陣列

資料來源： http://www.tcgs.tc.edu.tw/~sagit/cpp/q3.htm

一、陣列是什麼

在討論陣列是什麼之前，我們先來看一個問題，假設我們要連續輸入 5 個人的成績，然後再從最後一個人的成績印回來，這個程式可以寫成：

```
int a1, a2, a3, a4, a5;

cin >> a1 >> a2 >> a3 >> a4 >> a5;

cout << a5 << " " << a4 << " " << a3   << " " << a2 << " " << a1 << endl;
```

這樣的程式好像沒什麼問題，但是如果我們要把它改成輸入 100 個人的成績，這時候可能就要宣告 100 個變數，於是程式會變得很複雜。如果有像數學裡的「數列」，例如 a_1、a_2、a_3、...、a_n 這樣的變數，問題是不是會變得比較簡單。而這樣的變數，就是我們要介紹的「陣列」。

陣列(Array)是一組相同型態的連續變數，它們使用同一個變數名稱，而另外用一個索引值來指定使用第幾個變數，我們可以把陣列想像成一排連續的儲存格，例如：

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	...	s[98]	s[99]
3	7	12	5	8	2	...	0	1

在 C/C++ 裡要宣告一個陣列，可以使用下面的語法：

變數型態 陣列名稱[元素個數];

例如：

```
int s[100];
```

即是宣告 s 是一個 int 的陣列，它有 100 個元素，分別是 s[0]、s[1]、s[2]、...、s[99]，特別注意的是它的索引值是從 0 開始到 99，而非 1 到 100。在程式中，索引值除了直接使用數字外，也可以使用變數或運算式，例如：s[i+2]。另外，在宣告陣列的時候也可以和其他非陣列的變數一起宣告，例如：

```
int a[100], b[200], c, d;
```

在宣告陣列的時候，我們也可以直接給它起始值，例如：

```
int a[5]={1, 2, 3, 4, 5};
```

如果 {} 裡的個數和陣列元素的個數相同，則可以省略陣列的元素個數，例如：

int a[]={1, 2, 3, 4, 5};

而如果起始值的個數比元素的個數少，則後面的元素會自動歸零。而要將一個陣列所有元素的起始值歸零，則可以寫成：

int a[100]={0};

接下來我們用陣列把上面的問題改寫：

int a[100], i;

for(i=0; i<100; i++) cin >> a[i];

for(i=99; i>=0; i--) cout << a[i] << " ";

cout << endl;

而如果我們要把陣列的大小改成 1000，可能要修改程式的許多地方，所以我們可以用 #define 指令宣告一個常數，之後凡是和陣列大小有關的數字都改用那個常數，就可以省去很多麻煩了，例如：

```
#include <iostream>
#define MAX 100
using namespace std;
int main()
{
   int a[MAX], i;
   for(i=0; i<MAX; i++) cin >> a[i];
   for(i=MAX-1; i>=0; i--) cout << a[i] << " ";
   cout << endl;
   return 0;
}
```

開頭的指令要自己一行，而且後面是不用加分號；的，並且要寫在 main() 函數的上面才行。上面的例子我們宣告了一個常數 MAX (自定的常數通常用大寫)，它的值為 100，然後所有和陣列大小有關的數字全部改用 MAX 或其運算式取代。之後如果要修改陣列的大小，就只需要修改 #define 那行的數字就行了，而不用從頭到尾檢查有哪些數字和陣列大小有關。

二、二維陣列與多維陣列

前一段我們提到的陣列，我們稱之為一維陣列，我們也可以把它想成是一個一排的儲存格，例如：

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]
13	54	4	34	6	72	15	22	1

有時候我們會需要宣告成有兩層的索引值，例如：

```
int s[3][8];
```

即宣告陣列 s 有 3x8=24 個元素，分別是 s[0][0]、s[0][1]、s[0][2]、…、s[2][6]、s[2][7]等。這種陣列我們稱之為二維陣列。 我們可以把它想成是一個表格，例如：

s[0][0]	s[0][1]	s[0][2]	s[0][3]	s[0][4]	s[0][4]	s[0][6]	s[0][7]
?	?	?	?	?	?	?	?
s[1][0]	s[1][1]	s[1][2]	s[1][3]	s[1][4]	s[1][4]	s[1][6]	s[1][7]
?	?	?	?	?	?	?	?
s[2][0]	s[2][1]	s[2][2]	s[2][3]	s[2][4]	s[2][4]	s[2][6]	s[2][7]
?	?	?	?	?	?	?	?

