



คอมพิวเตอร์โอลิมปิกเอเชียแปซิฟิก  
๑๐ พฤษภาคม ๒๕๕๑

โจทย์	ลูกแก้ว (BEADS)	ถนน (ROADS)	สายพันธุกรรม (DNA)
ข้อมูลนำเข้า	standard input		
ข้อมูลส่งออก	โต้ตอบกับไลบรารี	standard output	
ข้อจำกัดของเวลา	๒ วินาที	๑ วินาที	๑ วินาที
ข้อจำกัดการใช้หน่วยความจำหลัก	๒๕๖ เมกะไบต์	๑๒๘ เมกะไบต์	๑๒๘ เมกะไบต์
คะแนน	๑๐๐	๑๐๐	๑๐๐
	๓๐๐		

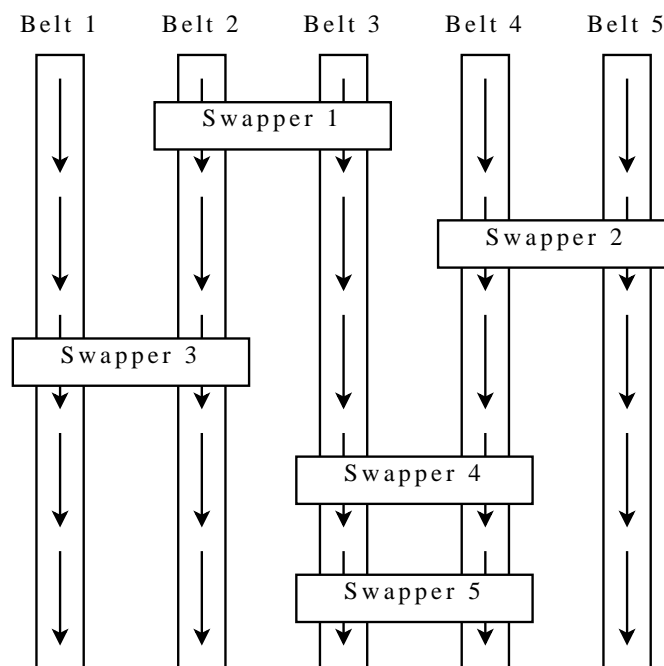
ภาษาโปรแกรม	Compiler Directive
C	<code>gcc -o abc abc.c -std=c99 -O2 -DCONTEST -s -static -lm</code>
C++	<code>g++ -o abc abc.cpp -O2 -DCONTEST -s -static</code>
ปาสคาล	<code>fpc -O1 -XS -dCONTEST abc.pas</code>

เวลาสอบ: ๕ ชั่วโมง  
โจทย์ทั้งหมด ๓ ข้อ  
พยายามทำทุกข้อ

## ลูกแก้ว (Beads)

โปรเฟสเซอร์ X ได้เผยโฉมนวัตกรรมล่าสุดให้ประจักษ์แก่ชาวโลก และขนานนามมันว่า เครื่องสลับลูกแก้วสะท้านโลกกันต์ (The Ultimate Bead Swapper – UBS) ฟังจากชื่อก็คงจะนึกออกว่าเครื่องนี้รับหน้าที่จัดลำดับของลูกแก้วเสียใหม่ให้น่าตื่นตามากขึ้น โดยลูกแก้วจะถูกสลับไปมาภายในรางที่อยู่ในเครื่อง

เครื่อง UBS ประกอบด้วยราง (belt)  $N$  ราง วางตัวขนานกันจากบนลงล่าง รางเหล่านี้ถูกกำหนดหมายเลขเป็น 1 ถึง  $N$  จากซ้ายไปขวา ตัวรางถูกออกแบบมาเป็นพิเศษเพื่อบังคับให้ลูกแก้ววิ่งด้วยความเร็วสม่ำเสมอและเท่ากันทุกราง ในเครื่อง UBS มีตัวสลับราง (swapper)  $M$  ตัวติดตั้งคร่อมอยู่ระหว่างคู่รางที่อยู่ติดกัน แต่จะไม่วางตัวอยู่ ณ ระดับเดียวกันตามแนวดิ่ง (นั่นคือตัวสลับรางทั้งหมดสามารถถูกนำมาจัดอันดับตามระยะทางจากจุดบนสุดได้โดยไม่กำกวม) เรากำหนดหมายเลขให้ตัวสลับรางจาก 1 ถึง  $M$  โดยนับลงมาจากด้านบนสุด รูปประกอบที่ 1 แสดงโครงสร้างของเครื่อง UBS เมื่อมองจากด้านหน้า



