UCPC 2024

전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2024

Finals

Official Problemset

본선 문제

2024년 8월 3일, 11:00 → 16:00

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원

STARTLINK SOLVED.AC LG전자 NEXON

HYUNDAI AutoEver MOLOCO SAMSUNG SOFTWARE MEMBERSHIP 박승원 (veydpz)

김동현 (kdh9949) 정재헌 (Gravekper) pjshwa 임지환 (raararaara)

kclee2172 이종서 (leejseo) 이상헌 (evenharder) 김준겸 (ryute)

UCPC 2024

Finals — 본선대회

문제 목록

문제지에 있는 문제가 총 13문제가 맞는지 확인하시기 바랍니다.

A 관광 코스

B Rolling Rick

C 배고픈 무토를 위한 피자 만들기

D SoleMap

E 돌 놓기 게임

F 4색 정리

G $\prod_{i=1}^{N}(R_i - L_i + 1)$개의 트리

H 감옥

I 러시안 회전초밥

J Distance Sum Maximization

K 스레드

L 밤양갱

M 지루함 줄이기

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 관광 코스

시간 제한 2 초 메모리 제한 1024 MB

UCPC 왕국에는 왕국 전체를 둘러볼 수 있는 N개의 구간으로 이루어진 원형 관광 코스가 있다. 각 구간에는 다음 구간으로 갈 수 있는 셔틀버스가 하나 있으며, $1 \le i < N$에 대해 i번째 구간에서는 $i+1$번째 구간으로, N번째 구간에서는 1번째 구간으로 이동할 수 있다.

이제 북극에서 온 N명의 관광객이 관광 코스를 이용해서 UCPC 왕국을 둘러볼 예정이다. i번째 관광객은 i번째 구간부터 시작해서 셔틀버스를 타고 총 N개의 구간을 관광한다.

각 구간은 설원과 사막 중 하나이다. 각 관광객은 호감도 1을 가지고 시작 지점부터 관광을 시작하며, 설원 구간을 지날 때마다 호감도가 1 증가하고 사막 구간을 지날 때마다 호감도가 1 감소한다. 각 관광객은 관광 도중 호감도가 0이 되는 즉시 관광을 중지하고 자신의 나라로 떠나버린다. 관광 코스의 N개의 구간을 모두 둘러본 뒤 호감도가 1 이상이라면 그 관광객은 UCPC 왕국의 비싼 기념품을 구매하고 자신의 나라로 돌아간다.

북극에 살고 있는 당신은 각 관광객의 기념품 구매 여부를 알고 있고, 이 정보를 활용하여 UCPC 왕국의 관광 코스의 구조를 알아내야 한다. 1번부터 N번까지 관광객의 기념품 구매 여부가 주어졌을 때 가능한 관광 코스의 구조 중 하나를 출력해 보자.

입력

첫 줄에 관광 코스 구간의 수인 N이 주어진다. $(1 \le N \le 500\,000)$

둘째 줄에 i번째 관광객의 기념품 구매 여부를 나타내는 길이 N의 문자열이 주어진다. i번째 문자는 i번 관광객의 기념품 구매 여부를 나타내며, 기념품을 구매했다면 `O`, 구매하지 않았다면 `X`이다.

출력

주어진 입력으로 가능한 UCPC 왕국의 관광 코스가 존재한다면, 첫 줄에 `YES`를 출력하고 둘째 줄에 길이 N의 문자열을 출력한다. i번째 문자에는 i번째 구간이 설원이라면 `+`, 사막이라면 `-`를 출력한다.

주어진 입력으로 가능한 관광 코스가 존재하지 않는다면 첫 줄에 `NO`를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 0X0X0	YES +-+-+
6 XXXXXX	YES +--+--
5 XXX0X	NO

문제 B. Rolling Rick

시간 제한 8 초 메모리 제한 1024 MB

책상 위에 가로 W, 세로 H인 직사각형 모양 종이가 있고, 종이 위에 가로 A, 세로 B, 높이 C인 직육면체 Rick이 놓여있다. 처음에는 Rick의 바닥면 왼쪽과 위쪽 변이 종이의 왼쪽, 위쪽 변과 겹쳐있다.

우리는 Rick을 오른쪽 또는 아래쪽으로 굴릴 수 있다. Rick을 굴릴 때에는 Rick의 바닥면 오른쪽 또는 아래쪽 모서리를 고정시킨 채로 오른쪽 또는 아래쪽으로 회전시킨다. Rick의 모든 표면에 마르지 않는 페인트가 묻어 있어서 Rick과 닿은 바닥 영역에 페인트가 묻는데, 책상을 깨끗이 써야 하므로 Rick이 종이를 벗어나도록 굴리면 안 된다.

Rick을 적절히 굴려서 Rick의 바닥면 오른쪽과 아래쪽 변이 종이의 오른쪽, 아래쪽 변과 겹치도록 옮길 때, 종이에 페인트가 묻는 영역의 넓이를 최대화하여라.

입력

첫 줄에 Rick의 크기 A, B, C와 모눈종이의 크기 W, H가 공백을 사이에 두고 주어진다. ($1 \le A, B, C, W, H \le 1\,000\,000$; $A < W$; $B < H$)

모든 입력 값은 정수이다.

출력

만약 Rick을 오른쪽 아래로 옮길 수 없다면 첫 줄에 `-1`만을 출력한다.

그렇지 않으면 첫 줄에 종이에 페인트가 묻는 영역 넓이의 최댓값을 출력하고, 둘째 줄에 Rick을 굴리는 방법을 `R`과 `D`만으로 이루어진 문자열로 출력한다. i번째 글자가 `R`이면 i번째에 Rick을 오른쪽으로, `D`이면 아래쪽으로 굴리는 것을 의미한다.

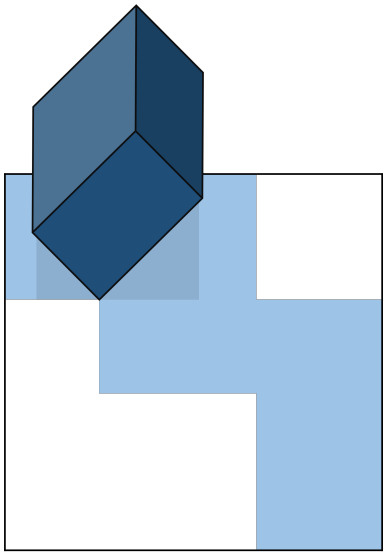
가능한 방법이 여러 가지라면 그중 아무거나 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 1 1 24 10	33 RRRRRRRRRRRRRRRRRRRRRRRDDDDDDDDDD
3 4 5 6 7	-1
3 4 5 12 12	79 RDRD

노트

아래 그림은 3번째 예제에서 Rick을 굴리는 그림이다.



