

資訊競賽選手培訓 – 首部曲 – Uva Online Judge

資料來源：<http://www.tcgs.tc.edu.tw/~sagit/cpp/q2.htm>

一、UVa Online Judge — [Uva On-Line Judge Home](#)

UVa Online Judge ([Uva On-Line Judge Home](#)，請使用 Firefox 開啟) 是個老牌且很有知名度的線上裁判系統 (Online-Judge)，提供相當多的題目供使用者練習，使用者可以註冊一個帳號，然後將自己寫好的程式碼上傳，系統會幫你測是程式是否正確。也因為 UVa Online Judge 這個網站與 ACM/ICPC 這個國際型的程式競賽有合作，並將它的題目加入題庫中，因此許多人便直接將 UVa Online Judge 題庫的題目稱為 ACM 題目(其實還有很多題目不是出自於 ACM/ICPC)。

要使用這個系統，首先我們先到左邊 [Login](#) 選單裡的 [Register](#) 註冊一個帳號，其中 Online Judge ID 這個欄位請填寫 00000，註冊完後請至註冊時填寫的信箱收取認證信，點選其中的連結，即可啟用這個帳號。

接下來可以在左邊 [Online Judge](#) 選單裡的 [Browse Problems](#) 瀏覽題目，題庫主要分成兩個部分，一部分是 [Problem Set Volumes](#) 題號從 100-999，另一部分是 [Contest Volumes](#)，題號從 10000-11586 (持續增加中)，兩個部分的題目都有難易不同的題目，我們可以看題目右邊的裁判次數以及被解出來的百分比，以瞭解該題的難易度。通常裁判次數越高而且被解出來百分比越高，代表題目越簡單，各位同學可以先挑簡單的來做，等到程式夠熟練了，再向比較難的題目挑戰。

如果題目看不太懂，可以到 [Lucky 貓的 ACM 園地](#)，站長整理了一些題目的中文翻譯，而且有題目的難易度區分。雖然沒有全部的題目，不過也有 500 多題，而且還會不斷增加，所以對初學者來說已經夠用了。等這些題目做完，如果還是意猶未盡，可以再向原文的題目挑戰。

寫完題目後，可以按該題右上角 [Submit](#) 按鈕，或是直接按左邊 [Online Judge](#) 選單的 [Quick Submit](#) 將你的程式碼上傳。要注意的是，如果你是用 Dev-C++，記得把 `system("PAUSE");` 這行拿掉或是在前面加上 `//` 讓它變成註解，不然程式會無法通過裁判。送出後可以到 [My Submissions](#) 看送出的結果，而在 [My Statistics](#) 裡可以看到解出了多少題。

因為 UVa Online Judge 的題目有時候會考慮到比較多的細節(也就是心機比較重)，所以如果題目上的範例資料都對了但是裁判結果卻是錯的，可以到 [UVa Online Judge 討論](#)

[區](#)，於右上角的文字方塊輸入題號後按 [Search](#)，就可以找到其他使用者的討論，也許會發現一些線索或是沒有注意到的地方。

二、連續輸入

在許多 Online Judge 系統以及一些程式比賽的題目中，經常需要連續輸入資料，一般說來有這三種：

情況一：給你一個數 N ，表示之後有連續 N 個值要輸入，例如：

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

我們可以用 `for` 迴圈來處理，例如：

```
int n, i, a[MAX];
cin >> n;
for(i=0; i<n; i++) {
    cin >> a[i];
}
```

情況二：連續輸入，直到輸入的數字為 0 時，例如：

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

我們可以用 `while` 迴圈來處理，例如：

```
int a, b;
while(1) {
    cin >> a >> b;
    if( (a==0) && (b==0) ) break;
    ...
}
```

上面的 `while(1)` 是無窮迴圈，所以我們要設定當 `a==0` 時讓它跳出。

情況三：連續輸入，直到沒有資料可以讀取為止，例如：

```
1 1
1 2
1 3
1 4
```

我們一樣用 `while` 迴圈來處理，在 C 的情況，可以利用 `scanf()` 函數的傳回值來判斷，例如：

```
int a, b;
while(1) {
    if( scanf("%d %d", &a, &b) < 2) break;
    ...
}
```

C++ 沒有類似的功能，但是我們可以用 `fail()` 函數來判斷輸入是不是出了問題，例如：

```
int a, b;
while(1) {
    cin >> a >> b;
    if( cin.fail() ) break;
    ...
}
```

上面的例子中，有時候也可以用 `eof()` 函數取代 `fail()` 函數。

最後還有一種情況是同一列有不定個數的資料要讀取，這個需要字串的處理，留到以後我們再說明。

三、練習題 (基本題，註冊完可以進行簡單測試)

1. Q10055：Hashmat the brave warrior (PS.要使用 `long long int`)
2. Q10071：Back to High School Physics
3. Q10499：The Land of Justice
4. Q10812：Beat the Spread!

