Question 5a

Assume we have a graph G with original edge weights and its corresponding MST, denoted as T. We want to show that after adding a positive constant, C, to all the edge weights in G, resulting in a new graph G', the MST of G' (denoted as T') remains the same as T.

To prove this, we will employ a hypothesis reduction proof. We will assume that T and T' are different, and then demonstrate that this leads to a contradiction.

Hypothesis:
Let's assume that T and T' are not equal. This implies that there must exist an edge, denoted as ***c'***, in T' that is not present in T, or an edge, denoted as ***c***, in T that is absent in T'.

Proof:

Case 1: ***c'*** is in T' but not in T.

1. Since T is an MST of G, it consists of edges with the smallest weights in G.
2. When we add a positive constant, C, to all edge weights in G to obtain G', the weight of ***c'*** in G' must be greater than the weight of its corresponding edge ***e*** in G. This is because we increased all the weights uniformly.
3. However, this contradicts the fact that T' is an MST of G', as T' should include the edges with the smallest weights in G'. Therefore, **c'** cannot exist in T'.

Case 2: ***c*** is in T but not in T'.

1. Like Case 1, since T' is an MST of G', it should consist of the edges with the smallest weights in G'.
2. Considering the rescaling of edge weights, the weight of e in G must be greater than the weight of its corresponding edge **c'** in G'.
3. Again, this contradicts the fact that T is an MST of G, as T should contain the edges with the smallest weights in G. Hence, **c** cannot exist in T'.

Since both cases result in contradictions, our assumption that T and T' are different is false. We can conclude that T and T' are indeed equal, implying that adding a positive constant to all edge weights of a graph G does not impact its MST. We have shown that rescaling the edge weights of a graph G by adding a positive constant to all of them does not affect the Minimum Spanning Tree (MST).

Question 5b

Hypothesis:
Let G be a graph that contains edges with negative edge weights.

Proof:

⇒ Let T be the MST generated by Prim's algorithm on G. We need to show that T is indeed the MST. Assume, for the sake of contradiction, that T is not the MST of G. This means there exists another tree T' with a smaller total weight than T.

⇒ Consider the first edge, e = (u, v), in T' that is not present in T. Let e' = (u', v') be the corresponding edge in T, such that u' is in the tree T and v' is not.

⇒ Since T' has a smaller total weight than T, the weight of e' must be greater than the weight of e. However, since G contains edges with negative weights, the weight of e' could also be negative.

⇒ Now, let's consider the cut C formed by removing T' from G. This cut partitions the vertices of G into two sets: S (vertices in T') and V - S (vertices not in T'). Since e = (u, v) is in T' but not in T, it must cross the cut C. Without loss of generality, let u be in S and v be in V - S.

⇒ Recall that e' = (u', v') was chosen by Prim's algorithm instead of e. Since e' was selected, its weight must be smaller than or equal to the weight of any other edge crossing the cut C. However, if the weight of e' is negative, and it is smaller than the weight of e, it contradicts the fact that the weight of e' is the smallest among all edges crossing the cut C.

Therefore, our assumption that T' exists must be false, and T is indeed the MST of G.

Conclusion:
Prim's algorithm remains valid and correctly identifies the MST even when the graph G contains edges with negative edge weights. By considering a contradiction, we have shown that the MST generated by Prim's algorithm, T, is the correct MST of G.