CS 461 – ARTIFICIAL INTELLIGENCE

HOMEWORK # 2

Group Name : Puzzle_Busters

Group Members :

Muhammad Ramish Saeed

Mohamed Aymen Abidi

Kamel Ben Kmala

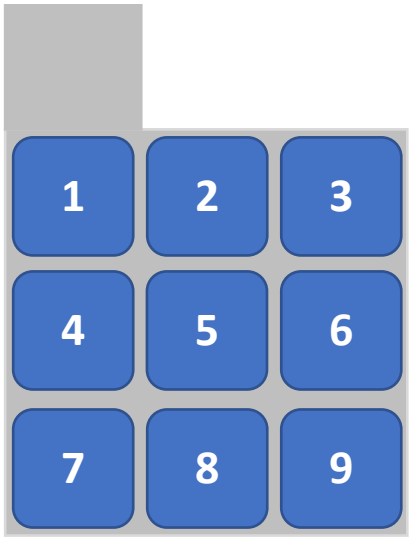
Yassine Gazzah

Dorra El Mekki

a. Write a **'puzzle generator'** first. Starting from the **goal state of 9-puzzle**:

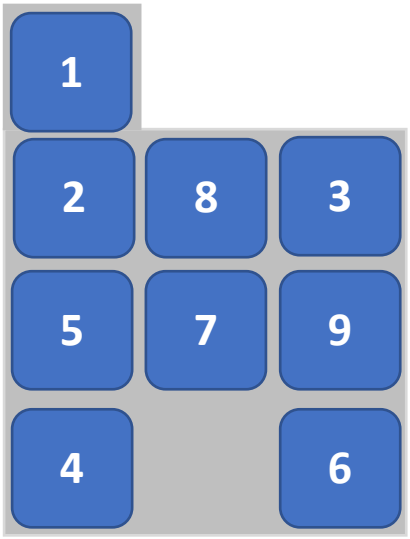
Very important :

The 9-puzzle is a game in which there are nine 1x1 tiles arranged on a form composed of a 1x1 square and a 3x3 square (cf. image below) so that there is one 1x1 uncovered (empty) area on the form. The tiles are labeled 1 through 9, and initially they are shuffled. The idea is to reach the goal state (cf. image below) from a given initial state by moving tiles one at a time.



In order to do that, the tile which have "1" as a value must be the one placed on the 1x1 square because if another tile was placed there the puzzle will be unsolvable. In fact, the operation used to reach the goal state from the initial state (shuffled tiles) must be a bijective operation. In simple terms, the inverse-operation must lead to the initial state when starting from the goal state.

The generator returns a reasonably **garbled initial state** (S) by randomly shuffling the puzzle.



(!) Notice that the generator thus guarantees that this initial state S will be solvable.

- How to be sure that the generated puzzle is solvable ?

We find two different manners to generate a solvable puzzle with shuffled tiles.

First : The first in based on the following steps :

1. start out with the "correct" 9-puzzle.
2. Generate a random number of "miss-placings" to be applied to the board.
3. Generate N list of valid miss-placings based off the random generated number.

Since, each time, we start with a valid board, then applying a set of forced mis-directions, we will always end up a completely legitimate start game (solvable puzzle).

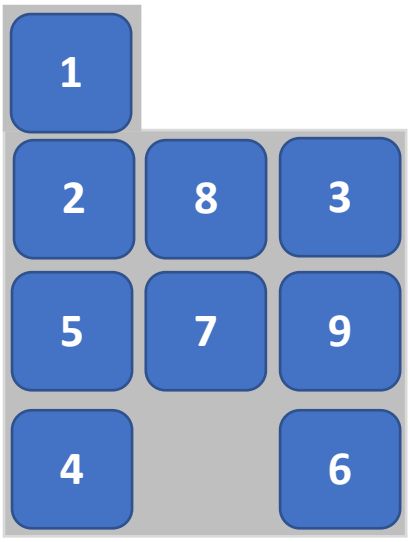
Second :

The second manner to generate a solvable puzzle with shuffled tiles is by checking the number of inversions after the generating a random puzzle (but the "1" value tile must be placed on the 1x1 square). In fact,the same as for the 8-puzzle, there are many unsolvable configurations of a 9-puzzle. The basic trick is to check **the number of possible inversions**. If you have an **even number** of inversions, then the puzzle is **solvable**. If you have an **odd number** of inversions, then you have an **unsolvable** puzzle. So we are going to work with an 8-puzzle which have tiles labeled 2 through 9 instead of 1 through 8.

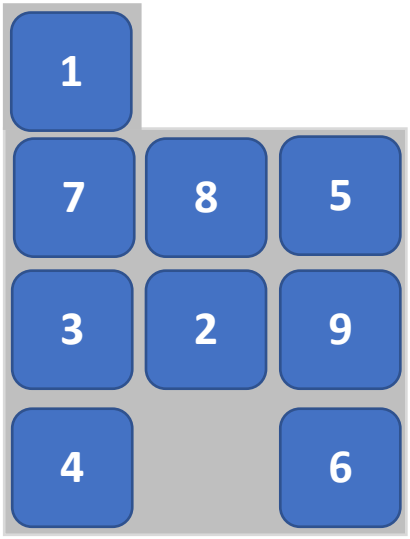
To find the total number of inversions, count the number of smaller numbers that appear after every number in the array:

```
Count_invers = 0
For square in [0..8]
    For other_square in [square .. 9]
        If (square.value > 0 and other_square.value > 0 and square.value > other_square.value )
            Count_invers ++
```

Example :



0+5+0+1+2+2+0= 10



5+5+4+1+0+2+0 = 17

EVEN	**ODD**
Solvable	**Unsolvable**

⇨ We choose to work with the second manner.

Branch and Bound

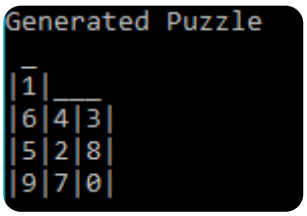
The search for an answer node can often be speeded by using an "intelligent" ranking function, also called an approximate cost function to avoid searching in sub-trees that do not contain an answer node. It is similar to backtracking technique but uses BFS-like search.

Cost function

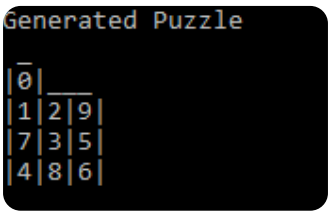
Each node X in the search tree is associated with a cost. The cost function is useful for determining the next E-node. The next E-node is the one with least cost. The cost function is defined as :

```
C(X) = g(X) + h(X) where

g(X) = cost of reaching the current node from the root

h(X) = cost of reaching an answer node from X.
```

b. The 10 distinct initial states of 9-puzzle obtained by the puzzle generator :



Puzzle - 1



puzzle - 2

****NOTE****: The rest of the 8 puzzle generators initial state will be given in the screenshot window i.e from puzzle 3 and onwards for ease.**

Also we acknowledge the font is quite small but if we had used a larger font then the number of screenshots would have been too large so just to avoid that we kept it small.

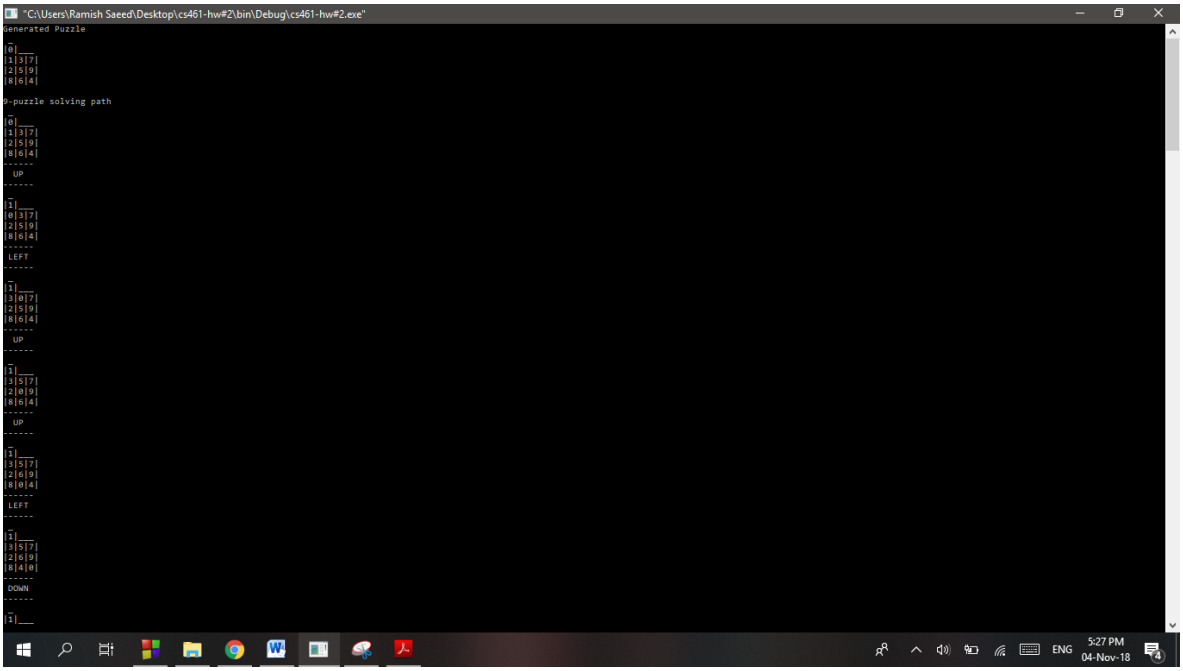
c. Solving each of these 10 puzzle instances :

