CS 140A - Assignment 2
Due Monday, November 3

In this assignment you will be working with stacks and queues. You must use a stack for the first part of this assignment and you must use queues for the second part of this assignment. You will not receive credit for using other data structures to implement these programs. You should use a linked list to implement the stack and queue.

Write a program that reads in a file and checks for balancing symbols. You should be able to handle the following symbols (), {}, and []. The symbols may be nested throughout the program. You will use a stack for this program. When you read an opening symbol you should push it on the stack. When you read a closing symbol you should pop a symbol off the stack and make sure that the symbols are of the same type. If there is an error with the symbol matching, an error message should be displayed specifying the line of the first unmatched symbol.

Traffic lights at intersections often need to be analyzed for their efficiency. Write a program that uses a queue to simulate and evaluate a traffic signal. The arrival pattern of cars will be represented in a file as the arrival times (in seconds) of each car. You may assume that the arrival times in the file are in order. You should process the cars in the file and then print out the average waiting time at the intersection as well as the average number of cars that go through the intersection during a green light.

You can implement this simulation as follows. First, read the arrival file and push the cars onto an *arrival* queue. Next, set a simulation time variable to 0, and assume the light is initially red. The light stays red for one minute. Pop the first car from the *arrival* queue and check its arrival time. If it is less than 60 seconds, push the car onto a *waiting_at_intersection* queue. Repeat this process until the next car's arrival time is 60 seconds or more. Then compute the amount of time the waiting cars must wait until the light turns green (60 - arrival time). At time 60 the light turns green for 20 seconds, and each car requires 2 seconds to pass through the intersection (thus up to 10 cars can pass during a green light). Pop up to 10 cars from the *waiting_at_intersection* queue (be sure to keep track of the number of cars that went through). Update your timer to $60 + 20 = 80$, and then repeat the above process until both queues are empty. Print the average waiting time for cars at the traffic light and the average number of cars through each green light.

A simple example is shown below. The arrival file contains:

```
  3
 10
 15
 24
 30
 32
 44
 48
 50
 53
 58
 85
 90
110
  0   (0 marks the end of the simulation)
```

The light would turn green twice. The first time the light would turn green (at time 60), 11 cars would already be waiting. Only 10 would be allowed through. The second light (at time 140) would allow the one car left as well as the others that later arrived. The average number of cars per green light would be $(10 + 4)/2 = 7$. The average waiting time in seconds would be
$(57 + 50 + 45 + 36 + 30 + 28 + 16 + 12 + 10 + 7 + 82 + 55 + 50 + 30)/14 = 36.3$

Submit your program electronically using the instructions provided by the TA. Be sure to use good programming style and include meaningful, thorough comments. Remember that no credit will be given if your program does not compile.