

제5회 kriiicon

제한 사항

모든 문제의 메모리 제한은 1024MB입니다.

각 문제의 시간 제한은 아래와 같습니다.

August14 Beginner Contest	: 1s
Donut-shaped Enclosure	: 1.5s
Fully Generate	: 0.5s
hi	: 0.5s
Joyful KMP	: 1s
Logistical Metropolis	: 2s
New Occurrences	: 1s
Polynomial Quine	: 0.5s
Repeating Subsequence Tests	: 1.5s
Unifying Values	: 0.5s
Window XOR	: 1s
Young Zebra	: 0.5s

August14 Beginner Contest

august14는 백준 온라인 저지(BOJ)에서 열심히 문제를 풀었던 사람 중 한 사람이다. 그는 문제를 세 가지 종류로 나눠 판단한다.

1. 푼 문제 : 이런 문제들은 **쉬운 문제**이다.
2. 시도했지만 풀지 못한 문제 : 이런 문제들은 손을 대어 본 문제들로 **어려운 문제**이다.
3. 시도조차 하지 않은 문제 : 이런 문제들은 손도 못 댄 문제들로 **매우 어려운 문제**이다.

august14는 문제 풀기를 갓 시작한 사람들을 위해 쉬운 문제들을 모아 대회를 하나 열고 싶다. 이를 위해 몇몇 문제들을 BOJ에서 뽑아 그 문제들의 난이도를 판별하고 쉬운 문제들을 고르려고 한다. 당신은 이를 도와주는 프로그램을 작성해주면 된다. august14가 갑자기 문제를 풀어서 난이도의 판별 기준이 달라질 수 있음에 조심하라. "<https://www.acmicpc.net/user/august14>"의 목록을 참조하는 것이 가장 정확하다.

입력

입력의 첫 번째 줄에는 august14가 뽑은 문제의 개수 $N(1 \leq N \leq 100,000)$ 이 주어진다.

다음 N 개의 줄의 i 번째 줄에는 august14가 BOJ에서 뽑은 문제의 ID가 정수로 주어진다. 이 정수는 -10^6 이상 10^6 이하의 값을 가진다.

출력

출력은 총 N 개의 줄로 이루어져야 하며, i 번째 줄에는 i 번째로 주어진 문제의 난이도를 출력해야 한다. 이 문제가 august14의 기준으로 쉬운 문제면 “E”를, 어려운 문제면 “H”를, 매우 어려운 문제면 “VH”를 출력하면 된다.

채점 기준

99%이상의 줄이 정답과 같으면 2점을 획득할 수 있다.

모든 줄이 정답과 같으면 7점을 획득할 수 있다.

입력 예제	출력 예제
6	E
1000	E
11392	E
13182	VH
-1111	H
13062	VH
100000	

Donut-shaped Enclosure

2차원 평면에서 보통 사용되는 거리 체계는 유클리드(Euclid) 거리다. 유클리드 거리에서 두 점 (x_1, y_1) 과 (x_2, y_2) 사이의 거리는 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 로 나타난다.

이 문제에서는 유클리드 거리 대신 체비쇼프(Chebyshev) 거리를 다룬다. 체비쇼프 거리에서 (x_1, y_1) 과 (x_2, y_2) 사이의 거리는 $\max(|x_1 - x_2|, |y_1 - y_2|)$ 로 나타난다.

어떤 거리 체계에 대하여 반지름이 r 인 원은 특정한 점 (x_c, y_c) 와의 거리가 r 인 점 (x, y) 들의 집합이다. 원의 형태는 어떤 거리 체계를 쓰느냐 에 따라 다른데, (그림 1)에서 보이듯 유클리드 거리에서는 우리가 보통 아는 원형이고, 체비쇼프 거리에서는 한 변의 길이가 $2r$ 인 정사각형이 원이 된다.

