BLG 233E DATA STRUCTURES AND LABORATORY

EXPERIMENT 7 – LABYRINTH WITH STACKS

IMPORTANT REMINDERS

1. It is not allowed to use USB sticks during the lab sessions.
2. You should unplug your ethernet cables during the lab sessions.
3. Any reference book or help material (C++) is allowed.

In this experiment, you are required to write a labyrinth solver application by yourself.

- In every step, you can move right, left, up or down.
- While traversing the labyrinth, you should show location updates on the labyrinth step by step.
- You can represent the path with 'o' , while traversing the labyrinth.
- If you encounter any dead ends, you should move back in the labyrinth. Also the path leads to dead end should be removed.
- If the labyrinth includes any cycles, your application should detect the cycles and still find a path from the initial position to the exit position. You are expected to use an extra stack to solve the cycle problem.
- At the final stage, a path from initial position to exit position should be shown on the labyrinth. You are also expected to store all the points on the path in an extra stack and print the points (i.e., x and y positions of points) on the path to the console using this extra stack at the end of the program.
- The example labyrinth is given below. In this representation, #'s represent walls, the empty spaces represent paths, o represent initial position and e presents exit positions.

```
char Maze[][] =
{
"#o###################",
"# # # # #",
"# ### # ###### ## # #",
"# # # # # # # #",
"# # ### ######## ## #",
"# # # ## # #",
"### # # # ## # # ####",
"# # # # # # #",
"# # # # # ######### #",
"# # # # # # # #",
"# # # # # ##### #",
"# # ##### # # #####",
"# # # #",
"###################E#" }
```

```
#o###################
#     #      #    # #
# ### # ###### ## # #
# # #        # #  # #
# # ### ######## ## #
#     # # ##   #    #
### # # # ## # # ####
# # # # #    #      #
# # # # # ######### #
# # # # # #       # #
# # #   # # #####   #
# # ##### # #   #####
#         #   #     #
###################E#
```