

2020

서울대학교 프로그래밍 경시대회

Division 2

주최 및 주관



후원

NAVER



서울대학교 컴퓨터공학부
Seoul National University
Dept. of Computer Science and Engineering



2020년 9월 12일

참가자를 위한 도움말

- 대회 시간은 13:00부터 17:00까지입니다. 대회가 진행되는 동안 인터넷 검색 및 라이브러리 코드 사용은 **전면 허용됩니다**. 하지만 다른 참가자나 외부인과 풀이, 코드를 공유할 수 없습니다. 또한, 대회 시간 동안 이 문제지를 외부로 유출하거나 공유할 수 없습니다.
- 대회는 Baekjoon Online Judge(<https://www.acmicpc.net/>) 플랫폼을 이용하여 진행됩니다. 별도로 제공되는 계정 정보를 이용하여 로그인하신 뒤 코드 제출 및 결과 확인 등을 하실 수 있습니다.
- 제출하실 답안 코드는 C11, C++14, C++17, Java, Java (OpenJDK), Python 3, PyPy3 **로만** 작성하셔야 합니다. 단, C++14와 C++17을 제외한 언어의 경우 시간 제한 및 메모리 제한 안에 문제를 풀 수 있음이 보장되지 않습니다.
- 모든 입력은 **표준 입력**으로 주어지고, 모든 출력은 **표준 출력**으로 합니다.
- 프로세스가 0이 아닌 값을 리턴하거나, 표준 오류 스트림 (`stderr`)에 값을 출력할 경우 런타임 에러를 받습니다.
- 문제에 대한 질의 사항은 대회 페이지의 질문 기능을 사용해 주시기 바랍니다. 이 때 대답해 드리기 어려운 질문에 대해서는 “답변을 드릴 수 없습니다”로 답변할 수 있습니다.
- 이 문제지는 참고용입니다. 문제지에 적힌 문제와 플랫폼에 올라온 문제가 다를 경우, 플랫폼의 문제가 우선합니다.

채점 결과

맞았습니다!! 제출하신 답안이 모든 테스트 데이터를 정해진 시간 안에 통과하여 정답으로 인정되었음을 의미합니다.

틀렸습니다 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

컴파일 에러 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다. 제출 기록을 클릭하여 오류가 발생한 부분을 볼 수 있습니다.

런타임 에러 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적으로 종료되었음을 의미합니다.

시간 초과 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

출력 형식이 잘못되었습니다 답을 올바르게 구했으나, 답안 프로그램의 출력 형식이 문제에 나와 있는 출력 형식과 다를 수 있음을 의미합니다. 대표적으로 줄 뒤에 의미 없는 공백을 두 칸 이상 출력하거나 의미 없는 빈 줄을 출력할 경우 이 결과를 받을 수 있습니다.

출력 초과 답안 프로그램이 정답에 비해 너무 많은 출력을 했음을 의미합니다. 이 결과는 **틀렸습니다**와 같은 의미를 갖습니다.

만약 여러 원인으로 인해 **맞았습니다!!**가 아닌 다른 결과를 얻으셨다면, 그 중 어떤 것도 결과가 될 수 있습니다. 예를 들어 답도 잘못되었고 비정상적인 동작도 수행하는 코드를 제출하신 경우 대부분 **런타임 에러**를 받으시게 되지만, 경우에 따라서 **틀렸습니다**를 받을 수도 있습니다.

문제 정보

문제 번호	제목	시간 제한	메모리 제한
A	수면 패턴	1.0s	1024MB
B	단어 개수 세기	2.0s	
C	넴모넴모 2020	3.0s	
D	신기한 연산	1.0s	
E	여우 신탁	1.0s	
F	빈 문자열 만들기	1.0s	
G	문제지 나르기	3.0s	
H	$2 \times M$ 타일링	1.0s	
I	버거운 버거	3.0s	

문제지와 플랫폼의 제한 조건이 다를 경우 플랫폼을 우선으로 참고하세요.

Problem A. 수면 패턴

사회적 거리두기와 휴학으로 인해 집콕 생활에 익숙해진 민티는 수면 패턴이 자꾸 깨져서 걱정이다.

민티는 건강을 위해 앞으로는 꼭 일주일에 T 시간 이상 잠을 자기로 결심했다. 매일 같은 시간 자고 일어나는 것은 아직 너무 힘들기 때문에, 평일에 자고 일어난 시각을 기록해 두고 주말에 모자란 만큼 몰아서 자기로 했다. 민티는 한번 잠들면 최소 1시간 이상 잠을 자며, 가끔은 일어나자마자 너무 졸려서 바로 다시 잠들기도 한다.

평일이 다 지나고, 민티에게 마음껏 잘 수 있는 48시간의 주말이 주어졌다. 민티는 평일에 자신이 적어 놓은 기록을 보고 주말 동안 몇 시간을 자야 하는지 계산하려고 한다.

민티가 평일 동안 잠든 시간과 일어난 시간이 주어질 때, 주말에 자야 할 최소 시간을 출력하라.

입력 형식

일주일에 자야 할 시간을 나타내는 정수 T 와 민티가 평일 동안 잠든 횟수를 나타내는 정수 N 이 주어진다.

다음 줄부터 N 개의 줄에 민티가 잠든 시간이 주어진다. 각 줄에 D_1, H_1, D_2, H_2 의 순서로 민티가 잠든 시각과 일어난 시각이 주어진다. D_1 은 잠든 요일, H_1 은 잠든 시각, D_2 는 일어난 요일, H_2 는 일어난 시각을 의미한다. 요일은 “Mon”, “Tue”, “Wed”, “Thu”, “Fri” 중 하나이며, 각각 월요일, 화요일, 수요일, 목요일, 금요일을 의미한다. 시각은 0 이상 23 이하의 정수로 주어지며, 각각 0시부터 23시까지의 시각을 의미한다.

