

# 2021 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 18 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「相對或絕對誤差不超過  $\epsilon$  皆視為正確」， $\epsilon$  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	1024 MB
B	北極熊大遷徙研究	1 s	1024 MB
C	猜數字	1 s	1024 MB
D	南極企鵝大遷徙	1 s	1024 MB
E	貓咪排隊買早餐	2 s	1024 MB
F	翻轉隊伍	1 s	1024 MB
G	城市分類	2 s	1024 MB
H	旅遊	8 s	1024 MB

# 2021 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

24 47
-------

#### Sample Output 1

71
----

#### Sample Input 2

33 20
-------

#### Sample Output 2

53
----

*This page is intentionally left blank.*

## B. 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	Sample Output 1
4	3 1

**Sample Input 2**

5
---

**Sample Output 2**

4 1
-----

**Sample Input 3**

14
----

**Sample Output 3**

5 9
-----

## C. 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 }
```



## D. 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.*

## E. 貓咪排隊買早餐

Problem ID: cats

最近喵喵來到了貓咪國旅行，這裡充滿各種品種的可愛貓咪，如 bb 貓、加菲貓、電貓、小肥貓……等，只要你能想到的，這邊都能夠看到。

在這邊待了幾天的喵喵，發現貓咪國有一家非常熱門的早餐店——NPSC。每天早上，都會有數以萬計的貓咪在店門口排隊，想要吃 NPSC 的早餐。

看到這麼多貓咪在吃 NPSC 的早餐，難道喵喵不會想吃吃看嗎？

喵喵還真的不想吃！比起吃 NPSC 的早餐，喵喵更喜歡觀察那些排在隊伍中的可愛貓咪們！

現在，喵喵發現有  $N$  隻貓咪正在排隊，從左到右的品種分別為  $a_1, a_2, \dots, a_N$ 。為了方便，喵喵告訴你的品種都已經換成正整數來表示了，也就是說  $a_1, a_2, \dots, a_N$  皆為不超過  $10^6$  的正整數。

此外，喵喵還發現，如果品種相同的貓咪排在相鄰的位置，那他們便會開始玩耍起來，進而感受不到因漫長排隊所產生的無聊感，這也是喵喵最喜歡看到的現象。

充滿好奇心的喵喵，看著這  $N$  隻正在排隊的貓咪，不禁心想，如果他能將至多 1 隻貓咪的品種改變成任何品種，喵喵可以看到「最長連續相同品種的貓咪數量」最大可以是多少呢？

也許你會問，喵喵要如何改變貓咪的品種呢？這個問題簡單！喵喵可以用染毛劑來幫貓咪換顏色，貓咪們便可以假裝改變品種囉！由於喵喵手邊只有 1 個染毛劑，因此至多只能幫 1 隻貓咪改變品種。

其次，你可能會好奇，「連續相同品種的貓咪數量」的意思是什麼？舉個例子，若存在一個區間  $[l, r]$ ，滿足  $a_l = a_{l+1} = \dots = a_r$ ，那麼  $r - l + 1$  便是「連續相同品種的貓咪數量」。

### Input

輸入第一行有一個非負整數  $N$ ，代表隊伍中貓咪的數量。

輸入第二行有  $N$  個正整數  $a_1, a_2, \dots, a_N$ ，分別代表從左至右每隻貓咪的品種。

- $1 \leq N \leq 10^6$
- $1 \leq a_i \leq 10^6$

## Output

輸出一個整數於一行，代表在喵喵可以改變至多 1 隻貓咪品種的情況下，喵喵可以看到「最長連續相同品種的貓咪數量」最大可以是多少。

### Sample Input 1

3 1 2 1	Sample Output 1 3
------------	----------------------

### Sample Input 2

5 1 2 1 2 1	Sample Output 2 3
----------------	----------------------