陣列的元素和一般的變數一樣，在宣告的時候它的值是一個亂數，如果要把它們全部歸零，則可以寫成：

```
int s[3][8]={0};
```

下面的程式是用兩層的迴圈輸入整個陣列的值：

```
#define MAX_X 3
#define MAX_Y 8

int i, j;
for(i=0; i<MAX_X; i++) {
   for(j=0; j<MAX_Y; j++) cin >> s[i][j];
}
```

而同樣的，如果我們宣告三層以上的索引值，則可以稱之為三維陣列或多維陣列，例如：

```
int x[10][10][10];
```

三、動態宣告陣列

有些時候，我們沒辦法預先知道需要宣告多大的陣列，而是等到使用者輸入完資料後才做決定，這時候可以使用動態宣告陣列的技巧。首先，先宣告一個指標：

```
int *a;
```

上面的變數 a 前面多一個星號 *，表示 a 是一個指標變數而不是一個正常的整數變數，接下來我們用下面的語法取得一段記憶體並讓它指向該位置：

```
a=(int *)malloc(sizeof(int)*n);   // C 的寫法

a=new int[n];                     // C++ 的寫法
```

在 C 的寫法中，malloc( ) 函數的功能是向系統取得一段記憶體，單位是 Bytes，我們使用 sizeof(int) 指令取得一個 int 的大小，再把它乘上所需要的個數即可。而 C++ 則是直接用 new 指令，後面接上 int[n] 代表要取得 n 個 int 整數的大小。以上的寫法可以把 int 換成其他的變數型別。

取得記憶體之後，就可以把它當成一般陣列來使用，例如：a[i]。不過，動態取得記憶體之後，當程式結束前，要手動將取得的記憶體釋放回去，否則會造成佔用記憶體的情況。而要釋放記憶體，則可以分別使用 free( ) 函數和 delete 指令，

寫法如下：

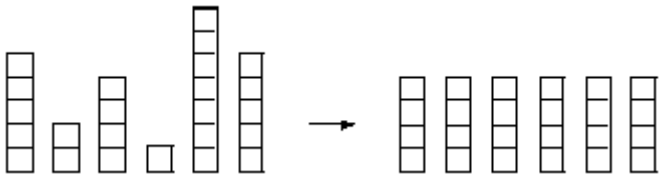
```
free(a)              // C 的寫法

delete [] a;         // C++ 的寫法
```

四、練習題

1. Q591：Box of Bricks
2. Q10370：Above Average (請參考格式化輸出)
3. Q484：The Department of Redundancy Department

591 - Box of Bricks *(Miguel A. Revilla 1998-03-10)*

Little Bob likes playing with his box of bricks. He puts the bricks one upon another and builds stacks of different height. ``Look, I've built a wall!'', he tells his older sister Alice. ``Nah, you should make all stacks the same height. Then you would have a real wall.'', she retorts. After a little con- sideration, Bob sees that she is right. So he sets out to rearrange the bricks, one by one, such that all stacks are the same height afterwards. But since Bob is lazy he wants to do this with the minimum number of bricks moved. Can you help?



Input

The input consists of several data sets. Each set begins with a line containing the number n of stacks Bob has built. The next line contains n numbers, the heights h_i of the n stacks. You may assume $1 \leq n \leq 50$ and $1 \leq h_i \leq 100$.

The total number of bricks will be divisible by the number of stacks. Thus, it is always possible to rearrange the bricks such that all stacks have the same height.

The input is terminated by a set starting with $n = 0$. This set should not be processed.

Output

For each set, first print the number of the set, as shown in the sample output. Then print the line ``The minimum number of moves is k.'', where k is the minimum number of bricks that have to be moved in order to make all the stacks the same height.

Output a blank line after each set.

Sample Input

```
6
5 2 4 1 7 5
0
```

Sample Output

```
Set #1
The minimum number of moves is 5.
```

MyNote:

10370 - Above Average

It is said that 90% of frosh expect to be above average in their class. You are to provide a reality check.

The first line of standard input contains an integer C, the number of test cases. C data sets follow. Each data set begins with an integer, N, the number of people in the class (1 <= N <= 1000). N integers follow, separated by spaces or newlines, each giving the final grade (an integer between 0 and 100) of a student in the class. For each case you are to output a line giving the percentage of students whose grade is above average, rounded to 3 decimal places.

Sample Input

5

5 50 50 70 80 100

7 100 95 90 80 70 60 50

3 70 90 80

3 70 90 81

9 100 99 98 97 96 95 94 93 91

Output for Sample Input

40.000%

57.143%

33.333%

66.667%

55.556%

MyNote:

484 - The Department of Redundancy Department

Write a program that will remove all duplicates from a sequence of integers and print the list of unique integers occuring in the input sequence, along with the number of occurrences of each.

Input

The input file will contain a sequence of integers (positive, negative, and/or zero). The input file may be arbitrarily long.

Output

The output for this program will be a sequence of ordered pairs, separated by newlines. The first element of the pair must be an integer from the input file. The second element must be the number of times that that particular integer appeared in the input file. The elements in each pair are to be separated by space characters. The integers are to appear in the order in which they were contained in the input file.

Sample Input

3 1 2 2 1 3 5 3 3 2

Sample Output

3 4
1 2
2 3
5 1

MyNote: