

Beads

Professor X has recently revealed his latest invention to the world: the Ultimate Bead Swapper (UBS). As the name implies, it can make a sequence of beads much more interesting by swapping some beads in it!

The UBS has N conveyor belts placed in north-south direction in parallel. The conveyor belts are numbered 1 to N from left to right. Every belt moves from north to south at the same speed. There are M swappers placed between adjacent conveyors. No two swappers are equally far from the north end of the UBS. (In other words, they can be totally ordered according to how far they are from the north end.) The swappers are numbered 1 to M from north to south. Figure 1 shows the UBS when viewed from above.

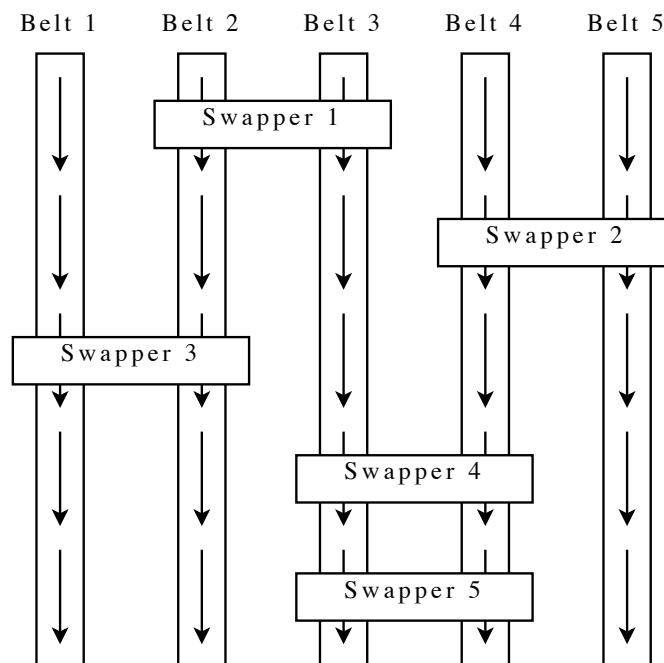


Figure 1: An Ultimate Bead Swapper with 5 conveyor belts and 5 swappers.

To use the UBS, N beads are placed at the north end of the conveyor belts at the same time so that they form a horizontal row as they move along the belt. When two beads come under a swapper, the bead on the right conveyor belt is moved to the left conveyor belt, and the bead on the left conveyor belt is moved to the right conveyor. After being swapped, the two beads do not break the horizontal row. Figure 2 illustrates the behavior of a swapper.

Task

Write a program that, given the number of conveyor belts N , the number of swappers M , and the positions of each swapper, answer questions of the form:

Given K and J , for the bead that is placed on Belt K at the north end of the UBS, which belt is the bead on after all beads just move past Swapper J ?

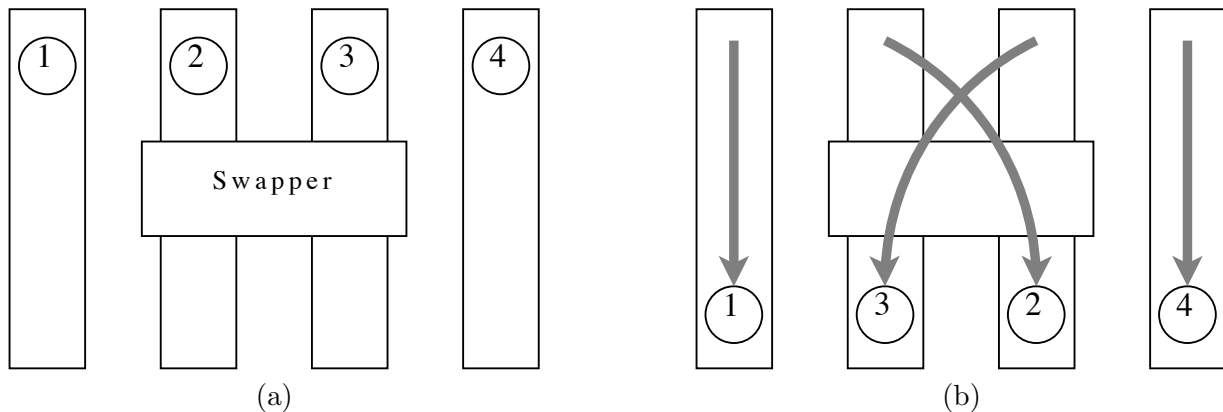


Figure 2: (a) Four beads move along the conveyor belts. (b) Bead 2 and 3 trade places after going under the swapper.