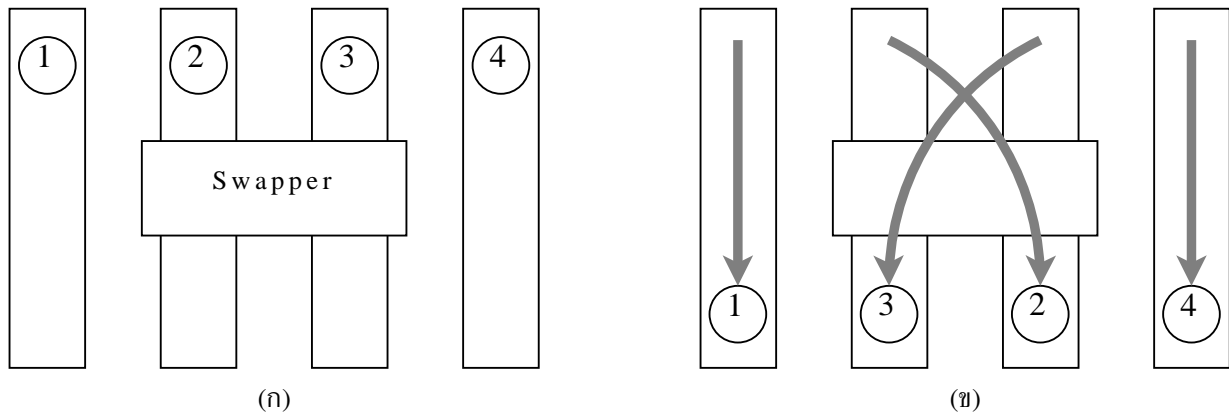
รูปที่ 1: เครื่องสลับลูกแก้วสะท้านโลกกันต์ (UBS) ที่ประกอบด้วยราง (belt) 5 ราง และตัวสลับราง (swapper) 5 ตัว

ในการเดินเครื่อง UBS ลูกแก้ว  $N$  ลูกจะถูกป้อนเข้าสู่รางทั้ง  $N$  ที่ด้านบนของเครื่องพร้อม ๆ กัน ลูกแก้วสองลูกที่ใหญ่มาถึงตัวสลับรางจะถูกสลับตำแหน่งกันโดยลูกที่อยู่บนรางฝั่งซ้ายจะถูกตีให้มาวิ่งอยู่บนรางฝั่งขวา และทำนองเดียวกัน ลูกแก้วบนรางฝั่งขวาจะไปวิ่งบนรางฝั่งซ้าย ลูกแก้วที่ถูกสลับที่ไม่มีทางกระเด็นออกนอกราง รูปประกอบที่ 2 แสดงการทำงานของตัวสลับราง

### งานของคุณ

จงเขียนโปรแกรมเพื่อรับค่าจำนวนราง ( $N$ ) จำนวนตัวสลับราง ( $M$ ) และตำแหน่งทั้งหมดของตัวสลับราง จากนั้นตอบคำถามในรูปแบบดังนี้

เมื่อกำหนดค่า  $K$  และ  $J$  ให้ ตอบคำถามว่าลูกแก้วที่ถูกป้อนเข้าสู่รางหมายเลข  $K$  ด้านบนของเครื่อง UBS จะถูกสลับไปอยู่ที่รางหมายเลขใดหลังจากวิ่งผ่านตัวสลับรางหมายเลข  $J$



รูปที่ 2: (ก) ลูกแก้วสี่ลูกไหลลงมาตามราง (ข) ลูกแก้วบนรางที่สองและสามสลับที่กันหลังจากวิ่งผ่านตัวสลับราง

### ข้อมูลนำเข้า

โปรแกรมของคุณจะอ่านข้อมูลจาก standard input บรรทัดแรกระบุจำนวนราง  $N$  ( $1 \leq N \leq 300,000$ ) และจำนวนตัวสลับราง  $M$  ( $1 \leq M \leq 300,000$ )

ตำแหน่งของตัวสลับรางกำหนดไว้ใน  $M$  บรรทัดถัดมาตามลำดับจากบนลงล่าง แต่ละบรรทัดระบุจำนวนเต็ม  $P$  ( $1 \leq P \leq M - 1$ ) ซึ่งหมายความว่าตัวสลับรางวางคร่อมไว้ระหว่างรางที่  $P$  และ  $P + 1$

