BLG 233E

DATA STRUCTURES AND LABORATORY

CRN: 11146

REPORT OF HOMEWORK 1

Submission Date: 02.11.12

STUDENT NAME: TUĞRUL YATAĞAN

STUDENT NUMBER: 040100117

1. Introduction

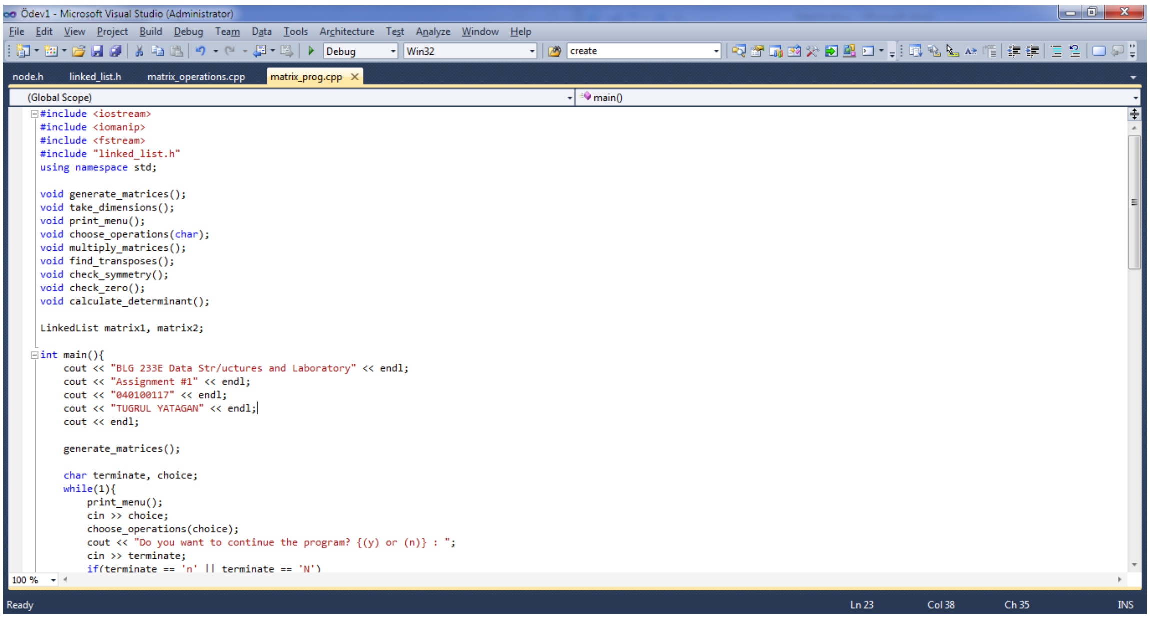
In this homework, the aim was create linked lists to represent matrices. On these matrices, the program applies some operations such as multiplication, transpose, checking the symmetry property and checking the zero matrix property. Each element of the matrix represented as a node in the linked list.

Each node have an integer value and two pointers (right and down) for right and down elements of the matrix. When the program starts, it asks the user the dimension of the first matrix. This matrix’s elements assigned randomly between 0 and 9. Then the program 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 listed. In this list, there are 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.

