CMPUT 366 Programming Assignment 2 Report Part 1
Greg Knoblauch & Bryce Cartman

Why 1: The tiles run from 0 to 120 (bottom-left to top-right) then 121 to 241, etc as we treat each tiling separately and thus we will need higher indexes to represent further tiles from the first tile. Each tileCoding is exactly 120 tiles. Example being index 726 could correspond to tiling 7 as it is a larger number than say 605 from tiling 6, as an example.

Why 2: With x and y equaling 0.1, it will be in the first tile of the first seven tilings because the input space is only moving by a fraction of -0.6/8 per tiling which is small. A small interval of movement coupled with the fact that x and y are just above and to the right of (0,0) index, the bottom left corner, gives it the ability to stay in the input space for 7 tilings.

Why 3: On the eight tiling, the point will be in the 13th tile because despite the input space moving very little (-0.6/8) each tiling, when the point goes out of the 11x11 tiling, we must add a new layer of tiles to accommodate this. The new layer of tiles is constructed to both the top and the right of the original 11x11 tiling as the input space moves **down and to the left** each tiling, creating a gap at the top and to the right that must be filled.

Why 4: The 13th tile corresponds with tile 859 because while the previous 7 tilings increment by 120, the 8th tiling must increment by 120 **plus** 11+2 = 13, as we are adding a new layer of tiles to both the top and the right of the original 11x11 tiling.

Why 5: The largest possible index is 967 because if we start in the top right corner, say x and y equaling 5.99 or 6.0, that means that for each tiling, the increment (-0.6/8) **down and to the left** will put that original point out of the input space each time it moves and thus new tiles will need to be added to correct for it.

Why 6: The second and fourth examples should have many common tiles because the x values are the same and the y values are very close to each other (2.0 vs. 2.1). They are not all the same results though because that 0.1 difference in example four is enough of a difference to push the point outside the input space and because of that, more tiles need to be added which in turn results in a higher index than the second example would produce (a difference of 11 index for tiling 5 and 6).