NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
PESHAWAR CAMPUS

Problem Set:	Assignment: A04	**Semester:**	Fall 2019
Points:	*10*		
Date Set:	*See Autograder*	**Due Date:**	*See Autograder*
Course:	CS218 - Data Structures	**Instructor:**	Dr. Nauman

1 Circular Linked List

Here is a description of what you need to do:

Create the circular linked list class as we did during our lecture. All the operations should be implemented including the ones did in lab: `__str__`, `_get_last`, `insert`, `remove` and `remove_at`. You can start with this assignment and only leave the remove_at function for after the lab when it will be covered (or just code it yourself).

Make sure your code is bug free – even if we had a bug in the code we discussed in class! After that, add methods in the class to support the following new operations:

1.1 Length

Add a function called `len` that returns the length of the list i.e. the number of elements in the list. The logic is simple: loop over the whole list and keep track of a counter. At the end, return the counter. You might want to separate the cases in which the list has just one element versus when it has more than one.

1.2 Index-based Retrieval

Add a function called `get` that takes one parameter – an index – and returns the value at that index. For instance, if we have a list `lst`:

```
[1, 2, 5, 4, 2]
```

and we call `lst.get(2)`, it should return 5. If the function is called on a list which has no elements, the function should *raise* an `IndexError` type exception. For other indices, it should just keep going over the list until that index is found *in a round-robin fashion*. For example, for the above list, if we call `lst.get(7)`, it should return 5.

The logic for the retrieval is again quite simple: loop over the whole list and keep track of a counter. When the counter reaches the desired value, simply return the value at that position.

1.3 Push

Once you have the insert and length functions, push is easy. You just need to call the insert function and pass the value for position based on the length of the list.

1.4 Pop

Similarly, when you have the get, len and remove_at functions, pop is easy. Just get the value at the last element, remove this last element based and return the value.

2 Submission

Use `python run.py local` to ensure all tests are passing and then submit your assignment using `python run.py remote`

If you wish to request an extension, use the autograder UI to do so. Each student gets a maximum of 3 extension days per semester.