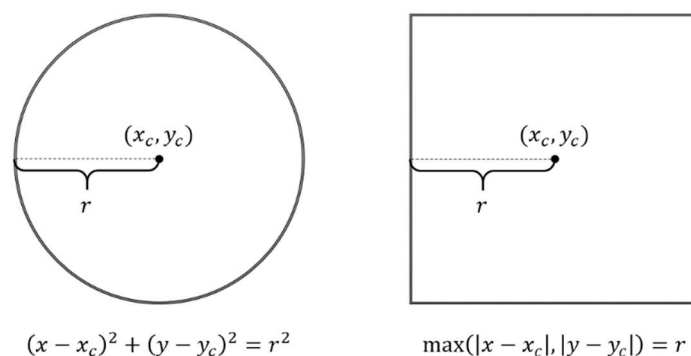


그림 1 : 유클리드 거리와 체비쇼프 거리에서의 원

또한, 어떤 거리 체계에 대하여 안쪽 반지름이 l 이고, 바깥쪽 반지름이 r 인 도넛은 특정한 점 (x_c, y_c) 와의 거리가 l 이상 r 이하인 점 (x, y) 들의 집합이다.

(그림 2)에서 왼쪽은 유클리드 거리에서의 도넛이고, 오른쪽은 체비쇼프 거리에서의 도넛이다. 회색 영역이 도넛에 포함된 영역이다.

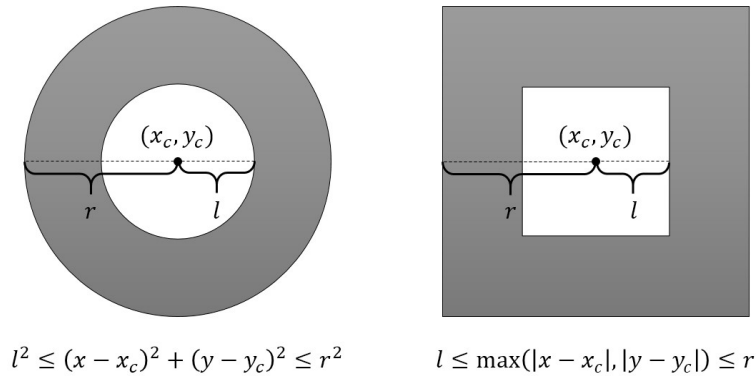


그림 2 : 유클리드 거리와 체비쇼프 거리에서의 도넛

범수는 N 개의 점 P_1, P_2, \dots, P_N 과 체비쇼프 거리에서의 안쪽 반지름이 L 이고 바깥쪽 반지름이 R 인 도넛을 가지고 놓고 있다. P_i 는 (x_i, y_i) 에 위치하며, 도넛 중심의 위치는 중심의 좌표를 격자점에 두는 것만 지키면, 범수가 마음껏 바꿀 수 있다. 만약 P_i 가 도넛 영역에 들어가 있다면, 범수는 강제적으로 S_i 점을 얻게 되고, 도넛 영역에 들어가 있지 않다면 아무 점수도 얻지 않는다. 이 때, 범수가 얻을 수 있는 점수의 최댓값을 구하는 프로그램을 작성하라.

입력

입력의 첫 번째 줄에는 세 정수 $N, L, R (1 \leq N \leq 10^5, 1 \leq L \leq R \leq 10^9)$ 이 공백 하나로 구분되어 주어진다.

다음 N 개의 줄의 i 번째 줄에는 $x_i, y_i, S_i (-10^9 \leq x_i, y_i \leq 10^9, -10^4 \leq S_i \leq 10^4)$ 가 공백 하나로 구분되어 주어진다. 같은 위치에 여러 점이 있을 수 있다.

출력

첫 번째 줄에 범수가 얻을 수 있는 점수의 최댓값을 출력한다.

채점 기준

모든 정수가 1이상 10^3 이하의 수로 주어지는 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
4 1 1 0 1 10 0 -1 10 1 0 -100 -1 0 -100	20
4 1 2 0 1 10 0 -1 10 1 0 -100 -1 0 -100	10

첫 번째 예제는 도넛의 중심을 (1,0)이나 (-1,0)에 두면 된다.

두 번째 예제는 도넛이 너무 커졌기 때문에, -100을 피해서 10하나를 포함시키는 것이 최적이다.