Q10055 : Problem A

Hashmat the brave warrior

Input: standard input

Output: standard output

Hashmat is a brave warrior who with his group of young soldiers moves from one place to another to fight against his opponents. Before fighting he just calculates one thing, the difference between his soldier number and the opponent's soldier number. From this difference he decides whether to fight or not. Hashmat's soldier number is never greater than his opponent.

Input

The input contains two integer numbers in every line. These two numbers in each line denotes the number of soldiers in Hashmat's army and his opponent's army or vice versa. The input numbers are **not greater than 2^{32}** . Input is terminated by End of File.

Output

For each line of input, print the difference of number of soldiers between Hashmat's army and his opponent's army. Each output should be in separate line.

Sample Input:

```
10 12
10 14
100 200
```

Sample Output:

```
2
4
100
```

MyNote:

Q10071: Problem B

Back to High School Physics

Input: standard input
Output: standard output

A particle has initial velocity and constant acceleration. If its velocity after certain time is v then what will its displacement be in twice of that time?

Input

The input will contain two **integers** in each line. Each line makes one set of input. These two integers denote the value of v ($-100 \leq v \leq 100$) and t ($0 \leq t \leq 200$) (t means at the time the particle gains that velocity)

Output

For each line of input print a single integer in one line denoting the displacement in double of that time.

Sample Input

```
0 0
5 12
```

Sample Output

```
0
120
```

MyNote:

Q10499: Problem H

The Land of Justice

Input: standard input
Output: standard output
Time Limit: 4 seconds

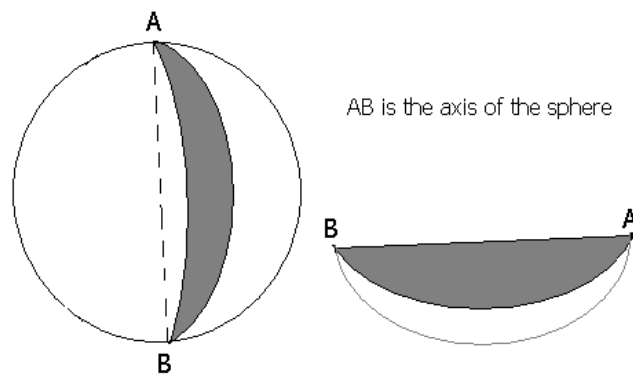
In the Land of Justice the selling price of everything is fixed all over the country. Nobody can buy a thing and sell it in double price. But, that created problems for the businessmen. They left their business and went to the production. So, after some days everybody was in production and nobody in business. And the people didn't get their necessary things though the country was self-sufficient in every sector.

The government became very much anxious. But, they were intelligent enough to call the mathematicians.

The mathematicians gave a solution. They suggested setting the surface area of an object as its selling-unit instead of its volume. Actually the clever mathematicians were very much interested to establish their own business.

Now, the government asks the programmers to build the software that would calculate the profit things.

Here your job is to calculate the business profit for a solid sphere. A businessman buys a complete sphere and to maximize his profit he divides it in n equal parts. All cut should go through the axis of the sphere. And every part should look like the picture below:



Input

You are given a sequence of integers N ($0 < N < 2^{31}$), indicating the numbers of parts of the sphere. The input file is **terminated with a negative number**. This number should not be processed.

Output

Calculate the profit over the sold pieces. The result should be in percentage and rounded to the nearest integer.

Sample input

2
2
-1

Sample output

50%
50%

Problemsetter: Tazrian Khan, University of Dhaka

MyNote:

Q10812: Problem D

Beat the Spread!



Superbowl Sunday is nearly here. In order to pass the time waiting for the half-time commercials and wardrobe malfunctions, the local hackers have organized a betting pool on the game. Members place their bets on the sum of the two final scores, or on the absolute difference between the two scores.

Given the winning numbers for each type of bet, can you deduce the final scores?

The first line of input contains n , the number of test cases. n lines follow, each representing a test case. Each test case gives s and d , non-negative integers representing the sum and (absolute) difference between the two final scores. For each test case, output a line giving the two final scores, largest first. If there are no such scores, output a line containing "impossible". Recall that football scores are always non-negative integers.

Sample Input

```
2
40 20
20 40
```

Output for Sample Input

```
30 10
Impossible
```

MyNote: