# HW Assignment 5

Marshall Slagle
7/30/18

## Program Code

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


struct employeeRank //Declaring a Structure with specific members
{
    int level; //Declaring integer to hold Level
    float salary; //Declaring float to hold salary
};

struct employee //Declaring a Structure with specific members
{
    int id; //Declaring integer to hold ID
    char name[20]; //Declare character array to hold name as string
    int joinYear; //Declaring integer to hold Joinyear
    struct employeeRank rank; //Declaring sub structure called rank to hold rank
specific information
};

int SearchEmployee(struct employee EmployeeArray[], int N, struct employee key);
//Method Declaration
int CountEmployee_Salary(struct employee EmployeeArray[], int N, float E1, float E2);
//Method Declaration
void PrintMethod(struct employee EmployeeArray[], int search, int count, float E1,
float E2); //Method Declaration



void PrintMethod(struct employee EmployeeArray[], int search, int count, float E1,
float E2) //Method to print outputs
{
    if(search>=0) //if statment to determine if returned search value is not lower
then 0
    {
        printf("\nEmployee Idex: %d Matched the Key\n", search); //prints employee
index
        printf("*********************************\n"); //prints border
        printf("\nEmployee info:\nName: %s\nID: %d\nYear Joined: %d\nEmployee Level:
%d\nSalary: %f\n", EmployeeArray[search].name, EmployeeArray[search].id,
EmployeeArray[search].joinYear,EmployeeArray[search].rank.level,
EmployeeArray[search].rank.salary); //prints members of structure
        printf("\n*****************************\n\n"); //prints border
    }
    else //else statement if returned search is -1
    {
        printf("\n* No Employee matched the provided key *\n"); //prints error message
    }

    if(count>0) //if statement to determins if returned count value is greater then 0
    {
        printf("\n%d Employee(s) have a Salary between %f and %f\n\n", count, E1, E2);
//prints number of employees within range
        printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n"); //prints
border
    }
    else
    {
        printf("\n** No Employee(s) has a Salary between %f and %f **\n\n", E1, E2);
//prints error message
        printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n"); //prints
ending seperation border

    }
}

int SearchEmployee(struct employee EmployeeArray[], int N, struct employee key)
//Method to match key with array of structures
{
    bool idB; //Declare boolean idB
    bool nameB; //Declare boolean nameB
    bool joinYearB; //Declare boolean joinYear B
    bool levelB; //Declare boolean levelB
    bool salaryB; //Declare boolean salaryB
    //boolean variables are used to store truthfullness of each member within each
structrue of the array

