**Dokuz Eylül University**

**Electrical and Electronics Engineering Department**

**EED 1010 ALGORITHMS AND PROGRAMMING COURSE**

**PROJECT FOR MIDTERM EXAMINATION**

Write a C program which will open the random access (binary) file “YOURNAME\_YOURSURNAME.bin” (e.g. FAIK\_YAREN.bin) and process the records from the file. There are different .bin files for each student, therefore the results of your program will be different then all the other students’ results (THERE IS NO NEED FOR CHEATING!).

The structure used for records in “YOURNAME\_YOURSURNAME.bin” file is as follows:

struct student{ //Structure for the records in the file

char Name[30];

char Surname[30];

char address[50];

unsigned long long No;

unsigned int st\_class;

unsigned int Section;

int grade;

} ;

typedef struct student Students;

The prototypes of the functions that will be used in your program are as follows (you will write these functions):

int read\_student\_database(char \*, Students []);

void count\_students\_in\_sections(Students [], int, Statistics []);

double calculate\_statistics(Students [], int, Statistics []);

void display\_and\_save\_statistics(char \*,Statistics [], int, double);

void sort\_and\_save\_data(char \*, Students [], int);

and the type Statistics is given below:

struct stats{ //Structure for number of students, maximum, minimum and average

int no; //number of students

int max; //maximum

int min; //minimum

double avg; //average

};

typedef struct stats Statistics;

Write the functions, for which the prototypes are given above (DO NOT CHANGE THE PROTOTYPE! OTHERWISE YOUR SOLUTION WILL NOT BE GRADED!).

1. **int read\_student\_database(char \*, Students []) function**: main() function will send two arguments to this function while calling it. The first argument is a string, that is the name of the random access (.bin) file to be opened. The second argument is an array whose elements are of type Students. **JOB OF FUNCTION**: This function will open the random access (.bin) file given to you. It will store each record from the file into array as elements of that array, i.e. it will transfer data from file to memory. It will close the file. It will count and return the number of records in the file, so the return value will be the number of records in the file, i.e. the number of all students registered to the course.
2. **void count\_students\_in\_sections(Students [], int, Statistics []) function**: main() function will send three arguments to this function while calling it. The first argument is an array whose elements are of type Students. The second argument is an integer representing the number of array elements (this number represents array elements that are used, of course array will be declared to have more elements than this number). The third argument is an array whose elements are of type Statistics. **JOB OF FUNCTION:** This function will count the number of students in each section. It will process elements of Students type array one by one. Note that there are six sections (1 – 6) in the course and section information is stored in “Section” member of type Students. The number of students in each section will be stored in “No” member of type Statistics, i.e. the “No” member of first array element of Statistics type array will hold the number of students in section 1.

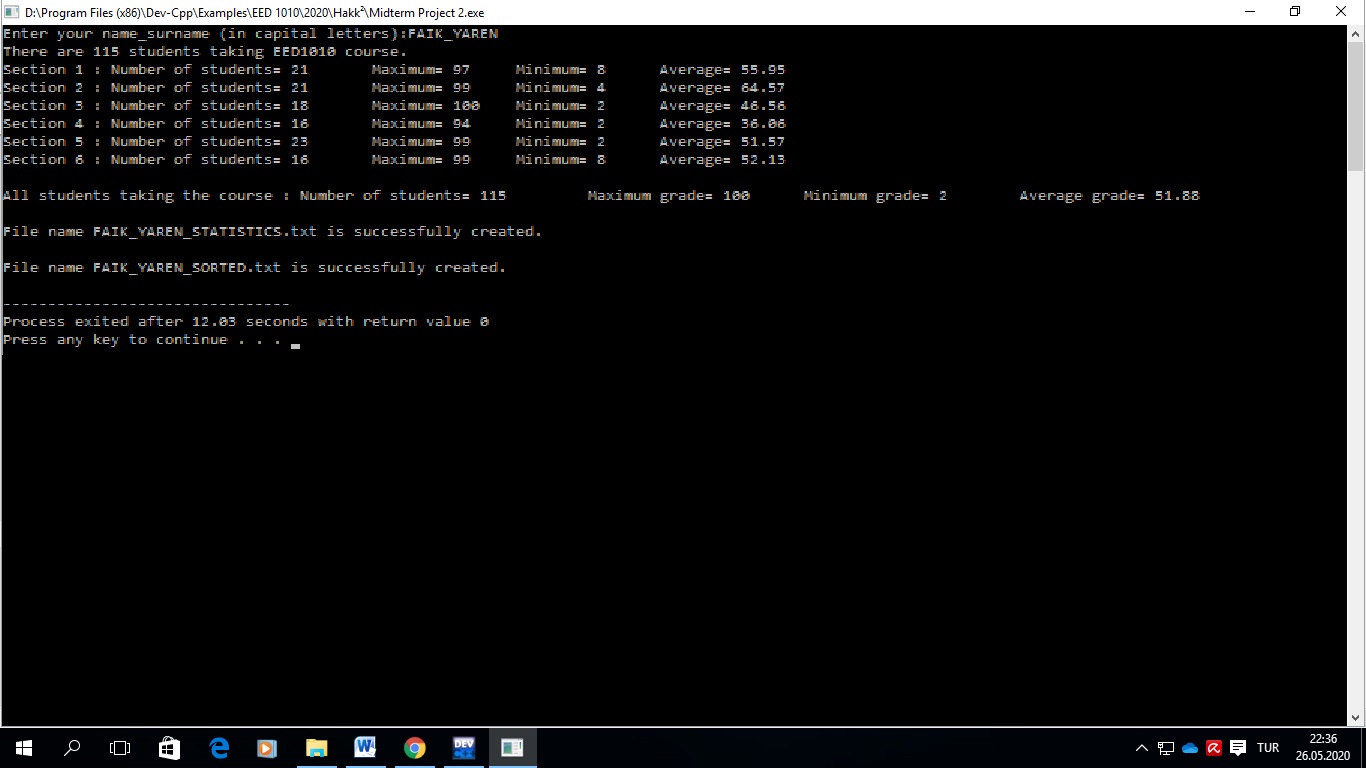
1. **double calculate\_statistics(Students [], int, Statistics []) function**: main() function will send three arguments to this function while calling it. The first argument is an array whose elements are of type Students. The second argument is an integer representing the number of array elements (this number represents array elements that are used, of course array will be declared to have more elements than this number). The third argument is an array whose elements are of type Statistics. **JOB OF FUNCTION:** This function will process elements of Student type array one by one. It will find maximum, minimum and average of grades in each section. Note that there are six sections (1 – 6) in the course. The maximum, minimum and average of grades in that section will be stored in “max”, “min” and “avg” members of corresponding array element of type Statistics, respectively. In other words, “max” member, “min” member and “avg” member of first array element of Statistics type array will hold the maximum grade, minimum grade and average of grades of students in section 1, respectively. This function also computes and returns the average of the grades of all students registered to the course, so the return value will be the average of grades of all students.

