



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>

시험시간: 5 시간

3 문제

All questions should be attempted

구슬 목걸이

최근 김 교수는 구슬 목걸이를 만드는 시스템 UBS(Ultimate Bead Swapper, 최고의 구슬 교환기)를 만들었다. UBS(최고의 구슬 교환기)는 인접한 구슬을 교환하여 매력적인 구슬 목걸이를 만들어 낸다.

UBS는 상-하 방향으로 나란히 놓여있는 N 개의 컨베이어 벨트로 이루어져 있다. N 개의 컨베이어 벨트는 왼쪽부터 오른쪽으로 1부터 N 까지 번호가 매겨져 있다. 벨트는 위에서 아래 방향으로 모두 같은 속도로 움직인다. 인접한 컨베이어 벨트 사이에는 M 개의 구슬 교환기가 있다. 위로부터 같은 거리에는 단 하나의 구슬 교환기만 존재한다. 즉, M 개의 구슬 교환기를 위로부터의 거리가 적은 교환기로부터 순서대로 1부터 M 까지 번호를 매길 수 있다. (그림 1. 참조)

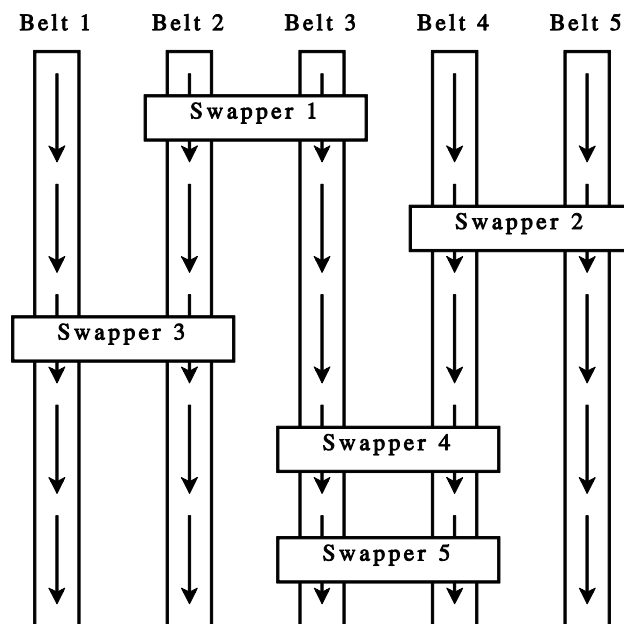


그림 1. 5개의 컨베이어 벨트와 5개의 구슬 교환기로 이루어진 UBS(Ultimate Bead Swapper)

UBS를 사용하기 위해서는 모든 컨베이어 벨트의 맨 윗부분(시작부분)에 하나씩의 구슬을 놓아둔다. N 개의 컨베이어 벨트에 있는 N 개의 구슬이 같은 속도로 같이 내려오다가 구슬 교환기(swapper)를 만나면 교환기가 걸쳐져 있는 두 컨베이어 벨트의 구슬들은 서로 위치가 바뀐다. 구슬이 바뀌어진 후에 모든 구슬은 같은 수평위치를 유지하며 같은 속도로 내려온다 (그림 2. 참조). 모든 구슬이 컨베이어 벨트의 맨 아랫부분(끝부분)에 도착하면 하나의 구슬 목걸이가 만들어지게 된다.

해야 할 작업

컨베이어 벨트의 수 N 과 구슬 교환기의 수 M , 그리고 구슬 교환기의 위치가 주어질 때 아래의 질문에 답하는 프로그램을 작성하라.

주어진 K, J 에 대하여, 컨베이어 벨트 번호 K 의 맨 윗부분에 있던 구슬이 구슬 교환기 J 번이 있는 위치를 지난 후에는 어느 컨베이어 벨트에 있는가?