    for(int i=0; i<N; i++) //for loop to access each index of the structure array
    {
        if(EmployeeArray[i].id==key.id) //if statement to determine the truthfullness
of the ID member with the key
        {
            idB=true; //initializing idB to true
        }
        else //else statment to set boolean value to false to avoid misidentifying the
truthfullness of the comparison on each iteration of the for loop
        {
            idB=false; //initializing idb to false
        }
        if(strcmp(EmployeeArray[i].name,key.name)==0) //if statement to determine the
truthfullness of the Name member with the key
```

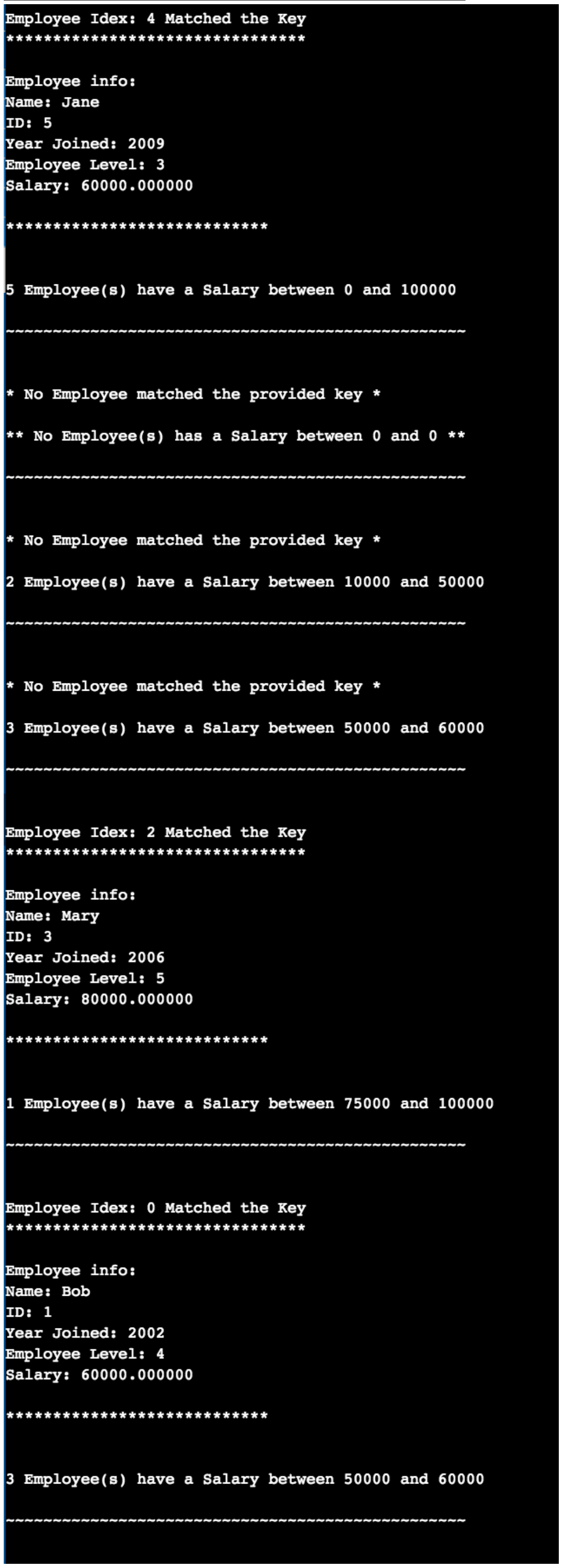
```
        {
            nameB=true; //initializing nameB to true
        }
        else //else statment to set boolean value to false to avoid misidentifying the
truthfullness of the comparison on each iteration of the for loop
        {
            nameB=false; //initializing nameB to false
        }
        if(EmployeeArray[i].joinYear==key.joinYear) //if statement to determine the
truthfullness of the joinYear member with the key
        {
            joinYearB=true; //initializing joinYearB to true
        }
        else //else statment to set boolean value to false to avoid misidentifying the
truthfullness of the comparison on each iteration of the for loop
        {
            joinYearB=false; //initializing joinYearB to false
        }
        if(EmployeeArray[i].rank.level==key.rank.level) //if statement to determine
the truthfullness of the Level member with the key
        {
            levelB=true; //initializing levelB to true
        }
        else //else statment to set boolean value to false to avoid misidentifying the
truthfullness of the comparison on each iteration of the for loop
        {
            levelB=false; //initializing levelB to false
        }
        if(EmployeeArray[i].rank.salary==key.rank.salary) //if statement to determine
the truthfullness of the Salary member with the key
        {
            salaryB=true; //initializing salaryB to true
        }
        else //else statment to set boolean value to false to avoid misidentifying the
truthfullness of the comparison on each iteration of the for loop
        {
            salaryB=false; //initializing slarayB to false
        }

        if(idB==true && nameB==true && joinYearB==true && levelB==true &&
salaryB==true) //if all boolean values representing each member is true then the index
of the current iteration will be returned
        {
            return i; //returns index to main method
        }
    }
    return -1; //return -1 as default if no iteration of the for loop returned a
truthfull value
```

```
}

int CountEmployee_Salary(struct employee EmployeeArray[], int N, float E1, float E2)
//Method to compare amount of employees within a given salary range
{
    int count = 0; //initializing count value to 0 as a base

    for(int i=0; i<N; i++) //for loop to add 1 to the count variable if the salary is
within th eprovided range sent as a parameter
    {
        if(EmployeeArray[i].rank.salary<=E2 && EmployeeArray[i].rank.salary>=E1) //if
statement to determine if salary is between the two value sent as parameters
        {
            count++; //adds 1 to count value
        }
    }

    return count; //returns count value to main method
}


int main() //Main Method
{
    struct employee EmployeeArray[5]; //Declaring Array of structures
    int N = sizeof(EmployeeArray)/sizeof(EmployeeArray[0]); //initializing N variable
to size of structure array

