

UCPC greetings 23/24

Good Bye, BOJ 2023!

Official Problemset

주최 전국 대학생 프로그래밍 대회 동아리 연합

후원 STARTLINK, SAMSUNG SOFTWARE MEMBERSHIP, FURIOSA, Presto, HYUNDAI MOBIS, HYUNDAI AutoEver, Jane Street, MOLOCO, LG전자, 한빛미디어 Hanbit Media, Inc.



Good Bye, BOJ 2023!

문제지에 있는 문제가 총 7문제가 맞는지 확인하시기 바랍니다.

문제 목록

A 2023은 무엇이 특별할까?

B 거짓말

C 스티커 재배치

D 23E

E 전기 전송

F 등수

G 오직 5%의 사람들만이 이 문제를 풀 수 있습니다

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 2023은 무엇이 특별할까?

시간 제한	2 초
메모리 제한	1024 MB

백준 온라인 저지의 송년대회 Good Bye, BOJ 2023!의 개최일은 2023년 12월 31일이다. 키파는 대회가 개최된다는 사실이 기뻐 2023과 2024를 뚫어져라 보다가 무언가 특별하다는 사실을 깨달았다.

그렇다. 내년의 연도인 2024가 올해 연도의 끝 두 자리인 23으로 나누어 떨어진다! 다음에도 이런 연도가 오려면 6년 뒤인 2029/2030년이 되어야 한다.

양의 정수 N이 주어진다. N년과 $N+1$년이 이러한 조건을 만족하는 관계인지 판별하는 프로그램을 작성하라.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다.

이후 T개의 줄에 걸쳐 테스트 케이스가 한 줄에 하나씩 주어진다. 각 테스트 케이스는 한 줄로 구성되어 있으며, 각각 한 개의 양의 정수 N이 주어진다.

출력

각 테스트 케이스에 대해, $N+1$이 N의 끝 두 자리로 나누어 떨어진다면 `Good`을, 그렇지 않다면 `Bye`를 한 줄에 하나씩 차례로 출력하여라.

제한

- $1 \le T \le 8910$
- $1000 \le N \le 9999$
- N은 100의 배수가 아니다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 2023 2024 2029	Good Bye Good

이 페이지는 공백입니다

문제 B. 거짓말

시간 제한	2 초
메모리 제한	1024 MB

당신 앞에는 N명의 사람들이 있다. 각 사람은 자신을 포함하여 몇 명 이상이 거짓말을 하고 있다고 말하거나, 몇 명 이하의 사람이 거짓말을 하고 있다고 말한다.

예를 들어, 각 사람이 다음과 같이 주장한다고 하자.

- 첫 번째 사람이 이렇게 말한다: "1명 이상이 거짓말을 하고 있다!"
- 두 번째와 세 번째 사람이 이렇게 말한다: "1명 이하가 거짓말을 하고 있다!"
- 네 번째 사람이 이렇게 말한다: "2명 이상이 거짓말을 하고 있다!"

이 경우에 가능한 시나리오는 다음과 같다.

- 네 번째 사람만 거짓말을 하고 있다.
- 두 번째와 세 번째 사람이 거짓말을 하고 있다.

사람들의 주장이 주어질 때, 거짓말을 하는 사람의 수로 가능한 것을 모두 구해라.

입력

첫째 줄에 사람의 수 N이 주어진다.

둘째 줄에 N개의 정수 $k_1, \cdots, k_N$이 공백을 사이에 두고 주어진다.

- k_i가 양의 정수라면, i번째 사람이 "k_i명 이상이 거짓말을 하고 있다"고 말했다는 뜻이다.
- k_i가 0이거나 음의 정수라면, i번째 사람이 "$-k_i$명 이하가 거짓말을 하고 있다"고 말했다는 뜻이다.

출력

첫째 줄에 거짓말을 하는 사람의 수로 가능한 수의 개수를 출력한다.

둘째 줄에 가능한 수들을 공백을 사이에 두고 오름차순으로 출력한다.

