题目描述：给定一个正整型数组表示待系统执行的任务列表，数组的每一个元素代表一个任务，元素的值表示该任务的类型。请计算执行完所有任务所需的最短时间。任务执行规则如下：
1、任务可以按任意顺序执行，且每个任务执行耗时间均为1个时间单位。
2、两个同类型的任务之间必须有长度为N个单位的冷却时间，比如：N为2时，在时间K执行了类型3的任务，那么K+1和K+2两个时间不能执行类型3任务。
3、系统在任何一个单位时间内都可以执行一个任务，或者等待状态。
说明：数组最大长度为1000,数组最大值1000.

输入描述：第一行记录一个用半角逗号分隔的数组，数组长度不超过1000，数组元素的值不超过1000
第二行记录任务冷却时间，N为正整数，N<=100。

输出描述：输出为执行完所有任务所需的最短时间。

补充说明：

示例1
输入：2,2,2,3
2
输出：7
说明：时间1：执行类型2任务。
时间2：执行类型3的任务（因为冷却时间为2，所以时间2不能执行类型2的任务）。
时间3：系统等待（仍然在类型2的冷却时间）。
时间4：执行类型2任务。
时间5：系统等待。
时间6：系统等待。
时间7：执行类型2任务。
因此总共耗时7。

```
import java.util.Scanner;

import java.util.*;
// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String[] arr = in.nextLine().split(",");
        int n = Integer.parseInt(in.nextLine());

        Map<String,Integer> map = new HashMap<>();

        int dMax = 0;
        int dMaxN = 0;

        for(String i: arr){
            map.put(i,map.getOrDefault(i,0)+1);
            if(map.get(i) > dMax){
                dMax= map.get(i);
            }
        }

        for(int v: map.values()){
            if(v == dMax){
```

```
                    dMaxN++;
                }
            }

            int result = Math.max((dMax-1)*(n+1)+dMaxN,arr.length);
            System.out.println(result);
        }
    }
```