    EmployeeArray[0].id = 1; //initializing index 0 Id Member to 1
    strcpy(EmployeeArray[0].name,"Bob"); //initializing index 0 name Member to Bob
    EmployeeArray[0].joinYear = 2002; //initializing index 0 joinYear Member to 2002
    EmployeeArray[0].rank.level = 4; //initializing index 0 level Member to 4
    EmployeeArray[0].rank.salary = 60000; //initializing index 0 salary Member to
60000

    EmployeeArray[1].id = 2; //initializing index 1 Id Member to 2
    strcpy(EmployeeArray[1].name,"Billy"); //initializing index 1 name Member to Billy
    EmployeeArray[1].joinYear = 2005; //initializing index 1 joinYear Member to 2005
    EmployeeArray[1].rank.level = 3; //initializing index 1 level Member to 3
    EmployeeArray[1].rank.salary = 50000; //initializing index 1 salary Member to
50000

    EmployeeArray[2].id = 3; //initializing index 2 ID Member to 3
    strcpy(EmployeeArray[2].name,"Mary"); //initializing index 2 name Member to Mary
    EmployeeArray[2].joinYear = 2006; //initializing index 2 joinYear Member to 2006
    EmployeeArray[2].rank.level = 5; //initializing index 2 level Member to 5
    EmployeeArray[2].rank.salary = 80000; //initializing index 2 salary Member to
80000
```

```
    EmployeeArray[3].id = 4; //initializing index 3 ID Member to 4
    strcpy(EmployeeArray[3].name,"John"); //initializing index 3 name Member to John
    EmployeeArray[3].joinYear = 2008; //initializing index 3 joinYear Member to 2008
    EmployeeArray[3].rank.level = 2; //initializing index 3 level Member to 2
    EmployeeArray[3].rank.salary = 40000; //initializing index 3 salary Member to
40000

    EmployeeArray[4].id = 5; //initializing index 4 ID Member to 5
    strcpy(EmployeeArray[4].name,"Jane"); //initializing index 4 name Member to Jane
    EmployeeArray[4].joinYear = 2009; //initializing index 4 joinYear Member to 2009
    EmployeeArray[4].rank.level = 3; //initializing index 4 level Member to 3
    EmployeeArray[4].rank.salary = 60000; //initializing index 4 salary Member to
60000

    struct employee Key; //Declaring Key of type employee structure
    Key.id = 5; //initializing ID Member to 5
    strcpy(Key.name,"Jane"); //initializing name Member to Jane
    Key.joinYear = 2009; //initializing joinYear Member to 2009
    Key.rank.level = 3; //initializing level Member to 3
    Key.rank.salary = 60000; //initializing salary Member to 60000

    int search; //Declare search variable to store return from SearchEmployee Method
    int count; //Declare count variable to store return from CountEmployee_Salary
    float E1; //Declare E1 variable to store lower bound of salary range
    float E2; //Declare E2 variable to store upper bound of salary range

    E1=0; E2=100000; //initializing E1 and E2 to 0 and 100000
    search = SearchEmployee(EmployeeArray, N, Key); //storing return from method call
in variable
    count = CountEmployee_Salary(EmployeeArray, N, E1, E2); //storing return from
method call in variable
    PrintMethod(EmployeeArray, search, count, E1, E2); //calling print Method

    Key.id = 5; //initializing ID Member to 5
    strcpy(Key.name,"Jane"); //initializing name Member to Jane
    Key.joinYear = 2008; //initializing joinYear Member to 2008
    Key.rank.level = 2; //initializing level Member to 2
    Key.rank.salary = 60000; //initializing salary Member to 60000

    E1=0; E2=0; //initializing E1 and E2 to 0 and 100000
    search = SearchEmployee(EmployeeArray, N, Key); //storing return from method call
in variable
    count = CountEmployee_Salary(EmployeeArray, N, E1, E2); //storing return from
method call in variable
    PrintMethod(EmployeeArray, search, count, E1, E2); //calling print Method

    Key.id = 5; //initializing ID Member to 5
    strcpy(Key.name,"John"); //initializing name Member to John
    Key.joinYear = 2009; //initializing joinYear Member to 2009
    Key.rank.level = 3; //initializing level Member to 3
    Key.rank.salary = 40000; //initializing salary Member to 40000