1. **void display\_and\_save\_statistics(char \*, Statistics [], int, double) function**: main() function will send four arguments to this function while calling it. The first argument is a string representing a file name. The second argument is an array whose elements are of type Statistics. The third argument is an integer representing the number of all students registered to the course. The fourth argument is a double value representing the average of all students’ grades. **JOB OF FUNCTION:** This function will process elements of Statistics type array one by one. It will find the maximum and minimum of grades of all students. It will display the statistics such as number of students, maximum, minimum and average of grades of students for each section on a separate line on the user screen. Then display number of students, maximum, minimum and average of grades of all students registered to the course. Averages will be displayed by using 2 digits precision after decimal point. A sample user screen will be given. This function will create a sequential access file (text file) whose name is sent by main() function. It will write the statistics to the text file in a similar format that is used to display this information on user screen. A sample data file created by calling this function will be given to you (FAIK\_YAREN\_STATISTICS.txt). **(OF COURSE YOUR PROGRAM WILL PRODUCE A DIFFERENT DATA FILE!)**
2. **void sort\_and\_save\_data(char \*, Students [], int) function**: : main() function will send three arguments to this function while calling it. The first argument is a string representing a file name. The second argument is an array whose elements are of type Students. The third argument is an integer representing the number of all students registered to the course. **JOB OF FUNCTION:** This function will first sort the elements of Students type array by using their “grade” members in descending (decreasing) order. Then it will create a sequential access file (text file) whose name is sent by main() function. It will write elements of sorted Students type array one by one as records to the text file. But it will also include the letter grades as the last information in the records for every student. The letter grades will be determined by using the following table:

|  |  |
| --- | --- |
| **Grade** | **Letter Grade** |
| **Grade ≥ 90** | **AA** |
| **90 > Grade ≥ 85** | **BA** |
| **85 > Grade ≥ 80** | **BB** |
| **80 > Grade ≥ 75** | **CB** |
| **75 > Grade ≥ 70** | **CC** |
| **70 > Grade ≥ 65** | **DC** |
| **65 > Grade ≥ 60** | **DD** |
| **60 > Grade ≥ 50** | **FD** |
| **50 > Grade** | **FF** |

A sample data file created by calling this function will be given to you (FAIK\_YAREN\_SORTED.txt). **(OF COURSE YOUR PROGRAM WILL PRODUCE A DIFFERENT DATA FILE!)**

1. **main() function:** An array of Students type will be declared with 200 elements (this will be enough for the number of records in the binary file). An array with necessary number of elements of type Statistics will also be declared. The program will ask the user to enter his / her name in “NAME\_SURNAME” format. (Here you will enter YOURNAME\_YOURSURNAME when you run your program. So your program will open file “YOURNAME\_YOURSURNAME.bin” by calling read\_student\_database() function.) Then it will create file name as “NAME\_SURNAME.bin”. It will call read\_student\_database() and count\_students\_in\_sections() functions. Display the total number of students registered to the course. It will call calculate\_statistics() function. Then it will create the string “NAME\_SURNAME\_STATISTICS.txt” and send this as the first argument to display\_and\_save\_statistics() function. It will then create the string “NAME\_SURNAME\_SORTED.txt” and send this as the first argument to sort\_and\_save\_data() function. The program will end after returning from sort\_and\_save\_data() function. A sample user screen is given below:



**GRADING CRITERIA FOR YOUR MIDTERM PROJECT**

1. Your code: 50 points
2. Indentation and comments in your code: 5 points.
3. “YOURNAME\_YOURSURNAME\_STATISTICS.txt” file: 20 points.
4. “YOURNAME\_YOURSURNAME\_SORTED.txt” file: 25 points.

You will download and use the file named “YOURNAME\_YOURSURNAME.bin” as datafile for your program.

You will upload your code (Yourname\_Yoursurname\_Yourstudentnumber\_Midterm.cpp file), “YOURNAME\_YOURSURNAME\_STATISTICS.txt” file created by your program and “YOURNAME\_YOURSURNAME\_SORTED.txt” file created by your program.

NOTE: You are given sample files FAIK\_YAREN\_STATISTICS.txt and FAIK\_YAREN\_SORTED.txt . Your program will produce files using same formats as in these files. **So inspect these files and use the same format for your own files**. (Remember that 45 points of your grade will be evaluated from these files!) For example, in FAIK\_YAREN\_SORTED.txt file the sixth column represents st\_class member values, the seventh column represents Section member values of records.