



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.

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.

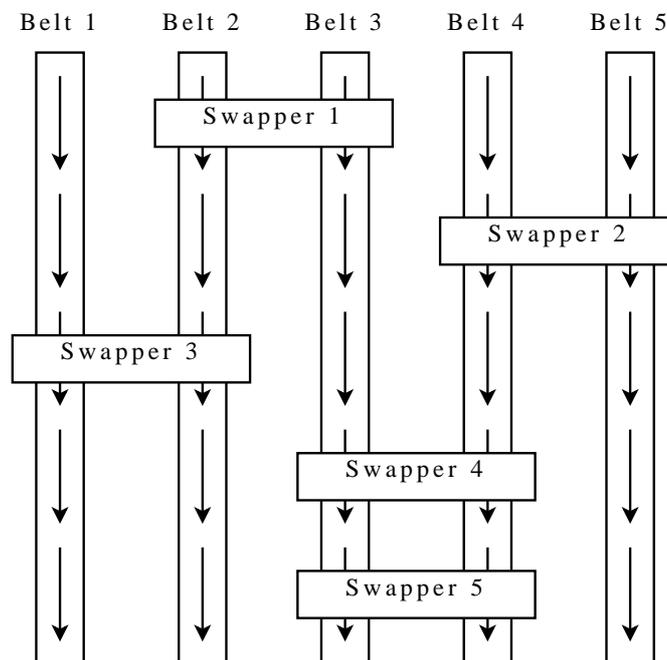


Figure 1: An Ultimate Bead Swapper with 5 conveyor belts and 5 swappers.

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 ?

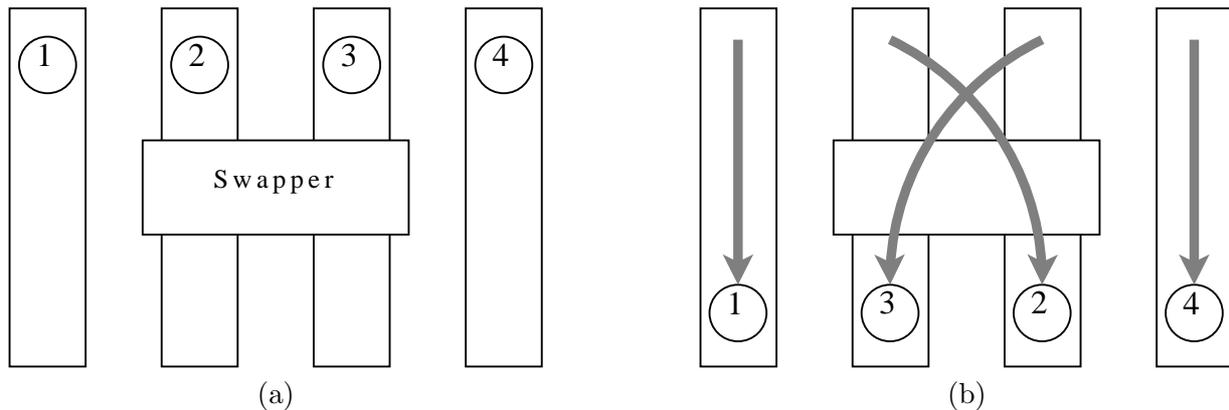


Figure 2: (a) Four beads move along the conveyor belts. (b) Bead 2 and 3 trade places after going under the swapper.

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.

Roads

The *Kingdom of New Asia* contains N villages connected by M roads. Some roads are made of cobblestones, and others are made of concrete. Keeping roads free-of-charge needs a lot of money, and it seems impossible for the Kingdom to maintain every road. A new road maintaining plan is needed.

The King has decided that the Kingdom will keep as few free roads as possible, but every two distinct villages must be connected by *one and only one* path of free roads. Also, although concrete roads are more suitable for modern traffic, the King thinks walking on cobblestones is interesting. As a result, he decides that *exactly* K cobblestone roads will be kept free.

For example, suppose the villages and the roads in New Asia are as in Figure 1a. If the King wants to keep two cobblestone roads free, then the Kingdom can keep roads (1,2), (2,3), (3,4), and (3,5) free as in Figure 1b. This plan satisfies the King's criteria because (1) every two villages are connected via one and only one path of free roads, (2) there are as few free roads as possible, and (3) there are exactly two cobblestone roads: (2,3) and (3,4).

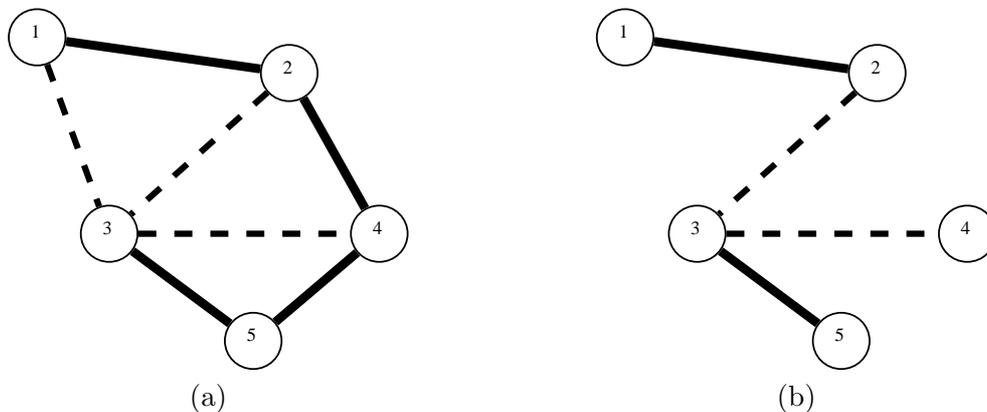


Figure 1: (a) An example configuration of villages and roads in the Kingdom of New Asia. Solid lines denote concrete roads, and dashed lines denote cobblestone roads. (b) A road maintaining plan that keeps two cobblestone roads free. Only free roads are shown.

