Algorithm Design and Programming II

Lab 5 (30 Points)

Objectives:

- Learn how to use structure array and type casting.
- Learn how to use a header file.

Description:

- Modify your functions from the pre-lab and lab 4: ***createArray***, ***getArraySize***, and ***freeArray***.
- In ***createArray*** function, create an array of 10 Record structures where 42000 ≤ Salary ≤ 89900, using ***rand()*** function, and 0 ≤ Employee ID (EID) ≤ 9.
- Each Record should look like as follows:

```
typedef struct {
        int EID;
        double salary;
} Record;
```

- Store an integer array size at the beginning of the array.

After updating your createArray(), your array should look like this:

size	employee[0]	employee[1]	employee[2]	…	employee[9]
10	**EID**: 0 **Salary**: 35290.55	**EID**: 1 **Salary**: 46677.89	**EID**: 2 **Salary**: 59788.56	…	**EID**: 9 **Salary**:67789.89

Main function steps:

- Call ***createArray*** function.
- Sum up each employee's salary in the array.
- Print out the result as shown in the **Example** below.
- Free the allocated array using your ***freeArray*** function.

Every user-defined function must have a comment describing:

- What function does;
- What parameter values are;
- What value it returns.

Example from terminal window:

$ **gcc main.c lab5.c -Wall -Werror**

$ **./a.out**

The summation of 10 employees' salary is $**1802044.56**

Grading Criteria:

- Main program: 6 points
- createArray function: 12 points
- getArraySize function: 6 points
- freeArray function: 6 points

Note:

- If your code does not compile with **–Wall** and **–Werror**, you will receive a **zero** for this assignment.
- You need to finish at least **three** peer reviews within three days of this lab. Otherwise, you will get a 20% penalty.
- You will lose points if you don't have enough comments.