NAME: SECTION:

USERNAME:

CS 2301
Exam 3
B-Term 2005

Questions 1 - 4:	_____	(20)
Question 5:	_____	(15)
Question 6:	_____	(15)
Question 7:	_____	(40)
Question 8:	_____	(10)
TOTAL:	_____	(100)

Fill in your name, section, and username. DO NOT OPEN THIS EXAM UNTIL YOU ARE TOLD TO DO SO.

The first four questions are multiple choice. Circle the response that best answers each question. (5 points each)

1. Assume that a linked list is made up of nodes defined as follows:

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

   Suppose `cursor` points to a node in a linked list. What expression will be true when `cursor` points to the tail node of the list?

   (a) `(cursor == NULL)`

   (b) `(cursor->nextPtr == NULL)`

   (c) `(cursor->data == NULL)`

   (d) `(cursor.nextPtr == NULL)`

   (e) None of the above.

2. Consider the following pseudocode:

```
declare a stack of characters
while ( there are more characters in the word to read )
{
   read a character
   push the character on the stack
}
while ( the stack is not empty )
{
   pop a character off the stack
   write the popped character to the screen
}
```

   What is written to the screen for the input "carpets"?

   (a) serc

   (b) carpets

   (c) steprac

   (d) ccaarrppeettss

3. A queue is implemented with a linked list, keeping track of a head pointer and a tail pointer. Which of these pointers will change during an insertion into an empty queue?

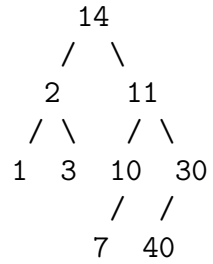
   (a) Neither changes

   (b) Only the head pointer changes.

   (c) Only the tail pointer changes.

   (d) Both change.

4. Consider this binary tree:



   What is the order of nodes visited using an in-order traversal of the binary tree?

   (a) 1 2 3 7 10 11 14 30 40

   (b) 1 2 3 14 7 10 11 40 30

   (c) 1 3 2 7 10 40 30 11 14

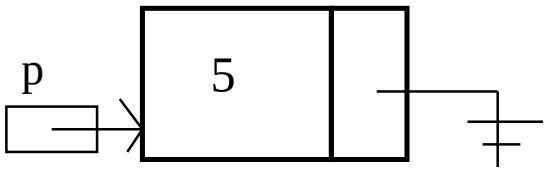
   (d) 14 2 1 3 11 10 7 30 40

5. (15 points) In each part below, you are given a segment of C code. Draw a picture that shows the final result of the execution of the code segment. Your picture should indicate the value of every declared variable and the value of every field in every node. Assume you are using the `struct listNode` definition given in Problem 1.

   Example: If the code segment is

```
struct listNode *p;
p = malloc(sizeof(struct listNode));
p->data = 5;
p->nextPtr = NULL;
```

   the picture should look like this:



(a)
```
struct listNode *p, *q;
p = malloc(sizeof(struct listNode));
p->data = 5;
p->nextPtr = NULL;
q = malloc(sizeof(struct listNode));
q->data = 10;
q->nextPtr = p;
p->nextPtr = p;
```

(b)
```
struct listNode *p, *q, *r;
p = malloc(sizeof(struct listNode));
p->data = 5;
p->nextPtr = NULL;
q = malloc(sizeof(struct listNode));
q->data = 10;
q->nextPtr = NULL;
p->nextPtr =q;
r = p->nextPtr;
free (p);
p = r;
```

(c)

```
struct listNode *p;
struct listNode *q=NULL;
p = malloc(sizeof(struct listNode));
p->data = 5;
p->nextPtr = q;
q = malloc(sizeof(struct listNode));
q->data = 10;
q->nextPtr = p->nextPtr;
```

6. (10 points) Suppose that `p` is a pointer to a node in a linked list, and `*p` is not the tail node. What steps must be taken to remove the node after `*p`? Use one short English sentence for each step.

7. (40 points) Here is the heading and pre- and post-conditions for a linked list function (using the `struct listNode` definition as given in Problem 1):

```
int count42s(struct listNode *head)
/* Precondition: head is the head pointer of a linked list.
** The linked list might be empty or it might be non-empty.
** Postcondition: The return value is the number of occurrences
** of the value 42 in the data field of a node on the linked list.
** The list itself is unchanged. */
```

(a) Implement the above function *non-recursively*.

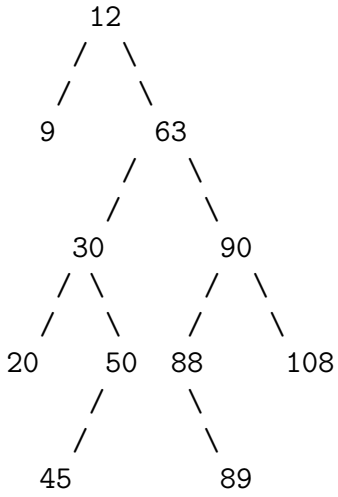
(b) Implement the above function *recursively*.

(c) Explain why the function does not need to use a double pointer for the parameter `head`.

8. (15 points)

(a) Draw a picture of the binary search tree that would be created if the following values were inserted into the search tree in this order:
52, 97, 112, 65, 77, 49, 86, 80

(b) Here is a binary search tree:



Draw a picture of a binary search tree that will result if the value 63 is deleted from the given tree.