Full Name: ANSWER KEY

Email Address (UW Net ID): @uw.edu

Section:

CSE 160 Summer 2024 - Midterm Exam

Instructions:

- You have **55 minutes** to complete this exam.
- The exam is **closed book**, including no calculators, computers, phones, watches or other electronics.
- You are allowed a single sheet of notes for yourself.
- We also provide a syntax reference sheet.
- Turn in *all sheets* of this exam, together and in the same order when you are finished.
- When time has been called, you must put down your pencil and stop writing.
    - **Points will be deducted if you are still writing after time has been called.**
- You may only use parts and features of Python that have been covered in class up to this point.
- You may ask questions by raising your hand, and a TA will come over to you.

Good luck!

Question	**Topic**
Question 1	Expressions
Question 2	Loops, Ifs
Question 3	Files
Question 4	Ifs, Loops, Lists
Question 5	Functions

Question 1)

All of the expressions in the left hand column are going to be printed, what value would be output to the terminal if printed. If evaluating the expression would result in an error, write "Error" in both the value and type columns. If you are writing a string please include a "_" where a space should be.

Expression	Value After Evaluating	Type
`100 - (27 // 2)`	87	Int
`27 % 5 - 10`	-8	int
`'Python in CSE' + 160`	Error	Error
`['hello', 1, 3, [5, 6], 22][1:4:2]`	[1, [5, 6]]	List
`my_lst = [22, 29, 23][3]` `[5, 4, 7, 1, 8, 9].append(my_lst)`	Error	Error
`[5, 4, 7, 1, 8, 9].sort()`	None	None Type
`'Good' + 'Job!'`	'GoodJob!'	String

Question 2)
You are given a list of barcodes. An example could be:

```
example_barcodes =          [[1, 1, 0, 0, 0, 1, 1],
                            [1, 0, 1, 0, 0, 1, 0],
                            [0, 1, 0, 1, 1, 0, 1],
                            [1, 1, 1, 0, 1, 0, 0],
                            [0, 0, 1, 1, 0, 1, 1]]
```

This barcode list is subject to change, so all code should be written without hard coding the length of `example_barcodes`. All barcodes are the same length and consist of 0s and 1s. For problems a and b, there is no need to write a function.

a. Write a code snippet to count the number of 1s in each barcode and store the counts in a list. In the example above, the expected output would be `[4, 3, 4, 4, 4]`.

```
one_counts = []
for barcode in barcodes:
  total = 0
  for value in barcode:
    total += value
  one_counts.append(total)
```

b. Write a code snippet to sum the values at different integers in the barcodes and store them in a list. The goal is to sum each column. In the example above, the expected output would be `[3, 3, 3, 2, 2, 3, 3]`.

```
column_sums = [0] * len(barcodes[0])
for barcode in barcodes:
  for i in range(len(barcode)):
    column_sums[i] += barcode[i]
```

Question 3) You have a large file called `element.txt` that contains information about all of the elements in the periodic table in the following format:

```
H Hydrogen 1.008 amu
He Helium 4.003 amu
…
```
(more lines not shown)
```
…
Rn Radon 222.01 amu
```

You wish to break up the longer file into smaller `.txt` files that contain information about each element and have come up with the following lines of code

```
file = open("elements.txt")
for line in file:
      element = open("info.txt", "w")
      element.write(line)
file.close()
```

a. Why would this code not work when trying to split up the longer element.txt file into smaller files for each element?

Using open("info.txt", "w") will only create one file called info.txt rather than creating multiple files.

b. What would be inside the `info.txt` file as the code it is written? (write it out or describe it)

`Rn Radon 222.01 amu` or something like "the last line of the .txt file".

c. While running the first line of code (`file = open("elements.txt")`) you get a `FileNotFoundError`. You know for sure that you saved `elements.txt` somewhere, although you forgot exactly where. Given what you know about file I/O, why did you receive this error even though `"elements.txt"` exists on your computer?

The current working directory or absolute file name is something to check.

Question 4) You run the code below.

```
numbers = [5, 8, 13, 20, 25]
result = []
for num in numbers:
      if num % 2 == 0:
            result.append(num * 2)
      elif num > 15 or num < 8:
            result.append(num)
      else:
            if num % 5 == 0:
                  result.append(num + 5)
            else:
                 result.append(num - 3)
print(result)
```

What will be output to your terminal?

[5, 16, 10, 40, 25]

Question 5)
You are given a list of lists of integers called `temperatures`, where each list contains all the recorded temperatures of a city. You are given another list called `cities` that contains the names of all of the cities that each list in `temperatures` corresponds to. For example, index 0 of `temperatures` contains the recorded temperatures for the city in index 0 of `cities`, and so on.

Write a function temperature_averaging that creates a dictionary of key-value pairs where the key is the name of the city and the value is the average recorded temperature for that city.

For example, given

```
temperatures = [[78, 92, 24], [34, 12], [40, 4, 2, 67, 8]]
cities = ["Paris", "Tokyo", "Seattle"]
```

Your code should return the following dictionary:
`{"Paris" : 64.6666667, "Tokyo": 23, "Seattle" :  38.6}`

You should not use the built-in python function `sum`. You may assume that the lists `temperatures` and `cities` are already defined. Write your code in the provided function header below. Do not do any rounding.

```
def temperature_averaging(temperatures, cities):

    city_avg_temp = {}

    for i in range(len(cities)):
        city = cities[i]
        temp_list = temperatures[i]
        temp_sum += 0
        count = 0

        for temp in temp_lst:
            temp_sum += temp
            count += 1

    avg_temp = temp_sum / count
    city_avg_temp[city] = avg_temp
    return city_avg_temp
```

Extra Credit: Draw your favorite dessert.