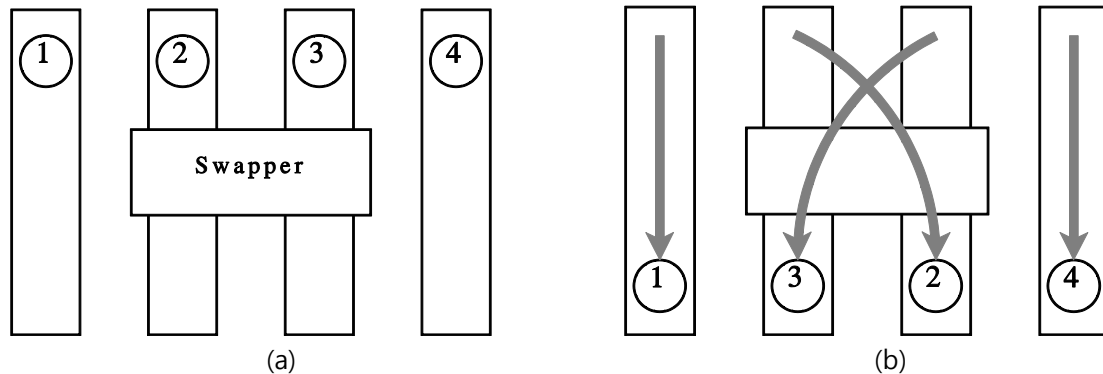


그림 2. (a) 4개의 구슬이 구슬 교환기를 지나기 전 상태

(b) 구슬 교환기에 의해 2번 구슬과 3번 구슬의 위치가 서로 바뀜

입력

모든 입력은 표준 입력으로 한다. 첫 번째 줄에는 컨베이어 벨트의 수 $N(1 \leq N \leq 300,000)$ 과 구슬 교환기의 수 $M(1 \leq M \leq 300,000)$ 이 주어진다. 구슬 교환기들은 위에서부터 아래의 순서로 M 줄에 걸쳐 주어진다. 각 줄에는 하나의 정수 $P(1 \leq P \leq M-1)$ 가 주어지는데 이는 컨베이어 벨트 P 와 $P+1$ 사이에 구슬 교환기가 있음을 의미한다.

상호작용

모든 입력을 읽은 후, 여러분의 프로그램은 테이블 1.에 명시된 라이브러리에 있는 함수를 아래에 지시된 순서대로 호출하여야 한다.

1. 질문의 수 $Q(1 \leq Q \leq 300,000)$ 를 찾기 위해 `getNumQuestions` 함수를 호출하여야 한다.
2. Q 번 동안 아래의 함수를 호출한다.
 - (a) `getQuestion` 함수를 호출하여 질문을 하나 받아온다.
 - (b) `answer` 함수를 호출하여 여러분들이 구한 답을 전해준다.

`getNumQuestions` 함수는 처음에 단 한번만 호출하여야 한다. `getQuestion`과 `answer` 함수는 반드시 교대로 호출되어야 한다. 즉, `getQuestion` 함수를 호출한 후 `answer` 함수를 호출하기 까지 `getQuestion` 함수를 호출하면 안 된다. 거꾸로도 역시 마찬가지다. 만약 이것을 지키지 않으면 0점을 받게 될 것이다.

프로그래밍 지침

만약 Pascal 언어로 작성된 프로그램을 제출 할 경우, 프로그램 소스 코드 내에는 다음의 문장이 반드시 포함되어 있어야 한다:

```
uses beadslib;
```

만약 C나 C++ 언어로 작성된 프로그램을 제출 할 경우, 프로그램 소스 코드 내에는 다음의 문장이 반드시 포함되어 있어야 한다:

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

Function Prototype	Description
Pascal function getNumQuestions():integer C and C++ int getNumQuestions()	질문의 수를 되돌려주는 함수이다.
Pascal procedure getQuestion(var K:integer, var J:integer) C void getQuestion(int *K, int *J) C++ void getQuestion(int &K, int &J)	하나의 질문 K와 J를 가져온다. K는 시작할 때(맨 윗부분)의 구슬이 있는 컨베이어 벨트 번호이고, J는 구슬 교환기 번호이다.
Pascal procedure answer(x:integer) C and C++ void answer(int x)	가장 최근의 getQuestion 함수에 대하여 답을 구한 컨베이어 벨트의 위치 x를 전해준다.

