

WEEK-1

Q-1

```
#include<iostream>

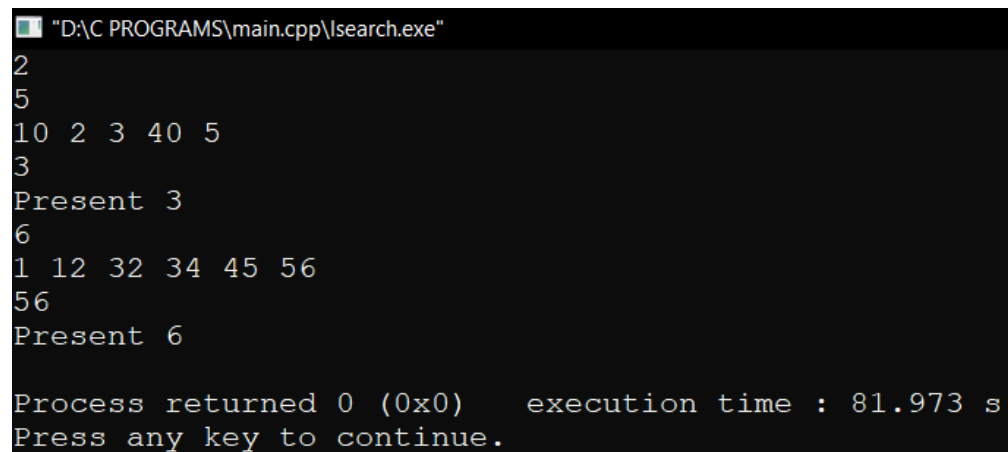
using namespace std;

void linear_search(int arr[],int n,int key)
{
    int comparisions=0,flag=0;
    for (int i=0;i<n;i++)
    {
        comparisions++;
        if (arr[i]==key)
        {
            cout<<"Present"<<" "<<comparisions<<endl;
            flag=1;
            break;
        }
    }
    if (flag==0)
        cout<<"Not Present"<<" "<<comparisions<<endl;
}

int main()
{
    int t;
    cin>>t;
    while (t-->0)
    {
```

```
int n;  
  
cin>>n;  
  
int arr[n];  
  
for (int i=0;i<n;i++)  
    cin>>arr[i];  
  
int key;  
  
cin>>key;  
  
    linear search(arr,n,key);  
}  
  
return 0;  
}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\search.exe"  
2  
5  
10 2 3 40 5  
3  
Present 3  
6  
1 12 32 34 45 56  
56  
Present 6  
  
Process returned 0 (0x0)    execution time : 81.973 s  
Press any key to continue.  
_
```

Q-2

```
#include<iostream>

using namespace std;

int binar_srch(int arr[],int n,int key)
{
    int mid=0;

    int i=0;

    int r=n-1;

    while(i<r)
    {
        mid=(i+r)/2;

        if(key==arr[mid])
        {
            return mid;

            break;
        }

        else if(key>arr[mid])
        {
            i=mid+1;
        }

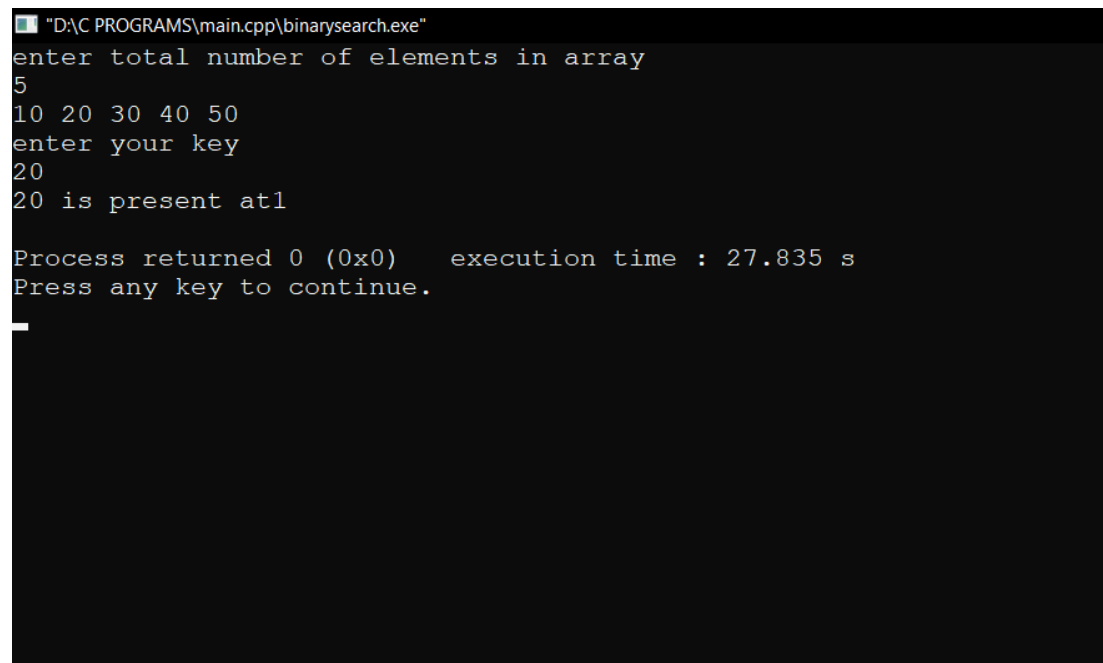
        else
        {
            r=mid-1;
        }

        if(i>r)

        cout<<"element not found"<<endl;}
```

```
int main(){  
    int arr[100];  
    int n,key,c=0;  
    cout<<"enter total number of elements in array"<<endl;  
    cin>>n;  
    for(int i=0;i<n;i++)  
    {  
        cin>>arr[i];  
    }  
    cout<<"enter your key"<<endl;  
    cin>>key;  
    c=binar_srch(arr,n,key);  
    cout<<key<<" "<<"is present at"<<c<<endl;  
}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\binarysearch.exe"  
enter total number of elements in array  
5  
10 20 30 40 50  
enter your key  
20  
20 is present at 1  
  
Process returned 0 (0x0)   execution time : 27.835 s  
Press any key to continue.  
_
```

