2026 臺灣大學程式解題社程式解題競賽
測試賽

- 本次比賽共 8 題，含本封面共 24 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。
- 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。

Problem	Problem Name	Time Limit	Memory Limit
A	北極熊大遷徙	1 s	2048 MB
B	南極企鵝大遷徙	1 s	2048 MB
C	猜數字	1 s	2048 MB
D	北極熊大遷徙研究	1 s	2048 MB
E	雞蛋糕	1 s	2048 MB
F	旅行	1 s	2048 MB
G	猜三個數字	1 s	2048 MB
H	巴乙己放石頭問題	1 s	2048 MB

2026 臺灣大學程式解題社程式解題競賽
輸入輸出範例

C 程式範例：

```c
#include <stdio.h>
int main()
{
    int cases;
    scanf("%d", &cases);
    for (int i = 0; i < cases; ++i)
    {
        long long a, b;
        scanf("%lld %lld", &a, &b);
        printf("%lld\n", a + b);
    }
    return 0;
}
```

C++ 程式範例：

```cpp
#include <iostream>
int main()
{
    int cases;
    std::cin >> cases;
    for (int i = 0; i < cases; ++i)
    {
        long long a, b;
        std::cin >> a >> b;
        std::cout << a + b << std::endl;
    }
    return 0;
}
```

A. 北極熊大遷徙

Problem ID: add

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: floatadd

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	71.5600000000

Sample Input 2	Sample Output 2
24.23000 47.33000	71.5600000000

Sample Input 3	Sample Output 3
24.230 47.330	71.5600000000

This page is intentionally left blank.

C. 猜數字

Problem ID: guess

本題為互動題。

我在心中想了一個介於 1 到 1024 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 11 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 1024 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：

- “<” 如果我想的數字比你猜的數字小
- “>” 如果我想的數字比你猜的數字大
- “=” 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 11 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：

```c
#include <stdio.h>
int main() {
  printf("500\n");
  fflush(stdout);
}
```

以下是 C++ 程式 flush 的範例：

```cpp
#include <iostream>
int main() {
  std::cout << "500\n";
  std::cout << std::flush;
}
```

以下是 Python 程式 flush 的範例：

```python
import sys
print(500)
sys.stdout.flush()
```

以下是 Java 程式 flush 的範例：

```java
System.out.println(500);
System.out.flush();
```

以下是 Kotlin 程式 flush 的範例：

```kotlin
fun main() {
  println(500);
  System.out.flush();
}
```

以下是 Rust 程式 flush 的範例：

```rust
use std::io::{self, Write};
fn main() {
  println!("500");
  let _ = io::stdout().flush();
}
```

Sample Input	Sample Output
> = <	123 789 456

This page is intentionally left blank.

D. 北極熊大遷徙研究

Problem ID: revadd

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 原本可能會有多少隻北極熊。如果有多種滿足下方條件的答案，輸出任何一種皆會被視為正確。

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

Sample Input 1	Sample Output 1
4	3 1

Sample Input 2	Sample Output 2
5	4 1

Sample Input 3	Sample Output 3
14	5 9

Note

在 Sample Input 1 中，輸出 0 4 也會被視為正確。

E. 雞蛋糕

Problem ID: cake

小冰塊現在正在雞蛋糕的烤爐上，要是再不離開，他就要融化成一灘水了！

烤爐有 N 個橫列，每列從左至右有 N 塊雞蛋糕，現在小冰塊正位於烤爐第一列第一個雞蛋糕的邊緣，他需要通過這些雞蛋糕，從最後一列最後一個雞蛋糕向右離開。

這些雞蛋糕各自有一面是冷的，而另一面則是熱的，要是小冰塊走到熱的那一面上，他便會快速融化、蒸發。因此他在移動的途中只能站在雞蛋糕冷的那一面上。首先，他會站上第一列第一個雞蛋糕，之後的每一步可以往上下左右移動，但是除了從最後一列的最後一個雞蛋糕離開烤爐以外，他都不能往烤爐的邊界以外移動。

