문제 1. 항공 노선도

시간 제한: 2초
메모리 제한: 1024MB

Alice는 JOI 왕국에 살고 있다. 그는 IOI 공화국에 사는 Bob을 초대하게 되어 JOI 왕국 내의 항공 노선도를 Bob에게 보내게 되었다. JOI 왕국은 0번 섬부터 $N-1$번 섬까지 총 N 개의 섬으로 이루어진 섬나라이다. JOI 왕국에는 M 개의 항공 노선이 있어서 $i+1$ 번째 $(0 \leq i \leq M-1)$ 항공 노선은 A_i번 섬과 B_i번 섬을 양방향으로 잇는다. 같은 쌍의 섬을 잇는 서로 다른 두 개의 항공 노선은 존재하지 않는다. JOI 왕국에서 IOI 공화국으로 정보를 보내기 위해서는, JOI 왕국이 운영하는 전보를 써야 한다. 하지만 전보를 쓸 때, 정점 번호와 간선 번호가 무작위로 섞일 것이다.

정확히 전보는 다음과 같은 순서로 보내지게 된다. G를 Alice가 보내는 그래프라고 하자. (V를 G의 정점 개수, U를 G의 간선 개수라고 하자.)

- Alice는 G의 정점 개수 V와 간선 개수 U를 지정한다. 또한, G의 각 정점에 $0, 1, \cdots, V-1$의 번호를 G의 각 간선에 $0, 1, \cdots, U-1$의 번호를 각각 붙인다.
- Alice는 인자 $C_0, C_1, \cdots, C_{U-1}$ 및 $D_0, D_1, \cdots, D_{U-1}$을 지정한다. 이 인자는 G의 간선을 의미한다. 즉, 각 j $(0 \leq j \leq U-1)$ 에 대해, G의 간선 j는 정점 C_j와 D_j를 연결한다.
- JOI 왕국에 의해 G의 정점 번호가 무작위로 섞인다. 우선 JOI 왕국은 $0, 1, \cdots, V-1$로 이루어진 순열 $p[0], p[1], \cdots, p[V-1]$을 생성한다. 다음, $C_0, C_1, \cdots, C_{U-1}$을 $p[C_0], p[C_1], \cdots, p[C_{U-1}]$로, $D_0, D_1, \cdots, D_{U-1}$을 $p[D_0], p[D_1], \cdots, p[D_{U-1}]$로 바꾼다.
- 이어서, JOI 왕국에 의해 G의 간선 번호가 무작위로 섞인다. 우선 JOI 왕국은 $0, 1, \cdots, U-1$로 이루어진 순열 $q[0], q[1], \cdots, q[U-1]$을 생성한다. 다음, $C_0, C_1, \cdots, C_{U-1}$을 $C_{q[0]}, C_{q[1]}, \cdots, C_{q[U-1]}$로, $D_0, D_1, \cdots, D_{U-1}$을 $D_{q[0]}, D_{q[1]}, \cdots, D_{q[U-1]}$로 바꾼다.
- V, U의 값과 바뀐 후의 인자 $C_0, C_1, \cdots, C_{U-1}$ 및 $D_0, D_1, \cdots, D_{U-1}$가 Bob에게 전달된다.

단, 이 전보로 보낼 수 있는 그래프는 단순 그래프 뿐이다. 단순 그래프는 다중간선이나 루프가 없는 그래프이다. 즉, 각 i, j $(0 \leq i < j \leq U-1)$에 대해 $(C_i, D_i) \neq (C_j, D_j)$ 와 $(C_i, D_i) \neq (D_j, C_j)$를 만족하며 또한 모든 i $(0 \leq i \leq U-1)$ 에 대해 $C_i \neq D_i$ 를 만족하는 그래프를 보내는 것이 가능하다.

Alice는 가능한 한 적은 정점 수의 그래프로 Bob에게 JOI 왕국 내의 항공 노선도를 보내고 싶다.