테이블 1. 상호작용 라이브러리

연습용 라이브러리와 예제 프로그램

여러분에게는 예제 라이브러리와 예제 프로그램의 소스 코드가 하나의 압축파일로 제공이 된다. 이 압축파일에는 각 Pascal, C 와 C++ 언어를 위한 세 개의 디렉토리 - pascal, c, cpp - 가 있다. 각 각의 디렉토리에는 연습을 위한 상호작용 라이브러리의 소스코드와 지침에 맞게 작성된 예제 프로그램의 소스 코드가 들어있다.

Pascal 언어 부분에는 연습용 상호작용 라이브러리가 beadslib 유닛(unit)에 포함되어 있고, 소스코드는 파일 beadslib.pas에 주어졌다. 파일 sample.pas에는 라이브러리를 바르게 사용하는 예제 프로그램이 들어있다.

C 언어 부분에는 연습용 라이브러리 함수의 프로토타입(prototype)들이 beadslib.h에 주어졌고, 함수의 내용은 beadslib.c에 들어있다. 파일 sample.c에는 라이브러리를 바르게 사용하는 예제 프로그램이 들어있다.

C++ 언어 부분에는 연습용 라이브러리 함수의 프로토타입(prototype)들이 beadslib.h에 주어졌고(파일의 내용은 C언어 부분의 것과 같지않음), 함수의 내용은 beadslib.cpp에 들어있다. 파일 sample.cpp에는 라이브러리를 바르게 사용하는 예제 프로그램이 들어있다.

연습용 라이브러리는 다음과 같이 동작한다.

- 연습용 라이브러리의 함수 getNumQuestions를 호출하면 파일 questions.txt에서 질문의 수를 읽고, 읽은 값을 되돌려준다.
- 함수 getQuestion을 호출하면 파일 questions.txt에서 K와 J를 읽는다.
- 함수 answer를 호출하면 인수 x를 표준출력장치로 출력한다.
- 잘못된 순서로 함수를 호출할 때 마다 표준출력장치에 오류 메시지가 출력된다.

파일 questions.txt의 첫 번째 줄에는 질문의 수 Q가 들어있다. 그 다음 Q개의 줄에는 각 줄에 두 정수 컨베이어 벨트의 번호 K와 구슬 교환기 번호 J가 주어졌다.

입력 예	예제 questions.txt의 내용
5 5	2
2	3 4
4	5 5
1	
3	
3	

(이 예는 그림 1과 같다)

예제 상호작용

Function Call	함수의 리턴 값과 설명
getNumQuestions();	2 이 프로그램은 두 개의 질문이 있다.
Pascal getQuestion(K, J); C getQuestion(&K, &J); C++ getQuestion(K, J);	K=3, J=4 UBS의 컨베이어 벨트 3번의 맨 윗부분에 있던 구슬이 구슬 교환기 4번이 있는 위치를 지난 후에는 몇 번의 컨베이어 벨트에 있는가?
answer(1);	모든 구슬이 구슬 교환기 4번이 있는 위치를 지난 후, 질문에서의 구슬은 1번 컨베이어 벨트에 있다.
Pascal getQuestion(K, J); C getQuestion(&K, &J); C++ getQuestion(K, J);	K=5, J=5 UBS의 컨베이어 벨트 5번의 맨 윗부분에 있던 구슬이 구슬 교환기 5번이 있는 위치를 지난 후에는 몇 번의 컨베이어 벨트에 있는가?
answer(4);	모든 구슬이 구슬 교환기 5번이 있는 위치를 지난 후, 질문에서의 구슬은 4번 컨베이어 벨트에 있다.