Q-3

```
#include <bits/stdc++.h>

using namespace std;

void jsearch(int arr[], int size, int key){

    int c = 0;

    int i = 0;

    int j = sqrt(size);

    while (arr[j] < key && i < size){

        c++;

        i = j;

        j += sqrt(size);

        if (j > size - 1)

            j = size - 1;

    }

    for (int k = i; k <= j; k++){

        c++;

        if (arr[k] == key){

            cout << key << " is present\n";

            cout << "no of comparision : " << c << endl;

            return;

        }

    }

    cout << key << " is not present\n";

    cout << "no of comparision : " << c << endl;

}

int main(){
```

```
int x, y, z;

cin >> x;

while (x--){

cin >> y;

int arr[y];

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

cin >> arr[i];

cin >> z;

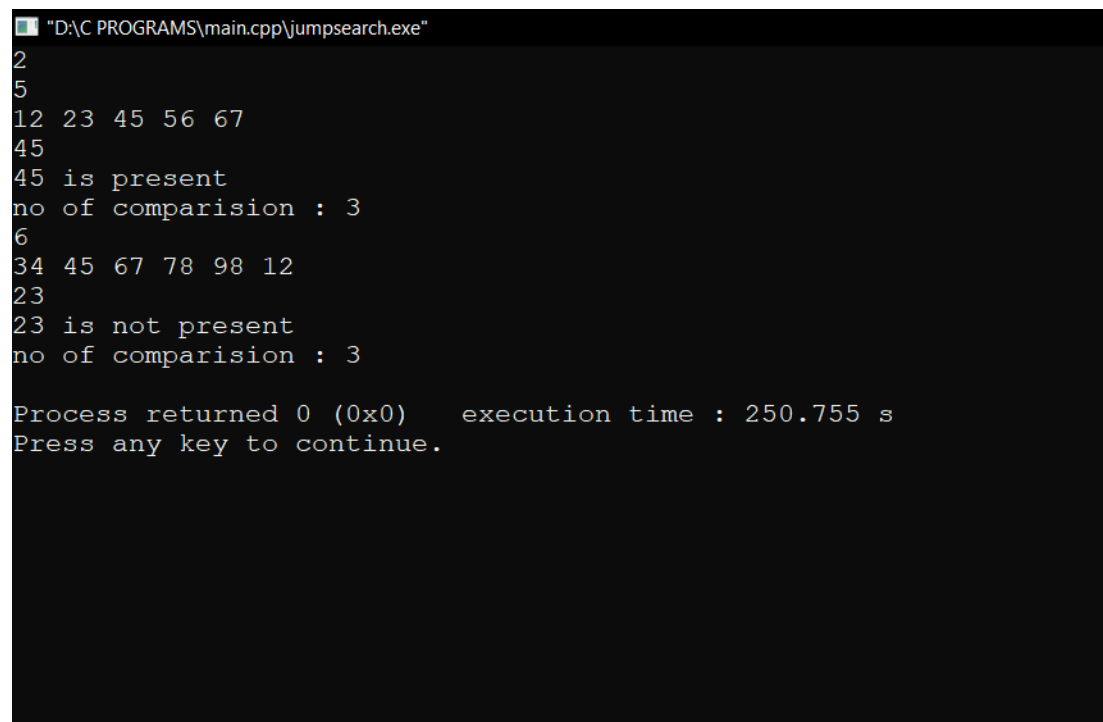
jsearch(arr, y - 1, z);

}

return 0;

}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
5
12 23 45 56 67
45
45 is present
no of comparision : 3
6
34 45 67 78 98 12
23
23 is not present
no of comparision : 3

Process returned 0 (0x0)   execution time : 250.755 s
Press any key to continue.
```