Fully Generate

양의 정수만으로 이루어진 단조 증가 수열 G 가 있다. 이 수열에서 G_i 는 i 가 1 이상의 정수일 때 정의되며, G 에서 i 가 등장하는 횟수를 나타낸다. 정확히 말하면, G 는 i 가 G_i 번 나타나는 수열이어야 한다. $G_1 = 1$ 이며, 이 때 G 는 유일하게 결정된다. G_1 에서 G_{12} 까지를 순서대로 적어보면 다음과 같다.

1, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 6, ...

1이 1번, 2가 2번, 3이 2번, 4가 3번, 5가 3번 등장하는 것을 볼 수 있다. n 이 주어질 때, G_1 에서 G_n 까지의 곱을 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 하나의 정수 $n(1 \leq n \leq 10^{12})$ 이 주어진다.

출력

G_1 에서 G_n 까지의 곱을 출력한다. 이 수가 매우 클 수 있으므로, 1,000,000,007로 나눈 나머지를 출력하도록 한다.

채점 기준

$n \leq 10^6$ 인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
1	1
2	2
3	4
4	12
5	36
6	144
7	576
8	2304
9	11520
10	57600
100	711574837

hi

길이가 L 인 원형 정수열 $[h_1, h_2, \dots, h_L]$ 이 있다고 하자. h_i 와 h_{i+1} 는 인접해 있으며, 특별히 h_L 과 h_1 이 인접해 있는 형태다. 이 원형 정수열의 점수는 같은 수들이 인접한 부분의 크기를 모두 구해서 곱한 것이다. 예를 들어 $h = [1, 1, 2, 2, 1, 3, 3, 3, 1, 1]$ 이면, 점수는 $4 \times 2 \times 1 \times 3 = 24$ 점이 된다.

1이상 N 이하의 정수만 포함하면서, 정수 a 를 정확히 C_a 개 포함하는 모든 서로 다른 원형 정수열에 대해서 점수의 총합을 구하는 프로그램을 작성하라. 두 원형 정수열 h 와 h' 이 다르다는 것은 $h_i \neq h'_i$ 인 i 가 존재한다는 것이다.

입력

첫 번째 줄에 하나의 정수 $N (1 \leq N \leq 50)$ 이 주어진다.

두 번째 줄에는 N 개의 정수 $C_1, C_2, \dots, C_N (1 \leq C_a \leq 100)$ 이 공백 하나로 구분되어 주어진다.

출력

첫 번째 줄에 정수 a 를 정확히 C_a 개 포함하는 모든 서로 다른 원형 정수열에 대해서 점수의 총합을 출력한다. 이 수는 매우 클 수 있으므로, 1,000,000,007로 나눈 나머지를 출력하도록 한다.

채점 기준

$N = 2$ 인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
2 2 2	18
3 2 3 4	7830
4 4 4 4 4	818559048
5 1 2 3 4 5	342934740
6 7 8 9 10 11 12	609539975

첫 번째 예제에서 만들어지는 원형 정수열은 다음의 6가지이다:

[1, 1, 2, 2] : 4점

[1, 2, 1, 2] : 1점

[1, 2, 2, 1] : 4점

[2, 1, 1, 2] : 4점

[2, 1, 2, 1] : 1점

[2, 2, 1, 1] : 4점

그러므로 총 18점이다.

Joyful KMP

최근 KMP(Knuth-Morris-Pratt)알고리즘을 배운 홍준이는 **실패함수**의 이해에 대해서 골몰하고 있다. 어떤 문자열 $S = s_1s_2 \cdots s_N$ 에 대해, 이 문자열의 실패함수는 N 개의 값 $f[1], f[2], \dots, f[N]$ 으로 나타나며, $f[i]$ 는 $s_1s_2 \cdots s_i$ 의 접두사가 되면서 동시에 접미사가 되는 길이 i 미만의 문자열 중 가장 긴 것의 길이이다. 만약 그런 문자열이 없다면 0 을 가진다. 예를 들어, $S = \text{"abcabd"}$ 의 실패함수는 다음과 같다.

