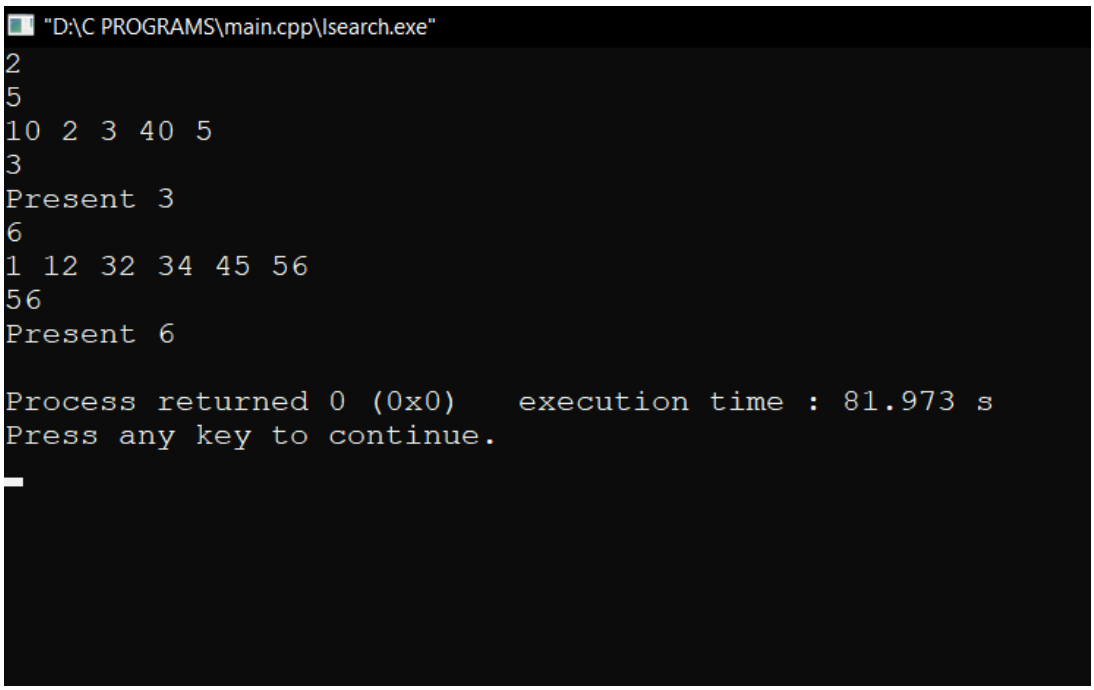
WEEK-1

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--)
   {
```

```
        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