Alice와 Bob의 통신을 구현하기 위해 다음 2개의 프로그램을 작성하여라.

1. 첫 번째 프로그램은 JOI 왕국의 섬의 수 N, JOI 왕국의 항공 노선의 수 M, JOI 왕국의 항공 노선을 의미하는 수열 A, B가 주어질 때, Alice가 보내는 그래프 G의 정보를 만든다.
2. 두 번째 프로그램은 Bob이 받은 그래프 G가 주어질 때, JOI 왕국의 항공 노선도를 복원한다.

구현 명세

당신은 파일 두 개를 제출해야 한다.

첫째 파일의 이름은 `Alice.cpp`이다. 이 파일은 Alice의 일을 나타내고 다음 함수를 구현해야 한다. 또한, `Alicelib.h`를 include해야 한다.

- `void Alice(int N, int M, int A[], int B[])`

  이 함수는 각 테스트케이스마다 정확히 한 번 불린다.

  - 인자 `N`은 JOI 왕국의 섬의 수를 의미한다.
  - 인자 `M`은 JOI 왕국의 항공 노선의 수 M을 의미한다.
  - 인자 `A[]`, `B[]`는 길이 M의 배열이며, JOI 왕국의 항공 노선을 의미한다.

함수 Alice 안에서는 다음의 함수를 호출해서 보낼 그래프 G의 정보를 설정한다.

- `void InitG(int V, int U)`
  이 함수를 호출하는 것으로 G의 정점 개수와 간선 개수를 설정한다.
  * 인자 V는 G의 정점 개수를 의미한다. V는 1 이상 1500 이하의 정수여야 한다. 이 범위 밖의 수로 함수를 호출한 경우 **오답** [1]이 된다.
  * 인자 U는 G의 간선 개수를 의미한다. U는 0 이상 $V(V-1)/2$ 이하의 정수여야 한다. 이 범위 밖의 수로 함수를 호출한 경우 **오답** [2]이 된다.
- `void MakeG(int pos, int C, int D)`
  이 함수는 G의 간선을 설정한다.
  * 인자 pos는 G의 정점 번호를 의미한다. pos는 0 이상 $U-1$ 이하의 정수여야 한다. 이 범위 밖의 수로 함수를 호출한 경우 **오답** [3]이 된다. 또한, 같은 인자 pos로 함수를 두 번 이상 호출할 수 없다. 같은 인자로 두 번 이상 호출한 경우 **오답** [4]이 된다.
  * 인자 C와 D는 pos번 간선의 양쪽의 정점을 의미한다. C와 D는 0 이상 $V-1$ 이하의 정수여야 한다. 또한, C $\neq$ D를 만족해야 한다. C 혹은 D가 이 조건을 만족하지 않은 경우, **오답** [5]가 된다.

  여기서 U와 V는 initG에서 설정된 값이다.

함수 Alice 안에서는 함수 InitG를 한 번 호출한 후, 함수 MakeG를 정확히 U번 호출해야 한다. 함수 InitG가 두 번 이상 호출된 경우 **오답** [6]이 된다. 함수 InitG가 호출되기 전에 함수 MakeG가 호출된 경우 **오답** [7]이 된다. 함수 Alice의 실행이 완료될 때 함수 InitG가 호출되지 않은 경우 혹은 MakeG의 호출 횟수가 U가 아니었던 경우 **오답** [8]이 된다. 함수 Alice의 실행이 종료될 때 지정한 그래프 G가 단순 그래프가 아닌 경우, **오답** [9]이 된다.

Alice의 호출이 오답으로 판정된 경우 그 시점에서 프로그램은 종료된다.

둘째 파일의 이름은 Bob.cpp이다. 이 파일은 Bob의 일을 나타내고 다음 함수를 구현해야 한다. 또한, Bob.h를 include해야 한다.

