

2018 網際網路程式設計全國大賽

國中組模擬測試賽

- 本次比賽共 8 題，含本封面共 20 頁。
- 全部題目的輸入都來自**標準輸入**。輸入中可能包含多組輸入，以題目敘述為主。
- 全部題目的輸出皆輸出到螢幕 (**標準輸出**)。
輸出和裁判的答案必須完全一致，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 所有題目的時間限制請參考 Kattis 網頁上各題之標示。
- 比賽中上傳之程式碼，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 Time Limit Exceeded 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差小於 ϵ 皆視為正確」， ϵ 值以題目敘述為主。
舉例來說，假設 $\epsilon = 10^{-6}$ 且 a 是正確答案， b 是你的答案，如果符合 $\frac{|a-b|}{\max(|a|, |b|, 1)} \leq 10^{-6}$ ，就會被評測程式視為正確。

2018 網際網路程式設計全國大賽

輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main()
3 {
4     int cases;
5     scanf("%d", &cases);
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         scanf("%lld %lld", &a, &b);
10        printf("%lld\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main()
3 {
4     int cases;
5     std::cin >> cases;
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

A. 北極熊大遷徙

Problem ID: polarbear

因為全球暖化的關係，北極各處的浮冰正在慢慢融化之中。部份北極熊所在的浮冰已經融化到不堪居住的程度，於是這些北極熊興起遷徙的念頭。

已經融化到不堪居住的浮冰 A 上有 a 隻北極熊，牠們現在打算遷徙到有 b 隻北極熊居住的浮冰 B 。你要回答的是：經過北極熊大遷徙以後，浮冰 B 上總共會有多少隻北極熊。

Input

輸入只有一行，有兩個整數 a 和 b ，代表有 a 隻北極熊即將從浮冰 A 遷徙到原本有 b 隻北極熊的浮冰 B 。

- $1 \leq a, b < 2^{31}$

Output

輸出一行，表示浮冰 B 上最後會有多少隻北極熊。

Sample Input 1	Sample Output 1
24 47	71
Sample Input 2	Sample Output 2
33 20	53

This page is intentionally left blank.

B. 南極企鵝大遷徙

Problem ID: penguin

因為全球暖化的關係，南極各處的浮冰正在慢慢融化之中。部份企鵝居住地的浮冰已經大量融化，導致他們重要的食物來源「磷蝦」數量銳減，已到不堪居住的程度。已經融化到不堪居住的浮冰 A 上有 a 公斤的企鵝，牠們現在打算遷徙到有 b 公斤的企鵝居住的浮冰 B 。

你要回答的是：經過企鵝大遷徙以後，浮冰 B 上總共會有多少公斤的企鵝。

Input

輸入只有一行，有兩個浮點數 a 和 b ，代表有 a 公斤的企鵝即將從浮冰 A 遷徙到原本有 b 公斤重的企鵝的浮冰 B 。

- $0 \leq a, b \leq 50$
- a, b 的小數點後最多有五位。

Output

輸出一行，表示浮冰 B 上最後會有多少公斤的企鵝。

如果你的答案的絕對或相對誤差不超過 10^{-6} 都會被當作正確。

Sample Input 1	Sample Output 1
24.23 47.33	71.56
Sample Input 2	Sample Output 2
24.23000 47.33000	71.56000000000000
Sample Input 3	Sample Output 3
24.230 47.330	71.560001

This page is intentionally left blank.

C. 北極熊大遷徙研究

Problem ID: polarbear2

因為全球暖化的關係，北極各處的浮冰正在慢慢融化之中。部份北極熊所在的浮冰已經融化到不堪居住的程度，於是這些北極熊興起遷徙的念頭。

已經融化到不堪居住的浮冰 A 上有 a 隻北極熊，牠們曾經遷徙到有 b 隻北極熊居住的浮冰 B 。你是個學者，你正在研究北極熊的遷徙狀態。已知目前浮冰上已有 x 隻北極熊，你想知道在遷徙時有多少外來的北極熊 a 跟原生的北極熊 b 。