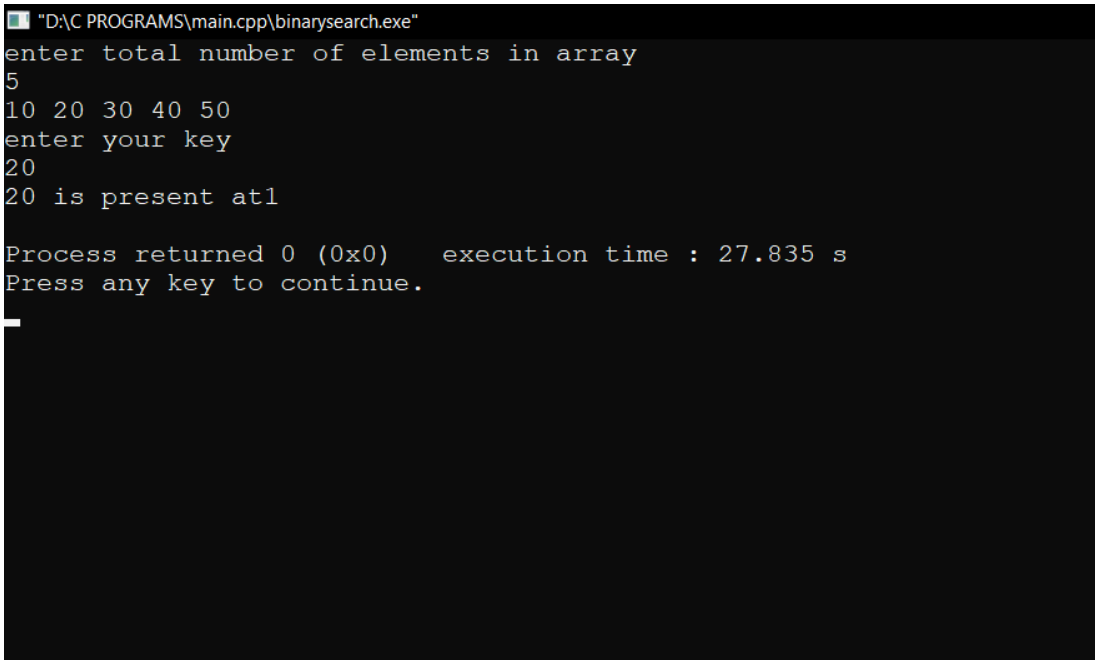


Q-2

```
#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

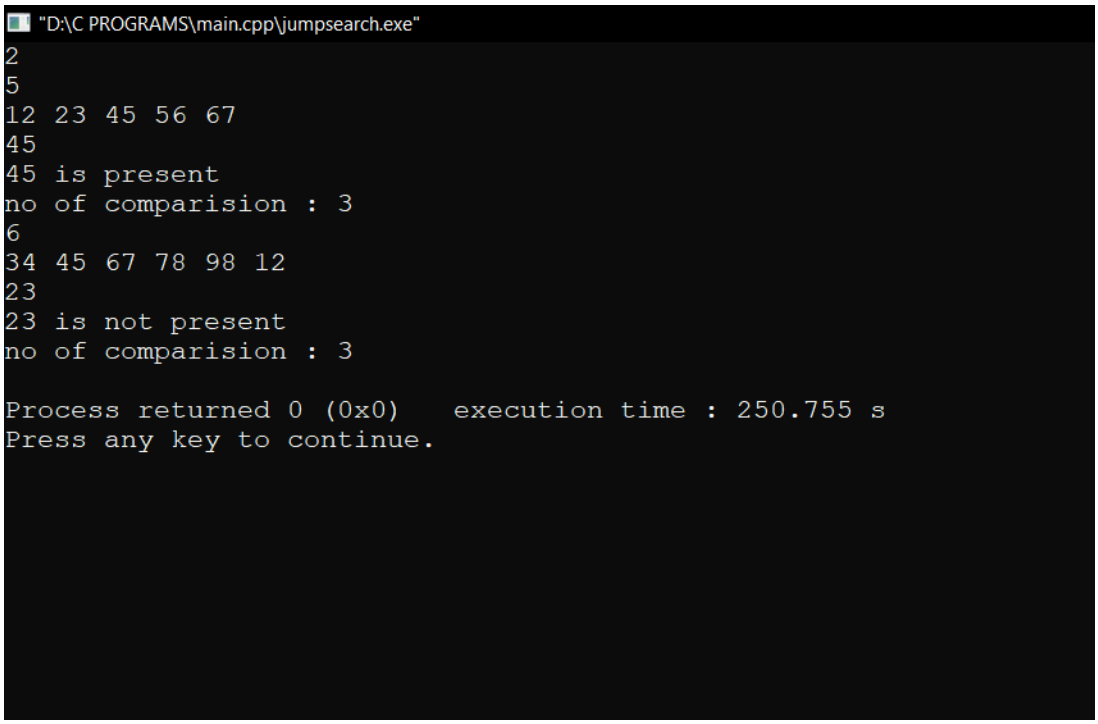


Q-3

