

TEST 7

1. 문자열로 구성된 리스트 strings와, 정수 n이 주어졌을 때, 각 문자열의 인덱스 n번째 글자를 기준으로 오름차순 정렬하려 합니다. 예를 들어 strings가 ["sun", "bed", "car"]이고 n이 1이면 각 단어의 인덱스 1의 문자 "u", "e", "a"로 string을 정렬합니다.

제한사항

- ▶ strings는 길이 1 이상, 50이하인 배열입니다.
- ▶ strings의 원소는 소문자 알파벳으로 이루어져 있습니다.
- ▶ strings의 원소는 길이 1 이상, 100이하인 문자열입니다.
- ▶ 모든 strings의 원소의 길이는 n보다 큼니다.
- ▶ 인덱스 1의 문자가 같은 문자열이 여럿 일 경우, 사전순으로 앞선 문자열이 앞쪽에 위치합니다.

입출력 예

s	n	return
["sun", "bed", "car"]	1	["car", "bed", "sun"]
["abce", "abcd", "cdx"]	2	["abcd", "abce", "cdx"]

#2>

abce와 abcd, cdx의 2번째 인덱스 값은 c, c, x 입니다. 따라서 정렬 후에는 cdx가 가장 뒤에 위치합니다. abce와 abcd는 사전순으로 정렬하면 abcd가 우선하므로, 답은 abcd, abce, cdx 입니다.

TEST 7

2. 카카오톡에 뜬 네 번째 별! 심심할 땐? 카카오톡 게임별~



카카오톡 게임별의 하반기 신규 서비스로 다트 게임을 출시하기로 했다. 다트 게임은 다트판에 다트를 세 차례 던져 그 점수의 합계로 실력을 겨루는 게임으로, 모두가 간단히 즐길 수 있다.

갓 입사한 진하는 코딩 실력을 인정받아 게임의 핵심 부분인 점수 계산 로직을 맡게 되었다. 다트 게임의 점수 계산 로직은 아래와 같다.

1. 다트 게임은 총 3번의 기회로 구성되며, 각 기회마다 얻을 수 있는 점수는 0 ~ 10점까지이다.
2. 점수와 함께 Single (S), Double (D), Triple (T) 영역이 존재하고 각 영역 당첨 시 점수에서 1제곱, 2제곱, 3제곱 (점수¹, 점수², 점수³)으로 계산된다.
3. 옵션으로 스타상 (*), 아차상 (#)이 존재하며 스타상 (*) 당첨 시 해당 점수와 바로 전에 얻은 점수를 각 2배로 만든다. 아차상 (#) 당첨 시 해당 점수는 마이너스된다.
4. 스타상 (*)은 첫 번째 기회에서도 나올 수 있다. 이 경우 첫 번째 스타상 (*)의 점수만 2배가 된다. 그리고 스타상 (*)의 효과는 다른 스타상 (*)의 효과와 중첩될 수 있다. 이 경우 중첩된 스타상 (*) 점수는 4배가 된다.
5. 스타상 (*)의 효과는 아차상 (#)의 효과와 중첩될 수 있다. 이 경우 중첩된 아차상 (#)의 점수는 -2배가 된다.
6. Single (S), Double (D), Triple (T)은 점수마다 하나씩 존재한다.
7. 스타상 (*), 아차상 (#)은 점수마다 둘 중 하나만 존재할 수 있으며, 존재하지 않을 수도 있다.

0 ~ 10의 정수와 문자 S, D, T, *, #로 구성된 문자열이 입력될 때 총점수를 반환하는 함수를 작성해주세요.

TEST 7

제한사항

▶ “점수|보너스[옵션]”으로 이루어진 문자열 3세트. Ex) 1S2D*3T

- > 점수는 0에서 10 사이의 정수이다.
- > 보너스는 S, D, T 중 하나이다.
- > 옵션은 *이나 # 중 하나이며, 없을 수도 있다.

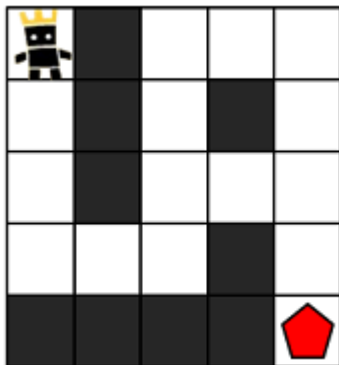
입출력 예

dartResult	answer	설명
1S2D*3T	37	$1^1 * 2 + 2^2 * 2 + 3^3$
1D2S#10S	9	$1^1 + 2^1 * (-1) + 10^1$
1D2S0T	3	$1^2 + 2^1 + 0^3$
1S*2T*3S	23	$1^1 * 2 * 2 + 2^3 * 2 + 3^1$
1T2D3D#	-4	$1^3 + 2^2 + 3^2 * (-1)$

TEST 7

3. ROR 게임은 두 팀으로 나누어서 진행하며, 상대 팀 진영을 먼저 파괴하면 이기는 게임입니다. 따라서 각 팀은 상대 팀 진영에 최대한 빨리 도착하는 것이 유리합니다.

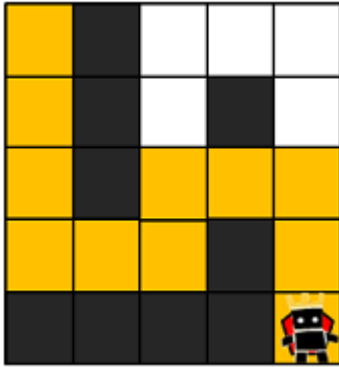
지금부터 당신은 한 팀의 팀원이 되어 게임을 진행하려고 합니다. 다음은 5 X 5 크기의 맵에, 당신의 캐릭터가 (행 : 1, 열 : 1) 위치에 있고, 상대 팀 진영은 (행 : 5, 열 : 5) 위치에 있는 경우의 예시입니다.



