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).

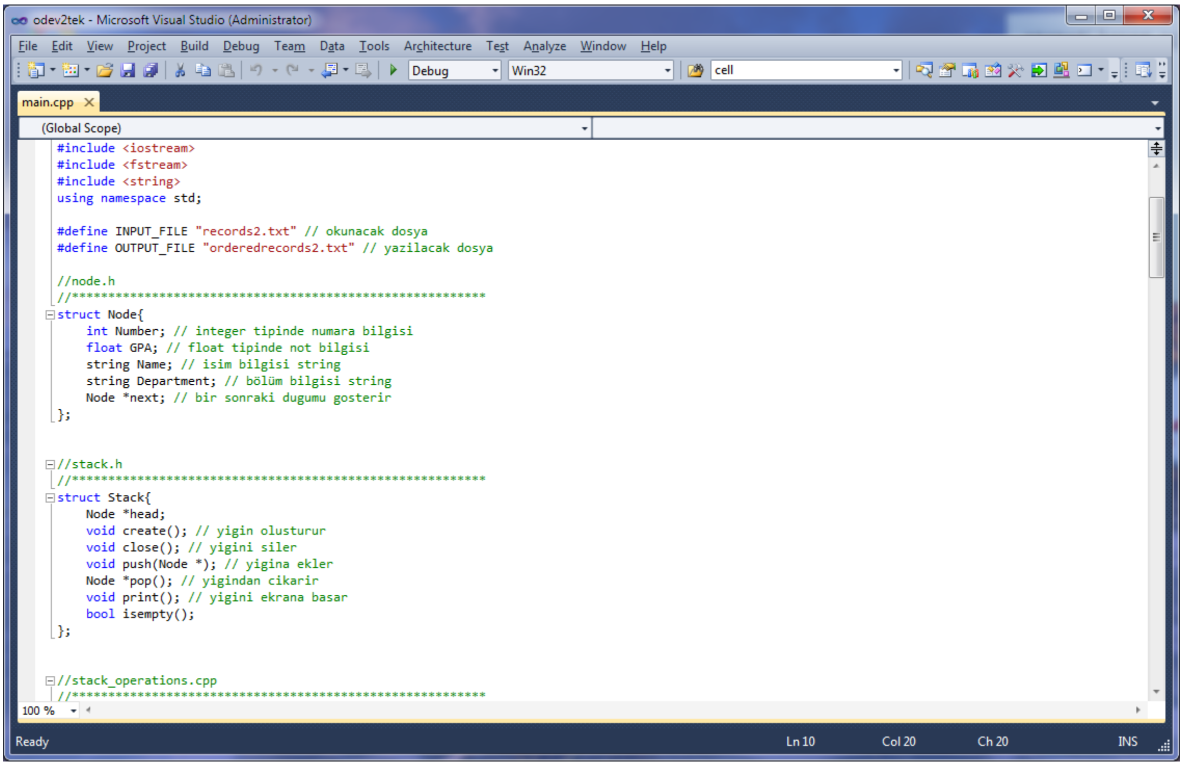
The program uses two stacks:

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.

