

CS-302 Design & Analysis of Algorithms (Fall 2013)

Assignment 2

Deadline: 21st October '2013

Total Marks: 40

- Your program should read input from a file and create an output file with the output as shown in sample output.
- Submissions would be accepted on SLATE only. Make exactly one source code file for each question in the assignment and attach it without putting it into a zip file.
- Students are encouraged to discuss the concepts been taught during the class in a group. Mention the names of students you have collaborated with.
- Students can be asked for concepts randomly if any plagiarism is observed in assignment. Any plagiarism case can result in a straight away **ZERO** in assignment at least or an **F** in this course.
- Assignments one day late subtract 10%; two days late loses 30%. After 2 days the assignment will be considered a zero.

Depth-First-Search

Q.No.1 The undirected unweighted graph with one selected vertex is given. Find the number of vertices in the connected component where the selected vertex belong (including the selected one).

Specifications

Input

The first line contains two integers n and s ($1 \leq s \leq n \leq 100$), where n - the number of vertices of the graph, and s - chosen vertex. The following n lines contains n numbers - the adjacency matrix of the graph in the MDM figure "0" means no edges between vertices, and the number "1" - its availability. It is guaranteed that the main diagonal of the matrix are always zero.

Output

Print the desired number of vertices

Example input

```

5 1
0 1 1 0 0
1 0 1 0 0
1 1 0 0 0
0 0 0 0 1
0 0 0 1 0

```

Example output

```

3

```

Q.No.2 The connected undirected graph without loops and multiple edges is given. You are allowed to remove the edges from it. Obtain a tree from the graph.

Specifications**Input**

The first line contains the number of vertices **n** ($1 \leq n \leq 100$) and the number of edges **m** of the graph. The next **m** pairs of numbers define the edges. It is guaranteed that the graph is connected.

Output

Print **n - 1** pairs of numbers - the edges that will be included in a tree. The edges can be displayed in any order.

Example input

```

4 4
1 2
2 3
3 4
4 1

```

Example output

```

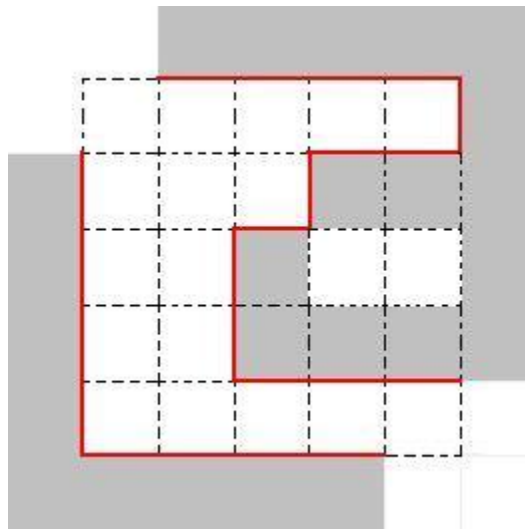
1 2
2 3
3 4

```

Breadth-First-Search

Painting maze

Q.No.3 Maze is a square, consisting of $N \times N$ segments. Each segment can be either empty or filled with stones. It is guaranteed that the upper-left and bottom right segments are empty. Surrounded by a maze of bottom, top, left and right walls, leave a space only the upper left and lower right corners. Director of the labyrinth has decided to paint the walls of the labyrinth, visible from inside.



View of the maze from above

Help him to calculate the amount of paint needed for this.

Specifications

Input

The first line contains the number N , then N lines of N symbols: the dot denotes an empty segment, the lattice - a segment of the wall.

$3 \leq N \leq 33$, the segment size of 3×3 , the height of the walls of 3 meters.

Output

Derive a single number - the area of the visible part of the interior walls of the labyrinth in square meters.

Example input

```

5
.....
...##
..#..
..###
.....

```

Example output

```

198

```

Dijkstra Algorithm

Q.No.4 The oriented weighted graph is given. Find the shortest path from the vertex **s** to the vertex **f**.

Specifications**Input**

The first line contains three numbers **n**, **s** and **f** ($1 \leq n \leq 100$, $1 \leq s, f \leq n$), where **n** is the number of vertices in a graph. Each of the next **n** lines contains **n** numbers - the adjacency matrix of the graph, where the number in the **i**-th line and **j**-th column corresponds to the edge from **i** to **j**: **-1** means the absence of the edge between the vertices, and any non-negative number - the presence of the edge of a given weight. The main diagonal of the matrix contains always zeroes.

Output

Print the required distance or **-1** if the path between the given vertices does not exist.

Example input

```

3 1 2
0 -1 2
3 0 -1
-1 4 0

```

Example output

```

6

```