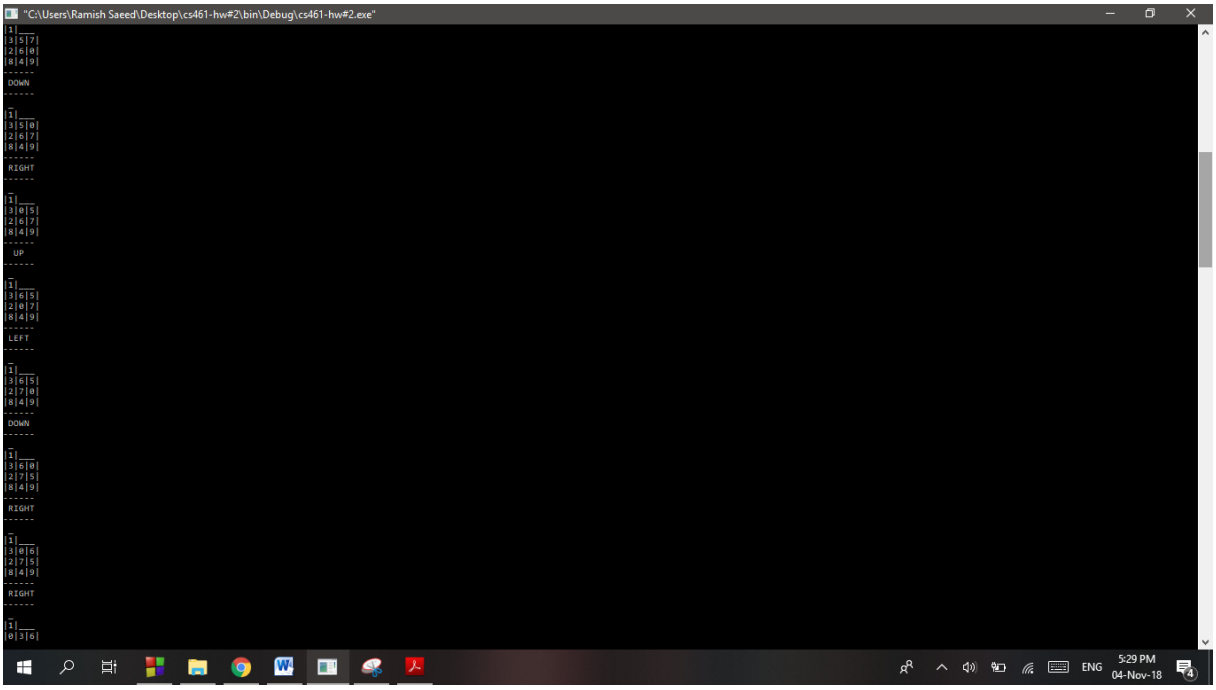
puzzle - 1

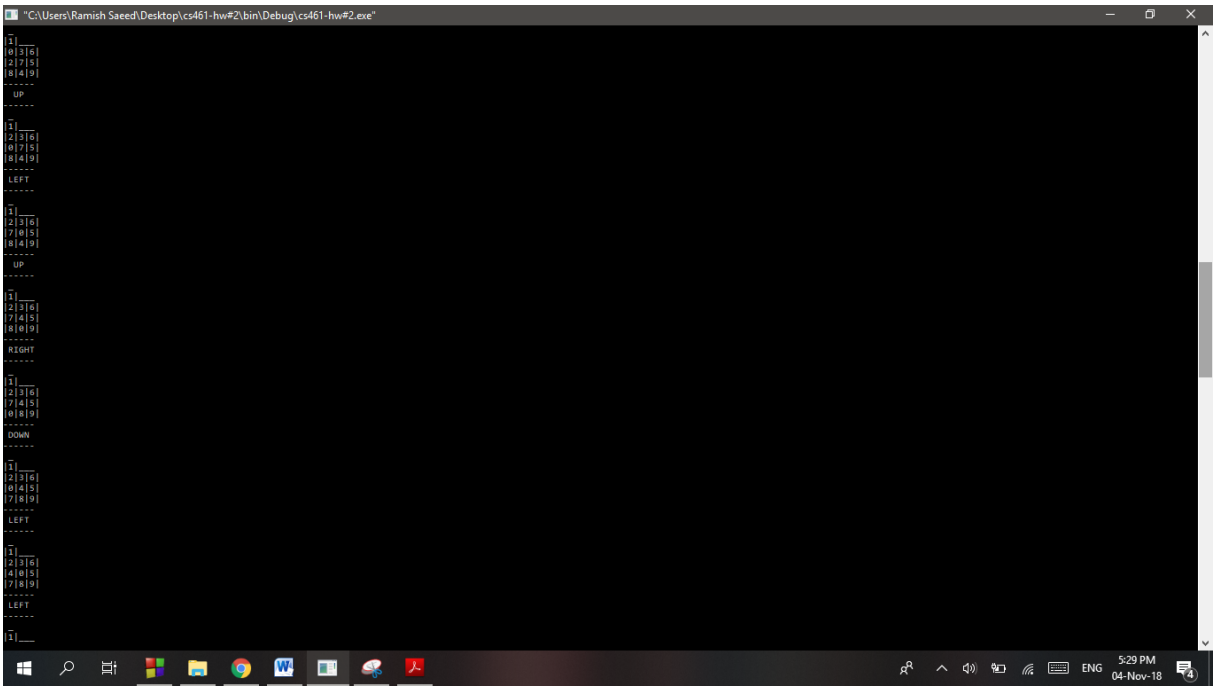
puzzle – 2

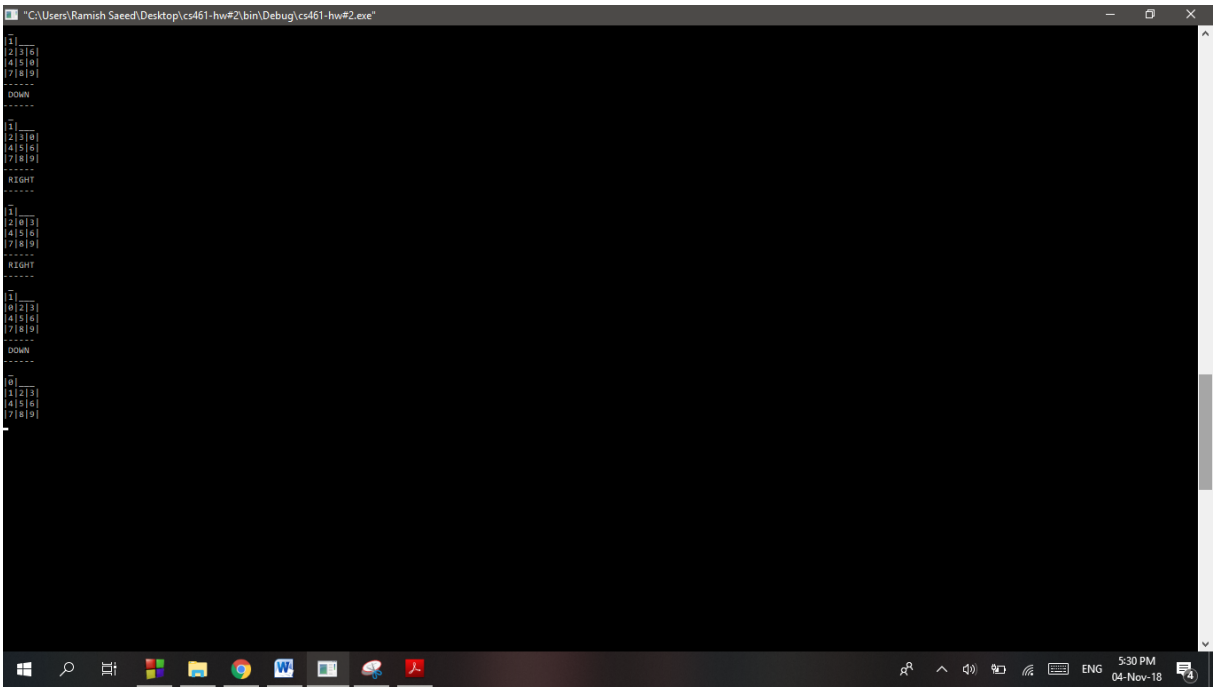
****NOTE**: Since it was taking too much time too cut and copy paste the puzzle from terminal, we are presenting the next 8 examples in the following format for our ease.**

puzzle – 3

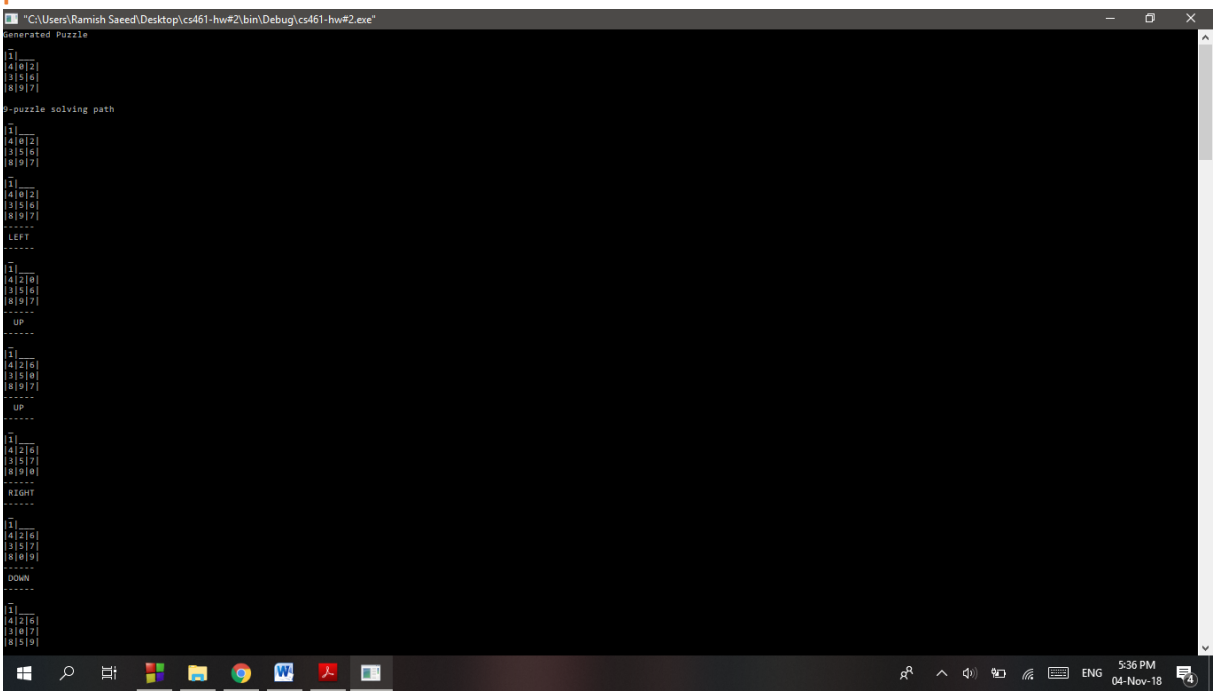

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
UP
LEFT
UP
LEFT
DOWN
ENG
5:27 PM
04-Nov-18

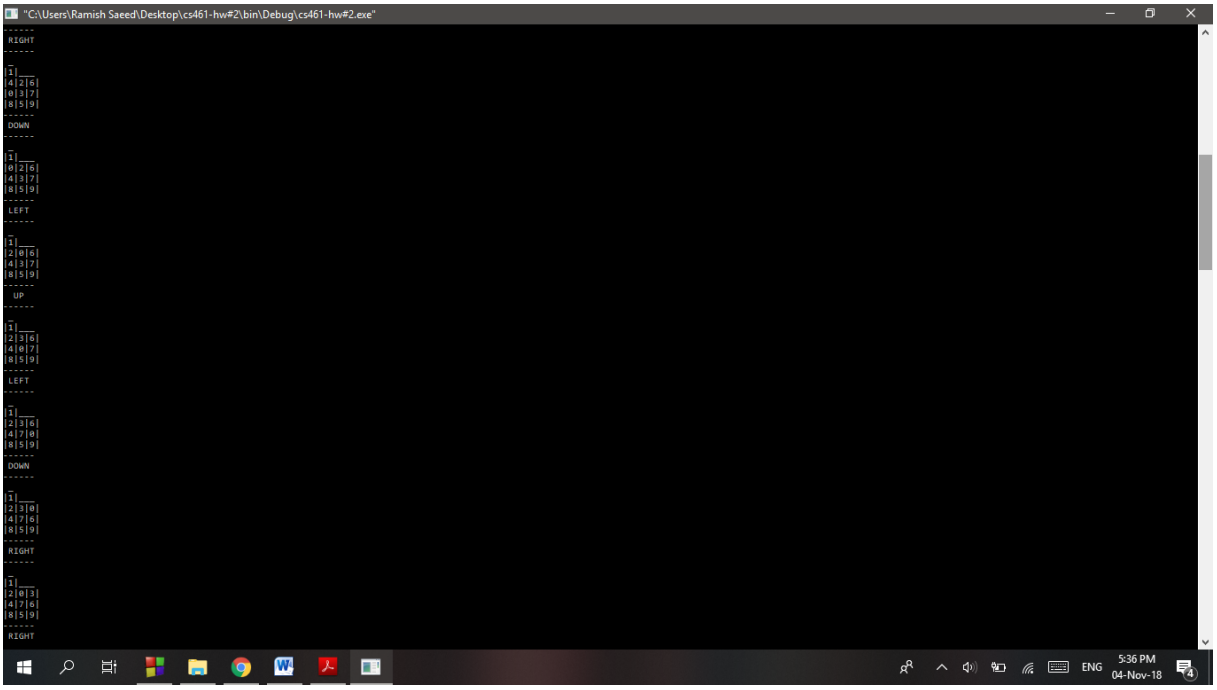

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
RIGHT
UP
LEFT
DOWN
RIGHT
RIGHT
ENG
5:29 PM
04-Nov-18

