



Asia-Pacific Informatics Olympiad 10th May 2008

TASK	BEADS	ROADS	DNA
input	standard input		
output	interaction	standard output	
time limit	2 second	1 second	1 second
memory limit	256 MB	128 MB	128 MB
points	100	100	100
	300		

Language	Compiler Directive
C	<code>gcc -o abc abc.c -std=c99 -O2 -DCONTEST -s -static -lm</code>
C++	<code>g++ -o abc abc.cpp -O2 -DCONTEST -s -static</code>
Pascal	<code>fpc -O1 -XS -dCONTEST abc.pas</code>

Duration: 5 hours

3 problems

All problems should be attempted.

串珠問題 (Beads)

X教授最近將他的一個最新發明公諸於世，那就是：終極串珠交換機器(Ultimate Bead Swapper，簡稱UBS)。就如同其命名，這個機器會將放置在機器上的一些串珠進行交換，製造出比較特別的串珠列。

這個UBS有 N 個輸送帶，以南北向平行放置。這些輸送帶由左而右依序從1編號到 N ，每一個輸送帶以相同的速度由北向南輸送。有 M 個交換器安置在相鄰的輸送帶上，且每個交換器位置和UBS北端的距離皆不相等(換句話說，這些交換器可以根據他們和UBS北端的距離大小排出一個前後順序)，因此可依這些交換器的位置由北而南從1編號到 M 。圖1顯示從上方看到的UBS 之示意圖。

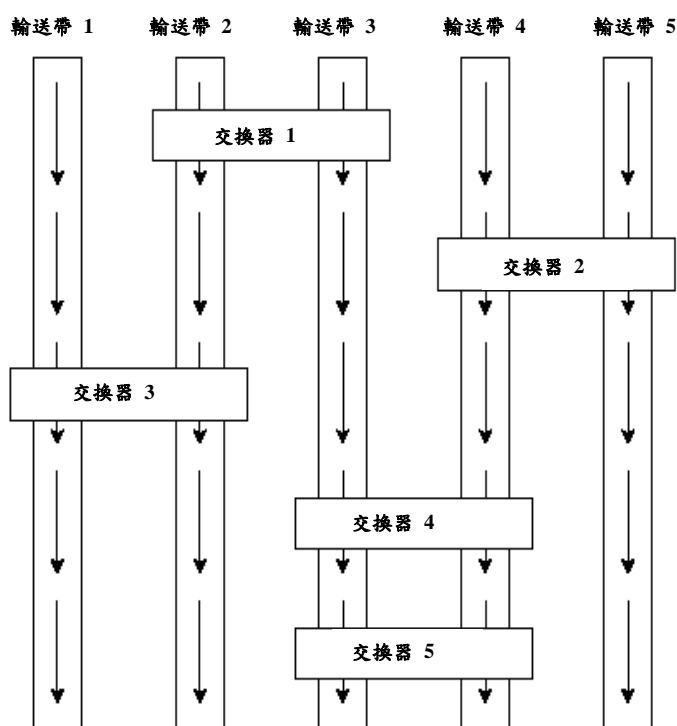


圖1:具有5個輸送帶及5個交換器的一個終極串珠交換機器(UBS).

當使用這個UBS時， N 個串珠會被同時擺在各個輸送帶的北端，因此當他們在輸送帶上移動時會排成一條水平線。當兩個串珠移動到一個交換器底下時，該交換器右邊輸送帶上的串珠會移到交換器左邊的輸送帶，左邊輸送帶上的串珠會同時移到右邊的輸送帶。當串珠交換之後，這兩個串珠仍維持和其他串珠排成一條水平線，圖2中顯示一個交換器的運作方式。