WEEK-2

Q-1

```
#include<iostream>

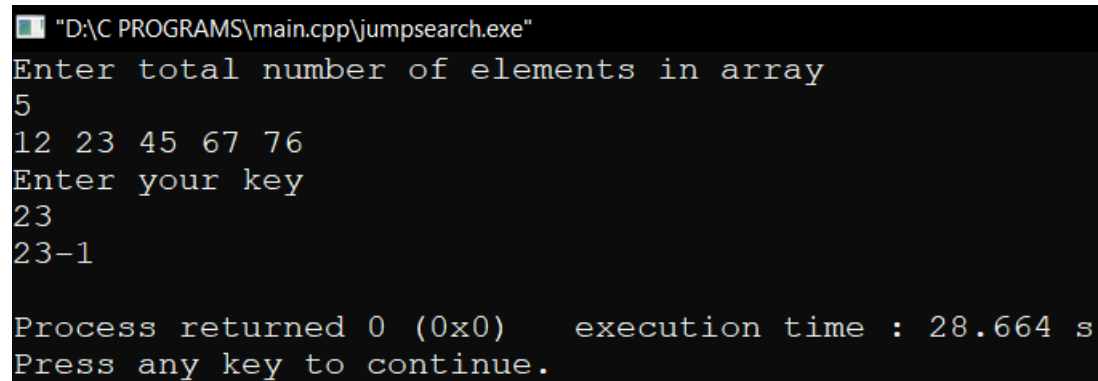
using namespace std;

int duplicate(int arr[],int n,int key)
{
    int flag=0;
    int count=0;
    for(int i=0;i<n;i++)
    {
        if(arr[i]==key&&flag==0)
        {
            flag=1;
            count++;
        }
        else if (arr[i]==key&&flag==1)
        {
            count++;
        }
    }
    return count;
}

int main()
{
    int arr[100];
    int n,key,c=0;
```

```
cout<<"Enter total number of elements in array"<<endl;
cin>>n;
for(int i=0;i<n;i++)
{
cin>>arr[i];
}
cout<<"Enter your key"<<endl;
cin>>key;
c=duplicate(arr,n,key);
cout<<key<<"-"<<c<<endl;
}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
Enter total number of elements in array
5
12 23 45 67 76
Enter your key
23
23-1

Process returned 0 (0x0)   execution time : 28.664 s
Press any key to continue.
```


Q-2

```
#include<iostream>

using namespace std;

int main()
{
    int arr[100];
    int n,key,c=0;
    cout<<"Enter total number of elements in array"<<endl;
    cin>>n;
    for(int i=0;i<n;i++)
    {
        cin>>arr[i];
    }
    for(int k=n-1;k>=0;k--)
    {
        int i=0;
        int j=k-1;
        while(i<j)
        {
            if(arr[i]+arr[j]==arr[k])
            {
                cout<<i<<" "<<j<<" "<<k<<endl;
                break;
            }
            else if(arr[i]+arr[j]>arr[k])
                j--;
        }
    }
}
```

```
else
```

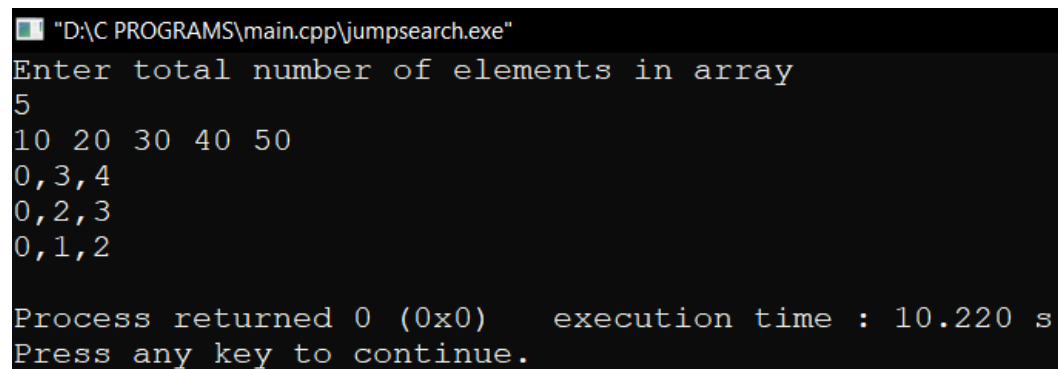
```
i++;
```

```
}
```

```
}
```

```
}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
Enter total number of elements in array
5
10 20 30 40 50
0,3,4
0,2,3
0,1,2

Process returned 0 (0x0)   execution time : 10.220 s
Press any key to continue.
_
```

Q-3

```
#include<iostream>

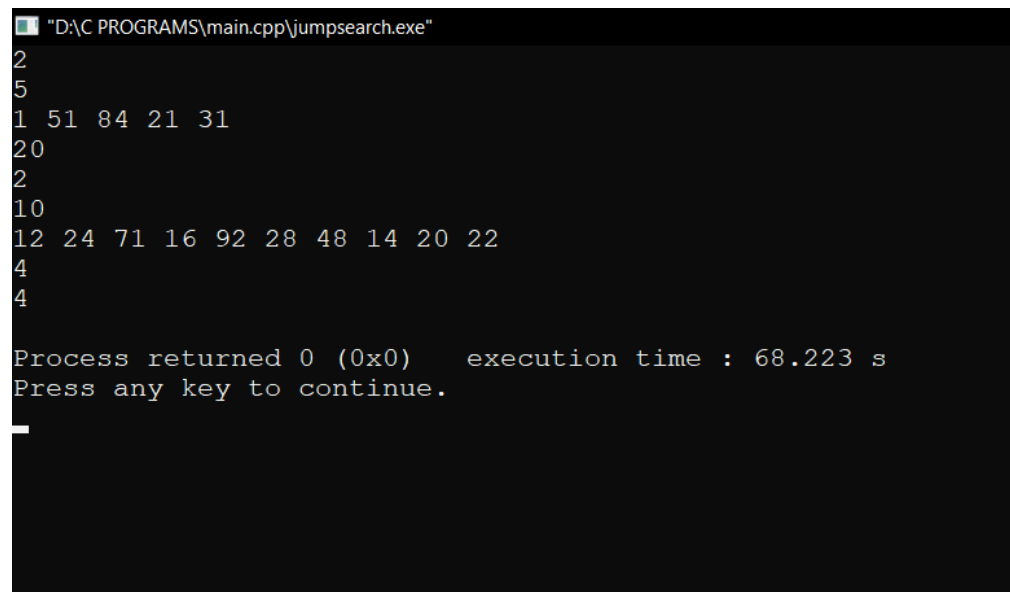
using namespace std;

int differenceequaltok(int arr[],int n,int key)
{
    int count=0;
    for (int i=0;i<n-1;i++)
    {
        for (int j=i+1;j<n;j++)
        {
            if (abs(arr[i]-arr[j])==key)
                count++;
        }
    }
    return count;
}

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];  
    int key;  
    cin>>key;  
    int count=diffrenceequaltok(arr,n,key);  
    cout<<count<<endl;  
}  
return 0;  
}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"  
2  
5  
1 51 84 21 31  
20  
2  
10  
12 24 71 16 92 28 48 14 20 22  
4  
4  
  
Process returned 0 (0x0)   execution time : 68.223 s  
Press any key to continue.  
_
```