각 수면은 서로 겹치지 않지만, 민티가 일어났다가 바로 잠들 수 있기 때문에 한 수면의 잠든 시각과 다른 수면의 일어난 시각이 동일한 경우가 있을 수 있다. 수면의 시작과 끝은 항상 평일이며, 입력은 잠든 시각이 빠른 순으로 정렬된 상태로 주어진다. 민티가 평일 동안 잠을 아예 자지 않는 경우가 존재할 수 있음에 유의하라.

출력 형식

민티가 주말에 더 자야 할 시간을 출력한다. 만약 주말에 잠을 자지 않아도 충분할 경우 0을, 주말 내내 자도 부족할 경우 -1을 출력한다.

제약 조건

- $0 \leq T \leq 168$
- $0 \leq N \leq 10$

예제

표준 입력(stdin)	표준 출력(stdout)
56 6 Mon 22 Tue 2 Tue 15 Tue 19 Tue 19 Wed 4 Wed 17 Thu 1 Thu 12 Thu 17 Fri 4 Fri 12	18
50 5 Mon 0 Mon 11 Tue 2 Tue 12 Wed 0 Wed 10 Thu 3 Thu 15 Fri 11 Fri 23	0
120 5 Mon 0 Mon 11 Tue 2 Tue 12 Wed 0 Wed 10 Thu 3 Thu 15 Fri 11 Fri 23	-1

Problem B. 단어 개수 세기

키파는 프랑스어를 배우고 있다. 프랑스어는 띄어쓰기 단위로 단어가 완벽하게 분리되는 언어는 아니다.

- 기본적으로는 띄어쓰기나 -(하이픈) 단위로 단어를 구분한다.
- 앞 단어가 *ce, je, ne, me, te, se, le, la, de, que* 혹은 *si*이고 뒤 단어가 모음(a, e, i, o, u, h)으로 시작하는 경우, 앞 단어의 마지막 모음이 사라지고, 대신 '(어포스트로피)가 붙으면서 이어진다.
 - 프랑스어에서 *h*는 언제나 묵음이므로, 이 문제에서는 일반적으로 알려진 모음 a, e, i, o, u는 물론이고 **h도 모음으로 취급**함에 유의하라.

예를 들어, *je*는 “나”라는 뜻이고, *aime*는 “(내가) 좋아한다”라는 뜻이고, (*les*) *pommes*는 “사과”라는 뜻이다. 이 단어들로 “나는 사과를 좋아한다”라는 프랑스어 문장을 만들 때는 *j'aime les pommes*와 같이 쓴다. 앞 단어가 *je*이고 뒤 단어의 모음인 *a*가 만났기 때문에 *e*가 사라지고 대신 어포스트로피가 붙은 것이다. 그래서, 프랑스어 초짜인 키파는 이런 식으로 프랑스어 문장의 단어를 쪼개기로 했다.

- 먼저 띄어쓰기와 -(하이픈)을 기준으로 “단어”를 쪼갬다.
- 각각의 “단어”에서, 위처럼 줄어들었을 가능성이 있는 경우(즉, *c', j', n', m', t', s', l', d', qu'*로 시작하고 어포스트로피 뒤 글자가 모음인 경우) 이 단어들을 한 번 더 분리해 준다.

그런데 생각해 보니 이 과정을 프랑스어 문장에만 적용할 수 있는 것은 아니었다!

임의의 문자열이 주어졌을 때, 키파의 발상대로 단어 단위를 만들었을 경우 몇 단어가 되는지를 구하는 프로그램을 작성하라.

입력 형식

첫째 줄에 “문장”을 나타내는 문자열이 주어진다. 이 문자열은 영어 소문자, 띄어쓰기, -(하이픈), '(어포스트로피)로만 이루어져 있다. 이때 띄어쓰기, 하이픈, 어포스트로피 중 어느 것도 인접해 있지 않고, 문장의 시작이나 끝에 있지 않다.

출력 형식

키파의 발상대로 입력된 “문장”을 단어로 쪼개었을 경우 몇 단어가 되는지를 출력하라.

제약 조건

- “문장”은 길이가 1 이상이고 5,000 이하인 문자열이다.
- “문장”은 영어 소문자, 띄어쓰기, -(하이픈), '(어포스트로피)로만 이루어져 있다.
- 띄어쓰기, 하이픈, 어포스트로피 중 어느 것도 인접해 있지 않고, 문장의 시작이나 끝에 있지 않다.

예제

표준 입력(stdin)	표준 출력(stdout)
qu'est-ce qu'il mange aujourd'hui	7

예제 설명

키파의 발상을 따라가면 다음과 같다.

- 먼저 띄어쓰기와 하이픈을 기준으로 “단어”로 분리해 준다: qu'est / ce / qu'il / mange / aujourd'hui.
- 첫 번째 “단어”인 qu'est는 qu'로 시작하고 모음인 e로 이어지기 때문에 que와 est가 만나서 줄어드는 것이다. 따라서 이 단어를 분리해 준다: que / est / ce / qu'il / mange / aujourd'hui.
- 세 번째 “단어” 역시 마찬가지로 분리해 준다: que / est / ce / que / il / mange / aujourd'hui.
- 마지막 “단어”는 어포스트로피가 있고 모음으로 취급되는 h로 이어지지만 앞 단어가 줄어들어 생긴 어포스트로피가 아니므로, 분리하지 않고 한 단어로 둔다.