제한

- $1 \le N \le 500\,000$
- $-N \le k_i \le N$ $(1 \le i \le N)$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 -1 -1 2	2 1 2
1 0	2 0 1

이 페이지는 공백입니다

문제 C. 스티커 재배치

시간 제한	2 초
메모리 제한	1024 MB

1번 스티커부터 M번 스티커까지 총 M종류의 스티커가 있다. j번 스티커에는 알파벳 소문자 s_j가 적혀 있다. 이때 같은 알파벳이 적힌 스티커가 여러 종류 있을 수 있다.

여러분은 N개의 칸으로 나누어진 보드판을 하나 갖고 있다. 각 칸에는 스티커가 하나씩 붙어 있고, i번째 칸에는 b_i번 스티커가 붙어 있다. 같은 종류의 스티커가 여러 개의 칸에 붙어 있는 것도 가능하다. 1번째 칸에 있는 스티커부터 N번째 칸에 있는 스티커까지 적힌 알파벳을 순서대로 읽으면 알파벳 소문자로 구성된 길이가 N인 문자열을 하나 얻는다. 이를 **스티커 문자열**이라고 정의한다.

여러분은 다음 시행을 통해 스티커 문자열을 바꿀 수 있다. 모든 시행을 완료하였을 때, 보드판에 빈칸이 있으면 안 된다.

- 스티커가 붙어 있는 칸의 스티커를 뗄 수 있다. 스티커는 강력 접착제로 붙어있기 때문에, j번 스티커를 떼는 데에는 d_j만큼의 비용이 든다. 스티커를 뗀 칸은 빈칸이 된다.
- 빈칸을 하나 골라서 스티커를 붙일 수 있다. 이때, 스티커를 새로 구매하여 붙이거나 이미 떼었던 스티커를 재활용하여 붙이는 것이 가능하다. 스티커를 붙일 때는 비용이 발생하지 않으며, j번 스티커를 한 장 구매하는 데에는 a_j만큼의 비용이 든다. 같은 종류의 스티커를 여러 장 구매하는 것도 가능하다.

여러분의 목표는, 적절하게 시행을 반복하여 스티커 문자열이 입력으로 주어진 길이가 $K(\leq N)$인 문자열 S를 부분 문자열로 포함하도록 만드는 것이다. 그러기 위한 최소 비용을 구하여라.

입력

첫 번째 줄에 정수 N, M, K가 공백으로 구분되어 주어진다.

두 번째 줄부터 M개의 줄에 걸쳐 스티커의 정보가 주어진다. 그중 j번째 줄에는 j번 스티커의 정보 s_j, d_j, a_j가 공백으로 구분되어 주어진다.

그다음 줄에 N개의 정수 $b_1, \cdots, b_N$이 공백으로 구분되어 주어진다. b_i는 보드판의 i번째 칸에 있는 스티커의 번호를 의미한다.

그다음 줄에 길이가 K인 알파벳 소문자로 구성된 문자열 S가 주어진다.

출력

스티커 문자열이 S를 부분 문자열로 포함하도록 만드는 데 필요한 최소 비용을 출력한다. 만약 S를 부분 문자열로 포함하도록 만들 수 없다면 -1을 출력한다.

제한

- $1 \leq N, M \leq 500$
- $1 \leq K \leq N$
- s_j는 알파벳 소문자이다. $(1 \leq j \leq M)$
- $1 \leq d_j, a_j \leq 1\,000$ $(1 \leq j \leq M)$
- $1 \leq b_i \leq M$ $(1 \leq i \leq N)$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 5 3 w 3 1 a 2 2 p 1 3 a 2 2 s 3 1 1 2 3 4 5 aaa	3
3 6 3 b 1 2 o 2 3 j 3 4 b 4 5 o 5 6 j 6 7 1 5 3 boj	0
3 4 3 g 1 1 o 2 2 o 3 3 d 4 4 3 2 1 bye	-1

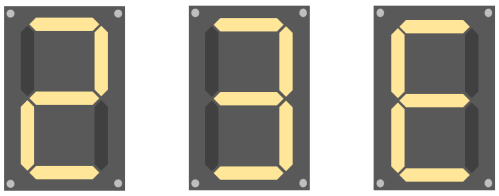
노트

문자열 T의 부분 문자열이란, 문자열 T에서 연속된 일부분에 해당하는 문자열을 의미한다. 예를 들어 문자열 T가 "goodbyeboj"이라면 "goo", "db", "j"는 T의 부분 문자열이고, "a", "god", "job"는 T의 부분 문자열이 아니다.