### การโต้ตอบกับระบบทดสอบ

หลังจากอ่านข้อมูลนำเข้าแล้ว โปรแกรมของคุณจะเรียกฟังก์ชันจากไลบรารีตามที่ระบุไว้ในตารางที่ 1 ฟังก์ชันเหล่านั้นต้องถูกเรียกตามลำดับต่อไปนี้

1. เรียกฟังก์ชัน `getNumQuestions` เพื่ออ่านค่า  $Q$  ( $1 \leq Q \leq 300,000$ ) แสดงจำนวนคำถามทั้งหมด
2. ปฏิบัติตามขั้นตอนต่อไปนี้  $Q$  ครั้ง
  - (a) เรียกฟังก์ชัน `getQuestion` เพื่อรับคำถามถัดไป
  - (b) เรียกฟังก์ชัน `answer` เพื่อตอบคำถามที่ได้รับมา

ขอย้ำว่า `getNumQuestions` ต้องถูกเรียกก่อนฟังก์ชันอื่นและเรียกเพียงครั้งเดียว `getQuestion` และ `answer` จะถูกเรียกสลับไปมา นั่นคือหลังจากเรียก `getQuestion` แล้ว โปรแกรมต้องไม่เรียก `getQuestion` ซ้ำอีกจนกว่าจะเรียก `answer` เสียก่อน และไม่เรียก `answer` ซ้ำอีกจนกว่าจะเรียก `getQuestion` ใหม่ หากโปรแกรมละเมิดข้อตกลงดังกล่าวในระหว่างการใช้ชุดทดสอบใด คุณจะได้รับคะแนน 0% จากชุดทดสอบนั้น ๆ

### คำสั่ง

หากโปรแกรมที่ส่งเป็นภาษาปาสคาล รหัสต้นฉบับต้องปรากฏบรรทัดต่อไปนี้

```
uses beadslib;
```

หากโปรแกรมที่ส่งเป็นภาษา C หรือ C++ รหัสต้นฉบับต้องปรากฏบรรทัดต่อไปนี้

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

โปรโตไทป์ของฟังก์ชัน	คำอธิบาย
ปาสคาล <pre>function getNumQuestions():integer</pre> C และ C++ <pre>int getNumQuestions()</pre>	คืนค่าจำนวนคำถามที่โปรแกรมต้องประมวลผล
ปาสคาล <pre>procedure getQuestion(var K:integer, var J:integer)</pre> C <pre>void getQuestion(int *K, int *J)</pre> C++ <pre>void getQuestion(int &amp;K, int &amp;J)</pre>	$K$ จะได้รับค่าที่แสดงถึงหมายเลขของรางที่ลูกแก้ววางไว้ในตำแหน่งบนสุดของ UBS $J$ จะได้รับค่าที่แสดงถึงหมายเลขของตัวสลักราง
ปาสคาล <pre>procedure answer(x:integer)</pre> C และ C++ <pre>void answer(int x)</pre>	จะส่งค่าคำตอบ $x$ ที่ใช้ตอบคำถาม สำหรับคำถามที่ได้จากเรียก <code>getQuestion</code> ครั้งสุดท้าย

ตารางที่ 1: โปรโตไทป์ของฟังก์ชัน ของไลบรารีทดสอบ

## ไลบรารีทดสอบและตัวอย่างโปรแกรม

คุณจะได้รับแฟ้ม zip ที่ประกอบด้วย รหัสต้นฉบับ (source code) สำหรับไลบรารีและตัวอย่างการใช้โปรแกรม แฟ้มที่ได้รับนั้นจะประกอบด้วย 3 โฟลเดอร์ ได้แก่ pascal c และ cpp สำหรับรหัสต้นฉบับที่เขียนในภาษาปาสคาล C และ C++ ตามลำดับ ในแต่ละโฟลเดอร์ประกอบด้วยรหัสต้นฉบับของไลบรารีทดสอบและรหัสต้นฉบับของตัวอย่างโปรแกรมที่มีการเรียกใช้ฟังก์ชันภายในไลบรารีอย่างถูกต้อง

สำหรับภาษาปาสคาล ไลบรารีทดสอบจะประกอบด้วยยูนิท beadslib ซึ่งรหัสต้นฉบับของยูนิทนี้จะอยู่ในแฟ้ม beadslib.pas และแฟ้ม sample.pas จะเป็นรหัสต้นฉบับของโปรแกรมตัวอย่างที่มีการเรียกใช้คำสั่งต่างๆ ในไลบรารีอย่างถูกต้อง

