CS36 C Programming Homework 4
20 points (Lesson 1 to Lesson 61)

1. You must have your student name, student ID, and program set number/description in the title banner at the very top of each program set. Late homework will not be accepted for whatever reasons you may have.

*********_**for this homework, you are to submit your Program sets to Canvas**_ under Homework 4 link*********

   a. Name your files: HW4_PS1_ lastname_firstinitial.c for Program Set 1 and HW4_PS2_ lastname_firstinitial.c for PS2 and so on. PS means program set. If there are two program sets you will submit two files one for each program set. Example if your name is Joe Smith then the filename will be HW4_PS1_smith_j.c
   b. You must submit your homework by the deadline at the specified upload link on Canvas under homework 4. If the deadline is past, Program Sets will not be graded. Homework submitted via email attachment, comment in Canvas, Canvas message, or by any other method is not accepted and will be given a zero for no submission.
   c. if you do not follow instructions on file naming provided in this section you will receive a zero for the question you did not correctly name the file.

2. Please format you output properly, for example all dollar amounts should be printed with 2 decimal places when specified. Make sure that your output values are correct (check the calculations).

3. Use only the 'tools' in the topics we covered up till lesson 61 only. **A zero grade will be given if any topics not covered from lesson 1 to lesson 61 is used in each of your program set.**

Each student is expected to do their own work. **IF IDENTICAL PROGRAMS ARE SUBMITTED, EACH IDENTICAL PROGRAM WILL RECEIVE A SCORE OF ZERO.**

Grading:

If a program does not compile the program set will receive a zero score. If the program compiles and run but does not have proper declaration of variables, syntax, logic, and displays the correct output (given in the sample test runs) with proper formatting as specified in the question, the program set will receive a zero score. Each program set must compile and must run correctly with proper declaration of variables, syntactically, logically, and display the correct output (given in the sample test runs) as specified then you will receive the full points for that question. Then points will be deducted for not having proper:

a. Comments 1 pt deducted for each infraction
   - Title Banner --Your name, description at the beginning of each program set.
   - Short description of the what each section or function of your codes do.

b. Consistency/Readability 1 pt deducted for each infraction
   - Spacing(separate each section of codes with a blank line
   - Separate each function with 2 blank lines
   - Proper Indentation (if statements, loops, switch see lessons example programs)
   - Proper naming of variables no a,b,c – use descriptive and mnemonics)

c. Required elements 1 pts deducted for each infraction
   - proper formatting for output when specified
   - all monetary values must be in 2 decimal places when specified in sample test runs

d. Use only 'tools' in the topics that have been covered in class (lesson 1 to lesson 61)

e. Output (you **must provide the specified number of test runs or your program set will receive a zero score**)
   - to be displayed at the end of the program listing(codes) and commented
   - must have the number of test runs as specified in each program set.
   - must use the data for test runs when they are provided for you in the question.

Points will be deducted from items a. to d. above until your Program Set reaches zero points.

USE ONLY THE TOOLS YOU HAVE BEEN TAUGHT IN CLASS ONLY, IF YOU USE ANYTHING WE HAVE NOT COVERED YET YOU WILL RECEIVE A ZERO FOR THAT PROGRAM SET .

NO GLOBAL VARIABLES ALLOWED and you must use function prototype for this homework 3. The break, continue and goto C commands are not allowed to be used. A zero will be given for the program set if your program contains a global variable, break, continue or goto command. The break command is allowed only as part of the switch statement.

Program Set 1

Write a C program to calculate salary raise for employees which uses the following functions for an array of structures for 6 employees. You must use functions for this program set as described in class lecture.

If salary is between	$ 0 <	$ 30000 the rate is 7.0%
If salary is between	$ 30000 <=	$ 40000 the rate is 5.5%
If salary is greater than	$ 40000	the rate is 4.0%

1. Allow the user to enter the employee name and salary for each employee and for each employee calculate the raise, and the new salary with the raise into an array of structures.
2. Sort the array of structures on name and print the array after the sort is completed.
3. Calculate and print the total Salary, total raises, and total new salary.
4. Save the array of structures to a text file.
5. Retrieve and print the text file.
6. Save the array of structures to a binary file.
7. Retrieve and print the binary file.

Calculation of Salary Raises

Name	Salary	Rate %	Raise	New Salary
Susan	25000.00	7.00	1750.00	26750.00
Jim	30000.00	5.50	1650.00	31650.00
Gloria	35000.00	5.50	1925.00	36925.00
Ros	40000.00	5.50	2200.00	42200.00
Ben	40001.00	4.00	1600.04	41601.04
Tim	45000.00	4.00	1800.00	46800.00
Total	215001.00		10925.04	225926.03

Your output should look like this sample test run(only one test run required using the data provided in the table above)

Sample Test Run

```
Enter the employee's name: Susan
Enter salary: 25000
Enter the employee's name: Jim
Enter salary: 30000
Enter the employee's name: Gloria
Enter salary: 35000
Enter the employee's name: Ros
Enter salary: 40000
Enter the employee's name: Ben
Enter salary: 40001
Enter the employee's name: Tim
Enter salary: 45000
Original Array of Structure before Sorting
```

Calculation of Salary Raises

Employee	Salary	Rate	Raise	New Salary
Susan	25000.00	7.00	1750.00	26750.00
Jim	30000.00	5.50	1650.00	31650.00
Gloria	35000.00	5.50	1925.00	36925.00
Ros	40000.00	5.50	2200.00	42200.00
Ben	40001.00	4.00	1600.04	41601.04
Tim	45000.00	4.00	1800.00	46800.00
Total	215001.00		10925.04	225926.03

