You are given a building floor map like so:

WWWWWWWWWWWWW

W E W E W

W W W

W W

W W W

W E W E W

WWWWWWWWWWWWW

W - wall

E - employee

[SPACE] - empty space

You need to find the best empty space to put a kitchen in.

The kitchen needs to be located in the empty space for which the sum of distances to all employees is minimal.

The distance from an empty space to an employee is the shortest path from the employee to the empty space.

Employees can only walk in north, south, east or west directions (no diagonals).

Employees cannot walk through walls (of course).

Write a program that receives as input a file with the floor

plan and outputs the best kitchen location.

If the floor plan does not allow for a kitchen (for example if the floor

plan does not allow some employees to reach some spaces) then state that it is so.

Code explanation

Class board – a class for creating the board – opening the file and reading the board from the file. While loading creating an array of empty slots and an array of employees.

Class Point – represents a point on the board with x and y coordinates

Class Qnode – represtents the que

Including an initialization function, cell checking function and a function to print the board.

Space - class a class the represents a piece on the board, it’s coordinates on the board and its data (‘w’ ‘ ’ or ‘e’ and its neighbors coordinates.

To find the shortest distances between an employee and an empty space, I chose to implement the BFS algorithm.

First, for each empty node there is a need to calculate the distance from to every of its neighboring nodes and fill the neighbor’s array.

After that, using BFS algorithm to find the minimal distance between the employee to every empty point and return the point that is at the same distance for every employee. Every time there isn’t a path from an employee to a clear space, the space is fined by a hundred to ensure that space will no be chosen as the minimal path.