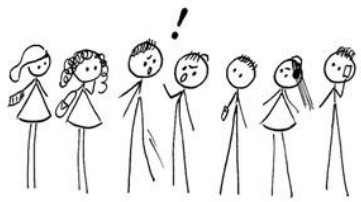
CMPS 2143 – Object Oriented Programming **Fall 2017**
Programming Assignment 5: 100 points **DUE: Wednesday, Dec. 6, 2017**

Purpose: To learn how to write a templated class and use it. To work with partners – you will be assigned partners based on the order of how you turned in the assignment. (That means late programmers will be partnered with late programmers.)

Problem: Team Queue: Most computer scientists know about data structures, such as *Queues* and *Priority Queues*. The *Team Queue*, however, is not so well known, though it occurs often in everyday life. A queue at a museum is often a team (tour group) queue, for example.

In a team queue, each element belongs to a team or group. If someone enters the queue, they first search the queue from head to tail to check if some of their *teammates* (someone of the same group) are already in the queue. If yes, they enter the queue right behind them (they cut into line). If not, they enter the queue at the tail and become the new last person in line. Dequeuing is done like in normal queues: elements are processed from head to tail in the order they appear in the team queue.



Your task is to write a program that simulates such a team queue.

Method: The first step is to implement a linked list-based implementation of a Queue class. Modify "enqueue" to work for the Team Queue. Next, template it. Finally, write a main program to use the template class to solve this problem. Or solve the problem first, then template it…

Input: The input file will contain one or more test cases. Each test case begins with the number of teams t ($1 <= t <= 10$). Then t team descriptions follow, each one consisting of the number of people belonging to the team and the people themselves. People are integers in the range 0..999. A team may consist of up to 100 elements.

Finally, a list of commands follows. There are three different kinds of commands:
- ENQUEUE x — enter element x into the team queue
- DEQUEUE — process the first element and remove it from the queue
- STOP — end of test case

The input will be terminated by a value of 0 for t.

Output: ***Output is to a file!*** For each test case, first print a line saying 'Scenario #k', where k is the number of the test case. Then, for each 'DEQUEUE' command, print the element that is dequeued on a single line. Print a blank line after each test case.

Turn in:
- Printouts of all files (3 of them)
- Printout of two input files (try 0 for number of teams and 0 for size of team)
- Printout of two output files
- Storage media containing all project files.
- Everything in a big envelope

Sample Input

```
2
3 101 102 103
3 201 202 203
ENQUEUE 101
ENQUEUE 201
ENQUEUE 102
ENQUEUE 202
ENQUEUE 103
ENQUEUE 203
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
STOP
2
5 251 252 253 254 255
6 261 262 263 264 265 26
ENQUEUE 251
ENQUEUE 261
ENQUEUE 252
ENQUEUE 253
ENQUEUE 254
ENQUEUE 255
DEQUEUE
DEQUEUE
ENQUEUE 262
ENQUEUE 263
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
STOP
0
```

Sample Output (not formatted)

```
Scenario #1
101
102
103
201
202
203
Scenario #2
251
252
253
254
255
261
….
```