**Special Instructions:**

NOTE: A Windows device may be required to run this program. This is because this program uses the system library, which may not be functional on Mac or Linux Devices. If needed, ‘system’ commands can be removed, and the program should function properly.

**Screenshots:**

Task 2:

Text

Description automatically generated

Text

Description automatically generated

A picture containing text

Description automatically generated

Task 3:

Did not have time to implement.

Task 4:

Text

Description automatically generated

Text

Description automatically generated

**Prompts:**

**Questions –**

1. The objectives explored in this lab include stacks and managing and adding items on the stack. We also explored queues which can be used to track the player’s history in this case. This is important to know in CS and beyond, as these structures are fundamental in many CS and math-based concepts, including the ideas of arrays and vectors.

3) While we did not have time to implement the 2-player version, some strategies that can be used include moving smaller disks first to prevent the other player from moving larger ones to a space. Another is cooperating with them in spaces so you can more efficiently move your disks across the pegs. You can also try and ignore the other player but at some point, that will not really be possible.

4) While we did not have time to implement the queue class vector, it could be done by making a vector and storing the moves in each space. This would allow for printing out of the move history. In terms of automation, which was implemented, we used the idea of recursion to create the best method of completing the game. This can be seen after the game is won and shows the most optimal steps for completing the tower of Hanoi. Moving steps can be calculated manually via 2^disks – 1.