Concordia University
Computer Science & Software Engineering
Faculty of Engineering and Computer Science

Concordia University
Department of Computer Science and Software Engineering
COMP 348: Principles of Programming Languages
Winter 2019

Assignment 3
Evaluation: 100 pts
(5% of your final grade)
Due date and time: Before Friday April 12, 2019 at 23:59

Part 1: Object-oriented programming with Ruby

Question 1 (10 pts)

Write a Ruby method that takes an array of strings as an argument, sorts it alphabetically, then iterates and uses a code block to display each string element which is a palindrome word, along with its characters count. For example, if we pass the following argument array ["adam","noon","john","rotator","level","sara"], then the output should be:

```
List of palindrome words:
level, n_of_char: 5
noon, n_of_char: 4
rotator, n_of_char: 7
```

Question 2 (20 pts)

Automated Readability Index (ARI) is used for testing readability in English text.
The mathematical formula for calculating ARI in a text document is as follows:

$$ARI = 4.71 \times (characters/word) + 0.5 \times (words/sentence) - 21.43$$

Where:

$$characters = \Sigma letters, numbers\ and\ punctuation$$
$$words = \Sigma spaces$$
$$sentences = \Sigma full\ stops$$

The following table shows the educational grade level that corresponds to each ARI score:

Score	Age	Grade Level
1	5-6	Kindergarten
2	6-7	First/Second Grade
3	7-9	Third Grade
4	9-10	Fourth Grade
5	10-11	Fifth Grade
6	11-12	Sixth Grade
7	12-13	Seventh Grade
8	13-14	Eighth Grade
9	14-15	Ninth Grade
10	15-16	Tenth Grade
11	16-17	Eleventh Grade
12	17-18	Twelfth grade
13	18-24	College student
14	24+	Professor

Write a Ruby method that can read any document file, count number of characters, words, and sentences and then apply the ARI formula to find out the grade level required for a person to read the opened document. For example, for the following "paragraph.txt" file:

After the Lord Stanley of Preston was appointed by Queen Victoria as Governor General of Canada on June 11, 1888, he and his family became highly enthusiastic about ice hockey. Stanley was first exposed to the game at Montreal's 1889 Winter Carnival, where he saw the Montreal Victorias play the Montreal Hockey Club. The Montreal Gazette reported that he "expressed his great delight with the game of hockey and the expertise of the players". During that time, organized ice hockey in Canada was still in its infancy and only Montreal and Ottawa had anything resembling leagues.

A Ruby method call: **calcARI**("paragraph.txt") should output the following:

Total # of characters: 474
Total # of words: 96
Total # of sentences: 4
Automated Readability Index: 13.8
Grade level: 18-24 (College student)

Note: For the above example, if you consider the punctuations as character as well, you may achieve 579 characters and the grade level of "Professor" which is still accepted.

Question 3 (30 pts)

In this question you create an electronic devices inventory catalogue using Ruby. Your program should read an electronic devices listings file which contains the information of all the electronic devices along with their features. Each line contains the following 9 specifications:

Category, Battery life, Model number, color, Manufacturer, Status, Year Built, Price, Features

Example of listing lines:

```
12hrs,Smartphone,18131A,used,Apple,600$,{Bluetooth,Water
resistant,finger print reader,16GB},white,2016

Used,Smartwatch,Samsung,{activity tracker,
Bluetooth,water resistant},2017,250$,black,3947t4f,9hrs

Apple,Laptop,silver,809F435RW,{Face ID,Retina display,
Core-i5},new,2019,1800$,18hrs
```

***Note*:** order of listing features varies from one line to another as it was entered by employees not trained for proper data entry. Also, assume the following specifications possible values (case insensitive):

Specifications List	**Values**
Category	{Smartphone, Tablet, Laptop, Smartwatch}
Battery life	Number of hours followed by "hrs"
Model Number	combination of letters and numbers not ending with "hrs"
Color	{silver, white, black, burgundy, blue}
Manufacturer	{Apple, Samsung, Google, Lenovo,LG}
Status	{used, new, refurbished}
Year Built	Any Year
Price	Any Number followed by $
Features	Any set of features inside a curly bracket

Your program should have three methods as follows:

1. **convertListingsToCatalougue**: This method
   - Reads the listing file line by line
   - Recognizes and extracts different listing features
   - Instantiates appropriate objects of different classes and subclasses that you can define to hold the listing information for different manufacturer. The order of features for each object should be according to the above table. Any parent class should have properties that count the number of listings of that manufacturer or that model along with other common behavior or properties. In the listings examples above, you can create an object of class (device-category) and store all the listing information in that class and also update the parent (device_manufacturer) common class properties accordingly.

2. **AddToInventory**: a method that accepts a new listing as a single line of unordered listing specifications, add the line to the original listing file and add an appropriate object to the catalogue based on the listing specifications.

3. **saveCatalogueToFile**: a method that traverses all created catalogue objects and stores them to an output alphabetically according to their **manufacture name**. Each listing features will follow a strict fixed order as opposed to the original file random order. The order is same as specifications listed in the above table. For example, the original three listings will look as follows in the output file:

```
Smartphone,12hrs,18131A, white,Apple,used,2016,
600$,{Bluetooth,Water resistant,finger print reader,16GB}

Laptop,18hrs,809F435RW,silver,Apple,new,2019,1800$, {Face
ID, Retina display, Core-i5}

Smartwatch,9hrs,3947t4f,black,Samsung,Used,
2017,250$,{activity tracker,Bluetooth,water resistant}
```

Part 2: Procedural programming with C:

Question 4 (20 pts)

Write a C function ***matrixTranspose*** that takes a two-dimensional array as its input argument then transposes its elements and prints the results.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

Develop the ***matrixTranspose*** method with two different methods:

1. Assume the matrix is a square matrix. This will make finding the transpose a simple swapping of the arrays' elements and it can be done in place.

2. Generalize your function to work with any NxM matrix where N≠M. You will need to properly handle dynamic memory allocation for creating the transpose matrix inside the function.

Question 5 (20 pts)

a) Write a C program that: creates and prints out a linked list of strings. Similar to java your program should include the node struct. Test your program for different lists of string.

b) Add the following functions to the program above:

- `void add_after(node_ptr &list, char a_word[], char word_after[])`

which inserts a node containing `"word_after"` in the linked list `"list"`, after the first occurrence of a node containing `"a_word"`. If `"list"` does not contain such a node, the function leaves it unchanged.

- `void delete_node(node_ptr &a_list, char a_word[])`

which deletes the first node in "a_list" which contains "a_word".

Sample input/output might be:

```
Enter first word (or '.' to end list): though
Enter next word (or '.' to end list): actions

Enter next word (or '.' to end list): speak
Enter next word (or '.' to end list): louder
Enter next word (or '.' to end list): than
Enter next word (or '.' to end list): words
Enter next word (or '.' to end list): .

THE LIST IS NOW:
though actions speak louder than words

AFTER WHICH WORD WOULD YOU LIKE TO ADD AN EXTRA WORD? though
WHICH WORD WOULD YOU LIKE TO ADD? many

THE LIST IS NOW:
though many actions speak louder than words
```

```
WHICH WORD WOULD YOU LIKE TO DELETE? Than

THE LIST IS NOW:
though many actions speak louder words
```

Submission:

- **Assignment must be done individually (no groups are permitted).**
  - Create one zip file, containing all files for your assignment.
  - Name your zip file this way:*a3_studentID*, where *studentID* is your ID number, for the second assignment, student 123456 would submit a zip file named a3_123456.zip
- Assignments must be submitted through Moodle.