NAME:

CS 2301
Exam 3
B-Term 2011

Questions 1-3:	_____	(15)
Question 4:	_____	(15)
Question 5:	_____	(20)
Question 6:	_____	(10)
Question 7:	_____	(15)
Question 8:	_____	(15)
Question 9:	_____	(10)
TOTAL:	_____	(100)

You may refer to one sheet of notes as you take this exam. Notes may not be shared between students during the exam. Please do not open the exam until you are told to do so.

For problems 1-3, circle the statement that best answers the question (5 points each).

1. Select the one FALSE statement about binary trees.

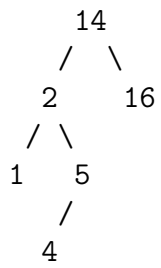
   (a) Every binary tree has at least one node.

   (b) Every non-empty binary tree has exactly one root node.

   (c) Every node in a binary tree has at most two children.

   (d) Every non-root node in a binary tree has exactly one parent.

2. Consider this binary search tree:



   Suppose we remove the root, replacing it with something from the left subtree. What will be the new value in the root of the tree?

   (a) 1

   (b) 2

   (c) 4

   (d) 5

   (e) 16

3. Suppose we use a fixed-length, wrap-around array to implement a queue. The array can hold 100 items. At some point in time there are ten items in the queue, stored at array positions [2] through [11]. If we enqueue another item, at what array position will it be enqueued?

   (a) 1

   (b) 2

   (c) 11

   (d) 12

4. (15 points) Write a C function called `printList` that displays the data field of each node in a linked list. *Your function must be recursive.* You may assume that the following definition is used for `struct listNode`:

```
struct listNode{
  int data;
  struct listNode *next;
};
```

Hint: use an if-statement, not a looping statement.

Here is the documentation and heading for the function; you need to write the body of the function.

```
// PRE:  head is a pointer to a linked list.  The list may be
//       either empty or non-empty.
// POST: the value in the data field of every node has been displayed.

void printList(struct listNode *head)
```

5. (20 points) For this problem, you will implement a function two different ways, once for a linked-list queue and once for a fixed-length array queue. Here is the prototype of the function you will implement:

```
// returns the number of elements currently in the queue
int currentQueueLength(void);
```

(a) Here are the global definitions available in the file Queue.c for a queue implemented as a fixed-length, wrap-around array:

```
typedef QUEUE_ELEMENT int;

struct queue{
  QUEUE_ELEMENT data[100];
  int front;    // the subscript of the element at the front of the queue
  int rear;     // the subscript of the element at the rear of the queue
  int count;    // the number of elements currently in the queue
};

struct queue Q;
```

Provide the complete function definition for `currentQueueLength()` using these definitions.

(b) Here are the global definitions available in the file Queue.c for a queue implemented as a linked list:

```
typedef QUEUE_ELEMENT int;

struct qNode{
  QUEUE_ELEMENT data;
  struct qNode *next;
};

struct queue{
  struct qNode *head;  // pointer to the first node in the linked list
  struct qNode *tail;  // pointer to the last node in the linked list
};

struct queue Q;
```

Note that in the linked list version, the queue is considered empty when `head` has the value NULL. Provide the complete function definition for `currentQueueLength()` using these definitions.

6. (10 points) Here is the main function from Homework 5:

```
// A demonstration project to simulate a queue in, say, a supermarket or a bank.
#include <stdio.h>
#include <stdlib.h> // Provides atol, rand, etc.
#include <ctype.h> // Provides tolower, etc.
#include "Simulation.h" // accesses the Simulation function
#include "Statistics.h" // Gathers and prints statistics about the simulation

int main(int argc, char **argv){
  int verbose = 0;             // controls printing out gory detail
  int seed = 1;                // seed for random number generator
  double simTime = 720;        // number of minutes to simulate
  double meanInterval = 2.0;   // interval between arrivals
  double meanServiceTime = 2;  // mean time to serve one customer
  int i;                       // for loop control variable

  for (i = 1; i < argc; i++) {
    switch (i) {
       case 1: simTime = atof(argv[1]);
               break;
       case 2: meanInterval = atof(argv[2]);
               break;
       case 3: meanServiceTime = atof(argv[3]);
               break;
       case 4: seed = atoi(argv[4]);
               srand(seed);
               break;
       case 5: if (tolower(*argv[5]) == 'v')
               verbose=1;
               break;
       default: printf("Usage: PA5 simTime interval serviceTime seed verbose\n")
                printf("Using defaults\n");
                break;
     } // switch (i)
  } // for (i ...

  Simulation(simTime, meanInterval, meanServiceTime, verbose);
  PrintStatistics(simTime);

  printf("Type any key to exit.\n");
  getchar();

  return 0;
}
```

(a) Assume the executable program is built, and is named `a.out`. The program is executed with this Linux command line:

```
./a.out 100 1
```

What is the value of each argument when the call to the function `Simulation` is made?

```
simTime =

meanInterval =

meanServiceTime =

verbose =
```

(b) Assume the executable program is built, and is named `a.out`. Provide a Linux command that will cause the main program to display the message:

```
Usage:  PA5 simTime interval serviceTime seed verbose
Using defaults
```

7. (15 points) Consider the following C code segment:

```
int i, j;    // loop control variables
int sum=0;
int table[10][20];

// compute the sum of all elements of table
for (j=0;  j<20; j++)
  for (i=0; i<10; i++)
     sum = sum + table[i][j];
```

(a) (10 points) Explain what this statement means:

The given code exhibits *temporal locality* with respect to the variable `sum`.

(b) (5 points) Assume this code is executed on a machine that utilizes cache memory. Would you expect the given code to execute more efficiently if arrays are stored in main memory in row-major order, or if arrays are stored in main memory in column-major order? Defend your answer.

8. (15 points) Assume that `a`, `b`, and `c` are 16-bit unsigned integers. `a` contains the value 65280 (binary value 1111 1111 0000 0000) and `b` contains the value 61680 (binary value 1111 0000 1111 0000). What will `c` contain after each of the following sets of statements executes? Give your answers in binary.

(a)
```
a = 65280;
b = 61680;
c = a | b;
```
c = _____

(b)
```
a = 65280;
b = 61680;
c = b >> 8;
```
c = _____

(c)
```
a = 65280;
b = 61680;
c = (~b) & a;
```
c = _____

(exam continues on next page)

9. (10 points)

   (a) In a single sentence, describe the purpose of a hash function.

   (b) In a single sentence, define the term *collision* as it pertains to hash tables.