

20 KU
22 PC

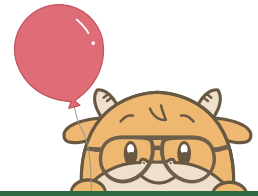
건국대학교 알고리즘 경진대회

Official Problemset

일시 2022년 12월 3일 13:00 ~ 16:00

주최 컴퓨터공학부 17 김명기 *riroan*
18 이승엽 *deLena0702*
19 김태현 *kth990303*
19 이동훈 *aru0504*

후원 김성렬 교수
Chris Ohk *utilforever*



문제 목록

- A 슬픈 건구스
- B 만주의 식사
- C 비숍 여행
- D 시험자리 배정하기
- E 즐거운 XOR
- F 킥보드로 등교하기
- G 보물 찾기 2
- H 볼링 아르바이트
- I 문자열 게임
- J 압도적 XOR 수

모든 문제의 메모리 제한은 512MB로 동일합니다.



문제 A. 슬픈 건구스

시간 제한 1 초
메모리 제한 512 MB

건구스는 슬프다.

건구스의 우는 소리는 정말 특이하다. 그의 울음소리를 글자로 나타내면, **g**로 시작해서 2개 이상의 **o**를 외친 뒤 곧바로 **se**로 끝맺는다.

건구스는 한 번 울 때, 우는 소리는 단 한 번 난다. 즉 **goosegoose**는 건구스의 울음소리가 아니다.

동물들이 한 번 울었을 때 나는 소리들이 주어진다. 그 중 건구스의 울음소리는 총 몇 번 주어졌을까?

입력

첫 번째 줄에 울음소리의 개수 N 이 주어진다. ($1 \leq N \leq 100$)

두 번째 줄부터 N 개의 줄에 걸쳐서 동물들이 한 번 울었을 때 나는 울음소리가 주어진다.

울음소리는 알파벳 소문자로 이루어져 있으며, 길이는 1000 이하이다.

출력

주어진 울음소리 중, 건구스의 울음소리가 몇 번 주어졌는지 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 goose gse goosegoose goooooooooose goosese	2



문제 B. 만주의 식사

시간 제한 1 초
메모리 제한 512 MB

만주의 밥그릇 N 개에 각각 a_1, a_2, \dots, a_N 개의 츄르가 있다. 만주는 매일 밥그릇 중 한 개를 골라 츄르를 한 개 먹는다. 모든 밥그릇에 있는 츄르의 개수가 같아지기 위해서는 최소 며칠이 걸릴까?

입력

첫 번째 줄에 밥그릇의 개수 N 이 주어진다. ($1 \leq N \leq 200000$)

두 번째 줄에 밥그릇에 들어있는 츄르의 개수 a_1, a_2, \dots, a_N 가 공백으로 구분되어 주어진다. ($1 \leq a_i \leq 10^9$)

출력

모든 밥그릇에 있는 츄르의 개수가 같아지기 위해 만주가 먹어야 하는 최소 일수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 1 3 1 3 1	4
1 10	0



문제 C. 비숍 여행

시간 제한 1 초
메모리 제한 512 MB

크기가 무한한 체스판에 비숍의 위치가 주어진다. 체스판 위에 N 개의 동전이 놓여져 있으며 비숍이 동전이 놓인 칸에 도달하면 동전을 획득할 수 있다.

비숍을 임의의 횟수만큼 이동해서 획득할 수 있는 동전의 최대 개수를 구해보자.

단, 비숍은 현재 위치가 (x, y) 라면 $(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)$ 중 한 곳으로 이동할 수 있다.

입력

첫 번째 줄에 비숍이 놓여져 있는 좌표를 나타내는 정수 x, y 가 공백으로 구분되어 주어진다. ($1 \leq x, y \leq 10^9$)

두 번째 줄에 동전의 개수 N 이 주어진다. ($1 \leq N \leq 200000$)

세 번째 줄부터 N 개의 줄에 걸쳐 동전의 좌표를 나타내는 정수 a_x, a_y 가 공백으로 구분되어 주어진다. ($1 \leq a_x, a_y \leq 10^9$)

비숍의 위치와 동전의 위치가 중복되는 경우가 없고 위치가 중복된 동전은 존재하지 않는다.

출력

비숍을 임의의 횟수만큼 이동해서 획득할 수 있는 동전의 최대 개수를 출력한다.

입출력 예시

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



문제 D. 시험자리 배정하기

시간 제한 1 초
메모리 제한 512 MB

n 개의 좌석이 일렬로 배치되어 있고, 이곳에서 k 명이 시험을 보려고 한다. 부정행위를 방지하기 위해 한 사람이 자리에 앉으면 양 옆 자리는 반드시 비워두어야 한다.

