

## 문제 1. 스프링클러 2: 알팔파의 귀환

입력 파일: sprinklers2.in  
출력 파일: sprinklers2.out  
시간 제한: 2초  
메모리 제한: 256MB

농부 존은  $N \times N$  격자 모양의 작은 농지를 가지고 있다. ( $1 \leq N \leq 2000$ ) 모든  $1 \leq i, j \leq N$ 에 대해, 위에서  $i$  번째 열의 왼쪽에서  $j$  번째 격자 칸을  $(i, j)$ 로 표시한다. 이 농지에 옥수수와 알팔파를 심기 위해, 농부 존은 특수한 형태의 스프링클러를 설치해야 한다.

$(I, J)$ 에 위치한 옥수수 스프링클러는 왼쪽 아래 칸 전부에 (즉,  $I \leq i$ 와  $j \leq J$ 를 만족하는 모든  $(i, j)$  칸) 물을 뿌린다.

$(I, J)$ 에 위치한 알팔파 스프링클러는 오른쪽 위 칸 전부에 (즉,  $i \leq I$ 와  $J \leq j$ 를 만족하는 모든  $(i, j)$  칸) 물을 뿌린다.

한 개 이상의 옥수수 스프링클러가 물을 뿌린 격자 칸에는 옥수수를 재배할 수 있다. 또한 한 개 이상의 알팔파 스프링클러가 물을 뿌린 격자 칸에는 알팔파를 재배할 수 있다. 하지만 두 종류의 스프링클러가 모두 물을 뿌렸거나, 어떤 스프링클러도 물을 뿌리지 않은 격자 칸에는 아무것도 재배할 수 없다.

농부 존이 모든 격자 칸에 작물을 재배할 수 있게 (즉, 정확히 한 종류의 스프링클러로만 물이 뿌려지게) 스프링클러를 설치하는 방법의 수를  $10^9 + 7$ 로 나눈 나머지를 구하여라.

몇몇 격자 칸에는 소가 있다. 이 격자 칸에도 작물을 재배할 수 있지만, 스프링클러를 설치하지는 못한다.

### 입력 형식

첫째 줄에는  $N$ 이 주어진다.

각  $i$ 에 ( $1 \leq i \leq N$ ) 대해,  $i + 1$  번째 줄은 농지의  $i$  번째 행을 나타내는 길이  $N$ 의 문자열이다. 문자열의 각 문자는 'W' (격자 칸에 소가 있음.) 혹은 '.' (격자 칸에 소가 없음.)이다.

### 출력 형식

스프링클러를 설치하는 방법의 수를  $10^9 + 7$ 로 나눈 나머지를 출력하여라.

### 예제

sprinklers2.in	sprinklers2.out
2 .. ..	28

다음은 (1, 1) 위치에 옥수수가 자랄 수 있는 14가지 경우이다.

CC .C CA CC .C CA CA C. CA C. CC .C CC .C  
CC, CC, CC, .C, .C, .C, CA, CA, .A, .A, C., C., .., ..

역자 주: C는 옥수수, A는 알팔파이다.

sprinklers2.in	sprinklers2.out
4 ..W. ..WW WW.. ...W	2304

이는 아래 설명된 첫 번째 서브태스크의 조건을 만족한다.

### 배점

- 3-4번 테스트 케이스는  $N \leq 10$ 을 만족하고, 소가 없는 격자 칸이 최대 10개이다.

- 5-9번 테스트 케이스는  $N \leq 200$ 을 만족한다.
- 10-16번 테스트 케이스는 추가 제한조건이 없다.

## 문제 2. 운동

입력 파일: exercise.in  
출력 파일: exercise.out  
시간 제한: 2초  
메모리 제한: 512MB

농부 존은 소를 위한 아침 운동 계획을 (또) 생각해 냈다!

전과 같이 농부 존의  $N$  마리 ( $1 \leq N \leq 7500$ ) 소는 일렬로 서 있다.  $1 \leq i \leq N$ 을 만족하는  $i$ 에 대해, 왼쪽에서  $i$  번째 소에는 번호  $i$ 번이 붙어있다. 농부 존은 소에게 다음 단계를 처음 상태와 같아질 때까지 반복하라고 시킨다.

- 길이  $N$ 의 순열  $A$ 에 대해, 왼쪽에서  $i$  번째에 있었던 소가 왼쪽에서  $A_i$  번째로 이동한다.

예를 들어  $A = (1, 2, 3, 4, 5)$ 인 경우에는 소는 정확히 한 단계를 실행하고 처음 상태로 돌아온다.  $A = (2, 3, 1, 5, 4)$ 인 경우에는, 소는 정확히 여섯 단계를 실행하고 처음으로 돌아온다. 각 단계에서 소의 번호는 다음과 같다.

- 0번째 단계: (1, 2, 3, 4, 5)
- 1번째 단계: (3, 1, 2, 5, 4)
- 2번째 단계: (2, 3, 1, 4, 5)
- 3번째 단계: (1, 2, 3, 5, 4)
- 4번째 단계: (3, 1, 2, 4, 5)
- 5번째 단계: (2, 3, 1, 5, 4)
- 6번째 단계: (1, 2, 3, 4, 5)

길이  $N$ 의 순열  $A$ 로 가능한  $N!$  가지에 대해 운동 단계가 몇 단계인지를 모두 곱한 값을 계산하여라.

수가 매우 클 수 있기 때문에 답을  $M$ 으로 ( $10^8 \leq M \leq 10^9 + 7$ ,  $M$ 은 소수) 나눈 나머지를 출력하여라.