    E1=10000; E2=50000; //initializing E1 and E2 to 10000 and 50000
    search = SearchEmployee(EmployeeArray, N, Key); //storing return from method call
in variable
    count = CountEmployee_Salary(EmployeeArray, N, E1, E2); //storing return from
method call in variable
    PrintMethod(EmployeeArray, search, count, E1, E2); //calling print Method

    Key.id = 5; //initializing ID Member to 5
    strcpy(Key.name,"Bob"); //initializing name Member to Bob
    Key.joinYear = 2009; //initializing joinYear Member to 2009
    Key.rank.level = 5; //initializing level Member to 5
    Key.rank.salary = 60000; //initializing salary Member to 60000

    E1=50000; E2=60000; //initializing E1 and E2 to 50000 and 60000
    search = SearchEmployee(EmployeeArray, N, Key); //storing return from method call
in variable
    count = CountEmployee_Salary(EmployeeArray, N, E1, E2); //storing return from
method call in variable
    PrintMethod(EmployeeArray, search, count, E1, E2); //calling print Method

    Key.id = 3; //initializing ID Member to 3
    strcpy(Key.name,"Mary"); //initializing name Member to Mary
    Key.joinYear = 2006; //initializing joinYear Member to 2006
    Key.rank.level = 5; //initializing level Member to 5
    Key.rank.salary = 80000; //initializing salary Member to 80000

    E1=75000; E2=100000; //initializing E1 and E2 to 75000 and 100000
    search = SearchEmployee(EmployeeArray, N, Key); //storing return from method call
in variable
    count = CountEmployee_Salary(EmployeeArray, N, E1, E2); //storing return from
method call in variable
    PrintMethod(EmployeeArray, search, count, E1, E2); //calling print Method

    Key.id = 1; //initializing ID Member to 1
    strcpy(Key.name,"Bob"); //initializing name Member to Bob
    Key.joinYear = 2002; //initializing joinYear Member to 2002
    Key.rank.level = 4; //initializing level Member to 4
    Key.rank.salary = 60000; //initializing salary Member to 60000

    E1=50000; E2=60000; //initializing E1 and E2 to 50000 and 60000
    search = SearchEmployee(EmployeeArray, N, Key); //storing return from method call
in variable
    count = CountEmployee_Salary(EmployeeArray, N, E1, E2); //storing return from
method call in variable
```

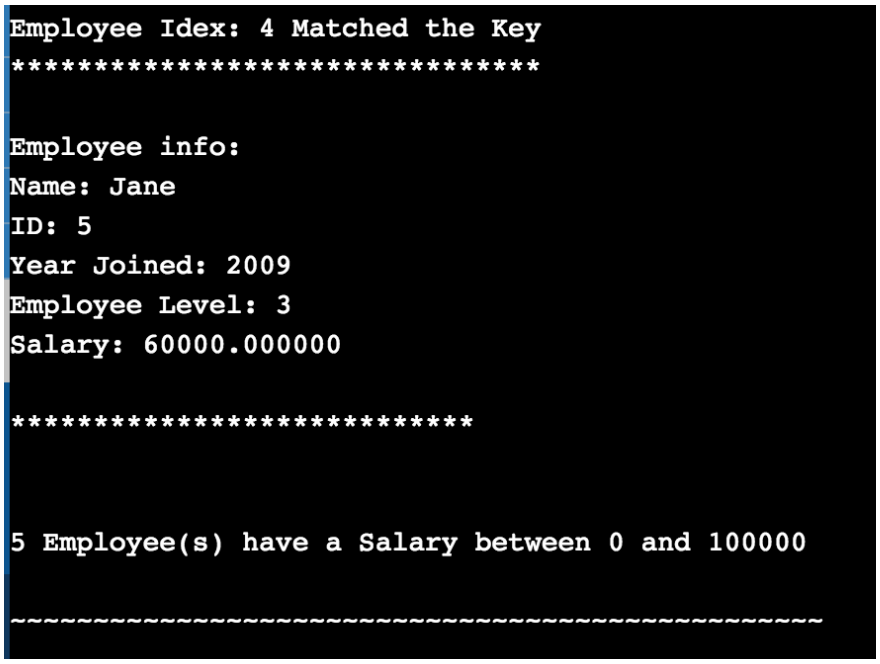
```
    PrintMethod(EmployeeArray, search, count, E1, E2); //calling print Method

