**BLG 233E**

**DATA STRUCTURES AND LABORATORY**

CRN: 11146

**REPORT OF HOMEWORK #2**

Submission Date: 30.11.12

**STUDENT NAME: TUĞRUL YATAĞAN**

**STUDENT NUMBER: 040100117**

**1. Introduction**

In this homework, program will implement a solution that uses stacks to sort student records according to GPAs in a university. An input file will be given which includes the necessary information for your struct (number, name, GPA, and department).

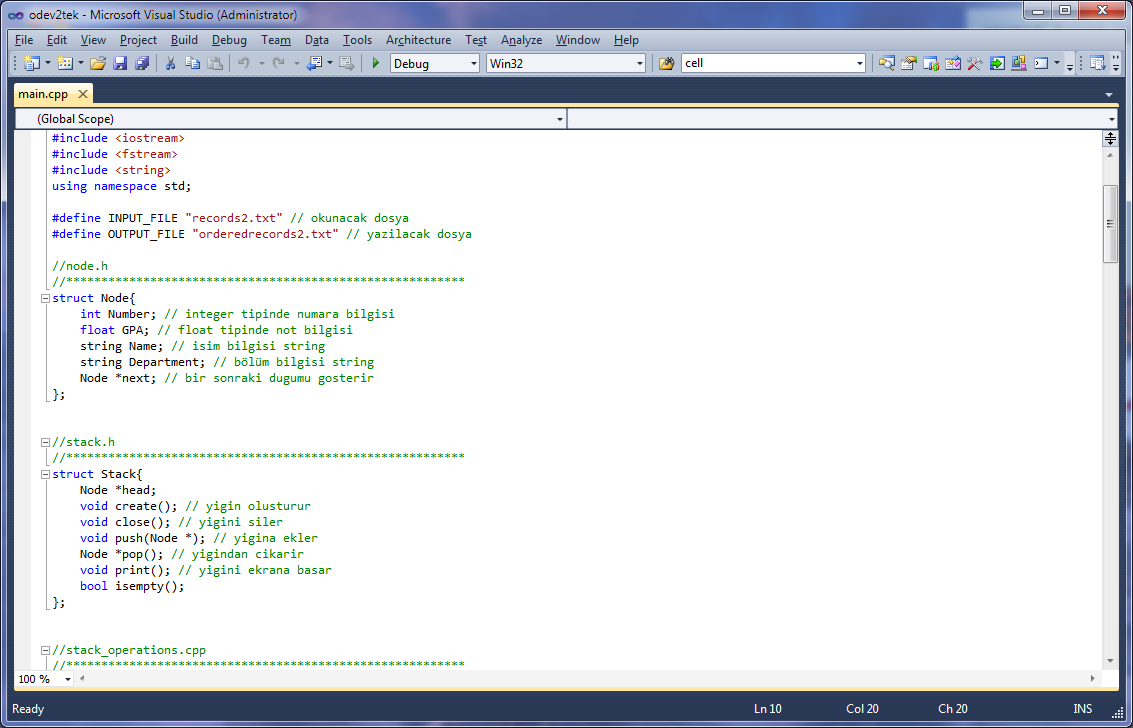
The program uses two stacks:

1) The program adds records in a sorted fashion to the first stack. The student record with the lowest GPA should be at the bottom of this stack, and the record with the highest GPA should be at the top.