你要回答的是：經過北極熊大遷徙之前，浮冰 A, B 上可能分別會有多少隻北極熊，需要一個可能的答案，但你也知道以前的北極熊族群不會太大，不會超過 1000 隻。

Input

輸入只有一行，只有一個整數 x ，表示你要研究的浮冰上有 x 隻北極熊。

- $0 \leq x \leq 2000$

Output

輸出一行，有兩個整數 a, b 並以一個空白隔開，分別表示浮冰 A, B 原有可能會有多少隻北極熊。

- $x = a + b$
- $0 \leq a, b \leq 1000$

Sample Input 1

4

Sample Output 1

3 1

Sample Input 2

5

Sample Output 2

4 1

Sample Input 3

14

Sample Output 3

5 9

D. 猜數字

Problem ID: guess

我在心中想了一個介於 1 到 1000 的整數，你有辦法猜到這個數字是多少嗎？每當你猜了一個數字，我可以告訴你猜的過低、過高或正確。但你最多只能猜 10 次，所以你要好好選擇你猜的數字。

互動說明

當你的程式打算要猜數字時，輸出一行且包含一個整數，這個整數必須介於 1 到 1000 之間。當你猜完數字後，記得要清空 (flush) 標準輸出 (standard out)。

當我們收到你的猜測後，會把你猜的結果回覆到你的標準輸入 (standard in)。回覆會是下列三種：

- “lower” 如果我想的數字比你猜的數字小
- “higher” 如果我想的數字比你猜的數字大
- “correct” 如果你猜到了

當你猜到了正確數字後，你的程式必須立刻結束 (exit)。如果你 10 次都猜錯了，你的程式將會被強制中止。

以下是 C 程式 flush 的範例：

```
1 #include <stdio.h>
2 int main()
3 {
4     printf( "500\n" );
5     fflush( stdout );
6 }
```

以下是 C++ 程式 flush 的範例：

```
1 #include <iostream>
2 int main()
3 {
4     std::cout << "500\n";
5     std::cout << std::flush;
6 }
```

E. 線上遊戲

Problem ID: game

瀚瀚最近迷上了線上遊戲，一天二十四小時有十六小時都在玩遊戲，這真是太不應該了。

但最近瀚瀚找到了一個絕妙的練功地點，為了讓一天二十四小時被完善的利用，身為一個會寫程式的人，瀚瀚就開發了一個外掛程式。於是，在外掛程式的幫助下，瀚瀚就可以好好練功，並且不花太多時間在上面。

人算不如天算，正當瀚瀚以為人生美滿時，迎來了一個難題。本來瀚瀚可以很巧妙地在每一單位時間恰好擊殺一隻小怪物，但這個練功地點每 X 單位時間會重生出一隻大魔王艾迪，當大魔王艾迪出現時，瀚瀚必須先花費 Y 單位時間去擊殺他，否則瀚瀚的角色就會死去，很不方便。

以第一筆範例測試資料為例 ($X = 5, Y = 2$)，至第一隻艾迪大魔王重生出來前，瀚瀚可以擊殺五隻小怪物，接著瀚瀚必須花費兩單位時間擊殺艾迪大魔王；在下一隻艾迪大魔王重生出來前，瀚瀚可以擊殺三隻小怪物，接著瀚瀚必須花費兩單位時間擊殺艾迪大魔王；如此持續地進行下去。

身為一個專業玩家，瀚瀚想知道在時間點 T 時，瀚瀚已經擊殺了幾隻小怪物以及幾隻大魔王艾迪，你能夠幫助他計算嗎？

Input

測試資料恰有一行，包含三個整數 T, X, Y 。

- $1 \leq T \leq 10^3$
- $1 \leq Y \leq X \leq 10^3$

Output

輸出兩個以空白隔開的數字於一行，其中第一個數字為瀚瀚擊殺的小怪物數量，第二個數字為瀚瀚擊殺的艾迪大魔王數量。

Sample Input 1	Sample Output 1
12 5 2	8 2
Sample Input 2	Sample Output 2
11 5 2	8 1
Sample Input 3	Sample Output 3
10 5 2	8 1
Sample Input 4	Sample Output 4
9 5 2	7 1
Sample Input 5	Sample Output 5
8 5 2	6 1
Sample Input 6	Sample Output 6
7 5 2	5 1
Sample Input 7	Sample Output 7
6 5 2	5 0
Sample Input 8	Sample Output 8
5 5 2	5 0
Sample Input 9	Sample Output 9
4 5 2	4 0