目前烤爐上的狀態可能無法讓他通行，幸運的是接下來雞蛋糕店的老闆會不斷的翻動這些雞蛋糕，每次的操作可能會是將一整列的雞蛋糕都翻面，或是將一整行的雞蛋糕都翻面。

現在小冰塊想要知道，有沒有可能經過數次（可能是沒有）翻面後，雞蛋糕上會出現一條可以讓它逃離的路徑？由於冰塊在室溫下也會漸漸融化，他更不可能會冒著被翻下去的風險，所以他不能在雞蛋糕上等待翻面，也就是說，要等待雞蛋糕店的老闆翻完雞蛋糕才能踩上這個雞蛋糕網格。

Input

輸入第一行有一個整數 N，接下來有 N 行，每行有一個長度為 N 的字串，第 i 行的第 j 個字元代表第 i 列第 j 個雞蛋糕的一開始狀態：如果是 `H` 代表熱的面朝上，反之是 `.` 則表示冷的面朝上。

- $1 \leq N \leq 10$

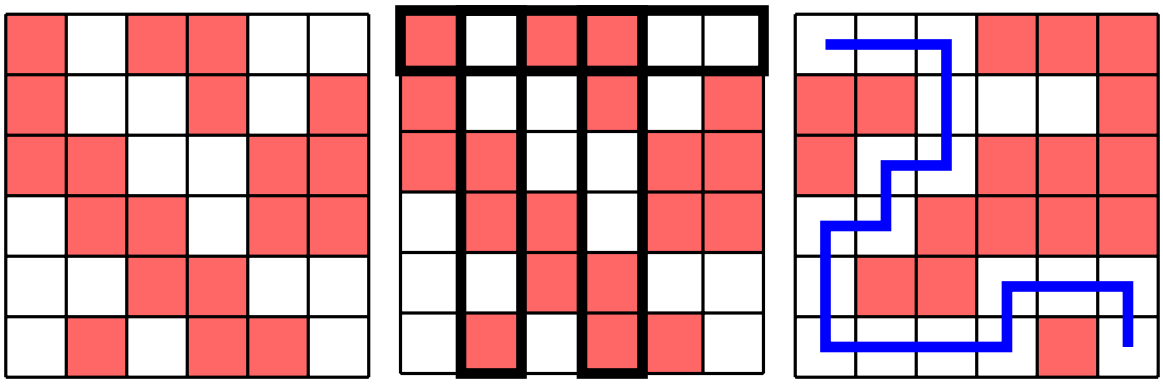
Output

如果小冰塊有可能逃離烤爐，輸出一行 `Yes`，否則輸出 `No`。

Sample Input 1	Sample Output 1
6 H.HH.. H..H.H HH..HH .HH.HH ..HH.. .H.HH.	Yes

Note

在範例當中，可以透過以下的方式翻轉達成條件。



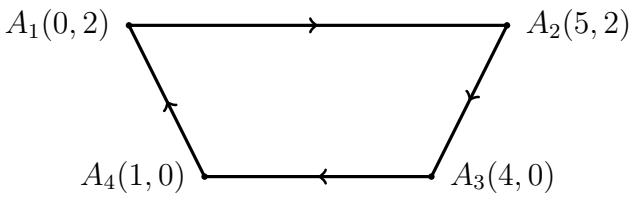
F. 旅行

Problem ID: travel

經過了一個學期的努力終於到假期了！小桃決定用一趟旅行來慶祝假期的開始。今天小桃拿出了一張地圖，並在上面標上一些她想要去的景點。地圖為一個二維平面，而每個景點都會是二維平面上的一個整數點。已知小桃一共想要去 N 個景點，且第 i 個景點在座標 (x_i, y_i) 上，任兩個景點的座標都不相同。

小桃有著獨特的方法來規劃她的旅行路線，首先她會先選擇 M 個相異的整數點 $A_1, A_2, \ldots, A_M$，接著她會從 A_1 出發，以最短距離移動到 A_2，再以最短距離移動到 A_3，以此類推。而當移動到 A_M 後再以最短距離回到 A_1。

舉例來說，若 $M = 4, A_1 = (0, 2), A_2 = (5, 2), A_3 = (4, 0), A_4 = (1, 0)$，那她移動的路線會是：