위와 같이 좌석을 배정한 경우의 수를 $f(n, k)$ 라고 할 때, $\sum_{i=1}^n f(n, i)$ 의 값을 구해보자.

단, 답이 너무 커질 수 있으니 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입력

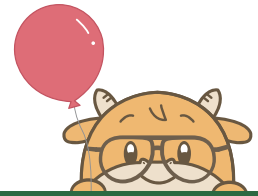
첫 번째 줄에 정수 n 이 주어진다. ($3 \leq n \leq 200000$)

출력

문제에서 주어진 식의 값을 $10^9 + 7$ 로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3	4
10	143



문제 E. 즐거운 XOR

시간 제한 2 초
메모리 제한 512 MB

N 개의 원소로 이루어진 수열 a_1, a_2, \dots, a_N 이 있다.

$1 \leq i < j < k \leq N$ 를 만족하는 (i, j, k) 중에서 $a_i \oplus a_j \oplus a_k$ 의 최댓값을 구해보자.

입력

첫 번째 줄에 원소의 개수 N 이 주어진다. ($3 \leq N \leq 200000$)

두 번째 줄에 N 개의 정수 a_1, a_2, \dots, a_N 이 공백으로 구분되어 주어진다. ($1 \leq a_i \leq 100$)

출력

$a_i \oplus a_j \oplus a_k$ 의 최댓값을 출력한다. ($1 \leq i < j < k \leq N$)

입출력 예시

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

노트

a 와 b 의 bitwise XOR인 $a \oplus b$ 는 2진법으로 표현했을 때 a 와 b 의 i 번째 자리가 같으면 $a \oplus b$ 의 i 번째 자리가 0이고, 서로 다르면 1이 되도록 계산한다.



문제 F. 키패드로 통학하기

시간 제한 2 초
메모리 제한 512 MB

건덕이는 이번 학기 수강신청을 실패해 1교시 수업을 잔뜩 듣게 되었다! 건덕이의 등교시간은 직장인의 출근시간과 겹치는 시간대였기 때문에, 지하철이 아닌 키패드를 구매해서 등교하기로 한다.

키패드는 배터리 1을 소모해 거리 1만큼 이동할 수 있으며, 배터리를 모두 사용하면 반드시 충전해야 한다.

건덕이의 집과 학교는 각각 0, L 에 위치해 있으며, 등갓길에는 총 N 개의 키패드 충전소가 순서대로 위치해 있다. 충전하느라 시간을 낭비한다면 지각할 게 뻔하기 때문에, 건덕이는 등교 중에 최대 K 번 충전하기로 했다. 충전소에 방문하면 키패드의 배터리가 가득 찬다.

키패드의 가격과 배터리 용량은 비례하며, 건덕이는 집에서 키패드를 모두 충전한 상태로 집을 나선다.

건덕이는 조건을 만족하는 키패드 중에서도 가장 싼 키패드를 구매하고자 한다. 건덕이가 구매할 키패드의 배터리 용량을 구해보자.

입력

첫 번째 줄에 학교까지의 거리, 키패드 충전소의 개수, 최대 충전 횟수를 나타내는 세 정수 L, N, K 가 공백으로 구분되어 주어진다.

두 번째 줄에 i 번째 충전소의 위치를 나타내는 N 개의 정수 A_i 가 공백으로 구분되어 주어진다.

- $5 \leq L \leq 200000$
- $3 \leq N \leq \min(L-1, 100000)$
- $0 \leq K \leq N$
- $1 \leq A_i < L$
- $1 \leq i < N$ 인 모든 i 에 대해서 $A_i < A_{i+1}$

출력

건덕이가 구매해야 할 키패드의 배터리 용량을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
10 3 2 1 3 6	4
5 4 2 1 2 3 4	2

노트

첫 번째 예제 입력에서 충전을 위해 방문하는 충전소는 $A_i = 3$ 지점의 충전소와 $A_i = 6$ 지점의 충전소이다.



문제 G. 보물 찾기 2

시간 제한 2.5 초
메모리 제한 512 MB

건덕이는 지난 보물찾기에서 보물을 찾는 데 성공했다. 이제는 배를 타고 세계 곳곳을 누비며 보물을 찾아나서는 보물 탐사대가 되었다.

건덕이는 주변 섬들의 지형이 담긴 가로 W 칸, 세로 H 칸의 지도를 구했다. 지도에는 주변 바다의 지형이 나타나있다. 바다와 암초로 이루어져있는데, 배는 암초 위를 지나다닐 수 없다. 지도의 가장 왼쪽 위는 $(1,1)$, 가장 오른쪽 아래는 (H,W) 이다.

