Programming Fundamental

Assignment - 06

Instructions:

- It is individual assignment, so try to do it by yourself.
- Assignment should be in zip file named as RollNumber_Name_PF.zip and name of your file should be according to tasks given in this assignment.
- Submit your Assignment on google classroom before October 31, 2022**, 11:59pm**. Late submissions will not be considered.
- The total Marks are 100. Question 1-5 (12 marks each), question 6 & 7 (20 marks each)
- Feel free to ask any question.

Task# 01:

A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

```
Enter your area code, exchange, and number: 415 555 1212
My number is (212) 767-8900
Your number is (415) 555-1212
```

Task# 02:

A point on the two-dimensional plane can be represented by two numbers: an x coordinate and a y coordinate. For example, (4,5) represents a point 4 units to the right of the vertical axis, and 5 units up from the horizontal axis. The sum of two points can be defined as a new point whose x coordinate is the sum of the x coordinates of the two points, and whose y coordinate is the sum of the y coordinates.

Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:

```
Enter coordinates for p1: 3 4
Enter coordinates for p2: 5 7
Coordinates of p1+p2 are: 8, 11
```

Task# 03:

Create a structure called employee that contains two members: an employee number (type int) and the employee's compensation (in dollars; type float). Ask the user to fill in this data for three employees, store it in three variables of type struct employee, and then display the information for each employee.

Task# 04:

Create a structure called time. Its three members, all type int, should be called hours, minutes, and seconds. Write a program that prompts the user to enter a time value in hours, minutes, and seconds. This can be in 12:59:59 format, or each number can be entered at a separate prompt ("Enter hours:", and so forth). The program should then store the time in a variable of type struct time, and finally print out the total number of seconds represented by this time value:

```
long totalsecs = t1.hours*3600 + t1.minutes*60 + t1.seconds
```

Task# 05:

Use the time structure from Task#4, and write a program that obtains two time values from the user in 12:59:59 format, stores them in struct time variables, converts each one to seconds (type int), adds these quantities, converts the result back to hours-minutes-seconds, stores the result in a time structure, and finally displays the result in 12:59:59 format.

Task# 06:

Write a program that uses a structure to store the following weather data for a particular month:

```
Total Rainfall
High Temperature
Low Temperature
Average Temperature
```

The program should have an array of 12 structures to hold weather data for an entire year. When the program runs, it should ask the user to enter data for each month. (The average temperature should be calculated.) Once the data are entered for all the months, the program should calculate and display the average monthly rainfall, the total rainfall for the year, the highest and lowest temperatures for the year (and the months they occurred in), and the average of all the monthly average temperatures.

Input Validation: Only accept temperatures within the range between –100 and +140 degrees Fahrenheit.

Task# 07:

Write a program that uses a structure to store the following data about a customer account:
Name
Address
City, State, and ZIP
Telephone Number
Account Balance
Date of Last Payment

The program should use an array of at least 10 structures. It should let the user enter data into the array, change the contents of any element, and display all the data stored in the array. The program should have a menu-driven user interface.

Also add the code that allows the user to search the structure array for a particular customer's account. It should take input a part of the customer's name as an input and then search for an account with a name that matches it. All accounts that match should be displayed. If no account matches, a message saying so should be displayed.

Input Validation: When the data for a new account is entered, be sure the user enters data for all the fields. No negative account balances should be entered.