CS2750 **System Programming and Tools**
Assignment # 5 Due Date: April 25, 2018

C Programming – Files and Queues

Write the code in C. I want to see the use of `git` and `Makefile`. Work on each part of the project in its own subdirectory. Use the `README` file to indicate what is deing done, especially if you have done something new, and how to execute the program.

Use top-down design to design and code the implementation. Use good documentation.

1. This program is meant to exercise the queues where each element is dynamically allocated as needed. You must use the queue ADT as given in the class using the dynamic memory allocation with pointers.

   In this program, you are required to perform a simulation of the grocery store checkout lanes and compare the performance of two different queueing methods for checkout. The program will be made up of two separate programs for simulation and one for generating the times of arrival and departure in the queue. Upon analysis, it is seen that we are more interested in the actual time spent by a customer in the queue than the time of departure. Therefore, we'll keep two parameters of each customer: time of arrival and the service time requirement.

   (a) Program to generate the time parameters for the customer
   - Write a program to generate random numbers to be associated with the time of arrival and the service time requirement for the customer. The time is a logical entity and can be expressed as a float, instead of the usual HH:MM:SS format. The arrival time will increase linearly with every customer (assuming that a customer arrives, on an average, every 5 seconds). Each customer will require between 100 and 400 seconds of service time.
     This program will generate a list of two fields (as specified above) and write them to a file named `customers`.

   (b) In the second phase, you will write two programs to simulate the grocery store checkout using the file `customer`. Each program will take its input from this file and produce a report to specify the following:
   - Number of customers serviced
   - Number of customers remaining
   - Average time spent by each customer in the line
   - Average waiting time for each customer
   - Average idle time for the checkout clerks

   Assume that there are 10 checkout counters, and all of them are available for checkout when you start. You have to write the programs to show the results from two different strategies:

   i. Simulate the grocery store checkout lanes as you see them. A customer arrives and goes to the lane which has the minimum number of waiting customers. You will have to have a queue corresponding to each checkout counter in this case. This can be easily simulated by an array of ten queues.

ii. Simulate the grocery store checkout where we still have ten checkout clerks but all the customers enter a single queue. Whenever a checkout clerk is done with the current customer, she waves the next customer in the queue to step to her counter and performs the service. This is like a bakery where you have to take a number for service and your number is called when the clerk is available.

2. This program is meant to be an introduction to Unix process handling and file processing.

   You are given a file of accounts that is not sorted in any manner. The file is binary and the structure of each record is:

```
typedef struct
{
    char name[40];
    int number;
    float balance;
} acct_info_t;
```

   The file has a header as well which is two integers long. This header is meant to verify the file type with a magic number and gives the number of records. Thus the structure of the file can be described as:

```
unsigned int    magic_num;
unsigned int    num_rec;
acct_info_t     acct[num_rec];
```

   The file to use for this project is called `acct_info` and is to be found in the class directory (`delmar:~bhatias/cs2750/acct_info`).

   Your job is to tell me the balance at a given record number. Since it is a binary file, you can search different areas of the file using different processes. *You should not be using sequential read through the file but go straight to the record using `fseek(3C)`.*

   Bonus 25 points: You can earn 25 extra points by performing a regular expression search of names using `regexec(3)`.