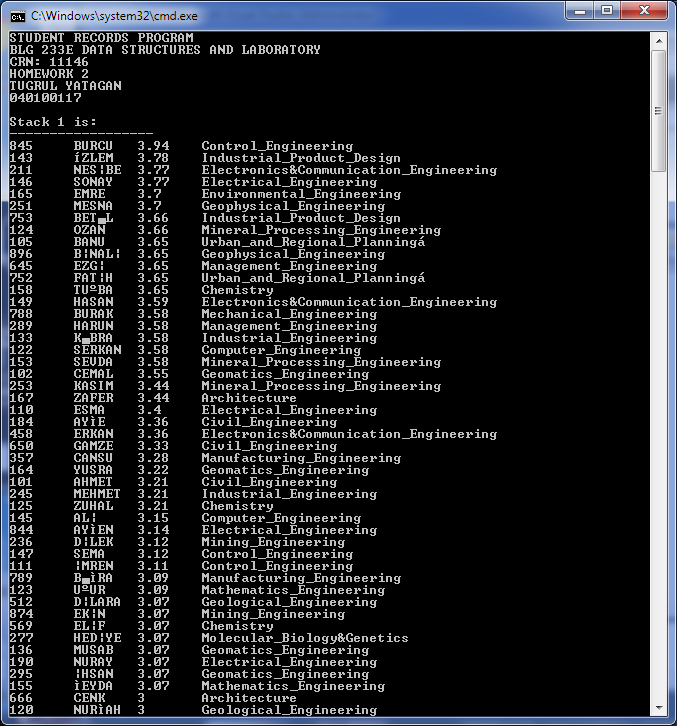
2) The program uses the second stack for temporarily storing records while adding a new record to the first stack. When adding to the first stack, you should check the top record and pop the records from the first stack until the GPA of the top record is lower than that of the record to be added. You should push these popped records onto the second stack, and after pushing the record to the first stack, you should move the contents of the second stack to the first stack.

**2. Development and Operating Environments**

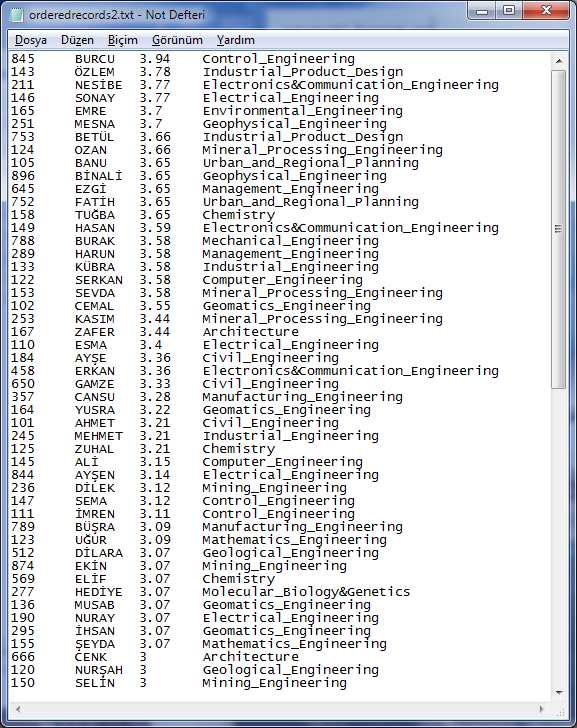
Microsoft Visual C++ 2010 environment has been used to write the source code in Windows 7 operation system and Microsoft Visual C++ 2010 compiler was used to compile the program.



The program compiled without warning or error. And finally the program is executed. Sample outcome is below:



Sample output file is below:



**3. Data Structures and Variables**

The program was design for **“records2.txt”** input file.Initially the program gives output as name “orderedrecords2.txt” which can be changeable.

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

struct Node{

int Number;

float GPA;

string Name;

string Department;

Node \*next;

};

Node structure is used for store student records information and necessary pointer for linked list stack.

struct Stack{

Node \*head;

void create();

void close();

void push(Node \*);

Node \*pop();

void print();

bool isempty();

};

Stack structure is used for linked list stack operations. In this scope create() function creates stack, close() functions deletes stack(), push(Node\*) functions adds nodes to the head of stack it takes pointer of new nodes, \*pop() functions deletes first node of stack and return it, print() function prints the stack to user, isempty() function checks if stack is empty or not.

void readfile();

void writefile();

readfile() functions reads the input file and add records to stack in ordered list notation, writefile() functions writes ordered stack to the text file.

**4. Conclusion**

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