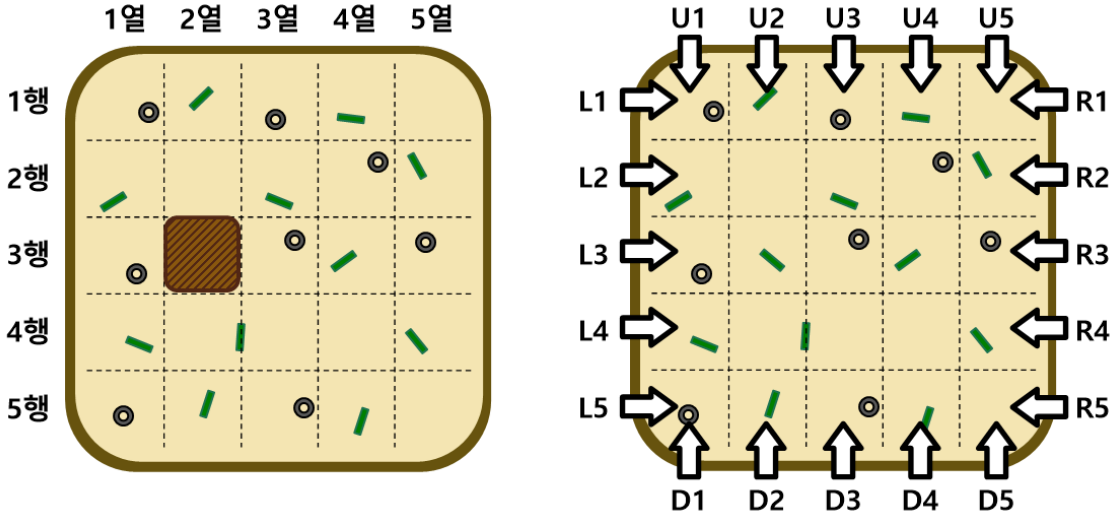
이 예제에서 Rick을 아래쪽, 오른쪽, 아래쪽, 오른쪽 순서대로 굴릴 수 있으나 페인트가 묻는 영역의 넓이가 74로 최적이 아니다.

문제 C. 배고픈 무토를 위한 피자 만들기

시간 제한 1초 메모리 제한 1024 MB

키네시스는 검은 마법사를 토벌하기 위해 츄츄 아일랜드를 탐험하던 중 배고픈 무토를 마주쳤다. 무토를 위해 매일 밥을 주고 있던 시미아는 밥투정하고 있는 무토를 위한 대형 피자를 만들기 위해 키네시스에게 도움을 요청했다. 큰 덩치를 가진 무토는 검은 마법사한테 가는 길을 막고 있기 때문에 키네시스는 어쩔 수 없이 시미아의 부탁을 들어주기로 했다.

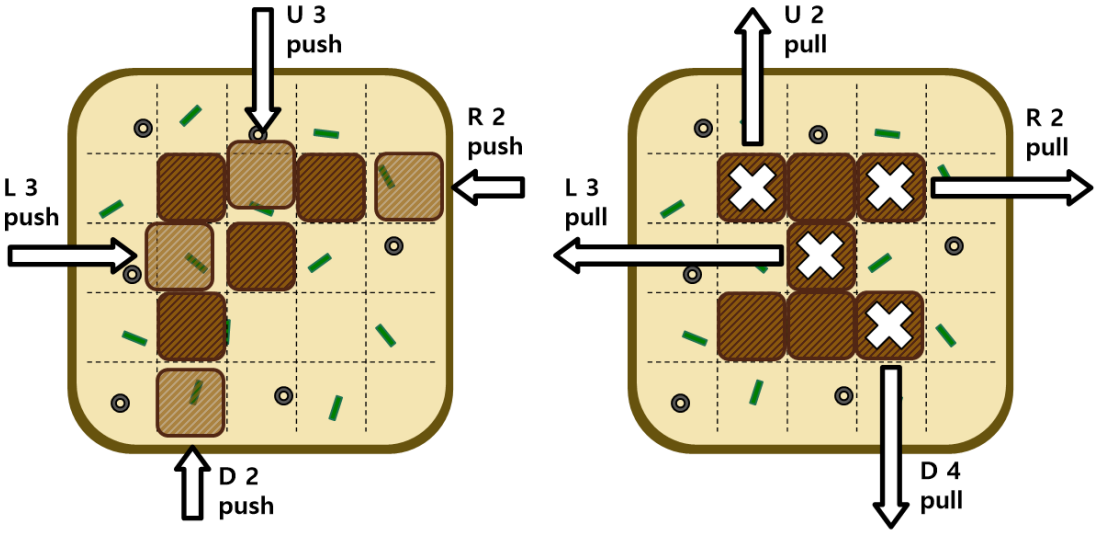
무토가 원하는 커다란 피자는 $N \times N$ 크기의 격자 모양이며, 피자의 가장 왼쪽 위 칸은 1행 1열이고, 가장 오른쪽 아래 칸은 N행 N열이다. 시미아가 준 레시피에는 피자에 올려야 하는 미트볼의 배치가 그려져 있으며, 키네시스는 레시피에 그려진 대로 미트볼을 토핑해야 한다. 처음에는 피자의 R행 C열에 시미아가 먼저 토핑한 미트볼이 하나 놓여 있다. 미트볼의 모양은 격자의 칸과 정확히 일치하며 격자 한 칸에 두 개 이상의 미트볼이 들어갈 수 없다.



$N = 5, R = 3, C = 2$일 때 피자의 상태와 피자 밖에서 염동력을 사용할 수 있는 위치

피자를 밟으면 먹을 수 없게 되어 버리기 때문에 키네시스는 피자 밖에서 염동력을 사용하여 피자를 완성해야 한다. 염동력을 1회 사용하는 과정은 아래와 같다.

1. 염동력을 사용할 위치를 정한다. 가능한 위치는 피자의 위쪽, 아래쪽, 왼쪽, 오른쪽 중 하나이다. 염동력은 키네시스가 위치한 곳에서 시작하여 위쪽이면 행이 증가하는 방향으로, 아래쪽이면 행이 감소하는 방향으로, 왼쪽이면 열이 증가하는 방향으로, 오른쪽이면 열이 감소하는 방향으로 사용한다.
2. 염동력을 사용할 행 또는 열의 번호를 정한다. 염동력은 행 또는 열과 평행하도록 사용해야 하며, 두 개 이상의 행 또는 열에 걸쳐서 사용할 수 없다.
3. 염동력을 사용한다. 사용할 수 있는 염동력은 두 종류이다.
    - 밀어넣기(push): 미트볼을 밀어 넣어서 처음 부딪힌 미트볼의 바로 앞 칸에 놓는다. 만약 밀어 넣기 전 염동력을 사용한 방향의 첫 번째 칸에 미트볼이 있거나, 그 방향에 미트볼이 없어서 밀어 넣은 미트볼이 피자 안에 위치할 수 없다면 밀어 넣은 미트볼은 사라진다.
    - 당기기(pull): 염동력에 처음 맞은 미트볼을 피자 밖으로 꺼낸다. 염동력을 사용한 방향에 미트볼이 없다면 아무 일도 일어나지 않는다.



