CS 241 DATA STRUCTURES SPRING 2022 MIDTERM II (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*

Given arr = [[5,2],[4,9]], which of the following would give 4?

A. >>> arr[0][0]
B. >>> arr[0][1]
C. >>> arr[1][0]
D. >>> arr[1][1]

Given arr = [“a sentence”, “a line”, “a phrase”], what is arr[-1][-1]?

A. TypeError
B. “a”
C. “e“
D. “s“

Suppose arr[“a key”] gives the list object “[5,7]”. What is the type of arr?

A. Boolean
B. Dictionary
C. Integer
D. List

Suppose arr[5:7] gives the list object “[5,7]”. What is the type of arr?

A. Dictionary
B. List
C. Set
D. Tuple

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

Python Lists and Python Strings can both contain characters that may be accessed individually using integer indices. What are similarities and differences between using Lists and Strings to hold characters?

In some cases, changing one Python object may change another Python object under a different name:

```
>>> A = [1]
>>> B = A
>>> B[0] = 2
>>> A[0]
2
```

Offer an opinion of whether this is helpful or not to a programmer and no fewer than two arguments supporting your opinion.

SECTION II: ORGANIZING DATA **60 Points**

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

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

```
>>> [3,2,1][-2] == 3
```

A. True
B. False

```
>>> 2 in [1,2,3][-1:]
```

A. True
B. False

```
>>> somelist = [4,3,2,1]
>>> all([somelist[i] > somelist[i+1] for i in range(len(somelist) - 1)])
```

A. True
B. False

```
>>> if True:
. . .       False
. . .   else:
. . .       True
```

A. True
B. False

Part 4: Class Writing: *40 Points*

We can aid the process of assigning lab sections using a sectionee class that contains:

- A constructor function that takes two inputs in the form of the sectionee's name (a string) and their/her/his availability (a list of booleans, where a True indicates that sectionee is available during that index time block)
- A getter method get_name to return the sectionee's name.
- A check_time method which takes an integer index as input, checks to see if the sectionee is available during that time slot, and returns True or False accordingly.

For example, the following piece of code should be viable given your class definition

```
>>> s = Sectionee("Vijaya", [True , False , False , False , True , True])
>>> print(p.get_name ())
>>> "Vijaya"
>>> print(p.check_time(3))
>>> False
```

SECTION III: DATA MANAGEMENT **100 Points**

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

Which of the follow gives a list of even numbers [0,2,4,6,8]:

A. [i for i in range(10) if i % 2]
B. [i for i in range(10) if i % 2 != 0]
C. list(range(10) if i % 2)
D. list(range(10) if not i % 2)

Which of the follow gives a list of even numbers [0,2,4,6,8]:

A. list(range(10))[: : 2]
B. list(range(10))[1 : : 2]
C. list(range(10))[: 2 :]
D. list(range(10))[1 : 2 :]

Say we assign var = [c for c in “one two three”]. What is the type of var?

A. “Character” (string of length one)
B. String that is not a character
C. Integer
D. List

Say we assign var = [x for x in str(1234)]. What is the type of var[1]?

A. “Character” (string of length one)
B. String that is not a character
C. Integer
D. List

Part 6: Written Response: 2 Questions @ 20 Points each = 40 Points

In Python, data can be stored in built-in data types (lists, dictionaries) or in class you write or import from a library. Discuss some cases under which you would use one or the other.

Suppose you wish to catalog your meals for the past week using Python. How could you design a class to track this information? Argue rhetorically or provide sample code or pseudo-code.

Reminder to practice positive nutrition habits through the end of the semester!

Part 7: Using Dictionaries *60 Points*

Consider a program “exchangeCalculation.py” that reads user input to look up current currency exchange rates in a provided dictionary and print them to console.

```
$ python exchangeCalculation.py
Conversion:
1.00 USD -> GBP
1.00 USD = 0.76 GBP
Conversion:
100 NGN to CNY
100 NGN = 1.53 CNY
Conversion:
```

Each value in the currencies dictionary is itself a dictionary that specifies the name of the currency and its current equivalent exchange rate relative to the Chinese Yuan “CNY”.

```
currencies = {
    “CNY”: { "name": "Chinese Yuan", "rate": 1.000 },
    "USD": { "name": "US dollar", "rate": 6.37 },
    "GBP": { "name": "British pound", "rate": 8.29  },
    "INR": { "name": "Indian rupee", "rate": 0.084 },
    "NGN": { "name": "Nigerian Naira", "rate":0.015 },
    "BRL": { "name": "Brazilian Real", "rate":1.35 },
# etc
}
```

Your program should prompt the user to input a request with “input(“Conversion\n”) and then the user should input a request of the form:

amount XXX -> YYY

Which yields a response of form:

amount XXX = *converted* YYY

For any other inputs, the output is not defined.

Write “exchangeCalculation.py”: