

**Assignment:** Write a simulation of a checkout line that records the number of customers served, the longest time waited by a customer, and the average time waited by all customers served. Use an abstract data type that implements a dynamically allocated circular queue to model the checkout line.

**Prohibition:** Use of the C++ Standard Template Library is prohibited in the implementation of this project.

**Program Files:** Project 4 consists of files **p04.cpp**, **Sim04.h**, **Sim04.cpp**, **Queue04.h**, **Queue04.cpp**, and **p04make**. Project file names are exactly as given. Failure to employ the foregoing names will result in a score of **zero (0)** for this project

Project files must be stored in the **root directory of your student account**. Failure to store project files in the root directory of your student account will result in a score of **zero (0)** for this project.

File	Description
<b>p04.cpp</b>	File <b>p04.cpp</b> contains functions that process command line arguments and manage the checkout line.
<b>Sim04.h</b>	File <b>Sim04.h</b> contains the definition of class <i>Sim</i> . Class <i>Sim</i> defines private data and member functions specific to the simulation.
<b>Sim04.cpp</b>	File <b>Sim04.cpp</b> contains the implementation of member functions of class <i>Sim</i> .
<b>Queue04.h</b>	File <b>Queue04.h</b> contains the definition of class <i>Queue</i> . Class <i>Queue</i> implements an integer queue by dynamically allocating an array of integers. A circular queue defined over the array.
<b>Queue04.cpp</b>	File <b>Queue04.cpp</b> contains the implementation of member functions of class <i>Queue</i> .
<b>p04make</b>	File <b>p04make</b> contains instructions for creating executable file <b>p04</b> . Instructions in file <b>p04make</b> are accepted by the UNIX utility <b>make</b> .

**Command Line:** Project 4 can be invoked with zero, one, or two program parameters. The first program parameter is the input file name. The second parameter is the output file name. Sample command lines together with corresponding actions by program **p04** are shown below. Boldfaced type indicates data entered at the keyboard by the user.

**\$ p04**

Enter the input file name: **i04.dat**

Enter the output file name: **o04.dat**

**\$ p04 i04.dat**

Enter the output file name: **o04.dat**

**\$ p04 i04.dat o04.dat**

**Input File:** File **i04.dat** in the class directory `~tt/cs2613/` contains a list of codes that represent actions on the checkout line. The first action occurs at time zero. Each time an integer code is read time advances 15 seconds. Refer to Figure 1. Input file format.

**Code Definition**

- 1** Integer code **1** indicates nothing happened.
- 2** Integer code **2** indicates a customer arrived and was put at the end of the line.
- 3** Integer code **3** indicates that a customer was served.
- 4** Integer code **4** indicates that a customer arrived at the same time a customer was served.

**Figure 1.** 1 2 4 3 2 1 3 2 1 2 4 2 3 3 2 4

**Input file format**

**Output File:**

The format of the output file is shown in Figure 2. Data shown in Figure 2 is produced by Program **p04**. The output file consists of three (3) columns. The first column contains a description of the data shown. The second column lists data produced by program **p04**. The last column lists the units of the data recorded in the second column. The widths of the columns are 30 spaces, 10 spaces, and 20 spaces respectively. Information in the first column is left justified. Information in the remaining two columns is right justified.

**Column Specification**

- 1 (left)** Text is left justified in a field of 30 spaces.
- 2 (middle)** Integer data are right justified in a field of 10 spaces.
- 3 (right)** Text is right justified in a field of 20 spaces.

**Figure 2.**

<b>Output file format</b>	Customers served	7	customers
	Maximum response	60	seconds
	Simulated mean response	36	seconds

Project 4 must employ the queue discussed lecture 16.

Project 4 must employ the simulation discussed in lecture 15.

Please note that the simulation is a derived class. The base class is **class Queue**.