Test 02 – CSD 1113 (Program Logic)

Overview:

You have been hired as a developer for one of the largest Banks in Canada. Your first task as a newly minted software developer is to develop a menu-driven application to process a list of transactions read in from a CSV file. The *main* program has been written, but the sub-procedures it invokes are incomplete.

Set-up (REQUIRED):

1. Download and Open the Test02.rap file from week 15 content
2. Use RAPTOR's **saveAs** feature and append your c# to the start of the filename
3. Familiarize yourself with the file and its procedures

Evaluation:

Test sums to 50 marks.

How-to-read Instructions:

The 7 questions that follow are designed to be completed in any order except for Q1. This should allow you to choose the questions you are most comfortable with first to build confidence. Each question corresponds to one procedure in the Test02.rap file provided. Additional comments are provided in each of the RAPTOR procedures.

Question 01 – printMenu (5 marks)

This procedure has no input and returns an integer between 1 and 5 representing the user's menu selection. The menu displayed should be like the following:

1. Calculate account balance
2. Add transaction
3. Find transaction(s) based on date
4. Find max debit or credit
5. Exit application

The procedure should then prompt the user for a selection.

Note: If an invalid selection is made the procedure should re-prompt the user until a valid selection is made.

Question 02 – readDataFromFile (8 marks)

This procedure has no input and returns a 2D array named **data** filled with the data.csv file's contents. The function must also return **length_of_array** populated with number of rows in the 2D array **data**. The format of each record data.csv is as follows:

- Date (dd-mm-yyyy)

- "DEBIT" or "CREDIT"
- Dollar amount of transaction

Your job is to read each record into a 2D array named **data**. The returned array should have rows in the format: [Date, DEBIT_CREDIT, Amount].

⇒ Date, DEBIT_CREDIT: strings
⇒ Amount: float

HINT: You may find the To_Integer(), To_Float(), and Get_Nth_String() procedures useful.

Question 03 – calculateAccountBalance (7 marks)

This procedure takes the 2D array **data** and the integer **length_of_array** as input and returns the account **balance** as a float.
IMPORTANT: DEBIT is an addition and CREDIT is a subtraction.

Question 04 – addTransaction (8 marks)

This procedure appends a transaction to end of the 2D array **data** and returns the updated array as well as the updated **length_of_array** value. The user should receive a prompt to enter the day of the month (0-31), "DEBIT" or "CREDIT", and the amount of the transaction.
REMEMBER: Rows in the 2D array are in the format [Date, DEBIT_CREDIT, Amount] where Date is represented in *dd-mm-yyyy* as a string.

Note: All inputs inside the procedure must be validated for correctness. You may assume the month is December and the year 2020.

Question 05 – findTransactionsOnDate (9 marks)

This procedure searches **data** array for all transactions on a that occurred on a specific date and returns a 2D array named **matches2DArray** with those records as its rows. The user should receive a prompt to enter the day of the month (1-31). Assume the month is December and the year 2020. You also need to return the number of rows in the **matches2DArray** as **length_of_matches_array**.

REMEMBER: Rows in the 2D array are in the format [Date, DEBIT_CREDIT, Amount] where Date is represented in *dd-mm-yyyy* as a string.

Note: All inputs inside the procedure must be validated for correctness. You may assume the month is December and the year 2020. Therefore, validate that day is correctly provided.

Question 06 – findMax (9 marks)

This procedure searches **data** array for the MAXIMUM "DEBIT" or "CREDIT" transactions. The decision to search for "DEBIT" or "CREDIT" value is provided by the **choice** parameter.
You also need to return a 1D array **maximum_Transaction** containing the maximum transaction information (i.e. [Date, DEBIT_CREDIT, Amount]) for "DEBIT" or "CREDIT" based on **choice**.

IMPORTANT: If there is a tie for the maximum transaction select any of these transactions of your choosing. Additionally, if no results are found return the string "ERROR: No DEBIT/CREDIT entries available" instead of a 1D array.

REMEMBER: Rows in the 2D array are in the format [Date, DEBIT_CREDIT, Amount].

Question 07 – writeDateToFile (4 marks)

This procedure takes the 2D array **data** as and the integer **length_of_array** input and writes each row in the array into the **data.csv** file. This procedure has no outputs.

******Congratulations you have reached the end of Test 02*******