SANTA AND GIFTS

Problem statement

Santa Claus has *n* gifts, he dreams to give them to children.

On the Christmas eve n events were organised and winner of each event was given a gift. Santa wants to give another special gift to a child who wins the maximum number of games. Help santa to find the child if the name of winners of each individual game are given.

Approach

- Here we need to find the string which appearing maximum number of times in array.
- Brute force approach – For each string count the no. it is appearing in array by another loop and if it is maximum then update the ans accordingly.
  Time complexity- O(n^2)

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    long long int n,m=0,c,i,j;
    cin>>n;
    string a[n];
    string ans;
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    for(i=0;i<n;i++)
    {
        c=0;
        for(j=0;j<n;j++)
        {
            if(a[j]==a[i])
            c++;
        }
        if(c>m)
        {ans=a[i];m=c;}
    }
    cout<<ans;
}
```

- Another approach is using sorting, all the same strings will be together and thus we can count the no. of each string in one loop.

Time complexity- O(nlogn)

- Use hashing – in a map the count of each string can be stored and that value can be accessed in O(1) time by using the string as key.

  Reference- https://www.geeksforgeeks.org/unordered_map-in-cpp-stl/

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    long long int n,maxi=0,i,j;
    cin>>n;
    string a[n];
    string ans;
    unordered_map<string,int>m;
    for(i=0;i<n;i++)
    {
        cin>>a[i];
    }
    for(i=0;i<n;i++)
    {
        if(m.find(a[i])==m.end())
        m[a[i]]=1;
        else m[a[i]]++;
        if(m[a[i]]>maxi)
        {
            ans=a[i];
            maxi=m[a[i]];
        }
    }
    cout<<ans;
}
```

Time complexity- O(n)