Problem Description

Consider a 7x14 map, where each position on the map has a value of 0, 1, 2, or 3 indicating the moving direction for that position. The representation: 0 for up, 1 for right, 2 for down, and 3 for left..

For example:

1 2 3 0 1 2 3 1 2 3 0 1 2 3

0 1 2 3 3 2 1 1 2 3 0 1 2 3

1 1 2 2 3 3 0 1 2 3 0 1 2 3

2 2 3 3 0 0 1 1 2 3 0 1 2 3

1 1 2 2 3 3 0 1 2 3 0 1 2 3

2 2 3 3 0 0 1 1 2 3 0 1 2 3

2 2 3 3 0 0 1 1 2 3 0 1 2 3

Start from the upper-left corner, follow the moving direction denoted by the value at each position. Stop when you reach the boundary of the map or when you reach a position that has been visited. Print the total number of visited positions, which should be 12 for the above example.

Complete the program by replacing with correct code:

```
#include <stdio.h>

#define ROWS 7

#define COLS 14

int map[ROWS+2][COLS+2];

int visited[ROWS+2][COLS+2];

int main(void)

{

    int i, j, direction, num;

    int d_row[] = { ??? , ??? , ??? , ??? };

    int d_col[] = { ??? , ??? , ??? , ??? };



    for (i=0; i<ROWS+2; i++) visited[ ??? ][ ??? ] = 1;
```

```
    for (i=0; i<ROWS+2; i++) visited[ ??? ][ ??? ] = 1;
    for (j=0; j<COLS+2; j++) visited[ ??? ][ ??? ] = 1;
    for (j=0; j<COLS+2; j++) visited[ ??? ][ ??? ] = 1;
    for (i=1; i<=ROWS; i++) {
        for (j=1; j<=COLS; j++) {
            scanf("%d", &map[i][j]);
        }
    }
    i = ??? ;
    j = ??? ;
    num = 0;
    while ( ??? ) {
        visited[i][j] = ??? ;
        num++;
        direction = ??? ;
        i = ??? ;
        j = ??? ;
    }

    printf("%d\n", num);
    return 0;
}
```

Input

The 7x14 map.

Output

Print the total number of visited positions.

Newline character at the end.

Sample Input

```
1 1 1 1 1 1 1 1 1 1 1 1 2
0 1 2 2 3 2 1 1 2 3 0 1 2 2
0 1 2 2 3 3 0 1 2 3 0 1 2 2
0 2 3 0 0 0 1 1 2 3 0 1 2 2
0 1 2 2 3 3 0 1 2 3 0 1 2 2
0 2 3 0 0 0 1 1 2 3 0 1 2 2
0 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Sample Output

```
38
```