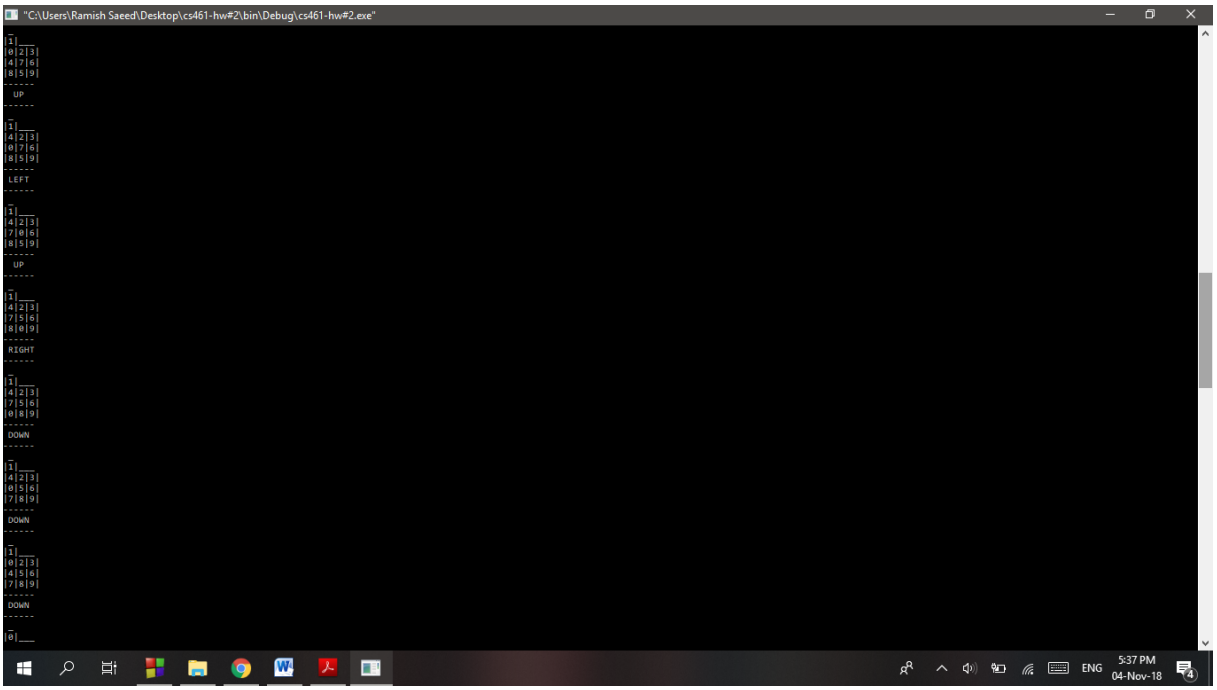

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
1|___
0|3|6|
2|7|5|
8|4|9|
UP
1|___
2|3|6|
0|7|5|
8|4|9|
LEFT
1|___
2|3|6|
7|0|5|
8|4|9|
UP
1|___
2|3|6|
7|4|5|
8|0|9|
RIGHT
1|___
2|3|6|
7|4|5|
0|8|9|
DOWN
1|___
2|3|6|
0|4|5|
7|8|9|
LEFT
1|___
2|3|6|
4|0|5|
7|8|9|
LEFT
1|___
ENG
5:29 PM
04-Nov-18

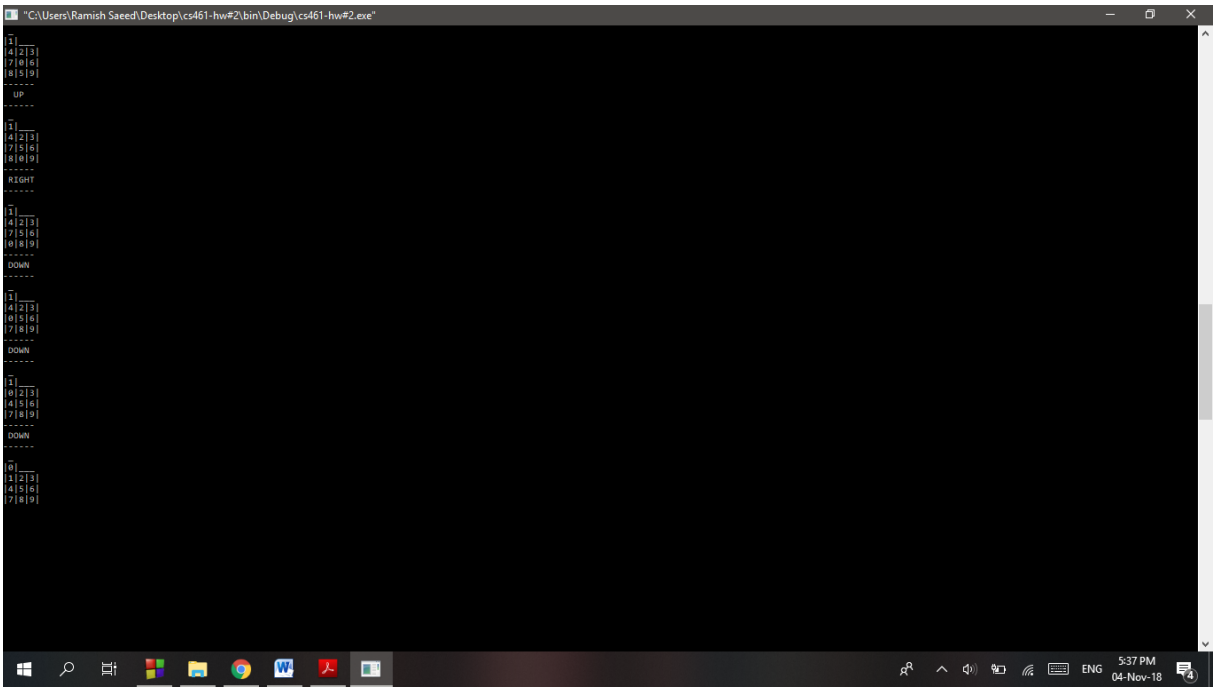

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
1|___
2|3|6|
4|5|0|
7|8|9|
DOWN
1|___
2|3|0|
4|5|6|
7|8|9|
RIGHT
1|___
2|0|3|
4|5|6|
7|8|9|
RIGHT
1|___
0|2|3|
4|5|6|
7|8|9|
DOWN
0|___
1|2|3|
4|5|6|
7|8|9|
ENG
5:30 PM
04-Nov-18

puzzle – 4

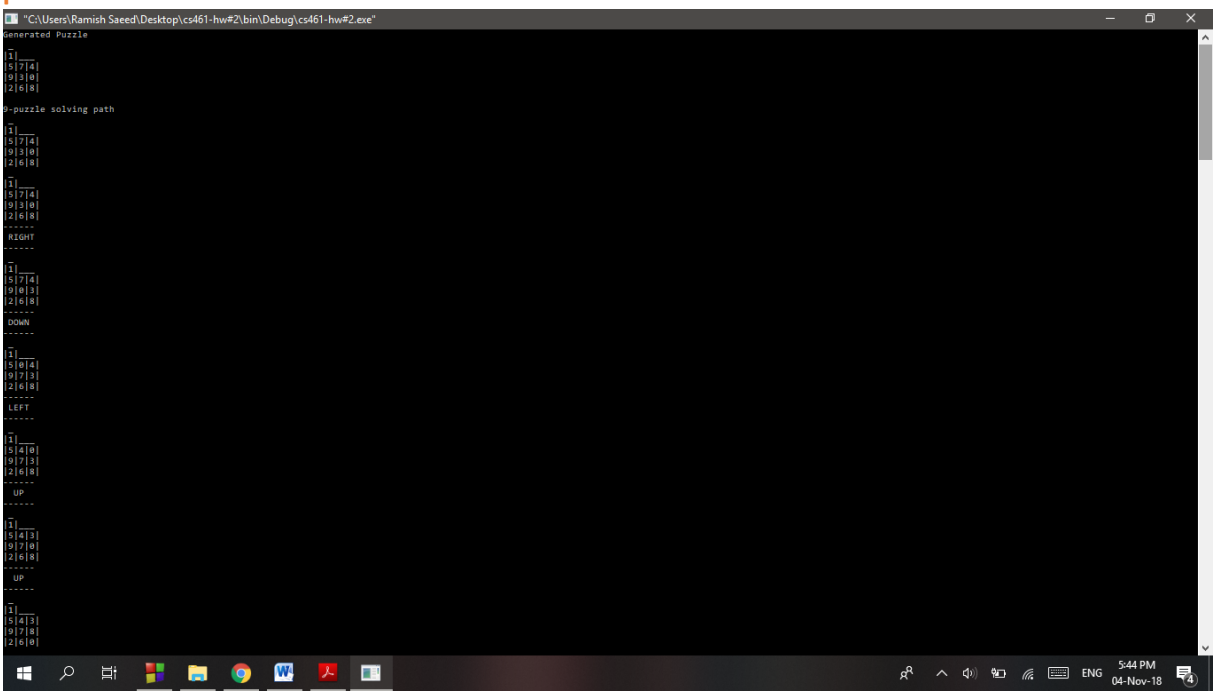

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
LEFT
UP
UP
RIGHT
DOWN
ENG
5:36 PM
04-Nov-18

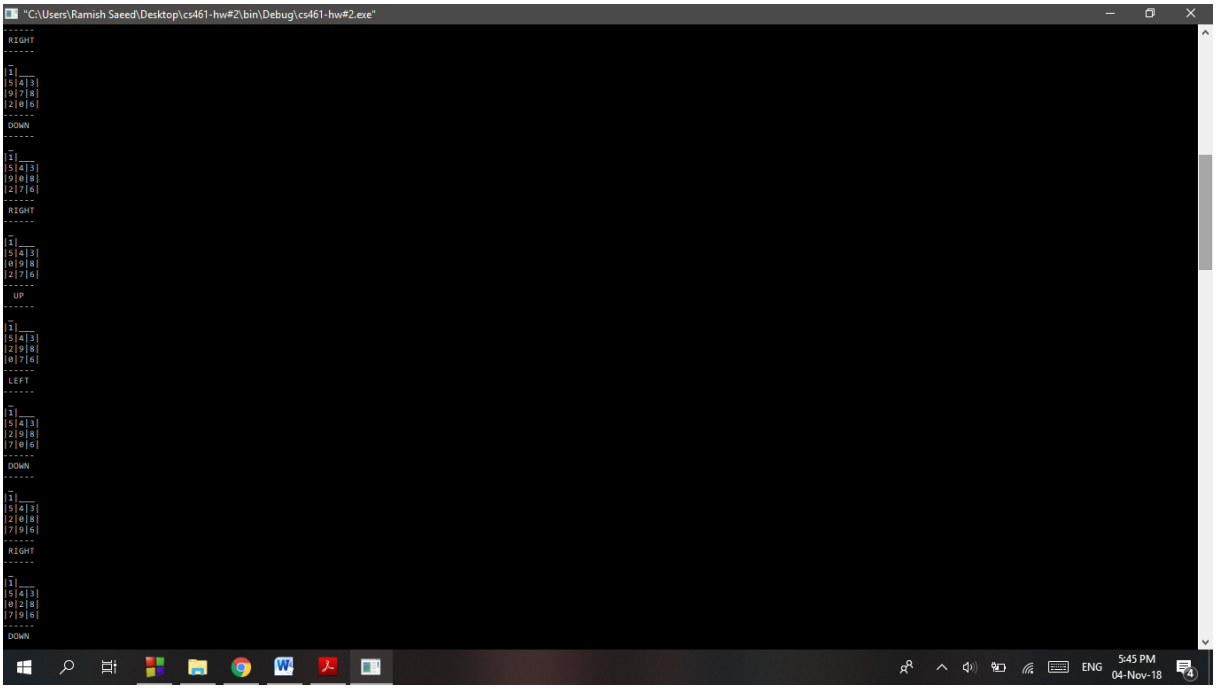

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
RIGHT
DOWN
LEFT
UP
LEFT
DOWN
RIGHT
RIGHT
ENG
5:36 PM
04-Nov-18

