**CEN 134 TAKE-HOME MIDTERM EXAM**

In the first code, I started with creating a struct type named “StudentMark”. This struct type will be able to store name, surname, midterm point and final point of a student. Then in main, I created a struct pointer named “STUDENT\_MARK” by using StudentMark struct type. Also to make sure it is empty, I equalized it to NULL because I would not know if it was empty after allocationg memory for it by using malloc.

I wanted to create a structure array to store all the student’s informations because I find using arrays more organized. Because of that I wanted the user to enter how many student’s information they want to store to be able to determine the size of the array.

After that I allocated memory for my structure array by using malloc function and then I checked if the program managed to allocate memory successfuly. If it can’t allocate memory successfuly, an error message will be written and the program will be terminated.

Then I wanted to fill the structure array I created. I wrote the function “studentInfo” to do this. This function takes the address of a structure array and the size of this array as parameters. I opened a for loop starting from the first structure in the array (i=0) to the last (i=studentAmount-1), entered name, surname, midterm point, final point information for all one by one also checked if the midterm and the final points were valid by using two other while loops inside the for loop. These while loops will keep going on till the user enters valid points. At the end of this function I returned what is inside the structure array by using “\*” operator. When I used this function in main and equalized to “\*STUDENT\_MARK”, I managed to fill the inside of the place STUDENT\_INFO pointer points.

I defined a file pointer named “fp” and opened a binary file in main in write mode and named it “marks\_190704001.txt”. While doing that, I put the address of the file to fp. I chose to open a binary file because some of the information the user enters will be numbers, and by using a binary file I might be able to reduce the memory usage. Then I checked if the file could be opened successfuly. The program will give an error message if the file can’t be opened properly and the program will be terminated.

I created a void function called “StoreStudentInfo” to be able to write what is inside of the place STUDENT\_INFO points to the file I opened. I got the address of a structure array, the size of it and the file I want to write onto as parameters. I firstly wrote the size of the structure array to the file by using “fwrite” function. I will use that information while reading the file. After that I wrote what is inside of the place STUDENT\_INFO pointer points again using fwrite function. Then used this function in main. Because I did not have nothing left to do with the file I opened also the memory I allocated, I freed the memory and closed the file.

-----------------------------------------------------------------------------------------------------------------

In the second code, I again created the structure type “StudentMark”. Then in main, I defined a file pointer named “fp” and opened the binary file “marks\_190704001.txt” in read mode. I checked if the file could be opened successfuly. The program will give an error message if the file can’t be opened properly and the program will be terminated. Then I decided to read from the file I opened to get the “studentAmount” I had written into the file in the first code. I read the first 4 bytes of the file which was the studentAmount by using “fread” function and I equalized the information I read to the integer named “studentAmount” I defined earlier in the second code. At first when I was coding, I didn’t write the studentAmount information into the file that’s why in the second code, I didn’t have a way to know how many student’s information was written in the file. Because of that I decided to ask the user, how many student’s information they would like to get. But then I thought it would be a problem if they entered a number bigger than the total student amount or if they entered a smaller value, the program would not be able to print all the information. That is why I decided to get the studentAmount directly from the first code by writing it on the file “marks\_190704001.txt”. After reading this information, the cursor on the file is right before the structure array’s information.

I again created a structure pointer called “STUDENT\_MARK” and again I allocated memory by using malloc function to create a structure array. If it can’t allocate memory successfuly, an error message will be written and the program will be terminated. This structure array will store the information the program reads from the file I opened.

I created a void function called “GetStudentInfo”. This function takes a structure array’s address, the size of it and the file’s address I will read from as parameters. By using fread function again, I continued reading from the file I had already opened and stored the information I read from the file in the structure array.

After that I created another function called “averageCalc” and used it in main. This function takes a structure array’s address and its size as parameters. I created a new array called “storeAverage” in this function. This array will store the average of all students so its size will be equal to studentAmount. Then I opened a for loop starting from the first student information to the last one. For each of them I got the information of midterm and final points then calculated the average of them and stored them in the storeAverage array. After storing one in the array, I printed it on the screen then continued for the next one and kept going till I printed the average of all students.

After all these, because I did not have nothing left to do with the file I opened also the memory I allocated, I freed the memory and closed the file in main.

**Yeşim Yalçın**

**190704001**