

Type

1. [꼬리잡기](#)
2. [도치피치피보족](#)
3. [팩토리얼 LCS](#)
4. [대테러 작전](#)
5. [문자의 값어치](#)
6. [바퀴벌레 잡기](#)

1. 꼬리잡기

유형: DFS/BFS

시간 제한: 1초

메모리 제한: 128MB

문제

창수는 친구들과 꼬리잡기 게임을 진행하였다. 창수가 친구들과 진행하는 꼬리잡기 게임은 매우 독특하다.

먼저 모든 사람들은 $N \times M$ 크기의 격자 맵의 각 칸에 자리를 잡고 각자 임의의 번호가 있는 꼬리를 부여받는다. (중복 번호 가능) 술래가 된 사람은 가장 왼쪽 위 지점에서부터 시작해서 상하좌우를 한 칸씩 움직이면서 사람들의 꼬리를 모으러 돌아다닌다. 상하좌우로 이동할 때 현재 지점에 있는 인원의 꼬리를 떼어야 다음 지점으로 이동할 수 있다. 이 때, 술래가 된 사람은 항상 자신의 꼬리 번호보다 큰 번호의 꼬리만 뺄 수 있다. (단, 자신이 여태까지 모은 꼬리 번호와 중복되는 꼬리는 뺄 수 없다.) 꼬리를 떼었다면 술래의 꼬리 번호는 방금 뺀 꼬리의 번호로 바뀐다.

더 이상 꼬리를 뺄 수 없을 때 게임이 끝나고, 자신의 꼬리를 포함해서 가장 많은 개수의 꼬리를 모은 사람이 해당 게임의 승리자가 된다. 만약 동일한 개수의 꼬리를 모았다면 꼬리 번호의 합이 제일 큰 사람이 승리자가 된다. 첫 판부터 술래가 된 창수는 자신이 얼마나 많은 꼬리를 모을 수 있고 이 때 꼬리 번호 합의 최대값이 궁금해져 동생인 창희에게 쌍안경과 무전기를 주며 옥상에서 사람들이 어디서 어떤 번호의 꼬리를 달고 있는지 살펴보고 최대 몇 개의 꼬리를 모을 수 있고 이 때 꼬리 번호 합의 최댓값을 알려달라고 했다. 우리의 목표는 불쌍한 창희를 도와 이를 알려주도록 하는 것이다.

2	3	4
1	7	5
3	1	6

2	3	4
1	7	5
3	1	6

2	3	4
1	7	5
3	1	6

2	3	4
1	7	5
3	1	6

2	3	4
1	7	5
3	1	6

술래 꼬리번호: 2 술래 꼬리번호: 3 술래 꼬리번호: 4 술래 꼬리번호: 5 술래 꼬리번호: 7

=> 최대 꼬리 개수 : 5

=> 최대 꼬리 번호 합 : 21

입력

첫 줄에 격자 맵의 세로의 길이 N 과 가로 길이 M 이 빈칸을 사이에 두고 입력된다. ($1 \leq N, M \leq 100$)

두 번째 줄부터 N 개의 줄에 걸쳐 각 지점의 꼬리의 숫자 T ($1 \leq T \leq 10000$, 중복 가능)가 빈칸을 사이에 두고 주어진다.

출력

첫째 줄에 창수가 모을 수 있는 꼬리 개수의 최댓값과 꼬리 번호 합의 최댓값을 빈칸을 사이에 두고 출력한다.

예제

```
3 3
2 3 4
1 7 5
3 1 6
```

[입력]

```
5 21
```

[출력]

2. 도치피치피보족

유형: 수학 (행렬, repeated squaring)

시간 제한: 2초

메모리 제한: 256MB

문제



김유정을 좋아하는 컴퓨터과학부 신입생 김피보는 김유정의 요기X 광고를 보다가 문득 피보나치 수열이 생각났다.

피보나치 수열(Fibonacci Sequence)이란

$$F_n := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F_{n-1} + F_{n-2} & \text{if } n > 1. \end{cases}$$

위와 같은 점화식으로 정의되고, 이 수열의 항들을 피보나치 수(Fibonacci Number)라 부른다.

피보나치 수열은 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...

피보나치 수를 하나씩 계산하던 김피보는 원래의 피보나치 수열이 너무 쉽게 느껴져서 피보나치 수열을 아래와 같이 새롭게 정의 하고, 이 수열을 **피보족나치 수열**이라 부르기로 하였다.

$$F_n := \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ F_{n-1} + F_{n-2} + F_{n-3} & \text{if } n > 2 \end{cases}$$

n 이 주어졌을때, **피보족나치 수열**의 n 번째 **피보족나치 수**를 계산해 보자.