}
```

## <u>Program Screenshot Output</u>



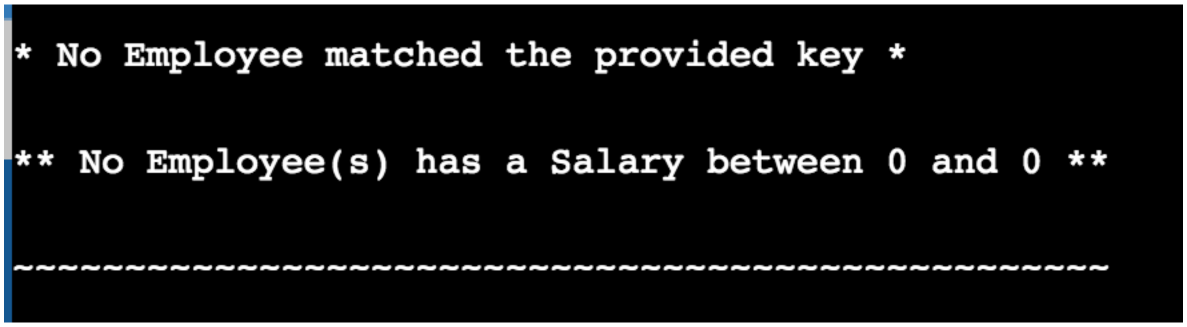
```
Employee Idex: 4 Matched the Key
********************************

Employee info:
Name: Jane
ID: 5
Year Joined: 2009
Employee Level: 3
Salary: 60000.000000

****************************


5 Employee(s) have a Salary between 0 and 100000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


* No Employee matched the provided key *

** No Employee(s) has a Salary between 0 and 0 **

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


* No Employee matched the provided key *

2 Employee(s) have a Salary between 10000 and 50000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


* No Employee matched the provided key *

3 Employee(s) have a Salary between 50000 and 60000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Employee Idex: 2 Matched the Key
********************************

Employee info:
Name: Mary
ID: 3
Year Joined: 2006
Employee Level: 5
Salary: 80000.000000

****************************


1 Employee(s) have a Salary between 75000 and 100000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Employee Idex: 0 Matched the Key
********************************

Employee info:
Name: Bob
ID: 1
Year Joined: 2002
Employee Level: 4
Salary: 60000.000000

****************************


3 Employee(s) have a Salary between 50000 and 60000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- Initial run.
- The screenshot above shows the full output on the initial run.
- The programs runs through data added to demonstrate different tests.



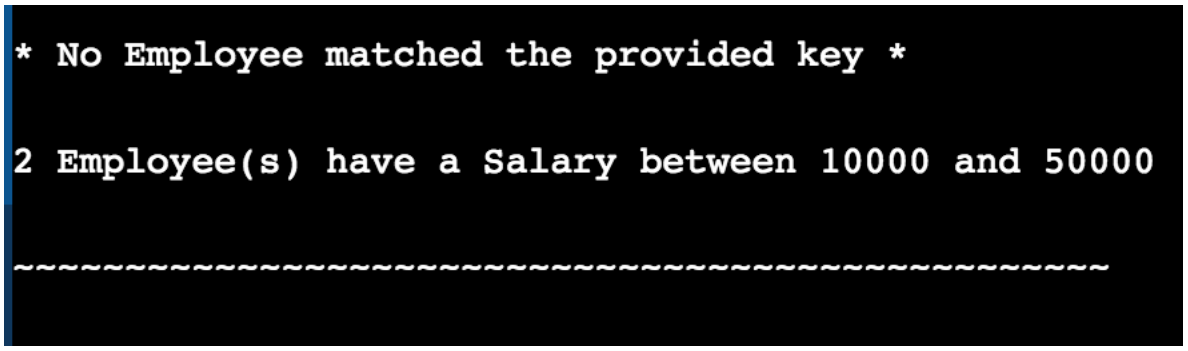
```
Employee Idex: 4 Matched the Key
********************************

Employee info:
Name: Jane
ID: 5
Year Joined: 2009
Employee Level: 3
Salary: 60000.000000

****************************


5 Employee(s) have a Salary between 0 and 100000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- This is the top portion of the output.
- This shows a complete match of the key with index 4 of the array of structures.
- This output also shows 5 Employees have a salary between a range of 0 to 100,000.



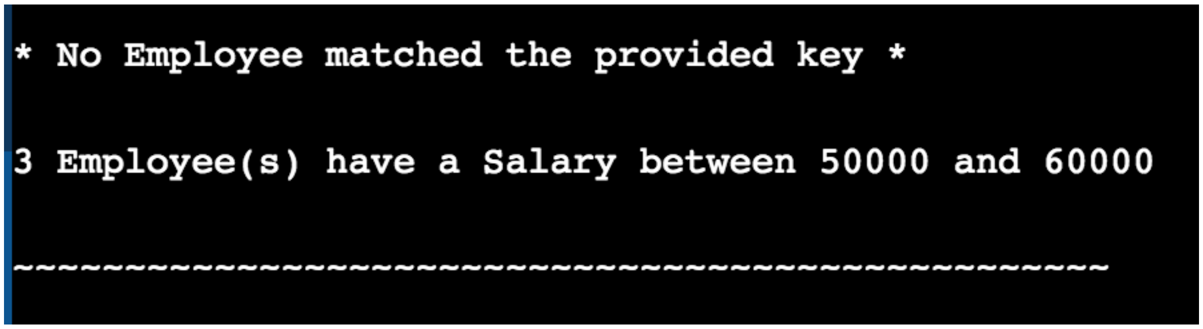
```
* No Employee matched the provided key *