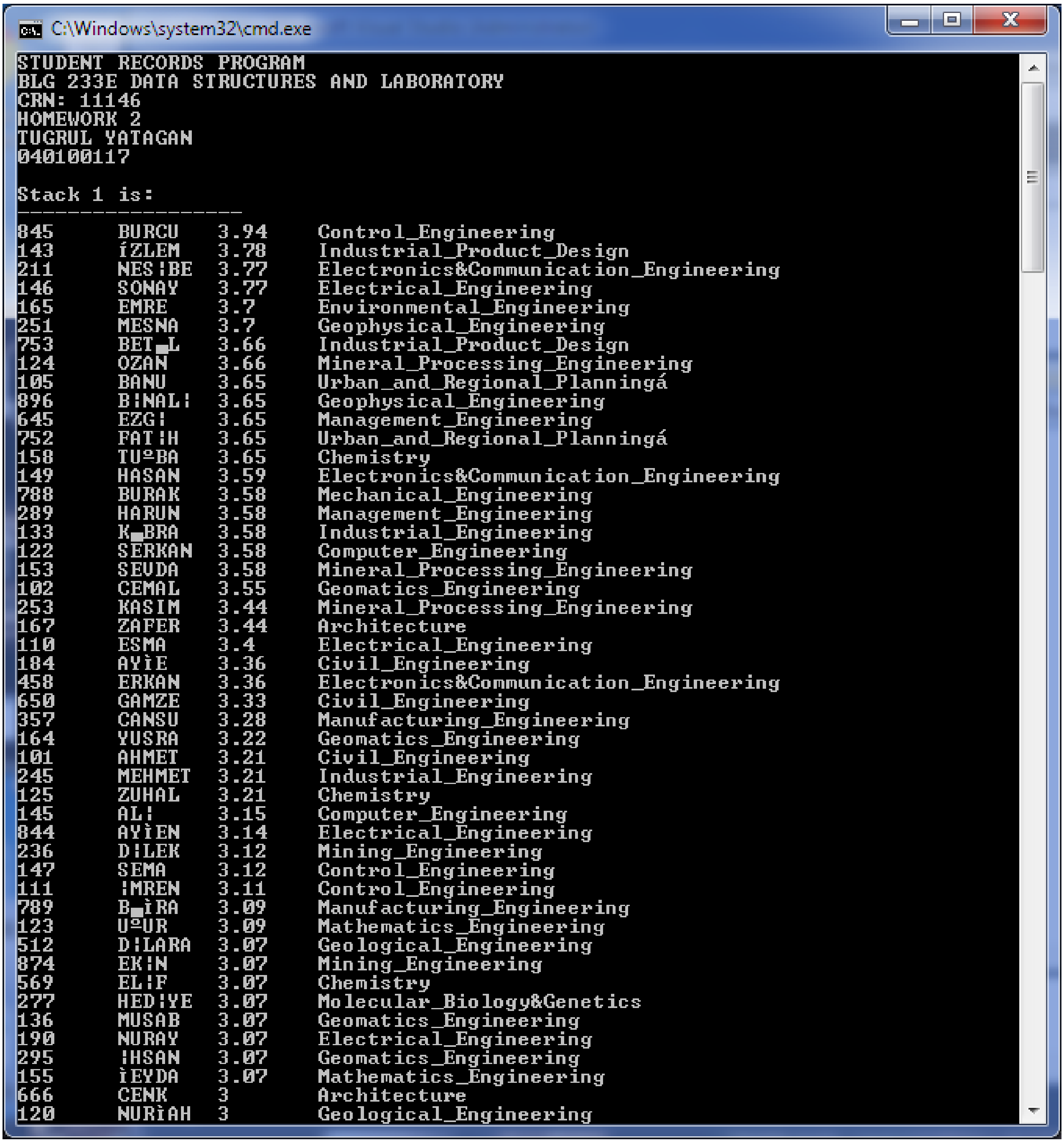
2. Development and Operating Environments

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.


odev2tek - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help
Debug
Win32
cell
main.cpp
(Global Scope)
100 %
Ready
Ln 10
Col 20
Ch 20
INS