WEEK-3

Q-1

```
#include<iostream>

using namespace std;

void insertion_sort(int A[],int n)
{
    int comparisions=0,shift=0;

    int i, j, x;

    for (i = 1; i < n; i++)
    {
        j = i - 1;
        x = A[i];
        while (j > -1 && A[j] > x)
        {
            comparisions++;

            A[j + 1] = A[j];

            j--;
        }
        shift++;

        A[j + 1] = x;
    }

    cout<<"Comparisions:"<<comparisions<<endl;
    cout<<"Shift:"<<shift<<endl;
    for (int i=0;i<n;i++)
        cout<<A[i]<<" ";
}
```

```

int main()
{
    int t;

    cin>>t;

    while (t--)

    {
        int n;

        cin>>n;

        int arr[n];

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

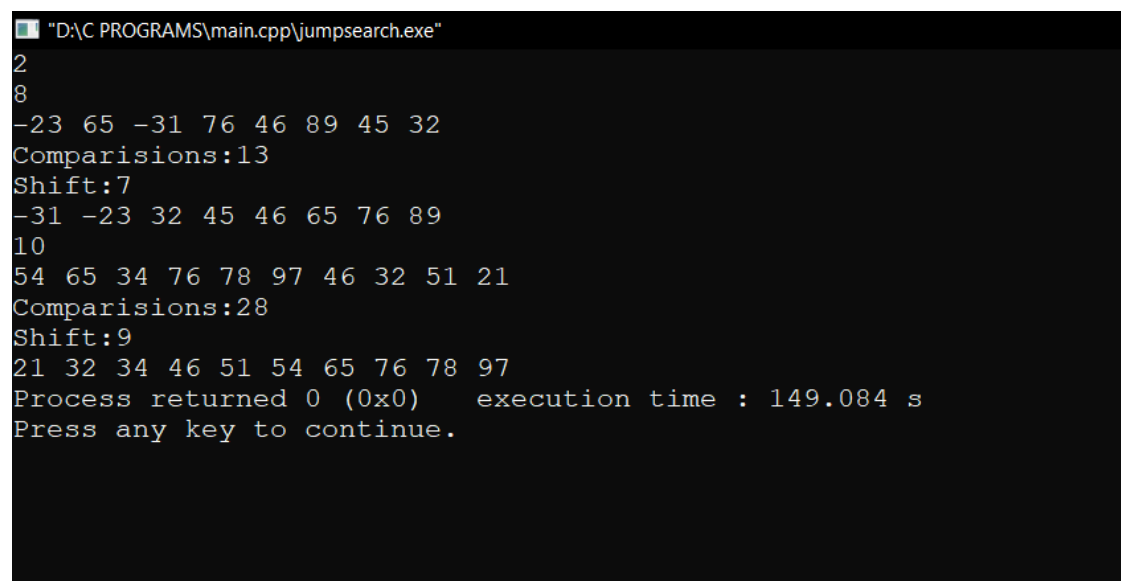
            cin>>arr[i];

        insertion_sort(arr,n);

    }
}

```

OUTPUT



```

"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
8
-23 65 -31 76 46 89 45 32
Comparisions:13
Shift:7
-31 -23 32 45 46 65 76 89
10
54 65 34 76 78 97 46 32 51 21
Comparisions:28
Shift:9
21 32 34 46 51 54 65 76 78 97
Process returned 0 (0x0)   execution time : 149.084 s
Press any key to continue.