i	1	2	3	4	5	6
s_i	a	b	c	a	b	d
$f[i]$	0	0	0	1	2	0

홍준이는 실패함수를 이해하기 위해서 어떤 문자열을 하나 놓고, 이 문자열과 같은 실패함수를 가지는 알파벳 소문자만으로 이루어진 문자열을 모두 구해 사전 순으로 나열하려고 한다. 옆에서 지켜보던 당신은 홍준이를 도와 주어진 문자열과 같은 실패함수를 가지는 문자열의 개수와 그 중 사전 순으로 K 번째인 문자열을 구해주는 프로그램을 작성하기로 했다.

입력

첫 번째 줄에 길이가 1이상 10^6 이하인 문자열이 주어진다. 이 문자열은 알파벳 소문자만으로 이루어져 있다.

두 번째 줄에 하나의 양의 정수 $K(1 \leq K \leq 9 \times 10^{18})$ 가 주어진다.

출력

첫 번째 줄에는 입력으로 주어진 문자열과 같은 실패 함수를 가지는 알파벳 소문자만으로 이루어진 문자열의 개수를 출력한다. 이 수는 매우 클 수 있으므로, 1,000,000,007로 나눈 나머지를 출력하도록 한다.

두 번째 줄에는 입력으로 주어진 문자열과 같은 실패 함수를 가지는 알파벳 소문자만으로 이루어진 문자열 중에서 사전 순으로 K 번째에 오는 문자열을 출력한다. K 가 너무 커서 이런 문자열이 존재하지 않는다면 “OVER”를 출력한다.

채점 기준

입력으로 주어진 문자열과 같은 실패 함수를 가지는 문자열의 개수만 정답과 같으면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
abab 100	650 dzdz
abab 1000	650 OVER

첫 번째 문자와 세 번째 문자가 같고, 두 번째 문자와 네 번째 문자가 같고, 첫 번째 문자와 두 번째 문자가 다른 꼴의 문자열만이 입력으로 주어진 문자열과 실패함수가 같다. 그러므로 $26 \times 25 = 650$ 이 abab와 실패함수가 같은 문자열의 개수이다.

그 중에서 사전 순으로 100번째에 오는 문자열은 dzdz이다.

Logistical Metropolis

우주제국의 황제 명우는 최근 서로 전쟁을 하던 N 개 행성을 어부지리로 점령했다. 이제 이 N 개의 행성에는 평화가 찾아왔다.

여기에 더해서 명우는 N 개 행성 간에 서로 이동이 가능하도록 $N - 1$ 쌍의 워프 게이트를 설치하려고 한다. 하나의 워프 게이트를 설치하면 두 행성 간의 이동이 가능하며, 어떤 행성들을 어떤 방식으로 연결하느냐에 따라 비용의 차이가 있을 수 있다. 명우는 현재 총 M 가지 쌍의 워프 게이트를 연결할 수 있다.

명우는 N 개의 행성 중 하나를 골라 **주요 행성**으로 삼아 자신의 우주 제국과 연결되는 워프 게이트를 만들 것이다. 명우는 선택된 주요 행성과 연결되는 워프 게이트를 비용을 따지지 않고 최대한 많이 만들고 싶다. 각각의 행성을 주요 행성으로 삼을 때 $N - 1$ 개의 워프 게이트를 설치하는 비용의 최솟값을 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 두 정수 N, M ($1 \leq N \leq 100,000$, $N - 1 \leq M \leq 300,000$)이 공백 하나로 구분되어 주어진다.

다음 M 개 줄의 각 줄에는 건설 가능한 워프 게이트의 정보를 나타내는 세 정수 x, y, c ($1 \leq x < y \leq N$, $1 \leq c \leq 10^9$)가 공백 하나로 구분되어 주어진다. 이는 x 번 행성과 y 번 행성을 잇는 워프 게이트를 건설하기 위해 비용이 c 필요하다는 뜻이다. 같은 두 행성에 대한 정보는 여러 번 주어지지 않는다. 모든 워프 게이트를 건설하면, 모든 두 행성간 이동이 가능한 것이 보장된다.