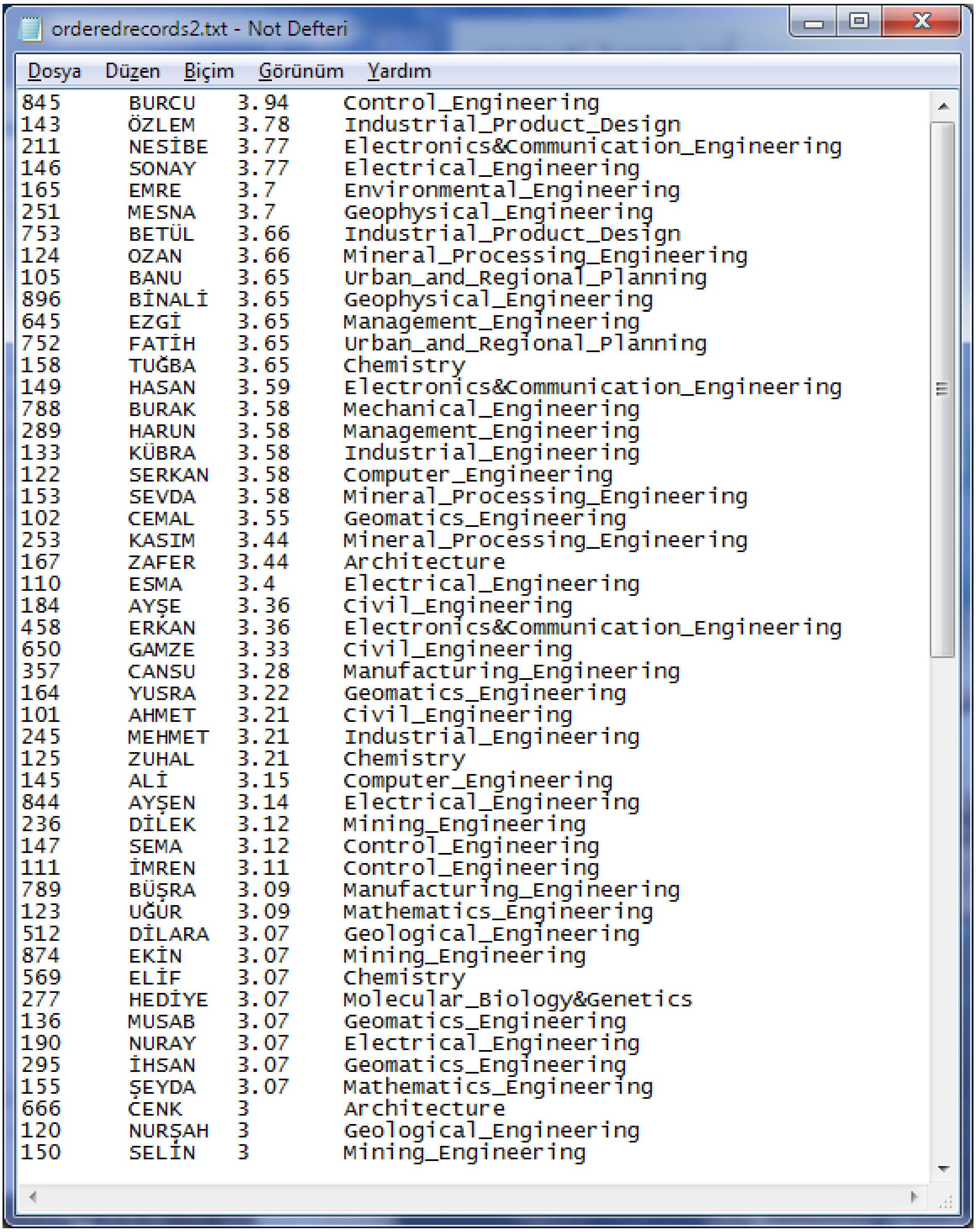
```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

#define INPUT_FILE "records2.txt" // okunacak dosya
#define OUTPUT_FILE "orderedrecords2.txt" // yazilacak dosya

//node.h
//*********************************************************
struct Node{
    int Number; // integer tipinde numara bilgisi
    float GPA; // float tipinde not bilgisi
    string Name; // isim bilgisi string
    string Department; // bölüm bilgisi string
    Node *next; // bir sonraki dugumu gosterir
};


//stack.h
//*********************************************************
struct Stack{
    Node *head;
    void create(); // yigin olusturur
    void close(); // yigini siler
    void push(Node *); // yigina ekler
    Node *pop(); // yigindan cikarir
    void print(); // yigini ekrana basar
    bool isempty();
};


//stack_operations.cpp
//*********************************************************
```

