and maximum elements presents in it by doing** **minimum comparisons by using divide and conquer approach.**	Date : 28/05/2021

Write the code with proper indentation

```
#include <bits/stdc++.h>
using namespace std;

#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;


#define F first
#define S second
#define PB push_back
#define MP make_pair



void firstlast(int n){
int i;int val;
int j;
vector<int>v;
// cout<<"Enter sorted Array"<<endl;
for (i=0;i<n;i++){
```

```
cin>>j;
v.push_back(j);
}
// cout<<"Enter value to be searched"<<endl;
cin>>val;


int mid=0;
int l=0;
int h=n-1;
i=l;
while(true){
if(h<l)
{cout<<"Not found"<<endl;
break;}

mid=(l+h)/2;
if (mid==0 || ( val>v[mid-1] && v[mid]==val)){
cout<<"First instance is "<<mid<<endl;break;
}
else if(v[mid]<val){
l=mid+1;
i=l;
}
else{
h=mid-1;
}
}

mid=0;
l=0;
h=n-1;
i=l;

while(true)
{
if(h<l)
{cout<<"Not found"<<endl;
break;}

mid=(l+h)/2;
if (mid==n-1 || ( val<v[mid+1] && v[mid]==val)){
cout<<"Last instance is "<<mid<<endl;break;
}
else if(v[mid]<val){

l=mid+1;
i=l;
}
else{
```

```
h=mid-1;
n=h;
}
}
}



void binarysearch(int n){
int i;int val;
int j;
vector<int>v;
for (i=0;i<n;i++){
cin>>j;
v.push_back(j);
}

cin>>val;
int mid=0;
int l=0;
int h=n-1;
i=l;

while(true){
if(h<l)
{cout<<"Not found"<<endl;
break;}

mid=(l+h)/2;
if (v[mid]==val){
cout<<"Element found at "<<mid<<endl;break;
}
else if(v[mid]<val){
l=mid+1;
i=l;
}
else{
h=mid-1;
n=h;
}
}

}

void peak(int n, int a[]){
int i;
for(i=0;i<n;i++){
if(i==0 && a[i]>a[i+1]){cout<<a[i]<<" "<<endl;}
else if(i==n-1 && a[i]>a[i-1]){cout<<a[i]<<" "<<endl;}
else if(i>0 && i<n-1 && a[i]>a[i-1] && a[i]>a[i+1]){cout<<a[i]<<" "<<endl;}
```

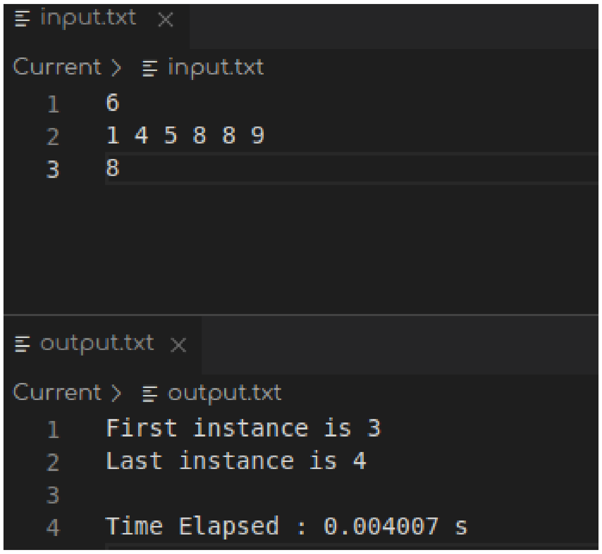
```
}
}




void noofinversions(int n,int a[]){
vector<int>ar;
int i;
for(i=0;i<n;i++){
ar.push_back(a[i]);
}

sort(ar.begin(),ar.end());
int s;s=0;int m;
for(i=0;i<n;i++){
auto k=find(ar.begin(),ar.end(),a[i]);
s=s+distance(ar.begin(),k);
ar.erase(k);
}
cout<<s<<endl;
}



pair<int,int> maxminele(int a[],int l,int h,int n)
{
int minm=INT16_MAX;
int maxm=INT16_MIN;
if(l==h)
{
maxm = a[l];
minm = a[h];
return make_pair(minm,maxm);
}

if(h-l==1)
{
if(a[l]>a[h]){
maxm=a[l];
minm=a[h];
}
else{
maxm=a[h];
minm=a[l];
}

return make_pair(minm,maxm);


}
```

```
int mid=(l+h)/2;
pair<int,int>p1=maxminele(a,l,mid,n);
pair<int,int>p2=maxminele(a,mid+1,h,n);



minm=min(p1.first,p2.first);
maxm=max(p1.second,p2.second);

return make_pair(minm,maxm);
}


int main(){
fast;
int n;
int t = 1;
cin >> n;
int a[n];
for(int i=0;i<n;i++){
cin>>a[i];
}
while(t--){


//i

firstlast(n);


//ii

binarysearch(n);

// //iii

peak(n,a);

// //iv

noofinversions(n,a);

// //v

pair<int,int>pp=maxminele(a,0,n-1,n);
cout<<"Minimum "<<pp.first<<" "<<"Maximum"<<" "<<pp.second<<endl;
}



#ifndef ONLINE_JUDGE
cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
#endif
return 0;

}
```