- `void Bob(int V, int U, int C[], int D[])`
  이 함수는 각 테스트케이스마다 정확히 한 번 불린다.
  - 인자 V은 그래프 G의 정점 개수를 의미한다.
  - 인자 U은 그래프 G의 간선 개수를 의미한다.
  - 인자 C[], D[]는 길이 U의 배열이며, 그래프 G를 의미한다.

  함수 Bob 안에서는 다음의 함수를 호출해서 복원한 JOI 왕국의 항공 노선도의 정보를 복원한다.

  - `void InitMap(int N, int M)`
    이 함수를 호출하는 것으로 JOI 왕국의 섬의 수와 JOI 왕국의 항공 노선의 수를 복원한다.
    * 인자 N은 복원한 JOI 왕국의 섬의 수를 의미한다. N은 실제 JOI 왕국의 섬의 수와 일치해야 한다. 일치하지 않을 경우, **오답** [10]이 된다.
    * 인자 M은 복원한 JOI 왕국의 항공 노선의 수를 의미한다. N은 실제 JOI 왕국의 항공 노선의 수와 일치해야 한다. 일치하지 않을 경우, **오답** [11]이 된다.
  - `void MakeMap(int A, int B)`
    이 함수를 호출하는 것으로 JOI 왕국의 항공 노선을 복원한다.
    * 인자 A와 B는 JOI 왕국의 **A**번 섬과 **B**번 섬을 잇는 항공 노선이 존재한다는 것을 의미한다. A와 B는 0 이상 $N-1$ 이하의 정수여야 한다. 또한, A $\neq$ B를 만족해야 한다. A 혹은 B가 이 조건을 만족하지 않은 경우 **오답** [12]가 된다. JOI 왕국의 섬 A와 섬 B를 연결하는 항공 노선이 없을 경우 **오답** [13]이 된다. 또한, 과거에 호출한 것과 같은 항공 노선을 호출해서는 안 된다. MakeMap(A, B)를 호출할 때 이미 MakeMap(A, B) 또는 MakeMap(B, A)를 호출한 경우 **오답** [14]이 된다.

여기서 N은 `initMap`에서 설정된 값이다.

함수 `Bob` 안에서는 함수 `InitMap`을 한 번 호출한 후, 함수 `MakeMap`을 정확히 M번 호출해야 한다. 함수 `InitMap`이 두 번 이상 호출된 경우 **오답** [**15**]이 된다. 함수 InitMap이 호출되기 전에, 함수 `MakeMap`이 호출된 경우 **오답** [**16**]이 된다. 함수 `Bob`의 실행이 완료 될 때, 함수 `InitMap`이 호출되지 않은 경우 혹은 `MakeMap`의 호출 횟수가 M이 아니었던 경우 **오답** [**17**]이 된다. 여기서 M은 `initMap`에서 설정된 값이다.

`Bob`의 호출이 오답으로 판정된 경우 그 시점에서 프로그램은 종료된다.

채점은 다음과 같은 방식으로 진행된다. 오답으로 판정된 경우 그 시점에서 프로그램은 종료된다.

(1) JOI 왕국의 항공 노선도의 정보를 인자로 하여 함수 `Alice`를 한 번 호출한다.

(2) 함수 `Alice`안에서 설정된 그래프를 G라고 하자. G의 정점 번호와 간선 번호를 뒤섞은 그래프를 의미하는 정보를 인자로 하여 함수 `Bob`을 한 번 호출한다.

(3) 채점이 종료된다.

참고 사항

- 당신의 프로그램은 내부에서 사용할 목적으로 함수나 전역변수를 사용할 수 있다. 제출된 파일들은 같이 컴파일되어 하나의 실행 파일이 된다. 모든 글로벌 변수나 함수는 충돌을 피하기 위하여 `static`으로 선언되어야 한다. 채점될 때는, Alice과 Bob에 해당하는 두 프로세스로 나누어서 실행될 것이다. Alice의 프로세스와 Bob의 프로세스는 전역변수를 공유할 수 없다.
- 당신의 프로그램은 표준 입출력을 사용해서는 안 된다. 당신의 프로그램은 어떠한 방법으로도 다른 파일에 접근해서는 안 된다. 단, 당신의 프로그램은 디버그 목적으로 표준 에러출력에 출력할 수 있다.