시간 및 메모리 제한

프로그램은 2초 이내에 수행되어야 하고, 사용 메모리는 256MB를 넘을 수 없다.

채점방법

각 입력에 대해 지시한대로 함수를 호출하고 또한 답이 맞으면 만점, 틀리면 0점이 주어진다.

테스트 데이터에서 M과 Q가 10,000 이하인 경우의 점수는 20점이다.

도로

뉴 아시아 왕국에는 M 개의 도로로 연결된 N 개의 마을이 있다. 어떤 도로들은 자갈로, 그리고 다른 도로들은 콘크리트로 포장되어 있다. 무료도로를 유지하기 위해서는 많은 경비가 필요하기 때문에 이 왕국에서 모든 도로를 무료화 하는 것은 불가능하여 보인다. 따라서 새로운 도로유지 방안이 필요하다.

왕은 무료도로의 수를 가능한 한 최소화하되, 어떤 마을에서 다른 마을로 갈 때 반드시 오직 하나의 무료도로로 이루어진 경로가 있도록 결정하였다. 또한 콘크리트 도로가 현대 교통에 적합하지만, 왕은 자갈길을 걷는 것이 매우 흥미 있다고 생각하였다. 결과적으로 왕은 정확히 K 개의 자갈도로를 무료도로로 유지하기로 결정하였다.

예를 들어 뉴 아시아 왕국의 마을과 도로는 그림 1a와 같다. 만일 왕이 2개의 자갈도로를 무료로 하고자 하면, 왕국은 그림 1b와 같이 (1, 2), (2, 3), (3, 4), (3, 5)를 무료로 하여야 한다. 이 계획은 왕의 요구사항을 만족하는데 그 이유는 (1) 모든 두 마을이 무료도로를 통하여 연결되어 있으며, (2) 가능한 가장 적은 수의 무료도로를 가지고 있고, (3) 정확히 두 개의 무료 자갈도로를 포함한다: (2, 3)과 (3, 4)

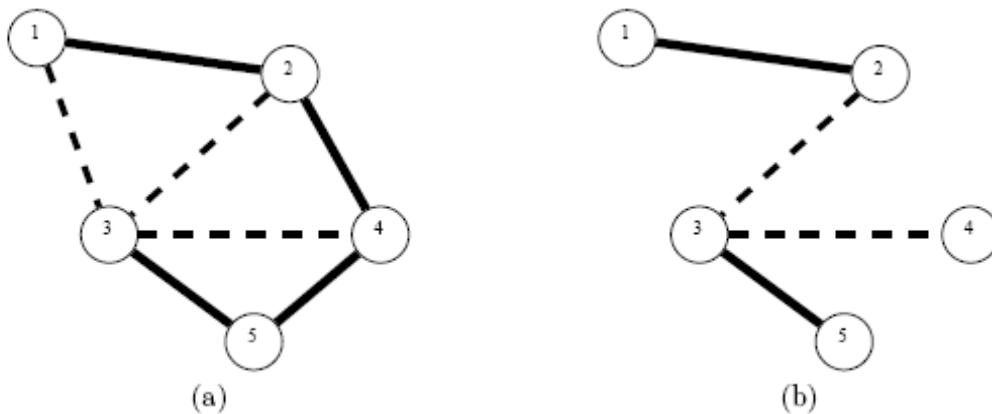


그림1: (a) 뉴 아시아 왕국의 마을과 도로 예. 실선은 콘크리트 도로를 점선은 자갈도로를 뜻함. (b) 정확히 두 개의 무료 자갈도로를 포함하는 도로유지방안. 무료도로만 표시되었음.