Task

Given a description of roads in New Asia and the number of cobblestone roads that the King wants to keep free, write a program to determine if there is a road maintaining plan that satisfies the King's criteria, and output a valid plan if there is one.

Input

The first line contains three integers separated by one space:

- N , the number of villages ($1 \leq N \leq 20,000$),
- M , the number of roads ($1 \leq M \leq 100,000$), and
- K , the number of cobblestone roads the King wants to keep free ($0 \leq K \leq N - 1$).

The following M lines describes the roads in New Asia, which are numbered 1 to M . The $(i + 1)$ st line describes Road i . It contains three integers separated by one space:

- u_i and v_i , the two villages connected by Road i . Villages are numbered 1 to N , and
- c_i , the type of Road i ; $c_i = 0$ if Road i is made of cobblestone, and $c_i = 1$ if it is made of concrete.

There will be no more than one road connecting a pair of villages.

Output

If there is no road maintaining plan that satisfies the King's criteria, your program should print **no solution** on the first line of the output.

Otherwise, your program should output a valid road maintaining plan by listing roads that will be kept free, one road per line. To list a road, print the line in the input that describes it. The roads can be listed in any order. If there are more than one valid plan, you can output any such plan.

Sample Input

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

(This input agrees with Figure 1a.)

Sample Output

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

(This output agrees with Figure 1b.)

Time and Memory Limits

Your program must terminate in 1 second and use no more than 128 MB of memory.

Scoring

The score for each input scenario will be 100% if the correct answer is outputted and 0% otherwise.

In test scenarios worthing 20 points, K will be at most 10.

DNA

One interesting use of computer is to analyze biological data such as DNA sequences. Biologically, a strand of DNA is a chain of nucleotides Adenine, Cytosine, Guanine, and Thymine. The four nucleotides are represented by characters A, C, G, and T, respectively. Thus, a strand of DNA can be represented by a string of these four characters. We call such a string a *DNA sequence*.

It is possible that the biologists cannot determine some nucleotides in a DNA strand. In such a case, the character N is used to represent an unknown nucleotides in the DNA sequence of the strand. In other words, N is a wildcard character for any one character among A, C, G or T. We call a DNA sequence with one or more character N an *incomplete sequence*; otherwise, it is called a *complete sequence*. A complete sequence is said to *agree with* an incomplete sequence if it is a result of substituting each N in the incomplete sequence with one of the four nucleotides. For example, ACCCT agrees with ACNNT, but AGGAT does not.

Researchers often order the four nucleotides the way we order the English alphabets: A comes before C, C comes before G, G comes before T. A DNA sequence is classified as *form-1* if every nucleotide in it is the same as or comes before the nucleotides immediately to its right. For example, AACCGT is form-1, but AACGTC is not.

In general, a sequence is *form- j* , for $j > 1$, if it is a form- $(j - 1)$ or it is a concatenation of a form- $(j - 1)$ sequence and a form-1 sequence. For example, AACCC, ACACC, and ACACA are form-3, but GCACAC and ACACACA are not.

Again, researchers order DNA sequences lexicographically the way we order words in a dictionary. As such, the first form-3 sequence of length 5 is AAAAA, and the last is TTTTT. As another example, consider the incomplete sequence ACANNCNNG. The first seven form-3 sequences that agree with it are:

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

Task

Write a program to find the R th form- K sequence that agrees with the given incomplete sequence of length M .

Input

The first line contains three integers separated by one space: M ($1 \leq M \leq 50,000$), K ($1 \leq K \leq 10$), and R ($1 \leq R \leq 2 \times 10^{12}$). The second line contains a string of length M , which is the incomplete sequence. It is guaranteed that the number of form- K sequences that agrees with the incomplete sequence is not greater than 4×10^{18} , so it can be represented by a `long long` in C and C++ or an `Int64` in Pascal. Moreover, R does not exceed the number of form- K sequences that agree with the given incomplete sequence.

Output

On the first line, print the R th form- K sequence that agrees with the incomplete sequence in the input.

Sample Input 1

```
9 3 5
ACANNCNNG
```

Sample Output 1

```
ACAAACCCG
```

Sample Input 2

```
5 4 10
ACANN
```

Sample Output 2

```
ACAGC
```

Programming Remark

In C and C++, you should use `long long` data type. The following piece of code show how to read and write a `long long` from and to standard input/output:

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

In Pascal, you should use `Int64`. No special instructions are needed to manipulate data of this type.

Time and Memory Limits

Your program must terminate in 1 second and use no more than 128 MB of memory.

Scoring

The score for each input scenario will be 100% if the correct answer is outputted and 0% otherwise.

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