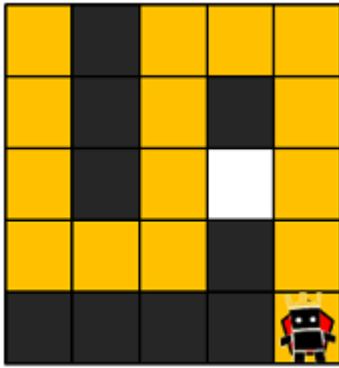
위 그림에서 검은색 부분은 벽으로 막혀있어 갈 수 없는 길이며, 흰색 부분은 갈 수 있는 길입니다. 캐릭터가 움직일 때는 동, 서, 남, 북 방향으로 한 칸씩 이동하며, 게임 맵을 벗어난 길은 갈 수 없습니다. 아래 예시는 캐릭터가 상대 팀 진영으로 가는 두 가지 방법을 나타내고 있습니다.

TEST 7

> 첫 번째 방법은 11개의 칸을 지나서 상대 팀 진영에 도착했습니다.

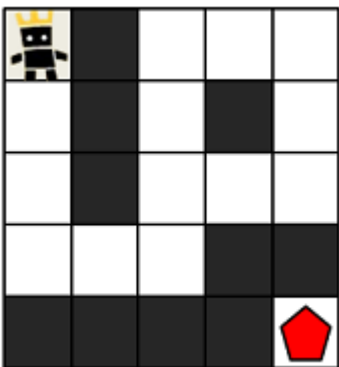


> 두 번째 방법은 15개의 칸을 지나서 상대 팀 진영에 도착했습니다.



위 예시에서는 첫 번째 방법보다 더 빠르게 상대팀 진영에 도착하는 방법은 없으므로, 이 방법이 상대 팀 진영으로 가는 가장 빠른 방법입니다.

만약, 상대 팀이 자신의 팀 진영 주위에 벽을 세워두었다면 상대 팀 진영에 도착하지 못할 수도 있습니다. 예를 들어, 다음과 같은 경우에 당신의 캐릭터는 상대 팀 진영에 도착할 수 없습니다.



게임 맵의 상태 maps가 매개변수로 주어질 때, 캐릭터가 상대 팀 진영에 도착하기 위해서 지나가야 하는 칸의 개수의 최솟값을 return하도록 solution 함수를 완성해주세요. 단 상대 팀 진영에 도착할 수 없을 때는 -1을 return 해주세요.

TEST 7

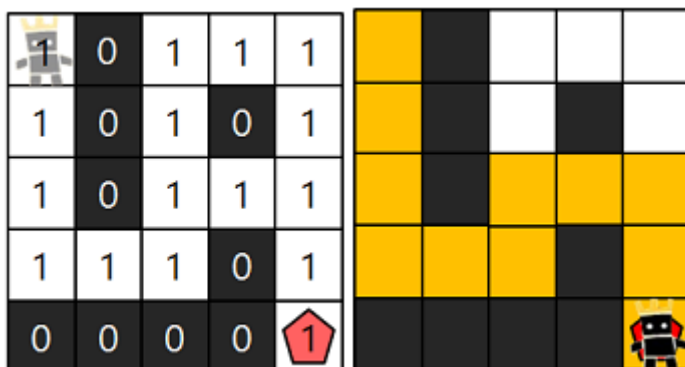
제한사항

- ▶ maps는 $n \times m$ 크기의 게임 맵의 상태가 들어있는 2차원 배열로, n 과 m 은 각각 1 이상 100 이하의 자연수입니다.
 > n 과 m 은 서로 같을 수도, 다를 수도 있지만, n 과 m 이 모두 1인 경우는 입력으로 주어지지 않습니다.
- ▶ maps는 0과 1로만 이루어져 있으며, 0은 벽이 있는 자리, 1은 벽이 없는 자리를 나타냅니다.
- ▶ 처음에 캐릭터는 게임 맵의 좌측 상단인 $(1, 1)$ 위치에 있으며, 상대방 진영은 게임 맵의 우측 하단인 (n, m) 의 위치에 있습니다.

입출력 예

maps	answer
[[1, 0, 1, 1, 1], [1, 0, 1, 0, 1], [1, 0, 1, 1, 1], [1, 1, 1, 0, 1], [0, 0, 0, 0, 1]]	11
[[1, 0, 1, 1, 1], [1, 0, 1, 0, 1], [1, 0, 1, 1, 1], [1, 1, 1, 0, 0], [0, 0, 0, 0, 1]]	-1

#1> 주어진 데이터는 다음과 같으며, 캐릭터가 적 팀의 진영까지 이동하는 가장 빠른 길은 옆의 그림과 같습니다.



따라서 총 11칸을 캐릭터가 지나갔으므로 11을 return 하면 됩니다.

#2> 문제의 예시와 같으며, 상대 팀 진영에 도달할 방법이 없습니다. 따라서 -1을 return합니다.