Sample questions for our last class:

1. If we try putting the following key-values in a hash table that uses a bucket array of size 17, uses identity as hash code function for integers, the MAD compressor with a = 3 and b = 5, and quadratic probing algorithm to deal with collision, what will be the content of bucket array at the end?

   **key-values:**
   (5, "five")
   (12, "twelve")
   (39, "thirty nine")
   (22, "twenty two")

2. What is the output of the following on the given input if we use `sum_of_component` for hash code function for strings?

   ```
   void count(const vector<string> & v){
       unordered_map<string, int> counts;
       for (int i=0; i<v.size(); i++)
           counts[v[i]]++;

       for (auto itr=counts.begin(); itr!=counts.end(); itr++)
           cout << itr->first << "t" << itr->second << endl;
   }
   ```

   input : {"stop", "tops", "pot", "opt", "opts", "tops", "opt"}

3. In previous question, what is the load factor if unordered_map uses separate chaining?

4. Draw the graph that is given in the following adjacency matrix.

   ```
   0 3 1 0 0
   3 0 1 0 0
   0 1 0 1 2
   0 2 1 0 0
   0 0 0 0 0
   ```

5. Describe an algorithm (in pseudo-code or in fewer than five sentences) that takes a simple connected graph as input and outputs one cycle in the graph or prints "no cycles" if the graph is a tree.