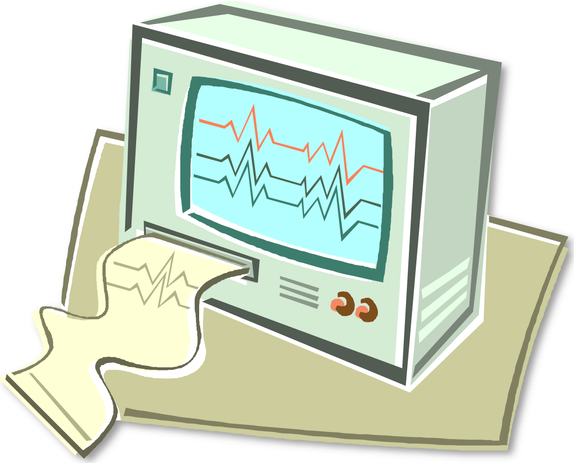
California State University, Sacramento
College of Engineering and Computer Science

Computer Science 130: Data Structures and Algorithm Analysis

Assignment #2 – Radix Sort

Overview

For this assignment, you will implement one of the most clever sorting algorithms of all time – Herman Hollerith's Radix Sort.



To properly test your program, I'm going to provide some data files. The format of each file will be the same.

Also, as not to make this assignment too difficult, all the keys will be base-10 numbers. In other words, you can assume their will be 10 buckets in your sort.

Part 1: Key-Value Class

Entry Class

To store the values in the array, we need to create a key-value class. In this case, I called it "Entry", but you are welcome any other name that you like.

```
class Entry
   public String key
   public String value
end class
```

Note, this is far different from the Node class used by the linked list. That class shouldn't be modified.

Your Updated Queue Class

So, how do you implement the buckets? Well, they are essentially queues.

The following is the modified interface for the Queue Class. You will only have to make some minor changes to your Assignment #1 code. Don't have to make any modifications to your Linked List class.

public class Queue		
Item	About()	Returns text about you – the author of this class.
void	Enqueue(Entry)	Enqueues a string onto the queue.
Entry	Dequeue()	Dequeues (removes) a string from the front of the queue.
Entry	Peek()	Returns the value on the front of the queue. Do not return the linked list node.
boolean	IsEmpty()	Returns true of the stack is empty.

Part 2: Input File Format

A number of test files will be provided to you for testing your code. The format is designed to be easy to read in multiple programming languages.

The first line of the data contains the total digits in the key. You should save this value and use it to control the number of passes the Radix Sort will perform. Naturally, this can also be inferred from the data itself, but its useful to have it explicitly stated.

The second line consists of the number of records (i.e., succeeding lines) in the file. You can use this variable to initialize an array. Note: Radix Sort can be performed on a Linked-List, but we didn't cover this in class. Please use whichever approach is easier (arrays are easier to understand).

```
Digits in the key
Number of records
Key 1, Value 1
Key 2, Value 2
Key 3, Value 3
...
Key n, Value n
```

The following is one of the most basic test files on the CSc 35 Website. It consists of only 13 records.

File: years.txt

```
4
13
2020,Great Toilet Paper Shortage
1839,Sutter's Fort founded
1846,Bear Flag Revolt
1947,Sacramento State founded
1977,Star Wars was born
2017,Star Wars franchise almost destroyed
1964,Buffalo wings invented
1783,United States Constitution enacted
1850,California joins the United States
1848,Gold Rush begins
1776,American Revolution
1980,Pacman was released
1953,Cheese Whiz was invented (Nacho Era began)
```

13 records

Requirements

⚠ **You must use your linked-list and queue (modified) from Assignment #1.**

Do not use any built-in queue library class, etc… If you do, you will receive a zero.
No exceptions. No resubmissions.

- This **must** be completely all your code. If you share your solution with another student or re-use code from another class, you will receive a zero.

- You **must** use your Linked-List from Assignment #1.

- Your Queue must use your Linked-List class.

- You may use any programming language you are comfortable with. I strongly recommend not using C (C++, Java, C#, Visual Basic are all good choices).

Due Date

Due **October 19, 2021** by 11:59 pm.

Given you already have developed excellent programming skills in CSc 20, this shouldn't be a difficult assignment. **Do not send it to canvas.**

E-Mail the following to dcook@csus.edu:

- The source code.

- The main program that runs the tests.

- Output generated by your tests

⚠ **The e-mail server will delete all attachments that have file extensions it deems dangerous. This includes .py, .exe, and many more.**

So, please send a ZIP File containing all your files.