小桃希望旅行路線能夠經過所有的景點，同時她不希望旅行的路線經過重複的點，意即除了起終點相同外，路線不會相交。此外，小桃希望挑選的點數量至少為 3 且至多為 2000。現在給你全部的景點，你能幫幫小桃找到一條符合所有條件的路線嗎？

Input

輸入第一行有一個正整數 N，表示小桃想要去的景點數量接下來 N 行，第 i 行有兩個用空格隔開的整數 x_i, y_i，代表第 i 個景點的 x 座標與 y 座標。

- $3 \leq N \leq 1000$
- $-10^6 \leq x_i, y_i \leq 10^6$
- $\forall i \neq j, (x_i, y_i) \neq (x_j, y_j)$

Output

第一行請輸出一個正整數 M，表示你要挑選的整數點。

接下來 M 行，第 i 行請輸出兩個用空格隔開的整數 a_i, b_i，代表第 i 個挑選的點的 x 座標與 y 座標。

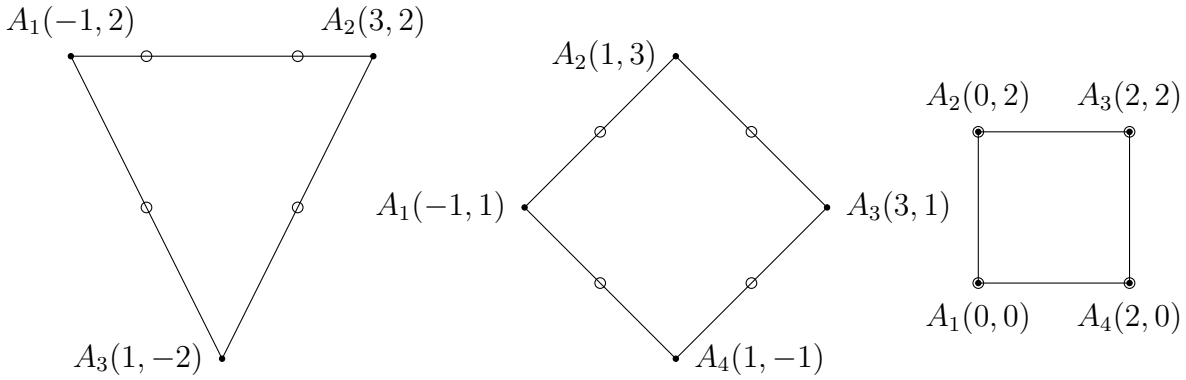
你的答案會被視為正確若你符合以下條件：

- $3 \leq M \leq 2000$
- $-10^9 \leq a_i, b_i \leq 10^9$
- $\forall i \neq j, (a_i, b_i) \neq (a_j, b_j)$
- 旅行的路線會經過所有景點
- 旅行的路線不會經過重複的點

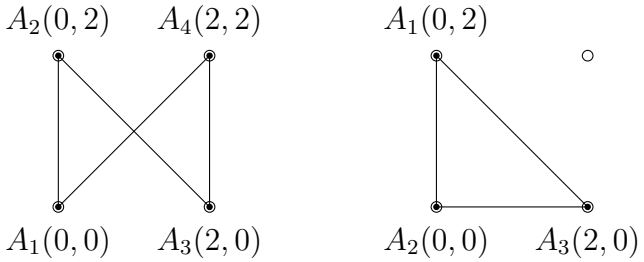
Sample Input 1	Sample Output 1
4 0 0 0 2 2 0 2 2	3 -1 2 3 2 1 -2

Note

範例測試資料一中，以下為一些合法的旅行路線，這裡空心的圓點代表景點。



以下為一些不合法的旅行路線，左邊的因為路線經過了點 $(1,1)$ 兩次，右邊的因為她沒有經過點 $(2,2)$。



你可以運用以下方格紙來幫助你思考：

This page is intentionally left blank.