```

Q-2

```
#include<iostream>

using namespace std;

void swap(int *x, int *y){

    int temp = *x;

    *x = *y;

    *y = temp;

}

void Selection_sort(int arr[],int n){

    int comp=0,swaps=0;

    int i,j,pos=0,min=0;

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

    {

        min=arr[i];

        pos=i;

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

        {

            comp++;

            if (min>arr[j])

            {

                min=arr[j];

                pos=j;

            } }

        if (pos!=i)

        {swap(&arr[pos],&arr[i]);

            swaps++;

        }
```

```

    }}

    cout<<"Comparisions:"<<comp<<endl;

    cout<<"Swaps:"<<swaps<<endl;

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

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

}

int main(){

    int t;

    cin>>t;

    while (t-- ) {

        int n;

        cin>>n;

        int arr[n];

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

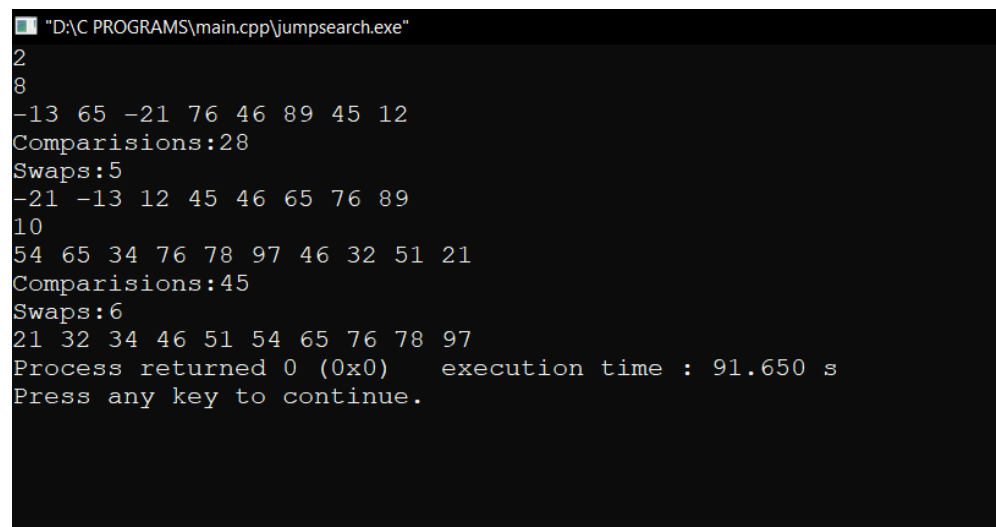
            cin>>arr[i];

        Selection_sort(arr,n);

    }}

```

OUTPUT



```

"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
8
-13 65 -21 76 46 89 45 12
Comparisions:28
Swaps:5
-21 -13 12 45 46 65 76 89
10
54 65 34 76 78 97 46 32 51 21
Comparisions:45
Swaps:6
21 32 34 46 51 54 65 76 78 97
Process returned 0 (0x0)   execution time : 91.650 s
Press any key to continue.

```


Q-3

```
#include<iostream>

using namespace std;

void merge(int arr[],int l,int mid,int h)
{
    int count=0;
    int i=l,j=mid+1;
    int temp[h-l+1];
    int k=0;

    while (i<=mid && j<=h)
    {
        if (arr[i]<arr[j])
            temp[k++]=arr[i++];
        else
        {
            temp[k++]=arr[j++];
            count+=mid-i+1;
        }
    }

    for (;i<=mid;)
        temp[k++]=arr[i++];

    for (;j<=h;)
        temp[k++]=arr[j++];
}
```

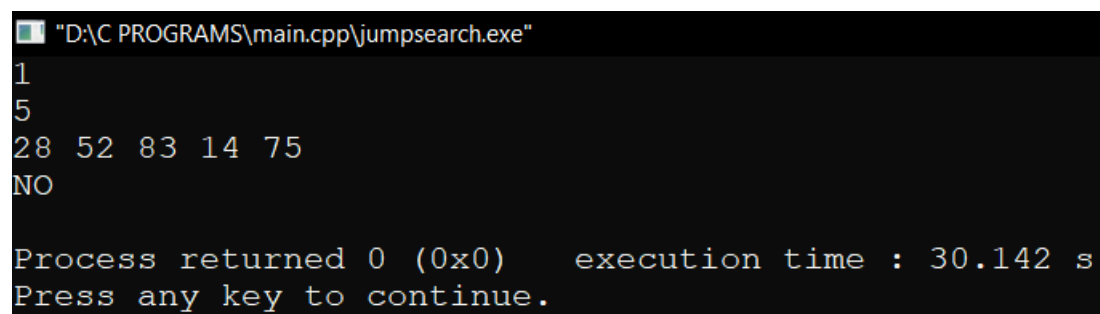
```

        for (int f=0;f<k;f++)
            arr[f+l]=temp[f];
    }
void merge_sort(int arr[],int l,int h)
{
    if (l<h)
    {
        int mid=l+(h-l)/2;
        merge_sort(arr,l,mid);
        merge_sort(arr,mid+1,h);
        merge(arr,l,mid,h);
    }
}
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];
        merge_sort(arr,0,n-1);
        int flag=0;

```

```
    for (int i=0;i<n-1;i++)  
    {  
        if (arr[i]==arr[i+1])  
        {  
            cout<<"YES"<<endl;  
            flag=1;  
        }  
    }  
    if (flag==0)  
        cout<<"NO"<<endl;  
}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"  
1  
5  
28 52 83 14 75  
NO  
  
Process returned 0 (0x0)   execution time : 30.142 s  
Press any key to continue.  
_
```