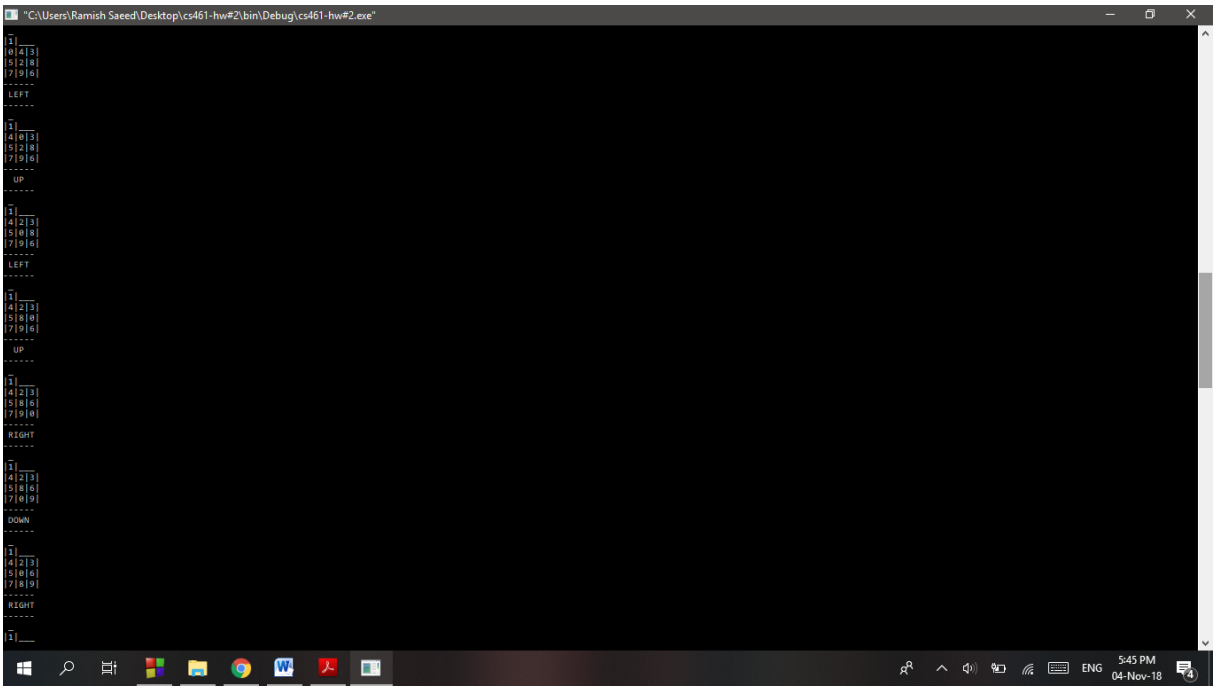

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
UP
LEFT
UP
RIGHT
DOWN
DOWN
DOWN
ENG
5:37 PM
04-Nov-18

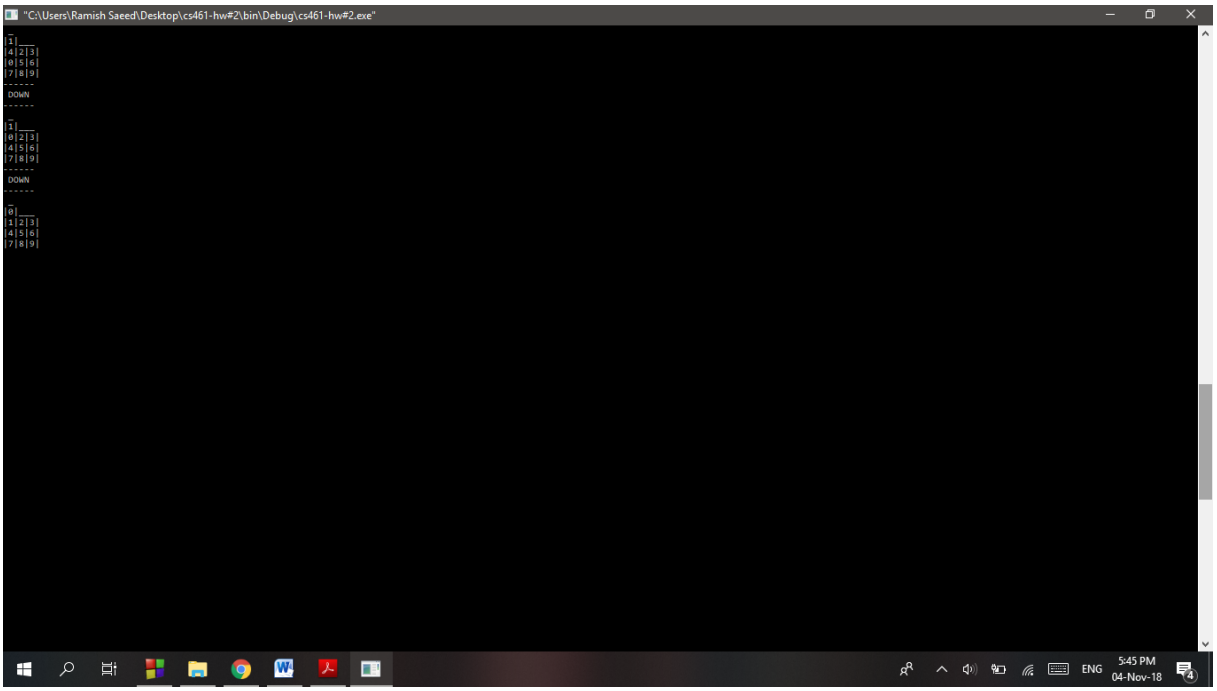

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
UP
RIGHT
DOWN
DOWN
DOWN
ENG
5:37 PM
04-Nov-18

puzzle – 5

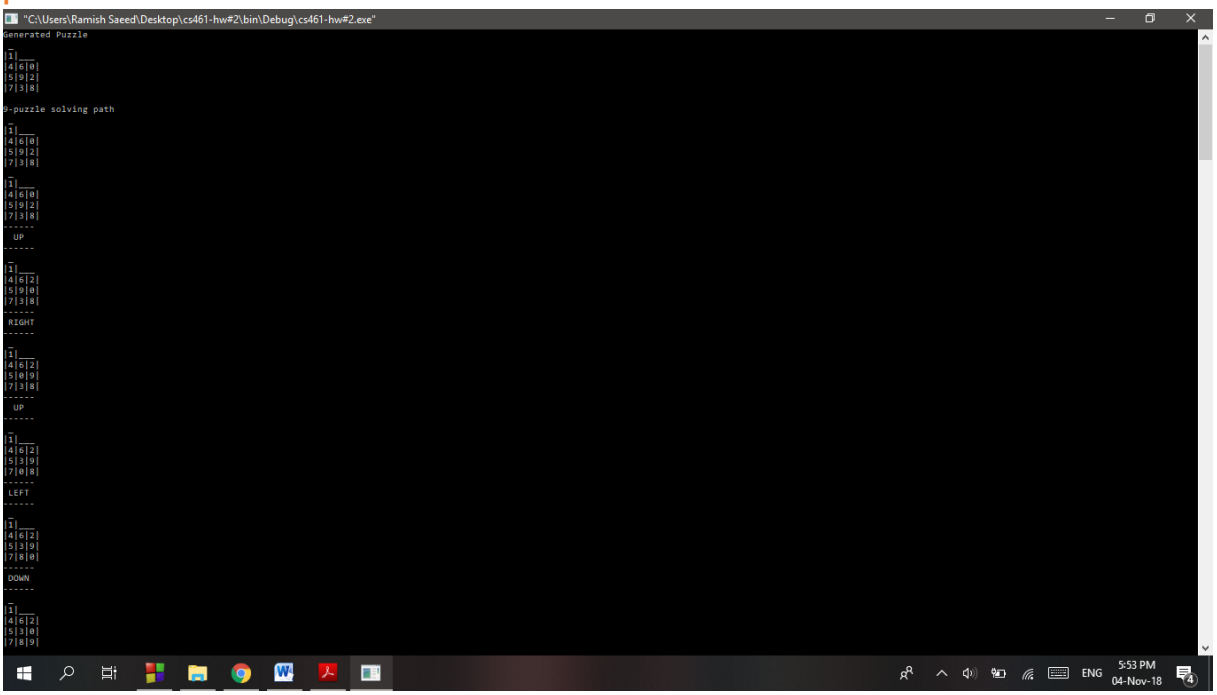

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
RIGHT
DOWN
LEFT
UP
UP
ENG
5:44 PM
04-Nov-18

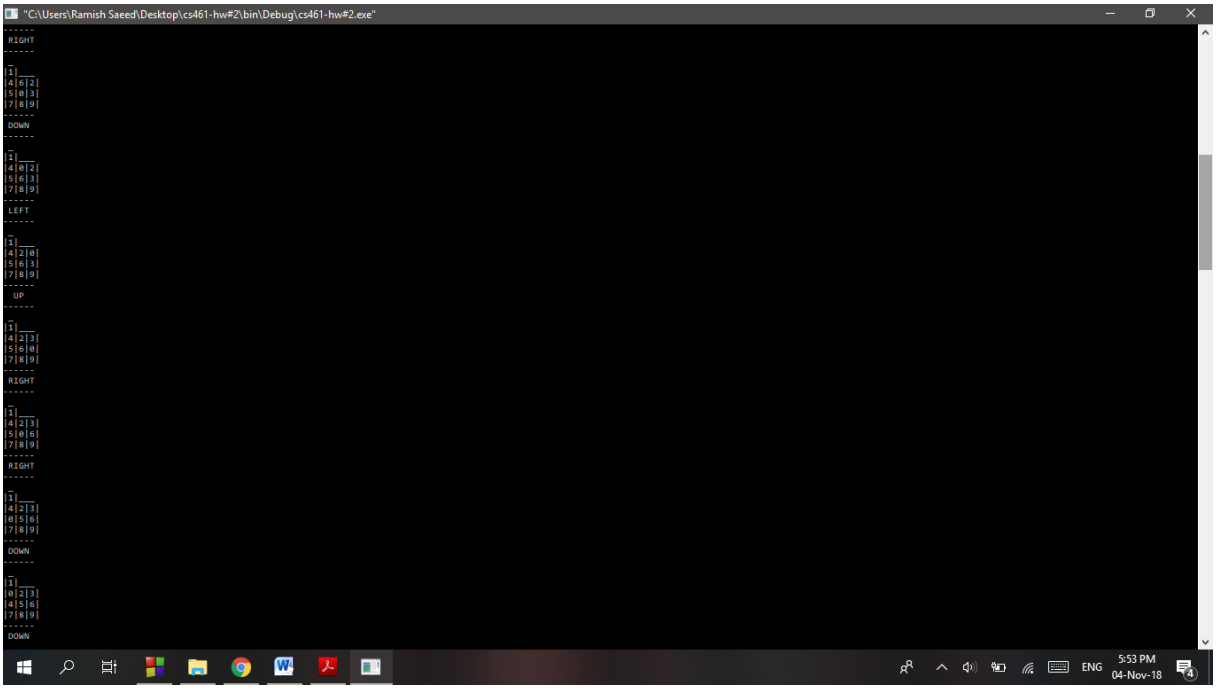

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
RIGHT
DOWN
RIGHT
UP
LEFT
DOWN
RIGHT
DOWN
ENG
5:45 PM
04-Nov-18

