Programming Assignment: Pentominoes
Due: April 12, 2019

A pentomino consists of five 1 x 1 squares glued together. Any two squares glued together share an entire edge. Assuming that one can pick up a pentomino and turn it over, there are 12 distinct pentominoes. A pentomino consists of a number (usually all of them) of pentominoes and a shape that must be assembled with no overlap from the given pentominoes (aka pieces) using each piece exactly once.

Write a complete program that accepts (from stdin) a description of a polycube puzzle and finds all solutions and outputs them (to stdout). **All solutions printed must be inequivalent under reflection or rotation** (unless a rotation does not leave the prism invariant).

Each piece (polycube) is described by a string consisting of the letters {’N’, ’S’, ‘E’, ‘W’} meaning {North, South, East, West}, as well as small positive integers (one digit only). These indicate directions for assembling the cubes into the piece. For example, the string “SW1S1E’ consists of five squares: Put down one square, move 1 unit South, attach a second square, move 1 unit West and put down a third square, move back to the previous square, move 1 unit South, attach a square, go back to the previous square , and finally move 1 unit East and attach the fifth square.

THERE IS NO KNOWN EFFICIENT SOLUTION!

However, you can be (and are encouraged to be) efficiently inefficient! Use brute force to try all possibilities. Try to recognize as early as possible that a current partial solution has no way to be completed.

Input File Format:

N (# of pieces)
Followed by N strings, one on each line, where each string describes a piece as explained above
Followed by
H W (dimensions of the rectangle to be assembled from the pieces)

For example:

```
12
EEEE
EEES
EES1E
SSEE
WSEE
SEEN
SEES
SES1E
SW1S1E
SSW1E
ESEE
ESES

6 10
```

Output File Format:

P (# of solutions)
Followed by P strings, one per line, where each string describes one complete solution. Each string will contain H x W characters, and each character names one of the pieces. ‘A’ is the name of the first piece in the input description, ‘B’ the second, etc. Suppose character i (0-based) is ‘C’. Then, if i = W x h + w, we know that the square at location (h, w) is part of piece ‘C’.