따라서 총 7개의 단어가 있다. 이 문장의 뜻은 “오늘 그는 무엇을 먹습니까?”이다.

Problem C. 녀모녀모 2020

오래된 테트리스 게임판 위에 수수께끼의 생물 “녀모”들이 살기 시작했다. 이 게임판은 가로로 10^9 칸, 세로로 N 층 크기이고, 녀모 한 마리는 한 층의 한 칸을 차지하고 산다. 편의상 왼쪽에서부터 x 번째, 아래쪽에서부터 y 층을 (x, y) 로 표기하자.

y 층에는 a_y 마리의 녀모들이 살고 있다. 녀모들은 붙어있는 걸 좋아하기 때문에 $(1, y), \dots, (a_y, y)$ 칸에 나란히 살고 있으며, 중력의 영향을 받기 때문에 모든 $1 \leq y \leq N - 1$ 에 대해 $a_y \geq a_{y+1}$ 이다.

				...
				...
				...

테트리스 게임판에 살고 있는 녀모들. 이 경우 $N = 3$, $a_1 = 3$, $a_2 = 3$, $a_3 = 2$ 이다.

테트리스를 하고 싶은 레프는 레이저를 이용해서 녀모들을 치워버리려고 한다. (x, y) 에 레이저를 설치하면 왼쪽에서 x 번째 칸에 살고 있는 녀모들 중 y 층 이상에 살고 있는 녀모들, y 층에 있는 녀모들 중 (x, y) 보다 오른쪽에 있는 녀모들이 모두 사라진다. 그 이외의 녀모는 당장 사라지지 않는다.

				...
				
				...

$(1, 2)$ 에 레이저를 설치한 모습. 총 4마리의 녀모가 레이저에 맞아 사라진다.

레이저를 설치할 수 있는 위치는 총 Q 개가 있다. 레프를 위해 각 위치에 레이저를 설치했을 때 몇 마리의 녀모를 없앨 수 있는지 알려주자. 단, 실제로 레이저를 설치하는 것이 아닌 설치 계획만 하는 것이기 때문에, 설치 계획끼리 서로 영향을 주고받지는 않는다.

입력 형식

첫째 줄에 정수 N , Q 가 공백으로 구분되어 주어진다. N 은 게임판의 세로 크기, Q 는 레이저를 설치할 수 있는 위치의 수를 의미한다.

둘째 줄에는 N 개의 정수 $a_1, \dots, a_N$ 이 공백을 사이에 두고 주어진다. 이는 i 층에 a_i 마리의 녀모가 살고 있다는 의미이다.

셋째 줄부터 Q 개의 줄에 걸쳐 레이저를 설치할 수 있는 위치가 주어진다. $(i + 2)$ 번째 줄에는 두 정수 x_i 와 y_i 가 공백을 사이에 두고 주어지는데, 이는 (x_i, y_i) 에 레이저를 설치할 수 있다는 의미이다.

출력 형식

Q 개의 줄에 걸쳐 답을 출력한다. i 번째 줄에는 (x_i, y_i) 에 레이저를 설치하면 몇 마리의 뱀모를 제거할 수 있는지 출력한다.

제약 조건

- $1 \leq N, Q \leq 250,000$
- $1 \leq a_i \leq 10^9 \quad (1 \leq i \leq N)$
- $a_1 \geq a_2 \geq \dots \geq a_N$
- $1 \leq x_i \leq 10^9, 1 \leq y_i \leq n \quad (1 \leq i \leq Q)$

예제

표준 입력(stdin)	표준 출력(stdout)
3 11	5
3 3 2	4
1 1	2
1 2	4
1 3	3
2 1	1
2 2	2
2 3	1
3 1	0
3 2	0
4 1	0
4 2	
3 3	

Problem D. 신기한 연산

재현이는 문제를 풀다가 신기한 연산을 발견했다. 이 연산을 사용하면 홀수 번 등장하는 원소가 **단 하나** 있는 원소들의 나열에서 그 원소를 빠르게 찾을 수 있다. 예를 들어 수열 $(1, 3, 2, 1, 2)$ 에 이 연산을 적용하면 유일하게 한 번만 등장하는 수인 3을 얻을 수 있다.

재현이는 이 연산에 큰 감명을 받고, 이 연산을 사용하는 문제를 반드시 출제하기로 결심했다. 그리고 마침내 출제할 기회가 주어지자, 다음과 같은 문제를 야심차게 내놓았다.

N 종류의 알파벳 소문자로 이루어진 M 글자의 문자열과 Q 개의 구간들이 주어진다. 문자열에서 각각의 구간에 해당하는 부분 문자열을 뽑았을 때, 각 부분 문자열에는 N 종류의 알파벳이 **전부** 등장하며 그중 홀수 번 등장하는 알파벳이 **단 한 종류** 있음이 보장된다.

이때 각 구간에 대해 해당하는 부분 문자열에서 홀수 번 등장하는 알파벳을 찾는 프로그램을 작성하라.

문제의 데이터를 준비하던 재현이는 구간들은 쉽게 만들 수 있었으나, 이 구간들에 대해 위의 조건을 만족하는 문자열을 도저히 만들 수 없었다. 그래서 재현이는 데이터를 만드는 작업을 또다른 출제진인 당신에게 맡기기로 했다. 재현이를 도와서 데이터를 만들어 주자.

입력 형식

첫 줄에 세 개의 정수 N , M , Q 가 주어진다. N 은 사용해야 하는 알파벳의 종류 수, M 은 만들고자 하는 문자열의 길이, Q 는 재현이가 만들어 놓은 구간들의 개수를 의미한다.