F. 測試資料系統

Problem ID: shell

你知道網際網路程式設計全國大賽 (NPSC) 嗎？沒錯，就是你正在比的這個比賽！

每年，NPSC 的題目都是由若干位裁判分別負責出題而成的，其中，最重要也是最困難的環節就是把所有裁判產生出的測試資料整合起來。每位裁判都有自己產生測試資料的習慣以及偏好，如果沒有好好協調，肯定是無法讓每位裁判產生的測試資料好好的儲存起來。

因此，爲了讓之後 NPSC 出題的過程能更順暢，現在想請你寫一支測試資料的管理系統！

這個系統在一開始，必須提供一個檔案目錄供所有裁判使用 (/)，裁判們可以在這個檔案目錄底下，新增或刪除一個目錄、新增或刪除一個檔案，或是列出一個目錄所包含的資訊，嚴格來說，必須有以下幾個功能：

- `mkdir DIR`: 新增路徑爲 DIR 的目錄
- `rmdir DIR`: 刪除路徑爲 DIR 的目錄
- `ls DIR L`: 列出路徑爲 DIR 這個目錄下，最多 L 層的資訊。
- `touch FILE`: 新增 FILE 這個檔案
- `rm FILE`: 刪除 FILE 這個的檔案

其中，DIR 爲 /D1/D2/.../Dk 的形式；而 FILE 爲 /D1/D2/.../Dk/F 的形式。

1. 對於 `mkdir` 必須滿足在最一開始所提供的目錄 (/) 底下有 D1 這個目錄，且 /D1 這個目錄底下必須有 D2 這個目錄，以此類推。如果 Dk 已經存在、中途有任何一個目錄不存在、Dk 是一個檔案或是 DIR 恰爲 /，便回報錯誤；否則，便建立 Dk 這個目錄。
2. 對於 `rmdir` 必須滿足在最一開始所提供的目錄 (/) 底下有 D1 這個目錄，且 /D1 這個目錄底下必須有 D2 這個目錄，以此類推。如果 Dk 並不存在、中途有任何一個目錄不存在、Dk 是一個檔案、Dk 這個目錄底下還有存在其他目錄或檔案、或是 DIR 恰爲 /，便回報錯誤；否則，便刪除 Dk 這個目錄。
3. 對於 `ls` 必須滿足在最一開始所提供的目錄 (/) 底下有 D1 這個目錄，且 /D1 這個目錄底下必須有 D2 這個目錄，以此類推。如果 Dk 並不存在、中途有任何一個目錄不存在、或是 Dk 是一個檔案便回報錯誤；否則，請輸出 Dk 這個目錄往下最多 L 層的資訊。如果第 L 層當中有目錄，且該目錄底下還有其他目錄或檔案，請再目錄名稱後多輸出一個 '+'，請以字典序的順序依序輸出**該目錄下的目錄及檔案**，如果該目錄底下還有目錄或檔案，且還未達到 L 層，請先進入該目錄，並輸出其中的內容。舉例來說，如果 / 底下有一個

目錄 tmp，tmp 底下有兩個目錄分別為 tmpdir1 及 tmpdir2，tmpdir1 底下有一個檔案為 tmpfile：

指令	輸出
ls / 0 /+	
ls / 1 /	/tmp/
ls / 2 /	/tmp/ /tmp/tmpdir1/ /tmp/tmpdir2/
ls / 3 /	/tmp/ /tmp/tmpdir1/ /tmp/tmpdir1/tmpfile /tmp/tmpdir2/

- 對於 touch 必須滿足在最一開始所提供的目錄 (/) 底下有 D1 這個目錄，且 /D1 這個目錄底下必須有 D2 這個目錄，以此類推。如果 Dk 不存在、中途有任何一個目錄不存在、Dk 這個目錄底下已經存在 F、或是 FILE 恰為 /，便回報錯誤；否則，便建立 F 這個檔案。
- 對於 rm 必須滿足在最一開始所提供的目錄 (/) 底下有 D1 這個目錄，且 /D1 這個目錄底下必須有 D2 這個目錄，以此類推。如果 Dk 並不存在、中途有任何一個目錄不存在、F 是一個目錄、Dk 這個目錄底下不存在 F 這個檔案、或是 FILE 恰為 /，便回報錯誤；否則，便刪除 F 這個檔案。