당신은 대회 홈페이지의 아카이브에서 프로그램을 테스트하기 위한 목적의 샘플 그레이더를 받을 수 있다. 아카이브는 당신의 프로그램의 예제 소스 또한 첨부되어 있다. 샘플 그레이더는 파일 `grader.cpp`이다. 당신의 프로그램을 테스트하기 위해서, `grader.cpp`, `Alice.cpp`, `Bob.cpp`, `Alicelib.h`, `Boblib.h`를 같은 디렉토리 안에 놓고, 컴파일하기 위해 다음 커맨드를 실행하여라.

- `g++ -std=c++14 -O2 -o grader grader.cpp Alice.cpp Bob.cpp`

컴파일이 성공적이면, 파일 `grader`가 생성된다.

실제 그레이더와 샘플 그레이더는 다름에 주의하여라. 샘플 그레이더는 하나의 프로세스에서 실행되며, 입력을 표준 입력으로부터 받고, 출력을 표준 출력에 출력한다.

입력 형식

샘플 그레이더는 표준 입력에서 다음과 같은 형식으로 입력받는다.

- 첫째 줄에는 정수 N, M이 주어진다. 이는 JOI 왕국의 섬의 수가 N이며, JOI 왕국의 항공 노선의 수가 M이라는 것을 의미한다.
- 다음 M 개의 줄에는 M 개의 항공 노선의 정보가 주어진다. M개의 줄 중 $i+1$ 번째 $(0 \leq i \leq M-1)$ 줄에는 두 개의 정수 A_i, B_i가 주어진다. 이는 JOI 왕국의 항공 노선의 정보를 의미한다.

출력 형식

프로그램이 정상적으로 종료되었다면, 샘플 그레이더는 다음과 같은 정보를 표준 출력에 출력한다. (따옴표는 출력하지 않는다.)

- 프로그램의 실행 중에 오답으로 판단된 경우, 오답의 종류를 "`Wrong Answer [1]`"과 같은 형식으로 출력하고, 실행이 종료된다.
- `Alice`와 `Bob` 모두 오답이 아닌 경우, "`Accepted`"를 출력하고 또한 V의 값이 출력된다.

프로그램이 다양한 오답의 종류에 속해 있으면 샘플 그레이더는 그중 하나만 출력할 것이다.

제한

- $1 \leq N \leq 1\ 000$.
- $0 \leq M \leq N(N-1)/2$.
- $0 \leq A_i \leq N-1$ $(0 \leq i \leq M-1)$.
- $0 \leq B_i \leq N-1$ $(0 \leq i \leq M-1)$.
- $A_i \neq B_i$ $(0 \leq i \leq M-1)$.
- $(A_i, B_i) \neq (A_j, B_j)$ 이며, $(A_i, B_i) \neq (B_j, A_j)$ $(0 \leq i < j \leq M-1)$.

서브태스크 1 (22 점)

- $N \leq 22$

서브태스크 2 (15 점)

- $N \leq 40$

서브태스크 3 (63 점)

추가 제한조건이 없다.

배점

- 서브태스크 1 혹은 2에서는, 각 서브태스크의 모든 테스트 케이스를 맞은 경우, 만점이다.
- 서브태스크 3에서는, 모든 테스트 케이스를 맞은 경우, $V - N$의 최댓값을 MaxDiff로 해서, 다음 기준으로 채점한다.
  - $101 \leq$ MaxDiff이면, 0점.
  - $21 \leq$ MaxDiff ≤ 100이면, $13 + \left\lfloor \dfrac{100 - \text{MaxDiff}}{4} \right\rfloor$점. 여기서, x를 넘지 않는 최대의 정수를 $\lfloor x \rfloor$로 표현한다.
  - $13 \leq$ MaxDiff ≤ 20이면, $33 + (20 - \text{MaxDiff}) \times 3$점.
  - MaxDiff ≤ 12이면, 63점.

