CS 241 DATA STRUCTURES FALL 2021 MIDTERM I
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 returns True if x is a string, such as “hello”?

A. >>> x == str
B. >>> x == type(“hi”)
C. >>> type(x) == type(“hi”)
D. >>> x == “hi”

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

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

Which of the following is a “data” object (assuming sortable as in HW2)?

A. >>> “hi”
B. >>> print
C. >>> print(“hi”)
D. >>> sortable.Sortable

Which of the following is a “function or method” object?

A. >>> “hi”
B. >>> print
C. >>> print(“hi”)
D. >>> sortable.Sortable

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

Suppose you wish to create a file sum.py that when run using “python sum.py” at the command line prints the result of the sum operation “2 + 2” and does nothing else. Write the contents of the file sum.py:

Suppose you have some variable “x” and you wish to determine if it is a single character. Write the function isCharacter that returns True if x is a string of length exactly 1, returns False otherwise, and does not cause an error regardless of what input is provided.

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 False:
. . .       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*

One way to improve the performance of data structures sorting numerical data is to split integers up with an easily computable function. Create the function fourSplits which given a single integer input “x” returns the following:

- If x is greater than zero and even, return the string “pos even”
- If x is lesser than zero and even, return the string “neg even”
- If x is greater than zero and odd, return the string “pos odd”
- If x is lesser than zero and even, return the string “neg odd”

You do not need to consider zero.

```
>>> fourSplits(10)
pos even
>>> fourSplits(3)
pos odd
>>> fourSplits(-1)
neg odd
>>> fourSplits(-6784)
neg even
```

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*

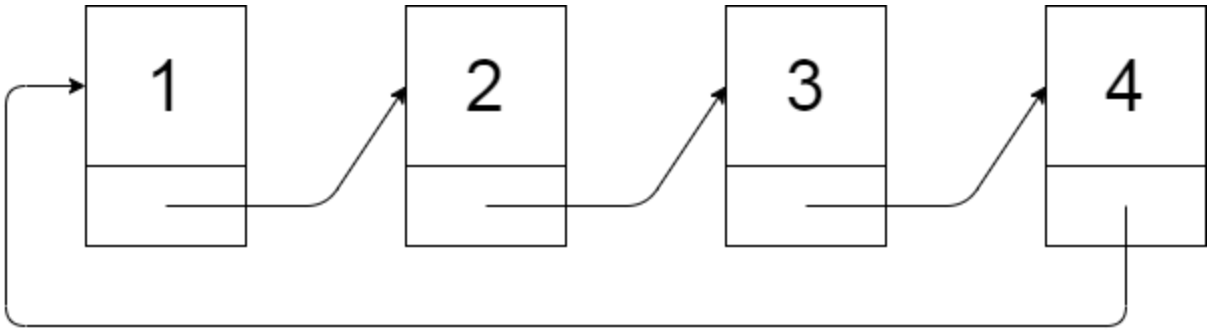
Write the function “ackerman” which takes as input two natural numbers and produces as output a single natural number. It is defined recursively as follows:

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

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 either 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.