### Radix Sort ( Due 02 Nov 2020 )

In this assignment you will modify the Radix Sort algorithm so that it sorts strings that have a mix of **lower case letters and digits** - similar to your UT EIDs.

In this version of Radix Sort we use Queues very naturally. Let us consider the following set of positive integers:

311, 96, 495, 137, 158, 84, 145, 63

We will sort these numbers with three passes. The number of passes is dependent on the number of digits of the largest number - in this case it is 495.

In the first pass we will go through and sort the numbers according to the digits in the units place (or the right most digit) and place the numbers in 10 queues for each of the digits 0 through 9. After this has been done we will dequeue the numbers in all the queues and enqueue them in an eleventh queue.

In the second pass, we will go through and sort the numbers according to the digits in the tens place and enqueue the numbers in the 10 queues. After this has been done, we will dequeue the numbers in all the queues and enqueue them in the eleventh queue - and so on.

Here are what the passes will look like:

Pass 0:

Q10: 311, 96, 495, 137, 158, 84, 145, 63

Pass 1:

Q0:

Q1: 311

Q2:

Q3: 63

Q4: 84

Q5: 495, 145

Q6: 96

Q7: 137

Q8: 158

Q9:

Q10: 311, 63, 84, 495, 145, 96, 137, 158

Pass 2:

Q0:

Q1: 311

Q2:

Q3: 137

Q4: 145

Q5: 158

Q6: 63

Q7:

Q8: 84

Q9: 495, 96

Q10: 311, 137, 145, 158, 63, 84, 495, 96

Pass 3: Assume that the 2-digit numbers have a leading zero

Q0: 63, 84, 96

Q1: 137, 145, 158

Q2:

Q3: 311

Q4: 495

Q5:

Q6:

Q7:

Q8:

Q9:

Q10: 63, 84, 96, 137, 145, 158, 311, 495

Now dequeue Q10 and the numbers are sorted.

For this assignment you will modify this algorithm so that it will take strings as input that have **lower case letters and digits**. Assume that the input is correct. Remember that according to the ASCII table digits come before letters.

Here are some recommendations that we would like to make. You do not have to follow our recommendations.

* Instead of naming the queues Q0, Q1, and so on. Create a list and keep your Queue objects there.
* Maintain a dictionary where the key is a character (either a digit or a lower case letter) and the value is an index in the above list.

**Input:**  
You will be given a file [radix.in](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/radix.in) in the following format. The first line in the input will be a single integer *n* that states the number of strings to follow. This will be followed by *n* lines where there will be a single string per line. Each string will have between 1 and 10 characters inclusive. The characters will be either the digits 0 through 9 or the letters 'a' through 'z'.

**Output:**  
You will [output a list](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/radix.out)having the strings in sorted order.

For this assignment you may work with a partner. Both of you must read the paper on [Pair Programming](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/PairProg-CACM-1999.pdf) and abide by the ground rules as stated in that paper. If you are working with a partner submit only one program but have both your names and eids in the header. If you are working by yourself remove the two lines regarding your partner.

Here is the template of the file called [Radix.py](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/Radix.py) that you will be turning in. We are looking for clean and structured design using the [standard coding conventions](https://www.python.org/dev/peps/pep-0008/)in Python. You may not change the function signatures but you may add as many helper functions as needed. The file will have a header of the following form:

# File: Radix.py

# Description:

# Student Name:

# Student UT EID:

# Partner Name:

# Partner UT EID:

# Course Name: CS 313E

# Unique Number:

# Date Created:

# Date Last Modified:

To run this code on the command line on the Mac you will do

python3 Radix.py < radix.in

To run the code on the command line on a Windows machine you will do

python Radix.py < radix.in

Use the [Canvas](http://canvas.utexas.edu/)system to submit your **Radix.py** file. We should receive your work by 11 PM on Monday, 02 Nov 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

* Your Python program should have the proper header.
* Your code must run before submission.
* You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.