출력

N 개의 줄에 걸쳐 정답을 출력한다. i 번째 줄에는 i 번 행성을 주요 행성으로 삼을 때, 모든 행성간 이동이 가능하도록 $N - 1$ 개의 워프 게이트를 설치하는 비용의 최솟값이 출력되어야 한다.

채점 기준

$N \leq 1,000$, $M \leq 3,000$, $c \leq 10^6$ 인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
4 4	7
1 2 1	6
2 3 2	6
3 4 3	8
1 4 4	

New Occurrences

두 문자열 S 와 P 에 대해, S 에서 P 가 등장하는 횟수를 나타내는 함수 $f(S,P)$ 가 있다. 즉 $f(S,P)$ 는 P 가 S 에서 연속한 부분 문자열로 등장하는 횟수를 나타낸다. 예를 들어 $f("ababa","aba") = 2$ 이고, $f("aaaaa","aa") = 4$ 이다.

여기에 더해서, 어떤 문자열 S 에 대해서 가능한 모든 문자열의 등장횟수 제공의 합을 나타내는 함수 $g(S)$ 가 있다. 즉, $g(S)$ 는 가능한 모든 문자열 P 에 대해서 $[f(S,P)]^2$ 의 합을 구한 값이다.

N 개의 문자로 이루어진 문자열 $S = c_1c_2 \dots c_N$ 이 주어진다. $R_i = g(c_1c_2 \dots c_i)$ (S 의 처음 i 자리에 대한 등장횟수 제공의 합)이라고 하면, R 값의 변화치를 통해 문자열의 뒤쪽에 한 글자가 추가될 때 마다 새롭게 찾을 수 있는 부분문자열이 얼마나 많아지는지 짐작해볼 수 있다. 이를 위해 R_1 에서 R_N 까지를 모두 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 길이가 1이상 10^5 이하인 문자열 S 가 주어진다. S 는 알파벳 소문자만으로 구성된다.

출력

N 개의 줄에 걸쳐 정답을 출력한다. i 번째 줄에는 R_i 만이 출력되어야 한다.

채점 기준

$n \leq 5,000$ 인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
aaa	1 5 14
abab	1 3 8 16
abaaba	1 3 8 16 25 41

Polynomial Quine

정수계수 다항식 $f(x) = a_{N-1}x^{N-1} + \dots + a_1x + a_0$ 가 다음과 같은 두 조건을 모두 만족하면 $N - 1$ 차 다항식 콰인(Quine)이라고 한다.

- 1) $0 \leq i < N$ 인 모든 정수 i 에 대해 $0 \leq a_i < N$.
- 2) $0 \leq i < N$ 인 모든 정수 i 에 대해 $f(i) \equiv a_i \pmod{N}$.

놀랍게도 $1 \leq N \leq 200$ 인 범위에서는 $N - 1$ 차 다항식 콰인을 모두 구해보면 정확히 N 개가 구해진다!

N 이 주어질 때 모든 $N - 1$ 차 다항식 콰인을 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 하나의 정수 $N(1 \leq N \leq 200)$ 이 주어진다.

출력

N 개의 줄에 걸쳐 한 줄에 하나씩 $N - 1$ 차 다항식 콰인을 출력한다. a_{N-1} 에서 a_0 을 공백 하나로 구분하여 출력해야 하며, 같은 다항식을 여러 번 출력하면 안 된다.

다항식을 출력하는 순서는 a_{N-1} 가 작은 순서대로, 만약 a_{N-1} 이 같다면 a_{N-2} 가 작은 순서대로, ...만약 a_{N-1} 에서 a_1 이 모두 같다면 a_0 이 작은 순서대로 출력해야 한다.

