Arrays as Hashmap

arr: [2, 4, 4, 2, 5, 6, 8, 2]

frequency → no. of times elements are present.

2 → 3
4 → 2
5 → 1
6 → 1
8 → 1.

	0	1	2	3	4	5	6	7	8
freq:	0	0	3	0	2	1	1	0	1

freq[8] → 1

freq[6] → 1

freq[2] → 3.

Step 1:- Find the maximum element in the array

```
int max = Integer.MIN_VALUE;
for(int i=0; i<n; i++){
    if(arr[i]>max){
```

```
for(
    if(arr[i]>max){
    max=arr[i];
    }
}
int freq[] = new int[max+1]
```

Step2:- Traverse the given array

```
freq[arr[i]]++;
```

[2,4,4,2,5,6,8,2]

0 1 2 3 4 5 6 7 8

[0 0 3 0 2 1 1 0 1]

```
for(int i=0; i<n; i++){
  freq[arr[i]]++;
  }
```

freq[2]++

2,4 4 2 5 6 2 8

Step3: Print the frequency of each element

```
for(int i=0; i<n; i++){
 if(freq[arr[i]]!=0){
  S.O.Pln(arr[i]+" "+freq[arr[i]]);
    freq[arr[i]]=0;
  }
}
```

```
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++){
            arr[i] = sc.nextInt();
        }
        int max = Integer.MIN_VALUE;
        for(int i=0;i<n;i++){
            if(arr[i]>max){
                max = arr[i];
            }
        }
        int freq[] = new int[max+1];
        for(int i=0;i<n;i++){
            freq[arr[i]]++;
        }
        int maxfreq = Integer.MIN_VALUE;
        int key = arr[0];
        for(int i=0;i<n;i++){
            if(freq[arr[i]]>maxfreq){
                maxfreq=freq[arr[i]];
                key= arr[i];
            }
        }
        System.out.println(key);
    }
```

Print freq of Alphabet in string

str: abcdaccd

a → 2 → 97
b → 1 → 98
c → 3 → 99
d → 1 → 100

freq: [0 1 2 ..... 97 98 99 100
2 1 3 1]

```
int freq[] = new int[256];
char ch[] = str.toCharArray();
['a','b','c','d','a','c','c','d']
for(int i=0; i<ch.length; i++){
    freq[ch[i]]++;
}
for(int i=0; i<ch.length; i++){
    if(freq[ch[i]] != 0){
S.O.Pln(ch[i]+" " + freq[ch[i]]);
    freq[ch[i]]=0;
```

a 0
b 1
c 3
d 1
a 2

Maximum Freq Character

str = abcdaccd

a -> 2
b -> 1
c -> 3
d -> 2

Soln:- 1. Find freq of each character

```
int maxfreq = Integer.MIN_VALUE;
char key = ch[0];
for(int i=0; i<ch.length; i++) {
    if(freq[ch[i]] > maxfreq){
        maxfreq = freq[ch[i]];
        key = ch[i];
    }
}
S.o.p(key);
```

Good String checker

Str = abacbc

To be good string, frequency of all characters should be same.

a → 2
b → 2
c → 2

Example 2:

addc

a-1
d-2
c-1

X (No)

Solution:-

1. Find frequency of each character

Str: abacbc

ch[] = ['a', 'b', 'a', 'c', 'b', 'c']

freq: [0 . . . . . 97 98 99 100 . . . . . 255]
2 2 2

2, 2, 2, 2, 3, 2

boolean isgood = true;

for(int i=1; i<ch.length; i++) {

```
boolean
for(int i=1; i<ch.length; i++) {
  if(freq[ch[i-1]] != freq[ch[i]]) {
        isgood=false;
      s.o.pln("false");
        break;
    }
  }
  if(isgood==true){
    s.o.pln("true");
  }
```