CS-230 Data Structures
Test 2
Part III (Programming Question)
3 points

Important Instructions:

1. Answer either Question 1 or Question 2 to fulfill the requirements of this exam.

2. Answer both questions for a bonus of 2 points.

3. It is your responsibility to identify which question should be graded for the fulfillment of the requirements of this exam and which question should be graded as the bonus.
   - Identify the question that you choose to be graded for the fulfillment of the requirements of this exam and submit the zip file containing all files in the solution to that problem in the drop box labeled as **Part III** on Angel.
   - Identify the question that you choose to be graded for the bonus points and submit the zip file containing all files in the solution to that problem in the drop box labeled as **Bonus** on Angel.

4. It is your responsibility to submit the correct zip files containing ALL components. If for whatever reason I cannot open the form file or code file within a project, your score on that project will be zero (0). Please ask and seek help if you are not sure how to zip a file.

1. **Splitting a linked list, at a given node, into two sublists.**

   Add the following operation to the class `LinkedList`:

```
void divideAt( LinkedList<NodeType> &secondList, const NodeType & item)
   // Divide the list at the node with the data item into two sublists.
   // Postcondition: firstNode and lastNode point to the first and last
   //          nodes of the first sublist.
   //          secondList.firstNode and secondList.lastNode point to
   //                the first and last nodes of the second sublist.
```

   Consider the following statements:

```
LinkedList<int> myList;
LinkedList<int> otherList;
```

   Suppose `myList` points to the list with the following elements (in this order):

```
34 65 18 39 27 89 12
```

   The statement

```
myList.divideAt( otherList, 18);
```

   divides `myList` into two sublists: `myList` points to the list with the elements

```
34 65,
```

   and `otherList` points to the sublist with the elements

```
18 39 27 89 12.
```

- Write the definition of the function template to implement the operation `divideAt`. Also write a program to test your function.
- Submit a zip file containing all components of the new class `LinkedList`.

2. Write a recursive function to generate the following pattern of stars:

```
    *
   * *
  * * *
 * * * *
  * * *
   * *
    *
```

Also, write a program that prompts the user to enter the number of stars in the width of the pattern and uses the recursive function to generate the pattern. For example, specifying 4 as the width generates the preceding pattern. Submit a zip file containing all components of the application.