APS 105 — Computer Fundamentals

Lab #2: More Complex Calculations

Winter 2022

You must use `examify.ca` to electronically submit your solution by 11:59 pm on **Saturday, January 29, 2022**.

Objective

In this lab you will be writing three C programs. They require knowledge of if-statements and math library functions. Input and output must be done using `scanf` and `printf`.

In the sample output examples that follow, the text that would be entered by the program user is displayed using a **bold** font. The text <**enter**> or <**space**> stands for the user pressing the `enter` or `space` bar key on the keyboard. When you execute your programs, the user's text will not be displayed in bold and <**enter**> will not be shown.

In Visual Studio Code, place your solution for each part in a separate directory, so that Visual Studio Code will not attempt to compile multiple files and link them together.

Part 1 — Monthly Installment Calculator

You decide to purchase a car, and you will need a loan to purchase your car. You will need a small down payment up front and pay the rest (with some interest) by monthly payments over a fixed term (ranging from a few months to several years). Given a purchase price P (in dollars), a down payment D (in dollars), a finance term n (in months), and the monthly interest rate r, the monthly payment M (in dollars) that you will be required to pay over n months is calculated using the following formula:

$$M = \frac{(P - D) \times r \times (1 + r)^n}{(1 + r)^n - 1}$$

For example, if the monthly interest rate is 0.4%, a customer can buy a $\$21000$ car with just $\$1000$ down payment and around $\$458.78$ per month for 48 months. Since you want to study different scenarios, write a `C` program that will assist you to quickly calculate the monthly payments you will have to make. Your program reads in the purchase price of that fancy car (an integer value), the amount of down payment (an integer value), the finance term (an integer value representing the number of months), the monthly interest rate (a real number representing the monthly interest percentage), and then prints the monthly payment rounded to 2 decimal places. You may assume that inputs are valid numbers and the finance term and the interest rate are not zero.
Here is a sample output from an execution of the program:

```
Enter the purchase price P: 21000<enter>
Enter the amount of down payment D: 1000<enter>
Enter the finance term (in months): 48<enter>
Enter the monthly interest rate (in percent): 0.4<enter>

The monthly payment is: 458.78
```

Note: To compute x^y, use the `pow` function in the `math.h` library. To use the math.h library, you will need to add the following line to the beginning of your program:

```
#include <math.h>
```

Part 2 — Diciphering a code

You want to use a 4-digit combination lock on a safe where you put your belongings when you go skiing. As a matter of good practice, you want to change the combination every time you go. Since you do not want to worry about people finding out what your combination is, you write it on a paper using a coding scheme. If anyone reads your piece of paper, they will still not know your combination.
The scheme you use takes the real combination and swaps the 1st and the 4th digit of the combination, and replaces each of the 2nd and 3rd digits by their 9's complement. In other words, a combination of the form `abcd` is encoded as `d(9-b)(9-c)a`.
For example, if the actual combination is 0428, the encrypted combination will be 8570 (note that 0 and 8 are swapped, 4 is replaced by 9-4, and 2 is replaced by 9-2). The nice thing about his coding scheme is that you can apply it again on the encrypted combination to calculate the real one.
Since you are now taking APS 105, your task is to write a `C` program to implement this code-decode scheme. The user (you!) will, then, enter an encrypted combination and the program will output the real combination. A sample output from an execution of the program appears below:

```
Enter an encrypted 4-digit combination:  8021<enter>
The real combination is:  1978
```

Note: You may assume that the entered combination is a valid 4-digit integer number. When reading the 4-digit combination from user input, you are not allowed to scan in individual characters; instead, you should scan in a single integer using scanf.

Part 3 — Arrival Time Calculator

Nidaa, Issac and Amy are going on a road trip from Chelsea, Quebec to Mississauga, Ontario. They used Ammar's application that estimates the trip time in hours. Peter wanted to know their arrival time as he will wait in Mississauga to greet them. However, Ammar's application does not provide them with the arrival time. You, taking APS 105 – like everyone in this question, decided to write a `C` program that takes in the current time in hours and minutes, and the trip time in hours from Ammar's application. Your code should tell them if they will arrive the same day or next day and their arrival time in hours and minutes rounded to the **lowest** minute. You can safely assume that the trip time is lower than 24 hours, i.e. maximum trip time is 23:59. A sample output from an execution of the program appears below:

```
Enter current time: 15<space>30<enter>
Enter trip time: 3.1<enter>

Current time is 15:30
Arrival Time is same day 18:36
```

Marking

This lab will be marked out of 10, 2 marks for Part 1, 4 marks for Part 2, and 4 marks for Part 3. The automarker may use multiple test cases in each part, and if this is the case, the marks for this part will be evenly split between the test cases. The test cases used for marking may or may not be the same as the test cases that are made available to you. The deadline is strictly enforced **(11:59 pm on Saturday, January 29, 2022)**, so avoid last minute submissions.