건덕이의 배는 매번 인접한 8칸 중 한 칸으로 이동할 수 있다. (r,c) 와 인접한 칸은 $\max(|r-x|, |c-y|) = 1$ 인 (x,y) 이다. 안전한 항해를 위해 지도 바깥으로는 나가지 않는다.

바다의 물살이 지도 기준 왼쪽에서 오른쪽으로 빠르게 흐르고 있어서, 물살을 타고 가는 데에는 연료가 들지 않지만, 그 외에는 한 칸당 1의 연료가 소모된다. 예를 들어, 건덕이가 현재 (r,c) 에 위치해 있다면 $(r-1,c+1), (r,c+1), (r+1,c+1)$ 로는 연료 소모 없이 이동할 수 있고, 그 외의 칸으로는 1의 연료를 소모해야 한다.

건덕이의 손에 보물지도가 주어졌다. 보물을 찾기까지 소모해야 하는 연료의 최솟값을 구해 주자!

입력

첫 번째 줄에 보물지도의 세로 길이 H , 가로 길이 W 가 주어진다. ($3 \leq H, W \leq 500$)

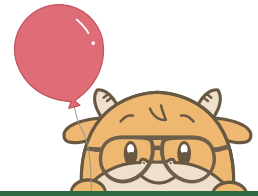
두 번째 줄부터 H 개의 줄에 걸쳐 지도가 주어진다. 지도는 $. \# * K$ 중 하나로만 나타내져 있으며, 각각 바다, 암초, 보물, 배를 의미한다. 배와 보물은 지도에 한 번만 주어진다.

출력

물살을 헤쳐 보물을 찾기까지 소모하는 연료의 최솟값을 출력한다. 도달할 수 없는 경우에는 -1 을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 5* K.#.. #...#	0
3 5K *.#.. #...#	4
3 3 K.. ### *..	-1



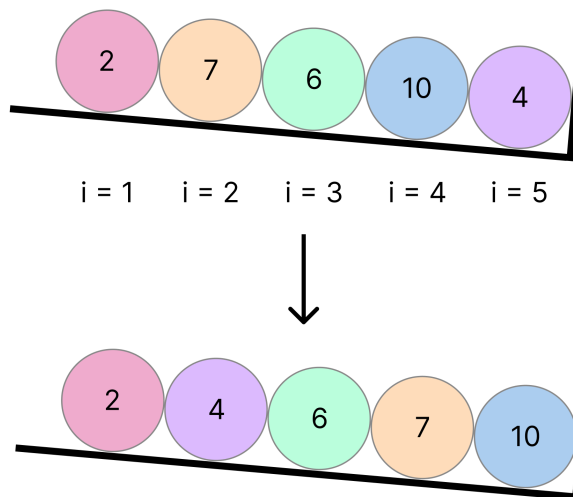
문제 H. 볼링장 아르바이트

시간 제한 1 초
메모리 제한 512 MB

건구스는 볼링장에서 아르바이트를 하고 있다. 건구스의 퇴근 전 마지막 업무는 N 개의 볼링공의 순서를 볼링공의 무게 순서대로 정리하는 것이다. 즉, 모든 i, j ($1 \leq i < j \leq N$)에 대해 $w_i \leq w_j$ 가 되도록 정리한다. 특이하게도 이 볼링장의 볼링공들은 높낮이가 기울어진 레일 위에 놓여있어 레일의 중간에는 볼링공을 넣을 수 없다. 따라서 다음과 같은 방법으로 볼링공을 이동한다.

1. i 번째 볼링공을 꺼낸다. ($1 \leq i \leq N$)
2. 꺼낸 볼링공을 맨 위에 넣는다.

예를 들어 $[2, 7, 6, 10, 4]$ 에서 4번째 볼링공을 이동한다면, $[10, 2, 7, 6, 4]$ 가 된다. 이 볼링공들을 정리한다면, $[2, 4, 6, 7, 10]$ 이 되어야 한다.



건구스는 최대한 일찍 퇴근하기 위해 가장 적은 이동 횟수로 볼링공을 정리하고 싶다. 높은 위치부터 낮은 위치까지 레일에 놓여진 순서대로 i 번째 볼링공 무게 w_i ($1 \leq i \leq N$)가 주어졌을 때, 건구스를 도와 최소 이동 횟수를 구하는 프로그램을 작성하자.

입력

첫째 줄에 볼링공의 개수 N ($1 \leq N \leq 200000$)이 주어진다.