밀어넣기(push), 당기기(pull) 염동력을 사용하는 모습

시미아의 레시피대로 피자를 토핑했다면 무토가 피자를 맛있게 먹고 다음 지역으로 갈 수 있는 길을 열어줄 것이다. 하지만 검은 마법사가 언제 세상을 파멸시킬지 모르기 때문에 키네시스는 피자를 만드는 데에 시간을 지체할 수 없다.

염동력을 $2N^2$번 이하로 사용하여 피자를 완성하는 방법 중 하나를 출력해 보자.

입력

첫 줄에 피자의 크기 N이 주어진다. $(3 \le N \le 50)$

둘째 줄에 시미아가 처음에 토핑한 미트볼의 행과 열을 의미하는 정수 R, C가 공백을 사이에 두고 주어진다. $(1 \le R, C \le N)$

이후 N개의 줄에 걸쳐 시미아의 레시피대로 그려진 미트볼 배치가 주어진다. 각 줄에는 길이가 N인 문자열이 주어진다. 레시피의 i번째 줄의 j번째 문자는 i행 j열의 미트볼 여부를 의미하며 `.`는 빈 칸, `#`는 미트볼이 놓인 칸을 의미한다.

출력

만약 염동력을 $2N^2$번 이하로 사용하여 피자를 완성할 수 있다면 첫 줄에 키네시스가 사용한 염동력의 횟수 M을 출력한다. 단, M이 최소일 필요는 없다. $(0 \le M \le 2N^2)$

M이 1 이상이라면 둘째 줄부터 M개의 줄에 걸쳐 키네시스가 수행한 행동을 한 줄에 하나씩 출력한다. 각 줄에는 염동력을 사용한 위치 (위쪽: `U`, 아래쪽: `D`, 왼쪽: `L`, 오른쪽: `R`), 행 또는 열의 번호 X $(1 \le X \le N)$, 염동력의 종류(밀어넣기는 `push`, 당기기는 `pull`)을 공백으로 구분하여 출력한다.

만약 염동력을 $2N^2$번 이하로 사용하여 피자를 완성할 수 없다면 첫 줄에 `-1`만을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 2 2 .#. ##. ...	2 U 2 push L 2 push
3 1 2 ... #.# .#.	6 D 2 push D 2 push L 2 push R 2 push U 2 pull U 2 pull
4 1 1	1 L 1 pull

문제 D. SoleMap

시간 제한 2 초 메모리 제한 1024 MB

HYUNDAI AutoEver

모든 지도는 '단 하나'(Sole)로 통한다, 현대오토에버 차세대 내비게이션 지도 'SoleMap'

처음 가는 길을 운전할 때, 갈림길을 잘못 들거나 차선을 잘못 타서 곤란해하는 사람을 지켜본 경험이 누구나 한 번쯤은 있을 겁니다. 내비게이션이 있어도 초행길은 내비게이션 화면과 실제 도로를 머릿속에서 바로 대입하기 어려운 경우가 많습니다. SoleMap은 이런 운전자들의 불편을 해소하기 위해 현대오토에버가 구축 중인 차세대 내비게이션 지도입니다. SoleMap은 '단 하나의'라는 뜻을 가진 형용사 Sole에 지도(Map)를 더해 통합의 의미를 강조한 이름입니다. SoleMap은 2024년 안에 실제 내비게이션 탑재를 목표로 하고 있습니다.

직선나라의 도로 구조는 N개의 도시가 일직선으로 늘어서 있고, 모든 $1 \le i \le (N-1)$에 대해 i번 도시와 $(i+1)$번 도시가 양방향 w_i-차선 도로로 직접 연결되어 있는 형태로 생각할 수 있습니다. 이러한 직선나라에는 매일같이 u_j번 도시에서 v_j번 도시까지를 이동하는 차량 x_j대가 있습니다. 따라서 직선나라에는 무려 $\sum_j x_j$대의 차량이 매일 이동합니다. 직선나라의 이동 경로는 유일하지만, 어떤 차로를 이용하느냐에 따라 더 빨리 이동하거나 차량 정체 때문에 더 느리게 이동할 수 있습니다. 그래서 차도 구분 없이 경로만 찾아 주는 기존의 내비게이션은 큰 도움이 되지 못했습니다.

직선나라의 대통령 키파는 기존의 내비게이션과 확연히 구별되는 SoleMap을 시범 도입하였습니다. 그러자 내비게이션이 차로 단위로 가장 빠른 길을 잘 알려준다는 소식은 입소문을 타게 되었고, 결국 직선나라의 모든 내비게이션은 SoleMap이 되었습니다. SoleMap 때문에 갑자기 자차 이용자가 많아져 도로가 무너지진 않을지 걱정되었던 키파는 현대오토에버에게 '도로 부담'이라는 값을 계산하도록 지시합니다.

다행히도 키파는 대통령을 하기 전 수학과 공학을 깊이 공부했기 때문에, 실무자가 직접 유의미한 지표를 만들어 내기 위해 머리를 싸매도 되지 않도록 도로 부담을 엄밀하게 정의해 주었습니다. 각 도로에 대해 **도로 부담**은, 매일 그 도로를 이용하는 차량의 수 c에 대해 c대의 차량을 w개의 차로가 적절히 분담했을 때, 각 차로를 지나는 차량 대수의 제곱의 합의 최솟값입니다.

예를 들어 어떤 도로에 대해 $c = 4$, $w = 3$인 경우, 다음과 같이 차로가 차량을 분담할 수 있습니다:

- 한 차로에 4대의 차량이 모두 다니는 경우, $4^2 + 0^2 + 0^2 = 16$
- 한 차로에 3대의 차량이, 다른 한 차로에 나머지 1대의 차량이 다니는 경우, $3^2 + 1^2 + 0^2 = 10$
- 한 차로에 2대의 차량이, 다른 한 차로에 나머지 2대의 차량이 다니는 경우, $2^2 + 2^2 + 0^2 = 8$
- 한 차로에 2대의 차량이, 다른 한 차로에 1대의 차량이, 나머지 한 차로에 나머지 1대의 차량이 다니는 경우, $2^2 + 1^2 + 1^2 = 6$

이중 최솟값인 6이 도로 부담이 됩니다.