예제

이 함수는 그레이더의 예제 입력과 해당하는 함수 호출을 보여준다.

예제 입력 1	함수 호출의 예			
	호출	반환값	호출	반환값
	`Alice(...)`			
			`InitG(4,3)`	
				(없음)
			`MakeG(0,0,1)`	
				(없음)
			`MakeG(1,0,2)`	
				(없음)
			`MakeG(2,0,3)`	
`4 3`				(없음)
`0 1`		(없음)		
`0 2`	`Bob(...)`			
`0 3`			`InitMap(4,3)`	
				(없음)
			`MakeMap(0,1)`	
				(없음)
			`MakeMap(0,2)`	
				(없음)
			`MakeMap(0,3)`	
				(없음)
		(없음)		

이 때, `Alice(...)`, `Bob(...)`에 주어진 인자는 다음과 같다.

인자	Alice(...)	Bob(...)
N	4	
M	3	
V		4
U		3
A	{0,0,0}	
B	{1,2,3}	
C		{2,2,2}
D		{3,0,1}

예제 입력 2

`5 7`
`0 1`
`0 2`
`1 3`
`1 4`
`3 4`
`2 3`
`2 4`

문제 2. 비타로의 파티

시간 제한: 2초
메모리 제한: 512MB

비버가 사는 마을 N개가 높이가 감소하는 순으로 1번부터 N번까지 번호가 붙어있다. 어떤 두 마을도 같은 높이에 있지 않다. 서로 다른 마을을 단방향으로 잇는 수로 M 개가 존재한다. i 번째 $(1 \leq i \leq M)$ 수로는 S_i 번 마을에서 E_i번 마을로 흐른다. 이 수로는 높은 마을에서 낮은 마을로 흐른다. 수로를 거슬로 올라갈 수는 없다.

비버 비타로는 N 명의 친구가 있고 N 개의 마을에 각각 한 마리씩 살고 있다.

비타로는 친구를 초대해서 파티를 Q 번 할 것이다. j 번째 $(1 \leq j \leq Q)$ 파티에는 Y_j 마리의 비버가 너무 바빠서 참석하지 못한다. j 번째 파티는 T_j번 마을에서 열리기 때문에, 수로를 통해서 T_j번 마을로 올 수 없는 친구들도 참석할 수 없다. 다른 친구들은 파티에 참석한다.

각 친구는 수로를 통해서 파티가 개최되는 마을로 이동한다. 이동하는 경로가 여러 개 존재할 수도 있다. 하지만 비타로의 친구들은 수로를 너무 좋아해서, 여러 개의 경로가 있는 경우에는 거치는 수로의 개수가 최대가 되는 경로로 이동한다.

비타로는 각각의 파티에 대해, 가장 많은 수의 수로를 거친 친구가 몇 개의 수로를 경유했는지가 궁금했다.

입력 형식

표준 입력에서 다음 입력이 주어진다.

- 첫째 줄에는 세 정수 N, M, Q가 공백으로 구분되어 주어진다. 이는 마을이 N 개 있고, 수로가 M 개 있으며, 비타로가 Q 번의 파티를 연다는 것을 의미한다.
- 다음 M 개의 줄에는 수로의 정보가 주어진다. M 개의 줄의 i 번째 $(1 \leq i \leq M)$ 줄에는, 정수 S_i, E_i가 공백으로 구분되어 주어진다. 이는 S_i번 마을 부터 E_i번 마을까지 단방향인 수로가 흐르고 있다는 것을 의미한다.
- 다음 Q개의 줄에는 비타로가 여는 파티의 정보가 주어진다. Q 개의 줄의 j 번째 $(1 \leq j \leq Q)$ 줄에는, 정수 T_j, Y_j와 Y_j개의 정수 $C_{j,1}, C_{j,2}, \cdots, C_{j,Y_j}$가 공백으로 구분되어 주어진다. 이는, j 번째 파티는 T_j번 마을에서 열리고, $C_{j,1}, C_{j,2}, \cdots, C_{j,Y_j}$번 마을에 사는 친구는 너무 바빠서 참석할 수 없다는 것을 의미한다.