입력

첫째 줄에 자연수 n 이 주어진다. ($3 \leq n \leq 1,000,000,000$)

출력

첫째 줄에 피보족나치 수열의 n 번째 피보족나치 수를 1,000,000,007로 나눈 나머지를 출력하시오.

예제

입력	출력
5	7
10	149
123456789	501428683
999999999	992281711

예제 설명) 새롭게 정의한 피보족나치 수열의 값을 나열해보면

0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136, ... 이다.

이 피보족나치 수열의 5 번째 항은 7이고, 10 번째 항은 149이다.

해설

기본 피보나치를 행렬로 나타내면 아래와 같다.

$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}, \quad \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \cdot \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}$$

문제의 피보나치 수열은 $F_{n+3} = F_n + F_{n+1} + F_{n+2}$.

그러므로,

$$\begin{pmatrix} F_{n+3} \\ F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} F_{n+2} \\ F_{n+1} \\ F_n \end{pmatrix}, \quad \begin{pmatrix} F_{n+2} \\ F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^n \cdot \begin{pmatrix} F_2 = 1 \\ F_1 = 1 \\ F_0 = 0 \end{pmatrix},$$

주어지는 수 n 의 범위가 10억까지 이므로 차례로 접근시 시간초과.

$$T = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{ 이라고 하면,}$$

$$T^{10} = T^{1010_2} = T^{8+2} = T^8 \cdot T^2$$

$$T^{15} = T^{1111_2} = T^{8+4+2+1} = T^8 \cdot T^4 \cdot T^2 \cdot T^1 \text{ 과 같이 나타낼 수 있다.}$$

이러한 repeated squaring의 방법을 활용하면

$O(3^3 \log(n))$ 의 시간복잡도로 정답을 구할 수 있다.

3. 팩토리얼 LCS

유형: 동적계획법 (Dynamic Programming, DP)

시간 제한: 1초

메모리 제한: 128MB

문제

LCS(Longest Common Sequence), 즉 “최장 공통 부분수열”은 주어진 여러 개의 수열 모두의 부분 수열이 되는 수열들 중에 가장 길이(수열의 원소 수)가 긴 것을 의미한다. 그리고 구해낸 LCS의 길이를 “LCS 길이”라고 정의한다. 예를 들어, 수열 (4 3 2 1)과 수열(4 2 3 1)에서 공통 부분 수열들은 {(4), (3), (2), (1), (4 3), (4 2), (4 1), (3 1), (2 1), (4 2 1)}이다. 따라서 주어진 두 수열의 LCS는 이 중 가장 길이가 긴 (4 2 1) 이고 그 길이는 3이다.

해당 내용을 바탕으로 주어진 수열들의 “LCS 길이”를 구하는 문제를 풀어보도록 하자.

첫 번째로 두 수 A와 B가 주어진다. 주어진 숫자 A, B의 팩토리얼을 각각 한자리 숫자로 구성된 수열로 간주하여 두 수열의 LCS 길이를 구해보자. 예를 들어 A=9, B=10이라 하였을 때, $9! = 362880$ 이고 $10! = 3628800$ 이다. 따라서 LCS는 362880 이고 LCS 길이는 6이다.

입력

첫 번째 줄에 두 개의 숫자가 입력된다. ($1 \leq A, B \leq 3000$)

출력

주어진 두 수의 팩토리얼 숫자의 LCS 길이를 출력한다.

예제

입력	출력
9 10	6
100 200	113
2999 2999	9198

4. 대테러작전

유형: 최소 신장 트리 (Minimum Spanning Tree, MST)

시간 제한: 1초

메모리 제한: 128MB

문제



현재 대한민국 수도 서울에 테러가 발생했다. 테러 발생지역의 개수는 N 개이고 각각의 좌표는 x , y 좌표와 고도인 z 로 이루어져 있다. 현재 가용한 헬기의 대수는 1대인데 테러발생지역인 두 지점 (x_1, y_1, z_1) 와 (x_2, y_2, z_2) 사이를 이동할 때 드는 비용은

$$\min(|x_1 - x_2|, |y_1 - y_2|, |z_1 - z_2|)$$

이다. 당신의 임무는 헬기를 이용하여 최소비용으로 모든 지점에 도달하는 것이다.