둘째 줄에는 볼링공의 무게를 나타내는 정수 w_i ($1 \leq i \leq N, 1 \leq w_i \leq 1000000$)가 공백으로 구분되어 주어진다.

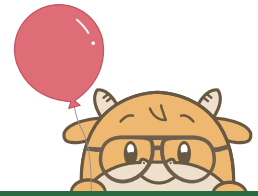
출력

모든 볼링공을 정리한 후, 즉 모든 i, j ($1 \leq i < j \leq N$)에 대해 $w_i \leq w_j$ 가 되도록 하는 최소 이동 횟수를 출력한다.



입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 2 7 6 10 4	3
7 2 3 2 3 1 2 4	3
3 1 2 3	0



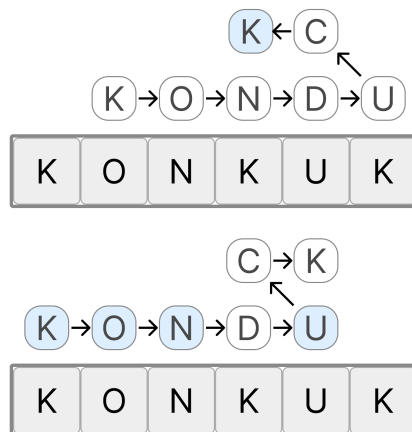
문제 I. 문자열 게임

시간 제한 1 초
메모리 제한 512 MB

건덕이는 문자가 일렬로 적혀있는 보드에서 게임을 하고 있다. 보드에는 N 개의 알파벳 대문자가 나란히 적혀있다. 건덕이는 또 다른 M 자리 영어 단어를 가지고 게임을 진행한다.

우선 보드의 한 지점에서 첫 번째 문자가 보드의 문자와 일치하는지 확인한다. L(왼쪽) 또는 R(오른쪽) 방향으로 이동한 후에 다음 문자와 보드의 문자가 일치하는지 확인한다. 건덕이가 가지고 있는 문장이 종료하면 총 매칭된 문자의 개수를 확인한다.

예를 들어, 보드에 "KONKUK"이라는 문자가 적혀있고, "KONDUCK"이라는 단어로 게임을 시작한다면, 2번째 문자부터 RRRLL 순으로 이동한다면 마지막 1개의 문자만이 매칭된다. 가장 많은 문자와 매칭하기 위해서는 1 번째 문자부터 RRRLLR 순으로 이동하여 4개의 문자가 매칭되는 경우이다.



보드와 가지고 있는 단어가 주어졌을 때, 건덕이가 얻을 수 있는 최대 점수를 구하는 프로그램을 작성해보자.

입력

첫째 줄에 보드의 길이와 건덕이가 가지고 있는 문자열의 길이를 나타내는 정수 N, M 이 공백으로 구분되어 주어진다. ($2 \leq N, M \leq 5000$)

둘째 줄에는 보드의 문자가 순서대로 주어진다.

셋째 줄에는 가지고 있는 문자열이 주어진다.

이때, 두 문자열은 모두 알파벳 대문자로만 구성되어 있다.

출력

첫째 줄에 최대 매칭 횟수를 출력한다.



입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 7 KONKUK KONDUCK	4
2 7 AB AAAAAAA	4
3 10 AAA HELLOWORLD	0

노트

Python3 를 이용하면 시간 초과가 될 수 있으니, PyPy3 를 이용해 주시기 바랍니다.



문제 J. 압도적 XOR수

시간 제한 1 초
메모리 제한 512 MB

양의 정수 A 에 대해 A 보다 큰 2의 제곱수 중 가장 작은 값을 P 라고 할 때, $B = A \oplus (P - 1)$ 라고 하자.

이 때, 양의 정수 K 에 대해 $A \geq B \cdot (2^K - 1)$ 일 경우 A 를 **압도적 XOR수**라고 한다.

N 과 K 가 주어지면 1부터 N 까지의 정수 중 **압도적 XOR수**의 개수를 구해보자.

입력

첫 번째 줄에 N, K 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 10^{15}, 1 \leq K \leq 50$)

출력

1부터 N 까지의 **압도적 XOR수**의 개수를 출력한다.

문제에서 사용되는 모든 정수의 범위는 64비트 정수 이내이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
243 4	22
243 7	7
10000000000000000 2	437050046578689

노트

a 와 b 의 bitwise XOR인 $a \oplus b$ 는 2진법으로 표현했을 때 a 와 b 의 i 번째 자리가 같으면 $a \oplus b$ 의 i 번째 자리가 0이고, 서로 다르면 1이 되도록 계산한다.

두 번째 입력에서 **압도적 XOR수**는 1, 3, 7, 15, 31, 63, 127로 총 7개이다.