이후 Q 개의 줄에 걸쳐 구간들이 주어진다. $i + 1$ 번째 줄에는 두 개의 정수 s_i 와 e_i 가 공백을 사이에 두고 주어지며, 각각 구간의 시작 위치와 끝 위치를 의미한다. 각 구간의 너비는 $2N - 1$ 이상의 홀수이며, 같은 구간이 여러 번 주어지지 않음이 보장된다.

출력 형식

첫 줄에 알파벳 소문자만으로 이루어진 M 글자의 문자열을 출력한다. 이 문자열은 정확히 N 종류의 알파벳으로 이루어져 있어야 하며, 입력으로 주어진 모든 구간에 대해 재현이가 출제하려는 문제의 조건을 만족해야 한다.

이러한 문자열이 존재하는 경우만이 입력으로 주어짐이 보장된다.

제약 조건

- $1 \leq N \leq 26$
- $2N - 1 \leq M \leq 100,000$
- $1 \leq Q \leq 100,000$

예제

표준 입력(stdin)	표준 출력(stdout)
3 10 4 1 9 3 7 2 6 4 10	baccbabbac

예제 설명

첫 번째 구간 [1, 9]에 해당하는 부분 문자열은 baccbabba으로, 알파벳 a가 홀수 번 등장한다.

만약 구간으로 [4, 8]이 주어진다면 cbabb에는 모든 알파벳이 홀수 번 등장하므로 이 문자열은 답이 될 수 없다.

만약 구간으로 [5, 9]가 주어진다면 babba에는 알파벳이 두 종류밖에 사용되지 않았으므로 이 문자열은 답이 될 수 없다.

Problem E. 여우 신탁

“여우신님, 여우신님, 번호 하나만 내려 주세요...”

누구나 1 이상 45 이하의 정수 여섯 개를 잘 골라서 인생이 달라지거나, 혹은 1 이상 10,000 이하의 정수 하나를 잘 골라서 **맞았습니다!!**를 받는 헛된 꿈을 꾸다. 이렇듯 일상 생활에서 신기가 담긴 수가 필요한 상황은 얼마든지 일어난다. 이는 여우 마을의 여우들에게도 마찬가지다. 다만 여우 마을과 현실의 차이점은, 여우 마을에서는 여우신에게 부탁을 드리면 실제로 여우신께서 신탁으로 수를 하나 내려 주신다는 점이다.

여우 마을의 여우들은 수를 셀 때 0부터 세기 때문에, X 개의 수 중 하나가 필요할 때는 여우신에게 “0 이상 X **미만**의 수를 하나 내려 주세요”와 같이 부탁을 드린다. 그러면 여우신은 0 이상 $X - 1$ 이하의 정수 하나를 **균일한** 확률로 골라서 신탁으로 내려 준다.

그러나 너무 많은 여우들이 여우신을 찾게 되자, 여우신은 매번 새로운 수를 고르기가 귀찮아졌다. 그래서 여우신은 여우들이 올린 부탁을 여러 개 모아서 한꺼번에 처리하기로 했다. 첫 번째 여우에게는 제대로 수를 골라서 내려 주고, 그 다음부터는 바로 전에 내려 줬던 수를 새로 부탁받은 값으로 나눈 **나머지**를 내려 주기로 한 것이다.

예를 들어 여우 세 마리가 차례대로 10개의 수 중 하나, 3개의 수 중 하나, 5개의 수 중 하나를 필요로 할 경우 여우신은 다음과 같은 방법으로 신탁을 내린다.

1. 0 이상 9 이하의 정수를 균일한 확률로 하나 골라서 첫 번째 여우에게 내려 준다.
2. 첫 번째 여우에게 내려 준 수를 3으로 나눈 나머지를 두 번째 여우에게 내려 준다.
3. 두 번째 여우에게 내려 준 수를 5로 나눈 나머지를 마지막 여우에게 내려 준다.

물론 신탁을 받는 여우들은 눈치채지 못하겠지만, 이렇게 내려 주는 신탁은 공정한 신탁이 아닐 수 있다. 여우신이 반성하고 다시 제대로 신탁을 내릴 수 있도록, 마지막 여우가 받게 되는 수의 분포를 계산해서 보여주자.

입력 형식

첫 줄에 신탁을 요청한 여우의 수를 의미하는 정수 N 이 주어진다.

다음 줄에는 N 개의 정수 $x_1, x_2, \dots, x_N$ 이 공백을 사이에 두고 차례대로 주어진다. x_i 는 i 번째 여우가 x_i 개의 수 중 하나를 필요로 함을 의미한다.

출력 형식

마지막 여우가 받게 되는 수의 기댓값을 출력한다. 절대 또는 상대 오차가 10^{-9} 이하면 정답으로 처리된다.

제약 조건

- $1 \leq N \leq 300,000$
- $1 \leq x_i \leq 10^6$ ($1 \leq i \leq N$)

예제

표준 입력(stdin)	표준 출력(stdout)
1 6	2.5
3 9 4 2	0.444444444444
3 10 3 5	0.9

예제 설명

첫 번째 예제의 경우 여우신은 0부터 5까지의 정수 중 하나를 균일한 확률로 내려 주게 된다.

두 번째 예제의 경우 마지막 여우는 $\frac{5}{9}$ 확률로 0, $\frac{4}{9}$ 확률로 1을 받게 된다.

Problem F. 빈 문자열 만들기

0과 1로만 이루어져 있으며, 0의 개수와 1의 개수가 동일한 문자열 S 가 주어진다. 당신은 S 에 다음과 같은 작업을 여러 번 수행할 수 있다:

S 의 길이 $2k$ 인 연속한 부분문자열이 앞 k 개 문자가 모두 동일하고, 또한 뒤 k 개 문자가 서로 동일하며, 0과 1을 모두 포함할 때, 그 부분문자열을 제거할 수 있다.