채점 기준

N 이 소수(1과 자기 자신으로만 나뉘어 떨어지는 수. 1은 소수가 아니다.)인 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
1	0
2	0 0 1 0
3	0 0 0 1 2 2 2 1 1
4	0 0 0 0 0 2 0 2 2 1 2 1 2 3 2 3

Repeating Subsequence Tests

어떤 문자열의 부분열(subsequence)은, 문자열에서 몇몇 문자를 지웠을 때 (혹은 하나도 지우지 않아도 되고, 모두 지워도 된다.) 얻어지는 문자열이다. 예를 들어, “ababA”의 서로 다른 부분열은 “”(빈 문자열), “a”, “b”, “A”, “ab”, “aa”, “aA”, “ba”, “bb”, “bA”, “aba”, “abb”, “abA”, “aab”, “aaA”, “bab”, “baA”, “bbA”, “abab”, “abaA”, “abbA”, “aabA”, “babA”, “ababA”로 총 24개다. 이 문제에서는 대소문자를 구별하는 것에 유의하라.

길이가 N 인 문자열 S 가 주어진다. 이 때, 이 문자열의 연속한 부분문자열 Q 개에 대해 각각의 서로 다른 부분열의 개수를 구하는 프로그램을 작성하라.

입력을 받는 시간을 줄이기 위해 다음과 같은 의사 난수 생성기(Pseudo-Random Generator)를 사용한다. a_0, b_0, p, q, r 이 주어지면, $i \geq 1$ 일 때 a_i, b_i, x_i, y_i 는 다음과 같이 정의된다. 먼저 $X = 1,000,000,007$ 라고 하자.

$$\begin{aligned} a_i &= (p \cdot a_{i-1} + q \cdot b_{i-1} + r) \bmod X \\ b_i &= (p \cdot b_{i-1} + q \cdot a_{i-1} + r) \bmod X \\ x_i &= \min(a_i \bmod N, b_i \bmod N) + 1 \\ y_i &= \max(a_i \bmod N, b_i \bmod N) + 1 \end{aligned}$$

$u \bmod v$ 는 u 를 v 로 나눈 나머지를 의미한다. 이 때, i 번째 부분 문자열은 S 의 x_i 번째 문자에서 y_i 번째 문자 까지를 포함하는 문자열이다. 의사 난수 생성기가 난수를 생성하는 방식에는 의미가 없으며, 임의의 입력이 주어져도 적절한 시간 내에 정답을 출력해내는 알고리즘이 존재한다.

입력

첫 번째 줄에 문자열 S 가 주어진다. 이 문자열의 길이 N 은 1이상 10^6 이하이며, S 는 알파벳 대문자, 소문자만으로 구성되어 있다.

두 번째 줄에는 6개의 정수 Q, a_0, b_0, p, q, r 이 공백 하나로 구분되어 주어진다. $1 \leq Q \leq 10^6$ 을 만족하며, $0 \leq a_0, b_0, p, q, r < X = 1,000,000,007$ 을 만족한다.

출력

출력이 오래 걸릴 수 있기 때문에, i 번째 부분문자열의 서로 다른 부분열의 개수를 z_i 라고 할 때, $z_i \bmod X (= 1,000,000,007)$ 를 모두 bitwise-XOR하여 첫 번째 줄에 출력하도록 한다.

채점 기준

$N \leq 10^5, Q \leq 10^5$ 인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
ababA 2 4 6 8 10 12	16

$x_1 = 1, y_1 = 5$ 로 계산되어, 문자열 “ababA”의 서로 다른 부분열의 개수를 계산해야 하며, 위에서 구했듯 $z_1 = 24$ 이다.

$x_1 = 3, y_1 = 5$ 로 계산되어, 문자열 “abA”의 서로 다른 문자열의 개수를 계산해야 하며, 문자들이 서로 다르기 때문에 중복이 없어 $z_2 = 8$ 이다.

두 수를 1,000,000,007로 나눈 나머지를 bitwise-XOR하면 16이 되므로, 16을 출력하면 된다.

Unifying Values

N 개의 수로 이루어진 수열이 주어진다. 우리는 이 수열을 **둘 이상**의 연속한 부분으로 나눈 다음 각 부분에 속하는 수들의 합을 구할 수 있다. 이 때, 합을 구한 모든 수가 같아지도록 수열을 나누는 방법의 개수를 구하는 프로그램을 작성하라.