## F. 翻轉隊伍

Problem ID: reverse

NPSC 國中正在校外教學，其中你恰好是一個有  $N$  位學生的隊伍領隊，學生們的座號分別是  $1 \sim N$ ，行動時他們總是照著座號排成一列。

麻煩的是，這些學生非常喜歡聊天，如果隊伍太吵而被扣秩序成績那就不好了！

經過調查後，你發現這座號  $i$  的學生有興趣的話題為  $a_i$ ，有趣的是，這些學生只會跟和自己**相鄰且興趣相同**的人聊天，這就代表只需要盡量讓相鄰且興趣不同的對數越多，整支隊伍就會越安靜！

不過調整隊伍是一個很麻煩的任務，基於一些問題，你只能從中選出一段連續的學生，並將他們頭尾翻轉來組合出新的隊伍，而且由於時間緊迫，你只能執行至多  $K$  次這個操作。

舉例來說，假設現在一支 5 個人的隊伍，他們有興趣的話題編號依序是 1 1 1 2 2，你可以把位置區間  $[2, 4]$  的學生翻轉使得有興趣的話題編號變成依序是 1 2 1 1 2，然後再翻轉  $[4, 5]$  的學生變成 1 2 1 2 1，如此一來這支隊伍就有 4 對相鄰且興趣不同的人，而且這也是 2 次操作可以達到的最佳對數。

請你撰寫一支程式，計算經過至多  $K$  次翻轉後，隊伍最多可以有多少對相鄰且興趣相異的學生。

### Input

輸入的第一行有兩個正整數  $N, K$ ，代表隊伍的人數、你可以執行的操作數至多有  $K$  次。

第二行有  $N$  個以空格分開的整數  $a_1, a_2, \dots, a_N$ ，代表第  $i$  個人有興趣的話題為  $a_i$ 。

- $1 \leq N \leq 5 \times 10^5$
- $0 \leq K \leq N$
- $1 \leq a_i \leq N$

## Output

輸出只有一行一個數字，代表經過至多  $K$  次翻轉後，隊伍最多可以有多少對相鄰且興趣相異的學生。

### Sample Input 1

5 2 1 1 1 2 2	Sample Output 1 4
------------------	----------------------

### Sample Input 2

7 1 2 2 3 3 1 1 4	Sample Output 2 5
----------------------	----------------------

### Sample Input 3

7 2 2 2 3 3 1 1 4	Sample Output 3 6
----------------------	----------------------

## G. 城市分類

Problem ID: cities

NPSC 王國有  $N$  個城市以及  $M$  條道路，城市編號為 1 到  $N$ ，而一條道路連接兩個相異的城市，使得任兩個城市皆可以透過若干條道路抵達彼此。為了交通的穩定性，NPSC 王國的道路設計還有另一個特點：對於任一條道路（假設它連接城市  $x$  與  $y$ ），則如果從  $x$  開始經過這條道路抵達  $y$ ，那麼至多只有一種方法可以再從  $y$  走回  $x$  而途中不經過重複的城市與道路（包含  $(x, y)$  這條道路）。

雖然這種道路的設計大量的減少了人們在同一些城市之間不斷繞來繞去的現象，由於 NPSC 國沒有設置良好的路標，因此還是時常發生有人迷路的事件。身為交通部部長的你，決定將所有城市分成若干類，使得每一條道路都連接兩個不同類的城市。如此一來，人們就能比較方便的確認自己行駛的方向。

當然，如果城市被分成太多類的話也會造成反效果，使得交通系統變得太複雜。因此，你希望將城市在分成最少類的同時，也達到上述的效果。

### Input

輸入第一行有兩個非負整數  $N$  和  $M$ ，分別代表城市以及道路的數量。接著  $M$  行，第  $i$  行有兩個正整數  $u_i, v_i$ ，代表第  $i$  條道路連接編號  $u_i$  以及編號  $v_i$  的城市。

- $1 \leq N \leq 5 \times 10^5$
- $1 \leq u_i, v_i \leq N$
- 保證  $u_i \neq v_i$ ，且任兩個城市之間至多只有一條道路

### Output

輸出兩行，第一行有一個正整數  $K$ ，代表最少所需的城市類別。接著一行輸出  $N$  個由空白隔開且位於 1 到  $K$  之間的正整數，第  $i$  個代表編號為  $i$  的城市被歸為哪一類。如果有多種符合條件且類別數最少的分類方法，只要輸出任一種分類方法即可。

**Sample Input 1**

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

**Sample Output 1**