SoleMap의 프로그래머인 당신이, 직선나라의 교통 상황이 주어지면 각 도로의 도로 부담을 계산해 승진의 기회를 노려 봅시다!

입력

첫 줄에 직선나라의 도시의 수 N과 직선나라를 이동하는 차량 정보를 나타내는 정수 M이 공백을 사이에 두고 주어집니다. $(2 \le N \le 500\,000;\ 1 \le M \le 500\,000)$

둘째 줄에 $(N-1)$개의 정수 $w_1, w_2, \cdots, w_{N-1}$이 공백을 사이에 두고 주어집니다. $(1 \le w_i \le 10^9)$ 이는 각 $1 \le i \le (N-1)$에 대해, i번 도시와 $(i+1)$번 도시를 잇는 도로는 w_i-차로라는 뜻입니다.

다음 M개의 줄에 직선나라의 교통 상황 정보가 주어집니다. 각 $1 \le j \le M$에 대해, $(j+2)$번째 줄에는 u_j, v_j, x_j가 공백을 사이에 두고 주어집니다. $(1 \le u_j < v_j \le N;\ 1 \le x_j \le 10^9)$ 이는 매일 u_j번 도시에서 v_j번 도시를 다니는 차량이 x_j대 있다는 뜻입니다.

주어지는 모든 x_j의 합은 10^9을 넘지 않습니다.

출력

$(N-1)$개의 줄을 출력합니다.

각 $1 \le i \le (N-1)$에 대해, i번째 줄에는 i번 도시와 $(i+1)$번 도시를 잇는 도로의 도로 부담을 출력합니다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2 1 3 4 1 3 3 2 4 1	9 6 1

노트

다음과 같이 매일 도로를 이용하는 차량의 수를 계산할 수 있습니다.

- 1번 도시와 2번 도시를 잇는 도로: $3+0=3$대
- 2번 도시와 3번 도시를 잇는 도로: $3+1=4$대
- 3번 도시와 4번 도시를 잇는 도로: $0+1=1$대

다음과 같이 각 도로의 도로 부담을 계산할 수 있습니다.

- 1번 도시와 2번 도시를 잇는 도로의 도로 부담: 차선이 하나밖에 없으므로 $3^2=9$
- 2번 도시와 3번 도시를 잇는 도로의 도로 부담: 앞서 설명한 대로 6
- 3번 도시와 4번 도시를 잇는 도로의 도로 부담: 차가 한 대밖에 없으므로, 어느 차선으로 달리든 $1^2+0^2+0^2+0^2=1$

문제 E. 돌 놓기 게임

시간 제한 1초 메모리 제한 1024 MB

N개의 칸이 원형으로 배치된 게임판을 갖고 철수와 영희가 게임을 하려고 한다. i번째 칸에는 음이 아닌 정수 x_i가 쓰여 있다. 각 칸에는 돌을 최대 하나씩 놓을 수 있다.

철수는 검은 돌을 N개 갖고 있고, 영희는 흰 돌을 N개 갖고 있다. 먼저 두 사람이 정해진 몇 개의 칸에 돌을 놓는다. 이후 철수부터 시작해서 번갈아 가며 턴을 진행한다. 자신의 턴에는 자신의 돌이 놓인 칸과 인접한 빈칸이 있을 때, 그중 하나를 골라 자신의 돌을 놓는다. 그러한 칸이 없다면 그대로 상대의 턴으로 넘어간다. 아무도 돌을 놓을 수 없는 경우 게임이 종료된다.

철수와 영희는 각자 자신의 돌이 놓인 칸의 점수의 합을 최대화하려고 한다. 턴을 진행하기 전의 돌의 배치가 주어지고 두 사람 모두가 최적의 수를 두었을 때 각자의 점수를 구하여라.

입력

첫 줄에 N이 주어진다. $(3 \le N \le 200\,000)$

둘째 줄에 N개 칸의 돌 배치를 나타내는 정수 $c_1, \cdots, c_N$이 공백을 사이에 두고 주어진다. $(0 \le c_i \le 2)$ $c_i = 1$이면 i번째 칸에 검은 돌이 놓여 있음을, $c_i = 2$이면 흰 돌이 놓여 있음을, $c_i = 0$이면 어떤 색의 돌도 놓여 있지 않음을 의미한다.

셋째 줄에 N개 칸에 적힌 정수 $x_1, \cdots, x_N$이 공백을 사이에 두고 주어진다. $(0 \le x_i \le 10^9)$

출력

첫 줄에 철수와 영희가 얻게 되는 각자의 점수를 공백을 사이에 두고 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
9 0 1 0 2 0 0 2 0 0 1 2 3 4 5 6 7 8 9	12 33
4 0 0 0 0 6 9 8 8	0 0
8 1 0 0 0 0 0 0 1 2 9 4 8 1 8 5 0	37 0
36 1 1 0 0 2 2 0 1 0 0 0 0 2 0 0 0 1 0 0 0 1 0 0 2 0 0 0 2 0 0 0 1 0 0 0 1 18 23 18 20 40 30 19 15 13 11 19 21 12 25 43 37 23 21 10 4 9 7 3 60 54 32 18 39 42 55 71 92 4 2 40 1	493 458

* 예제 4번의 경우 입력이 가로로 길어 문제지에서는 줄을 나누어 표시합니다. 정확한 예제는 플랫폼을 참고하시기 바랍니다.

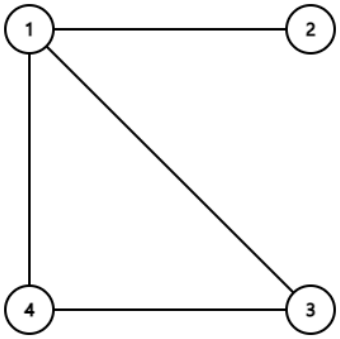
문제 F. 4색 정리

시간 제한 4 초 메모리 제한 1024 MB

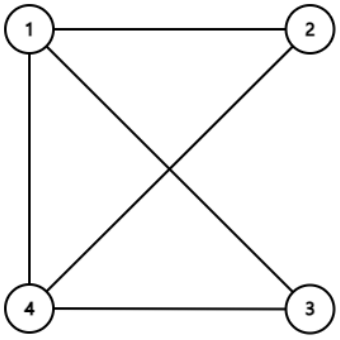
아래 조건을 만족하는 그래프 G가 주어진다.

- G의 정점은 N개, 간선은 M개이며, 정점에는 1번부터 N번까지 정수 번호가 매겨져 있다.
- 다음 과정으로 그려진 그림의 선분은 원의 둘레를 제외한 원의 내부에서 교차하지 않는다.
  1. 원을 그리고, 원 위에 일정한 간격으로 N개의 점을 찍는다. 이 점을 순서대로 $A_1, \cdots, A_N$이라 하자.
  2. G에서 u번 정점과 v번 정점이 연결되어 있다면, 점 A_u와 점 A_v를 선분으로 잇는다.

