CSE 115
Introduction to Computer Science I
Announcement

For lab exam 2 only:

We will allow students to take the exam even if lab entry ticket is not yet earned.

Lab entry ticket must be completed within two days of student's scheduled lab time for grade on lab exam 2 attempt to be counted.
Reminder

As usual, all students are eligible for the lab exam 2 make-up after lab exam 3.
Road map

▶ exercise/review ◀

HTML

Front End JavaScript
def writeFile(filename, contents):
    with open(filename, 'w') as f:
        for item in contents:
            f.write(item+'\n')

The write function expects a string. To print other types of values, first convert them to an equivalent string.

For example:
    f.write(str(7))
import csv

def writeCSVFile(filename, dataTable):
    with open(filename, 'w', newline='') as f:
        writer = csv.writer(f)
        for record in dataTable:
            writer.writerow(record)

This writes the members of record on one line, separated by commas.
dt = [['abc', 'def'], ['ghij', 'klmn']]
writeCSVFile('file1', dt)

writeCSVFile('file2', dt[0])

abc,def
ghij,klmn

a,b,c
d,e,f
Exercise

*Lab Activity 4, part (c) didn’t go as well as expected

We expect you to be able to expand on the concepts we cover

This is not a course about memorization

This is a course about creative problem solving using a set of concepts taught in class

That said.. part (c) of last week’s lab may have taken this too far for a timed assignment. Let’s cover an example similar to that question
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

To solve this we’ll follow our same accumulation pattern, however we will replace our entire value each time a condition is true. To solve this we will iterate over the dictionary while

1. Keep track of the maximum value seen
2. Keep track of the key where that max value was stored
3. Update both max key and value whenever we see a value large than the max found
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```
Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```

“accumulator” variables
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```

Need access to both key and value.
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```

Whenever a new max value is found, update key and value
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```

After the loop all values have been checked and `max_value` stores the maximum value in the entire dictionary.

Return the key associated with that value.
Exercise

However, this solution has one major flaw

What happens if every value is negative?
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
def find_max_key(dictionary):
    max_value = 0
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```

If all values are negative, the condition is never true and we’ll return ""
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

```python
import math

def find_max_key(dictionary):
    max_value = -math.inf
    max_key = ""
    for key in dictionary:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key
```

That's better

Since we’re finding the max value we can use -math.inf, though this will be different depending on our goal.
Exercise

Define a function that takes a dictionary of strings to floats as a parameter and returns the key associated with the maximum value in the dictionary.

def find_max_key(dictionary):
    keys = list(dictionary.keys())
    max_key = keys[0]
    max_value = dictionary[max_key]
    for key in keys:
        value = dictionary[key]
        if value > max_value:
            max_value = value
            max_key = key
    return max_key

Here's another approach - seed the process with items from the input.

This works even if the values are not numeric.
Road map

exercise/review

▶ HTML ◀

Front End JavaScript
HTML

Hyper Text Markup Language

Hyper Text: Text that can contain links to other resources

Markup Language: Special markers add information to the text that is not displayed. In HTML we use tags that tell the browser how to display the text

HTML is not a programming language
Save this in a file with a .html extension and open it in a web browser to see the web page below.
HTML - Elements

<html>
<head></head>
<body>
<h1>First Web Page</h1>
<p>My content</p>
<div id="myDiv"></div>
</body>
</html>

HTML uses angle brackets to define elements.

Each element has an open tag <h1> and close tag </h1>

Everything between the open and close tag is the content of that element.

In this example we used header 1 (h1) and paragraph (p) tags to display text with different sizes.
Elements can contain properties which are defined in the open tag of the element

These properties are key-value pairs

We have an empty division with a property named id with a value of “myDiv”
HTML

HTML is not a programming language

This is as much as we cover HTML in this course.

For much, much more information about HTML and other web technologies visit w3schools

https://www.w3schools.com
Road map

eexercise/review

HTML

▶ Front End JavaScript ◀
Instead of learning more HTML we will instead write JavaScript code to add more power to our web pages.

We’ll “import” our javascript code by adding a script element at the bottom of the body element with a src (source) property containing our JavaScript filename.

This runs our script once the body is loaded.
var myDiv = document.getElementById("myDiv");
myDiv.innerHTML = "Content added from JavaScript";

We’ll save this code in a file named “myCode.js” and it will run once the content of our HTML page is loaded.

Here we call the `document.getElementById` method with the id of an element as an argument.

The element is an object with a key “innerHTML” whose value is the content of the element.
Front End JavaScript

The script runs when the body loads and sets the content of “myDiv” resulting in this page

First Web Page

My content

Content added from JavaScript
Front End JavaScript

For more visit w3schools tutorial on DOM (Document Object Model) manipulation with JavaScript

https://www.w3schools.com/js/js_htmldom.asp