```
#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



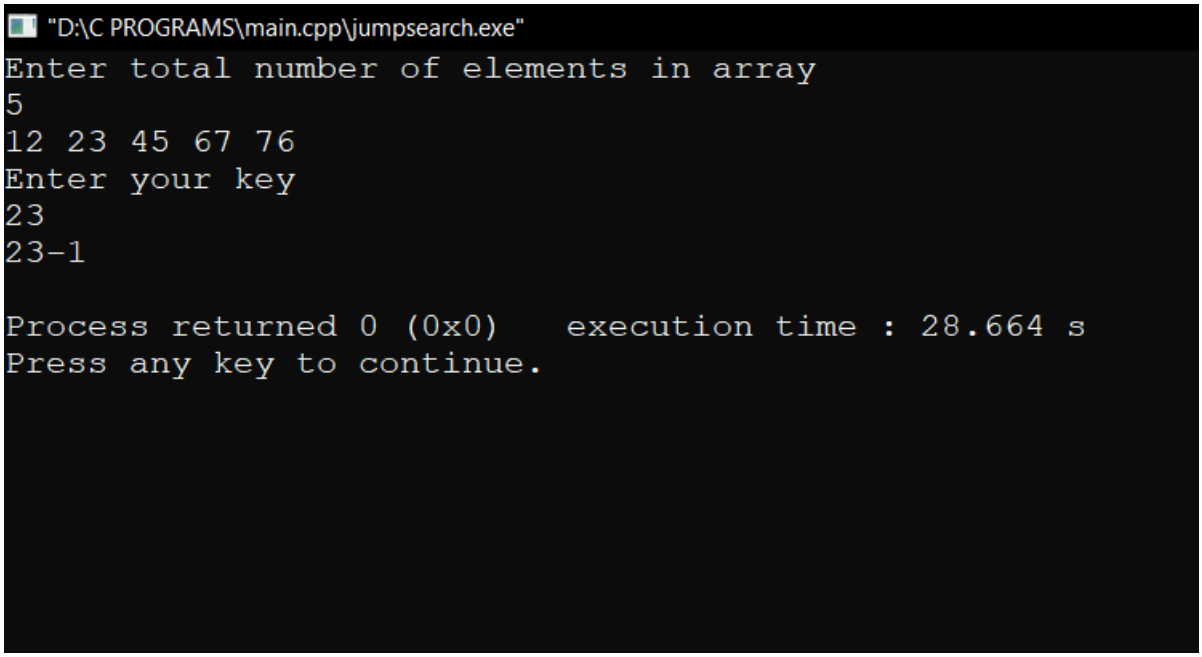
WEEK-2

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

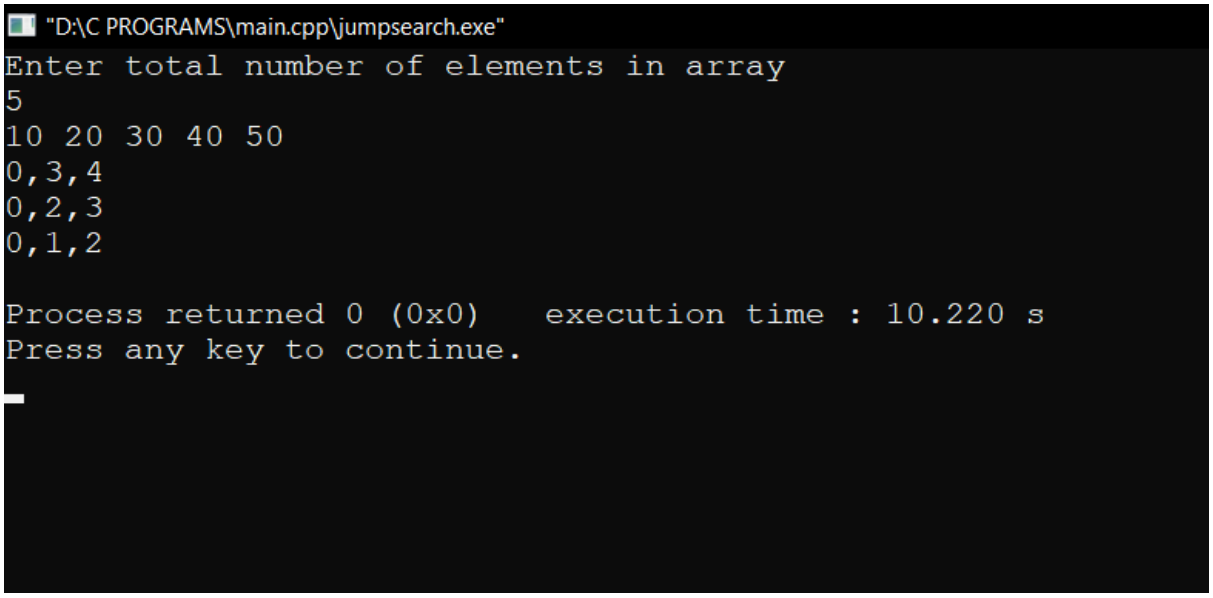


Q-2

```