สำหรับภาษา C โปรโตไทป์ (prototype) ของฟังก์ชันต่างๆ ที่มีในไลบรารีจะจัดเก็บไว้ใน beadslib.h ส่วนรหัสต้นฉบับของแต่ละฟังก์ชันจะถูกจัดเก็บไว้ใน beadslib.c และสำหรับแฟ้ม sample.c จะเป็นรหัสต้นฉบับของโปรแกรมตัวอย่างที่มีการเรียกใช้คำสั่งต่างๆ ในไลบรารีอย่างถูกต้อง

สำหรับภาษา C++ โปรโตไทป์ (prototype) ของฟังก์ชันต่างๆ ที่มีในไลบรารีจะจัดเก็บไว้ใน beadslib.h (ไฟล์เฮดเดอร์นี้จะแตกต่างจากไฟล์เฮดเดอร์ของภาษา C) รหัสต้นฉบับของแต่ละฟังก์ชันจะถูกจัดเก็บไว้ใน beadslib.cpp และสำหรับแฟ้ม sample.cpp จะเป็นรหัสต้นฉบับของโปรแกรมตัวอย่างที่มีการเรียกใช้คำสั่งต่างๆ ในไลบรารีอย่างถูกต้อง

ไลบรารีทดสอบจะมีลักษณะการทำงานดังนี้ (ให้พิจารณาตารางที่ 1 ประกอบ)

- เมื่อฟังก์ชัน `getNumQuestions` ของไลบรารีถูกเรียก มันจะเปิดแฟ้ม `questions.txt` เพื่ออ่านค่าจำนวนของคำถามและคืน (return) ค่านั้นออกมา
- เมื่อคำสั่ง `getQuestion` ถูกเรียกใช้ มันจะอ่านค่า  $K$  และ  $J$  จากแฟ้ม `questions.txt`
- เมื่อคำสั่ง `answer` ถูกเรียกใช้ มันจะพิมพ์ค่าของพารามิเตอร์  $x$  ออกไปยัง standard output
- ไลบรารีจะพิมพ์ข้อความที่ผิดพลาดออกไปยัง standard output ทุกครั้งที่ฟังก์ชันถูกเรียกด้วยลำดับที่ผิดพลาด

ในแฟ้ม `questions.txt` จะมีรูปแบบดังนี้คือ บรรทัดแรกจะประกอบด้วยจำนวนคำถาม ( $Q$ ), สำหรับต่อมาอีก  $Q$  บรรทัดในแต่ละบรรทัดจะประกอบด้วยจำนวนเต็ม 2 จำนวน คือ หมายเลขของรางที่เป็นจุดเริ่มต้นของลูกแก้ว ( $K$ ) และหมายเลขของตัวสลับราง ( $J$ )

**ตัวอย่างข้อมูลนำเข้า**

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

**ตัวอย่างข้อมูลในแฟ้ม `questions.txt`**

```
2
3 4
5 5
```

(ข้อมูลนำเข้าชุดนี้สอดคล้องรูปประกอบที่ 1)

**ตัวอย่างการโต้ตอบ**

การเรียกฟังก์ชัน	ค่าที่รับคืนและคำอธิบาย
<code>getNumQuestions();</code>	2 โปรแกรมจะถูกลูกแก้วทั้งหมด 2 คำถาม
ปาสคาล <code>getQuestion(K, J);</code> C <code>getQuestion(&amp;K, &amp;J);</code> C++ <code>getQuestion(K, J);</code>	$K=3, J=4$ คำถาม: พิจารณาลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 3 ในช่วงที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 4 ถามว่า ลูกแก้วลูกนี้อยู่บนรางที่เท่าใด?
<code>answer(1);</code>	หลังจากที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 4 ลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 3 อยู่บนรางที่ 1
ปาสคาล <code>getQuestion(K, J);</code> C <code>getQuestion(&amp;K, &amp;J);</code> C++ <code>getQuestion(K, J);</code>	$K=5, J=5$ คำถาม: พิจารณาลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 5 ในช่วงที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 5 ถามว่า ลูกแก้วลูกนี้อยู่บนรางที่เท่าใด?
<code>answer(4);</code>	หลังจากที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 5 ลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 5 อยู่บนรางที่ 4

**ข้อจำกัดเรื่องเวลาและหน่วยความจำ**

โปรแกรมจะต้องทำงานเสร็จสิ้นใน 2 วินาทีและใช้หน่วยความจำไม่เกิน 256 MB

### การให้คะแนน

ในแต่ละชุดทดสอบ ผู้แข่งขันจะได้คะแนนเต็ม ถ้าการทำงานของโปรแกรมที่เขียนขึ้นมีการเรียกใช้ฟังก์ชัน ที่เป็นไปตามรูปแบบที่กำหนดให้ และสามารถตอบคำถามทุกข้อในชุดทดสอบนั้นได้อย่างถูกต้อง มิฉะนั้นแล้วในชุดทดสอบนั้น จะไม่ได้รับการพิจารณาให้คะแนนเลย

