

UVA10474 : Where is the Marble?

題目資料

(UVA10474) Where is the Marble :

每顆大理石都有一整數編號(編號可重複)，請找出定編號的大理石在哪裡？最多 65 筆測資。每筆測資會給兩個整數 N 及 Q ，表示此測資有 N 顆大理石以及 Q 個問題；之後會給出 N 個整數(每顆大理石的編號)，再給出 Q 個整數。請先將大理石依編號由小到大排序，再找出 Q 個整數分別是第幾顆大理石？

所有輸入都是正整數且小於等於 10000，當 $N=0$ 且 $Q=0$ 時，測資結束。

每個測資都要輸出這是第幾筆測資，然後每個問題輸出一行結果，結果的格式有兩種，取決於是否有找到大理石，格式如下：

- 'x found at y'，第一顆編號為 x 的大理石出現在位置 y ，位置編號為 1, 2, 3, ... N 。
- 'x not found'，編號為 x 的大理石沒有出現。

標準輸入串

```
4 1
2
3
5
1
5
5 2
1
3
3
3
1
2
3
0 0
```

正確輸出串

輸出是第幾筆測資並回答問題。

CASE# 1:

5 found at 4

CASE# 2:

2 not found 3

found at 3

解答

思考	內容	C/C++
思考一	用 while 迴圈重複運算，只要 scanf 有讀到 2 個整數，且 n 不等於 0，就繼續運算。	<code>while(scanf("%d%d", &n, &q)==2 && n){}</code>
思考二	用 sort 將陣列由小到大排序。	<code>sort(stone, stone+n);</code>
思考三	使用反手拍計算處理了幾個問題。	<code>while(q--){ //找大理石所在位置 //輸出結果}</code>
思考四	用 lower_bound() 找大理石所在位置。	<code>int place = lower_bound(stone, stone+n, x)-stone;</code>
思考五	用 if 判斷是否有找到指定的大理石，並輸出結果。	<code>if(stone[place]==x){ printf("%d found at %d\n", x, place+1);} else{printf("%d not found\n", x);}</code>
思考六	把大理石陣列全部設為 0。	<code>for(int i=0; i<n; i++)stone[i]=0;</code>

程式碼

```
#include <iostream>
#include <cstdio>
#include <algorithm>
```

```
using namespace std;
```

```
const int maxn = 10000;
```

```
int main()
{
    int n, q, x, stone[maxn], kase = 0;
    while(scanf("%d%d", &n, &q)==2 && n){
        printf("CASE# %d:\n", ++kase);
        for(int i=0; i<n; i++)cin >> stone[i];
        sort(stone, stone+n);
        while(q--){
            cin >> x;
            int place = lower_bound(stone, stone+n, x)-stone;
            if(stone[place]==x)printf("%d found at %d\n", x, place+1);
        }
    }
}
```

```
        else printf("%d not found\n", x);
    }
    for(int i=0; i<n; i++)stone[i]=0;
}
return 0;
}
```