

SYSC 2100, Fall 2006 Midterm
October 31, 2006
Duration: 80 minutes
Instructor: T. Kunz

Name:

Student number:

Question 1. Recursion (10 marks)

1. Write a recursive Java method *writeLine* that writes a character repeatedly to form a line of n characters. For example, *writeLine*('*', 5) produces the line

2. Write a recursive method *writeBlock* that uses *writeLine* to write m lines of n characters each. For example, *writeBlock*('*', 5, 3) produces the output

Question 2. ADT List (10 marks)

1. In the discussion in class, we described methods *displayList* and *replace* for the ADT List. As described, these methods exist outside of the ADT; that is, they are not operations of the ADT. Instead, their implementations are written in terms of the ADT's operations.
 - a. What is an advantage and a disadvantage of the way *displayList* and *replace* are implemented?
 - b. What is an advantage and a disadvantage of adding the operations *displayList* and *replace* to the ADT?

2. Write a method to merge two linked lists of integers that are sorted into ascending order. The result should be a third linked list that is the sorted combination of the original lists. Do not destroy the original lists.

Question 3. ADT Stack (10 marks)

1. Write a pseudo-code method $isInL(s)$ that uses a stack to determine whether a string s is in the language L , where $L = \{w: w \text{ is of the form } A^n B^n \text{ for some } n \geq 0\}$
2. Suppose that you have a stack $aStack$ and an empty auxiliary stack $auxStack$. Show how you can do each of the following tasks using only the operations of the ADT stack (i.e., write the pseudo-code for the following tasks):
 - a. Count the number of items in $aStack$, leaving $aStack$ unchanged.
 - b. Delete every occurrence of a specified item from $aStack$, leaving the order of the remaining items unchanged.

Question 4. Algorithm Efficiency and Sorting (10 marks)

1. Suppose that your implementation of a particular algorithm appears in Java as

```
for (int pass = 1; pass <= n; ++pass) {  
    for (int index = 0; index < n; ++index) {  
        for (int count = 1; count < 10; ++count) {  
            ...  
        } // end for  
    } // end for  
} // end for
```

The previous code shows only the repetition in the algorithm, not the computations that occur within the loops. These computations, however, are independent of n . What is the order of the algorithm? Justify your answer.

2. Apply the selection sort, bubble sort, and insertion sort to
- An inverted array: 8 6 4 2
 - An ordered array: 2 4 6 8

Show the resulting array after each change.