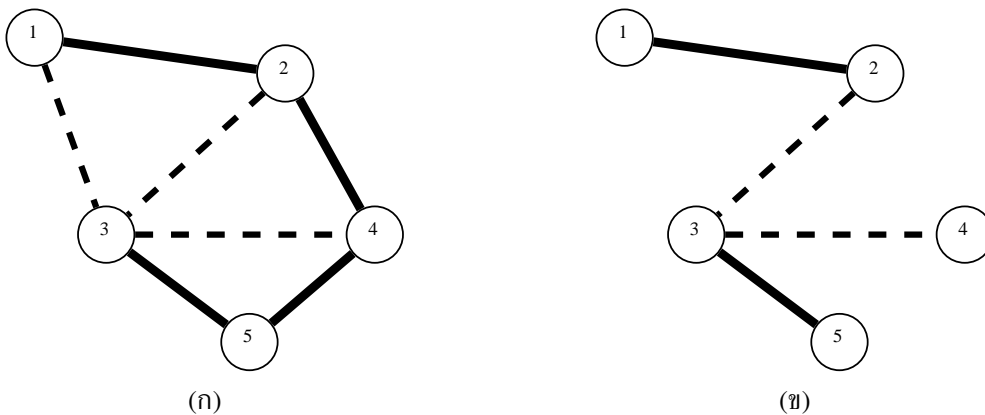
ในกรณีของชุดทดสอบที่มีค่า 20 คะแนน จะมีค่า  $M$  และ  $Q$  ไม่เกิน 10,000

## ถนน (Roads)

ในอาณาจักรเอเชียยุคใหม่ที่มีหมู่บ้านทั้งหมด  $N$  หมู่บ้าน โดยที่หมู่บ้านดังกล่าวมีเชื่อมต่อกันโดยใช้ถนนทั้งหมด  $M$  เส้น ทั้งนี้ ถนนดังกล่าวบางเส้นทำด้วยหินอ่อน ในขณะที่บางเส้นลาดด้วยคอนกรีต การที่จะทำถนนทั้งหมดให้ไม่มีค่าผ่านทางเลยจะต้องอาศัยเงินจำนวนมากในการทำนุบำรุง ซึ่งมันเป็นไปได้เลยในกรณีของอาณาจักรเอเชียยุคใหม่นี้ นั่นหมายความว่าถนนบางเส้นมีการเก็บค่าผ่านทาง ในขณะที่บางเส้นไม่มี เพราะฉะนั้นจึงมีความจำเป็นในการวางแผนของรูปแบบถนนทั้งหมดในอาณาจักร

ประมุขของอาณาจักรต้องการให้มีถนนที่ไม่มีค่าผ่านทางมีจำนวนน้อยที่สุดเท่าที่เป็นไปได้ กล่าวคือทุก ๆ สองหมู่บ้านที่แตกต่างกันใด ๆ จะต้องเชื่อมต่อให้ไปมาหากันได้ด้วยเส้นทางซึ่งประกอบด้วยถนนที่ไม่มีค่าผ่านทาง *จำนวนหนึ่งเส้น* เสมอ ไม่มากไม่น้อยไปกว่านี้ ทั้งนี้ถึงแม้ว่าการใช้ถนนคอนกรีตมีความเหมาะสมกับการคมนาคมยุคใหม่มากกว่า แต่ประมุขก็มีความคิดว่าการเดินเท้าบนถนนหินอ่อนก็มีความน่าสนใจและมีความสวยงามมากกว่า เพราะฉะนั้นจึงได้ตัดสินใจว่าภายในถนนที่ไม่เก็บค่าผ่านทางทั้งหมดนั้น จะต้องประกอบด้วยถนนหินอ่อน  $K$  เส้นพอดี

