

## Assignment #3

Due: Tuesday Dec 10, 2013 23:59

### Learning Outcome:

Upon successful completion of this assignment, you will have demonstrated the ability to:

- read in large pre-formatted data sets from sequential text files
- use structure types
- use arrays
- write functions
- write data to sequential text files

### Introduction:

You are hired to develop a program to calculate the salespeople's weekly pay for a large chemical company based on the sales information from a sequential text file. The result will be stored in a file (file name: `pay.txt`) in descending order with respect to the weekly pay for each salesperson.

### Calculation rules:

The company pays its salespeople on a commission basis. The salespeople receive base salary \$200 per week **plus** commission of their gross sales for that week. The commission is the **product** of the gross sales and the commission rate for that salesperson. There are five (5) levels of commission rate.

Level	1	2	3	4	5
Rate	2%	4%	6%	8%	10%

For example, a salesperson at level 2 (commission rate: 4%) who sells \$5000 worth of chemicals in a week receives the weekly pay at: \$200 plus 4% of \$5000, or a total of \$400.

### Sales Information:

The sales information for last week is stored in a sequential text file (file name: `sales.dat`). The format of the data is as follows:

```
salesNum:lastName:firstName:salesLevel:salesAmount
```

where:

salesNum: 4-digit integer

lastName: salesperson's last name, no more than 20 characters long.

firstName: salesperson's first name, no more than 20 characters long.

salesLevel: int, commission level for each salesperson

salesAmount: double, the amount the salesperson committed for the week

The five (5) fields are delimited by colon (:).

The sample data may be like the following. However, there may be much more data in the real file. Your program should be able to handle the file with more data.

You can assume there are no more than 1000 salespeople in this company. However, your program should be easy to modify for future company growth with more salespeople.

```
1001:Brendes:Howard:1:750.92
1002:Jervis:Michael:1:8900.23
1003:Chen:Jessica:2:2009.34
1004:Ding:Sheng:2:3475.32
1005:Mok:Anna:3:569.87
1006:Ng:Kelly:5:1000.00
1007:Brown:Maggie:4:7899.32
1008:Frost:Harry:3:3764.34
```

**Result file (file name: pay.txt):**

Weekly Payment Report					
SNum	Last Name	First Name	Lvl	Sales Amt	Week Pay
1007	Brown	Maggie	4	\$7899.32	\$831.95
1008	Frost	Harry	3	\$3764.34	\$425.86
1002	Jervis	Michael	1	\$8900.23	\$378.00
1004	Ding	Sheng	2	\$3475.32	\$339.01
1006	Ng	Kelly	5	\$1000.00	\$300.00
1003	Chen	Jessica	2	\$2009.34	\$280.37
1005	Mok	Anna	3	\$569.87	\$234.19
1001	Brendes	Howard	1	\$750.92	\$215.02

**Note:** the last column on the file is calculated based on the sales information on the file “sales.dat” and the calculation rules. It was sorted in descending order. You can apply bubble sort, selection sort, or other sorting algorithms on this column. In addition, you need to consider the sorting along with the salesperson’s other information, e.g., sNum, LastName, and so on. You can consider all the information as an object (structure type). More description will be available below.

## Specifications:

Your assignment **MUST** include the following functions/structure/array definition. Not only must you code them but you must also make use of them in your program. You are allowed to add more functions if you wish but these should be considered the minimum set.

- **A salesPerson structure type:**

A salesperson is considered as an object with related information defined as a structure type.

The structure type is given as follows:

```
struct salesPerson{
    int salesNum;
    char lastName[31];
    char firstName[31];
    int salesLevel;
    double salesAmount;
    double salesPay;
};
```

This struct should be written in a separate file, named as “sales.h”. In your main c program, you need to include this file as:

#include “sales.h”

- **Commission Level/ Rate:**

```
float level[5] = {0.02, 0.04, 0.06, 0.08, 0.10};
```

Note, this is for the five (5) levels rate. The levels start from 1 to 5. The index of the array starts from 0.

- **FILE\* openRead(char fName[])**

A string file name is passed to this function. The function returns the file pointer if the file can be opened for “read” successfully. It exits the program if the file cannot be opened.

- **FILE\* openWrite(char fName[])**

A string file name is passed to this function. The function returns the file pointer if the file can be opened for “write” successfully. It exits the program if the file cannot be opened.

- **int setArray(FILE \*fp, struct salesPerson sRec[])**

This function goes through each record of the file (sales.dat) and writes the information to the struct array sRec []. The actual number of records in the file is returned.

Parameter (\*fp) is the FILE pointer for the file (sales.dat).

Parameter (sRec []) is the array of struct salesPerson.

Return integer, which stands for the actual number of records in the file (sales.dat). You can use this number to check the end of the array with records afterwards.

Note, this function basically only reads the data from the sequential text file and fill the array of struct salesPerson. At this moment, the member salesPay from struct salesPerson will be empty, which will be calculated in function calcPay() as below.

- **void calcPay(struct salesPerson sRec[], float lvl[], int rNum)**

This function calculates weekly pay for each salesperson based on their level and sales amounts.

Parameter (sRec[]) is the array of struct, which is set in the function setArray().

Parameter (lvl) is commission level/rate array, which will be replaced by the argument in the function call :

```
float level[5] = {0.02, 0.04, 0.06, 0.08, 0.10};
```

Parameter (rNum) is the actual number of records in the file (`sales.dat`). Also, it stands for the actual number of elements that holding valid data in the array sRec[].

- **void sortPay(struct salesPerson sRec[], int recNum)**

This function sorts the array of struct (sRec[]) in descending order based on its one member salesPay (which is calculated in function calcPay())

Parameter (rNum) is the actual number of records in the file (`sales.dat`). Also, it stands for the actual number of elements that holding valid data in the array sRec[].

**Hint:** You can apply any sorting algorithms to sort on the member salesPay of the struct `sRec[j]`. it could be represented as `sRec[j].salesPay`.  
j is the index of the array of struct, which stands for a salesPerson object.

You may need the convenient assignment operation for two struct objects, e.g.,  
struct salesPerson a;  
a = sRec[j];

- **void writePay(struct salesPerson sRec[], int rNum, FILE \*fp)**

This function writes the struct array to result file, with salesPay in descending order (which is sorted in function sortPay()).

Parameter (sRec[]) is the array of struct, which is sorted in the function sortPay().

Parameter (rNum) is the actual number of elements that holding valid data in the array sRec[].

Parameter (\*fp) is the file to write, which is replaced with argument (`pay.txt`) while the function is called.

Note, your program needs to produce the same format for the result file. To make your program portable (display properly on different platforms), make sure there are no more than 80 characters each line in the result report.

## Submission Requirements

Test your program on the C platforms that your instructor has specified. For submission purposes, your program must work on matrix, our Linux cluster. Submit the typescript on Blackboard.

## **STUDENT OATH**

A signed declaration of honesty must be included in your assignment. Put the following declaration in comment in your C program.

I declare that the attached assignment is wholly my own work in accordance with Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) or distributed to other students.

Name \_\_\_\_\_ Student ID \_\_\_\_\_

## **GRADING**

This assignment is worth 9% of your final grade.