CS 241 DATA STRUCTURES SPRING 2022 MIDTERM I (PRACTICE)
Instructor Calvin Deutschbein

Roster Name	
Sign here to affirm the Honor Code	

This exam will be timed to take 60 Minutes.

It will be scored out of 200 Points.

It will make up 20% of Final Grade.

SECTION I: PYTHON **40 Points**

Part 1: Multiple Choice: *4 Questions @ 5 Points each =* *20 Points*

Which of the following is a way to combine multiple elements of data into a single object?

A. Using multiple variables to save multiple values
B. Using higher order function objects like cons(...) that return multiple values
C. Using print statements to output multiple values to console
D. Converting multiple values into the same type, e.g. into strings

Which of the following would allow use of methods from within sortable.py (as with HW0)?

A. >>> import sortList()
B. >>> from * import sortList
C. >>> import sortList.py
D. >>> from sortList import *

Which of the following may never be of type None (assuming a HW1 SortList)?

A. self.data
B. self.next
C. self.index(...)
D. self.size()

Which of the following is a function and **NOT** a method of a class?

A. c3ns(...)
B. __add__()
C. SortList.size()
D. SortList.next

Part 2: Short Response: 2 Questions @ 10 Points each = 20 Points

Create a function l_o_n_g() that accepts one argument, a string s, prints the string, and returns a new string containing all of the characters in s interspersed with underscores.

Create a function isFurther that takes two “points”, each given as an x and y value, for a total of four arguments (x1,y1,x2,y2) and returns True if (x1,y1) is further from the origin (0,0) than (x2,y2), and False otherwise.

SECTION II: CONDITIONALS **40 Points**

Part 3: True/False: *4 Questions @ 5 Points each =* *20 Points*

Specify whether the code snippets return boolean values True or False.

```
>>> x = False
>>> x == True
```

A. True
B. False

```
>>> false = True
>>> False
```

A. True
B. False

```
>>> if None:
. . .       True
. . .   else:
. . .       False
```

A. True
B. False

```
>>> x = False
>>> if not x:
. . .       x = True
>>> not x
```

C. True
D. False

Part 4: Free Response: *20 Points*

Given a cons list or a SortList, write the function for cons or a method for Sortlist called getMean that takes an input list and returns the mean of the values in the list as either a rounded integer or a float.

```
>>> getMean(cons(2,cons(3,cons(10))))
5
>>> getMean(cons(2,cons(3)))
2.5
>>> getMean(cons(2))
2
>>> getMean(None)
>>>
```

SECTION III: Data, Method, Class **40 Points**

Part 5: Multiple Choice: *4 Questions @ 5 Points each =* *20 Points*

Given some x, determine whether it is a data, method/function, class object or None.

```
>>> x = print
>>> type(x)
```

A. Data
B. Method or Function
C. Class
D. None

```
>>> x = print("hi")
>>> type(x)
```

A. Data
B. Method or Function
C. Class
D. None

```
>>> x = "hi"
>>> type(x)
```

A. Data
B. Method or Function
C. Class
D. None

```
>>> import sortable
>>> x = sortable.Sortable()
>>> type(x)
```

A. Data
B. Method or Function
C. Class
D. None

Part 6: Written Response: 4 Questions @ 5 Points each = 20 Points

As an object oriented language, everything in Python is an object of some kind. Describe briefly the uses of and differences between the following kinds of objects and the Python “None”

What is a data object?

What is a method or function object?

What is a class object?

What is None?

SECTION IV: RECURSION **80 Points**

Part 7: Coding Exercises *2 Questions @ 40 Points each =* *80 Points*

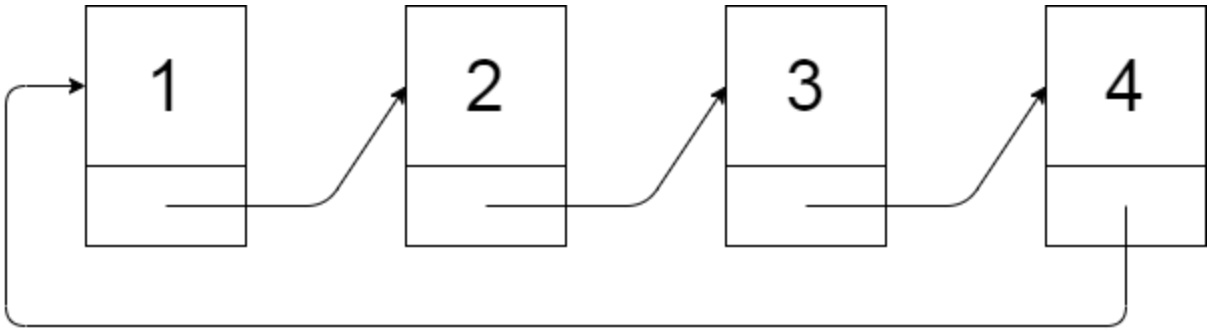
Write the function “ackerman” which takes as input two natural numbers and produces as output a single natural number. It is defined recursively as follows:

$$\begin{aligned} \mathrm{A}(0, n) &= n + 1 \\ \mathrm{A}(m + 1, 0) &= \mathrm{A}(m, 1) \\ \mathrm{A}(m + 1, n + 1) &= \mathrm{A}(m, A(m + 1, n)) \end{aligned}$$

Implement the class CircularList.

A circularly linked list is similar to a simple linked list but has no last element. Instead, what would be the last element links to the first element, rather than to nothing (or “None”).

Visually, we can represent this data structure as containing a data field (that may contain a value such as an integer) and a next field, similar to simply linked lists. It is simply the case that every element of the list has some next element.



A CircularList need not necessarily be sorted or contain any particular type of data. However, the next field must always be another circularly linked list.

In your class, include a constructor (initializer), and an insert method. You do not need to include other list methods such as size, contains, or remove. You may use any of a recursive data structure, a node implementation with an inner class, or a functional implementation if it is your preference to do so, but most have some notion of a data structure with two fields holding all the relevant information.