- Assignment 2 Information -

Start by declaring an array of servers (shop checkouts) and a variable to indicate the number of servers. Each server element should contain members for the server's priority, customer finish time, idle (or busy) flag - and other variables for keeping the statistics to be printed at the end of the program. The server array can be a global or a member a class object with an appropriate name.

The main() should start by asking for a filename, then read from the file the number of servers and the priorities which are stored in the server array. Main() should then read the customer arrival events from the file and process them (as explained below).

To help process the customer arrival and server events you should implement a priority queue. The elements of the priority queue should contain thee members: the *event type, event time* and the *service time* (for customers). The element at the top of the priority queue has the smallest *event time*.

The *event type* can be a *customer arriving* at the shop or a *customer completing payment* at a server (i.e. checkout). For example, the test data file has 5 servers. You could use the numbers 0 to 4 to indicate when a customer completes payment at a *server* (*customer complete payment* event) and the number 6 (or -1) to indicate a customer arriving at the shop (*customer arrival* event).

You should also implement a FIFO queue for storing the customers that arrive when all the servers (checkouts) are busy. The FIFO elements should store the customer's arrival time and the service time.

After initialising the server array from the file (and other objects), the main() should manage the events. The following algorithm is one example of how this could be done:

```
Read 1st CustomerArrival event from file and add it to the priority queue
do
        Get next event from event priority queue
        If Event = CustomerArrival
                Add customer to customer FIFO queue
                Read next CustomerArrival event from file
                If not EOF add Event to event priority queue
        Else // must be a CustomerCompletePayment event
                Set server[Event] to idle (and do its stats)
        End if
        While customer FIFO not empty and idle server available
                Get Next Customer from FIFO
                Find fastest idle server (see note below)...
                set server's idle flag to busy
                calculate server's finish time (and do its stats)...
                add CustomerCompletePayment event to priority queue
        end while
While event priority queue not empty and FIFO not Empty and servers busy
Print stats
```

Note: to speed up finding the fastest server (i.e. checkout) you could use another priority queue!