任務

當給定輸送帶數目 N ，交換器數目 M ，以及每個交換器的位置，請你寫一個程式回答以下形式的問題：

給定一組 K 及 J 值，請問起初被放在UBS中編號 K 的輸送帶北端之串珠，在所有串珠剛通過編號 J 的交換器後，會位於哪一個編號的輸送帶上？

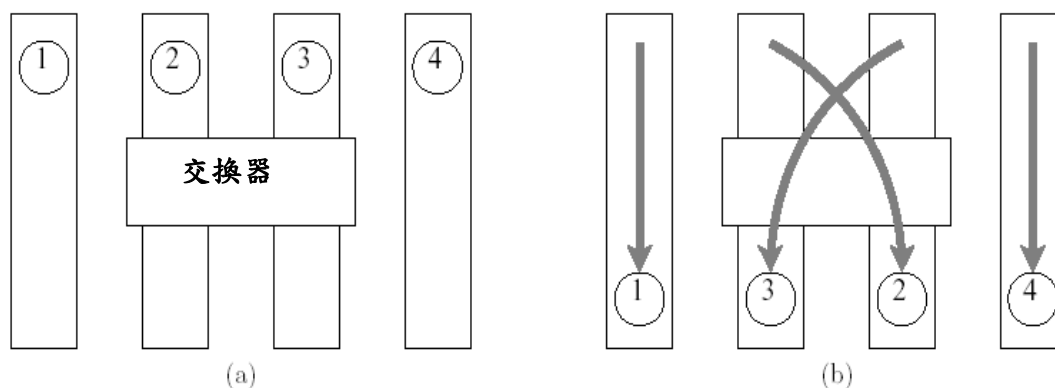


圖2: (a)四個串珠在輸送帶上被移動。(b)串珠2及串珠3在經過交換器底下後交換彼此所在的輸送帶。

輸入說明

你的程式必須由標準輸入讀進資料。第一行包含輸送帶數目 N ($1 \leq N \leq 300,000$) 以及交換器數目 M ($1 \leq M \leq 300,000$)。

接下來的 M 行分別輸入由北而南的交換器資料，每一行包含一個整數值 P ($1 \leq P \leq M - 1$)，表示該交換器跨越於編號 P 及編號 $P + 1$ 的輸送帶上。

函式呼叫方式

當讀取以上所描述的輸入資料後，你的程式必須呼叫以下表1中所列的函式，且必須依下述順序呼叫這些函式：

1. 呼叫函式 `getNumQuestions` 取得 Q 值 ($1 \leq Q \leq 300,000$)， Q 值表示接下來要給定的問題數目。
2. 接下來必須執行 Q 次以下指令：
 - (a) 呼叫函式 `getQuestion` 取得一組問題。
 - (b) 呼叫函式 `answer` 來回答所取到該組問題的解答。

我們要特別強調，函式 `getNumQuestions` 必須先被呼叫，且只能被呼叫一次。函式 `getQuestion` 及 `answer` 必須被交替呼叫：在呼叫函式 `getQuestion` 後，你的程式必須先呼叫函式 `answer` 後，才能再呼叫 `getQuestion`，反之亦同。若你的程式在執行一組測試資料時違反這個規定，你在該組測試資料將得到0分。

程式撰寫指令

若你要送出的原始碼是採用 Pascal 程式語言，你的原始碼中必須包含以下敘述：

```
uses beadslib;
```

若你要送出的原始碼是採用 C 或 C++ 程式語言，你的原始碼中必須包含以下敘述：

```
#include "beadslib.h"
```

表 1: 函式庫呼叫方式

函式宣告	函式功能描述
Pascal function getNumQuestions():integer C and C++ int getNumQuestions()	回傳你的程式必須回答的問題數目。
Pascal procedure getQuestion(var K:integer, var J:integer) C void getQuestion(int *K, int *J) C++ void getQuestion(int &K, int &J)	K 會被設定成一個輸送帶編號，來指定起初被放在UBS該輸送帶北端之串珠。 J 會被設定成所指定交換器的編號。
Pascal procedure answer(x:integer) C and C++ void answer(int x)	對最近一次呼叫getQuestion 讀取到的問題，回答其解答為x。