문제 D. 23E

시간 제한	2 초
메모리 제한	1024 MB

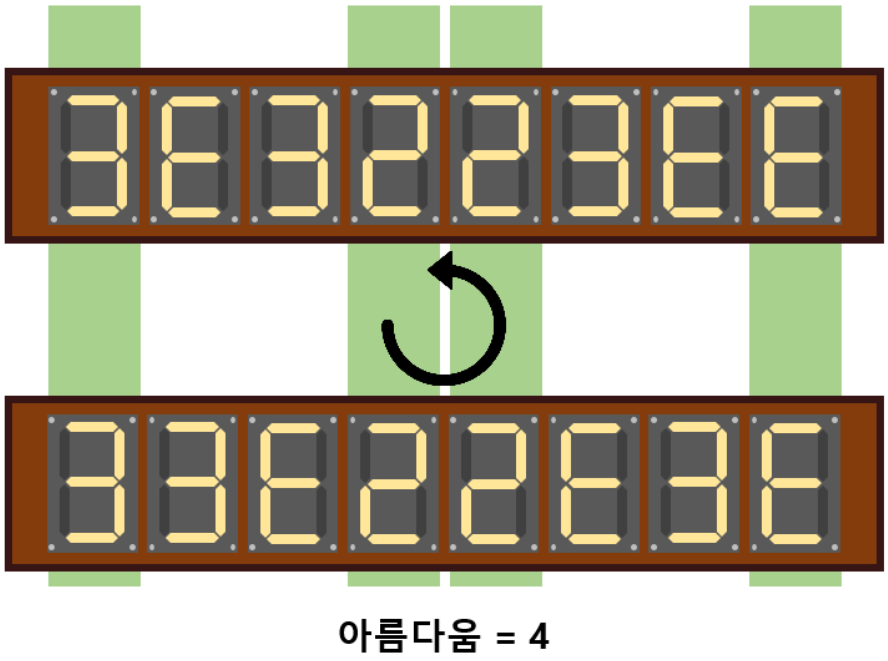
동현이는 2023년의 끝을 조금 특별하게 기념하기 위하여 7-Segment Display를 이용한 장식물을 만들기로 했다. 동현이는 N개의 Display가 일렬로 꼭 맞게 들어가는 틀을 준비하고, N개의 Display에 각각 2, 3, E 중 하나를 표시하여 틀에 모두 끼워 넣었다.



동현이는 짝수를 좋아하기 때문에 N은 짝수로 정하였다. 그리고, 동현이는 점대칭 역시 좋아하기 때문에 장식물을 180도 돌렸을 때 원래 모습과 비슷할수록 아름다운 장식물이라고 생각한다. 원래 장식물과 180도 돌린 장식물에서 위치와 표시된 글자가 모두 일치하는 Display의 개수를 장식물의 **아름다움**이라고 하자. 구체적으로는 아래와 같이 정의한다.

- 원래 장식물과 180도 돌린 장식물의 왼쪽에서 i번째 Display에 표시된 글자를 각각 S_i, T_i라고 하자. $(1 \le i \le N)$
- $f(2) = 2$, $f(3) = E$, $f(E) = 3$ 이라고 하자.
- $T_i = f(S_{N+1-i})$ 이다.
- 장식물의 아름다움은 $S_i = T_i$ 인 i $(1 \le i \le N)$ 의 개수이다.

예제 1의 입력에 해당하는 처음 장식물은 아래 그림과 같이 아름다움이 4임을 알 수 있다.



