You have 50 minutes. You may use one prepared 8.5 × 11 sheet of paper. All questions concern the C++ programming language. **Check your answers.**

For the multiple choice questions (marked [MC]) circle the letter of the best answer, even if no answer is ideal. For the other questions, write the answer after the question in a clear and readable way.

The first two questions use the following structure type definition.

```cpp
struct Feline
{
    int     size;
    const char* coat;
    Feline(int s, const char* c)
    {
        size = s;
        coat = c;
    }
};
```

1. [MC] Suppose that you have already created a variable called cat of type Feline. Which of the following statements will set the 'size' field of cat to hold 10?
   
   (a) s.Feline = 10;
   (b) cat.s = 10;
   (c) cat.size = 10;
   (d) Feline.size = 10;
   (e) size.cat = 10;

2. [MC] Which of the following statements or sequences of statements will create a variable p of type Feline* and make p point to a new Feline structure whose 'size' field holds 8 and whose 'coat’ field holds “tabby”?

   (a) new Feline* p(8, "tabby");
   (b) Feline* p = new Feline*; p->size = 8; p->coat = "tabby";
   (c) Feline* p = new Feline(8, "tabby");
   (d) Feline* p(8, "tabby");
   (e) Feline* p = new Feline*; size.p = 8; coat.p = "tabby";
Types ListCell and List are as defined in class. Here are their definitions.

```c
struct ListCell
{
    int head;
    ListCell* tail;

    ListCell(int h, ListCell* t)
    {
        head = h;
        tail = t;
    }
};
```

**typedef ListCell* List;**

You can assume that constant emptyList and functions isEmpty(L), head(L), tail(L) and cons(x, L) have been defined as in class.

3. [MC] Which of the following will create variable L, of type List, and make it point to a new linked list holding 4 and 7, in that order? (Using our conceptual list notation, it must make L refer to list [4, 7].)

   (a) List L = new ListCell(4, new ListCell(7, NULL));
   (b) List L = new ListCell(7, new ListCell(4, NULL));
   (c) List L = new List(4, new List(7, NULL));
   (d) List L = new ListCell(4, 7);
   (e) List L = new ListCell(7, 5);

4. Suppose that variables L and n have already been created. L has type ListCell* and points to a linked list of length three, and n has type int. Which of the following sets variable n to the second integer in list L?

   (a) n = L->head->tail
   (b) n = L->tail->tail;
   (c) n = L->tail->head;
   (d) n = L->2;
   (e) n = L[1];
5. Write a C++ definition of function twice(s) that takes a null-terminated string s and yields a null-terminated string that holds string s twice in a row. For example, twice("fig") must return "figfig" and twice("rabbit") must return "rabbitrabbit". You may use functions from the cstring library. A heading is given.

    char* twice(const char* s)
6. Suppose that \( \text{sum}(L) \) is intended to return the sum of the values in list \( L \). For example, \( \text{sum}([8, 2, 5]) = 8 + 2 + 5 = 15 \) and \( \text{sum}([9, 7]) = 9 + 7 = 16 \).

(a) Using the conceptual view of lists discussed in class, complete the following equations so that, taken together, they define \( \text{sum}(L) \) for list \( L \). Use conceptual notation, not C++ notation, for this part. See the bottom of this page for a brief summary of conceptual list notation.

\[
\begin{align*}
\text{sum}([]) &= \underline{\quad} \\
\text{sum}(L) &= \underline{\quad} \\
&\quad \text{ (when } L \neq []) \\
\end{align*}
\]

(b) Following your equations from part (a) closely, write a C++ definition of \( \text{sum}(L) \). It must not change any of the cells in list \( L \). **Do not use any kind of loop for this definition.** A heading is given.

\[
\text{int sum(List L)}
\]

Summary of conceptual list notation.

- \([\quad] \) is an empty list
- isEmpty([\quad]) is true
- head([2, 4, 6, 8]) = 2
- tail([2, 4, 6, 8]) = [4, 6, 8]
- 2 : [4, 6, 8] = [2, 4, 6, 8]