模擬函式庫及範例程式

你會被提供一個zip檔，其中包含範例函式庫及程式的原始碼。該檔案中包含三個目錄：pascal、c、及cpp，分別對應到Pascal、C、及C++三種程式語言的原始碼。每一個目錄中包含一個模擬函式庫的原始碼，以及以正確順序呼叫該函式庫中各函式的一個程式之原始碼。

對於Pascal程式語言，模擬函式庫被包含在beadslib單元中，其原始碼在beadslib.pas檔中。檔案 sample.pas是正確使用該函式庫的範例程式之原始碼。

對於C程式語言，模擬函式庫中函式的宣告在beadslib.h檔中，其原始碼在 beadslib.c檔中。檔案 sample.c是正確使用該函式庫的範例程式之原始碼。

對於C++程式語言，模擬函式庫的函式的宣告在beadslib.h檔中(但該檔案與C程式語言對應的beadslib.h 檔不同)，其原始碼在 beadslib.cpp檔中。檔案 sample.cpp 是正確使用該函式庫的範例程式之原始碼。

該模擬函式庫的運作方式如下：

- 當呼叫函式庫中的getNumQuestions函式時，它會打開questions.txt檔，讀取問題的數目，並回傳它讀到的數值。
- 當呼叫getQuestion函式時，它會從questions.txt讀取K及J的值。
- 當呼叫answer函式時，它會將參數 x 之值印出到標準輸出。
- 每當有函式被呼叫的順序錯誤時，函式庫會印出一個錯誤訊息到標準輸出。

questions.txt 檔案的格式如下：第一行包含問題的個數 Q ，接下來的 Q 行，每一行包含兩個整數值 K 及 J ，其中 K 表示輸送帶的編號，而 J 表示交換器的編號。

輸入範例

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

questions.txt 的內容範例

```
2
3 4
5 5
```

(此範例符合圖1的描述)

函式呼叫範例

呼叫函式	回傳值及解釋
getNumQuestions();	2 表示程式必須回答兩個問題。
Pascal getQuestion(K, J); C getQuestion(&K, &J); C++ getQuestion(K, J);	K=3, J=4 對於起初被放在編號3的輸送帶北端之串珠，在所有串珠剛通過編號4的交換器後，會被交換到哪一個編號的輸送帶?
answer(1);	回答當所有串珠剛通過編號4的交換器後，該串珠會在編號1的輸送帶上。
Pascal getQuestion(K, J); C getQuestion(&K, &J); C++ getQuestion(K, J);	K=5, J=5 對於起初被放在編號5的輸送帶北端之串珠，在所有串珠剛通過編號5的交換器後，會被交換到哪一個編號的輸送帶?
answer(4);	回答當所有串珠剛通過編號5的交換器後，該串珠會在編號4的輸送帶上。

時間及記憶體空間限制

你的程式必須在2秒鐘之內執行結束，且使用的記憶體空間不得超過256 MB

評分

對於每組測試資料，如果你的程式符合上述函式呼叫的規定且答對所有問題，則可完全得到該組測試資料對應的所有分數，否則得到0分。

在共20分的測試資料組別中，其M和Q值最大為10,000。

道路問題 (Roads)

新亞洲王國內共有 M 條道路連結 N 個村莊。有些道路是鵝卵石路，有些則是混凝土路。要保持所有道路免收費使用就需要龐大的支出，因此這似乎是不可能的。所以王國需要一個新的道路維護計畫。

國王最終決定要保留越少道路越好，但是每兩個不同的村莊之間一定要保留一條，且只有一條，免費的路徑。此外，雖然混凝土路比較符合現今道路需求，國王認為走在鵝卵石路上是比較有趣的，因此他決定至少要保留 K 條免費的鵝卵石路。

舉例來說，假設新亞洲王國的村莊與道路如圖1(a)所示，如果國王要保留兩條免費的鵝卵石路，哪就可以如圖1(b)將道路 (1, 2), (2, 3), (3, 4) 和 (3, 5) 保留為免費通行。此安排符合國王的要求因為 (1)每兩個村莊之間都有一條免費道路，(2)免費道路越少越好，(3) 所有免費道路中剛好有兩條鵝卵石路，即 (2, 3) 和 (3, 4)。