장식물에서 서로 다른 두 Display의 위치를 서로 바꾸는 것을 **교환**이라고 하자. 동현이는 처음에 만든 장식물에서 교환을 여러 차례 반복하여 장식물을 더욱 아름답게 만들려고 하는데, 교환은 귀찮은 일이므로 최대 K번까지만 하려고 한다. 동현이를 위해, 각 정수 $i(0 \le i \le K)$에 대해 처음 장식물에서 교환을 최대 i번 수행했을 때 얻을 수 있는 장식물의 최대 아름다움을 모두 구하자.

입력

첫째 줄에 동현이가 처음에 만든 장식물에 들어간 7-Segment Display의 개수 N, 최대 교환 횟수 K가 공백으로 구분되어 주어진다.

둘째 줄에 처음 장식물의 각 Display에 표시된 글자를 나타내는 길이 N의 문자열 S가 주어진다. S의 i번째 글자는 장식물의 왼쪽에서 i번째 Display에 표시된 글자이다.

출력

$K+1$개의 줄에 걸쳐, 이 중 i번째 줄에는 처음 장식물에서 교환을 최대 $i-1$번 수행했을 때 얻을 수 있는 장식물의 최대 아름다움을 출력한다.

제한

- $2 \le N, K \le 500\,000$
- N은 짝수이다.
- S의 각 문자는 2, 3, E 중 하나이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
8 2 3E3223EE	4 8 8
6 2 2233EE	0 2 6

문제 E. 전기 전송

시간 제한	3 초
메모리 제한	1024 MB

당신은 주원 전력 공사의 사장이다. 주원 전력 공사는 1번부터 N번까지 총 N개의 전력탑을 보유하고 있다.

그 중 $N-1$개의 전력탑 쌍은 서로 전선으로 이어져 있고, 전선에는 1번부터 $N-1$번까지 번호가 붙어 있다. i번 전선은 x_i번 전력탑과 y_i번 전력탑을 잇는다. 모든 전력탑은 전선을 통해 서로 이어져 있다.

전기의 수요는 시시각각 변하므로, 한 전력탑에서 다른 전력탑으로 전기를 보내야 하는 경우가 자주 생긴다. 문제는, 전기를 보낼 때 전선의 품질에 따라 손실이 발생한다는 것이다.

i번 전선의 저항은 r_i이고, 손실계수는 z_i이다. 한 전력탑에서 i번 전선을 통해 이어진 전력탑으로 전기를 e만큼 보내면, $\left\lfloor \frac{\max(e-r_i,0)}{z_i} \right\rfloor$ 만큼의 전기만이 도착하게 된다.

당신은 Q개의 질의를 처리해야 한다.

- a번 전력탑에서 e만큼의 전기를 b번 전력탑으로 보내려고 할 때, b번 전력탑에서 받을 수 있는 전기의 최댓값은 얼마인가?

입력

첫째 줄에 전력탑의 수 N이 주어진다.

다음 $N-1$개의 줄에 x_i, y_i, r_i, z_i가 공백을 사이에 두고 차례로 주어진다.

$N+1$번째 줄에 질의의 수 Q가 주어진다.

다음 Q개의 줄에 질의 a, b, e가 공백을 사이에 두고 차례로 주어진다.

출력

각 질의의 답을 순서대로 한 줄에 하나씩 출력한다.

제한

- $2 \le N \le 100\,000$
- $1 \le x_i, y_i \le N$ $(1 \le i \le N-1)$
- $1 \le r_i, z_i \le 10^9$ $(1 \le i \le N-1)$
- $1 \le Q \le 100\,000$
- $1 \le a, b \le N$, $a \ne b$
- $1 \le e \le 10^{18}$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 1 2 20 23 3 1 1 4 3 4 2 2 3 5 3 1 3 5 4 23 4 5 23 1 5 1	9 7 0

문제 F. 등수

시간 제한	3 초
메모리 제한	1024 MB

> 소프트웨어에 진심인 LG전자는 개발자들이 문제 해결 능력을 겨루고 지적 즐거움을 누릴 수 있도록 2012년부터 매년 LG Code Jam을 개최하고 있다.
>
> LG Code Jam에서 우수한 성적을 거둔 개발자는 코딩 전문가로 선발되어 비즈니스 문제 해결 및 개발자 교육과 같은 다양한 분야에 기여하고 있다.
>
> 특히, 신입 개발자라도 뛰어난 실력을 갖추면 전문가 타이틀을 획득할 수 있다.
>
> 이와 함께 LG전자는 2024년 LG 대학생 프로그래밍 경진대회를 개최하여 실력 있는 학생들에게도 지적 즐거움을 제공할 예정이다.