Input

測試資料第一行包含一個整數 N ，代表總共輸入的指令數。接下來 N 行，分別包含以下其中一種形式：

- mkdir DIR: 新增路徑為 DIR 的目錄
- rmdir DIR: 刪除路徑為 DIR 的目錄
- ls DIR L: 列出路徑為 DIR 這個目錄下，最多 L 層的資訊。
- touch FILE: 新增 FILE 這個檔案
- rm FILE: 刪除 FILE 這個的檔案

- $1 \leq N \leq 500$
- $1 \leq |\text{DIR}|, |\text{FILE}| \leq 100$
- $1 \leq L \leq 100$
- DIR 滿足題目敘述中的格式
- FILE 滿足題目敘述中的格式
- D_i, F 僅包含大小寫英文字母數字 (a-zA-Z0-9) 及點 ('.'))
- $|D_i|, |F| \geq 1$

Output

對於每個指令如果產生錯誤，請輸出 "ERROR" 於一行（不包含引號）。對於每個正確執行的 `ls DIR L` 指令，輸出相對應的內容。

Sample Input 1	Sample Output 1
<pre>8 mkdir /tmp mkdir /tmp/tmpdir1 mkdir /tmp/tmpdir2 touch /tmp/tmpdir1/tmpfile ls / 0 ls / 1 ls / 2 ls / 3</pre>	<pre>/+ / /tmp/+ / /tmp/ /tmp/tmpdir1/+ /tmp/tmpdir2/ / /tmp/ /tmp/tmpdir1/ /tmp/tmpdir1/tmpfile /tmp/tmpdir2/</pre>

Sample Input 2	Sample Output 2
<pre> 18 touch /c mkdir / mkdir /a/b mkdir /a touch /a/b mkdir /a mkdir /c rmdir /c/d rmdir /c rmdir /a mkdir /a/e touch /a/e rm /a/e rm /a/b ls / 0 ls / 1 ls / 2 ls / 3 </pre>	<pre> ERROR ERROR ERROR ERROR ERROR ERROR ERROR ERROR ERROR /+ / /a/+ /c / /a/ /a/e/ /c / /a/ /a/e/ /c </pre>

Sample Input 3	Sample Output 3
<pre> 9 mkdir /npsc mkdir /npsc/shell touch /npsc/shell/s1.in touch /npsc/shell/s2.in touch /npsc/shell/s3.in ls /npsc 0 ls /npsc/shell 0 ls /npsc/shell/s1.in 0 ls /npsc/shell 100 </pre>	<pre> /npsc/+ /npsc/shell/+ ERROR /npsc/shell/ /npsc/shell/s1.in /npsc/shell/s2.in /npsc/shell/s3.in </pre>

G. 最大不連續和問題

Problem ID: sum

在數學中，某個數列的子數列是從最初數列通過去除某些元素，但不破壞餘下元素的相對位置（在前或在後）而形成的新數列。例如， $1, 3, 4$ 即是 $1, 2, 3, 4, 5$ 的一個子數列。

最大連續和問題是一個子數列相關的經典問題，目標是在數列中找到一個連續的子數列，使該子數列的和最大。例如，對一個數列 $-2, 1, -3, 4, -1, 2, 1, -5, 4$ ，其連續子數列中和最大的是 $4, -1, 2, 1$ 其和為 6。

在某個小島舉行的 ACM-ICPC 區域賽中，常常會有許多經典問題。身為一個專業的競賽選手，艾迪十分熟練各種經典題的做法，因此也常常在該賽區神速般獲得各種 **Accepted**。

但就在這次競賽中，艾迪稍微遇到了小波折。這次艾迪遇到的問題已經不是經典題了，而是只有一字之差的**最大不連續和問題**！即要在數列中找到一個**不連續的子數列**，使得該子數列的總和最大。

