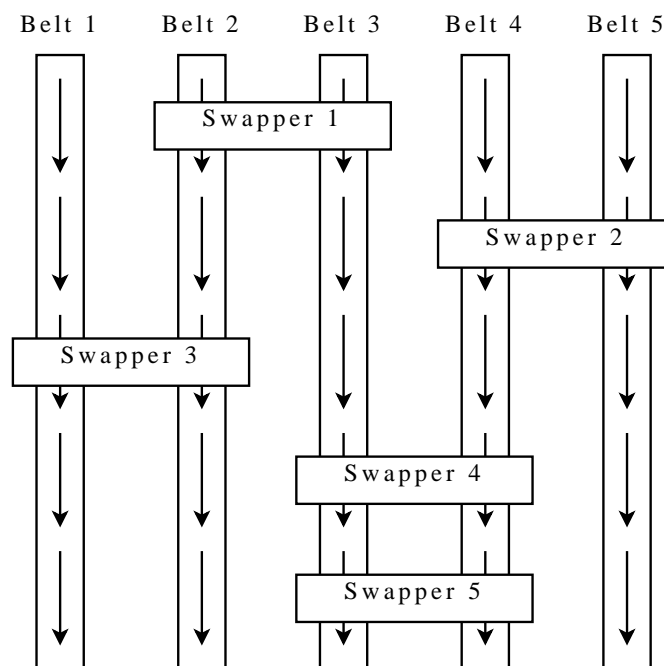


## ลูกแก้ว (Beads)

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



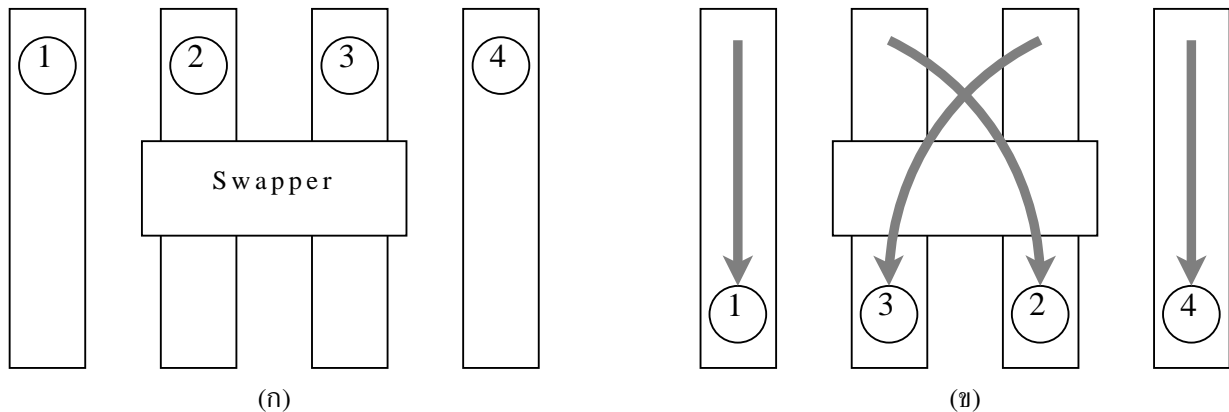
รูปที่ 1: เครื่องสลับลูกแก้วสะท้านโลกกันต์ (UBS) ที่ประกอบด้วยราง (belt) 5 ราง และตัวสลับราง (swapper) 5 ตัว

ในการเดินเครื่อง UBS ลูกแก้ว  $N$  ลูกจะถูกป้อนเข้าสู่รางทั้ง  $N$  ที่ด้านบนของเครื่องพร้อม ๆ กัน ลูกแก้วสองลูกที่ไหลมาถึงตัวสลับรางจะถูกสลับตำแหน่งกันโดยลูกที่อยู่บนรางฝั่งซ้ายจะถูกติดให้มาวิ่งอยู่บนรางฝั่งขวา 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

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

ขออย่าว่า `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
<p>ปาสคาล</p> <pre>function getNumQuestions():integer</pre> <p>C และ C++</p> <pre>int getNumQuestions()</pre>	คืนค่าจำนวนคำถามที่โปรแกรมต้องประมวลผล
<p>ปาสคาล</p> <pre>procedure getQuestion(var K:integer, var J:integer)</pre> <p>C</p> <pre>void getQuestion(int *K, int *J)</pre> <p>C++</p> <pre>void getQuestion(int &amp;K, int &amp;J)</pre>	<p><math>K</math> จะได้รับค่าที่แสดงถึงหมายเลขของรางที่ลูกแก้ววางไว้ในตำแหน่งบนสุดของ UBS</p> <p><math>J</math> จะได้รับค่าที่แสดงถึงหมายเลขของตัวสลักราง</p>
<p>ปาสคาล</p> <pre>procedure answer(x:integer)</pre> <p>C และ C++</p> <pre>void answer(int x)</pre>	จะส่งคำตอบ $x$ 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 (ให้พิจารณตารางที่ 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
<p>ปาสคาล</p> <pre>getQuestion(K, J);</pre> <p>C</p> <pre>getQuestion(&amp;K, &amp;J);</pre> <p>C++</p> <pre>getQuestion(K, J);</pre>	<p><math>K=3, J=4</math></p> <p>คำถาม: พิจารณาลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 3 ในช่วงที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 4 ถามว่า ลูกแก้วลูกนี้อยู่บนรางที่เท่าใด?</p>
<code>answer(1);</code>	หลังจากที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 4 ลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 3 อยู่บนรางที่ 1
<p>ปาสคาล</p> <pre>getQuestion(K, J);</pre> <p>C</p> <pre>getQuestion(&amp;K, &amp;J);</pre> <p>C++</p> <pre>getQuestion(K, J);</pre>	<p><math>K=5, J=5</math></p> <p>คำถาม: พิจารณาลูกแก้วที่ถูกใส่ที่ส่วนบนของรางที่ 5 ในช่วงที่ลูกแก้วทั้งหมดเพิ่งผ่านพ้นตัวสลับรางที่ 5 ถามว่า ลูกแก้วลูกนี้อยู่บนรางที่เท่าใด?</p>
<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