운 좋은 금성이는 2024년 LG 대학생 프로그래밍 경진대회에 참가하여 K등을 기록하는 꿈을 꾸고 있다. 이번 LG 대학생 프로그래밍 경진대회에는 총 N명이 참가할 것이다.

금성이는 자신이 K등을 기록했을 때 나올 수 있는 서로 다른 대회 결과의 경우의 수를 구하고 싶다. 두 대회의 결과가 서로 다르다는 것은 두 대회에서의 등수가 다른 참가자가 1명 이상 존재한다는 뜻이다.

단, 동점을 기록한 사람들은 같은 등수가 된다. 예를 들어, 공동 2등이 3명일 수 있고, 이 경우 다음 등수는 5등이 된다.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다.

다음 줄부터 각 테스트 케이스마다 N과 K의 값이 한 줄에 공백으로 구분되어 주어진다.

출력

각 테스트 케이스에 대해 문제의 정답을 10^9+7로 나눈 나머지를 한 줄에 출력한다.

제한

- $1 \le T \le 100\,000$
- $1 \le K \le N \le 5\,000$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 4 1 4 2 4 3 4 4	26 18 18 13

이 페이지는 공백입니다

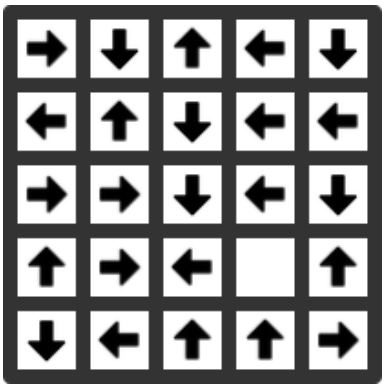
문제 G. 오직 5%의 사람들만이 이 문제를 풀 수 있습니다

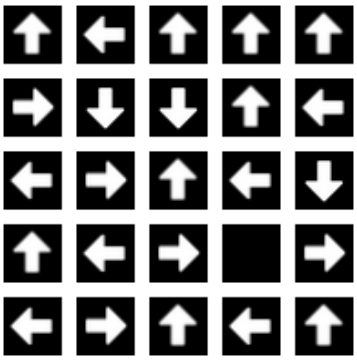
시간 제한	1 초
메모리 제한	1024 MB

이 문제는 인터랙티브 문제입니다.

이따금, 광고에서는 도발적인 내용과 제목을 넣어 사람들이 게임을 하도록 유도하는 경우가 있다. 코코아는 이런 도발에 매우 취약해서, 이런 광고가 나오면 반드시 해당 게임을 완벽히 정복해야 직성이 풀린다. 오늘도 어김없이 코코아는 '오직 5%의 사람들만이 이 문제를 풀 수 있습니다'라는 제목의 광고를 보았고, 이제 코코아는 이 게임을 완벽히 정복해야만 한다! 코코아가 하려는 게임은 다음과 같다.

게임은 직사각형 양면 격자판에서 진행된다. 격자판은 앞면과 뒷면에 상하좌우 중 한 방향을 가리키는 화살표가 그려져 있는 일반 칸과 양면 모두 아무것도 그려져 있지 않은 목표 칸으로 구성되어 있다. 처음에 모든 칸은 앞면으로 되어 있다.

