

Graph Theory - S001 (E017)

Solved Challenges 1/2

[Back To Challenges List](#)**Iterations Count All Zero****ID:11091 Solved By 690 Users**

The program must accept an integer matrix of size **R*C** and an integer **K** as the input. For each occurrence of **K** in the matrix, the program must replace **K** and all the adjacent non-zero cell values with zero which are to its **left, right, top** and **bottom**. The program must repeat the process until all the values become zero. The program must print how many times the process has to be performed to convert all the cell values to zero.

Boundary Condition(s): $2 \leq R, C \leq 50$ **Input Format:**

The first line contains **R** and **C** separated by a space.

The next **R** lines, each containing **C** integers separated by a space.

The (**R+2**)nd line contains **K**.

Output Format:

The first line contains an integer representing the number of times the above process has to be performed to convert all the cell values to zero.

Example Input/Output 1:

Input:

```
5 5
5 6 0 5 6
1 8 8 0 2
5 5 5 0 6
4 5 5 5 0
8 8 8 8 8
6
```

Output:

```
2
```

Explanation:

After performing the process for the first occurrence of **6**, the matrix becomes
0 0 0 5 6

0 0 0 2

0 0 0 6

0 0 0 0

0 0 0 0

After performing the process for the second occurrence of **6**, the matrix becomes

0 0 0 0

0 0 0 0

0 0 0 0

0 0 0 0

0 0 0 0

Now, all the cell values in the matrix become zero.

Hence the output is 2

Example Input/Output 2:

Input:

4 5

5 0 5 6

1 0 8 1 0

0 5 0 6

4 5 0 5 2

5

Output:

4

Max Execution Time Limit: 500 millisecs

Ambiance

C (gcc 8.x) ▼



Reset

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int R,C;
5  void traverse(int matrix[R][C], int row,int col)
6  {
7      if(row>=0 && row<R && col>=0 && col<C)
8      {
9          if(matrix[row][col]==0)
10         {
11             return;
12         }
13         matrix[row][col] = 0;
14         traverse(matrix,row,col+1);
15         traverse(matrix,row,col-1);
16         traverse(matrix,row+1,col);
17         traverse(matrix,row-1,col);
18     }
19 }
20 int main()
21 {
22     scanf("%d%d",&R,&C);
23     int matrix[R][C];
24
25     for(int row=0;row<R;row++)
26         for(int col=0;col<C;col++)
27             scanf("%d",&matrix[row][col]);
28     int k,count=0;
29     scanf("%d",&k);
30     for(int row=0;row<R;row++)
31     {
32         for(int col=0;col<C;col++)
33         {
34             if(matrix[row][col]==k)
35             {
36                 count++;
37                 traverse(matrix,row,col);
38             }
39         }
40     }
41
42
43     printf("%d",count);
44
45 }
```

**i SkillRack Message : Iterations Count All Zero**

```
#include<stdio.h>
#include<stdlib.h>
int R,C;
void traverse(matrix[][C], int row,int col)
{
if(row>=0 && row<R && col>=0 && col<C)
{
if(matrix[row][col]==0)
return;
matrix[row][col] = 0;
traverse(matrix,row,col+1);
traverse(matrix,row,col-1);
traverse(matrix,row+1,col);
traverse(matrix,row-1,col);
}
}
int main()
{
int R,C;
scanf("%d%d",&R,&C);
int matrix[R][C];

for(int row=0;row<R;row++)
for(int col=0;col<C;col++)
scanf("%d",&matrix[R][C]);
int k,sourceR,sourceC,count=0;
scanf("%d",&k);
for(int row=0;row<R;row++)
for(int col=0;col<C;col++)
{
if(matrix[row][col]==k)
{
count++;
traverse(matrix,sourceR,sourceC);
}
}

printf("%d",count);

}
```

Please wait while we run the program 

☐ Run with a custom test case (Input/Output)