例如，對一個數列 $1, 1, 1, -1, -3, 10$ ，其不連續的子數列中和最大的是 $1, 1, 1, 10$ 其和為 13。

由於艾迪已經驚慌失措了，你能夠幫助他解決這個問題嗎？

Input

測試資料第一行有一個數字 N ，表示數列的長度。

第二行有 N 個整數 A_1, A_2, \dots, A_N ，表示數列。

- $3 \leq N \leq 10^6$
- $|A_i| \leq 10^9$

Output

輸出一個數字於一行，表示最大不連續和的值。

Sample Input 1	Sample Output 1
3 -1 -2 -3	-4

Sample Input 2	Sample Output 2
6 1 1 1 -1 -3 10	13

Sample Input 3	Sample Output 3
10 9 4 3 8 1 5 10 7 2 6	54

H. When in doubt, C4!

Problem ID: bomb

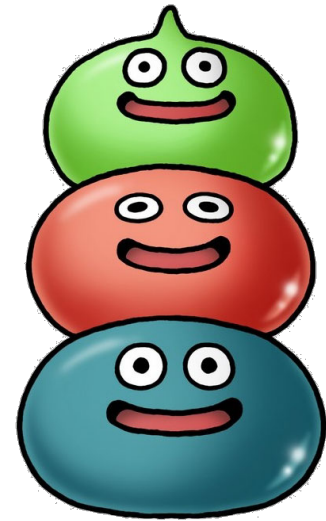
「不管遇上什麼問題，一顆炸彈總能解決你的困擾——如果不行就兩顆。解決不了問題那是你當量不夠，這是普世真理。」

當今的帝國審查官兼諸界巡視者暨煉獄星領主——郝仁，在去找女神渡鴉 12345 蹭飯的路上遇到一群不長眼的史萊姆擋路，江湖人稱炸彈超仁的他決定扔炸彈開路。

擋路的史萊姆們巧妙地排成了一個 $n \times m$ 的格狀陣型。其中郝仁視角中左上角那格的座標為 $(1, 1)$ ，右下角為 (n, m) 。每格中可能有若干隻史萊姆直直向上疊起來，也有可能一格內所有史萊姆都被嚇跑了，導致該格一隻史萊姆也沒有。

郝仁打算依序扔 q 顆炸彈來趕走史萊姆。第 i 顆炸彈會落於座標 (x_i, y_i) 的格子正中央，其爆炸半徑為 r_i 。而當第 i 顆炸彈落下後，所有座標 (x, y) 滿足 $(x - x_i)^2 + (y - y_i)^2 \leq r_i^2$ 的非空格子中，最上面的那隻史萊姆會被炸彈嚇跑。

身為一個專業的炸彈仁，忠實紀錄每顆炸彈的威力是他的職責所在。對於每顆炸彈，郝仁想要知道它嚇跑了幾隻史萊姆。但是史萊姆實在太多了，希望你可以寫個程式幫幫他。



Input

測試資料第一行有三個整數 n, m, q ，分別代表史萊姆陣型的長與寬，以及炸彈的數量。接下來 n 行，每行有 m 個字元，依序代表左上到右下每格包含的史萊姆數量。為了節省空間，郝人偷懶的使用了縮寫來記錄史萊姆數量。其中字元 0 到 9 代表 0 到 9 隻史萊姆、字元 a 到 z 代表 10 到 35 隻史萊姆、字元 A 到 Z 則代表 36 到 61 隻史萊姆。接下來 q 行，每行有三個數字 x_i, y_i, r_i ，依序代表每顆炸彈的落點座標以及爆炸半徑。

- $1 \leq n, m \leq 1000$
- $1 \leq q \leq 100000$
- $1 \leq x_i \leq n$
- $1 \leq y_i \leq m$
- $1 \leq r_i \leq 5000$
- 每格中的史萊姆數量不超過 61 隻

Output

請輸出 q 行，每行包含一個整數。第 i 行中的數字表示第 i 顆炸彈嚇跑了幾隻史萊姆。

Sample Input 1

```
2 3 3
240
222
1 1 2
2 3 2
1 2 3
```

Sample Output 1

```
4
4
3
```

Sample Input 2

```
2 4 10
np5c
6es7
1 3 0
2 1 1
1 3 1
2 4 2
1 1 4
2 3 3
2 3 1
1 4 0
1 3 0
1 4 4
```

Sample Output 2

```
1
3
4
5
8
8
3
1
0
7
```