C++을 사용하는 참가자는 KACTL에 있는 Barrett Reduction으로 알려진 코드를 유용하게 사용 할 수도 있다.  $b > 1$ 이 상수이지만 컴파일 타임에 알려지지 않은 경우,  $a\%b$  연산을 평소보다 몇 배 빠르게 할 수 있다. (유감스럽지만 Java에 대해서는 이런 최적화를 고려하고 있지는 않다.)

```
#include <bits/stdc++.h>
using namespace std;

typedef unsigned long long ull;
typedef __uint128_t L;
struct FastMod {
    ull b, m;
    FastMod(ull b) : b(b), m(ull((L(1) << 64) / b)) {}
    ull reduce(ull a) {
        ull q = (ull)((L(m) * a) >> 64);
        ull r = a - q * b; // can be proven that 0 <= r < 2*b
        return r >= b ? r - b : r;
    }
};
FastMod F(2);
```

```
int main() {  
    int M = 1000000007; F = FastMod(M);  
    ull x = 10ULL*M+3;  
    cout << x << " " << F.reduce(x) << "\n"; // 10000000073 3  
}
```

## 입력 형식

첫째 줄에  $N$ 과  $M$ 이 주어진다.

## 출력 형식

답을 출력하여라.

## 예제

exercise.in	exercise.out
5 1000000007	369329541

$1 \leq i \leq N$ 를 만족하는 각  $i$ 에 대해, 다음 배열의  $i$  번째 원소는 소가  $i$  단계를 실행하면 원래 위치로 돌아온다는 것을 의미한다.  $[1, 25, 20, 30, 24, 20]$ . 답은  $1^1 \cdot 2^{25} \cdot 3^{20} \cdot 4^{30} \cdot 5^{24} \cdot 6^{20} \equiv 369329541 \pmod{10^9 + 7}$  이다.

## 참고 사항

이 문제의 메모리 제한은 512MB이다.

## 배점

- 2번 테스트 케이스는  $N = 8$ 을 만족한다.
- 3-5번 테스트 케이스는  $N \leq 50$ 을 만족한다.
- 6-8번 테스트 케이스는  $N \leq 500$ 을 만족한다.
- 9-12번 테스트 케이스는  $N \leq 3000$ 을 만족한다.
- 13-16번 테스트 케이스는 추가 제한조건이 없다.

## 문제 3. 서커스

입력 파일: circus.in  
출력 파일: circus.out  
시간 제한: 2초  
메모리 제한: 256MB

농부 존의 서커스에 참가하는  $N$  마리 ( $1 \leq N \leq 10^5$ ) 소는 다음 연기를 준비하고 있다. 연기는 정점에 1 번부터  $N$  번까지 번호가 붙어있는 트리에서 진행된다. “시작 상태”는  $1 \leq K \leq N$  을 만족하는 정수  $K$  와 1 번부터  $K$  번까지의 소가 한 정점당 최대 한 마리씩 트리의 정점에 배치된 것이다.

연기하면서 소는 임의의 수 만큼 “동작”을 한다. 한 동작에서 한 마리의 소는 비어있는 인접한 정점으로 움직인다. 유한한 수의 동작을 통해 한 시작 상태에서 다른 시작 상태로 갈 수 있다면 두 시작 상태를 “동등”하다고 한다.

각  $1 \leq K \leq N$  에 대해 동치류의 수, 즉, 어떤 두 시작 상태도 동등하지 않도록 시작 상태를 고를 수 있는 최대 개수를 구하여라. 답이 매우 클 수 있으니  $10^9 + 7$  로 나눈 나머지를 출력하여라.

### 입력 형식

첫째 줄에는  $N$  이 주어진다.

$2 \leq i \leq N$  인  $i$  번째 줄에는  $a_i$  번 정점과  $b_i$  번 정점 사이에 간선이 있다는 의미에  $a_i$  와  $b_i$  가 주어진다.

### 출력 형식

각  $1 \leq i \leq N$  을 만족하는  $i$  에 대해,  $i$  번째 줄에  $K = i$  인 경우에 대한 답을  $10^9 + 7$  로 나눈 나머지를 출력하여라.

### 예제

circus.in	circus.out
5	1
1 2	1
2 3	3
3 4	24
3 5	120

$K = 1$  과  $K = 2$  인 경우에, 모든 시작 상태가 동등하다.

$K = 3$  인 경우에,  $c_i$  를  $i$  번 소의 위치라고 하자.  $(c_1, c_2, c_3) = (1, 2, 3)$  인 상태는  $(1, 2, 5)$  인 상태,  $(1, 3, 2)$  인 상태와 동등하다. 하지만  $(2, 1, 3)$  인 상태와 동등하지는 않다.

circus.in	circus.out
8	1
1 3	1
2 3	1
3 4	6
4 5	30
5 6	180
6 7	5040
6 8	40320

### 배점

- 3-4번 테스트 케이스는  $N \leq 8$  을 만족한다.
- 5-7번 테스트 케이스는  $N \leq 16$  을 만족한다.
- 8-10번 테스트 케이스는  $N \leq 100$  을 만족하고, 트리는 “별” 모양이다. 즉, 정점의 차수가 2 초과인 정점이 하나이다.

- 11-15번 테스트 케이스는  $N \leq 100$ 을 만족한다.
- 16-20번 테스트 케이스는 추가 제한조건이 없다.