2. Development and Operating Environments

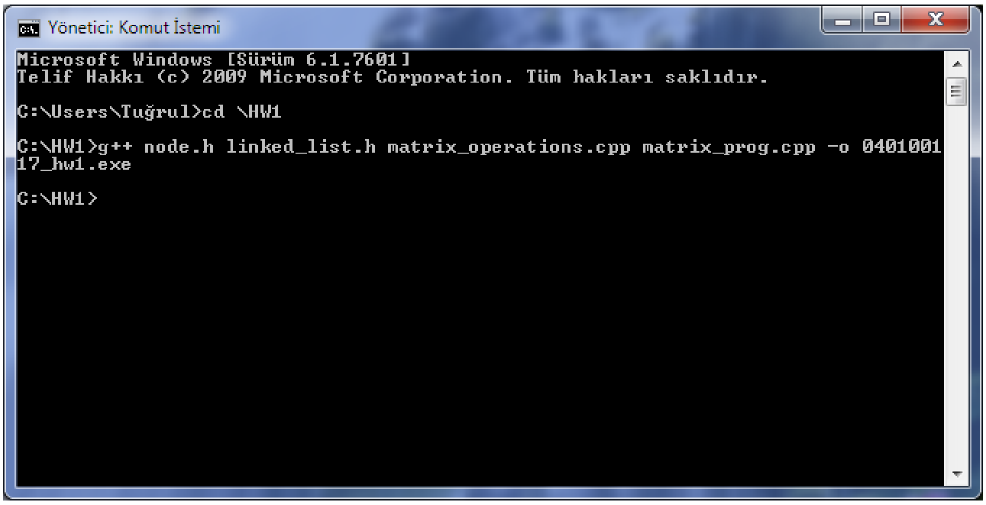
Microsoft Visual C++ 2010 environment has been used to write the source code in Windows 7 operation system and GNU C++ compiler was used to compile the program.


Ödev1 - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Win32
create
node.h
linked_list.h
matrix_operations.cpp
matrix_prog.cpp
(Global Scope)
main()
100 %
Ready
Ln 23
Col 38
Ch 35
INS

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include "linked_list.h"
using namespace std;

void generate_matrices();
void take_dimensions();
void print_menu();
void choose_operations(char);
void multiply_matrices();
void find_transposes();
void check_symmetry();
void check_zero();
void calculate_determinant();

LinkedList matrix1, matrix2;

int main(){
    cout << "BLG 233E Data Str/uctures and Laboratory" << endl;
    cout << "Assignment #1" << endl;
    cout << "040100117" << endl;
    cout << "TUGRUL YATAGAN" << endl;
    cout << endl;

    generate_matrices();

    char terminate, choice;
    while(1){
        print_menu();
        cin >> choice;
        choose_operations(choice);
        cout << "Do you want to continue the program? {(y) or (n)} : ";
        cin >> terminate;
        if(terminate == 'n' || terminate == 'N')
```

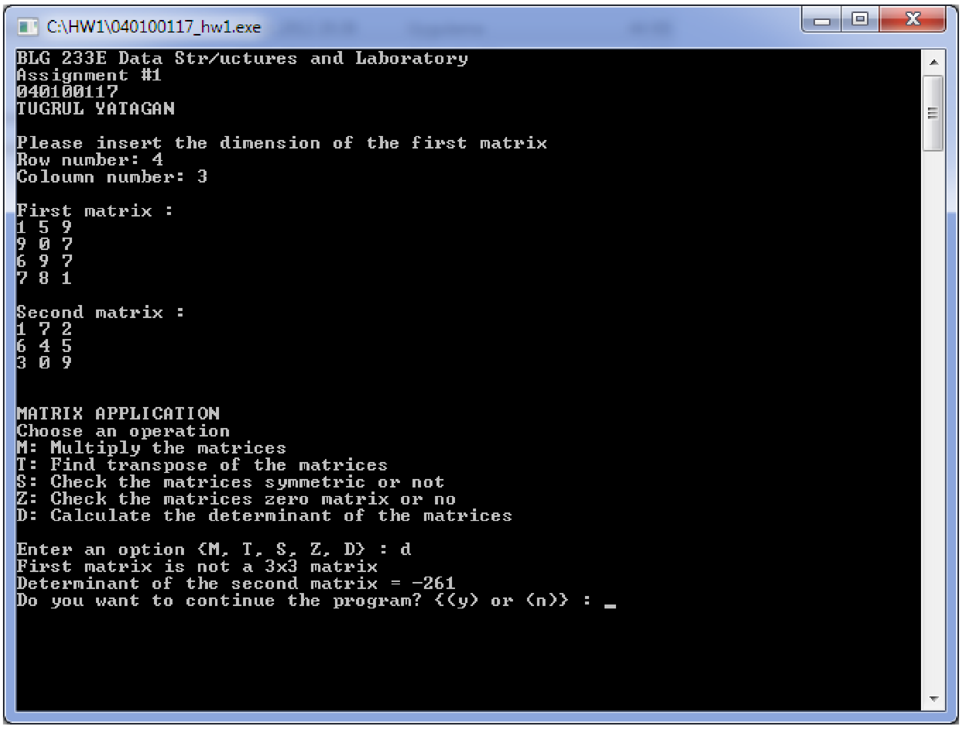
The program consist of 2 source codes which are named “matrix_operations.cpp” and “matrix_prog.cpp”, and it also consist of 2 header files which are named “node.h” and “linked_list.h”. matrix_operations.cpp consists linked list operation function definitions, matrix_prog.cpp consist user input-output operation functions, node.h consist the data structure of Node and linked_list.h consist the data structure of LinkedList.

To use GNU C++ as our compiler first we need to install MinGW and add compiler to Windows PATH, then we need to go Start>Run and write cmd, click OK. In the DOS command prompt, cd to the directory where the program is located. To compile, matrix_operations.cpp matrix_prog.cpp node.h linked_list.h -o 040100117_hw1.exe needs to be written, whereas linking the program is via 040100117_hw1.exe. The program compiled without warning or error. Compiling process is below:



```
Microsoft Windows [Sürüm 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\Tuğrul>cd \HW1

C:\HW1>g++ node.h linked_list.h matrix_operations.cpp matrix_prog.cpp -o 0401001
17_hw1.exe

C:\HW1>
```

And finally the program is executed. Outcome is below:



```
BLG 233E Data Str/uctures and Laboratory
Assignment #1
040100117
TUGRUL YATAGAN

Please insert the dimension of the first matrix
Row number: 4
Coloumn number: 3

First matrix :
1 5 9
9 0 7
6 9 7
7 8 1

Second matrix :
1 7 2
6 4 5
3 0 9


MATRIX APPLICATION
Choose an operation
M: Multiply the matrices
T: Find transpose of the matrices
S: Check the matrices symmetric or not
Z: Check the matrices zero matrix or no
D: Calculate the determinant of the matrices

Enter an option (M, T, S, Z, D) : d
First matrix is not a 3x3 matrix
Determinant of the second matrix = -261
Do you want to continue the program? ((y) or (n)) : _
```

3. Data Structures and Variables

As you see, 2 types of data structures were used in this homework, Node and LinkedList

```
struct Node{
	int value;
	Node *right;
	Node *down;
};
```

- value: value of matrix element
- right: node type pointer which points right after elements of matrix
- down: node type pointer which points down after elements of matrix

