

## CSCI 127, Final Practicum – May 3, 2018

Name \_\_\_\_\_

Question One. 25 points. Supply the missing function. The missing function should return the total number of calories in the meal. The variable **nutrition\_info** can contain any number of sublists, as can **my\_order**. The solution should be high quality; comments are not necessary.

**# The missing function goes here but write it below.**

```
# Each sublist contains the name of a McDonald's item, followed by the calories in that item
nutrition_info = [{"denali big mac", 850}, {"artisan grilled chicken", 440}, {"medium fries", 340},
                  {"large coke", 290}, {"chocolate chip cookie", 170}]
```

```
# The following order consists of 2 denali big macs, 1 medium fries and 1 large coke
my_order = [{"denali big mac", 2}, {"medium fries", 1}, {"large coke", 1}]
```

```
# For this example, the correct output is Calories in order = 2330
print("Calories in order =", calculate_calories(my_order, nutrition_info))
```

Question Two. 25 points. In the periodic table, the first four noble gases are Helium, Neon, Argon and Krypton. Write the two missing methods (comments are not necessary) so that all elements from Hydrogen (atomic number 1) through Krypton (atomic number 36) are processed correctly. For example, in the following program, this output should be produced:

*Hydrogen has atomic number 1.*

*Helium has atomic number 2 and is a noble gas.*

**class Element:**

**# The first missing goes here but write it below**

**def \_\_str\_\_(self):**

**answer = self.name + " has atomic number " + str(self.atomic\_number)**

**if self.is\_noble\_gas():**

**answer += " and is a noble gas."**

**else:**

**answer += "."**

**return answer**

**# The second missing method goes here but write it below**

**# -----**

**hydrogen = Element(1, "hydrogen")**

**print(hydrogen)**

**helium = Element(2, "helium")**

**print(helium)**

Question Three. 25 points. Supply the missing python statements such that an n by n numpy array is created. The array should contain alternating zeros and ones with a zero in the top left corner. Make your solution as simple as possible. Hint: consider using the concept of **slicing**.

Example 1: if n is 5, the output should be:

```
[[0 1 0 1 0]
 [1 0 1 0 1]
 [0 1 0 1 0]
 [1 0 1 0 1]
 [0 1 0 1 0]]
```

Example 2: if n is 6, the output should be:

```
[[0 1 0 1 0 1]
 [1 0 1 0 1 0]
 [0 1 0 1 0 1]
 [1 0 1 0 1 0]
 [0 1 0 1 0 1]
 [1 0 1 0 1 0]]
```

```
import numpy as np
```

```
size = int(input("Enter the size of the n by n board: "))
# The missing code goes here, but write it below
print(checkers)
```

Question Four. 25 points. Download the "Classics CSV Library" from <https://think.cs.vt.edu/corgis/csv/classics/classics.html>. Complete the Python program below such that it identifies the book with the most downloads. (You may assume there is only one such book but the solution should be general.) For this dataset, this output is produced:

*Pride and Prejudice by Austen, Jane has 36576 downloads*

```
import pandas as pd
```

```
data_frame = pd.read_csv("classics.csv")
```

```
# The solution goes here, but write it below. Comments are not necessary.
```

```
print(title, "by", author, "has", downloads, "downloads")
```