해야 할 작업

뉴 아시아 왕국의 도로와 왕이 원하는 무료자갈도로의 수가 주어졌을 때, 왕의 요구사항을 만족하는 도로유지방안이 있다면 그 방안을 출력하는 프로그램을 작성하는 것이다.

입력

첫 번째 줄에는 세 개의 정수가 하나의 빈칸을 사이에 두고 주어진다.

- N , 마을의 개수 ($1 \leq N \leq 20,000$),
- M , 도로의 개수 ($1 \leq M \leq 100,000$), 그리고
- K , 왕이 무료로 유지하기를 원하는 자갈도로의 개수 ($0 \leq K \leq N-1$)

다음 M 줄에는 1에서부터 M 까지의 도로가 기술된다. $(i + 1)$ 번째 줄은 도로 i 를 뜻한다. 여기에는 세 개의 정수가 하나의 빈칸을 사이에 두고 주어진다:

- u_i 와 v_i , 도로 i 에 의해 연결되는 두 마을. 마을은 1부터 N 으로 표시되며,
- c_i 는 도로 i 의 종류; $c_i = 0$ 은 자갈 도로, $c_i = 1$ 은 콘크리트 도로를 뜻한다.

인접한 두 마을을 연결하는 도로는 하나뿐이다.

출력

만일 왕의 요구사항을 만족하는 도로유지방안이 없다면, 여러분의 프로그램은 **no solution**을 첫 번째 줄에 출력하여야 한다.

그렇지 않은 경우, 여러분의 프로그램은 유효한 도로유지방안의 무료도로들을 한 줄에 하나씩 열거하여 출력하여야 한다. 도로는 입력 시 주어진 형태로 출력이 되어야 하나, 도로의 순서는 어떤 순서라도 무방하다. 만일 유효한 도로유지방안이 여러 개가 있는 경우, 그 중 하나만 출력하면 된다.

입력 예

5 7 2

1 3 0

4 5 1

3 2 0

5 3 1

4 3 0

1 2 1

4 2 1

(이 출력 예는 그림 1a와 같다)

출력 예

3 2 0

4 3 0

1 2 1

5 3 1

(이 출력 예는 그림 1b와 같다)

시간 및 메모리 제한

프로그램은 1초 이내에 수행되어야 하고, 사용 메모리는 128MB를 넘을 수 없다.

채점방법

각 입력에 대하여 정답이면 만점, 틀리면 0점이 주어진다.

테스트 데이터에서 K 값이 10 이하인 경우의 점수는 20점이다.

DNA

컴퓨터를 이용하여 해결하는 흥미로운 작업 중 하나는 **DNA** 서열과 같은 생물학적인 자료를 분석하는 것이다. 하나의 **DNA** 가닥은 뉴클레오티드인 아데닌, 사이토신, 구아닌, 티민으로 이루어진 체인이다. 이들 네 개의 뉴클레오티드는 각각 **A, C, G, T** 로 나타낸다. 따라서 **DNA** 가닥은 이들 네 문자의 문자열로 표현될 수 있다. 이 문자열을 **DNA 서열**이라 부른다.

DNA 가닥의 어떤 위치에 있는 뉴클레오티드가 정확하게 무엇인지 결정할 수 없는 경우가 있다. 이러한 경우, 이 가닥의 **DNA** 서열의 미확인 뉴클레오티드를 나타내는데 문자 **N** 을 사용한다. 달리 말하면, **N** 은 **A, C, G, T** 중 임의의 문자 하나를 위한 와일드카드 문자이다. 한 개 이상의 **N** 이 있는 **DNA** 서열을 **불완전서열**이라 부르고, 그렇지 않은 문자열을 **완전서열**이라 부른다. 불완전서열에 있는 **N** 각각에 대하여 이를 네 개의 뉴클레오티드 중 하나로 대체하여 완전서열을 얻을 수 있으면, 이 완전서열을 주어진 불완전서열과 **일치**한다고 말한다. 예를 들어, **ACCT** 는 **ACNNT** 과 일치하지만 **AGGAT** 는 일치하지 않는다.