Input

Your program should read from standard input. The first line contains the number of conveyor belts N ($1 \leq N \leq 300,000$) and the number of swappers M ($1 \leq M \leq 300,000$).

Swappers are listed from north to south in the following M lines. Each line contains one integer P ($1 \leq P \leq M - 1$), meaning that there is a swapper over conveyor belt P and $P + 1$.

Interaction

After reading the input above, your program should call functions from the following library specified in Table 1. The functions must be called in the following order:

1. It calls the function `getNumQuestions` to retrieve Q ($1 \leq Q \leq 300,000$), the number of questions it will be asked.
2. For Q times, it should:
 - (a) Call the function `getQuestion` to receive one question.
 - (b) Call the function `answer` to answer the question it just received.

We emphasize that `getNumQuestions` must be called first and only once. `getQuestion` and `answer` must be called alternately: after calling one `getQuestion`, your program may not call `getQuestion` until it calls `answer`, and vice versa. If your program violates this convention when running a test scenario, you will receive a 0% score for that test scenario.

Programming Instructions

If you submit a Pascal source code, the source code must contain the statement:

```
uses beadslib;
```

If you submit a C or C++ source code, the source code must contain the line:

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

| Function Prototype | Description |
|--|--|
| Pascal <code>function getNumQuestions():longint</code> C and C++ <code>int getNumQuestions()</code> | Return the number of questions your program is going to be asked. |
| Pascal <code>procedure getQuestion(var K:longint; var J:longint)</code> C <code>void getQuestion(int *K, int *J)</code> C++ <code>void getQuestion(int &K, int &J)</code> | K is set to the number of the conveyor belt that the bead is placed at the north end of the UBS. J is set to the number of the swapper. |
| Pascal <code>procedure answer(x:longint)</code> C and C++ <code>void answer(int x)</code> | Report that the answer to the question corresponding to the last <code>getQuestion</code> is <code>x</code> . |

Table 1: Interaction library.

Mock Libraries and Sample Programs

You will be provided a zip file containing source code of sample libraries and programs. The file contains three directories — `pascal`, `c`, and `cpp` — for source code in Pascal, C, and C++, respectively. Each directory will contain a source code of a mock interaction library, and the source code of a program that calls the library functions in the correct order.

For Pascal, the mock interaction library is contained in the unit `beadslib`, whose source code is given in `beadslib.pas`. The file `sample.pas` is the source code of the program that uses the library correctly.

For C, the prototypes of the mock library functions are given in `beadslib.h`. The functions are specified in `beadslib.c`. The file `sample.c` is the source code of the program that uses the library correctly.

For C++, the prototypes of the mock library functions are also given in `beadslib.h` (but the file is not the same as the one for C). The functions are specified in `beadslib.cpp`. The file `sample.cpp` is the source code of the program that uses the library correctly.

The mock library behaves as follows:

- When `getNumQuestions` of the mock library is called, it opens the file `questions.txt`, reads the number of questions, and returns what is read.
- When `getQuestion` is called, it reads `K` and `J` from `questions.txt`.
- When `answer` is called, it prints the argument `x` to standard output.
- The library prints an error message to standard output every time a function is called out of the correct order.

The file `questions.txt` has the following format. The first line contains the number of questions `Q`. Each of the next `Q` lines contains two integers `K`, the number of a conveyor belt, and `J`, the number of a swapper.

Sample Input

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

Sample Content of questions.txt

```
2
3 4
5 5
```

(This input agrees with Figure 1)

Sample Interaction

| Function Call | Return Value(s) and Explanation |
|---|--|
| <code>getNumQuestions();</code> | 2 The program will be asked two questions. |
| Pascal <code>getQuestion(K, J);</code> C <code>getQuestion(&K, &J);</code> C++ <code>getQuestion(K, J);</code> | K=3, J=4 For the bead that is placed on Belt 3 at the north end of the UBS, which belt is the bead on after all beads just move past Swapper 4? |
| <code>answer(1);</code> | After every bead passes Swapper 4, the bead in question is on Belt 1. |
| Pascal <code>getQuestion(K, J);</code> C <code>getQuestion(&K, &J);</code> C++ <code>getQuestion(K, J);</code> | K=5, J=5 For the bead that is placed on Belt 5 at the north end of the UBS, which belt is the bead on after all beads just move past Swapper 5? |
| <code>answer(4);</code> | After every bead passes Swapper 5, the bead in question is on Belt 4. |

Time and Memory Limits

Your program must terminate in 2 seconds and use no more than 256 MB of memory.

Scoring

The score for each input scenario will be 100% if your program follows the function calling convention above and answers every question correctly, and 0% otherwise.

In test scenarios worthing 20 points, M , and Q will be at most 10,000.