```cpp
// solution.cpp : 이 파일에는 'main' 함수가 포함됩니다. 거기서 프로그램 실행이 ↩
   시작되고 종료됩니다.
//

#include <iostream>
#include <string>
#include <vector>

using namespace std;

int solution(vector<vector<int>> board, vector<int> moves) {

    if (board.size() < 5 || board.size() > 30)
        return 0;

    int answer = 0;


    // 행열 입력 받은 것 반대로 바꿔준다.
    vector <int>::iterator iter;
    vector<vector<int>>::reverse_iterator iter_reverse;
    vector<vector<int>> r_board;

    // board[0]이 행의 개수가 된다.
    // [3,5,1,3,1] -> 각 row[0] 값에 해당..
    for (unsigned i = 0; i < board[0].size(); i++)
    {
        vector<int> row;
        // board.size가 열의 값이 된다.
        for (int j = board.size() - 1; j >= 0; j--)
        {
            row.push_back(board[j][i]);
        }

        r_board.push_back(row);
    }


    //vector<int>::reverse_iterator iter2;
    vector<int> box;
    //vector<int>::iterator iter;
    for (iter = moves.begin(); iter != moves.end(); iter++)
    {
        int move = *iter;

        vector<int>* board_cols;
        board_cols = &r_board[move - 1];
        if (board_cols == nullptr)
            continue;

        if (board_cols->size() == 0)
            continue;
        int doll = board_cols->back();

        // 0일 때는 추가 안해야 함
        if (doll == 0)
```

```
56          {
57              while (board_cols->size() != 0)
58              {
59                  board_cols->pop_back();
60                  if (board_cols->size() == 0)
61                      break;
62
63                  doll = board_cols->back();
64                  if (doll != 0)
65                      break;
66              }
67          }
68
69          if (board_cols->size() == 0)
70              continue;
71
72          board_cols->pop_back();
73          box.push_back(doll);
74
75          if (box.size() >= 2)
76          {
77              int i = box.at(box.size() - 1);
78              int j = box.at(box.size() - 2);
79              if (i == j)
80              {
81                  box.pop_back();
82                  box.pop_back();
83                  answer++;
84                  answer++;
85              }
86
87          }
88
89      }
90
91      return answer;
92  }
93
94  int main()
95  {
96      vector <vector<int>> board;
97      vector <int> board2;
98      vector <int> board3;
99      vector <int> board4;
100     vector <int> board5;
101     vector <int> board6;
102     vector <int> board7;
103
104     vector <int> moves;
105
106     //1, 5, 3, 5, 1, 2, 1, 4
107     moves.push_back(2);
108     moves.push_back(2);
109     moves.push_back(2);
110     moves.push_back(2);
111     moves.push_back(2);
```

```
112     moves.push_back(2);
113     moves.push_back(2);
114     moves.push_back(2);
115
116     //  [[0, 0, 0, 0, 0], [0, 0, 1, 0, 3], [0, 2, 5, 0, 1], [4, 2, 4, 4, 2],
          [3, 5, 1, 3, 1]], [1, 5, 3, 5, 1, 2, 1, 4]
117
118     board2.push_back(0);
119     board2.push_back(0);
120     board2.push_back(0);
121     board2.push_back(0);
122     board2.push_back(0);
123     board.push_back(board2);
124
125     board3.push_back(0);
126     board3.push_back(1);
127     board3.push_back(1);
128     board3.push_back(0);
129     board3.push_back(3);
130     board3.push_back(3);
131     board.push_back(board3);
132
133     board4.push_back(0);
134     board4.push_back(2);
135     board4.push_back(5);
136     board4.push_back(0);
137     board4.push_back(1);
138     board.push_back(board4);
139
140     board5.push_back(4);
141     board5.push_back(2);
142     board5.push_back(4);
143     board5.push_back(4);
144     board5.push_back(2);
145     board.push_back(board5);
146
147     board6.push_back(3);
148     board6.push_back(5);
149     board6.push_back(1);
150     board6.push_back(3);
151     board6.push_back(1);
152     board.push_back(board6);
153
154     int a = solution(board, moves);
155     std::cout << a << endl;
156 }
157
158
```