예를 들어 수열이 $[4, -3, 1, 0, 1]$ 이라고 하자. $[4, -3]$, $[1]$, $[0, 1]$ 의 세 부분으로 나누거나, $[4, -3]$, $[1, 0]$, $[1]$ 의 세 부분으로 나누면 모든 부분의 합이 1이 된다. 이외의 방법은 없으므로, 이 입력에 대한 답은 2가 된다.

입력

첫 번째 줄에 하나의 정수 $N(1 \leq N \leq 10^4)$ 이 주어진다.

두 번째 줄에는 수열에 포함된 N 개의 정수가 공백 하나로 구분되어 순서대로 주어진다. 각 정수는 -10^{14} 이상 10^{14} 이하의 정수다.

출력

수열을 나누는 방법의 개수를 출력한다. 이 수가 매우 클 수 있으므로, 1,000,000,007로 나눈 나머지를 출력하도록 한다.

채점 기준

수열에 포함된 정수가 -1 이상 1 이하인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
5 4 -3 1 0 1	2
6 1 1 1 1 1 1	3
4 100 200 300 400	0

Window XOR

길이가 N 인 수열 X 가 주어진다. 1이상 N 이하의 정수 K 가 주어질 때, X 를 한 번 변환하면 수열의 각 값은 다음과 같이 바뀐다. X' 은 변환된 이후의 수열이며, \oplus 는 Bitwise-XOR연산이다.

$$\begin{aligned}
 X'_1 &= X_1 \oplus X_2 \oplus \cdots \oplus X_K \\
 X'_2 &= X_2 \oplus X_3 \oplus \cdots \oplus X_{K+1} \\
 &\vdots \\
 X'_{N-K+1} &= X_{N-K+1} \oplus X_{N-K+2} \oplus \cdots \oplus X_{N-1} \oplus X_N \\
 X'_{N-K+2} &= X_{N-K+2} \oplus X_{N-K+3} \oplus \cdots \oplus X_N \oplus X_1 \\
 &\vdots \\
 X'_{N-1} &= X_{N-1} \oplus X_N \oplus \cdots \oplus X_{K-3} \oplus X_{K-2} \\
 X'_N &= X_N \oplus X_1 \oplus \cdots \oplus X_{K-2} \oplus X_{K-1}
 \end{aligned}$$

조금 더 편하게 표현하자면, $X_{i+N} = X_i$ 으로 봤을 때, $X'_i = X_i \oplus \cdots \oplus X_{i+K-1}$ 인 것이다.

수열 X 와 K 가 주어질 때, 수열 X 를 T 번 변환한 수열을 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 세 정수 N, K, T ($1 \leq K \leq N \leq 10^5, 1 \leq T \leq 10^{18}$)가 공백 하나로 구분되어 주어진다.

두 번째 줄에는 N 개의 정수 X_1, X_2, \dots, X_N ($0 \leq X_i \leq 10^9$)이 공백 하나로 구분되어 주어진다.

출력

주어진 X 를 T 번 변환한 수열을 A 라고 할 때, 첫 번째 줄에 A_1 에서 A_N 까지의 N 개의 정수를 공백 하나로 구분하여 순서대로 출력한다.

채점 기준

$K \leq 3$ 인 범위의 입력만 해결하면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
5 3 1 3 0 2 1 2	1 3 1 0 1
5 3 2 3 0 2 1 2	3 2 0 0 3
5 3 3 3 0 2 1 2	1 2 3 0 2
5 3 10000000000000000000 3 0 2 1 2	1 1 3 1 0

Young Zebra

요즘 젊은 얼룩말 사이에서는, 가족의 흑백 무늬를 새롭게 바꾸는 수술이 대 인기다. 젊은 얼룩말인 경우도 이 수술로 가족의 무늬를 바꾸는 것에 관심이 많다.

이 수술을 받기 전에 수술을 받는 얼룩말이 원하는 크기의 흑백 패턴을 준비해야 하고, 수술은 이 패턴을 무한히 이어 붙인 것을 기준으로 한다. 예를 들어, 2×3크기의 패턴과 이 패턴을 이어 붙인 그림이다.