ตัวอย่างเช่น สมมติว่าหมู่บ้านและถนนในอาณาจักรเอเชียยุคใหม่เป็นไปตามรูปประกอบที่ 1ก ถ้าประมุขต้องการให้มีถนนหินอ่อนที่ไม่มีค่าผ่านทางทั้งหมด 2 เส้น ดังนั้นอาณาจักรสามารถทำให้ถนนเส้น (1,2), (2,3), (3,4) และ (3,5) ไม่มีค่าผ่านทางได้ดังที่แสดงในรูปประกอบที่ 1ข ซึ่งแบบแผนดังกล่าวเป็นไปตามความต้องการของประมุขเนื่องจาก 1) หมู่บ้านสองหมู่บ้านใดๆ สามารถเชื่อมโยงกันได้ด้วยเส้นทางหนึ่งเส้นเท่านั้น และเส้นทางดังกล่าวประกอบด้วยถนนที่ไม่มีค่าผ่านทาง 2) มีจำนวนถนนที่ไม่มีค่าผ่านทางน้อยเส้นที่สุดเท่าที่เป็นไปได้ และ 3) มีถนนหินอ่อนที่ไม่มีค่าผ่านทาง 2 เส้นคือ (2,3) และ (3,4)



รูปที่ 1: (ก) ตัวอย่างแสดงการเชื่อมต่อของถนนกับหมู่บ้านทั้งหมดในอาณาจักรเอเชียยุคใหม่ โดยที่เส้นทึบแสดงถึงถนนคอนกรีต ในขณะที่เส้นประแสดงถึงถนนหินอ่อน (ข) แบบแผนซึ่งแสดงถึงถนนทั้งหมดที่ไม่มีค่าผ่านทาง โดยมีถนนหินอ่อน 2 เส้นตามความต้องการของประมุข

### งานของคุณ

เมื่อกำหนดรูปแบบการเชื่อมต่อของถนนในอาณาจักรเอเชียยุคใหม่ และจำนวนของถนนหินอ่อนที่ประมุขต้องการให้ไม่มีค่าผ่านทาง จงเขียนโปรแกรมเพื่อหาความเป็นไปได้ของแบบแผนที่เป็นไปตามความต้องการของประมุข และแสดงผลแบบแผนดังกล่าวมาหนึ่งรูปแบบ (ถ้ามี)

### ข้อมูลนำเข้า

บรรทัดแรกมีตัวเลขทั้งหมด 3 ตัวคั่นโดยช่องว่างหนึ่งช่อง และมีรายละเอียดดังนี้

- $N$  จำนวนของหมู่บ้าน ( $1 \leq N \leq 20,000$ )

- $M$  จำนวนของถนน ( $1 \leq M \leq 100,000$ ) และ
- $K$  จำนวนถนนหินอ่อนที่ประมุขต้องการให้ไม่มีการเก็บค่าผ่านทาง ( $0 \leq K \leq N - 1$ )

$M$  บรรทัดถัดไปเป็นการอธิบายข้อกำหนดของถนนแต่ละเส้นในอาณาจักรเอเชียยุคใหม่ตั้งแต่เส้นที่ 1 ถึงเส้นที่  $M$  ซึ่งบรรทัดที่  $i + 1$  จะเป็นข้อกำหนดของถนนเส้นที่  $i$  และในแต่ละบรรทัดมีตัวเลขทั้งหมด 3 ตัวคั่นโดยช่องว่างหนึ่งช่อง และมีรายละเอียดดังนี้

- $u_i$  และ  $v_i$  เป็นหมายเลขของหมู่บ้านสองแห่งที่เชื่อมต่อกันด้วยถนนเส้นที่  $i$  โดยที่หมายเลขของหมู่บ้านจะถูกระบุโดยตัวเลขตั้งแต่ 1 ถึง  $N$  และ
- $c_i$  เป็นชนิดของถนนเส้นที่  $i$  โดยที่  $c_i = 0$  ถ้าถนนเส้นที่  $i$  ทำด้วยหินอ่อน ในขณะที่  $c_i = 1$  ถ้าถนนเส้นดังกล่าวลาดด้วยคอนกรีต

ทั้งนี้ในแต่ละหมู่บ้านใดๆจะมีถนนเชื่อมโยงกันไม่เกินหนึ่งเส้นเท่านั้น

### ข้อมูลส่งออก

- กรณีที่มีแบบแผนที่ตรงกับความต้องการ ให้แสดงผลแบบดังกล่าวมาหนึ่งแบบที่แสดงถึงถนนที่ไม่มีการเก็บค่าผ่านทางทั้งหมด โดยพิมพ์ข้อมูลของถนนออกมาบรรทัดละหนึ่งถนน หากมีแบบมากกว่าหนึ่งแบบ ให้เลือกแสดงผลแบบใดก็ได้ ในการพิมพ์ข้อมูลถนนเส้นใด ๆ ให้พิมพ์ข้อมูลเช่นเดียวกับข้อมูลนำเข้าในบรรทัดที่กำหนดถนนเส้นนั้น คุณสามารถเรียงถนนในลำดับอย่างไรก็ได้
- กรณีที่ไม่มีแบบแผนที่ตรงกับความต้องการเลย ให้แสดงผลคำว่า `no solution` เป็นบรรทัดแรกและบรรทัดเดียว

