Every tile has a different tile number (index). Assuming that you number the tiles in the natural way, the tiles in the first tiling will run from 0 to 120, and the tiles in the second tiling will run from 121 to 241 (why?).

Math. Each tiling contains 121 tiles, and we sequence the tilings continuously. So the next group of tiles (a tiling) starts where the previous one ends and continues for 121. Thus. 121->241

A given `input` point will be in exactly one tile in each tiling. For example, the point from the first example in the training set above, in1=0.1 and in2=0.1, or 0.1,0.1, will be in the first tile of the first seven tilings, that is, in tiles 0, 121, 242, 363, 484, 605, 726 (why?). The point 0.1 is essentially 0.1/0.6 = 1/6th the length of a tile up and to the right of the tile origin. *Each new tiling effectively shifts the point 1/8th a tile. So for the 7th tiling, it has been shifted (7-1)/8th = 0.75 of the way to the origin. 7-1 because no shifting takes place in the first tiling. So still has not shifted enough to consume the point 1/6th = 0.17 away from the origin.. Therefore it's in the first tile.* In the eighth tiling this point will be in the 13th tile (why?), *By the 8th tiling, the point has been shifted by 7X1/8th = 0.875 which means that the point at 0.17 would no longer fall in this first tile. It be shifted up and to the right tile* which is tile 859 (why?) *The first tile of this tiling would be 847. But this point doesn't fall into that. It falls up a row and to the right a column. So 11+1 = 12 tiles beyond. So 847+12 = 859.*
`tilecode(0.1,0.1,tileIndices)`, then afterwards `tileIndices` will contain exactly these eight tile indices. The largest possible tile index is 967 (why?).*121 tiles / tiling X 8 tilings = 968 possible tiles. But 0 index based. So max index of 967*.

Part 2

Run SuperLearn.py. The last line of the file calls the test1 function which calls your learning algorithm on the four example points and prints out results. As a result of learning, your approximate function value should move 10% of the way from its original value towards the target value. The before value of the fourth point should be nonzero (why?) *The fourth point is non-zero because it is in similar to the second point - which has already been learned about and is thus non-zero. This is demonstrates the power of generalization.*

You should see the MSE coming down smoothly from about 0.25 to almost 0.1 and staying there. *Why does it not decrease closer to 0? It doesn't continue to decrease further because the target function itself is not deterministic. It has a variance of 0.1. So a function that perfectly predicted the future value would still be on average off because of this variance.*