B4M35KO, BE4M35KO
Practical Test – Knights Placement

1 Test Assignment

You task is to place as many Knights as possible on 8×8 chessboard such that no two Knights attack each other. Furthermore, there might be some Rooks already placed on some fixed positions on the chessboard – in that case, you have to place the Knights such that they are not attacked by the Rooks.

We remind you that Knight moves in an 'L-shape' as illustrated in Figure 1.

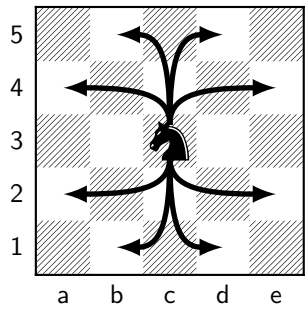

Figure 1: Knight moves illustration.

In case of the presence of a Rook on the chessboard, it threatens the whole row and column of the chessboard, as shown in Figure 2. We remind that you do not come up with the positions for the Rooks, these are given to you as an input. They just restrict some of the squares for your Knights.

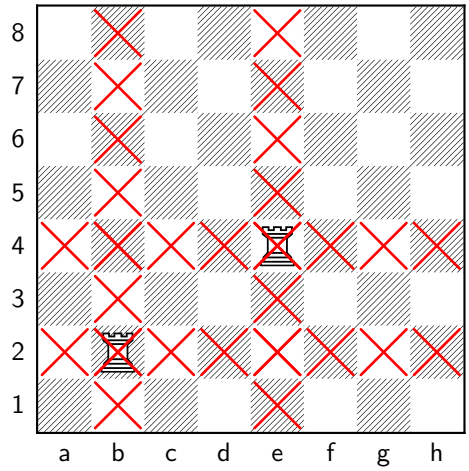

Figure 2: Red crosses represent the area threatened by the Rooks.

2 Assignment Evaluation (max. 10 points)

Upload your code to BRUTE. If your program will be able to pass all the tests you will obtain full 10 points. Supported programming languages are C++, Java and Python (see `https://cw.fel.cvut.cz/wiki/courses/ko/upload_system` for more details).

Your program will be called with two arguments: the first one is absolute path to input file and the second one is the absolute path to output file (the output file has to be created by your program).

2.1 Input File

The input file has the following form:

R
r_1
r_2
$\vdots$
r_R

Integer number R represents the number of Rooks present on the board (can be 0). Then R lines follow, containing strings r_1, r_2, ..., r_R representing the positions of the rooks. Each string r_i contains two characters (letter and number) representing the column and the row, respectively. E.g., if r_i = a3 then a Rook is placed on column 'a' and row '3' (or in chess terms, "a-file" and "the third rank").

2.2 Output File

The output file has the following form:

N
n_1
n_2
$\vdots$
n_N

The first line of the output file contains the value of the objective function N, i.e., the number of the Knights placed on the chessboard. The remaining N lines contain strings n_1, n_2, ..., n_N representing the positions of the Knights (again each n_i contains exactly two symbols - one character and one number representing the column and the row, respectively).

In the case when no Knight can be placed on the board, the output file should consist of value 0 only.

3 Examples

3.1 Example 1

Input:

```
0
```

This input represents the scenario without the Rooks.

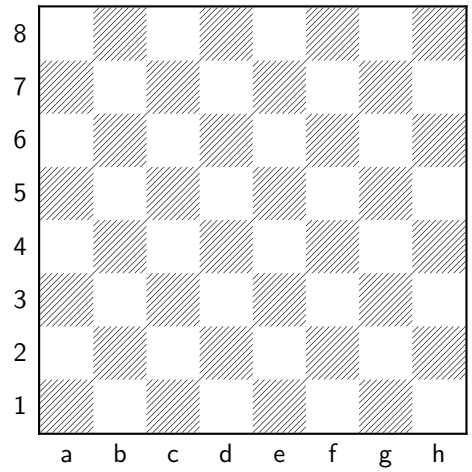

Figure 3: Input of Example 1.

Output:

```
32
b1
d1
f1
h1
a2
c2
e2
g2
b3
d3
f3
h3
a4
c4
e4
g4
b5
d5
f5
h5
a6
c6
```

```
e6
g6
b7
d7
f7
h7
a8
c8
e8
g8
```

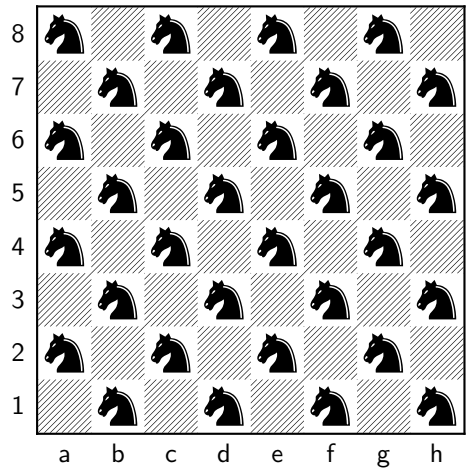

Figure 4: Solution of Example 1.

3.2 Example 2

The input represents the scenario with 2 Rooks shown in Figure 2.

Input:

```
2
b2
d5
```

Output:

```
20
a1
c1
e1
f1
g1
h1
a3
a4
e4
```

```
f4
g4
h4
a6
c6
h7
a8
c8
e8
g8
h8
```

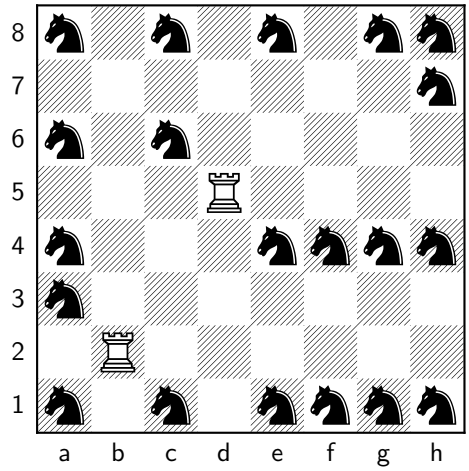

Figure 5: Solution of Example 2.

3.3 Example 3

The input represents a scenario where no Knight can be placed.

Input:

```
8
a1
b2
c3
d4
e5
f6
g7
h2
```

Output:

```
0
```

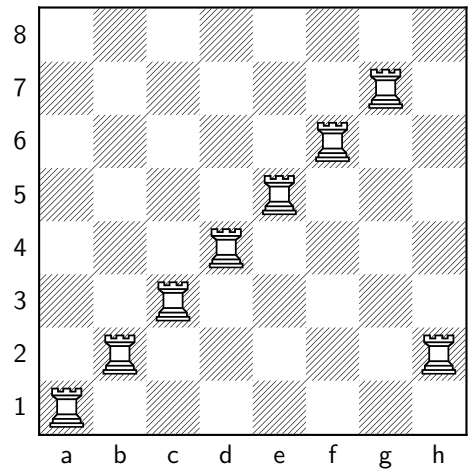

Figure 6: Solution of Example 3.