```
struct LinkedList{
	Node *head;
	int row, col;
	void create();
	void generaterand();
	void readfile();
	int scanright(int);
	int scandown(int);
	void print();
	void transpose();
	bool symmetry();
	bool zero();
	int determinant();
	void clear();
};
```

- head: node type pointer which points head of the linked list
- row, col: row and column number of matrix of linked list
- create(): creates empty linked list
- generaterand(): generates linked list matrix which all elements are random number
- readfile(): generates linked list matrix from matrix.txt input
- scanright(): reads nodes value left to right from up to down in a matrix
- scandown():reads nodes value up to down from left to right in a matrix
- print(): prints matrices to the screen
- transpose(): takes transposes of matrices
- symmetry(): checks if matrices symmetric matrix or not
- zero(): checks if matrices zero matrix or not
- determinant(): calculates determinant of matrices
- clear(): gives back allocated linked list to memory

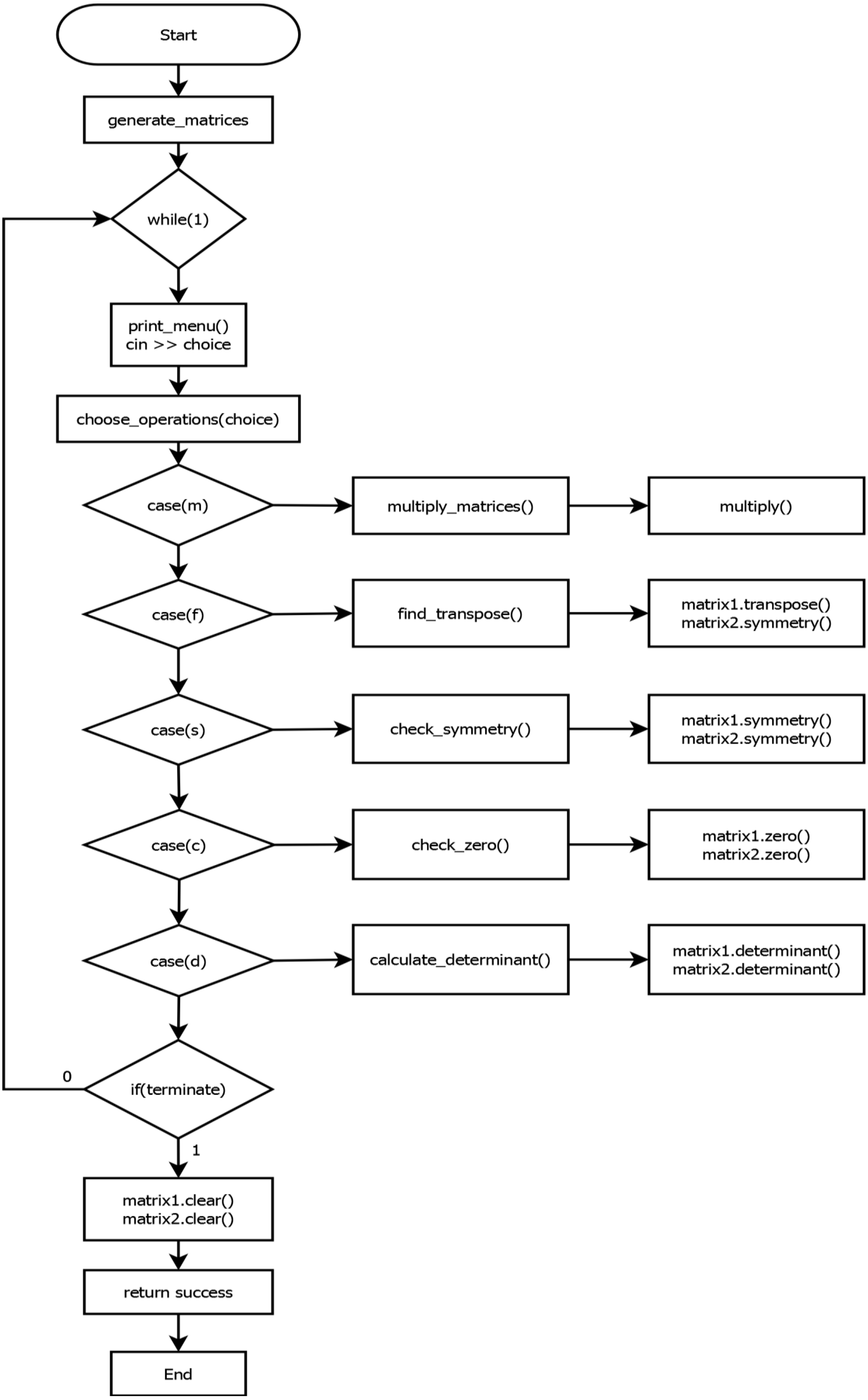
```
LinkedList matrix1, matrix2;
```

matrix1 and matrix2 are our objects which are type of LinkedList

```
extern LinkedList matrix1, matrix2;
```

matrix1 and matrix2 objects are used in matrix_operations.cpp with extern

4. Program Flow



5. Conclusion

In this homework, I have become more familiar with the concept of data structures, structs and linked list. I had the chance to intensify my knowledge about their structures.