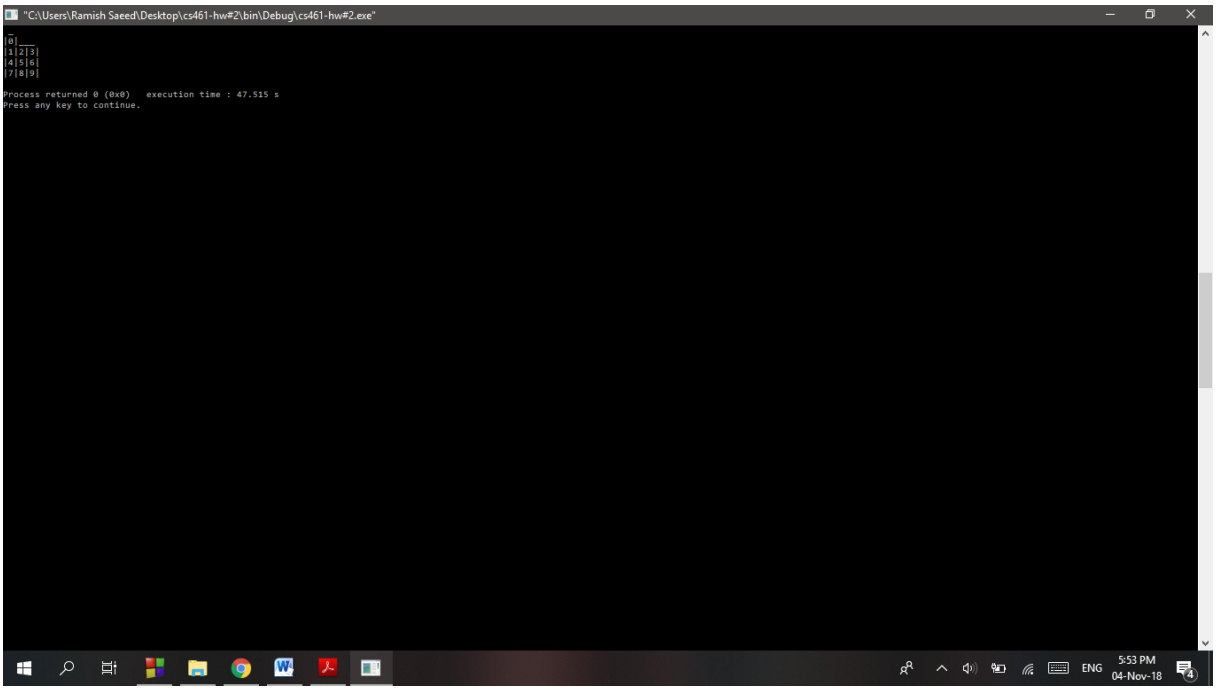

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
LEFT
UP
LEFT
UP
RIGHT
DOWN
RIGHT
ENG
5:45 PM
04-Nov-18


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
DOWN
ENG
5:45 PM
04-Nov-18

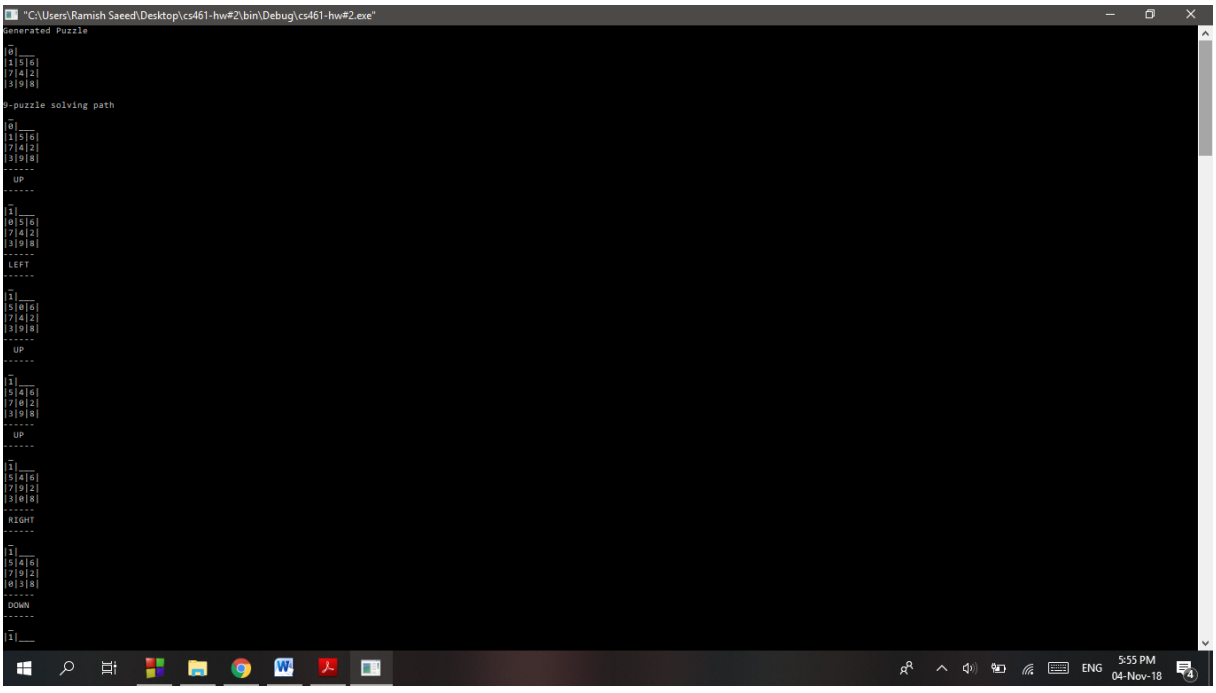
puzzle – 6


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
UP
RIGHT
UP
LEFT
DOWN
ENG
5:53 PM
04-Nov-18


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
RIGHT
DOWN
LEFT
UP
RIGHT
RIGHT
DOWN
DOWN
ENG
5:53 PM
04-Nov-18


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
_
|0|___
1	2	3
4	5	6
7	8	9
Process returned 0 (0x0) execution time : 47.515 s
Press any key to continue.
ENG
5:53 PM
04-Nov-18

puzzle – 7


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
_
|0|___
1	5	6
7	4	2
3	9	8
9-puzzle solving path
_
|0|___
1	5	6
7	4	2
3	9	8