연구자들은 네 개의 뉴클레오티드를 다음과 같이 영문자 순서대로 순서를 정한다: **A** 는 **C** 보다 앞에 나오고, **C** 는 **G** 보다 앞에 나오고, **G** 는 **T** 보다 앞에 나온다. **DNA** 서열에서 각 뉴클레오티드가 오른쪽 옆에 있는 뉴클레오티드와 같든지 혹은 오른쪽 옆에 있는 뉴클레오티드보다 이전 순서의 문자이면, 이 서열은 **형태-1** 로 분류된다. 예를 들어, **AACCGT** 는 형태-1 이지만 **AACGTC** 는 아니다.

일반적으로 **형태- j** 서열은 다음과 같이 정의된다: 이 서열이 형태- $(j-1)$ 이든지 혹은 어떤 형태- $(j-1)$ 서열 하나와 어떤 형태-1 서열 하나를 연결한 것이면 이를 형태- j 서열이라 한다. 예를 들어, **AACCC**, **ACACC** 와 **ACACA** 는 형태-3 서열이지만, **GCACAC** 와 **ACACACA** 는 형태-3 서열이 아니다.

연구자들은 사전에 단어들을 나열하는 순서인 사전식 순서대로 **DNA** 서열들의 순서를 정한다. 따라서 길이 5 인 형태-3 서열들 중 첫 번째 순서의 서열은 **AAAAA** 이고 마지막 순서의 서열은 **TTTTT** 이다. 다른 예로 불완전 서열 **ACANNCNNG** 를 고려해보자. 이 불완전서열과 일치하는 형태-3 서열들 중 첫 번째부터 순서대로 7 개를 나열하면 다음과 같다:

ACAAACAAG
ACAAACACG
ACAAACAGG
ACAAACCAG
ACAAACCCG
ACAAACCGG
ACAAACCTG

해야 할 작업

길이 M 인 불완전 서열과 일치하는 문자들 중 R 번째 순서의 형태- K 서열을 찾는 프로그램을 작성하시오.

입력

첫 번째 줄에 세 개의 정수가 빈칸 하나를 사이에 두고 주어진다: $M(1 \leq M \leq 50,000)$, $K(1 \leq K \leq 10)$ 과 $R(1 \leq R \leq 2 \times 10^{12})$. 두 번째 줄에 불완전서열인 길이 M 의 문자열이 주어진다. 불완전서열과 일치하는 형태- K 서열의 개수는 4×10^{18} 이하이다. 그러므로 이것은 C 나 C++에서 long long 으로 나타낼 수 있고 Pascal 에서는 Int64 로 나타낼 수 있다. 또한 R 은 주어진 불완전서열과 일치하는 형태- K 서열의 개수보다 같거나 작다.

출력

입력으로 주어진 불완전서열과 일치하는 형태- K 서열들 중 R 번째 것을 첫 번째 줄에 출력하시오.

입력 예 1

```
9 3 5
ACANNCNNG
```

출력 예 1

```
ACAAACCCG
```

입력 예 2

```
5 4 10
ACANN
```

출력 예 2

```
ACAGC
```

프로그래밍 유의사항

C 나 C++에서는 long long 자료형을 사용해야 한다. 다음 코드는 long long 형의 자료의 표준 입력 출력 방법을 보여준다.

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

Pascal 에서는 Int64 자료형을 사용해야 한다. 이 형의 자료를 다루는데 특별한 명령어는 필요하지 않는다.

시간 및 메모리 제한

프로그램은 1 초 이내에 수행되어야 하고, 사용 메모리는 128 MB 를 넘을 수 없다.

채점방법

각 입력에 대하여 정답이면 만점, 틀리면 0 점이 주어진다.

테스트 데이터에서 M 값이 10 이하인 경우의 점수는 20 점이다.