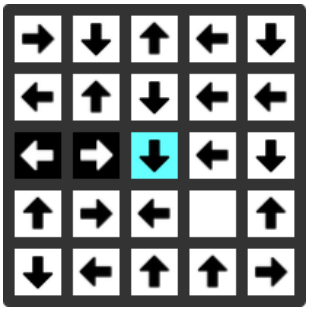
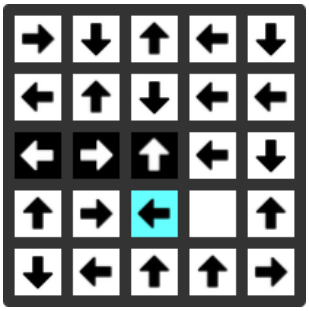
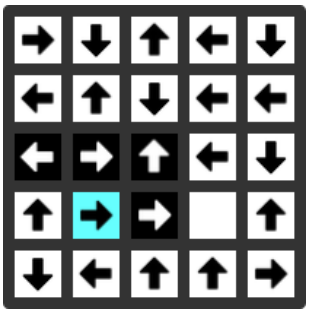
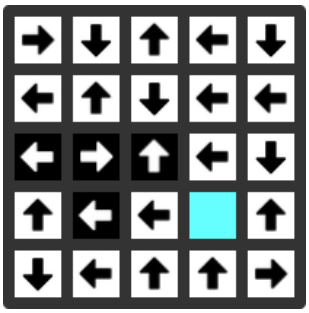




플레이어는 정해진 시작 칸에 공을 놓고 게임을 시작한다. 격자판에 있는 버튼을 누르면 공은 자신이 있는 칸에 있는 화살표가 가리키는 방향으로 굴러간다. 공이 어떤 칸 위에 있었다가 떠나면 그 칸은 위아래가 뒤집혀 앞면과 뒷면이 바뀐다.

만약 공이 굴러가던 방향과 다른 방향의 화살표가 있는 칸에 올라가거나 목표 칸에 올라가면 공은 그 칸에서 멈춘다.

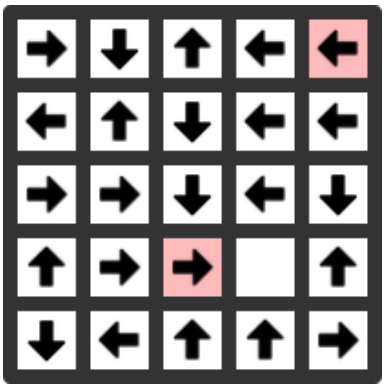
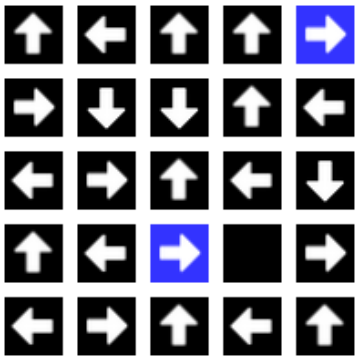
게임의 목적은 버튼을 눌러서 공을 목표 칸에 위치시키는 것이다. 공이 격자판 밖으로 떨어지면 게임은 실패이다.

위는 3행 1열에 공을 놓고 버튼을 4번 누르는 상황이다. 처음에 버튼을 누르면, 공이 3행 3열까지 이동하면서 3행 1열과 3행 2열에 있는 칸이 뒤집힌다. 그다음에 버튼을 한 번 더 누르면, 공이 한 칸 아래로 내려가면서 3행 3열에 있는 칸이 뒤집힌다. 그 후, 위 그림처럼 버튼을 두 번 더 누르면 공은 4행 4열에 있는 목표 칸에 도달한다.

코코아는 친구 치노의 깜짝 새해 선물로 $N \times M$ 크기 게임판을 만들어 주려고 한다. 하지만 코코아가 만들어 주려는 게임판은 조금 특별한 게임판이다. 바로 정확히 k번 버튼을 누른 후에 게임에서 이기는 게임판이다. 그런데 사소한 문제가 생겼다. k를 치노가 좋아하는 수로 정하고 싶은데 깜빡하고 k를 물어보지 않은 것이었다. 지금 와서 k를 물어보고 새로 게임판을 만든다면 깜짝 선물이 아니다!