예를 들어, 다음은 $N = 4$, $M = 4$일 때 조건을 만족하는 그래프이다.



그러나, 다음은 조건을 만족하는 그래프가 아니다.



그래프 G에 대해, G를 **색칠**한다는 것은 간선으로 연결된 두 정점의 색이 다르도록 모든 정점에 색을 부여하는 것을 의미한다.

입력 조건을 만족하는 임의의 그래프는 4개의 색으로 **색칠**하는 것이 항상 가능하다.

색칠에 사용하는 서로 다른 4개의 색을 각각 1, 2, 3, 4라고 하자.

4 이하의 양의 정수로 이루어진 서로 다른 K개의 순서쌍 (c_j, d_j)가 주어질 때, 색이 c_j인 정점과 d_j인 정점을 연결하는 간선이 존재하지 않도록 G를 **색칠**하여라.

입력

첫 줄에 G의 점의 개수 N, 간선의 개수 M, 순서쌍의 개수 K가 공백을 사이에 두고 주어진다. ($3 \le N \le 200000$; $1 \le M \le 400000$; $0 \le K \le 6$)

$1 \le i \le M$에 대해, $(i+1)$번째 줄에는 간선에 대한 정보 u_i, v_i가 공백을 사이에 두고 주어진다. ($1 \le u_i < v_i \le N$; $1 \le i_1 < i_2 \le M$이면 $(u_{i_1}, v_{i_1}) \ne (u_{i_1}, v_{i_2})$) 이는 그래프 G의 u_i번 정점과 v_i번 정점이 연결되어 있음을 의미한다.

$1 \le j \le K$에 대해, $(M+j+1)$번째 줄에는 c_j, d_j가 주어진다. ($1 \le c_j < d_j \le 4$; $1 \le j_1 < j_2 \le K$이면 $(c_{j_1}, d_{j_1}) \ne (c_{j_2}, d_{j_2})$)

출력

조건을 만족하도록 **색칠**할 수 없다면 첫 줄에 **-1**만을 출력한다.

조건을 만족하도록 **색칠**할 수 있다면 첫 줄에 N개의 정점의 색을 공백을 사이에 두고 순서대로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 5 1 1 2 2 3 3 4 1 4 2 4 2 3	2 4 2 1
3 3 5 1 2 2 3 1 3 1 3 1 4 2 3 2 4 3 4	-1

문제 G. $\prod_{i=1}^{N}(R_i - L_i + 1)$ 개의 트리

시간 제한 2 초 메모리 제한 1024 MB

1번 정점을 루트로 하는 정점 N개의 트리가 주어진다. 우리는 각 정점에 쓰일 음이 아닌 정수 a_i를 정할 것이다.
이때 a_i들은 다음 조건을 만족해야 한다. 모든 $1 \le i \le N$에 대해서,

- $a_i \le p_i$
- i번 서브트리 내부의 정점 j에 대해서 a_j를 모두 더한 값을 S_i라고 했을 때, $S_i \ge q_i$

위의 조건을 만족하는 $(a_1, a_2, \ldots, a_N)$에 대해 $\sum_{i=1}^{N} c_i a_i$의 최솟값을 $f(c_1, c_2, \ldots, c_N)$으로 정의하자.
다음 값을 998244353으로 나눈 나머지를 구하여라.

$$\sum_{c_1=L_1}^{R_1} \sum_{c_2=L_2}^{R_2} \cdots \sum_{c_N=L_N}^{R_N} f(c_1, c_2, \cdots, c_N)$$

입력

첫 줄에 정점의 개수 N이 주어진다. $(1 \le N \le 250)$

둘째 줄부터 $(N-1)$개의 줄에 걸쳐 트리 간선에 대한 정보가 주어진다. 이중 i번째 줄에는 두 개의 정수 s_i, e_i가 공백으로 구분되어 주어진다. $(1 \le s_i, e_i \le N)$ 이는 s_i와 e_i를 잇는 간선이 존재한다는 것이다.

$1 \le i \le N$에 대해, $(N+i)$번째 줄에는 p_i, q_i, L_i, R_i가 공백을 사이에 두고 주어진다. $(1 \le L_i \le R_i \le 250;\ 0 \le p_i, q_i \le 250;\ \sum_{i=1}^{N} p_i \le 250)$

주어진 그래프는 트리를 이루며, 주어진 조건을 만족하는 $(a_1, a_2, \ldots, a_N)$이 존재하는 입력만이 주어진다.

출력

문제에 주어진 값을 998244353으로 나눈 나머지를 출력한다. $998244353 = 119 \times 2^{23} + 1$은 소수이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 2 1 3 1 4 2 5 5 5 1 1 2 2 2 1 3 3 1 1 1 1	14
6 1 2 1 3 2 4 2 5 4 6 1 7 1 3 3 2 2 4 2 1 1 4 4 2 4 6 2 0 1 5 2 0 2 5	39072

노트

j번 정점이 i번 서브트리 내부에 포함된다는 것은 $j = i$이거나 i번 정점이 j번 정점의 조상이라는 것이다.

문제 H. 감옥

시간 제한 1초 메모리 제한 1024 MB

무한한 영역이 감옥이기 때문에 절대로 탈옥할 수 없어 악명 높은 좌표평면 감옥에 Q명의 죄수가 갇혀 있다.

감옥을 감시하는 교도관은 원점에 서 있으며, 죄수들을 볼 수 있는 시야가 있다. 교도관의 시야는 다음 조건을 만족함이 알려져 있다.

- 교도관의 시야는 꼭짓점이 N개인 단순다각형으로 둘러싸여 있으며, 단순다각형의 선분을 포함한 내부는 시야 내부, 선분을 제외한 외부는 시야 외부이다.
- 교도관이 특정 지점을 볼 수 있다면, 그 방향으로 그보다 더 가까운 곳도 볼 수 있다. 엄밀히 말해, 시야 내부의 임의의 점과 원점을 잇는 선분 위의 모든 점은 시야 내부에 있다.
- 원점 주변은 교도관이 언제나 볼 수 있다. 엄밀히 말해, 원점을 중심으로 하고 반지름이 ε인 원이 교도관의 시야 내부에 포함되는 $\varepsilon > 0$이 존재한다.

이 감옥에 갇힌 Q명의 죄수들이, 조금의 자유라도 쟁취하기 위해 힘을 모아 교도관의 시야에서 벗어나기 위해 시도하고 있다. 시야를 더욱 효과적으로 벗어나기 위해, 모든 $2 \le i \le Q$에 대해 i번 죄수가 $(i-1)$번 죄수가 시야 내에 있는지 여부에 따라 다음과 같이 순서대로 움직인다.