UP

_
|1|___
0	5	6
7	4	2
3	9	8

LEFT

_
|1|___
5	0	6
7	4	2
3	9	8

UP

_
|1|___
5	4	6
7	0	2
3	9	8

UP

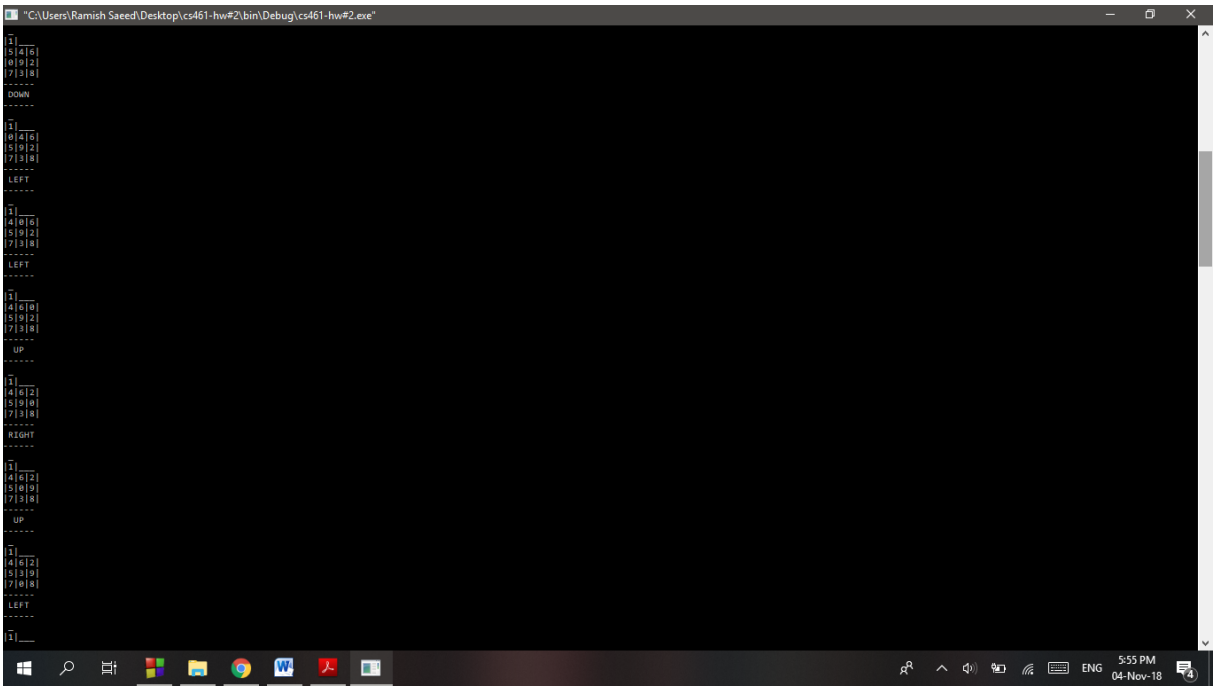
_
|1|___
5	4	6
7	9	2
3	0	8

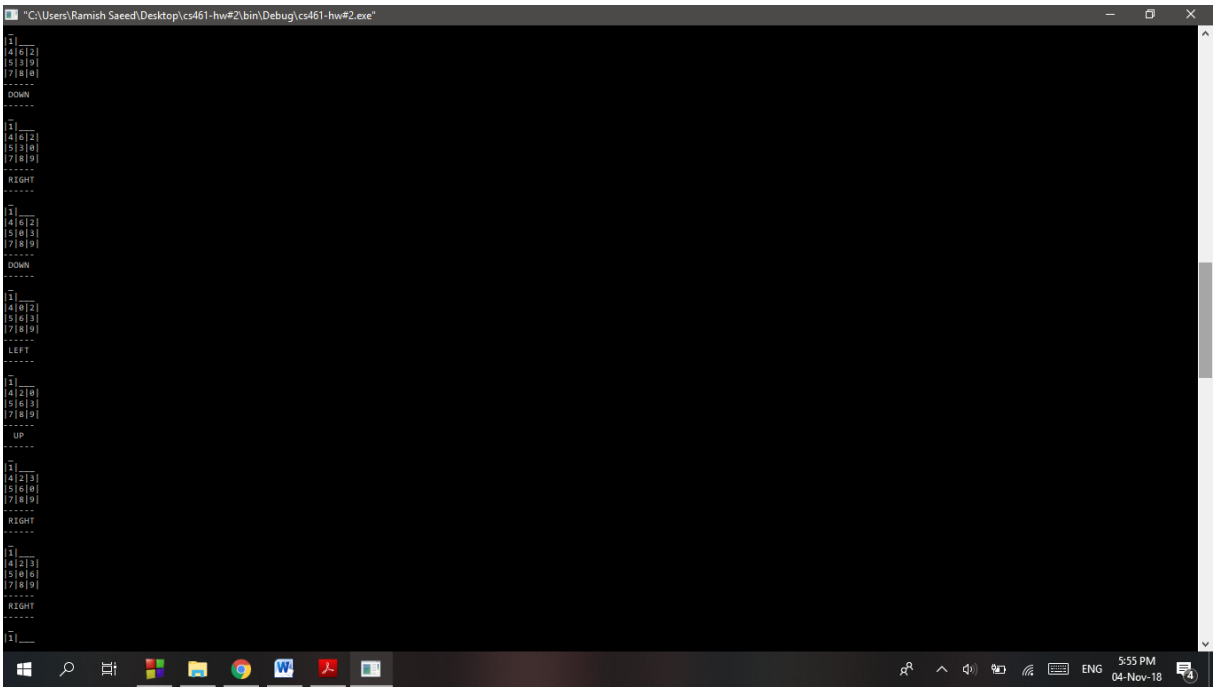
RIGHT

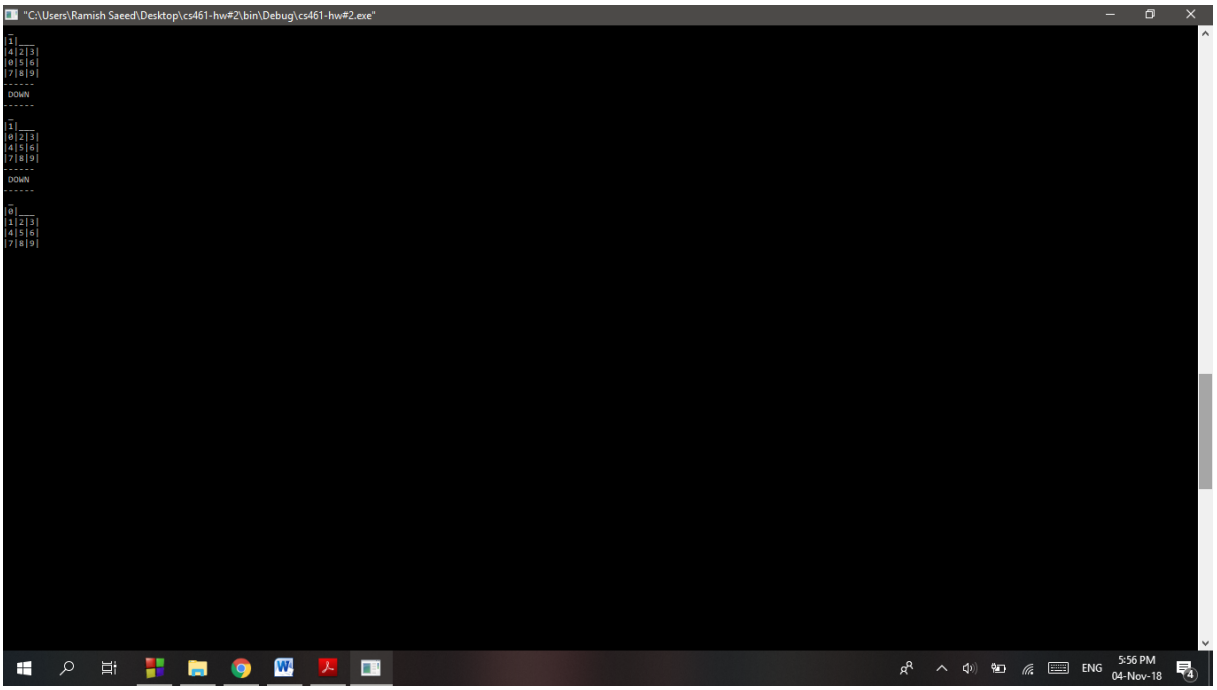
_
|1|___
5	4	6
7	9	2
0	3	8

DOWN

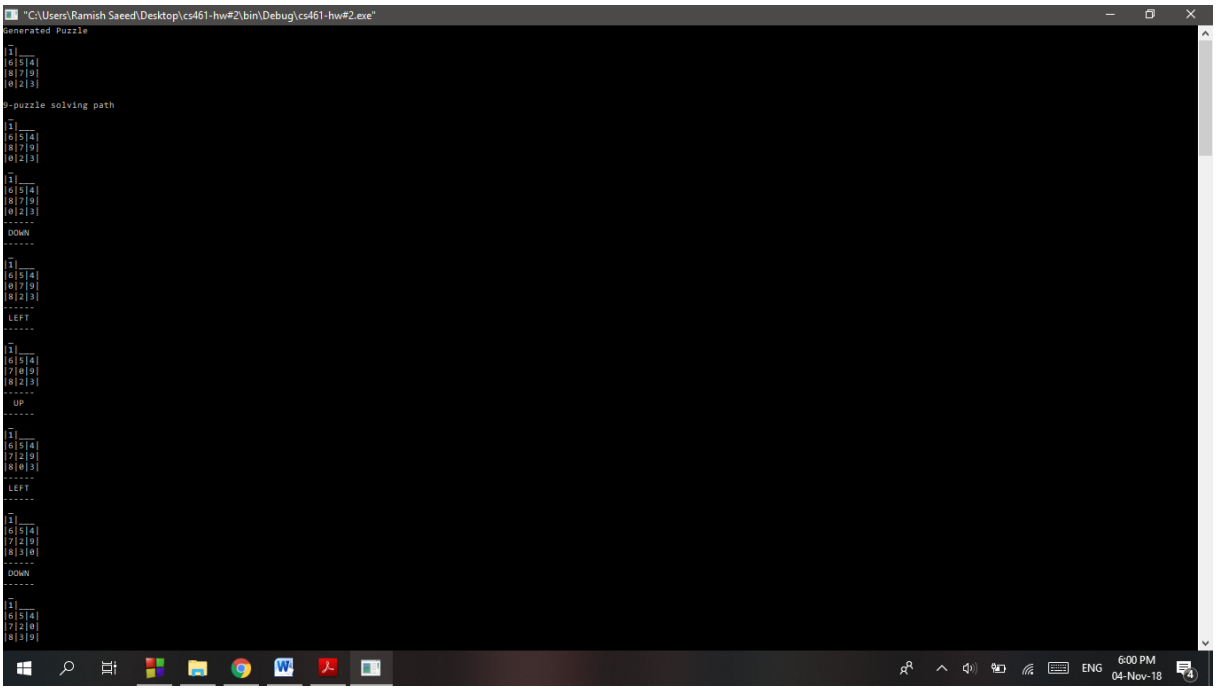
_
|1|___
ENG
5:55 PM
04-Nov-18

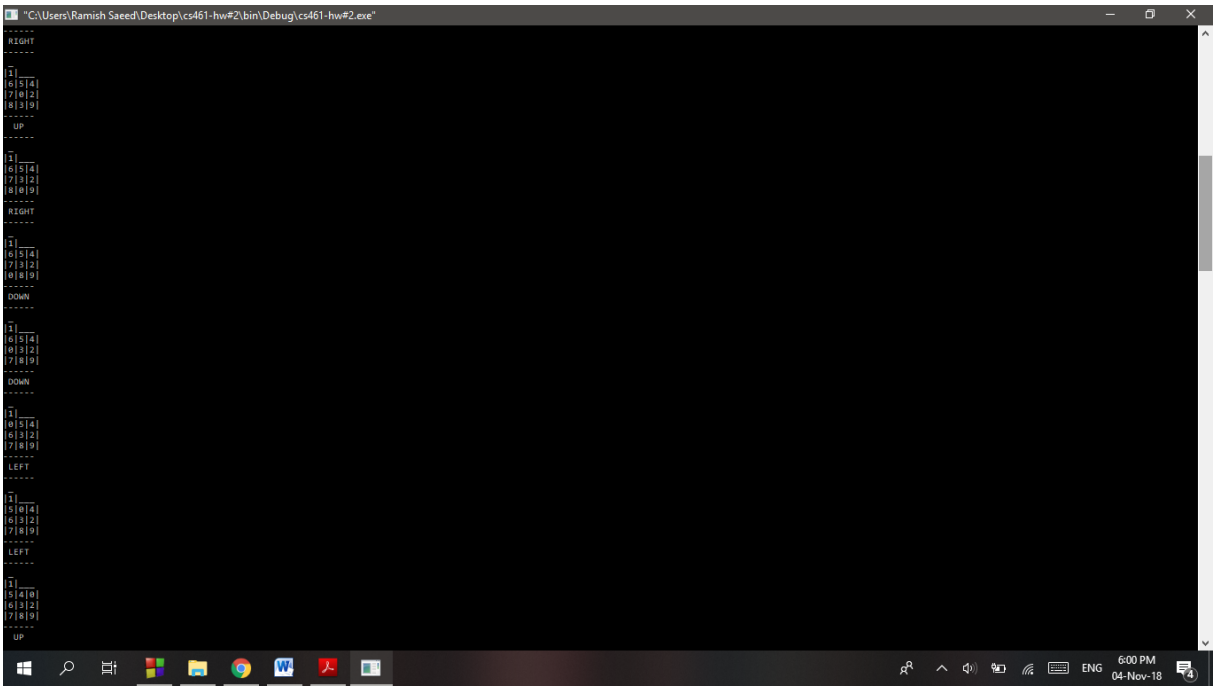

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
LEFT
LEFT
UP
RIGHT
UP
LEFT
ENG
5:55 PM
04-Nov-18

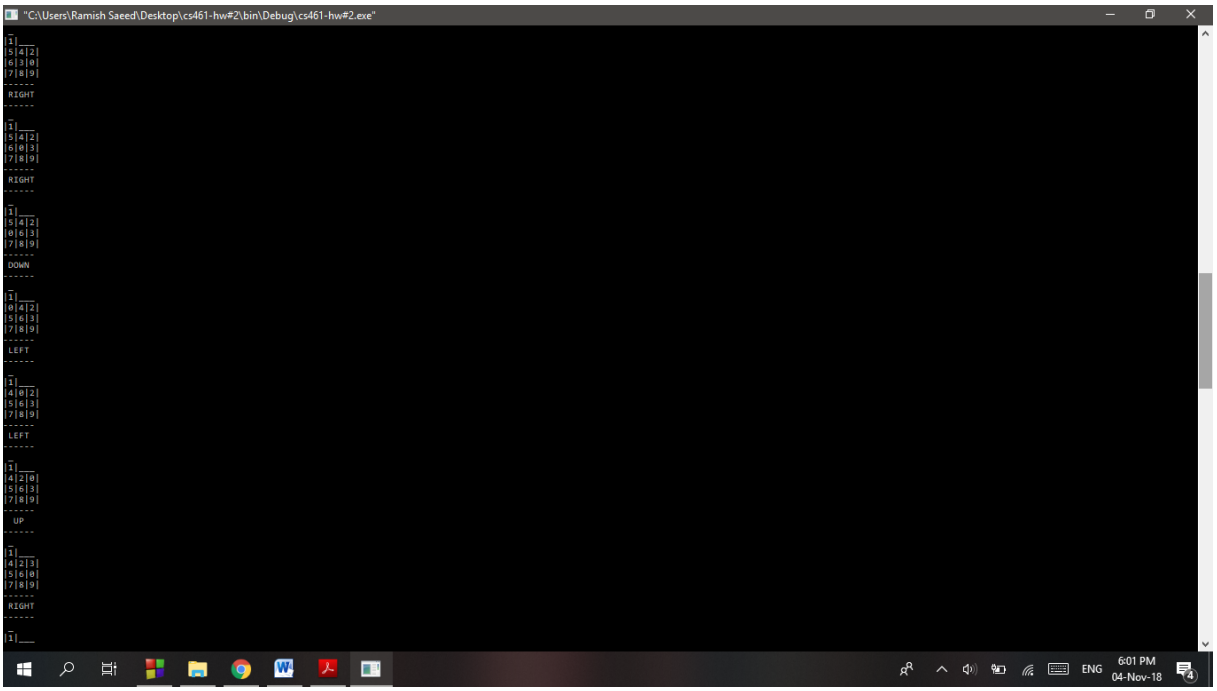

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
RIGHT
DOWN
LEFT
UP
RIGHT
RIGHT
ENG
5:55 PM
04-Nov-18

