문제 설명

땅따먹기 게임을 하려고 합니다. 땅따먹기 게임의 땅(land)은 총 N행 4열로 이루어져 있고, 모든 칸에는 점수가 쓰여 있습니다. 1행부터 땅을 밟으며 한 행씩 내려올 때, 각 행의 4칸 중 한 칸만 밟으면서 내려와야 합니다. **단, 땅따먹기 게임에는 한 행씩 내려올 때, 같은 열을 연속해서 밟을 수 없는 특수 규칙이 있습니다.**

예를 들면,

| 1 | 2 | 3 | 5 |

| 5 | 6 | 7 | 8 |

| 4 | 3 | 2 | 1 |

로 땅이 주어졌다면, 1행에서 네번째 칸 (5)를 밟았으면, 2행의 네번째 칸 (8)은 밟을 수 없습니다.

마지막 행까지 모두 내려왔을 때, 얻을 수 있는 점수의 최대값을 return하는 solution 함수를 완성해 주세요. 위 예의 경우, 1행의 네번째 칸 (5), 2행의 세번째 칸 (7), 3행의 첫번째 칸 (4) 땅을 밟아 16점이 최고점이 되므로 16을 return 하면 됩니다.

제한사항

* 행의 개수 N : 100,000 이하의 자연수
* 열의 개수는 4개이고, 땅(land)은 2차원 배열로 주어집니다.
* 점수 : 100 이하의 자연수

입출력 예

| **land** | **answer** |
| --- | --- |
| [[1,2,3,5],[5,6,7,8],[4,3,2,1]] | 16 |

입출력 예 설명

입출력 예 #1  
문제의 예시와 같습니다.

int hopscotch(vector<vector<int>> land)

{

int tmp = 0;

int maxSum = 0;

for (int i = 1; i < land.size(); i++)

{

for (int j = 0; j < 4; j++)

{

for (int k = 0; k < 4; k++)

{

if (j != k)

tmp = max(tmp, land[i][j] + land[i - 1][k]);

}

land[i][j] = tmp;

tmp = 0;

}

}

for (int i = 0; i < 4; i++)

{

maxSum = max(maxSum, land.back()[i]);

}

return maxSum;

}

int max(int a, int b)

{

return (a > b) ? a : b;

}

/\* 과정 설명

0행은 건너 뛰고 1행부터 시작

| 1 | 2 | 3 | 5 |

| 5 | 6 | 7 | 8 |

| 4 | 3 | 2 | 1 |

5(1, 0)를 기준으로 잡으면 5의 주소는 (1, 0)

(1 - 1)(0 이 아닌 나머지)를 각각 더해서 가장 큰수를 구해서 자신에게 덮어씌운다 그렇게 다른 열도 반복하면

| 1 | 2 | 3 | 5 |

| 10| 11| 12| 11|

| 4 | 3 | 2 | 1 |

그리고 다음 행으로 넘어가서도 똑같이 반복한

| 1 | 2 | 3 | 5 |

| 10| 11| 12| 11|

| 16| 15| 13| 13|

그러면 가장 큰 값은 16이 나온다

\*/