G. 猜三個數字

Problem ID: 3black

本題為互動題。

我在心中想了三個不同的整數，他們都介於 1 到 200，你有辦法猜到這三個數字是多少嗎？每當你猜了一組數字，我可以告訴你我想的數字有奇/偶數個在你猜的數字裡（注意到 0 是偶數）。但你最多只能猜 33 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 K，並在下一行輸出 K 個兩兩相異的整數，這些整數必須介於 1 到 200 之間（包含 $1, 200$）。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：

- 『odd』如果你猜的 K 個整數中有奇數個數字和我心中想的數字一樣
- 『even』如果你猜的 K 個整數中有偶數個數字和我心中想的數字一樣
- 『correct』如果 $K = 3$ 而且你猜的數字都和我心中想的數字一樣

當你猜到了正確數字後，你的程式必須立刻結束（exit）。我會回答至多你 33 次，如果這 33 次都沒有回覆你 correct，你的程式將會被強制中止。注意到我在過程中可能會改變心意，換一組數字，不過我保證就算換了一組數字，之前的回答依舊符合新的答案。

以下是 C 程式 flush 的範例：

```c
#include <stdio.h>
int main() {
  printf("500\n");
  fflush(stdout);
}
```

以下是 C++ 程式 flush 的範例：

```cpp
#include <iostream>
int main() {
  std::cout << "500\n";
  std::cout << std::flush;
}
```

以下是 Python 程式 flush 的範例：

```python
import sys
print(500)
sys.stdout.flush()
```

以下是 Java 程式 flush 的範例：

```java
System.out.println(500);
System.out.flush();
```

以下是 Kotlin 程式 flush 的範例：

```kotlin
fun main() {
  println(500);
  System.out.flush();
}
```

以下是 Rust 程式 flush 的範例：

```rust
use std::io::{self, Write};
fn main() {
  println!("500");
  let _ = io::stdout().flush();
}
```

Sample Input	Sample Output
odd even correct	5 1 2 3 4 5 4 1 2 4 5 3 2 4 3

This page is intentionally left blank.

H. 巴乙己放石頭問題

Problem ID: stone

巴乙己最喜歡收集石頭了，在眾多石頭中，他最喜歡的就是大石頭。今天，他來到全國最知名的珍貴石頭大會（National Precious Stone Conference，簡稱 NPSC），希望能帶走一些有趣的大石頭。

他準備了 N 個容量相同但長相不同的籃子，每個籃子最多能裝兩顆石頭，而且因為籃子寬度很窄，先放入的石頭會在籃子底部，後放入的石頭則會疊在先放入的石頭之上。巴乙己在前往 NPSC 的路上，已經手癢收集了若干顆大石頭並放入其中的一些籃子裡了。

巴乙己到了 NPSC 之後，發現大會準備了 N 種奇特的石頭，每個人每種石頭可以拿走兩顆，但是必須要依序拿第一種到第 N 種石頭。巴乙己有強迫症，他希望同一個籃子內裝的石頭不是同一種，而且一種石頭要拿就要兩顆都拿。到 NPSC 的路上所收集的石頭種類都不相同，而且跟大會準備的石頭種類都不一樣。他知道以他現在籃子的容量，不一定能裝得下 NPSC 送的所有石頭，但他也不想多思考如何好好最大化他能拿的石頭個數，因此他的策略如下：假設他正要拿第 i 種石頭，而且現在還有至少兩個籃子還沒裝滿，則他會選擇其中兩個籃子並將兩顆第 i 種石頭分別放入所選的兩個籃子。如果他找不到兩個還沒裝滿的籃子，或是 N 種石頭都拿完了，他就不會再拿任何石頭。

已知巴乙己到 NPSC 前，第 i 個籃子內已經裝了 a_i 個石頭，請問當他已經拿到不會再拿任何石頭時，籃子的狀態有幾種可能？因為狀態數可能很大，所以請回答這個數量除以 $10^9 + 7$ 的餘數。

我們稱兩個籃子的狀態不同，若存在某一個籃子由頂部到底部所放的石頭種類不同，或是放的石頭個數不同。

Input

第一行輸入一個正整數 N，意義如題目所述。

第二行輸入 N 個整數 $a_1, a_2, \ldots, a_N$，其中 a_i 代表到 NPSC 之前，第 i 個籃子裡已經放了 a_i 顆石頭。

- $1 \le N \le 10^6$
- $0 \le a_i \le 2$

Output

輸出一個整數，代表最後籃子可能的狀態數除以 $10^9 + 7$ 的餘數。

Sample Input 1	Sample Output 1
4 0 1 0 1	19

Sample Input 2	Sample Output 2
5 0 0 0 0 0	2490