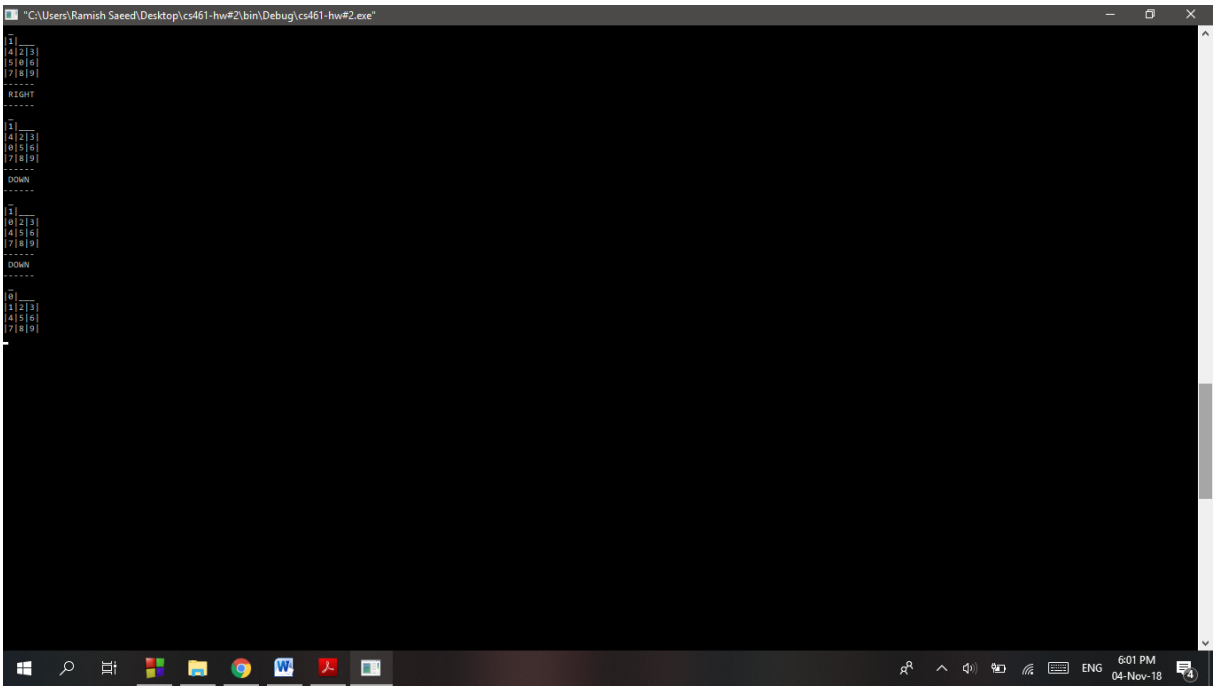

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
DOWN
ENG
5:56 PM
04-Nov-18

puzzle – 8

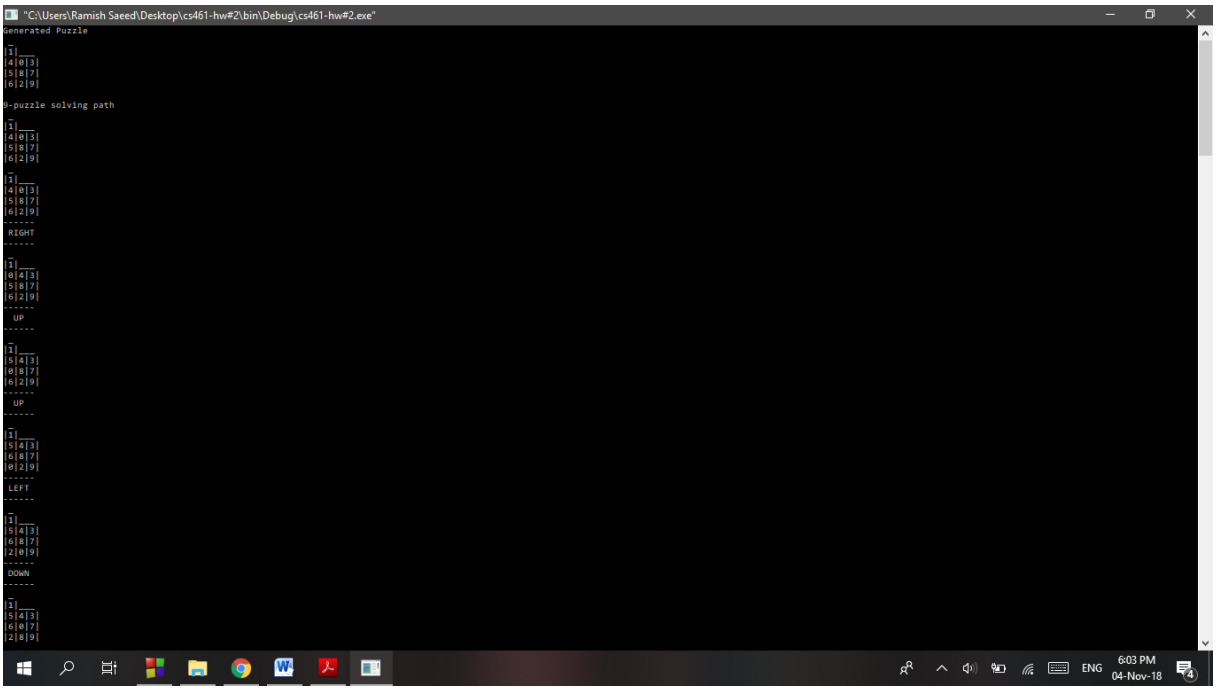

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
DOWN
LEFT
UP
LEFT
DOWN
ENG
6:00 PM
04-Nov-18

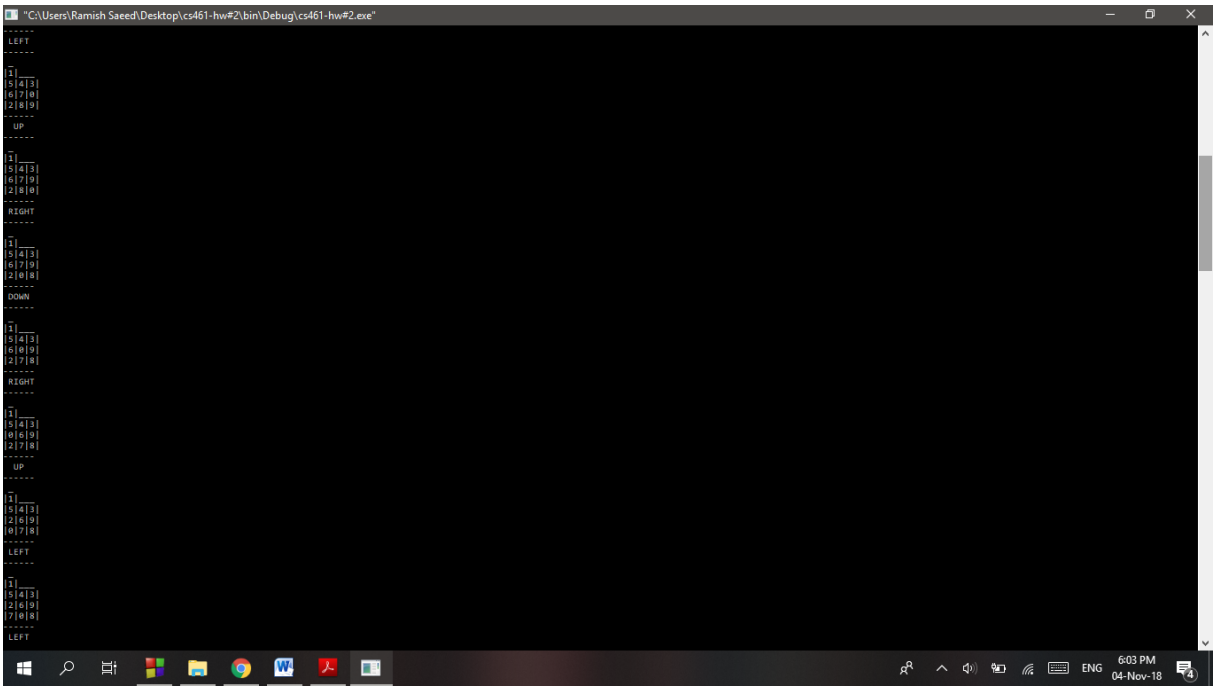

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
RIGHT
UP
RIGHT
DOWN
DOWN
LEFT
LEFT
UP
ENG
6:00 PM
04-Nov-18

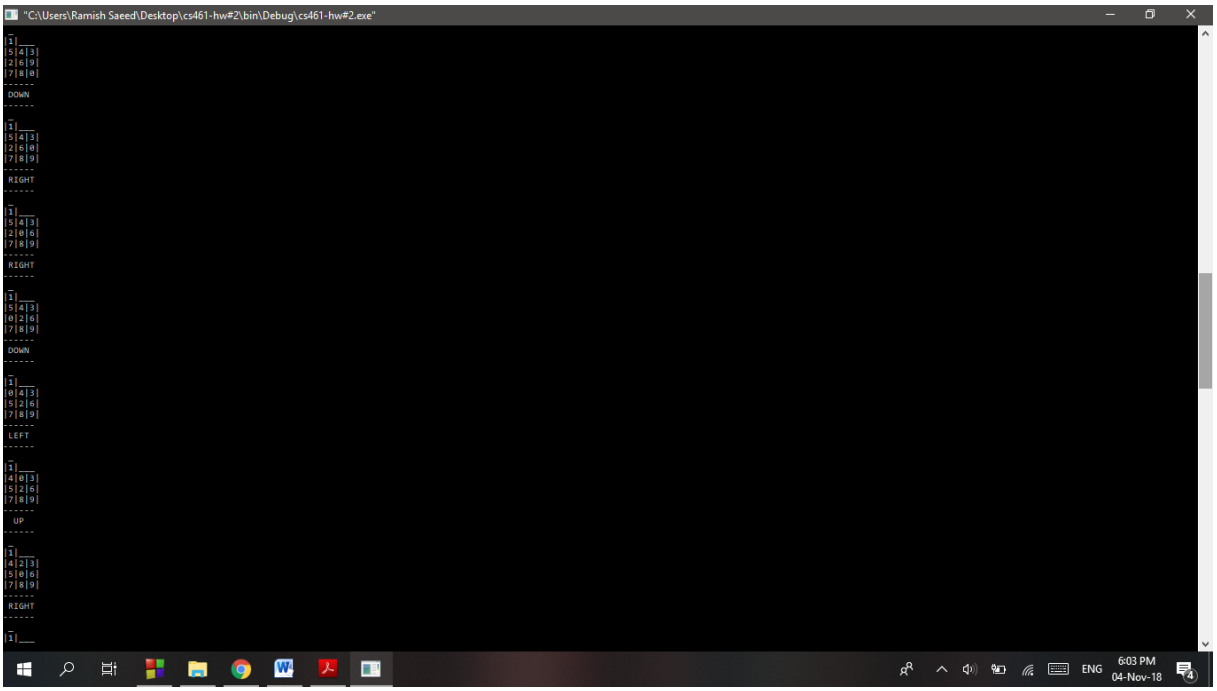

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
RIGHT
RIGHT
DOWN
LEFT
LEFT
UP
RIGHT
ENG
6:01 PM
04-Nov-18

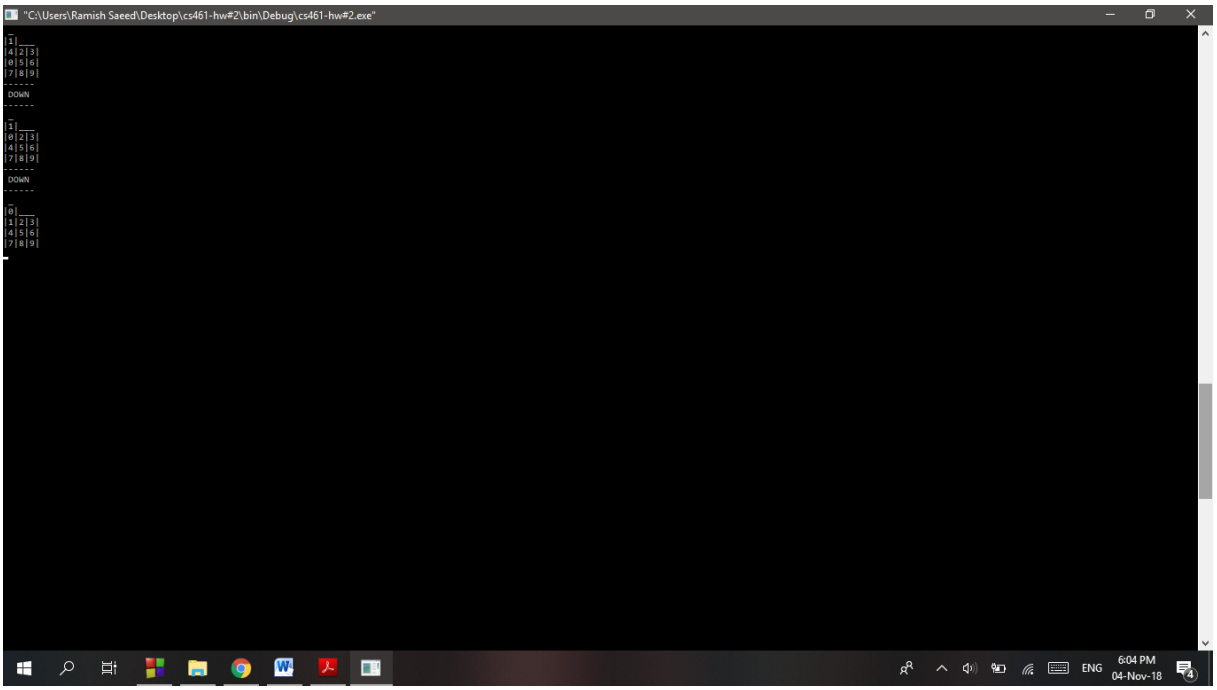

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
RIGHT
DOWN
DOWN
ENG
6:01 PM
04-Nov-18