#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

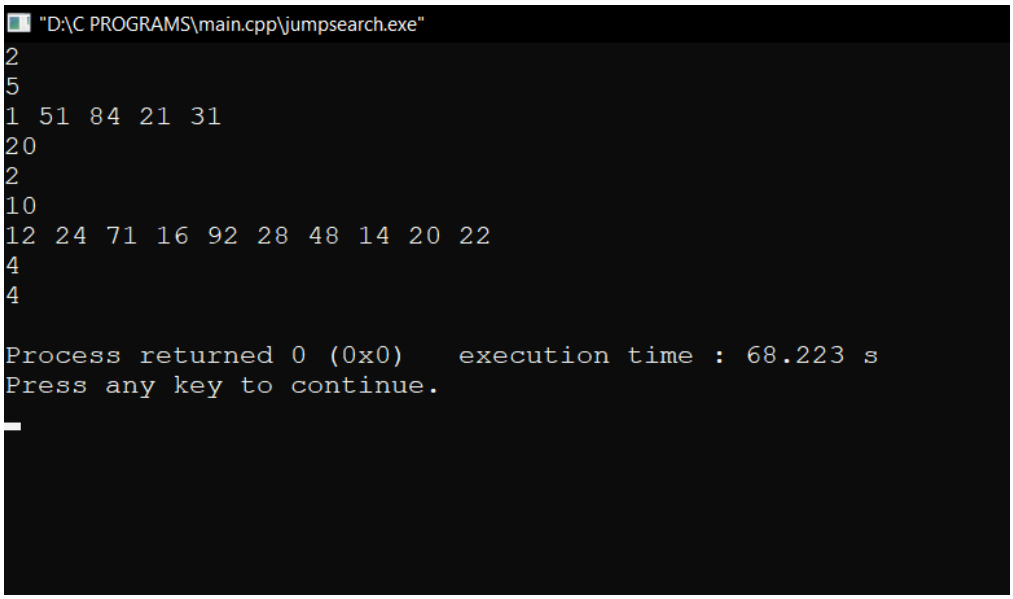


Q-3

```
#include<iostream>
using namespace std;
int diffrenceequaltok(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