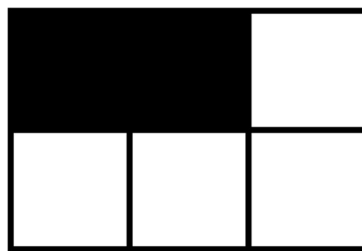


그림 3 : 2×3크기의 패턴

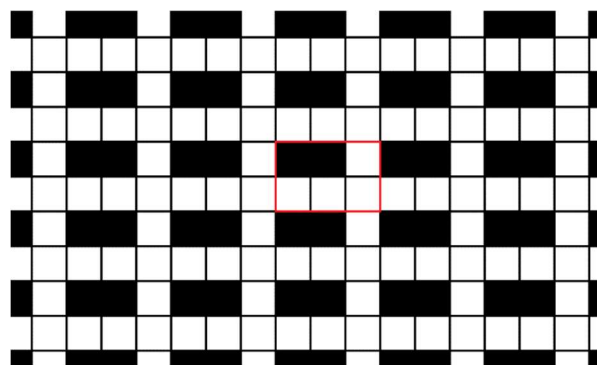


그림 4 : 위의 패턴을 무한히 이어 붙인 것의 일부.

빨간 직사각형은 위의 패턴을 나타내며, 뒤틀림 없이 그대로 맞춰서 이어 붙인다.
격자는 시술할 때 없어지므로 이상하게 시술이 될 것 같다는 걱정은 하지 않으셔도 됩니다.

경우는 먼저 패턴을 정한 후, 무한히 이어 붙여보는 것을 해보려고 한다. 이 때, 중요한 것은 무한히 이어 붙인 다음 각 칸이 이루는 연결 성분의 크기이다. 이것은 미관상 매우 중요한 문제다. 만약 색이 같은 두 칸이 상하좌우로 붙어 있다면 그 두 칸은 같은 연결 성분에 들어가는 것이다.

그림 2에서 각 칸이 속한 연결 성분의 크기를 구해보면, 검은색 칸이 속한 연결 성분의 크기는 모두 2이고, 흰색 칸이 속한 연결 성분에는 무한히 많은 칸이 포함된다. 즉, 크기가 무한대이다. 주어진 패턴을 무한히 이어 붙인 다음 각 칸이 속한 연결 성분의 크기를 구하는 프로그램을 작성하라.

입력

첫 번째 줄에 얼룩 패턴의 크기를 나타내는 두 정수 $N, M (1 \leq N, M \leq 400)$ 이 공백 하나로 구분되어 주어진다.

다음 N 개 줄의 각 줄에는 길이가 M 인 문자열이 주어진다. 모든 문자열은 'B' 또는 'W'로 이루어진 문자열이다. 'B'는 검은색, 'W'는 흰색을 나타낸다.

출력

총 N 개의 줄에 걸쳐 정답을 출력한다. i 번째 줄에는 M 개의 수가 공백 하나로 구분되어 출력되어야 하고, 그 중 j 번째 수는 입력에서 i 번째 행 j 번째 열에 주어진 칸이 속한 연결 성분의 크기여야 한다. 만약 그 크기가 무한히 크다면 -1로 출력해야 한다.

채점 기준

크기가 무한히 큰 연결 성분의 크기는 signed int 범위에 있는 임의의 정수

하나로 출력하고, 크기가 유한한 연결 성분의 크기는 모두 정확히 맞으면 2점을 획득할 수 있다.

모든 입력을 해결하면 7점을 획득할 수 있다.

입력 예제	출력 예제
2 3 BBW WWW	2 2 -1 -1 -1 -1
2 3 BWB WBW	2 1 2 2 1 2
5 5 WBWBW BBWBB WWWWW BBWBB WBWBW	4 12 -1 12 4 12 12 -1 12 12 -1 -1 -1 -1 -1 12 12 -1 12 12 4 12 -1 12 4
5 4 WWBW BWBB BBWW BWBB WWBW	8 8 -1 8 -1 8 -1 -1 -1 -1 2 2 -1 8 -1 -1 8 8 -1 8