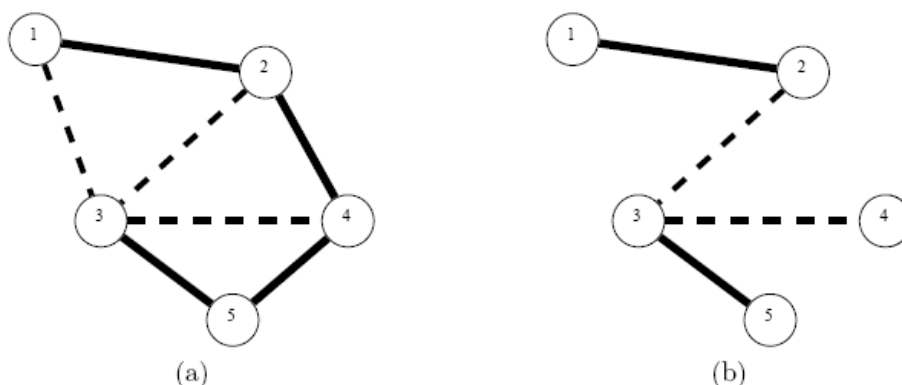


圖1：(a)新亞洲王國村莊與道路範例。實線代表混凝土路，虛線代表鵝卵石路。
(b)一個含有兩條免費鵝卵石路的道路維護計畫，圖中只畫出免費道路。

任務

給定新亞洲王國的道路圖，與國王所想要保持免費的鵝軟石路數量，請寫一個程式來判斷是否有可以符合國王要求的道路維護計畫，如果有，請輸出該計畫。

輸入說明

第一行有三個整數，以空白隔開。

- N ，代表村莊數 ($1 \leq N \leq 20,000$)
- M ，代表道路數 ($1 \leq M \leq 100,000$)
- K ，代表國王要求保留免費的鵝軟石路數量 ($0 \leq K \leq N - 1$)

接下來的 M 行描述新亞洲王國的道路狀況，第 $(i+1)$ 行代表道路 i ，道路編號以 1 到 M 代表之。每一行都有三個以空白隔開的整數：

- u_i 和 v_i ，代表道路 i 連接村莊 u_i 和 v_i ，村莊以數字 1 到 N 代表之。
- c_i ，代表道路 i 的型態：若 $c_i = 0$ 則道路 i 是鵝軟石路；若 $c_i = 1$ 則道路 i 是混凝土路。

每兩個村莊之間不會有超過一個連接兩村莊的道路。



輸出說明

若是不可能有一個符合國王要求的道路維護計畫，則在第一行輸出 no solution。否則，你的程式應該輸出一個可行的道路維護計畫中，該維持免費的道路。輸出的道路格式應與輸入資料中該道路的描述相同，輸出的道路可以以任何順序呈現。若是有一個以上符合要求的道路維護計畫，則只需要輸出其中一個即可。

輸入範例

```
5 7 2
1 3 0
4 5 1
3 2 0
5 3 1
4 3 0
1 2 1
4 2 1
```

(此範例符合圖1(a)的描述)

輸出範例

```
3 2 0
4 3 0
1 2 1
5 3 1
```

(此輸出符合圖1(b)的描述)

時間及記憶體空間限制

你的程式必須在1秒鐘之內執行結束，且使用的記憶體空間不得超過128 MB

評分

對於每組測試資料，如果輸出正確，就可得到 100% 的分數，否則就是 0% 的分數。此外，在共20分的測試資料組別中， K 值最大為10。

脫氧核糖核酸(DNA)

電腦的一項有趣的用途便是用來分析生物學的資料諸如 DNA 序列。就生物學而言，一般的 DNA 是由一連串的腺嘌呤(Adenine)、胞嘧啶(Cytosine)、鳥嘌呤(Guanine)、胸腺嘧啶(Thymine)等四種核苷酸(nucleotide)組成。這四種核苷酸將以字元 A、C、G 及 T 來分別表示。因此，一般的 DNA 可以用這四個字元的一連串組合來表示。我們稱這樣的字元串組合為一串 DNA 序列。

