# Part2. Dan Willis. Registration: UXK18QAU 100263247

Question 1 :

(It is unclear from the brief if just rows or just columns or both can wrap around so I assumed just columns as this was used in the example)

The successor function would need to be altered to remove the checks for all of the North and South moves as they can all happen at any time. However the opposite of the last move check I would still keep. East and West would be unchanged.

The makeMove function would need to be changed to use a modulus of the dimensions of the grid to allow the wrapping to take place. eg 2 + 1 % 3 is 0 and 0 -1 % 3 is 2.

Also the h function would need to be altered so that the Manhattan distance takes into account the wrap around.

Question 2 :

To expand on my answer given in question one.

The vertical shifts don’t affect the relative position of the tiles with each other. So a vertical down shift, followed by a move, followed by a vertical up shift will always be the same as just making the move and have the same cost. Therefore, I think you can ignore the vertical shifts for the main part of the algorithm and handle them at the beginning or the end instead. You’d do that by having either X different start states (each shifted vertically) or X different goal states (again shifted vertically) and running it normally then adding the vertical shifts at the very end. I think I’d use the X goal states and change my algorithm to search for the different goal states from the root goal state then append the shifts to the end depending on which goal state it finds. There would be as many goal states as there are rows in the grid.

Question 3 :

A sliding tile barrel/cylinder puzzle, so spinning/rotating the puzzle without moving the tiles is zero cost.