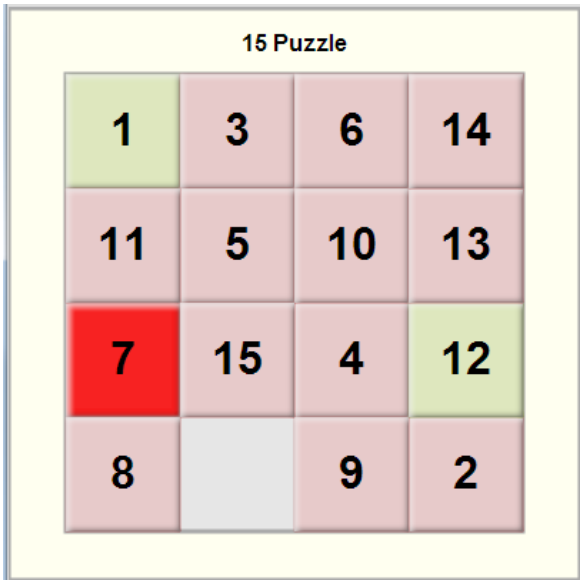
Task 4. 15-Puzzle Animation

Available marks: 23



The 15-Puzzle is a sliding-tile puzzle on a 4x4 grid with one hole, or missing tile. Each tile is labelled with a unique integer from 1 to 15. At each move one tile adjacent to the hole is moved into that position, effectively swapping the tile and the hole. The aim of the puzzle is to produce an arrangement where the numbers are in sequence in row order, with the hole at the bottom right.

Write a program that, given an initial layout of the puzzle and a sequence of moves, produces an SVG document that displays the moves in order, until either all moves have been processed, or an invalid move is encountered.

The first four input lines show the original layout, with spaces surrounding numbers (for visual alignment) and * representing the hole. Then follows the number of moves on a line by itself, and finally the list of moves on the last line. Each move is identified by the number on the tile. Moves are separated by a single space.

For example, the configuration at right could be the final state after interpreting this input file:

```
 1  3  6 14
11  5 10 13
 7 15 12  *
 8  9  4  2
6
12 4 9 7 15 5
```

SVG requirements

1. Each tile should be around 80 pixels square, have bevelled edges and its number should be centred using bold 32px text.
2. A tile can be in one of four states: *shuffled*, *home*, *selected* or *error*. *Shuffled* means out of its home position. Recommended fill colours are, respectively, #E7CACA, #DEE7BE, #FFFF66 and #F72424.
3. Each valid move is divided into two time periods:
    - The tile fill colour changes progressively from its current colour to the *selected* state colour during the first time period, then
    - the tile moves to the new position and changes colour progressively to the final state (*home* or *shuffled*, depending on the position) during a second time period.
4. Each time period is between 0.4s (for a quick animation) and 2s (for testing).
5. If the tile to be moved is not adjacent to the hole, it changes to the *error* state over one time period and the animation stops.

The SVG document generated by your program must be displayable in a suitable browser.

Assumptions

- The initial layout is valid, and contains the integers from 1 to 15 once each plus one asterisk.
- There may be any number of moves.
- Each move is a valid tile number.
- You may adapt the first part of the sample animation document provided to finalists to help prepare for this question.

Test files and assessment

Besides the sample file `task4sample.dat`, there are two test files used for assessment, `task4A.dat` and `task4B.dat`.

8 marks are allocated to the appearance of the tiles and smooth transitions, and 15 for completing the move sequences correctly.