

Part 2: Hide-and-seek

Suppose that instead of a single agent as in Part 1, you have adopted k agents. The problem is that these agents do not like one another, which means that they have to be positioned such that no two agents can see one another. Write a program called `arrange_pichus.py` that takes the filename of a map in the same format as Part 1 as well as a single parameter specifying the number k of agents that you have. You can assume $k \geq 1$. Assume two agents can see each other if they are on either the same row, column, or diagonal of the map, and there are no walls between them. An agent can only be positioned on empty squares (marked with `.`). It's okay if agents see you, and you obscure the view between agents, as if you were a wall. Your program should output a new version of the map, but with the agents' locations marked with `p`. Note that exactly one `p` will already be fixed in the input map file. If there is no solution, your program should just display `False`. Here's an example on the same sample output on the same map as in Part 1:

```
[<>djcran@silo ~] python3 arrange_pichus.py map1.txt 5
```

```
....XXX
.XXXp..
.p..X..
.X.X...
.X.X.Xp
pX.p.X@
```