|  |  |  |  |
| --- | --- | --- | --- |
|  |  | ***for-each* loops** |  |

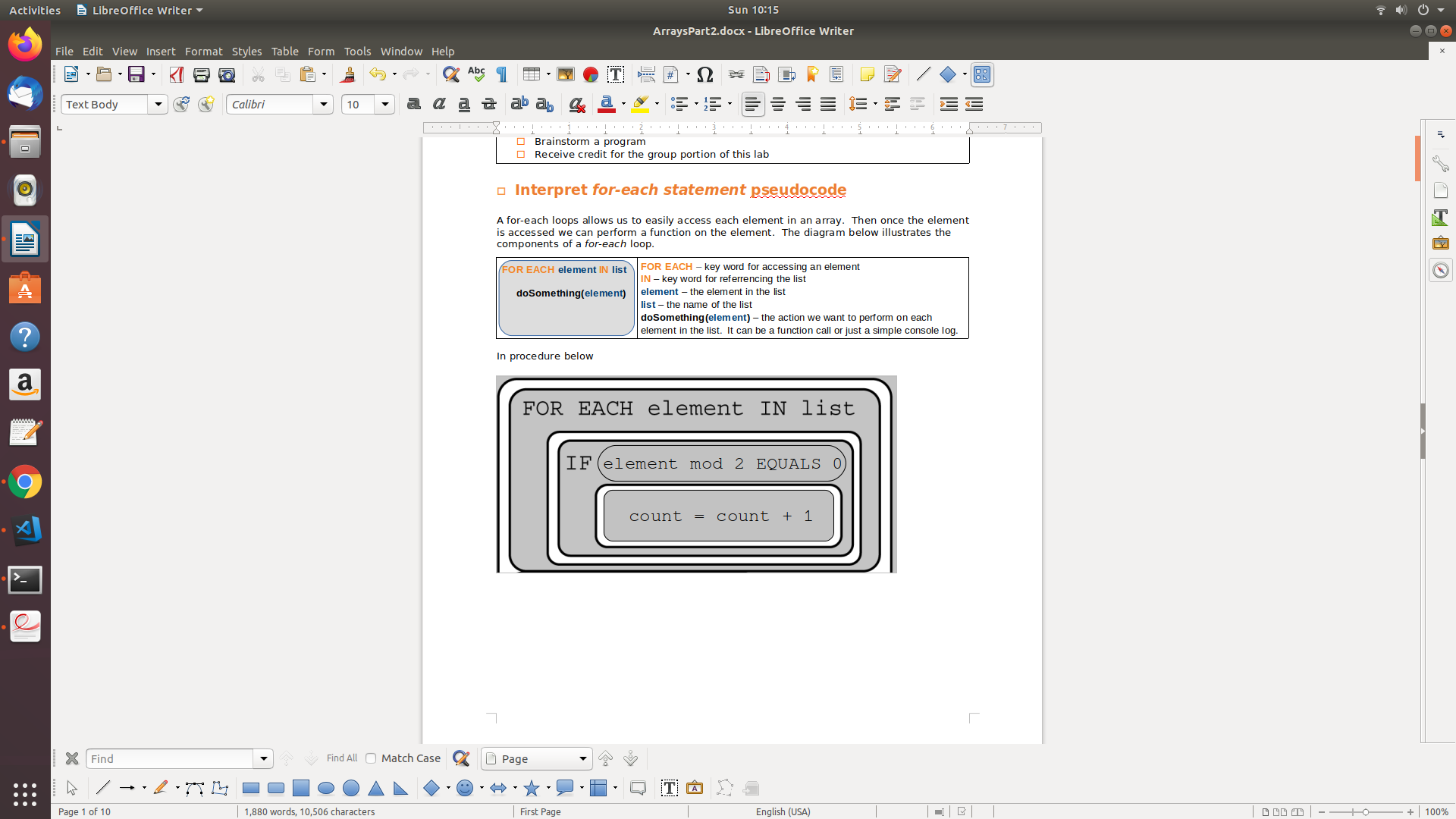
|  |
| --- |
| **Your Tasks (Mark these off as you go)** |
| * Interpret *for-each* loop pseudocode * Have Ms. Pluska check off the above tasks * Write a *for-each* loop * Brainstorm a program * Receive credit for the group portion of this lab |

* **Interpret *for-each statement* pseudocode**

A for-each loops allows us to easily access each element in an array. Then once the element is accessed we can perform a function on the element. The diagram below illustrates the components of a *for-each* loop.

|  |  |
| --- | --- |
|  | **FOR EACH** – key word for accessing an element  **IN** – key word for referrencing the list  **element** – the element in the list  **list** – the name of the list  **doSomething(element)** – the action we want to perform on each element in the list. It can be a function call or just a simple console log. |

In the example below we use a *for-each* loop to count all the even numbers in a list. For each element in the list, we are checking if it is divisible by two, and if it is, we increment *count*.