- $(i-1)$번 죄수가 이동한 후 시야 내부에 있었다면, i번 죄수는 $(i-1)$번 죄수가 있는 방향의 반대 방향을 본 뒤 둘 사이의 거리만큼 이동한다.
- $(i-1)$번 죄수가 이동한 후 시야 외부에 있었다면, i번 죄수는 $(i-1)$번 죄수가 있는 방향을 바라본 뒤, 둘 사이 거리의 절반만큼 이동한다. 단, 정수 격자점이 아닌 점은 오래 있기 불편하게 설계되었으므로, 이동한 후 가장 가까운 정수 격자점 중 원점과 가장 가까운 점으로 이동한다.

자유를 중시하는 당신이 교도관의 시야 정보를 소식통으로부터 입수했고, 이를 이용해서 죄수들에게 시야 안에 있는지 밖에 있는지를 알려주고자 한다. 현재 죄수들의 위치가 주어지면, 이동 후 각 죄수가 시야 내부에 있는지 외부에 있는지를 판별하는 프로그램을 작성하여라.

입력

첫 줄에 N과 Q가 공백을 사이에 두고 주어진다. $(3 \le N \le 100\,000;\ 1 \le Q \le 100\,000)$

둘째 줄부터 N개의 줄에 걸쳐 시야를 나타내는 단순다각형의 정보가 주어진다. 모든 $1 \le i \le N$에 대해, $(i+1)$번째 줄에는 x_i와 y_i가 공백을 사이에 두고 주어진다. $(-1\,000\,000 \le x_i, y_i \le 1\,000\,000)$ 이는 다각형의 반시계방향으로 i번째 점이 (x_i, y_i)라는 의미이다.

$(N+2)$번째 줄부터 Q개의 줄에 걸쳐 각 죄수의 정보가 주어진다. 모든 $1 \le j \le N$에 대해 $(N+j+1)$번째 줄에는 u_j와 v_j가 공백을 사이에 두고 주어진다. $(-1\,000\,000 \le u_j, v_j \le 1\,000\,000)$ 이는 j번 죄수의 처음 위치가 (u_j, v_j)라는 의미이다.

주어지는 모든 좌표는 정수이며, 주어지는 다각형은 문제에서 설명하는 시야 조건을 만족한다. 주어지는 다각형의 연속한 세 점은 한 직선 위에 있지 않다.

출력

총 Q개의 줄을 출력한다. 모든 $1 \le i \le Q$에 대해, i번째 죄수가 시야 내부에 있다면 `1`, 아니면 `0`을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 3 -1 -1 9 -1 -1 9 2 2 -2 3 8 0	1 0 1
6 5 0 -2 3 -10 14 -3 5 0 10 10 -5 5 0 0 -2 0 6 4 5 -5 -3 11	1 0 1 0 1

문제 I. 러시안 회전초밥

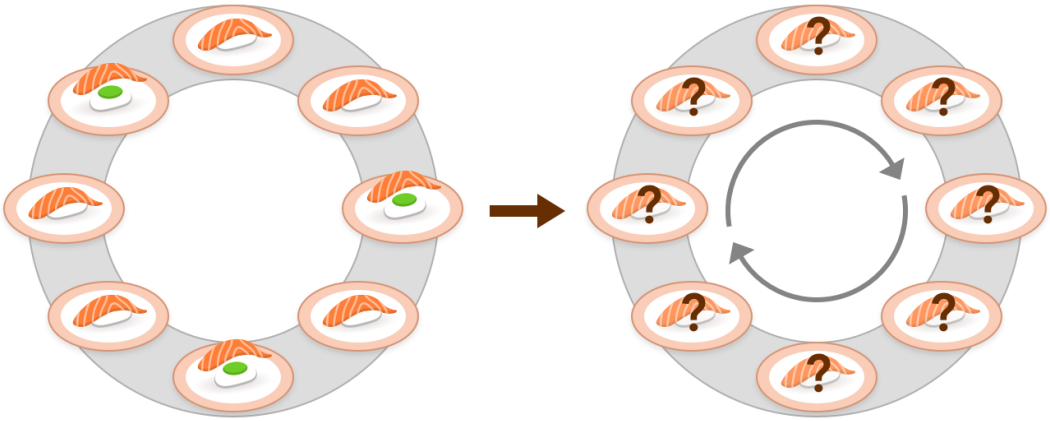
시간 제한 3 초 메모리 제한 1024 MB

진우는 세계 최고의 도박사이자 미식가이다. 진우는 좋아하는 아이돌 그룹의 사장인 도현이 "러시안 회전초밥"이라는 음식점을 새롭게 개업했다는 소식을 듣고 한 걸음에 달려갔다.

러시안 회전초밥의 대표 메뉴는, 당연하게도, 러시안 회전초밥이다. 이 메뉴를 주문하면 N개의 초밥과 함께 챌린지가 주어지는데, 챌린지에 성공한 도전자는 돈을 내지 않아도 된다. 챌린지의 목표는 한 번도 표정이 변하지 않고 K개의 초밥을 먹는 것이다. 이 챌린지가 어려운 이유는 몇몇 초밥에 매운 와사비가 듬뿍 들어가기 때문이다!

챌린지는 다음과 같이 진행된다. 먼저 도현은 원형 컨베이어 벨트 위에 일정한 간격으로 N개의 초밥을 배치한다. 도현은 도전자가 보는 앞에서 초밥 몇 개에 와사비를 넣어서 그 위치를 알 수 있게 한다. 와사비 초밥을 포함해서 모든 초밥은 생김새가 동일하여 구분할 수 없다.

그 다음 도전자는 눈을 가리고, 도현은 컨베이어 벨트를 무작위로 회전시킨다. 도전자가 다시 눈을 뜨면 컨베이어 벨트가 시계 방향으로 돌아가기 시작한다. 이제부터 도전자는 자신의 앞에 초밥이 놓일 때마다 즉시 그 초밥을 먹어야 한다. 즉, 도전자는 눈을 뜬 순간 앞에 놓인 초밥부터 **반시계 방향으로 연속한 초밥을 먹게 된다.**



도현은 더 많은 사람들에게 기회를 주기 위해 초밥 건너뛰기 쿠폰을 판매하고 있다. 도전자는 눈을 가리기 전에 쿠폰을 원하는 만큼 살 수 있다. 도전자가 쿠폰을 사용하면 앞에 놓인 초밥 하나를 먹지 않고 건너뛸 수 있다. 이렇게 건너뛴 초밥은 컨베이어 벨트에서 제거되며, 도현이 확인해서 와사비가 들었는지 알려준다.