** No Employee(s) has a Salary between 0 and 0 **

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- This is the next output.
- The output shows that no employee has matched the key with all members of the structure.
- The output shows that no Employee has a salary between 0 and 0.
- This range is used to demonstrate the strength of the method.



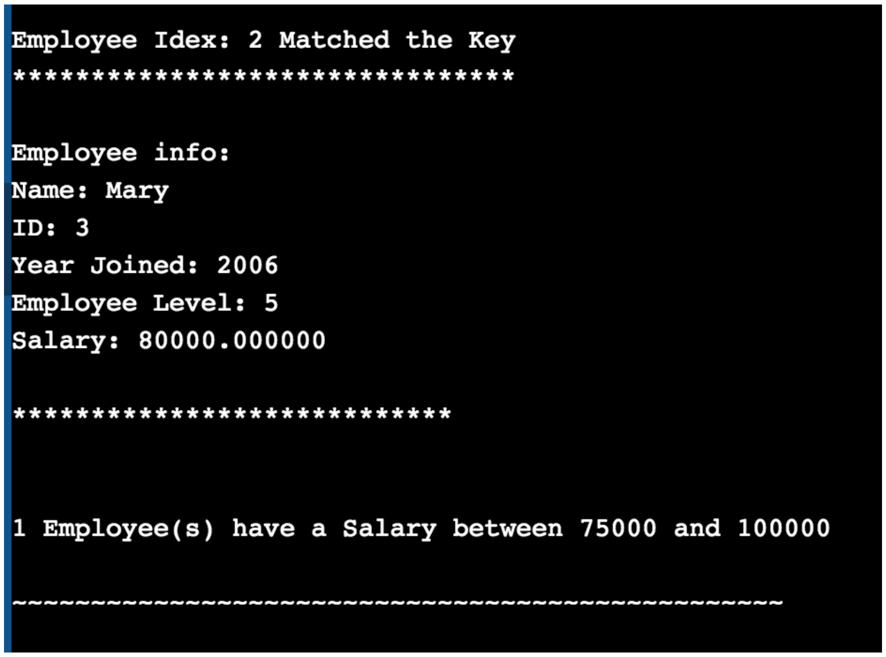
```
* No Employee matched the provided key *

2 Employee(s) have a Salary between 10000 and 50000


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- This is the next output.
- The output shows that no employee has matched the key with all members of the structure.
- The output shows that 2 Employees have a salary between 10,000 and 50,000.
- The range is inclusive on both ends. 10,000 and 50,000 are an acceptable salary within the range



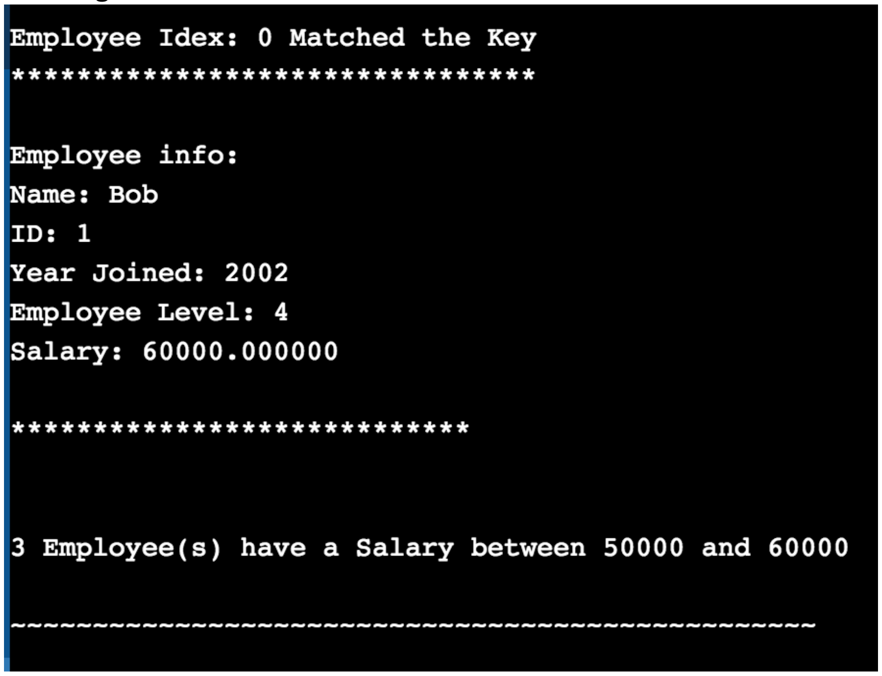
```
* No Employee matched the provided key *

