Q1. Count the number of pairs whose sum is equal to given k.
Const:-
1 <= n <= 100000
1 <= k <= 100000

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
int freq[100005]; //store freq count
int arr[100005];
int main(){
        int n,k;
        long long ans=0;
        scanf("%d%d",&n,&k);
        for(int i=0; i < n; i++)
                cin>>arr[i];
        for(int i=0; i < n; i++){
                freq[arr[i]]++;
        }
        for(int i=0; i <= k; i++){
                // for i==k-i
                // Suppose, k=2 and arr={1,1,1} freq[1] is 3 and freq[2-1] is also three
                // So we need to add freq[i]*(freq[k-i]-1) only
                if(i == k-i)
                        ans += (ll)freq[i]*(freq[k-i]-1);
                else
                        ans += (ll)freq[i]*freq[k-i];
        }
        ans = ans/2;
        cout<<ans;
        return 0;
}
```

Q2. Given a sorted array of size n. Find the upper bound of a given number, say k. (Upper bound is the element which is just greater than the given number)

```
#include<bits/stdc++.h>
using namespace std;
int main(){
        int n,k,arr[1000];
        cin>>n>>k;
        for(int i=0; i < n; i++)
                cin>>arr[i];
        for(int i=0; i<n ; i++)
                if(arr[i]<=k)
                        cout<<"No ";
                else cout<<"Yes ";
        cout<<endl;
        int lo=0, hi=n-1, ans=-1;
        while( lo <= hi ){
                int mid = (lo + hi)/2;
                if(arr[mid] > k){
                        ans = mid;
                        hi = mid-1;
                }
                else
                        lo = mid+1;
        }
        if( ans==-1 )
                cout<<"All numbers are smaller";
        else cout<<arr[ans]<<endl;
        return 0;
}
```

Q3. The strength of a person A is given number by k. Find the minimum number of opponents needed to defeat A if the strength of let’s say n opponents is sum of numbers from 1 to n. A can be defeated if the total strength of opponents is more than A’s strength.

Const:-

1 <= t <= 100000

1 <= k <= 10^18

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
int main(){
        int t;
        scanf("%d",&t);
        while(t--){
                ll k;
                ll lo = 1 , hi = 2e9, ans=-1;
                scanf("%lld", &k);
                while( lo <= hi ){
                        ll mid = (lo+hi)/2;
                        if( k < (mid*(mid+1))/2 ){
                                ans=mid;
                                hi = mid-1;
                        }
                        else
                                lo = mid+1;
                }
                printf("%lld\n", ans);
                return 0;
        }
}
```

Q4. Find the minimum size of subarray which contains all the letters of alphabet atleast once.

Const:-

1 <= n <= 100000

```
#include <bits/stdc++.h>
using namespace std;
int freq[27],n;
char arr[100005];
int totDisCharacters(){
        int cnt = 0;
        for(int i = 0; i<26 ; i++)
                if(freq[i])
                        cnt++;
        return cnt;
}
bool solve(int wind){
        //creating initial window
        for(int i = 0; i < wind; i++)
                freq[arr[i]-'a']++;
        if( totDisCharacters() == 26)
                return true;
        for(int i = wind; i < n; i++){
                //shifting window
                freq[ arr[i] - 'a' ]++;
                freq[arr[i-wind]-'a']--;

                if( totDisCharacters() == 26 )
                        return true;
        }
        for(int i=0; i<26 ; i++)
                freq[i]=0;
        return false;
}
int binarySearch(){
        int lo = 1, hi = n, ans =- 1;
        while(lo <= hi){
                int mid = (lo + hi)/2;
                if( solve(mid) == true ){
                        ans = mid;
                        hi = mid-1;
                }
                else lo = mid+1;
        }
        return ans;
}
int main(){
        // Input Format -> int n
        // Character Array-> like asdfabcdefghijklmnospqrstuvwxyzadf
        scanf("%d%*c",&n);
        for(int i=0; i<n ; i++){
                scanf("%c",arr+i);
        }
        cout<<binarySearch();
        return 0;
}
```