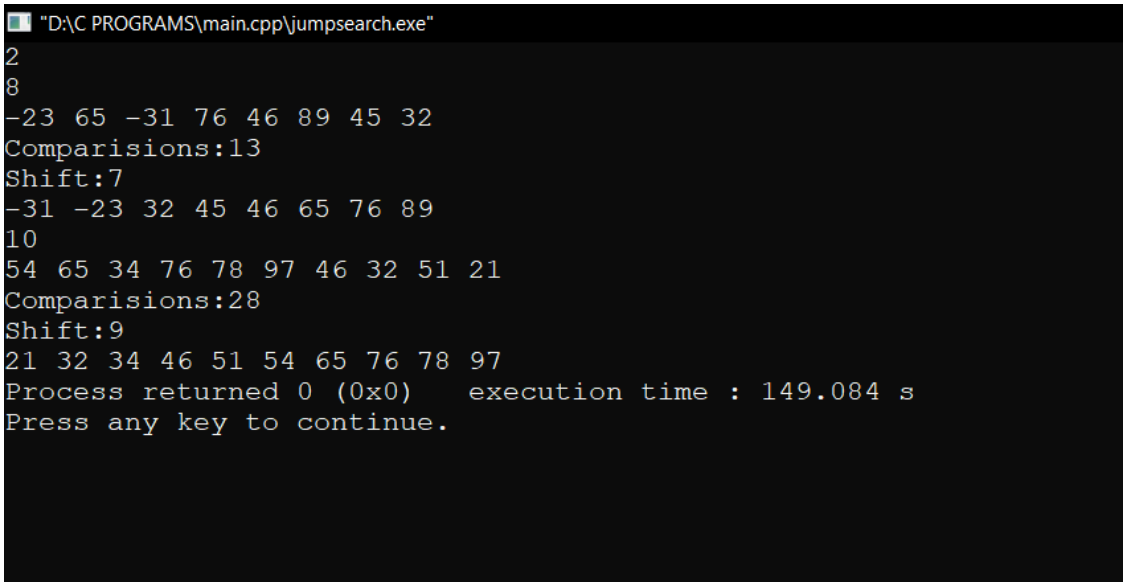
WEEK-3

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

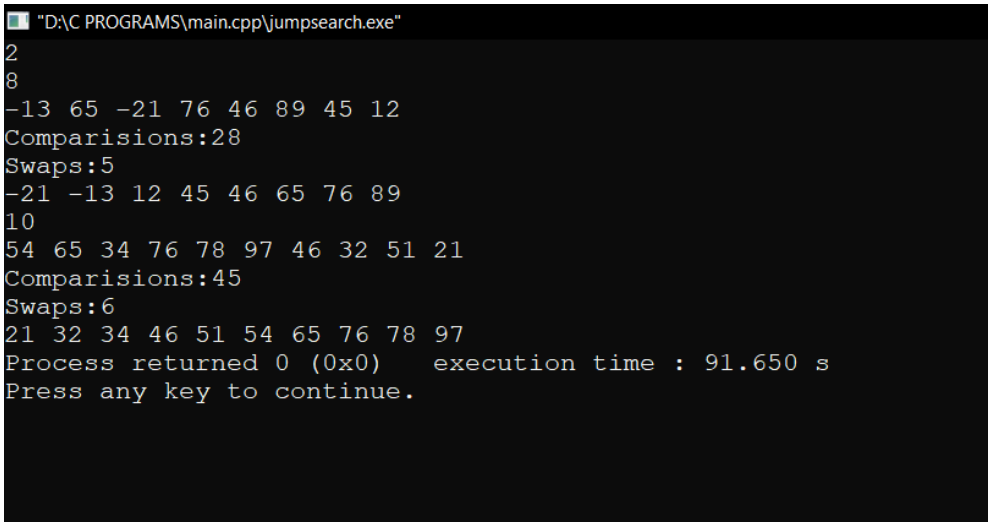


Q-2

```
#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



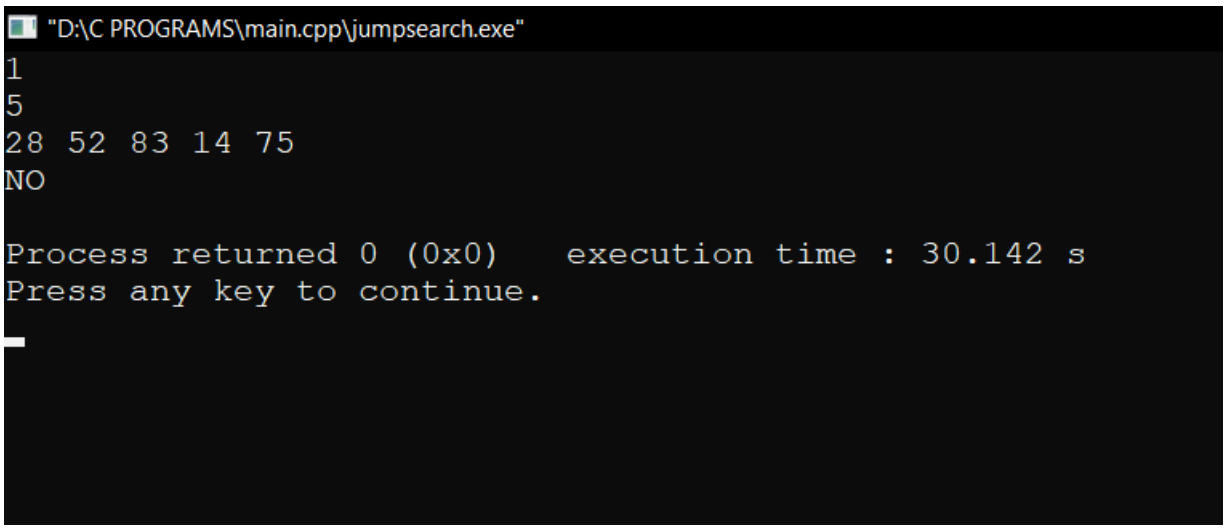
