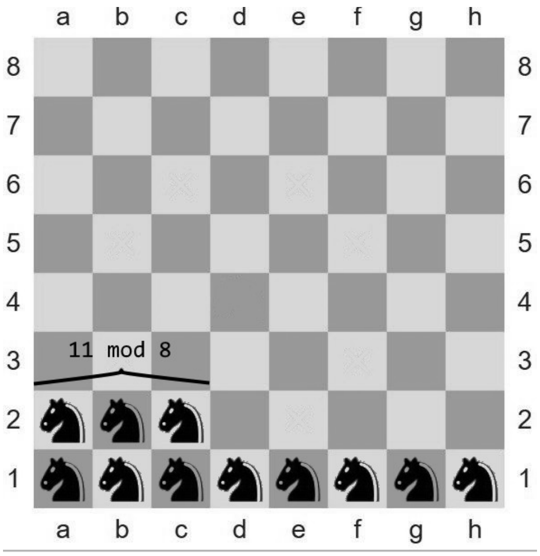
Problem A. Horseback Riding

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

It is well known that chess is a sport. But Dima, who is into horseback riding, considers it boring, and not a sport at all. His friend Sasha has decided to show him how complex and interesting chess can be.

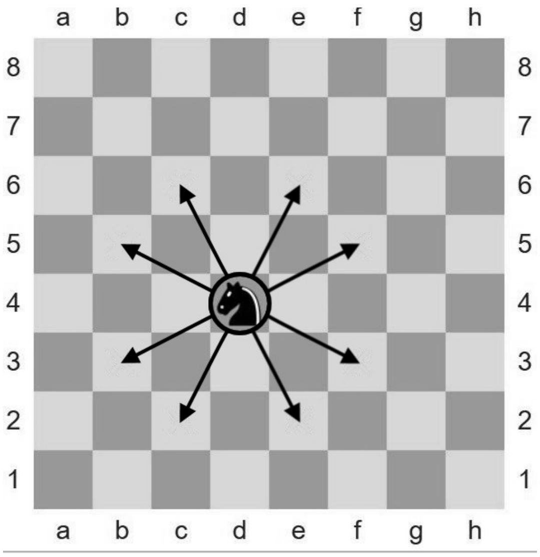
Sasha has placed k knights on a 8×8 chessboard. These knights are on a promenade and want to return to their stables. Unfortunately, they don't remember the way home.

Let us say that the knights are in their stables if the following conditions are satisfied: some (specifically, k div 8) lower rows are filled with knights, and the next row only contains knights in its leftmost cells (if $k \bmod 8 \neq 0$, then $k \bmod 8$ knights take leftmost positions in this row).



There are the desired cells for $k = 11$ knights, one per knight.

This is a chess problem, so the knights move as they do in chess: exactly two cells in one direction, and then exactly one cell in another direction.



Possible moves of a knight

Knight make moves one by one, in any order. In a single moment at most one knight can be located in any cell.

Dima spent two days solving this problem, and now asks you to find the sequence of knights moves, so that no two knight ever occupied the same cell, and all the knight ended up in the stables. You don't have to minimize the number of moves, but you are limited to no more than 1500 moves in total.

Input

The first line of input contains an integer k — the number of knights at the chessboard ($1 \le k \le 64$). The following k lines contains the positions of the knigths. Each position is formatted as `xy`, where `x` is the column letter, and `y` is the row digit.

Columns are numbered from «`a`» to «`h`» left-to-right, rows are numbered with digits from 1 to 8 bottom-to-top.

Output

The first line of output must contain the number of moves you need to get knights into the stables. Next, output these moves one per line, in the order they are made. Format your moves as `xy-zt`, whehe `x` and `z` are column letters, and `y` and `t` are row digits.

Remember that you don't have to minimize the number of moves, but you have to fit into the limit of 1500.

Examples

standard input	standard output
2 a5 f1	4 a5-b3 b3-a1 f1-d2 d2-b1
3 a1 b3 c2	5 b3-d2 d2-b1 c2-b4 b4-d3 d3-c1