

Q1.

```
#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;
```

```
    cout<<"enter t";
```

```
    cin>>t;
```

```
    while (t--)
```

```
    {
```

```
        int n;
```

```
        cout<<"enter n";
```

```
        cin>>n;
```

```
        char arr[n];
```

```
        cout<<"enter the character";
```

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

```
            cin>>arr[i];
```

```
        count_sort(arr,n);
```

```
    }
```

```
    return 0;
```

```
}
```

OUTPUT-

enter t3

enter n10

enter the character a e d w a d q a f p

a - 3

enter n15

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

No Duplicate Found

enter n20

enter the character g t l l t c w a w g l c w d s a a v c l

l - 4

Q2.

```
#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;

    cout<<"enter t";

    cin>>t;

    while (t-->0)
    {
        int n;

        cout<<"enter n";

        cin>>n;

        int arr[n];

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

            cin>>arr[i];

        int key;

        cout<<"enter key";

        cin>>key;

        merge_sort(arr,0,n-1);

        find_duplicates(arr,n,key);
    } return 0;}

```

OUTPUT-

enter t2

enter n10

64 28 97 40 12 72 84 24 38 10

enter key50

10+40=50

12+38=50

enter n15

56 10 72 91 29 3 41 45 61 20 11 39 9 12 94

enter key302

No such pair exist

Q3.

```
#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;

    cout<<"enter number 1";

    cin>>n1;

    int arr1[n1];

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

        cin>>arr1[i];

        int n2;

    cout<<"enter number 2";

    cin>>n2;

    int arr2[n2];

    for (int i=0;i<n2;i++)

        cin>>arr2[i];

    intersection(arr1,n1,arr2,n2);

    return 0;
}

```

OUTPUT-

enter number 1 7

10 10 34 39 55 76 85

enter number 2 12

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

10 10 34 55