生物學家不能夠確定一串 DNA 序列中某些核苷酸的情況是有可能會發生的。在種情況下，可以用字元 N 來表示出現在 DNA 序列中某一位置的未知核苷酸。換句話說，N 可以視為一個萬用字元而用來表示 A、C、G 或 T 其中的任一個字元。若一串 DNA 序列中含有一個或更多的字元 N 就稱作「不完整序列」；否則，稱作「完整序列」。一串完整序列會被認為與另一串完整序列吻合，是發生在當它可以是經由以四種核苷酸中的任一個來取代不完整序列中的每個 N 而得的結果。例如 ACCCT 與 ACNNT 是吻合的，但 AGGAT 則是不吻合的。

研究人員往往會照依英文字母順序來排序這四種核苷酸：A 在 C 之前，C 在 G 之前，G 在 T 之前。一串 DNA 序列將被分類成「類型 1」，如果此序列中每一個核苷酸與右邊緊鄰核苷酸是相同或者其字母順序是在右邊緊鄰核苷酸的字母順序之前。例如 AACCGT 是類型 1，但 AACGTC 則不是。

通常一串類型 j 序列 ($j > 1$) 可以是一串類型 $(j-1)$ 序列，或者是由一串類型 $(j-1)$ 序列與一串類型 1 序列串接而成的。例如 AACCC、ACACC 與 ACACA 是類型 3 序列，但 GCACAC 與 ACACACA 則不是。

再者，研究人員也會依照英文辭典中詞的編纂順序方式來排序 DNA 序列。據此，第 1 個長度為 5 的類型 3 序列將是 AAAAA，最後一個是 TTTTT。另一個例子是，當考慮一串不完整序列 ACANNCNNG 時，前 7 串與它吻合的類型 3 序列是：

```
ACAAACAAG
ACAAACACG
ACAAACAGG
ACAAACCAG
ACAAACCCG
ACAAACCGG
ACAAACCTG
```

任務 (Task)

寫一個程式，當給定一串長度為 M 的不完整序列後，尋找與它吻合的第 R 串類型 K 序列。

輸入 (Input)

輸入第一行有 3 個整數以空白鍵分隔： M ($1 \leq M \leq 50,000$)、 K ($1 \leq K \leq 10$) 和 R ($1 \leq R \leq 2 \times 10^{12}$)。第二行包含一串長度為 M 的字元序列，代表不完整序列。與此串不完整序列吻合的類型 K 序列的個數將不會大於 4×10^{18} ，所以可以使用 C 與 C++ 程式語言中 long long 或者 Pascal 程式語言中 Int64 資料型態表示。此外， R 不會大於與所給定的不完整序列吻合之類型 K 序列的個數。

輸出 (Output)

在第一行列印出與輸入的不完整序列吻合的第 R 串類型 K 序列。

**輸入範例一 (Sample Input 1)**

```
9 3 5
ACANNCNNG
```

輸出範例一 (Sample Output 1)

```
ACAAACCCG
```

輸入範例二 (Sample Input 2)

```
5 4 10
ACANN
```

輸出範例二 (Sample Output 2)

```
ACAGC
```

程式撰寫注意事項 (Programming Remark)

在C與C++程式語言，你應當使用long long的資料型態。下面一段程式碼說明如何從標準輸出/輸入讀取long long的資料，以及將long long的資料寫至標準輸出/輸入：

```
long long a;
scanf("%lld",&a);
printf("%lld\n",a);
```

在Pascal程式語言，你應當使用Int64的資料型態。處理這種資料型態的資料並不需要特別的指令。

時間與記憶體限制 (Time and Memory Limits)

你的程式必須在1秒內結束，並且不能使用超過128MB的記憶體。

評分(Scoring)

對每一個輸入測試資料，如果輸出是正確的可以得到100%的分數，否則為0%。
在共20分的測試資料組別中， M 最多為10。