puzzle – 9

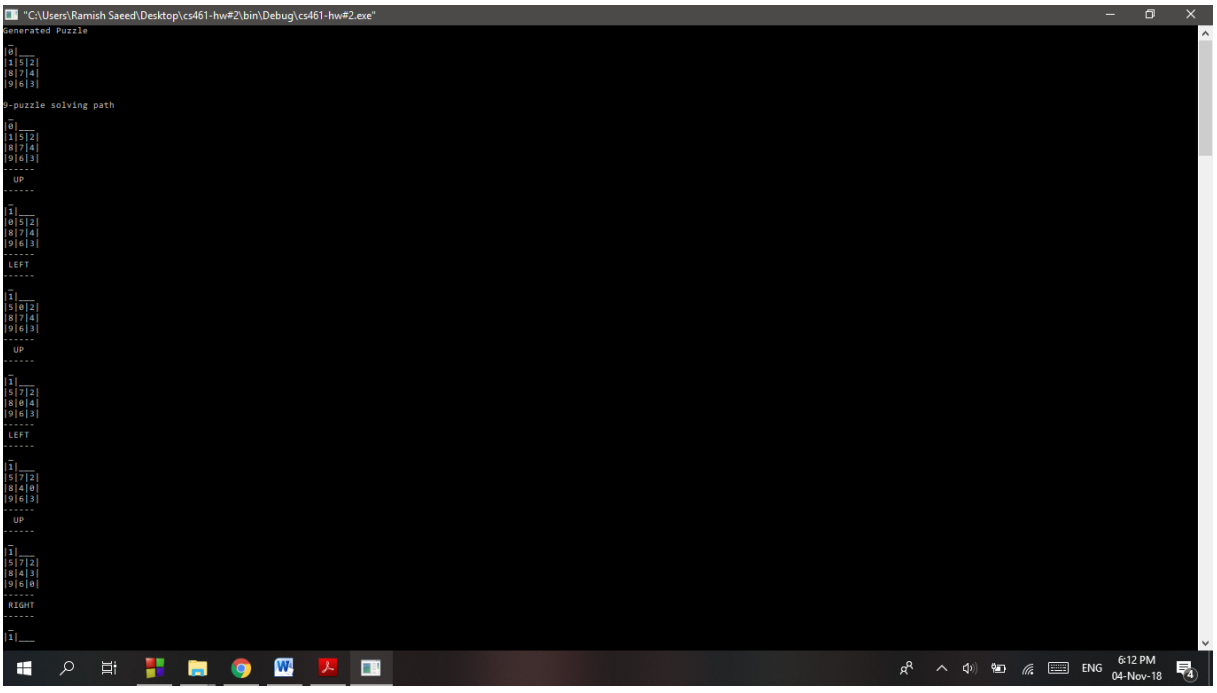

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
RIGHT
UP
UP
LEFT
DOWN
ENG
6:03 PM
04-Nov-18

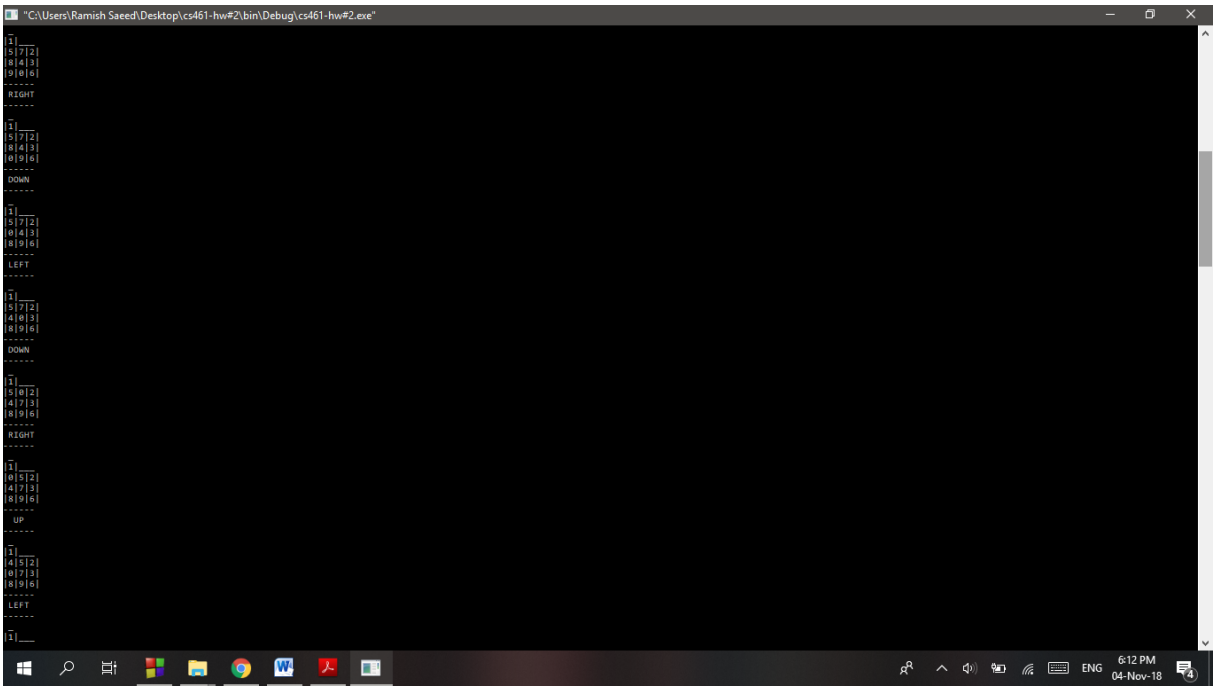

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
LEFT
UP
RIGHT
DOWN
RIGHT
UP
LEFT
LEFT
ENG
6:03 PM
04-Nov-18

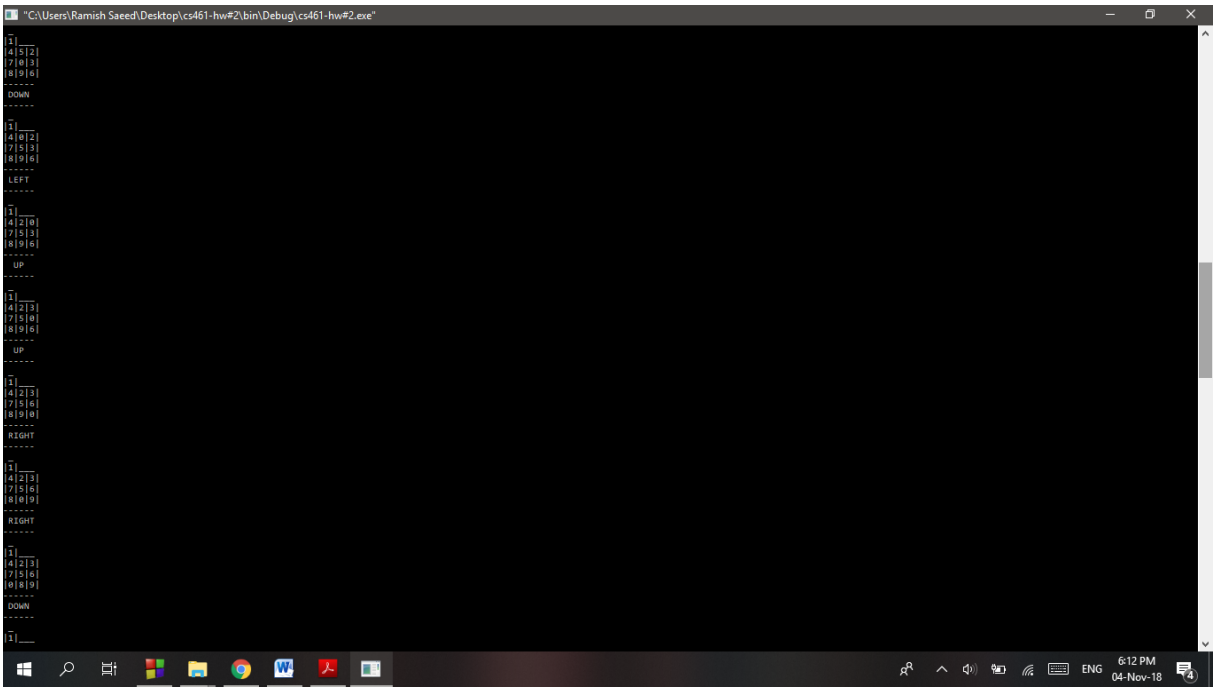

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
RIGHT
RIGHT
DOWN
LEFT
UP
RIGHT
ENG
6:03 PM
04-Nov-18

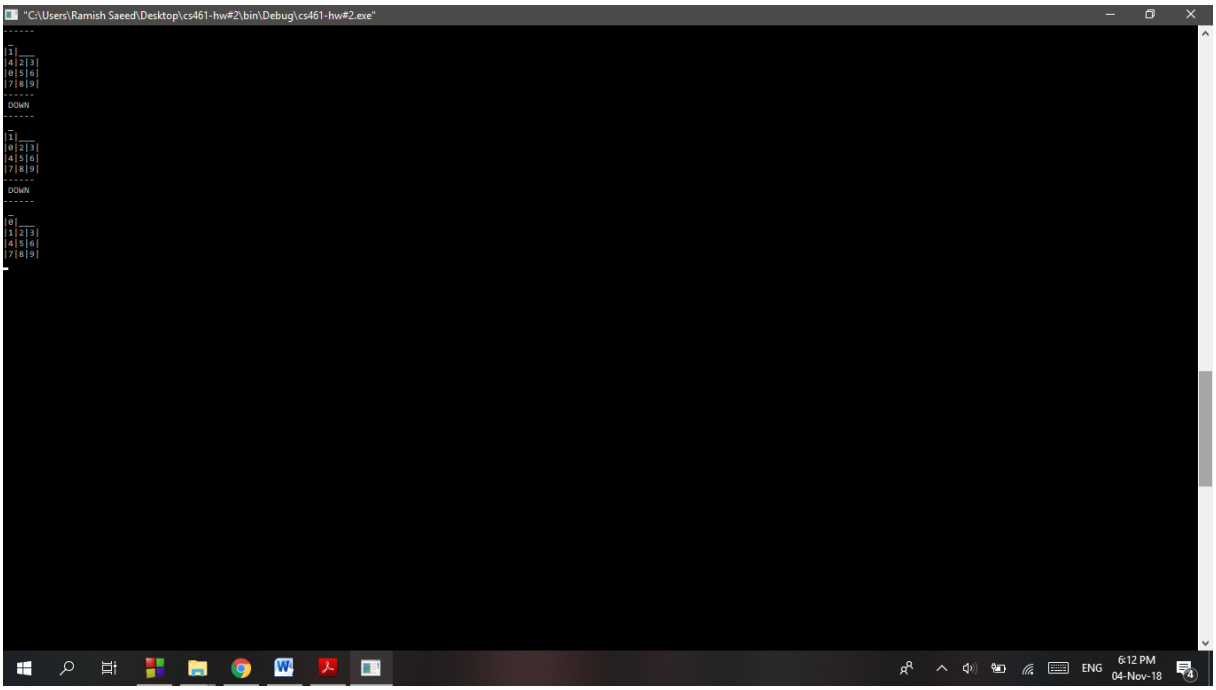

"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
DOWN
ENG
6:04 PM
04-Nov-18

puzzle – 10


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
Generated Puzzle
9-puzzle solving path
UP
LEFT
UP
LEFT
UP
RIGHT
ENG
6:12 PM
04-Nov-18


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
ENG
6:12 PM
04-Nov-18


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
ENG
6:12 PM
04-Nov-18


"C:\Users\Ramish Saeed\Desktop\cs461-hw#2\bin\Debug\cs461-hw#2.exe"
DOWN
DOWN
ENG
6:12 PM
04-Nov-18