CSE102 – Computer Programming

Homework #8

Functions and Selection

Due Date: 24/05/2023

Hand in: A student with number 20180000001 should hand in a zip file named 20180000001.zip for this homework via Microsoft Teams.

. Any deviation from the shared format may be penalized regardless of the correct execution.

8 Puzzle Simulation

For this C homework, you are asked to creating an 8-puzzle game. You need to use a struct data structure to represent the puzzle board. When your program starts, it should present the user with a menu consisting of three options: (1) Play game as a user, (2) Finish the game with PC, (3) Show the best score and (4) Exit. In order to provide a detailed and engaging user experience, your program should meet the following requirements for each menu option:

If the user selects "Play game as a user", your program should generate a random 8-puzzle board and save the initial state of the board to a txt file. During gameplay, the user should input a number and direction (e.g., "2-R" to move the number 2 to the right) to control the puzzle board. After each move, the program should save the updated state of the board to the same txt file. Ensure that your program checks for illegal moves and provides appropriate warning messages to the user. Once the user successfully completes the puzzle, your program should display a congratulatory message, print the total number of moves taken and score (Formula of the score= 1000 – 10 * number of moves). If user has the best score, program save that score to a txt file. After this, the program should return to the main menu.

If the user selects "Finish the game with PC", your program should solve the puzzle automatically. To do so, you should write a recursive function named **auto_finish**. This function should make random legal moves. Similar to the first option, the program should save the state of the puzzle board to a txt file after each move. When the computer completes the puzzle, the program should print the total number of moves taken. After this, the program should return to the main menu.

If the user selects " Show the best score", your program should display the best score by reading **best_score.txt** file.

If the user selects "Exit", your program should terminate gracefully.

Expected Outputs:

```
Welcome to the 8-Puzzle Game!
Please select an option:
1. Play game as a user
2. Finish the game with PC
3. Show the best score
4. Exit
> 1


1 3 8
2 5 6
7 _ 4

Enter your move (number-direction, e.g., 2-R): 5-D

1 3 8
2 _ 6
7 5 4

Enter your move (number-direction, e.g., 2-R): 2-R

1 3 8
_ 2 6
7 5 4

Enter your move (number-direction, e.g., 2-R): 1-L

1-L is an illegal move!!! Make a legal move!!!


1 3 8
_ 2 6
7 5 4

Enter your move (number-direction, e.g., 2-R):
```

```
...

1 2 3
4 5 6
7 8 _


Congratulations! You finished the game.

Total number of moves: 28
Your Score: 720
```

```
Welcome to the 8-Puzzle Game!
Please select an option:
1. Play game as a user
2. Finish the game with PC
3. Show the best score
4. Exit
> 2


1 4 7
5 2 6
8 _ 3

Computer Move: 8-R

1 4 7
5 2 6
_ 8 3

Computer Move: 5-D


1 4 7
_ 2 6
5 8 3

Computer Move: 2-L

1 4 7
2 _ 6
5 8 3

...
```

```
...

1 2 3
4 5 6
7 8 _


Computer finished the game.

Total number of computer moves: 36
```

```
 Welcome to the 8-Puzzle Game!
 Please select an option:
 1. Play game as a user
 2. Finish the game with PC
 3. Show the best score
 4. Exit
 > 3

 The best score is 840...
```

```
Welcome to the 8-Puzzle Game!
Please select an option:
1. Play game as a user
2. Finish the game with PC
3. Show the best score
4. Exit
> 4

Program terminated...
```

Note: Make sure to include appropriate comments and variable names in your code to make it easy to understand.

****Attach the photos of the outputs of the code to your file.**

****Do not forget to prepare a makefile (-50 points)**

General Rules:

1. We do not give you any function prototypes. We expect that you are experienced enough to understand when to use methods and name them. These will also be graded.
2. The program must be developed on given version of OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.
3. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it is working partially.

4. Zip your homework files before uploading them to MS Teams. The zip file must contain the C file with your solution and screenshots of the valid outputs of the program.
5. You can ask any question about the homework by sending an email to barisozcan@gtu.edu.tr or by using the homework channel on MS Teams page of the course.