**WRAV202: Practical 5**

Write a separate program for each of the following questions. Each program must read an input file and write to an output file.

You have to write your own Queue class. The most efficient way of implementing a queue is using a doubly linked list. Your class must provide at least the following four methods: Enqueue, Dequeue, Peek and Count (which keeps track of the number of elements in the queue). Use your queue implementation to answer the following questions.

# Question 1 – Queueing simulation

Professor Plum gives assignment assistance to all his students of Thursday mornings. All the students line up outside the Prof’s office early in the morning and the Prof calls them into the office one at a time. Each student receives a piece of candy every time they are called into the office (**even if they have already been called into the office previously**).

Prof Plum only gives help on one assignment question at a time, so students who need help with multiple questions are required to rejoin the queue to get help with their other questions. However, once the queue’s length drops to 20 or fewer students (not counting the student that the Prof is currently helping), Prof Plum will help each student with as many questions as they have left, so those students will not have to rejoin the queue again.

The question that you need to answer is, for a given list of students, how many pieces of candy will Prof Plum have handed out after processing the entire queue?

The input file consists of a list of student names and the number of questions that each student need help with (separated by a comma[[1]](#endnote-1)). For example:

Mike,3

Vusi,1

Julie,5

You program must read the students from the file, enqueue each into your Queue object, then dequeue from the queue (and enqueue some students again) to simulate Prof Plum’s behavior. Each time a student is dequeued you must increment the count of the number of pieces of candy that Prof Plum dispenses. Finally, the program must save the number of dispensed candies to “Outputs.txt”.

Test Inputs and the Expected Test Outputs are provided in the question folder.

# Question 2 – Vaccinations

Assume that your local vaccination center can give a maximum of 1000 vaccinations per day. Preference is given to older people but the center would also like to give some vaccinations to younger individuals. Accordingly, three queues are formed:

Queue 1 is for people 65 years or older.

Queue 2 is for people 40 years or older (but younger than 65).

Queue 3 is for people younger than 40.

The following procedure is **repeated** until the 1000 vaccine limit is reached:

*Four individuals from Queue 1 is given the vaccine (if fewer than four people remain in the queue then whatever number remains are given a vaccine). Then two people from Queue 2, followed by 1 person from Queue 3.*

You have to write a program to determine the name of the person who receives the last of the 1000 vaccines.

The input file consists of a list of names and their respective ages (separated by a comma).

Your program must read the individuals from the file and insert them into three queues. The vaccination procedure must then be simulated until no more vaccines are left. The name of the last person to vaccinated must be written to “Outputs.txt”.

Test Inputs and the Expected Test Outputs are provided in the question folder.

# SUBMISSION

You have to submit two files for this practical. Each file should have a *.cs* extension.

1. Go look at the String.Split method that C# provides to easily split delimited strings [↑](#endnote-ref-1)