Problem 1

The permutations ρ and σ are *not* conjugates. The reason is that they have different cycle structure (Lemma 6.2). Notice that ρ in cycle notation is simply $(132)(45)$, whereas $\sigma = (2543)$. Observe that ρ has two cycles of length 3 and 2, whereas σ has one cycle of size 1 and another of size 4. This is enough to conclude that their cycle structures do not match.

Problem 2

The answer is $N = 4$, i.e. we can apply σ^{-1} four times to get back to the identity permutation, in cycle notation $(1)(2)(3)(4)(5)(6)$. First we represent the shuffle/permutation in cycle notation, i.e. $\sigma^{-1} = (1)(2453)(6) = (2453)$, (recall that for convenience singleton loops are omitted from cycle notation).Then

$$\sigma^{-2} = (2453) \circ (2453) = (25)(43).$$

Hence

$$\sigma^{-4} = (25)(43) \circ (25)(43) = (2)(3)(4)(5).$$

Note that in fact the order of any cycle ρ of length n is n. That is, n is the smallest number such that ρ^n is the identity.

Problem 3

The graphs are *not* isomorphic. One way to see this is by coloring the graphs according to their degree, let us for example color the vertices with degree 5 red and the vertices with degree 4 green.

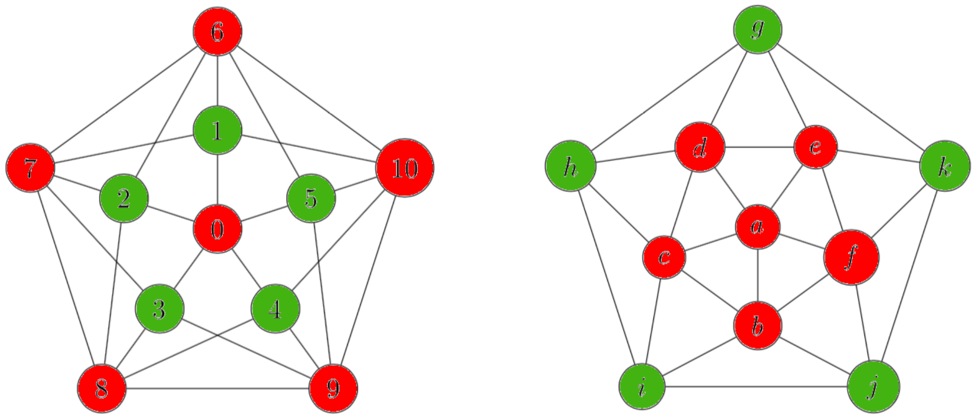

Figure 1. Graphs colored according to vertex degree.

We observe that in the second graph, all vertices with degree 5 are connected, but this is not the case in the first. Hence, there is no mapping from vertices of the first graph to vertices of the second graph, such that edge information is preserved, because their structure is different.

Problem 4

First we label the vertices of the Petersen graph arbitrarily (this is just so that we give names to nodes.

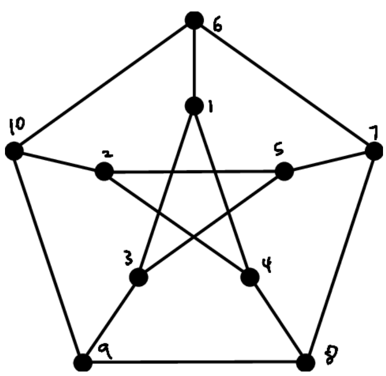

Figure 2.

Next we can color the graph greedily, via (for example) the following ordering: (1, 2, 7, 9, 5, 6, 8, 3, 4, 10). We use the convention that 1 = red, 2 = blue and 3 = green. An example run of the greedy algorithm would be the following. Start with coloring the first vertex in the ordering (incidentally this is 1) by 1, a.k.a. red. Then for the next vertex in the ordering (2) find the minimum color such that there is no conflict with the neighbours of 2, this is again (conveniently) 1/red. Analogous steps are performed for 7 and 9. Now (for vertex 5 which comes after 9) we must use a different color, since 5 shares an edge with 7, which is of color 1/red. The process repeats till we need to use a different color (3 or "green") for 3, 4 and 10.

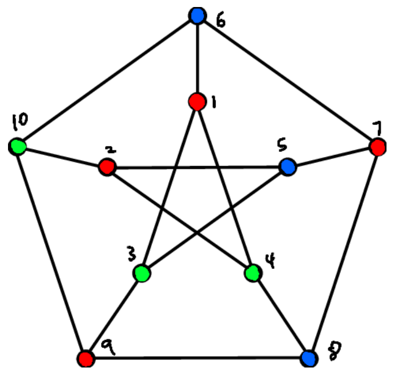

FIGURE 3.