WEEK-4

Q-1

```
#include <iostream>

using namespace std;

int comp=0;

int merge(int arr[],int l,int mid,int h)
{
    int count=0;
    int i=l,j=mid+1;
    int temp[h-l+1];
    int k=0;

    while (i<=mid && j<=h)
    {
        if (arr[i]<arr[j])
            temp[k++]=arr[i++];
        else
        {
            temp[k++]=arr[j++];
            count+=mid-i+1;
        }
        comp++;
    }
    for (;i<=mid;)
        temp[k++]=arr[i++];
```

```

    for (;j<=h;)
        temp[k++]=arr[j++];

    for (int f=0;f<k;f++)
        arr[f+l]=temp[f];

    return count;
}

int merge_sort(int arr[],int l,int h)
{
    int inversion=0;
    if (l<h)
    {
        int mid=l+(h-l)/2;
        inversion+=merge_sort(arr,l,mid);
        inversion+=merge_sort(arr,mid+1,h);
        inversion+=merge(arr,l,mid,h);
    }
    return inversion;
}

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];

int inv = merge_sort(arr,0,n-1);

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

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

    cout<<endl<<"Inversions:"<<inv<<endl;

cout<<"Comparisions:"<<comp<<endl;

comp=0;

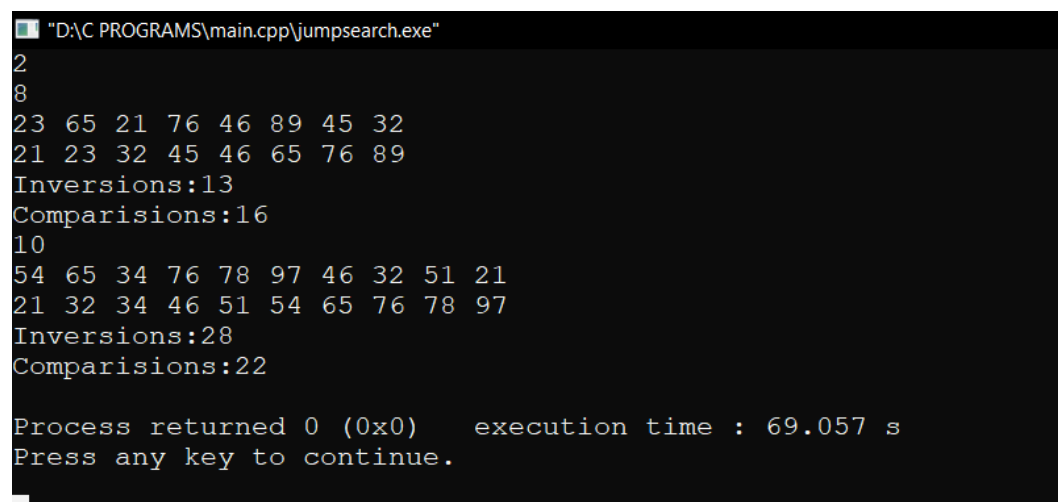
}

return 0;

}

```

OUTPUT



```

"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
8
23 65 21 76 46 89 45 32
21 23 32 45 46 65 76 89
Inversions:13
Comparisions:16
10
54 65 34 76 78 97 46 32 51 21
21 32 34 46 51 54 65 76 78 97
Inversions:28
Comparisions:22

Process returned 0 (0x0)   execution time : 69.057 s
Press any key to continue.

```

Q-2

```
#include <iostream>

using namespace std;

void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

int comp=0,swaps=0;

void quick_sort(int arr[],int lb,int ub)
{
    if (lb>=ub)
        return ;
    else{
        int pivot=arr[lb];
        int i=lb,j=ub;
        while (i<j)
        {
            comp++;
            while (pivot>=arr[i] && i<j)
                i++;
            while (pivot<arr[j])
                j--;
            if (i<j)
                swap(&arr[i],&arr[j]);
        }
    }
}
```

```

    }

    swaps++;

    swap(&arr[lb], &arr[j]);

    quick_sort(arr, lb, j-1);

    quick_sort(arr, j+1, ub);

}

}

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];

        quick_sort(arr, 0, n-1);

        for (int i=0; i<n; i++)
            cout << arr[i] << " ";

        cout << endl << "Swaps:" << swaps << endl;

        cout << "Comparisons:" << comp << endl;

        swaps=0;

        comp=0;

    }
}

```



```
    return 0;  
}
```

OUTPUT

```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"  
1  
8  
23 45 65 21 76 89 45 32  
21 23 32 45 45 65 76 89  
Swaps:5  
Comparisons:8  
  
Process returned 0 (0x0)    execution time : 30.962 s  
Press any key to continue.  
_
```

Q-3

```
#include<iostream>

using namespace std;

void merge(int arr[],int l,int mid,int h)
{
    int count=0;
    int i=l,j=mid+1;
    int temp[h-l+1];
    int k=0;
    while (i<=mid && j<=h)
    {
        if (arr[i]<arr[j])
            temp[k++]=arr[i++];
        else
        {
            temp[k++]=arr[j++];
            count+=mid-i+1;
        }
    }
    for (;i<=mid;)
        temp[k++]=arr[i++];

    for (;j<=h;)
        temp[k++]=arr[j++];

    for (int f=0;f<k;f++)
```

```

        arr[f+l]=temp[f];
    }
void merge_sort(int arr[],int l,int h)
{
    if (l<h)
    {
        int mid=l+(h-l)/2;
        merge_sort(arr,l,mid);
        merge_sort(arr,mid+1,h);
        merge(arr,l,mid,h);
    }
}
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];
        int k;
        cin>>k;
        merge_sort(arr,0,n-1);
    }
}