예를 들어, $S = "0111000011"$ 인 경우, S 의 2번째 문자부터 7번째 문자까지인 "111000"을 제거하는 것이 가능하다. 이 작업 후에는 제거된 부분의 앞부분과 뒷부분이 연결되어 $S = "0011"$ 이 된다. 그러면 이제 한번의 작업을 통해 "0011"을 제거할 수 있으므로 초기 $S = "0111000011"$ 는 두 번의 작업을 통해 빈 문자열로 만들 수 있다.

여러분의 목표는 최소 횟수의 작업을 통해 S 를 빈 문자열로 만드는 것이다. 최소 횟수의 작업으로 S 를 빈 문자열로 만드는 과정을 구하여라.

입력 형식

첫 줄에 0과 1로만 이루어진 문자열 S 가 주어진다. S 에 포함된 0의 개수와 1의 개수는 동일하다.

출력 형식

작업들을 통해 S 를 빈 문자열로 만드는 것이 불가능하다면 첫 줄에 -1을 출력한다.

그렇지 않은 경우, 첫 줄에 필요한 작업의 최소 횟수 K 를 출력한다. 그 다음 K 줄에 걸쳐 수행한 작업에 대한 정보를 출력한다.

K 개 줄 중 i 번째 줄에는 두 정수 b_i 와 e_i 를 공백을 사이에 두고 출력한다. 이는 i 번째 작업에서 S 의 b_i 번째 문자부터 e_i 번째 문자로 이루어진 문자열을 제거하였음을 뜻한다.

최소 횟수의 작업으로 문자열을 지우는 방법이 여러 가지인 경우에는 그 중 아무 것이나 출력해도 된다.

제약 조건

- S 에 포함된 문자의 개수는 1 이상 500,000 이하이다.
- S 에 포함된 0의 개수와 1의 개수는 동일하다.

예제

표준 입력(stdin)	표준 출력(stdout)
0101110100	4 2 3 4 5 3 6 1 2

예제 설명

처음에 $S = \text{"0101110100"}$ 인 상태이다.

S 의 i 번째 문자부터 j 번째 문자까지의 문자열을 $S[i : j]$ 라 하자.

$S[2 : 3] = \text{"10"}$ 을 제거하면 $S = \text{"01110100"}$ 이 된다.

그 후 $S[4 : 5] = \text{"10"}$ 을 제거하면 $S = \text{"011100"}$ 이 되고,

다음에 $S[3 : 6] = \text{"1100"}$ 을 제거하면 $S = \text{"01"}$ 이 된다.

마지막으로 $S[1 : 2] = \text{"01"}$ 을 제거하면 S 는 빈 문자열이 된다.

$S = \text{"0101110100"}$ 을 4번 미만의 작업을 통해 빈 문자열로 만드는 방법은 존재하지 않는다.

Problem G. 문제지 나르기

교준이는 온라인으로 진행되는 SNUPC 2020에 아쉬움이 남아, N 명의 대회 참가자들의 집으로 문제지를 직접 배달해주기로 했다.

여러분이 익히 알고 있듯 이 우주는 11차원이기 때문에, 참가자들의 집은 11차원 좌표의 한 점 $\mathbf{x} = (x_1, x_2, \dots, x_{11})$ 로 나타낼 수 있다. 이 우주에서 이동하기 위해서는 좌표축을 따라 나 있는 도로를 타고 이동해야 하기 때문에, 두 점 $\mathbf{x}, \mathbf{y}$ 사이를 이동하는 데 필요한 거리는 아래와 같다.

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{11} |x_k - y_k|$$

교준이는 문제지를 실은 차를 한 곳에 주차해둔 뒤 문제지를 나를 것이다. 11차원 세계에는 Q 개의 주차장이 있고, 주차장 역시 11차원 좌표계의 한 점 $\mathbf{y} = (y_1, \dots, y_{11})$ 으로 나타낼 수 있다.

문제지를 직접 들고 나르는 일은 힘들기 때문에, 교준이는 각 주차장의 위치에 대해 가장 멀리 떨어진 참가자의 집까지 거리를 알고 싶어한다. 교준이는 이 문제가 나쁘지 않다고 생각했는지, 문제지에 이 문제까지 적어서 여러분에게 가져다주려고 한다.

입력 형식

입력의 첫째 줄에는 참가자의 수 N , 주차장의 수 Q 가 공백을 사이에 두고 주어진다.

둘째 줄부터 N 개의 줄에 걸쳐 각 참가자의 집 좌표 $(x_{i,1}, \dots, x_{i,11})$ 가 주어진다. 구체적으로, $(i+1)$ 번째 줄에는 i 번째 참가자의 집 좌표를 나타내는 11개의 정수 $x_{i,1}, \dots, x_{i,11}$ 가 공백을 사이에 두고 주어진다.

$(N+2)$ 번째 줄부터 Q 개의 줄에 걸쳐 주차장의 좌표 $(y_{i,1}, \dots, y_{i,11})$ 가 주어진다. 구체적으로, $(N+1+i)$ 번째 줄에는 i 번째 주차장의 집 좌표를 나타내는 11개의 정수 $y_{i,1}, \dots, y_{i,11}$ 가 공백을 사이에 두고 주어진다.

출력 형식

Q 개의 줄에 걸쳐 답을 출력한다. i 번째 줄에는 i 번째 주차장과 가장 멀리 떨어진 참가자의 집까지 거리를 출력한다.

