CSE 30
Programming Abstractions: Python
Programming Assignment 5

In this project you will augment the *List* class discussed in lecture. This class is based on an underlying data structure called a *linked list*, to be discussed at length in lecture. It is intended to emulate (to some extent) the *list* class built into the Python language. The partially constructed List class can be found at

https://classes.soe.ucsc.edu/cse030/Spring21/Examples/Classes/list.py

and contains implementations of the following 11 functions.

```
__init__(self, s=None)
   """Initialize self, a List object."""

__len__(self)
   """Return the length of self."""

__str__(self)
   """Return a string representation of self."""

__repr__(self)
   """Return a detailed string representation of self."""

__iter__(self)
   """Return an iterator over self."""

__eq__(self, other)
   """
   Return True if self and other are the same sequence, False otherwise.
   """

append(self, x)
   """Add item x to back of List."""

clear(self)
   """Delete all items from List."""

copy(self)
   """Return a (shallow) copy of List."""

insert(self, i, x)
   """Add item x at position i of List, where -n<=i<=n and n=len(self)."""

pop(self, i=-1)
   """
   Delete item at position i of List, where -n<=i<=(n-1) and n=len(self).
   """
```

You will augment this class by implementing the 8 functions given below.

```
remove(self, x)
   """
   Delete leftmost occurrence of x in List. Raise ValueError if x is not
   contained in self.
   """

reverse(self)
   """Reverse the items of List."""

__getitem__(self, i)
   """
   Return item at position i of self, where -n<=i<=n-1 and n=len(self).
   """

__setitem__(self, i, x)
   """
   Overwrite item at position i of self by x, where -n<=i<=n-1 and
   n=len(self).
   """

__add__(self, other)
   """
   Return the concatenation of self with other. This function implements
   the operation self + other.
   """

__iadd__(self, other)
   """
   Replace self by the concatenation of self with other. This function
   implements the operation self += other.
   """

__mul__(self, n)
   """
   Return the n-fold concatenation of self with self, where n>=0. This
   function implements the operation self*n.
   """

__rmul__(self, n)
   """
   Return the n-fold concatenation of self with self, where n>=0, reversing
   the order of self and n. This function implements the operation n*self.
   """
```

A template for the augmented list can be found at.

https://classes.soe.ucsc.edu/cse030/Spring21/Examples/pa5/augmented_list_template.py

Change the name of this file to list.py and fill in the required function definitions. To complete this assignment, you must carefully study and fully understand how the given 11 functions create, maintain and operate on a linked list data structure. To do this, it is necessary that you draw, for each function, abstract pictures of memory illustrating the special cases that arise, as was done in lecture. Draw similar pictures as you write the remaining 8 functions. Test each function thoroughly by calling it from within a program you called TestList.py. This program is not required to do any particular thing, but should convince the graders that you have exercised every logical pathway and verified all special cases for each required function. Testing is not an afterthought, and should be done concurrently with coding.

Submit both files list.py and TestList.py to Gradescope before the deadline. As usual it is important to start early and get help as issues arise.