Q-3

```
#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



WEEK-4

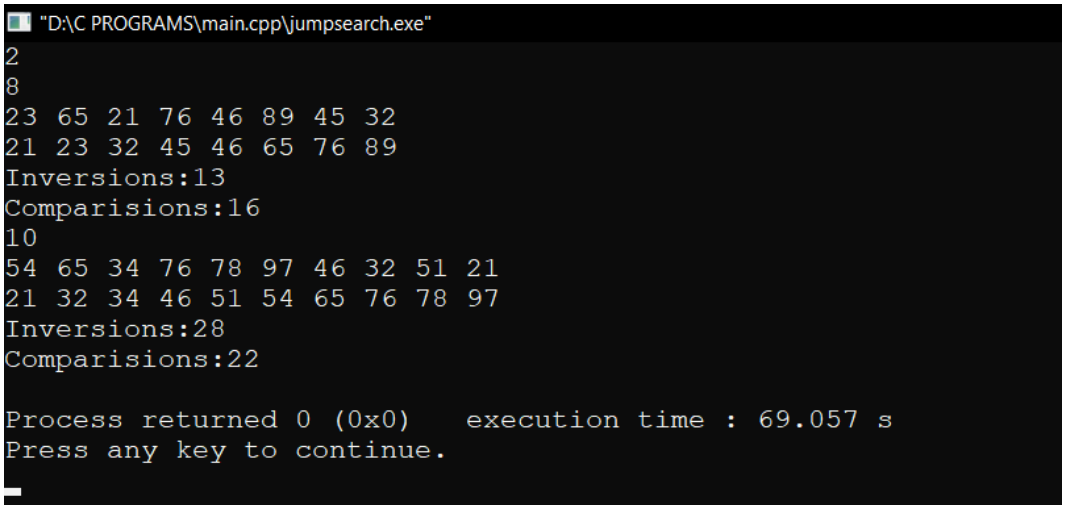
Q-1

```
#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



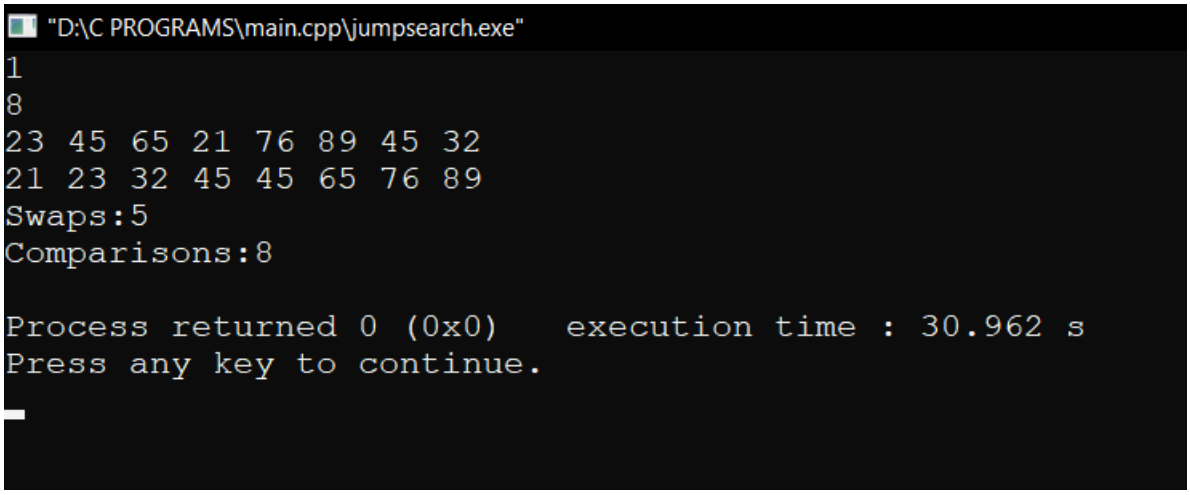
Q-2