제약 조건

- $1 \leq N, Q \leq 50,000$
- $-10^9 \leq x_{i,j}, y_{i,j} \leq 10^9 \quad (1 \leq i \leq N, 1 \leq j \leq 11)$

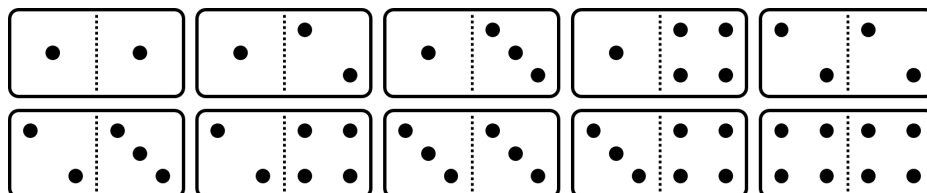
예제

표준 입력(stdin)	표준 출력(stdout)
2 2	87
0 1 2 3 4 5 6 7 8 9 10	312
0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10	
3 8 -4 2 4 6 0 -9 5 2 7	
10 34 2 -38 17 55 -23 30 -19 41 22	

Problem H. $2 \times M$ 타일링

상수는 프로그래밍 대회 상품으로 타일들이 들어 있는 상자를 받았다. 이 타일들은 두 개의 정사각형 칸이 변을 맞대고 붙어 있는 직사각형 모양이고, 각 칸에는 정해진 개수만큼의 점이 찍혀 있다. 편의상 한 칸에는 a 개, 다른 칸에는 b 개의 점이 찍힌 타일을 (a, b) 타일이라고 하자.

타일 상자는 크기에 따라서 안에 들어 있는 타일들의 구성이 달라지며, 상수가 받은 타일 상자는 크기가 K 인 상자이다. 이 상자에는 $1 \leq a \leq b \leq K$ 를 만족하는 모든 정수 쌍 a, b 에 대해 (a, b) 타일이 각각 하나씩, 총 $\frac{K(K+1)}{2}$ 개의 타일이 들어 있다.



$K = 4$ 인 경우의 예시

머릿속이 온통 알고리즘 문제뿐인 상수는 타일의 모양을 보자마자 유명한 “ $2 \times N$ 타일링” 문제가 생각났다. 그래서 상수도 상자에 들어 있는 타일들 중 M 개의 타일을 골라서 $2 \times M$ 격자판을 채워 보기로 했다.

타일을 놓을 때는 반드시 격자판의 칸에 맞춰서 놓아야 하며, 가로로 놓을지 세로로 놓을지는 자유롭게 정할 수 있다. 같은 방향이더라도 타일을 180도 돌려서 두 칸의 순서를 바꿔서 놓을 수도 있다. 당연히 타일이 격자판을 벗어나거나 두 타일이 겹쳐서는 안 되고, 따라서 M 개의 타일을 올바르게 놓으면 $2M$ 개의 칸을 전부 덮게 된다.

그러나 단순히 격자판을 채우는 것은 너무 쉽다고 생각한 상수는 타일들에 찍혀 있는 점들을 이용해서 다음과 같은 조건들을 추가로 만들었다.

- 격자판의 두 가로줄에는 서로 같은 개수만큼의 점이 찍혀 있어야 한다.
- 격자판의 모든 세로줄에는 점이 정확히 $K + 1$ 개씩 찍혀 있어야 한다.

상수가 주어진 조건에 맞게 격자판을 채울 수 있을지 알아보자.

입력 형식

첫 줄에 두 개의 정수 K 와 M 이 공백을 사이에 두고 주어진다. K 는 상수가 받은 타일 상자의 크기, M 은 상수가 채우려는 격자판의 너비를 의미한다.

출력 형식

상수가 조건에 맞게 격자판을 채울 수 있다면 첫 줄에 YES를 출력한다. 그리고 다음 두 줄에 걸쳐 각 줄에 격자판의 각 칸에 찍혀 있는 점의 개수를 의미하는 M 개의 정수를 공백으로 구분하여 출력한다. 단, 해당 칸을 덮는 타일이 세로로 놓여 있을 경우는 앞에 -를 붙여 음수로 출력한다.

격자판을 채울 수 없는 경우는 첫 줄에 NO를 출력한다.

제약 조건

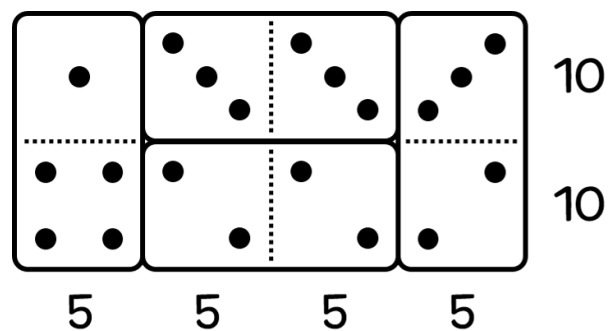
- $1 \leq K \leq 15$
- $1 \leq M \leq \frac{K(K+1)}{2}$