도전자가 와사비 초밥을 먹고 표정이 변하거나, 초밥을 너무 많이 건너뛰어서 K개의 초밥을 먹지 못하면 챌린지에 실패한다.

진우는 러시안 회전초밥 챌린지에 도전하려고 한다. 안타깝게도 진우는 매운 음식을 못 먹기 때문에 와사비 초밥은 피해야 한다. 진우는 세계 최고의 도박사이자 미식가라는 명성을 잃고 싶지 않으므로, **어떤 경우에도 챌린지에 실패하는 일이 없도록** 충분한 양의 쿠폰을 구매하려고 한다.

진우가 눈을 가리기 전에 확인한 와사비 초밥의 위치가 주어진다. 진우가 최선의 전략으로 챌린지에 도전한다면, 최소 몇 개의 쿠폰을 구매해야 반드시 챌린지에 성공할 수 있을까?

입력

첫 줄에 초밥의 개수 N과 챌린지에서 먹어야 하는 초밥의 개수 K가 공백을 사이에 두고 주어진다. $(1 \le K \le N \le 200\,000)$

둘째 줄에 문자 **O**와 **X**로 구성된 길이 N의 문자열이 주어진다. i번째 문자는 진우가 눈을 가리기 전에 **반시계 방향으로 i번째 위치에 놓인 초밥**이 와사비 초밥인지를 나타낸다. **O**는 와사비 초밥을, **X**는 와사비가 들지 않은 초밥을 나타낸다.

출력

진우가 반드시 챌린지에 성공하기 위해서 최소 몇 개의 쿠폰을 구매해야 하는지를 출력한다. 만약 몇 개의 쿠폰을 구매하더라도 챌린지에 실패할 가능성이 있다면, 대신 **-1**을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 2 0XX0XX	3
5 1 XX0XX	-1
4 4 XXXX	0
8 2 0XX0XX0X	5
8 1 X0XX00X0	6

문제 J. Distance Sum Maximization

시간 제한 3 초 메모리 제한 1024 MB

N개의 정점으로 이루어진 트리(사이클이 없는 무방향 연결 그래프)가 있다. 정점은 1번부터 N번까지 번호가 매겨져 있고, 간선은 1번부터 $(N-1)$번까지 번호가 매겨져 있다.

아래의 쿼리를 수행하는 프로그램을 작성하시오.

- u v : 정점 $x(1 \le x \le N)$에 대해, $\text{dist}(x, u) + \text{dist}(x, v)$의 최댓값을 출력한다. $(1 \le u, v \le N)$

이때 $\text{dist}(x, y)$는 정점 x에서 정점 y로 가는 최단경로 상의 간선 개수로 정의한다. 트리의 모든 정점 x에 대해 $\text{dist}(x, x) = 0$이다.

입력

첫째 줄에 트리의 정점 수 N가 주어진다. $(2 \le N \le 300\,000)$

다음 $(N-1)$개의 줄에는 트리의 정보가 주어진다. 이중 i번째 줄에는 i번 간선이 연결하는 두 정점 번호가 공백을 사이에 두고 주어진다.

다음 줄에 쿼리의 수 Q가 주어진다. $(2 \le Q \le 300\,000)$

다음 줄부터 Q개의 줄에는 쿼리의 정보가 한 줄에 하나씩 주어진다.

출력

Q개의 줄에 쿼리의 답을 순서대로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 1 2 2 3 2 4 4 5 3 1 3 1 5 2 3	6 5 5

문제 K. 스레드

시간 제한 3 초 메모리 제한 1024 MB

AI가 세상을 휩쓸고 있는 가운데, LG전자는 AI CPU를 탑재한 노트북 LG 그램, 방의 환경을 스스로 조절하는 에어컨 휘센 AI AIR 등 AI를 활용한 차세대 전자기기를 만들기 위해 전력을 다하고 있다.

이런 고성능 AI 연산을 가능케 하는 기술로 멀티스레딩을 빼놓을 수 없다. 스레드는 프로그램 내에서 병렬로 돌아가는 실행 경로를 의미하며, 컴퓨터는 이를 통해 여러 작업을 동시에 수행함으로써 효율성을 향상한다. 하지만 스레드 간에 공유되는 자원이 있을 때에는 동기화를 주의 깊게 맞춰 줘야 한다.

스레드를 막 배운 프로그래머 리프는 LG 그램을 열고, 정수형 변수 `x`를 선언한 뒤 N개의 스레드가 `x = x + 1`이라는 명령문을 실행하는 프로그램을 작성했다. `x`에 1을 더하는 이 명령문을 실행하려면 `x`를 읽는 작업과 `x`를 쓰는 작업이 필요한데, 사실 이 둘은 동시에 일어나는 게 아니라 다음 순서대로 일어난다.

- 절차 1: 스레드가 `x`의 값을 읽고 기억한다.
- 절차 2: 스레드가 자신이 기억한 값에 1을 더하고, 그 결과를 `x`에 덮어씌운다.

문제는 한 스레드의 절차 1과 2 사이에 다른 스레드가 간섭할 수 있다는 것이다. `x`의 초기값이 0이라고 할 때, 스레드 A와 B가 각각 절차 1을 진행하면 둘 다 0이란 값을 읽고 기억한다. 그 후 A와 B가 각각 절차 2를 진행하면 둘 다 `x`에 1을 쓰기 때문에, `x`의 값은 1이 된다. `x`를 1 늘리는 명령이 두 번 실행되더라도 `x`가 2만큼 늘어나지 않을 수도 있는 것이다. 그래서 프로그램을 돌렸더니 `x`의 값이 N이 아닌 다른 수가 되어 있는 것을 본 리프는 깜짝 놀랄 수밖에 없었다.

이제 우리가 LG 그램이 되어 스레드 N개를 원하는 순서로 실행시킬 수 있다고 생각해 보자. 각 스레드는 정확히 두 번 실행되어야 한다. 스레드가 처음 실행될 때는 절차 1을 진행하고, 두 번째로 실행될 때는 절차 2를 진행한다. 이렇게 스레드를 실행시키는 경우의 수는 $\frac{(2N)!}{2^N}$이다. 그렇다면 `x`의 초기값이 0이라고 할 때, N개의 스레드가 실행을 마쳤을 때 나올 수 있는 `x` 값의 분포는 어떻게 될까?

입력

첫 줄에 정수 N이 주어진다. ($1 \le N \le 200\,000$)

출력

첫 줄에 가능한 `x` 값의 개수 M을 출력한다.

다음 줄부터 M개의 줄에 걸쳐 한 줄에 하나씩, 가능한 `x` 값 하나와 그 `x` 값이 나오는 스레드 실행의 경우의 수를 998244353으로 나눈 나머지를 출력한다. 가능한 `x` 값이 여러 개라면 `x` 값에 대한 오름차순으로 모두 출력한다.