Array of Structure after Sorting

Calculation of Salary Raises

Employee	Salary	Rate	Raise	New Salary
Ben	40001.00	4.00	1600.04	41601.04
Gloria	35000.00	5.50	1925.00	36925.00
Jim	30000.00	5.50	1650.00	31650.00
Ros	40000.00	5.50	2200.00	42200.00
Susan	25000.00	7.00	1750.00	26750.00
Tim	45000.00	4.00	1800.00	46800.00
Total	215001.00		10925.04	225926.03

From Save Text file

Calculation of Salary Raises

Employee	Salary	Rate	Raise	New Salary
Ben	40001.00	4.00	1600.04	41601.04
Gloria	35000.00	5.50	1925.00	36925.00
Jim	30000.00	5.50	1650.00	31650.00
Ros	40000.00	5.50	2200.00	42200.00
Susan	25000.00	7.00	1750.00	26750.00
Tim	45000.00	4.00	1800.00	46800.00
Total	215001.00		10925.04	225926.03

From Save Binary file

Calculation of Salary Raises

Employee	Salary	Rate	Raise	New Salary
Ben	40001.00	4.00	1600.04	41601.04
Gloria	35000.00	5.50	1925.00	36925.00
Jim	30000.00	5.50	1650.00	31650.00
Ros	40000.00	5.50	2200.00	42200.00
Susan	25000.00	7.00	1750.00	26750.00
Tim	45000.00	4.00	1800.00	46800.00
Total	215001.00		10925.04	225926.03

Here is the template for your program. You must use this template for your program and you are not allowed to change any codes inside the template. Changing any codes in the given template will result in a zero for the whole program set. Write codes for those fonts in red. You only need one test run for all 6 employees. (download the hw4template.c and use it for your program)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define SIZE 6


struct Employee
{
        //declare your struct data members here

};

//function prototype here


int main()
{
        //declaration of variables
        struct Employee e[SIZE];
        float ts, tr, tn;

        //load data into struct, calculate raises, salaries and new salaries and print the
        //original array of struct
        load(e, SIZE);
        ARate(e, SIZE);
        calcRaise(e, SIZE);
        total(e, SIZE, &ts, &tr, &tn);
        printf("Original Array of Structure before Sorting\n\n");
        title();
        print(e, SIZE);
        printTotals(ts, tr, tn);

        //sort the struct of array and print the sorted struct of array
        sort(e, SIZE);
        total(e, SIZE, &ts, &tr, &tn);
        printf("Array of Structure after Sorting\n\n");
        title();
        print(e, SIZE);
        printTotals(ts, tr, tn);

        //**************Section for text files*****************
        printf("\n\n");
        printf("From Save Text file\n\n");
        title();
        //save the struct of array to a text file
        savetext(e, SIZE);

        //retrieve the text file and print the data
        gettext(e, SIZE);
        print(e, SIZE);
        total(e, SIZE, &ts, &tr, &tn);
        printTotals(ts, tr, tn);


        //**************Section for binary files*****************
        printf("\n\n");
        printf("From Save Binary file\n\n");
        title();

        //save struct of array to a binary file
        savebn(e,SIZE);

        //retrieve data from binary file and print the data
        getbn(e, SIZE);
        print(e, SIZE);
        total(e,SIZE,&ts, &tr, &tn);
        printTotals(ts, tr, tn);

       return 0;

}


void load (struct Employee e[], int n)
{
        //let user enter employee's name and salary
```

```
}


void ARate(struct Employee e[], int n)

{
        //assign each employee rate to the struct's rate data member here by checking salary //range

        //for example
        //        if(e[i].sal >= 0.00 && e[i].sal < 30000.00)
        //             e[i].rate = 7.00;

}

void calcRaise(struct Employee e[], int n)
{
        //calculate the raise and new salary for each employee and store into the
        //struct's raise and new salary data members
        //      raise = salary * rate/100
        //      new salary = salary + raise
}



void sort(struct Employee e[], int n)
{
        //sort the struct on employee name in ascending order

}


void total(struct Employee e[], int n, float *ts, float *tr, float *tn )

{
        //ts = total salary(sum the salaries of all 6 employees)
        //tr = total raise(sum the raises of all 6 employees)
        //tn = total new salary(sum of new salaries of all 6 employees)

}


void title()
{

printf("\t\t\tCalculation of Salary Raises\n\n");

printf("Employee\t");     printf("Salary\t\t"); printf("Rate %\t"); printf("  Raise\t\t");
printf("New Salary\n\n");
}


void print(struct Employee e[], int n)
{
        int i;

        for(i = 0; i < n; i++)
        {
        printf("%s\t\t", e[i].name); printf("%10.2f\t", e[i].sal); printf("%4.2f\t", e[i].rate);
        printf("%8.2f\t", e[i].raise); printf("%8.2f\t\n", e[i].newSal);
        }

}


void printTotals(float ts, float tr, float tn)
{

        printf("Total\t\t"); printf("%10.2f\t\t", ts); printf("%8.2f\t", tr); printf("%8.2f\n\n",
        tn);

}


void savetext(struct Employee e[], int n)
{

        //save the struct to text file employees.txt
}


void gettext(struct Employee e[], int n)

{
```

```
        //retrieve the data from employees.txt file
}


void savebn(struct Employee e[], int n)
{
        //save the struct to a binary file employees.bin
}

void getbn(struct Employee e[], int n)
{
        //retrieve the data from the employees.bin file
}
```