입력

첫째 줄에 테러지점발생 개수인 N ($1 \leq N \leq 100,000$) 과 다음 N 개 줄에는 각 지점의 x, y, z 좌표가 주어진다. ($-100 \leq x, y, z \leq 100$ 인 실수). 한 지점에 두 번이상 테러는 발생하지 않는다. 또한 헬기 출발지점의 좌표는 테러발생지점들 중 하나를 선택해야 한다.

출력

모든 지점에 도달하는데 필요한 최소 비용을 출력한다.

예제

```
5
11 -15 -15
14 -5 -15
-1 -1 -5
10 -4 -1
19 -4 19
```

[입력]

4.00

[출력]

```
3
7 9 -20
8 -15 23
7 3 3
```

[입력]

1.00

[출력]

```
8
1 1 1
-99 -99 -99
-50 50 -50
25 -25 25
-12 13 -14
-2 -5 -5
65 42 23
99 99 99
```

[입력]

125.00

[출력]

5. 문자의 값어치

유형: 자료구조 (직접 주소 테이블)

시간 제한: 1초

메모리 제한: 128MB

문제



조선시대에 살고 있는 돌쇠는 인쇄소에서 활자를 찍는 천민이다. 처음엔 양반들의 학업 증진을 위해 무료로 책을 찍어주었지만, 갈수록 힘이 부치는 탓에 한 글자당 값어치를 매겨 돈을 받기로 하였다. 돌쇠는 프로그래밍을 배운 적이 없기에 여러분들이 도와주어야 한다. 돌쇠가 받는 총 엽전의 양을 구하고 이를 현대 화폐인 won으로 바꾸어 계산해주자.

입력

첫째 줄에 테스트 케이스의 수인 N 과 주어지는 문자의 수인 K 가 이 공백을 사이에 두고 주어진다. ($0 < N \leq 5$, $0 < K \leq 100$). 그리고 K 줄에 각 문자와 해당 값어치가 주어진다. 그리고 정수 M 이 주어지는데, M 줄만큼 문자열이 주어진다. ($0 < M \leq 1000$). 그리고 각각의 문자열은 10000개 이하의 문자로 구성된다. 만약 주어진 문자가 문자열에 포함되어 있지 않다면 0으로 계산한다.

(참고로 100엽전 = 1won이다.)

출력

각 테스트의 케이스의 답은 'a.bb won'으로 표현된다. (예: 82.30 won, 0.01 won)

결과가 음수일 경우 "빚쟁이!" 라고 출력한다.

예제

1

7

a 3

W 10

A 100

, 10

k 7

. 3

I 13

8

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, `and what is the use of a book thought Alice `without pictures or conversation?'

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

[입력]

4.02 won

[출력]

6. 바퀴벌레 잡기

유형: DFS/BFS

시간 제한: 1초

메모리 제한: 128MB

문제



실험실에 바퀴벌레가 출현해 연막탄을 사용해 바퀴벌레를 잡으려 한다. 실험실의 크기는 $N \times M$ 이고, 벽 및 기둥들이 있다. 연막탄은 1 초마다 주위 8 방향으로 한칸씩 퍼져나간다. (물론 벽 및 기둥을 통과할 수는 없고, 벽이나 기둥 안에는 바퀴벌레가 없다.) 연막탄은 강력하여 바퀴벌레는 닿자마자 죽는다고 하자. 연막탄을 최적의 위치에서 터뜨린다고 할 때, 주어진 실험실의 모든 바퀴벌레를 죽이기 위해 필요한 최소 시간을 구하라. (불가능할 수도 있음에 유의하라.)

입력

첫째 줄에 실험실의 크기 N, M ($1 < N, M \leq 1000$)이 공백을 사이에 두고 주어지고, N 줄만큼 빈 공간을 나타내는 'O'와 벽 혹은 기둥을 나타내는 '#'이 길이가 M 인 문자열로 주어진다.

O: 빈 공간, #: 벽, 혹은 기둥

출력

주어진 실험실에서 모든 바퀴벌레를 죽이기 위해 연막탄이 퍼지는데 걸리는 최소 시간 (불가능한 경우 -1 을 출력한다.)

예제

```
6 6
000000
0#0#00
0#00#0
000#0#
00#000
0000#0
```

[입력]

5

[출력]

```
5 6
000000
0#0#00
0#00#0
000#0#
00#000
```

[입력]

-1

[출력]

```
2 2
0#
#0
```

[입력]

-1

[출력]