Take a **screenshot** of your output and show here

Question 3

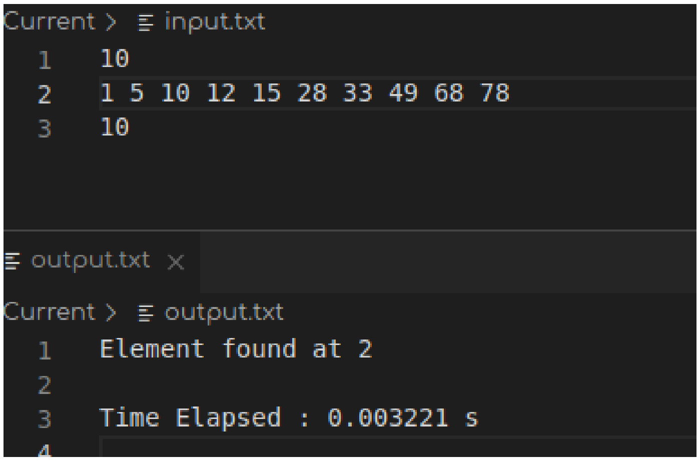
I) First and Last Instance of Element



```
6
1 4 5 8 8 9
8
```

```
First instance is 3
Last instance is 4

Time Elapsed : 0.004007 s
```

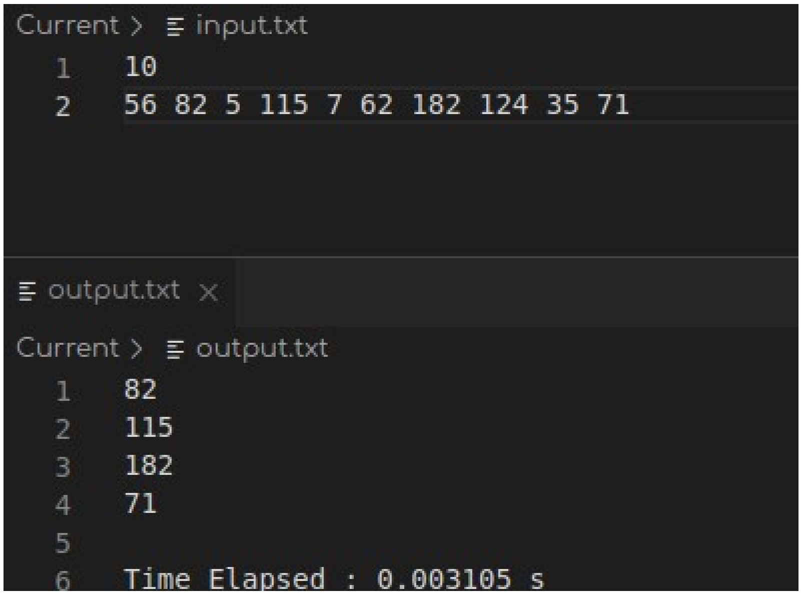
ii) Position of Element



```
10
1 5 10 12 15 28 33 49 68 78
10
```

```
Element found at 2

Time Elapsed : 0.003221 s
```