출력 형식

표준 출력으로 Q개의 줄을 출력하여라. j 번째 $(1 \leq j \leq Q)$ 줄은 j 번째 파티에 대해 가장 많은 수의 수로를 경유한 친구가 거친 수로의 개수를 출력하여라. 단, 파티에 친구가 한 명도 참석하지 않을 경우 대신 `-1`을 출력하여라.

제한

- $1 \leq N \leq 100\ 000$.
- $0 \leq M \leq 100\ 000$.
- $1 \leq Q \leq 100\ 000$.
- $1 \leq S_i < E_i \leq N$ $(1 \leq i \leq M)$.
- $(S_i, E_i) \neq (S_j, E_j)$ $(1 \leq i < j \leq M)$.
- $1 \leq T_j \leq N$ $(1 \leq j \leq Q)$.
- $0 \leq Y_j \leq N$ $(1 \leq j \leq Q)$.

- $1 \leq C_{j,1} < C_{j,2} < \cdots < C_{j,Y_j} \leq N$ $(1 \leq j \leq Q)$.
- $Y_1 + Y_2 + \cdots + Y_Q \leq 100\ 000$.

서브태스크 1 (7 점)

- $N \leq 1\ 000$.
- $M \leq 2\ 000$.
- $Q = 1$.

서브태스크 1 (7 점)

- $Q = 1$.

서브태스크 3 (86 점)

추가 제한조건이 없다.

예제

test	answer
5 6 3 1 2 2 4 3 4 1 3 3 5 4 5 4 1 1 5 2 2 3 2 3 1 4 5	1 3 0

첫 번째 파티에 참석한 친구 (2번, 3번, 4번 마을에 사는 친구) 중에서는 2번 마을에 사는 친구와 3번 마을에 살고있는 친구가 파티에 개최되는 4번 마을까지 거친 수로의 수가 제일 많다. 이는 1개이므로, 1을 출력한다.

두 번째 파티에 참석한 친구 (1번, 4번, 5번 마을에 사는 친구) 중에서는 1번 마을에 사는 친구가 파티에 개최되는 5번 마을까지 거친 수가 제일 많다. 이는 3개이므로, 3을 출력한다.

세 번째 파티에 참석하는 친구는, 파티가 개최되는 2번 마을에 사는 친구밖에 없다. 이 친구는 수로를 경유하지 않으므로, 0을 출력한다.

test	answer
12 17 10 1 2 2 3 3 4 1 5 2 6 3 7 4 8 5 6 6 7 7 8 5 9 6 10 7 11 8 12 9 10 10 11 11 12 6 3 1 7 12 3 7 1 2 3 4 5 6 7 11 3 1 3 5 9 2 1 9 8 4 1 2 3 4 1 1 1 12 0 10 3 1 6 10 11 8 2 3 5 6 7 9 10 11 8 7 2 3 4 5 6 7 8	1 -1 3 1 3 -1 5 2 4 4

문제 3. 보안 게이트

입력 파일:	`standard input`
출력 파일:	`standard output`
시간 제한:	5초
메모리 제한:	1536MB

당신은 Just Odd Invention이라는 회사를 아는가? 이 회사의 일은 "그저 기묘한 발명 (just odd invetion)"을 하는 것이다. 줄여서 JOI 회사라고 부른다.

JOI 회사에서는 기밀정보를 지키기 위해 입구에 보안 게이트를 설치했다. 회사를 출입하기 위해서는 반드시 보안 게이트를 거칠 필요가 있다. 또한, 한 번에 두 명 이상의 사람이 보안 게이트를 지날 수는 없다.