3 Employee(s) have a Salary between 50000 and 60000


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- This is the next output.
- The output shows that no employee has matched the key with all members of the structure.
- The output shows that 3 Employees have a salary between 50,000 and 60,000.
- The range is inclusive on both ends. 50,000 and 60,000 are an acceptable salary within the range



```
Employee Idex: 2 Matched the Key
********************************

Employee info:
Name: Mary
ID: 3
Year Joined: 2006
Employee Level: 5
Salary: 80000.000000

****************************


1 Employee(s) have a Salary between 75000 and 100000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- This is the next output.
- This shows a complete match of the key with index 2 of the array of structures.
- This output also shows 1 Employee has a salary between a range of 75,000 to 100,000.
- The range is inclusive on both ends. 75,000 and 100,000 are an acceptable salary within the range.



```
Employee Idex: 0 Matched the Key
********************************

Employee info:
Name: Bob
ID: 1
Year Joined: 2002
Employee Level: 4
Salary: 60000.000000

****************************


3 Employee(s) have a Salary between 50000 and 60000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

- This is the Final output.
- This shows a complete match of the key with index 0 of the array of structures.
- This output also shows 3 Employees have a salary between a range of 50,000 to 60,000.
- The range is inclusive on both ends. 50,000 and 60,000 are an acceptable salary within the range.

## Program Output

```
Employee Idex: 4 Matched the Key
********************************

Employee info:
Name: Jane
ID: 5
Year Joined: 2009
Employee Level: 3
Salary: 60000.000000

****************************


5 Employee(s) have a Salary between 0 and 100000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


* No Employee matched the provided key *

** No Employee(s) has a Salary between 0 and 0 **

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


* No Employee matched the provided key *

2 Employee(s) have a Salary between 10000 and 50000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


* No Employee matched the provided key *

3 Employee(s) have a Salary between 50000 and 60000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
Employee Idex: 2 Matched the Key
********************************

Employee info:
Name: Mary
ID: 3
Year Joined: 2006
Employee Level: 5
Salary: 80000.000000

****************************


1 Employee(s) have a Salary between 75000 and 100000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Employee Idex: 0 Matched the Key
********************************

Employee info:
Name: Bob
ID: 1
Year Joined: 2002
Employee Level: 4
Salary: 60000.000000

****************************


3 Employee(s) have a Salary between 50000 and 60000

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```