### ตัวอย่างข้อมูลนำเข้า

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

(ข้อมูลนำเข้าชุดนี้สอดคล้องรูปประกอบที่ 1ก)

### ตัวอย่างข้อมูลส่งออก

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

(ข้อมูลส่งออกชุดนี้สอดคล้องรูปประกอบที่ 1ข)

### ข้อจำกัดเรื่องเวลาและหน่วยความจำ

โปรแกรมจะต้องทำงานเสร็จสิ้นใน 1 วินาทีและใช้หน่วยความจำไม่เกิน 128 MB

### การให้คะแนน

คะแนนสำหรับแต่ละชุดทดสอบจะเป็น 100% ถ้าคำตอบถูกต้อง และเป็น 0% ถ้าหากไม่ถูกต้อง ในกรณีของชุดทดสอบที่มีค่า 20 คะแนน  $K$  จะมีค่าไม่เกิน 10



## สายพันธุกรรม (DNA)

เป็นเรื่องที่น่าสนใจมากที่เราสามารถนำคอมพิวเตอร์มาใช้ในการวิเคราะห์ข้อมูลทางชีววิทยา เช่น ข้อมูลของลำดับ DNA ได้ สาย DNA ประกอบด้วยนิวคลีโอไทด์ (Nucleotide) ซึ่งนิวคลีโอไทด์ทั้งสี่ชนิดนี้สามารถแทนด้วยตัวอักษร A, C, G, และ T ตามลำดับ ดังนั้นข้อมูลของสาย DNA จะสามารถแทนได้ด้วยสายอักขระที่ประกอบด้วยอักขระทั้งสี่ตัวนี้ เราจะเรียกสายอักขระนี้ว่า *ลำดับ DNA*

มีความเป็นไปได้ที่นักชีววิทยาไม่สามารถที่จะระบุชนิดของนิวคลีโอไทด์บางตัวในสาย DNA ในกรณีดังกล่าวนักชีววิทยาจะใช้ตัวอักษร N เพื่อแทน DNA ที่ไม่สามารถระบุชนิดได้ หรือกล่าวอีกนัยหนึ่งว่าตัวอักษร N เป็นอักขระตัวแทนที่ใช้แทนอักขระตัวใดก็ได้จาก A, C, G หรือ T เราเรียกลำดับ DNA ที่ประกอบด้วยตัวอักษร N ตั้งแต่หนึ่งตัวขึ้นไปว่า *ลำดับไม่สมบูรณ์* ในทางกลับกันเราจะเรียกลำดับ DNA ที่ไม่มีตัวอักษร N อยู่เลยว่าเป็น *ลำดับสมบูรณ์* และเราจะเรียกลำดับสมบูรณ์ตัวหนึ่งว่า *เข้ากันได้* กับลำดับไม่สมบูรณ์อีกตัวหนึ่งถ้าเราสามารถสร้างลำดับสมบูรณ์ตัวนั้นจากการแทนที่ตัวอักษร N ในลำดับไม่สมบูรณ์ด้วยตัวอักษรแทนนิวคลีโอไทด์ตัวใดตัวหนึ่งจากทั้งสี่ชนิด ตัวอย่างเช่น ACCCT เข้ากันได้กับ ACNNT แต่ AGGAT เข้ากันไม่ได้

นักวิจัยจะเรียงลำดับนิวคลีโอไทด์ทั้งสี่ตามลำดับของตัวอักษรภาษาอังกฤษ นั่นคือ A มาก่อน C, C มาก่อน G และ G มาก่อน T ลำดับ DNA จะถูกจัดประเภทเป็น *รูปแบบ-1* ถ้าทุกนิวคลีโอไทด์ในลำดับนั้นเป็นตัวเดียวกับนิวคลีโอไทด์ที่ติดกันทางขวาหรือเป็นตัวที่มีลำดับมาก่อน ตัวอย่างเช่น AACCGT เป็นลำดับรูปแบบ-1 แต่ AACGTC ไม่เป็น

