

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， $N \leq 100$ 。

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。

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
void code(int arr[]) {
```

```
    for (int i = 0; i < 1000; i++) {
```

```
        for (int j = i + 1; j <= 1000; j++) {
```

```
            if (arr[j] > arr[i]) {
```

```
                int tmp;
```

```
                tmp = arr[i];
```

```
                arr[i] = arr[j];
```

```
                arr[j] = tmp;
```

```
            }
```

```
        }
```

```
    }
```

```

}
int main () {
    int arr[1001] = {0};
    int n,count = 0;
    int ans = 0;
    string str;
    getline(cin, str);
    cin >> n;
    for (int i = 0; i< str.length(); i++) {
        int tmp = 0;
        while (i < str.length() && str[i]!=',') {
            tmp = tmp*10 + str[i] - '0';
            i++;
        }
        arr[tmp]++;
    }
    code(arr);
    while (arr[0] >0) {
        arr[0] --;
        ans ++;
        for (int i = 0; i<n; i++) {
            if (arr[0]==0 && arr[i+1]==0) break;
            if (arr[i+1] > 0) arr[i+1]--;
            ans ++;
        }
        code(arr);
    }
    cout << ans<< endl;
    return 0;
}

```