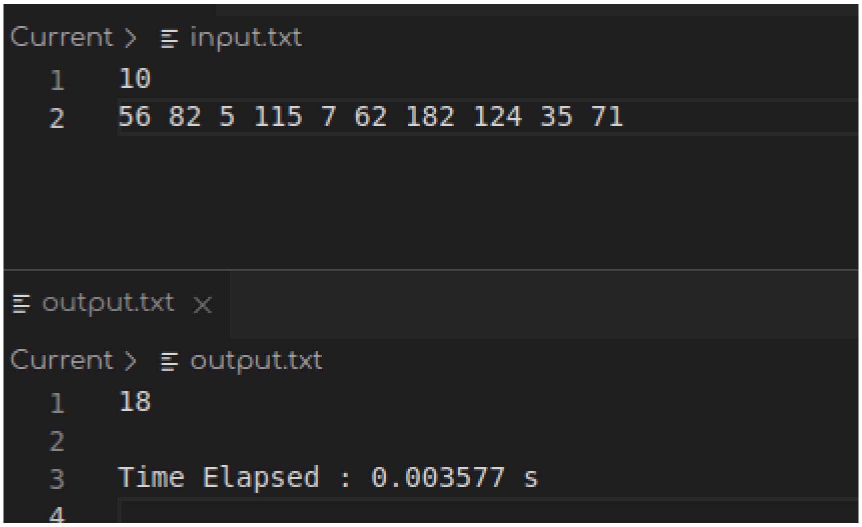
iii)Peak Element



```
10
56 82 5 115 7 62 182 124 35 71
```

```
82
115
182
71

Time Elapsed : 0.003105 s
```

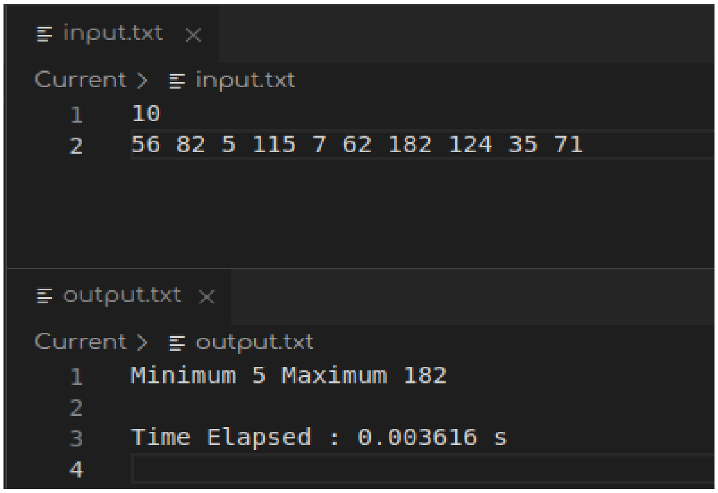
iv)No of inversions



```
10
56 82 5 115 7 62 182 124 35 71
```

```
18

Time Elapsed : 0.003577 s
```

v) Maximum and Minimum element



```
10
56 82 5 115 7 62 182 124 35 71
```

```
Minimum 5 Maximum 182

Time Elapsed : 0.003616 s

```

Instruction:

1 **Don’t try to copy and paste the code from each other or from the internet and write all the lab assignment in the above format only.**

2 **After writing all the lab assignments convert the word file to PDF then submit it in the google classroom in the assignment section.**

3 **All the file names must be your roll number in proper format**