```

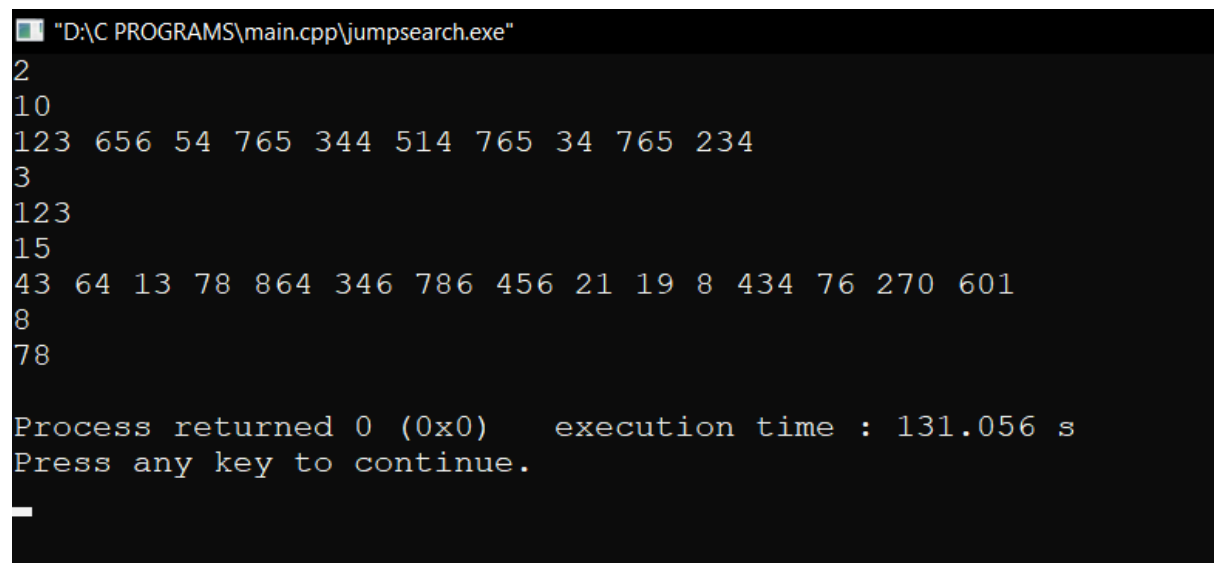
```
    int flag=0;

    cout<<arr[k-1]<<endl;

}

}
```

OUTPUT



```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
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

Process returned 0 (0x0)    execution time : 131.056 s
Press any key to continue.
```

WEEK -5

Q-1

```
#include <iostream>

#include<limits.h>

using namespace std;

void count_sort(char arr[],int n)
{
    int temp[26]={0};
    for (int i=0;i<n;i++)
        temp[arr[i]-97]++;
    int maxi=0;
    char res='$';
    for (int i=0;i<26;i++)
    {
        if (temp[i]>maxi)
        {
            maxi=temp[i];
            res=i+97;
        }
    }
    if (maxi==1)
        cout<<"No Duplicate Found"<<endl;
    else
        cout<<res<<" - "<<maxi<<endl;
}

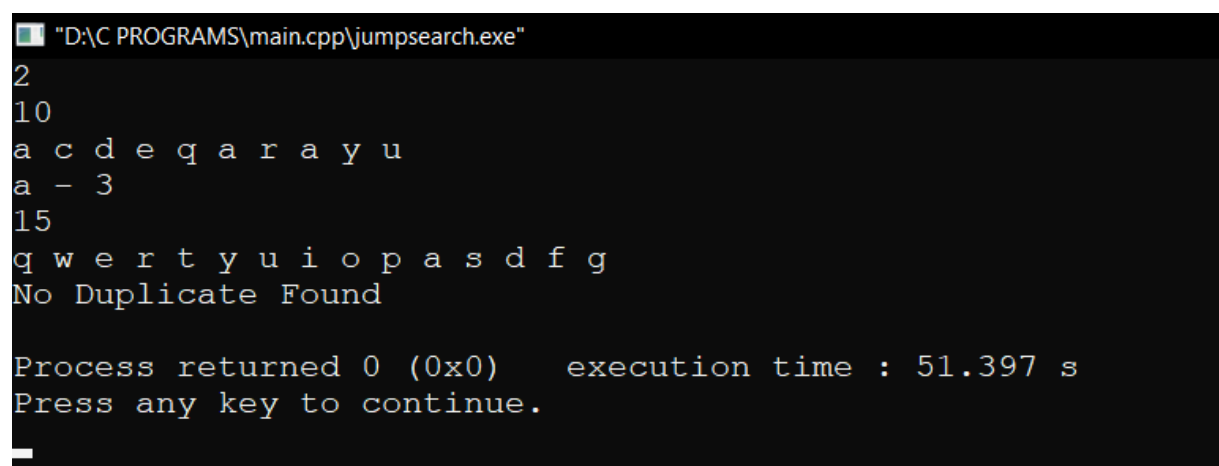
int main()
```

```

{
    int t;
    cin>>t;
    while (t--)
    {
        int n;
        cin>>n;
        char arr[n];
        for (int i=0;i<n;i++)
            cin>>arr[i];
        count_sort(arr,n);
    }
    return 0;
}

```

OUTPUT



```

"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
10
a c d e q a r a y u
a - 3
15
q w e r t y u i o p a s d f g
No Duplicate Found

Process returned 0 (0x0)   execution time : 51.397 s
Press any key to continue.

