Part B

1. (a) The answer is Yes. A simple way to think about it is to break the ties in some fashion and then run the stable matching algorithm on the resulting preference lists. We can for example break the ties lexicographically—that is, if a b is indifferent between two g_i, g_j then g_i appears on b's preference list before g_j if $i < j$ and if $j < i$ g_j appears before g_i. Similarly, if g is indifferent between b_i, b_j then b_i appears on g's preference list before b_j if $i < j$ and if $j < i$ b_j appears before b_i.

   Now that we have a set of concrete preference lists, we run the stable matching algorithm. We claim that the matching produced would have no strong instability. But the latter claim is true because any strong instability would be an instability for the match produced by the algorithm, yet we know that the algorithm produced a stable matching—a matching with no instabilities.

   (b) The answer is No. The following is a simple counterexample. Let $n = 2$, and consider the following preference lists:

b_1	$g_1 =_1 g_2$
b_2	*it can be anything*
g_1	$b_1 <^1 b_2$
g_2	$b_1 <^2 b_2$

   There is no matching without weak instability in this example, since regardless of who was matched with b_1, the other g together with b_1 would form a weak instability.

2. We do (b); it is possible to get a more desirable partner by falsely switching order of the preferences.

   Assume that we have three b's and three g's with the preference lists given in the table below, where the last column we have a false list of preferences of g_3.

b_1	b_2	b_3	g_1	g_2	g_3	g_3'
g_3	g_1	g_3	b_1	b_1	b_2	b_2
g_1	g_2	g_1	b_2	b_2	b_1	b_3
g_2	g_2	g_2	b_3	b_3	b_3	b_1

   Now run the algorithm with the true column, and note that we obtain the following pairing: $(b_1, g_3), (b_2, g_1), (b_3, g_2)$. On the other hand, if we run the algorithm with the false column we get the pairing $(b_1, g_1), (b_2, g_3), (b_3, g_2)$.

   So g_3 ends up with m_2 who is her true favorite.