```
2
1 2 1 2
```

**Sample Input 2**

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

**Sample Output 2**

```
2
1 2 1 2
```

**Sample Input 3**

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

**Sample Output 3**

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



## H. 旅遊

Problem ID: travel

小 N 是一個喜好旅遊的人。他最近發現 NPSC 島十分適合旅遊，因此他決定規劃一個在 NPSC 島旅遊的行程。

經過詳細的調查之後，發現 NPSC 島總共有  $N$  個主要的城鎮，由 0 編號到  $N - 1$ ，並且每個城鎮都有至少一條通往其它城鎮的單向道路，有些城鎮甚至會有超過一條通往同一個城鎮的道路。為了得到最佳的旅遊體驗，小 N 把從每個城鎮出發的所有道路都排好順序。小 N 計畫了  $T$  天的行程，每天遊覽一個城鎮，並且要按照自己排的道路順序選擇隔天的要前往的城鎮。

具體來說，小 N 在行程開始前會先用八個幸運數字  $A, B, C, D, E, F, a_0, a_1$  產生一個數列  $a_0, a_1, \dots, a_{T-1}$ ，其中對於所有  $i \geq 2$ ， $a_i = Aa_{i-1}^2 + Ba_{i-1}a_{i-2} + Ca_{i-2}^2 + Da_{i-1} + Ea_{i-2} + F$ 。接著，假設小 N 第  $i$  天在編號  $x$  的城鎮，且在小 N 的排序之後由該城鎮出發的各個道路分別通往編號  $p_{x,0}, p_{x,1}, \dots, p_{x,m_x-1}$  的城鎮，那麼他隔天就會前往編號  $p_{x,q}$  的城鎮，其中  $q$  是  $a_i$  除以  $m_x$  的餘數。

然而因為小 N 的行程實在是太長了，而且他的數列中數字也都非常大，所以請你幫小 N 寫一個程式，方便他用自己排的道路順序和第 1 天所在的城鎮計算出這  $T$  天分別會在哪個城鎮。

### Input

第一行有三個以空白隔開的非負整數  $N, S, T$ ，代表島上的城鎮數量、小 N 第 1 天所在的城鎮編號與他總共要移動的次數。

第二行有八個以空白隔開的非負整數  $A, B, C, D, E, F, a_0, a_1$ ，意義如題目所述。

接下來有  $N$  行，每行有數個以空白隔開的非負整數，第  $i$  行（從 0 開始編號）代表從第  $i$  個城鎮出發的所有道路。第一個數字  $m_i$  代表從該城鎮出發的道路總數，接下來  $m_i$  個數字依序代表每條道路連向哪一個城鎮，道路的順序即是按照小 N 規劃時決定的順序。

- $0 \leq S < N \leq 3 \times 10^4$
- $T \leq 5 \times 10^6$
- $1 \leq m_i$
- $m_0 + m_1 + \dots + m_{N-1} \leq 10^5$

- $0 \leq a_0, a_1, A, B, C, D, E, F \leq 10^5$

## Output

請輸出  $T$  行，依序代表小  $N$  每天所在的城鎮編號（因此第一行一定是  $S$ ）。

Sample Input 1	Sample Output 1
4 0 9	0
1 0 0 0 0 0 1 2	2
3 1 3 2	0
1 0	3
2 0 0	0
1 0	3
	0
	3
	0

Sample Input 2	Sample Output 2
4 1 8	1
1 2 3 4 5 6 7 8	0
2 2 1	2
2 0 3	1
2 1 0	0
2 2 1	2
	1
	0