BLG 233E, Data Structures and Laboratory, Fall 2012-2013
Assignment #1
Asst. Prof. Dr. Gülşen Cebiroğlu Eryiğit
Res. Asst. Doğan Altan

Due: November 2, 2012, 23.59 PM

Submission type: An archive file including all source code files.

In this homework, you will use linked lists to represent matrices. On these matrices, you will apply some operations such as multiplication, transpose, checking the symmetry property and checking the zero matrix property.
Each element of the matrix will be represented as a node in the linked list. Structure of a node is illustrated in Figure 1.

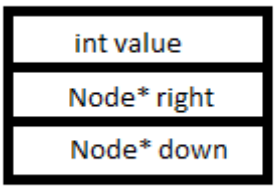

Figure 1: Structure of the Node

Each node will have an integer value and two pointers (right and down) for right and down elements of the matrix. When the program starts, it should ask the user the dimension of the first matrix. This matrix's elements should be assigned randomly between 0 and 9. Then the program should read second matrix's elements from "matrix.txt" file which is provided with the homework. In this file, first row represents the dimension of the matrix. After this, a menu should be listed. In this list, there must be 5 options.

1. Multiply the matrices and print the result to the screen. Give an error message to the user if these two matrices cannot be multiplied.
2. Find transpose of the matrices (both) and print them to the screen.
3. Check if the matrices are symmetric or not and give messages to the user about the result.
4. Check if the matrices are zero matrix or not and give messages to the user about the result.
5. Calculate the determinant of the matrices if they are 3x3 and print the result to the screen. If a matrix is not 3x3, you should print an error message to the screen.

After each operation, you should ask if the user wants to terminate the program. If user enters ‘y’, the menu should be listed one more time. Otherwise (if the user enters ‘n’), the program should be terminated and before termination, you should deallocate your linked list.

Your program will be run and graded on programming style and how it performs the required computation.

With regard to programming style, we expect the following:

(I) A top-down design should be evident.

(II) Comments and/or appropriate variable names should be used to make your program readable.

(III) Appropriate prompts and messages for input and output should be given to the user of the program.

Unless stated otherwise, more emphasis will be placed on program clarity than on program length, speed or size.

CHEATING: This is not a group assignment. It should be done individually. When a student receives information from another person about a program, it is considered cheating when the information is enough to precisely describe the code in a nontrivial part of the program. For the most part, oral discussions between students about the algorithms used in a program, or the program's design, is not considered cheating. In fact, these types of discussions are encouraged. The most common example of cheating occurs when a student copies all or part of a program from another student and then changes the names of some of the program variables and functions. If cheating is discovered, a report will be made recommending a course grade of "VF".