```

Q-2

```
#include <iostream>

using namespace std;

void merge(int arr[],int l,int mid,int h)
{
    int count=0;
    int i=l,j=mid+1;
    int temp[h-l+1];
    int k=0;
    while (i<=mid && j<=h)
    {
        if (arr[i]<arr[j])
            temp[k++]=arr[i++];
        else
        {
            temp[k++]=arr[j++];
            count+=mid-i+1;
        }
    }
    for (;i<=mid;)
        temp[k++]=arr[i++];

    for (;j<=h;)
        temp[k++]=arr[j++];

    for (int f=0;f<k;f++)
```

```

        arr[f+l]=temp[f];
    }
void merge_sort(int arr[],int l,int h)
{
    if (l<h)
    {
        int mid=l+(h-l)/2;
        merge_sort(arr,l,mid);
        merge_sort(arr,mid+1,h);
        merge(arr,l,mid,h);
    }
}
void find_duplicates(int arr[],int n,int k)
{
    int flag=0;
    int i=0,j=n-1;
    while (i<j)
    {
        if (arr[i]+arr[j]==k)
        {
            flag=1;
            cout<<arr[i]<<"+"<<arr[j]<<"="<<k<<endl;;
            i++;j--;
        }
        else if (arr[i]+arr[j]<k)
            i++;
    }
}

```



```

        else
            j--;
    }
    if (flag==0)
        cout<<"No such pair exist"<<endl;
}

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];
        int key;
        cin>>key;
        merge_sort(arr,0,n-1);
        find_duplicates(arr,n,key);
    }
    return 0;
}

```

OUTPUT

```
"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
2
10
64 28 97 40 12 72 84 24 38 10
50
10+40=50
12+38=50
15
56 10 72 91 29 3 41 45 61 20 11 39 9 12 94
302
No such pair exist

Process returned 0 (0x0)    execution time : 149.912 s
Press any key to continue.
```

Q-3

```
#include<iostream>

#include<limits.h>

using namespace std;

void intersection(int arr1[],int n1,int arr2[],int n2)
{
    int maxi1=INT_MIN;
    for (int i=0;i<n1;i++)
    {
        if (arr1[i]>maxi1)
            maxi1=arr1[i];
    }
    int temp1[maxi1+1]={0};
    for (int i=0;i<n1;i++)
        temp1[arr1[i]]++;
    int maxi2=INT_MIN;
    for (int i=0;i<n2;i++)
    {
        if (arr2[i]>maxi2)
            maxi2=arr2[i];
    }
    int temp2[maxi2+1]={0};
    for (int i=0;i<n2;i++)
        temp2[arr2[i]]++;

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

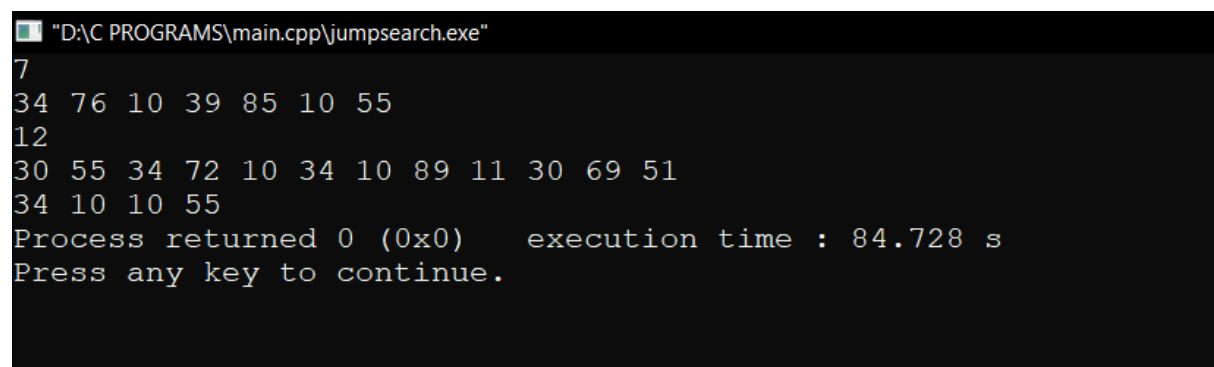
```

    {
        if (temp1[arr1[i]]!=0 && temp2[arr1[i]]!=0)
            cout<<arr1[i]<<" ";
    }
}

int main()
{
    int n1;
    cin>>n1;
    int arr1[n1];
    for (int i=0;i<n1;i++)
        cin>>arr1[i];
    int n2;
    cin>>n2;
    int arr2[n2];
    for (int i=0;i<n2;i++)
        cin>>arr2[i];
    intersection(arr1,n1,arr2,n2);
    return 0;
}

```

OUTPUT



```

"D:\C PROGRAMS\main.cpp\jumpsearch.exe"
7
34 76 10 39 85 10 55
12
30 55 34 72 10 34 10 89 11 30 69 51
34 10 10 55
Process returned 0 (0x0)   execution time : 84.728 s
Press any key to continue.

```

