

EECS2301E

Lab 8
Fall 2018

Lab Objectives

Given the 15 puzzle https://en.wikipedia.org/wiki/15_puzzle You should write a program to manipulate it (not to solve it just manipulate it).

Your program should be able to get commands from the user, the commands are to move the space up (u), down (d), left (l), or right (r). then your program should move the space and display the puzzle.

Part 1

In this part, just write the program to read and display the array

Reading the input

The input is on the form of a pair of integers (i,j) , then 15 digits (1..15).

the i,j pair indicates where is the space. Then the 15 numbers fill the array with the numbers 1 to 15.

For example if the input

2,2

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

the array is

$$\begin{bmatrix} 7 & 9 & 1 & 15 \\ 2 & 6 & 8 & 3 \\ 11 & 4 & & 10 \\ 5 & 14 & 13 & 12 \end{bmatrix}$$

Which is a 4x4 array with element (2,2) empty space.

Display the array

The array should be displayed as above (without the square brackets of course).

For ease of marking you should print the numbers in the matrix as

```
printf("%-4d",a[i][j]);
```

and the empty space as

`printf(" ");` where there are 4 spaces between the two quotation marks.

submit 2031 lab_8 a1.c

Part 2

In this part, you continue on last program (a1.c), after reading the matrix and displaying it, will read user commands to move the space. The commands are u,d,l,r as we explained before, and q to quit.

First, when you read a character (char) you have to deal with new lines (pressing Enter on the keyboard), tabs, and spaces. All these are considered characters in C. You have to be sure that you read a real character not *white spaces*.

When you read a valid command (u,d,l,r,q), then you execute it and display the new matrix (after executing the command).

First if the input is not one of these characters, you should display

No such command

Then, continue reading the next command.

If the command is one of the 5 chars (u,d,l,r,q) then you check if the move is valid or not. If the move is valid, then you execute it and display the new matrix.

If the move is not valid, then you display

Not a valid move

And continue reading the next command until you get 'q'.

The command is invalid if the move results in moving the space out of the matrix.

For example is the space is in the first row (row 0) and the move is u, you can not move the command up if it is in the top row (similarly, you can not move the space down if it is in the bottom row, or move it left if it is in the first column, or move it right if it is in the last column).

If both the command and move are valid, the move is performed and the new game/matrix is displayed

For example consider the given matrix

$$\begin{bmatrix} 7 & 9 & 1 & 15 \\ 2 & 6 & 8 & 3 \\ 11 & 4 & & 10 \\ 5 & 14 & 13 & 12 \end{bmatrix}$$

if the command is 'u' which means move the space up, the new matrix is

$$\begin{bmatrix} 7 & 9 & 1 & 15 \\ 2 & 6 & & 3 \\ 11 & 4 & 8 & 10 \\ 5 & 14 & 13 & 12 \end{bmatrix}$$

submit 2031 lab_8 a2.c