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



```
STUDENT RECORDS PROGRAM
BLG 233E DATA STRUCTURES AND LABORATORY
CRN: 11146
HOMEWORK 2
TUGRUL YATAGAN
040100117

Stack 1 is:
------------------
845     BURCU   3.94    Control_Engineering
143     ÍZLEM   3.78    Industrial_Product_Design
211     NES¦BE  3.77    Electronics&Communication_Engineering
146     SONAY   3.77    Electrical_Engineering
165     EMRE    3.7     Environmental_Engineering
251     MESNA   3.7     Geophysical_Engineering
753     BET▄L   3.66    Industrial_Product_Design
124     OZAN    3.66    Mineral_Processing_Engineering
105     BANU    3.65    Urban_and_Regional_Planningá
896     B¦NAL¦  3.65    Geophysical_Engineering
645     EZG¦    3.65    Management_Engineering
752     FAT¦H   3.65    Urban_and_Regional_Planningá
158     TUºBA   3.65    Chemistry
149     HASAN   3.59    Electronics&Communication_Engineering
788     BURAK   3.58    Mechanical_Engineering
289     HARUN   3.58    Management_Engineering
133     K▄BRA   3.58    Industrial_Engineering
122     SERKAN  3.58    Computer_Engineering
153     SEVDA   3.58    Mineral_Processing_Engineering
102     CEMAL   3.55    Geomatics_Engineering
253     KASIM   3.44    Mineral_Processing_Engineering
167     ZAFER   3.44    Architecture
110     ESMA    3.4     Electrical_Engineering
184     AYÌE    3.36    Civil_Engineering
458     ERKAN   3.36    Electronics&Communication_Engineering
650     GAMZE   3.33    Civil_Engineering
357     CANSU   3.28    Manufacturing_Engineering
164     YUSRA   3.22    Geomatics_Engineering
101     AHMET   3.21    Civil_Engineering
245     MEHMET  3.21    Industrial_Engineering
125     ZUHAL   3.21    Chemistry
145     AL¦     3.15    Computer_Engineering
844     AYÌEN   3.14    Electrical_Engineering
236     D¦LEK   3.12    Mining_Engineering
147     SEMA    3.12    Control_Engineering
111     ¦MREN   3.11    Control_Engineering
789     B▄ÌRA   3.09    Manufacturing_Engineering
123     UºUR    3.09    Mathematics_Engineering
512     D¦LARA  3.07    Geological_Engineering
874     EK¦N    3.07    Mining_Engineering
569     EL¦F    3.07    Chemistry
277     HED¦YE  3.07    Molecular_Biology&Genetics
136     MUSAB   3.07    Geomatics_Engineering
190     NURAY   3.07    Electrical_Engineering
295     ¦HSAN   3.07    Geomatics_Engineering
155     ÌEYDA   3.07    Mathematics_Engineering
666     CENK    3       Architecture
120     NURÌAH  3       Geological_Engineering
```

Sample output file is below:



```
845     BURCU   3.94    Control_Engineering
143     ÖZLEM   3.78    Industrial_Product_Design
211     NESİBE  3.77    Electronics&Communication_Engineering
146     SONAY   3.77    Electrical_Engineering
165     EMRE    3.7     Environmental_Engineering
251     MESNA   3.7     Geophysical_Engineering
753     BETÜL   3.66    Industrial_Product_Design
124     OZAN    3.66    Mineral_Processing_Engineering
105     BANU    3.65    Urban_and_Regional_Planning
896     BİNALİ  3.65    Geophysical_Engineering
645     EZGİ    3.65    Management_Engineering
752     FATİH   3.65    Urban_and_Regional_Planning
158     TUĞBA   3.65    Chemistry
149     HASAN   3.59    Electronics&Communication_Engineering
788     BURAK   3.58    Mechanical_Engineering
289     HARUN   3.58    Management_Engineering
133     KÜBRA   3.58    Industrial_Engineering
122     SERKAN  3.58    Computer_Engineering
153     SEVDA   3.58    Mineral_Processing_Engineering
102     CEMAL   3.55    Geomatics_Engineering
253     KASIM   3.44    Mineral_Processing_Engineering
167     ZAFER   3.44    Architecture
110     ESMA    3.4     Electrical_Engineering
184     AYŞE    3.36    Civil_Engineering
458     ERKAN   3.36    Electronics&Communication_Engineering
650     GAMZE   3.33    Civil_Engineering
357     CANSU   3.28    Manufacturing_Engineering
164     YUSRA   3.22    Geomatics_Engineering
101     AHMET   3.21    Civil_Engineering
245     MEHMET  3.21    Industrial_Engineering
125     ZUHAL   3.21    Chemistry
145     ALİ     3.15    Computer_Engineering
844     AYŞEN   3.14    Electrical_Engineering
236     DİLEK   3.12    Mining_Engineering
147     SEMA    3.12    Control_Engineering
111     İMREN   3.11    Control_Engineering
789     BÜŞRA   3.09    Manufacturing_Engineering
123     UĞUR    3.09    Mathematics_Engineering
512     DİLARA  3.07    Geological_Engineering
874     EKİN    3.07    Mining_Engineering
569     ELİF    3.07    Chemistry
277     HEDİYE  3.07    Molecular_Biology&Genetics
136     MUSAB   3.07    Geomatics_Engineering
190     NURAY   3.07    Electrical_Engineering
295     İHSAN   3.07    Geomatics_Engineering
155     ŞEYDA   3.07    Mathematics_Engineering
666     CENK    3       Architecture
120     NURŞAH  3       Geological_Engineering
150     SELİN   3       Mining_Engineering
```

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.