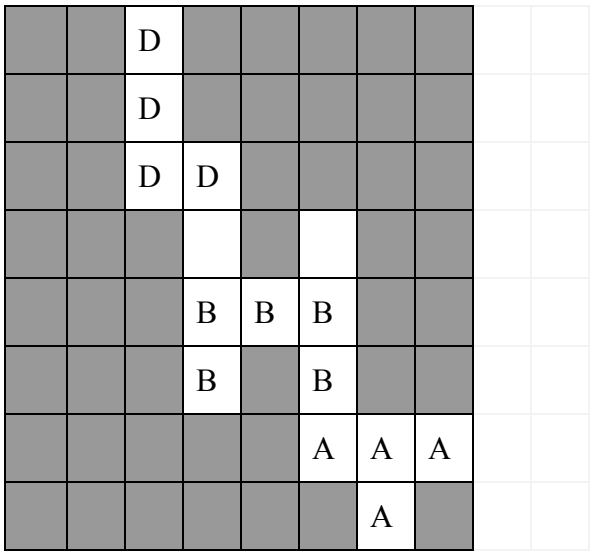
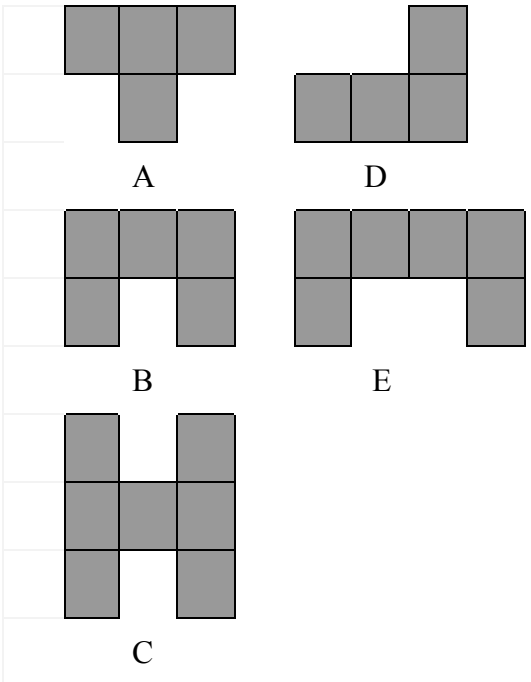
9. Puzzle

PROBLEM: Given an N x N grid that may have multiple cells designated as already filled with pieces of a puzzle, determine if the given pieces will fit in the puzzle using the following rules:

- Each input line will define a different puzzle and a different set of puzzle pieces.
- Each input line will define at most 5 puzzle pieces.
- Label the puzzle pieces in order A, B, C… E as needed.
- The pieces can be rotated 0, 90, 180 or 270 degrees in a clockwise direction. Use the smallest rotation necessary to make the piece fit.
- Puzzle pieces are placed in the puzzle in alphabetical order, from bottom to top and from left to right. That is one puzzle Piece A, then one puzzle Piece B...etc. Each new piece is added starting at the first bottom-most, left-most open grid location where it will fit. A piece is skipped only if it is impossible to be placed on the board. If a puzzle piece does not fit, skip it and try to place the next puzzle piece. After the last labeled puzzle piece is fit or skipped, start with puzzle Piece A again.





INPUT: There will be 10 sets of data. Each data set will consist of an integer *N* giving the size of the grid, an integer *P* giving the number of pieces that will be used, *P* strings defining the

pieces, and *N* strings giving the configuration of the board. The numbers and strings are separated by whitespace (spaces, tabs, newlines).

Each of the *P* pieces is a 3-character hex number. Convert each character to a binary number, and the 12 bits will fill a 3x4 grid with 1's and 0's from left to right, starting at the top row. The 1's represent locations that are filled to form contiguous puzzle pieces and the 0's represent locations that are not filled.

Each of the N strings (giving the configuration of the board) is a hex number that is N/4 characters long. Convert each to binary, and use them to fill the NxN grid from left to right, starting at the top row. The 1's represent locations that are filled and the 0's represent locations that are not filled.

The first set of Sample Input below defines the 5 puzzle pieces above and the 8x8 grid above.

OUTPUT: For each line of input, print the total number of degrees of rotation needed to fill the puzzle until no more pieces can be placed.

SAMPLE INPUT *(3 sets of data; your test data will have 10 sets of data)**:***
8 5 0E4 EA0 AEA 017 0F9 DF DF CF EB E3 EB F8 FD
8 5 720 EA0 575 170 0F9 FF FF F9 FD FD 01 A8 03
12 5 E40 075 AEA 170 F90 FFF FFF 1FF 8FF 07F 0CF A6F 1EF A0F 0FF ABF 03F

SAMPLE OUTPUT:

1. 90
2. 450
3. 1080

9. Puzzle

TEST DATA

TEST INPUT:

```
8 5 EEA 00F 088 88E CCC FF FF FF 83 87 00 00 C7
4 2 620 006 0 8 8 1
8 3 664 88F C44 FA FA 00 00 0D 40 00 80
8 4 332 88F 311 646 00 00 00 00 00 10 C0 E4
12 4 311 CE0 64C C60 888 888 888 FFF 888 888 888 FFF 888 888 888 FFF
12 5 311 C4C 266 C88 C60 088 088 088 088 FFF 088 088 088 FFF 088 088 088
4 3 064 8C0 003 0 0 0 0
8 5 064 8C0 003 666 C46 00 00 29 01 0C 40 09 41
4 3 F9F 88E 110  0 0 0 0
12 3 88F EAE 310 555 AAA 555 AAA 555 AAA 555 AAA 555 2AA 555 0AA
```

TEST OUTPUT:

1. 270
2. 270
3. 900
4. 990
5. 900
6. 720
7. 720
8. 810
9. 270
10. 0

9. Puzzle

	1	2	3	4	5	6	7	8	9	10	11	12
1	C	C	C	D	D	D	C	C				
2	C	C	C	D		D	B	C				
3	D	D	C	A	A		B	C				
4	D		C	A	A		B					
5	D	D	C	A	B	B	B					
6	C	C	C	X	B	B	B	B				
7	X	X	A	A	A			B				
8	X	X	X	A	A	X		B				
9												
10												
11												
12