예제

표준 입력(stdin)	표준 출력(stdout)
4 4	YES -1 3 3 -3 -4 2 2 -2
2 3	NO

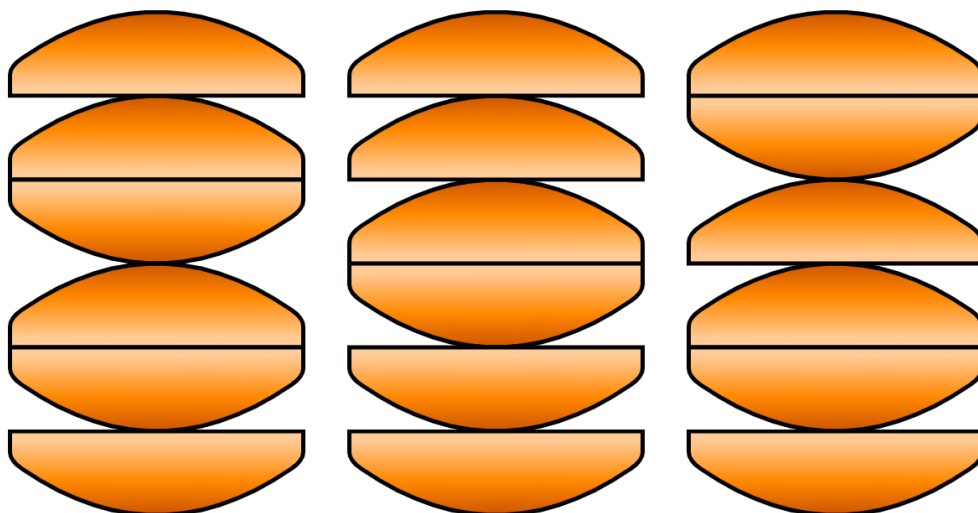
예제 설명

첫 번째 예제의 출력은 다음과 같은 배치를 나타낸다.



Problem I. 버거운 버거

드디어 산업기능요원 복무를 마친 키파는 *버거운* 직장에서 벗어나 새로운 직업에 도전하고자 햄버거집을 차렸다. 키파는 케이크를 여러 차례 만들면서 빵은 좀 구워 봤지만 햄버거를 만드는 것은 처음이었기 때문에, 위아래의 구분이 있는 빵을 적당히 구워서 햄버거 패티로 햄버거가 들어간 탄수화물 폭탄의 *버거운 버거*를 만들어 팔기로 했다.



버거운 버거의 예시.

버거운 버거의 엄밀한 정의는 다음과 같다.

- 속이 위로 온 빵 X 위에 속이 아래로 온 빵 Y 를 올린 것은 버거운 버거이다. 이때 X 를 Y 의 대응하는 쌍, Y 를 X 의 대응하는 쌍이라 정의하자.
- 속이 위로 온 빵 X 위에 버거운 버거를 올리고, 그 위에 속이 아래로 온 빵 Y 를 올린 것은 버거운 버거이다. 마찬가지로 이때 X 를 Y 의 대응하는 쌍, Y 를 X 의 대응하는 쌍이라 정의하자.
- 버거운 버거 위에 버거운 버거를 올린 것은 버거운 버거이다.
- 위 세 규칙으로 만들 수 없는 것은 버거운 버거가 아니다.

키파는 총 N 개의 빵 굽는 기계를 일렬로 세워 두고 동시에 다루고 있다. 하나의 기계는 빵을 하나만 구울 수 있다. 가장 왼쪽의 것부터 오른쪽으로 순서대로 1부터 번호를 붙이자. 키파가 가게를 운영하는 도중에 다음 둘 중 하나의 상황이 Q 번 발생할 수 있다.

- a 번 기계부터 b 번 기계까지의 빵이 곧 타려고 하기 때문에, 각각 뒤집어 줘야 한다.
- 손님이 a 번 기계부터 b 번 기계까지의 빵을 차곡차곡 쌓아 주기를 원한다. 즉, 이 과정 중 빵을 뒤집어 쌓으면 안 되고, 가장 아래에 a 번 기계에서 나온 빵, 그 위에 $(a+1)$ 번 기계에서 나온 빵, 이렇게 하여 가장 위에 b 번 기계에서 나온 빵이 순서대로 쌓여야 한다. 키파는 기계에 빵이 없으면 재료가 다 떨어진 것처럼 보인다고 생각했기 때문에, 한 주문이 끝난 이후 그 주문을 받기 이전의 상태대로 빵의 위아래를 맞춰서 채워 둔다.

그러나 손님들이 햄버거를 만들기를 원하는 빵을 쌓았을 때, 어떤 빵은 대응하는 쌓이 존재하지 않아 버거운 버거로서 실격일 수 있다. 키파는 각 손님의 주문대로 빵을 쌓은 뒤, 이 빵을 버거운 버거로 만들기 위해 쌓아둔 빵의 순서를 유지하면서 미리 구워 둔 빵을 최소한으로 집어넣고자 한다. 키파는 손님들의 건강에 관심이 많기 때문에 각 주문마다 버거운 버거의 높이, 즉 버거운 버거에 들어간 빵의 개수를 알고자 한다. 이를 구하는 프로그램을 작성해서 키파를 도와 주자.

입력 형식

첫 줄에 양의 정수 N 이 주어진다.

둘째 줄에 길이가 N 이고 (와)로만 구성된 문자열이 주어진다. 모든 $1 \leq i \leq N$ 에 대해 i 번째 문자가 (인 경우 속이 위로 온 빵이 i 번 기계에,)인 경우 속이 아래로 온 빵이 i 번 기계에 들어 있다는 의미이다.

셋째 줄에 양의 정수 Q 가 주어진다.

넷째 줄부터 Q 개의 줄에 세 개의 양의 정수 t, a, b 로 상황이 주어진다. t 는 2 이하이며, $1 \leq a \leq b \leq N$ 이다.

$t = 1$ 인 경우 a 번 기계부터 b 번 기계까지의 빵을 뒤집어 줘야 함을 의미한다.

$t = 2$ 인 경우 a 번 기계부터 b 번 기계까지의 빵을 꺼내 버거운 버거로 만들어 달라는 주문이 들어왔음을 의미한다.

출력 형식

각 주문($t = 2$)마다 버거운 버거의 높이를 한 줄에 하나씩 출력한다.

