Object Oriented Programming I
Assignment 1, Instructions

Implementing the 15 Puzzle in C++

1 Game Description

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

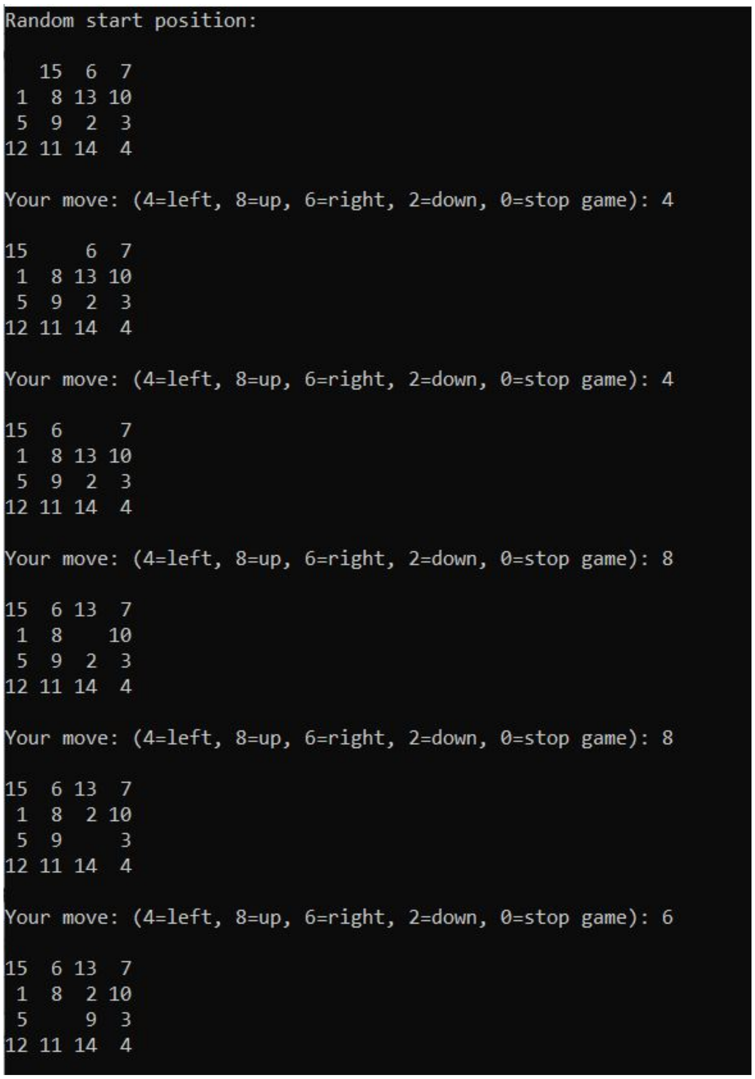
The numbers $1, 2, \ldots, 15$ are given on a 4×4 grid in a random order. One spot of the grid remains empty. The goal of the game is to bring the numbers into increasing orders by consecutively executing “moves” where a move means sliding a piece into the empty slot. Note that, at each stage, there are at most four possible moves, namely, “left”, “up”, “right”, and “down”, where, for instance “right” means that the piece on the left of the empty spot is slid right so that ends up in the empty spot. In the picture above, “right” slides 13 into the empty spot. For more details on the game, please see the Wikipedia article on it.

Fantastic video on the 15 puzzle: **https://www.youtube.com/watch?v=QxvnEwvgfeI**

2 C++ Program

Write a C++ program that allows the user to play a 15 puzzle game starting at a random position.

- The program should use pseudorandom numbers produced by **rand()** to set up a random position and print it on the screen.
- At each step, it should be checked if the user input is valid. Otherwise, the user should be asked to correct the input.
- If the user enters **0**, the program should be stopped.
- When the numbers are in increasing order (no matter where the empty spot is), “success” should be printed to the screen and the program should be stopped.
- The output of the program should look similar to the following (this game has not been completed).



```
Random start position:

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

Your move: (4=left, 8=up, 6=right, 2=down, 0=stop game): 4

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

Your move: (4=left, 8=up, 6=right, 2=down, 0=stop game): 4

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

Your move: (4=left, 8=up, 6=right, 2=down, 0=stop game): 8

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

Your move: (4=left, 8=up, 6=right, 2=down, 0=stop game): 8

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

Your move: (4=left, 8=up, 6=right, 2=down, 0=stop game): 6

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

3 Submission

- Put your complete solution into **one single** file `Assignment1.cpp`.
- Submit the file `Assignment1.cpp` in NTULearn.
- **Do not** submit any other file: Do not submit any Dev-C++ or XCode project files, do not submit any data files.
- **Important:** You must make sure that your program compiles and runs correctly under Dev-C++ or XCode.
- **Write your own program.** Do not copy from any sources. Submission will be checked with anti-plagiarism software.