```
#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



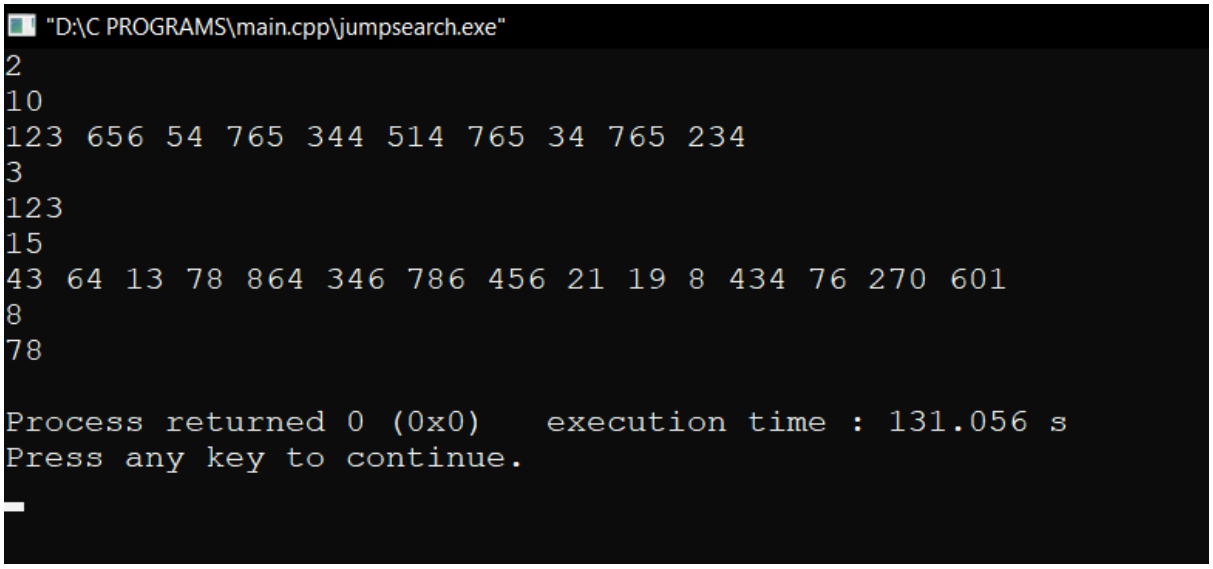
Q-3

```
#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



