int max Count = 3
int ans = 3

for (int i=0; i<10; i++)

if (freq[i] > max Count)
    max Count = freq[i];
    ans = i;

syso (ans);

$n = 2\ 9\ 3\ 4\ 3\ 1\ 6\ 7\ 3$

0-9

0	1	1	3	0	0	1	1	0	1
0	1	2	3	4	5	6	7	8	9

$n \% 10 = 7\ 6\ 8\ 3\ 4\ 3\ 2$

$n / 10 = 0$

Digit with highest frequecy
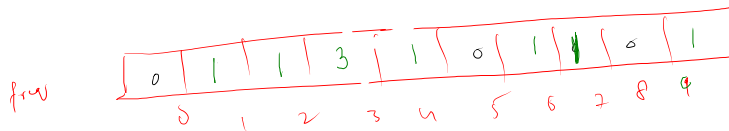
| Problem | Submissions | Leaderboard | Discussions |

Given a number as an integer input. Check which digit occurs the maximum number of times. Print that digit.   O(n)

$n\% 10 = 7\ 6\ 8\ 3\ 4\ 3\ 2$

$n / 10 = 0$

$n =$ $2\ 9\ 3\ 4\ 3\ 1\ 6\ 7\ 3 \% 10 = 3$

$n / 10 = 0$

freq

0	1	1	3	1	0	1	1	0	1
0	1	2	3	4	5	6	7	8	9

int maxCount = 0
ans = 0

for (int i=0; i<10; i++)

if (freq[i] > max Count)
    max Count = freq[i];
    ans = i;
)
) ⟶ syso (ans);

Given an array that contains only one-digit integers from 0-9. Print the digit which occurs in the array the maximum number of times. You must take a constant extra space and also you have to complete the question in the single traversal.

* Array is sorted

```java
public static void maxFreq(int arr[]){
    int n=arr.length;
    int freq[]=new int[10];

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

    int maxCount=0;
    int ans=0;

    for(int i=0;i<10;i++){
        if(freq[i]>maxCount){
            maxCount=freq[i];
            ans=i;
        }
    }

    System.out.println(ans);
}
```
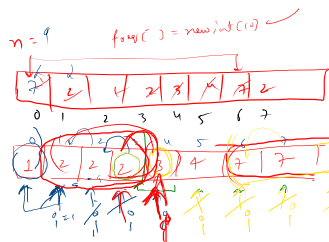
maxCount = 0
ans
int Count = 1

n = 9

for[ ] = new int[10]

| 2 | 2 | 1 | 2 | 3 | 4 | 7 | 2 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| 1 | 2 | 2 | 2 | | 4 | | 7 | 7 | 7 |

while (i < n) {
    if (aor[i] == arr[i-1])
    {
        count++;
    }
}

max count = 1 3

count = 1 2 3 1 2 3 4

ans = 1

Syso (ans);

I < 3

max count = 3
ans = arr[i-1]

else {
    if (max count < count)
    {
        maxCount = count
        ans = arr[i-1];
    }
    count = 1;
}
i + 1;
}

Sysout

if (max count < count)

ans = arr[i-1];

7

Syso (ans)

Lucky Number 26

Problem	Submissions	Leaderboard	Discussions

Given an array of integers arr, a lucky integer is an integer that has a frequency in the array equal to its value.
Return the largest lucky integer in the array. If there is no lucky integer return -1.

Sample Input 0

```
4
2
2
3
4
```

Sample Output 0

```
2
```