이 보안 게이트는 사람이 게이트를 통과할 때 사람이 회사로 들어왔는지 혹은 나왔는지에 대한 정보를 기록한다. 이제 JOI 회사의 사원 IOI군이 어떤 날의 게이트 기록을 보았다. 기록은 문자열 S로 표현된다. S의 i 번째 문자가 '('인 경우, 게이트를 i 번째로 통과한 사람이 회사로 들어갔다는 의미이고, S의 i 번째 문자가 ')'인 경우, 게이트를 i 번째로 통과한 사람이 회사에서 나왔다는 의미이다. IOI군은 이날의 시작 혹은 마지막 날의 JOI 회사 안에 아무도 없다는 것을 알고 있다. '(' 과 ')'로만 이루어져 있는데, 기록으로 등장할 수 없는 문자열이 있을 수 있음에 유의하여라. 예를 들어, `())(` 혹은 `(()`같은 문자열은 기록으로 등장할 수 없는데, JOI 회사 안에 있는 사람의 수가 음수가 되어야 하거나, 이날의 시작 혹은 마지막 날의 JOI 회사 안에 사람이 있었다는 의미이기 때문이다.

IOI군이 기록을 확인한 순간 S는 JOI 회사에 퍼져나간 바이러스로 인해 바뀌었다! 조사 이후에, 그는 수정이 다음과 같은 절차로 이루어져 있다고 가정했다:

- 처음에 S의 연속된 구간에 있는 문자열이 모두 다음과 같이 바뀌었다. 그 구간에 포함된 문자 전부에 대해 각 문자가 '('이라면 ')'로 바뀌고, ')'이라면 '('로 바뀌었다. 변화 후의 문자열을 S'이라고 하자. 여기서 변화한 구간의 길이가 0일 수도 있다, 즉, $S = S'$일 수도 있다.
- 다음에 S'의 0개 이상의 문자가 'x'로 바뀌었다. 변화 후의 문자열은 S''이다.

IOI군은 S의 정보를 기억하고 있지 않기 때문에, S''으로부터 S를 복원하려고 생각하고 있다. 이를 위해 IOI군은 우선 S'으로 가능한 문자열의 경우의 수를 세고 싶다. (S가 아님에 주의하여라.)

입력 형식

표준 입력에서 다음 입력이 주어진다.

- 첫째 줄에는 정수 N이 주어진다. 이는 S''의 길이가 N이라는 의미이다.
- 다음 줄에는 각 문자가 '(', ')' 혹은 'x'인 문자열 S''이 주어진다.

출력 형식

표준 출력으로 한 개의 줄을 출력하여라. 출력은 S'으로 가능한 문자열의 경우의 수를 1 000 000 007 로 나눈 나머지여야 한다. 만약 해당하는 S'이 없으면 0을 출력하여라.

제한

- $1 \leq N \leq 300$.

서브태스크 1 (4 점)

- $N \leq 100$.
- S''의 x의 개수는 최대 4개이다.

서브태스크 2 (8 점)

- $N \leq 100$.
- S''의 x의 개수는 최대 12개이다.

서브태스크 3 (18 점)

- $N \leq 100$.
- S''의 x의 개수는 최대 20개이다.

서브태스크 4 (43 점)

- $N \leq 100$.

서브태스크 5 (27 점)

추가 제한조건이 없다.

예제

standard input	standard output
4 x))x	3

이 입력 예제에서, $S' =$))((은 S가 될 수 있는 문자열이 존재하지 않기 때문에 불가능하다. 다음 세 문자열이 S'이 될 수 있다.

- $S' =$ ())(. 예를 들어, $S =$ ()()
- $S' =$ ()). 예를 들어, $S =$ ()()
- $S' =$)))). 예를 들어, $S =$ (())

이 세 문자열만 S'이 될 수 있으므로, 3을 출력한다.

standard input	standard output
10 xx(xx()x(x	45
5 x))x(	0
10 xxxxxxxxxx	684