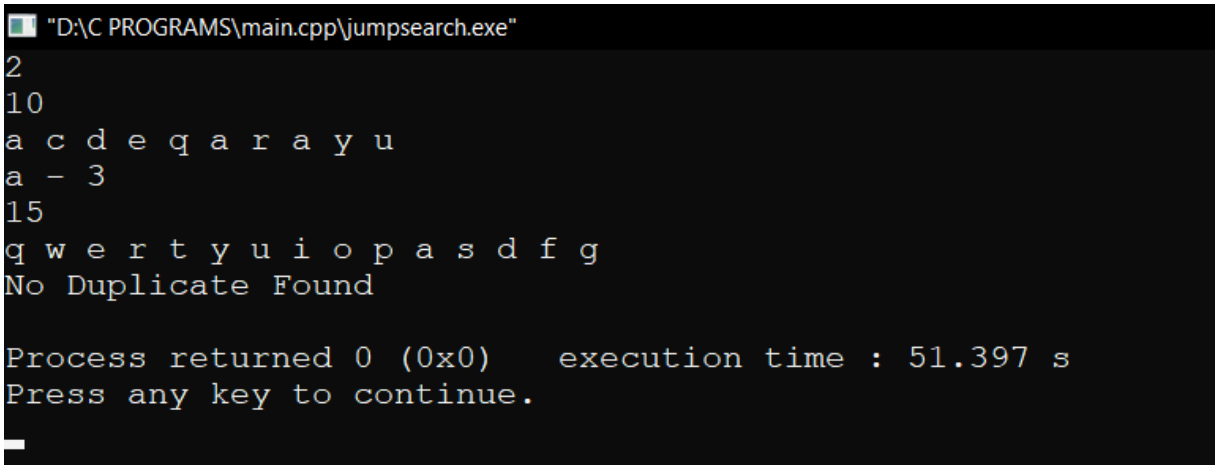
WEEK -5

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



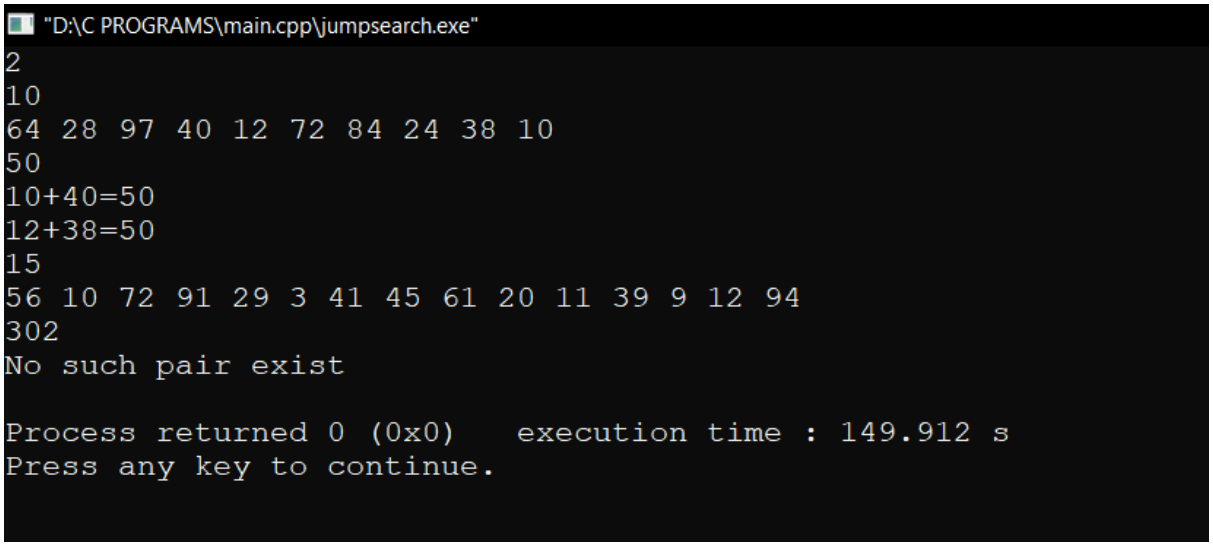
Q-2

```
#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