Brian Blalock

Assignment 5

Source Code

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


struct employeeRank {                   // representation of ranks employees can have
 int level;
 float salary;
};
struct employee {                       // representation of employee information
 int id;
 char name[20];
 int JoinYear;
 struct employeeRank rank;
};



//function declarations

int SearchEmployee (struct employee EmployeeArray[], int N, struct employee key);

int CountEmployee_Salary (struct employee EmployeeArray[], int N, float S1, float S2);


int main(){

        // employees in database
        struct employee Bia = {3452, "Bia", 2014, {3, 45000}};
        struct employee Chris = {6109, "Chris", 2016, {2, 30000}};
        struct employee Brian = {5421, "Brian", 2013, {4, 55000}};
        struct employee Kevin = { 7390, "Kevin", 2016, { 1 , 20000}};

        // make the database and a vairable to represent it's size
        struct employee Employees[] = {Bia,Chris,Brian,Kevin};
        int EmployeesSize = 4;
```

```
int menuOption = 1;    // used to store which option is chosen
while ( menuOption) {



        printf("Access Employee Database:\n 0.Exit\n 1.Search for employee\n 2.Count
employees in Salary Range\n"); // menu prompt

        scanf("%d", &menuOption);                    //store menu option

        if( menuOption == 1){                                                    // if
they are searching for an employee

                struct employee key;                                             // used
to store the information of the employee being searched for


                //prompt to get information for search
                printf("Enter employee id\n");
                scanf("%d", &key.id);
                printf("Enter employee name\n");
                scanf("%s", key.name);
                printf("Enter year employee was hired\n");
                scanf("%d", &key.JoinYear);
                printf("Enter employee's rank level\n");
                scanf("%d", &key.rank.level);
                printf("Enter employee salary\n");
                scanf("%f", &key.rank.salary);


                // get where the employee is in the database
                int indexOfEmployee = SearchEmployee(Employees, EmployeesSize,
key);

                // print the index of where the employee is
                printf( "Employee is located at index: %d\n\n", indexOfEmployee);
        }
        else if( menuOption == 2){                          // if they are counting
employees within a salary range

                float salaryLower;              // store lower bound
                float salaryUpper;              //store upper bound
```

```
                        // get bounds from user
                        printf( "Enter lower bound of salary range.\n");
                        scanf("%f", &salaryLower);
                        printf("Enter upper bound of salary range\n");
                        scanf("%f", &salaryUpper);

                        // count employees in the range
                        int numInRange = CountEmployee_Salary ( Employees, EmployeesSize,
salaryLower, salaryUpper);

                        //print what was found
                        printf("There are %d employees in the salary range\n\n", numInRange);
                }
        }

}

int SearchEmployee (struct employee EmployeeArray[], int N, struct employee key){



        for ( int i = 0 ; i < N ; i ++){      //loop through the array

                // if all the data from the current emploee in EmployeeArray equals the data in the
key
                if( EmployeeArray[i].id == key.id && strcmp(EmployeeArray[i].name, key.name)
== 0 && EmployeeArray[i].JoinYear == key.JoinYear && EmployeeArray[i].rank.level ==
key.rank.level && EmployeeArray[i].rank.salary == key.rank.salary)  {



                        return i;          // return the current index


                }

        }

        return -1;        // else return -1

}
```

```
int CountEmployee_Salary (struct employee EmployeeArray[], int N, float S1, float S2){

	int count = 0;   // store counter

	for ( int i = 0 ; i < N ; i ++){                              // loop through array
		float salary = EmployeeArray[i].rank.salary; //get current employee's salary

		if( salary  > S1 && salary < S2 ){                    // if the salary is in the range
			count++;                                                                //
increment counter;

		}
	}

	return count;   // return the counter

}
```

Console output:

```
$ ./project
Access Employee Database:
 0.Exit
 1.Search for employee
 2.Count employees in Salary Range
1
Enter employee id
3452
Enter employee name
Bia
Enter year employee was hired
2014
Enter employee's rank level
3
Enter employee salary
45000
Employee is located at index: 0

Access Employee Database:
 0.Exit
 1.Search for employee
 2.Count employees in Salary Range
1
```

Enter employee id
7390
Enter employee name
Kevin
Enter year employee was hired
2016
Enter employee's rank level
1
Enter employee salary
2000
Employee is located at index: -1

Access Employee Database:
 0.Exit
 1.Search for employee
 2.Count employees in Salary Range
1
Enter employee id
7390
Enter employee name
Kevin
Enter year employee was hired
2016
Enter employee's rank level
1
Enter employee salary
20000
Employee is located at index: 3

Access Employee Database:
 0.Exit
 1.Search for employee
 2.Count employees in Salary Range
2
Enter lower bound of salary range.
25000
Enter upper bound of salary range
46000
There are 2 employees in the salary range

Access Employee Database:
 0.Exit
 1.Search for employee

2.Count employees in Salary Range
2
Enter lower bound of salary range.
26000
Enter upper bound of salary range
27000
There are 0 employees in the salary range

Access Employee Database:
0.Exit
1.Search for employee
2.Count employees in Salary Range
0