제약 조건

- $1 \leq N \leq 1,000,000$
- $1 \leq Q \leq 300,000$

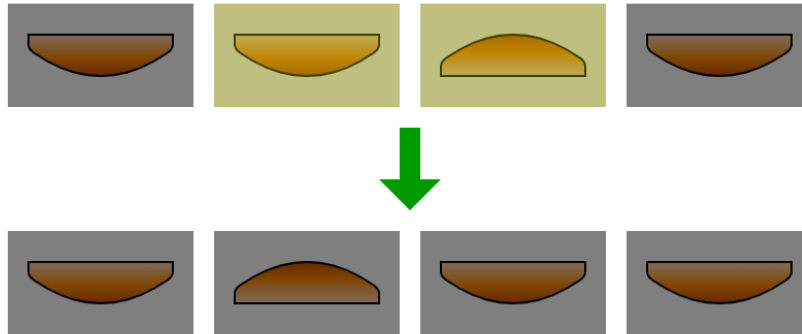
예제

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

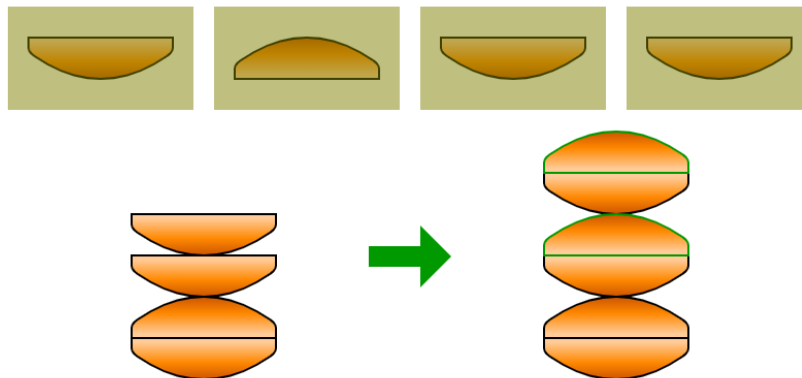
예제 설명

입력 예제에서는 총 다섯 번의 상황이 발생한다.

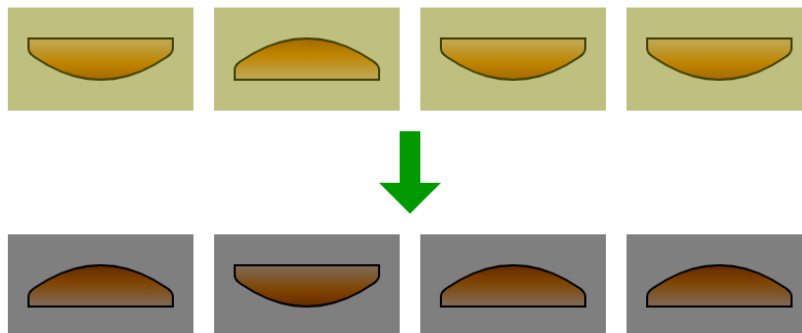
첫 번째 상황은 2번 기계부터 3번 기계까지에 들어 있는 빵을 모두 뒤집어 주어야 하는 상황이다. 기계 안에 있는 빵을 하나씩 뒤집음에 유의하라.



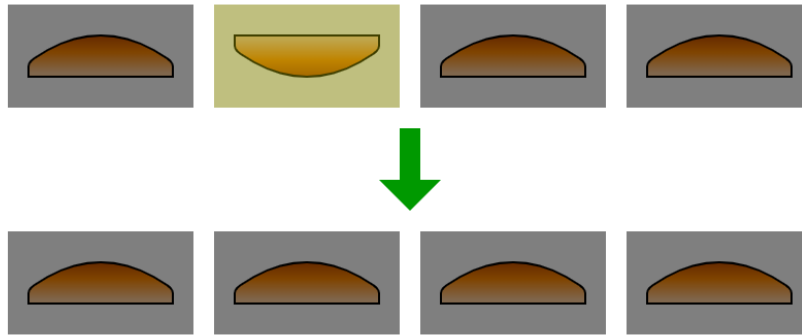
두 번째 상황은 손님이 1번 기계부터 4번 기계까지의 빵을 모두 꺼내 버거운 버거를 만들어 달라는 주문이다. 빵을 꺼내서 쌓으면 왼쪽 아래와 같이 되고, 두 개의 빵을 추가해 높이 6의 버거운 버거를 만들 수 있으며, 이보다 높이가 더 낮은 버거운 버거를 만들 수는 없다. 빵을 추가할 때는 기존에 쌓여 있던 빵의 순서는 유지되어야 하며, 기존 빵을 뒤집을 수 없음에 유의하라.



세 번째 상황은 1번 기계부터 4번 기계까지에 들어 있는 빵을 모두 뒤집어 주어야 하는 상황이다. 첫 번째 주문을 처리했지만, 키파는 기계에 빵이 비는 것을 싫어하기 때문에 모두 원래 방향으로 채워 두었음에 유의하라.



네 번째 상황은 2번 기계에 들어 있는 빵을 뒤집어 주어야 하는 상황이다.



다섯 번째 상황은 손님이 3번 기계부터 4번 기계까지의 빵을 모두 꺼내 버거운 버거를 만들어 달라는 주문이다. 빵을 꺼내서 쌓으면 왼쪽 아래와 같이 되고, 두 개의 빵을 추가해 높이 4의 버거운 버거를 만들 수 있으며, 이보다 높이가 더 낮은 버거운 버거를 만들 수는 없다. 빵을 추가할 때 기존에 쌓아둔 빵과 빵 사이뿐만 아니라 가장 위쪽 혹은 아래쪽에도 넣을 수 있으며, 빵을 넣는 방향은 상관없음에 유의하라.

