

[1차] 프렌즈4블록

Python3

문제 설명

프렌즈4블록

블라인드 공채를 통과한 신입 사원 라이언은 신규 게임 개발 업무를 맡게 되었다. 이번에 출시할 게임 제목은 "프렌즈4블록". 같은 모양의 카카오프렌즈 블록이 2×2 형태로 4개가 붙어있을 경우 사라지면서 점수를 얻는 게임이다.



만약 판이 위와 같이 주어질 경우, 라이언이 2×2 로 배치된 7개 블록과 콘이 2×2 로 배치된 4개 블록이 지워진다. 같은 블록은 여러 2×2 에 포함될 수 있으며, 지워지는 조건에 만족하는 2×2 모양이 여러 개 있다면 한꺼번에 지워진다.



블록이 지워진 후에 위에 있는 블록이 아래로 떨어져 빈 공간을 채우게 된다.



만약 빈 공간을 채운 후에 다시 2×2 형태로 같은 모양의 블록이 모이면 다시 지워지고 떨어지고를 반복하게 된다.



위 초기 배치를 문자로 표시하면 아래와 같다.

```
TTTANT
RRFACC
RRRFCC
TRRRAA
TTMMMF
TMMTTJ
```

각 문자는 라이언(R), 무지(M), 어피치(A), 프로도(F), 네오(N), 튜브(T), 제이지(J), 콘(C)을 의미한다

입력으로 블록의 첫 배치가 주어졌을 때, 지워지는 블록은 모두 몇 개인지 판단하는 프로그램을 제작하라.

입력 형식

- 입력으로 판의 높이 m , 폭 n 과 판의 배치 정보 `board`가 들어온다.
- $2 \leq n, m \leq 30$
- `board`는 길이 n 인 문자열 m 개의 배열로 주어진다. 블록을 나타내는 문자는 대문자 A에서 Z가 사용된다.

출력 형식

입력으로 주어진 판 정보를 가지고 몇 개의 블록이 지워질지 출력하라.

입출력 예제

m	n	board	answer
4	5	["CCBDE", "AAADE", "AAABF", "CCBBF"]	14
6	6	["TTTANT", "RRFACC", "RRRFCC", "TRRRAA", "TMMMF", "TMMTTJ"]	15

예제에 대한 설명

- 입출력 예제 1의 경우, 첫 번째에는 A 블록 6개가 지워지고, 두 번째에는 B 블록 4개와 C 블록 4개가 지워져, 모두 14개의 블록이 지워진다.
- 입출력 예제 2는 본문 설명에 있는 그림을 옮긴 것이다. 11개와 4개의 블록이 차례로 지워지며, 모두 15개의 블록이 지워진다.

[해설 보러가기](#)

solution.py

```

1 def solution(m, n, board):
2     board = list(map(list, board))
3     delete = 0
4     while (1):
5         check = [[0 for _ in range(n)] for _ in range(m)]
6
7         for i in range (m-1):
8             for j in range (n-1):
9                 if board[i][j] == board[i+1][j] == board[i][j+1] == board[i+1][j+1] and board[i][j] != 0:
10                     check[i][j] = 1
11                     check[i+1][j] = 1
12                     check[i][j+1] = 1
13                     check[i+1][j+1] = 1
14
15         deleteBefore = delete
16         for i in range(m):
17             for j in range (n):
18                 if check[i][j] == 1:
19                     board[i][j] = 0
20                     delete +=1
21         if deleteBefore == delete:
22             return delete
23         for i in range (1,m):
24             for j in range (n):
25                 if board[i][j] == 0:
26                     count = 0
27                     while(1):
28                         if i-count-1 < 0:
29                             break
30                         board[i-count][j] = board[i-count-1][j]
31                         board[i-count-1][j] = 0
32                         count += 1

```

리스트 변경

인덱스 -1 씩 개리만 앞전시켜
인덱스의 오류를 막고, 인접여부를 확인할 수 있으면 check에 표시.
빈칸 제거 이유는 이렇듯 이렇게 정하는 것만 때문.

삭제가 되었을 때
안된 것은 삭제 안해.

Check 표시 되었을 board를 0으로 변경.

0은 원본 제거함.

실행 결과

실행 결과가 여기에 표시됩니다.

초기화

코드 실행

제출 후 채점하기