그래서 코코아는 치노가 어떤 k를 좋아하더라도 대처할 수 있도록 게임판을 만들기로 했다. 코코아는 선물을 주기 직전에 치노에게 k를 물어본 후, 딱 20개 이하의 일반 칸에 그려진 화살표 방향을 순식간에 바꿔서 게임판을 만들려고 한다. 어떤 칸을 바꾸기로 했다면, 윗면과 아랫면에 그려진 화살표를 모두 바꿀 수 있다.

그러나 이런 게임판을 만드는 것은 화살표의 방향을 순식간에 수정하는 것보다 어려운 문제이다. 코코아는 고민을 해보았지만 마땅한 방안을 찾지 못하였다. 여러분이 코코아를 위해 초기 게임판을 만들고, k가 입력되었을 때 게임판을 수정하는 프로그램을 작성하자.

Interaction Protocol

당신의 프로그램은 아래의 과정을 통해 표준입력(stdin)과 표준출력(stdout)으로 채점기와 상호작용해야 한다.

우선, 당신의 프로그램은 두 개의 정수 N, M을 공백으로 구분하여 한 줄에 출력해야 한다. 이는 각각 격자판의 세로 길이와 가로 길이다.

이후, N개의 줄에 걸쳐 M개의 숫자를 공백 없이 출력한다. 각 숫자는 앞면의 상태를 나타낸다. 숫자가 0이면 해당 칸은 목표 칸이다. 그렇지 않으면 해당 칸은 일반 칸이고, 1이면 아래쪽, 2면 왼쪽, 3이면 위쪽, 4면 오른쪽 방향의 화살표가 그려져 있다. 이후, N개의 줄에 걸쳐 M개의 숫자를 공백없이 출력한다. 각 숫자는 뒷면의 상태를 나타내며, 앞면과 동일한 방식으로 표시한다.

앞면과 뒷면의 목표 칸의 위치는 동일해야 한다. 또한, 한 면 전체에서 목표 칸의 개수는 정확히 하나여야 한다.

이후, 시작 칸의 위치를 나타내는 정수 r, c를 공백으로 구분하여 한 줄에 출력해야 한다. 시작 칸은 r행 c열에 위치하게 된다. 맨 위쪽 행이 1행, 맨 왼쪽 열이 1열이다.

위의 사항들이 전부 출력되었다면 채점기는 양의 정수 k를 입력으로 준다. 당신의 프로그램은 프로그램은 k를 입력받고, 새로 바꾼 격자판을 출력해야 한다. 새로 바꾼 격자판은 앞면과 뒷면을 위와 동일한 방식으로 출력한다. 이때, 앞면과 뒷면 중 어느 하나라도 다른 칸의 개수는 20개를 넘을 수 없으며, 0의 위치는 기존의 출력과 동일해야 한다.

마지막으로 출력한 격자판에서 실제로 r행 c열부터 공을 굴리기 시작했을 때, 정확히 k번만에 공이 목표 칸에 도달해야 한다.

프로그램은 매 출력마다 표준 출력 버퍼를 비워야 한다. 출력 버퍼를 비우지 않을 시 예상치 못한 결과를 받을 수 있다. 언어별로 표준 출력 버퍼를 flush하는 방법은 다음과 같다.

- C: `fflush(stdout)`
- C++: `std::cout << std::flush`
- Java: `System.out.flush()`
- Python: `sys.stdout.flush()`

제한

- $1 \le N \le 100$
- $1 \le M \le 100$
- $1 \le r \le N$
- $1 \le c \le M$
- $1 \le k \le 10^6$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4	5 5 41321 23142 44121 34203 12334 32333 41132 24321 32404 24323 3 1 41322 23142 44121 34403 12334 32334 41132 24321 32404 24323

설명

해당 예제의 입출력의 빈 줄은 입출력이 어떤 방식으로 이루어지는지 이해를 돕기 위해 의도적으로 추가된 것이며, 실제 입출력에는 빈 줄이 나타나지 않는다.