# Part2. Ben Cotton. Registration: 100389259

Question 2.1

In the *move* function, the conditions that currently check whether the position of the blank space is within the grid’s bounds would be removed. Instead, they would be replaced with a condition that checks whether the updated position is within the bounds and updates the position to either the start or the end of the column. In the *manhattan* function, the circular moves would be taken into consideration. This could be done by checking if the difference in the row is the same as the height of the grid minus 1. Therefore, if a value would normally have had to cross the entirety of the height of the grid, now, it will only have to move 1 up or down. This means that the manhattan distances generated will be much lower and, thus, be solved faster and in fewer moves.

Question 2.2

When it comes to vertically shifting the rows, the storage of predecessors and the checking of goals states must change. Each state will have 2 (in the 3x3 example) different synonymous states. My solution to this problem would be implementing a circular search within a new function, *check\_states. check\_states* will loop through the 3 (in the 3x3 example) possible row shifts and yield them as a generator. The use of generators is essential to result in high efficiency. Generators do not commit the results to memory like a list, meaning search speed will increase. This is especially pertinent because with this wrapping modification to the *n*-tile problem, 3 to 4 moves can be made per state as opposed to the 2 to 4 previously (both using the 3x3 example). The generator results can then be iterated to check if any state within the results is already within the *predecessor* dictionary or if any states within the results are the goal state.

Question 2.3

The new *n*-tile problem could only physically be a cylinder or a sphere, it must connect from the top to the bottom. When using a cylinder, curved tiles can be slid around the circumference to the other side to create the effect of vertical movement from top to bottom.