ในกรณีทั่วไปลำดับ DNA จะเรียกว่า *รูปแบบ- $j$*  สำหรับ  $j > 1$  ถ้าลำดับนั้นเป็นลำดับรูปแบบ- $(j - 1)$  หรือ เกิดจากลำดับรูปแบบ- $(j - 1)$  ต่อกับลำดับรูปแบบ-1 ตัวอย่างเช่น AACCC, ACACC, และ ACACA เป็นลำดับรูปแบบ-3 แต่ GCACAC และ ACACACA ไม่ใช่

เช่นเดียวกันนักวิจัยเรียงลำดับ DNA ตามลำดับของคำในพจนานุกรมภาษาอังกฤษ ดังนั้นลำดับ DNA ตัวแรกในรูปแบบ-3 ที่มีความยาว 5 คือAAAAA และ ลำดับตัวสุดท้ายคือ TTTTT ตัวอย่างอีกอันหนึ่งของลำดับเจ็ดตัวแรกของลำดับสมบูรณ์รูปแบบ-3 ที่เข้ากันได้กับลำดับไม่สมบูรณ์ ACANNCNNG คือ:

```

ACAAACAAG
ACAAACACG
ACAAACAGG
ACAAACCAG
ACAAACCCG
ACAAACCGG
ACAAACCTG
    
```

### งานของคุณ

เขียนโปรแกรมเพื่อหาลำดับในรูปแบบ- $K$  ตัวที่  $R$  ที่เข้ากันได้กับลำดับไม่สมบูรณ์ที่มีความยาว  $M$  ที่กำหนดให้

### ข้อมูลนำเข้า

บรรทัดแรกประกอบด้วยตัวเลขจำนวนเต็มสามตัว  $M$  ( $1 \leq M \leq 50,000$ ),  $K$  ( $1 \leq K \leq 10$ ), และ  $R$  ( $1 \leq R \leq 2 \times 10^{12}$ ) แยกจากกันด้วยช่องว่างหนึ่งช่อง บรรทัดที่สองประกอบด้วยสายอักขระความยาว  $M$  ซึ่งเป็นลำดับไม่สมบูรณ์ เรารับประกันว่าลำดับรูปแบบ- $K$  ที่เข้ากันได้กับลำดับไม่สมบูรณ์จะมีจำนวนไม่เกิน  $4 \times 10^{18}$  ดังนั้นตัวเลขดังกล่าวจะสามารถแทนได้ด้วย long long ในภาษา C และ ภาษา C++ หรือ int64 ในภาษาปาสคาล นอกจากนี้  $R$  จะมีค่าไม่เกินจำนวนของลำดับรูปแบบ- $K$  ที่เข้ากันได้กับลำดับไม่สมบูรณ์ที่กำหนดให้

## ข้อมูลส่งออก

ให้พิมพ์ข้อมูลเพียงบรรทัดเดียวที่แสดงถึงลำดับในรูปแบบ- $K$  ตัวที่  $R$  ที่เข้ากันได้กับลำดับไม่สมบูรณ์ที่มีความยาว  $M$  ที่กำหนดให้

### ตัวอย่างข้อมูลนำเข้า 1

```
9 3 5
ACANNCNNG
```

### ตัวอย่างข้อมูลส่งออก 1

```
ACAAACCCG
```

### ตัวอย่างข้อมูลนำเข้า 2

```
5 4 10
ACANN
```

### ตัวอย่างข้อมูลส่งออก 2

```
ACAGC
```

## ข้อแนะนำในการเขียนโปรแกรม

ในภาษา C และ ภาษา C++ คุณควรประกาศชนิดข้อมูลเป็น long long ชุดคำสั่งต่อไปนี้แสดงตัวอย่างการอ่านและพิมพ์ค่าของข้อมูลชนิด long long จาก standard input/output

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

ในภาษาปาสคาล คุณควรประกาศชนิดข้อมูลเป็น Int64 ซึ่งไม่จำเป็นต้องใช้ชุดคำสั่งเฉพาะในการจัดการกับข้อมูลชนิดนี้

## ข้อจำกัดเรื่องเวลาและหน่วยความจำ

โปรแกรมของคุณจะต้องทำงานเสร็จภายใน 1 วินาที และใช้หน่วยความจำไม่เกิน 128 MB

## การให้คะแนน

ในแต่ละชุดข้อมูลทดสอบคุณจะได้ 100% ถ้าผลลัพธ์ถูกต้องในชุดข้อมูลนั้น และจะได้ 0% ในกรณีที่ตอบผิด ในกรณีของชุดทดสอบที่มีค่า 20 คะแนน  $M$  จะมีค่าไม่เกิน 10