$998\,244\,353 = 119 \times 2^{23} + 1$은 소수이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2	2 1 4 2 2
100	100 ... [89 more lines] ... 90 729889561 91 145721628 92 477239109 ... [8 more lines] ...

노트

두 스레드를 A와 B라고 하고, 각 스레드의 스텝을 A1과 A2, 그리고 B1과 B2라고 하자. 스레드 실행 순서에 따른 x 값은 다음과 같다.

- A1 A2 B1 B2: $x = 2$
- A1 B1 A2 B2: $x = 1$ (본문에서 이 예시를 사용하였다.)
- A1 B1 B2 A2: $x = 1$
- B1 A1 A2 B2: $x = 1$
- B1 A1 B2 A2: $x = 1$
- B1 B2 A1 A2: $x = 2$

예제 2는 출력이 너무 길어 일부만 표시하였다. 실제로는 줄을 생략하지 않고 모두 출력해야 한다.

문제 L. 밤양갱

시간 제한 2.5 초 메모리 제한 1024 MB

시우는 $N \times N$ 크기의 격자 모양 밤양갱을 만들었다. 밤양갱의 각 칸에는 1 이상 N^2 이하의 서로 다른 정수로 등급이 매겨져 있다. 이제 이 밤양갱을 작은 조각으로 잘라 UCPC 참가자들에게 나누어 주려 한다. 이때 하나의 밤양갱 조각은 상하 또는 좌우로 인접한 두 개의 칸으로 이루어지며, 밤양갱 조각의 등급은 그 조각을 이루는 칸의 등급 중 더 큰 쪽을 따른다. 조각의 등급이 작을수록 맛있는 밤양갱이 되기 때문에, 시우는 어떤 참가자들이 맛없는 밤양갱 조각을 받는 일이 없도록 가장 등급이 큰 밤양갱 조각의 등급이 최소가 되도록 밤양갱을 자르려 한다.

그러나 시우는 이번 UCPC의 참가자가 몇 명인지 까먹고 말았다! 다행히 모든 참가자에게 밤양갱 조각을 나누어줄 수 있도록 밤양갱을 만들긴 했지만, 대회 시작 전까지 밤양갱을 잘라 나누어 주기에 시간이 부족했다. 그래서 시우는 밤양갱을 나누어줄 수 있는 모든 참가자 수마다 밤양갱을 자르는 최적의 방법을 찾기로 했다.

1 이상 $N^2/2$ 이하의 모든 정수 i에 대해, i명의 참가자에게 밤양갱 조각을 나누어줄 수 있도록 밤양갱을 자를 때 가장 등급이 큰 밤양갱 조각의 등급의 최솟값을 구하시오.

입력

첫 줄에 밤양갱의 한 변의 길이 N이 주어진다. ($2 \le N \le 100$; N은 짝수)

둘째 줄부터 N개의 줄에 걸쳐 각 줄마다 N개의 정수가 공백으로 구분되어 주어진다. $i+1$번째 줄의 j번째 정수 a_{ij}는 밤양갱의 위에서 i번째, 왼쪽에서 j번째 칸에 매겨진 등급을 의미한다. ($1 \le a_{ij} \le N^2$; a_{ij}는 서로 다르다.)

출력

$N^2/2$개의 줄에 걸쳐, i번째 줄에 i명의 참가자에게 밤양갱 조각을 나누어줄 수 있도록 밤양갱을 자를 때 가장 등급이 큰 밤양갱 조각의 등급의 최솟값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 6 2 3 7 16 9 10 12 1 15 11 14 4 5 8 13	3 4 7 8 10 14 15 16

문제 M. 지루함 줄이기

시간 제한 1초 메모리 제한 1024 MB

Moloco는 전 세계의 다양한 회사들이 온라인 광고를 더 잘할 수 있도록 돕는 광고 기술 회사입니다. Moloco의 기술을 사용하면 인터넷 사용자들이 더 관련성 높은 광고를 볼 수 있게 되고, 광고주들은 효과적으로 광고를 할 수 있습니다.

종서는 유저가 한 홈페이지에 총 $2N$번 방문할 것을 예측하였고, 유저에게 0번 광고와 1번 광고 총 두 종류의 광고를 각각 N번씩 보여주고자 합니다. 홈페이지를 방문할 때마다 같은 종류의 광고가 계속 나오면 사람들이 광고에 익숙해져 광고 효과가 떨어질 수 있기 때문에, 두 가지의 광고를 잘 배치하여 광고 효과를 극대화하고자 합니다.

광고 효과를 정량적으로 파악하기 위해 Moloco의 엔지니어 종서는 "지루함"이라는 지표를 정의했습니다. $i \le j$에 대해, i번째부터 j번째까지의 광고로 이루어진 **구간의 지루함**은 구간 내에 있는 0번 광고의 개수와 1번 광고의 개수의 차이로 정의됩니다. 유저가 최종적으로 느끼는 **지루함**은 모든 구간의 지루함 중 최댓값입니다.

예를 들어, 광고를 `00110110`과 같은 순서로 배치하였을 때, 3번째 광고부터 7번째 광고까지의 지루함은 $|1-4|=3$입니다. 이 구간이 지루함이 가장 크기 때문에, 유저가 최종적으로 느끼는 지루함 역시 3입니다.

Moloco의 훌륭한 알고리즘을 통해 유저의 참여도를 높일 수 있는 효과적인 광고 배치를 알아냈으나, 종서는 "지루함"이라는 지표가 얼마나 잘 작동하는지를 알아보기 위해 지루함을 줄이려고 합니다. 그러나 이미 광고의 순서가 정해져, 지루함을 줄이기 위해서는 연달아 나오는 두 광고를 맞바꾸기 위해 1만큼의 비용을 소모해야만 합니다. 맞바꾸는 작업은 원하는 만큼 수행할 수 있습니다.

유저가 최종적으로 느끼는 지루함을 K 이하로 만드는 데 필요한 최소 비용은 얼마일까요?

입력

첫 줄에 N과 K가 공백을 사이에 두고 주어집니다. ($1 \le K \le N \le 500\,000$)

둘째 줄에는 초기 광고 배치를 나타내는 N개의 `0`과 N개의 `1`로만 이루어진 길이가 $2N$인 문자열이 주어집니다.

출력

지루함을 K 이하로 만들기 위해서 필요한 최소 비용을 출력합니다.

주어진 광고 순서의 지루함이 이미 K 이하인 경우 `0`을 출력합니다.

가능한 모든 입력에 대해 항상 지루함을 1로 만들 수 있음을 보일 수 있습니다. 즉, 항상 답이 존재합니다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2 00110110	1
4 2 11110000	3
4 1 10011001	2