Name: Parashara Ramesh	**Student Id**: A0285647M	**NUSNET Id**: E1216292

1. **Factor_evidence**

   a. I reused the "observe_evidence" code from Lab1.
   b. Post-processing involved assigning -1 to rows without observations.
   c. Rows with -1 and columns with observed values were removed.
   d. Since my implementation uses an OrderedDict the expected value order is still maintained

2. **_update_mrf_w_evidence**

   a. The function applies evidence to each factor and removes observed variables.
   b. When a variable is removed, the corresponding node and edge for that factor are also removed.
   c. The function ensures that no empty factors are returned.
   d. This process can result in split graphs with multiple connected components.

3. **_get_jt_clique_and_edges**

   a. Determine default elimination order for UGM and generate reconstituted graph using the algorithm covered in the lecture..
   b. Identify cliques in reconstituted graph with NetworkX.
   c. Create new clique/cluster graph, assigning each clique as a node and establish edges between cliques with non empty sepsets
   d. Return clique list and bidirectional clique edges.

4. **_get_clique_factors**

   a. After identifying cliques and their edges, the goal is to assign the input list of factors (from the original graph) to each clique while ensuring that each factor is used only once to determine the clique's potential (or factor).
   b. A greedy approach is employed to allocate unused factors to cliques until the minimum required number of factors “covers” each clique’s random variables.
   c. The "factor_product" operation is applied to potential factors for each clique, resulting in the clique's factor.
   d. Any remaining unused factors are assigned to a clique where they are within its scope.
   e. This process ensures that each clique has a final factor derived from applying "factor_product" to some of the input factors, with each factor being used exactly once.

5. **_get_clique_potentials**

   a. With the identified cliques and edges, a NetworkX clique graph is constructed.
   b. Considering potentially disjoint graphs due to evidence updates, a root node is selected for each component, and the sum-product algorithm is applied independently to each component.
   c. The message passing protocol is then used in both the upward and downward passes, with messages structured appropriately to marginalize any extra variables present in the sender node which is not in the sepset between 2 clique nodes.
   d. Using computed messages in both directions for all components, unnormalized clique probabilities (given evidence) are determined by combining the clique's factor with received messages from neighbors.
   e. Normalized clique probabilities are calculated by dividing by a normalizing factor obtained through marginalization of all random variables in the clique.
   f. The normalized clique probability is returned as the clique's potential.

6. **_get_node_marginal_probabilities**

   a. For each given query node, the corresponding clique it belongs to is identified by picking the first matching clique.
   b. Within this clique, the marginalization process is applied to eliminate nuisance random variables apart from the query random variable
   c. This process results in the marginal probability of the query node given the provided evidence.