The example above could have also been written as follows,

|  |  |
| --- | --- |
|  |  |

In the above example, the procedure *countEvens* is called each time an element is found in the list. Each element is passed to the procedure *countEvens*, where *count* is incremented if the element is even.

|  |
| --- |
| A student is creating a procedure to determine whether the weather for a particular month was considered very hot. The procedure takes as input a list containing daily high temperatures for a particular month. The procedure is intended to return true if the daily high temperature was at least 90 degrees for a majority of days in the month and return false otherwise.    < MISSING CODE A >  < MISSING CODE B > |
| 1. Which of the following can be used to replace <MISSING CODE A> so that the procedure works as intended? 2. Which of the following can be used to replace <MISSING CODE A> so that the procedure works as intended? |

|  |
| --- |
| The two code segments below are each intended to display the average of the numbers in the list numList. Assume that numList contains more than one value. |
| 1. Code segment I displays the correct average, but code segment II does not. 2. Code segment II displays the correct average, but code segment I does not. 3. Both code segments display the correct average, but code segment I requires more arithmetic operations than code segment II. 4. Both code segments display the correct average, but code segment II requires more arithmetic operations than code segment I. |
| The procedure below searches for the value target in list. It returns true if target is found and returns false otherwise.    < MISSING CODE > |
| 1. What should replace < MISSING CODE > for the procedure to work as intended? 2. Which of the following are true statements about the procedure?   I. It implements a binary search.  II. It implements a linear search.  III. It only works as intended when list is sorted. |

* **Have Ms. Pluska check off the above tasks**



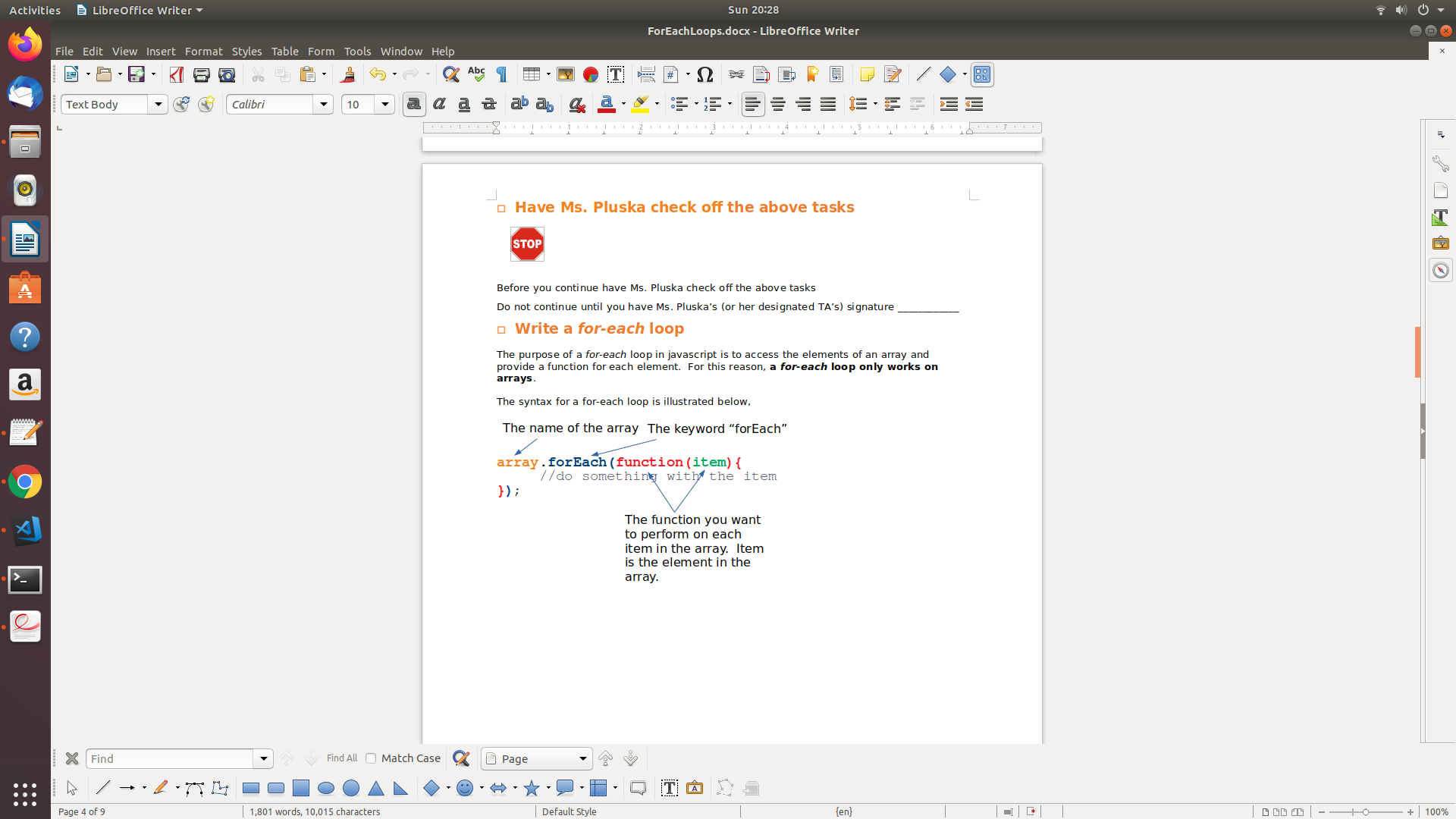
Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Write a *for-each* loop**

The purpose of a *for-each* loop in javascript is to access the elements of an array and provide a function for each element. For this reason, **a *for-each* loop only works on arrays**.

The syntax for a for-each loop is illustrated below,



The code snippet illustrates how to implement a *for-each* loop,

|  |  |
| --- | --- |
| **Code** | **Output** |
| var arr = ['Wall-E', 'Up', 'Coco'];  arr.forEach(function(item) {  console.log(item);  }); | Wall-E  Up  Coco |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Consider an array of cards called *dealt.* The value of each card can be accessed using the .getValue() function. How to use getValue() to access a cards value is illustrated below,   |  |  | | --- | --- | |  | Card a = new Card();  a.getValue();//returns 2 | |  | Carb b = new Card();  b.getValue(); |  1. Write a forEach loop that could be used to access each Card in the *dealt* array and print its value to the console. |
| 1. Write a forEach loop that could be used to locate the Card with the lowest value in the *dealt* array and print its value to the console. |

* **Have Ms. Pluska check of the above tasks**



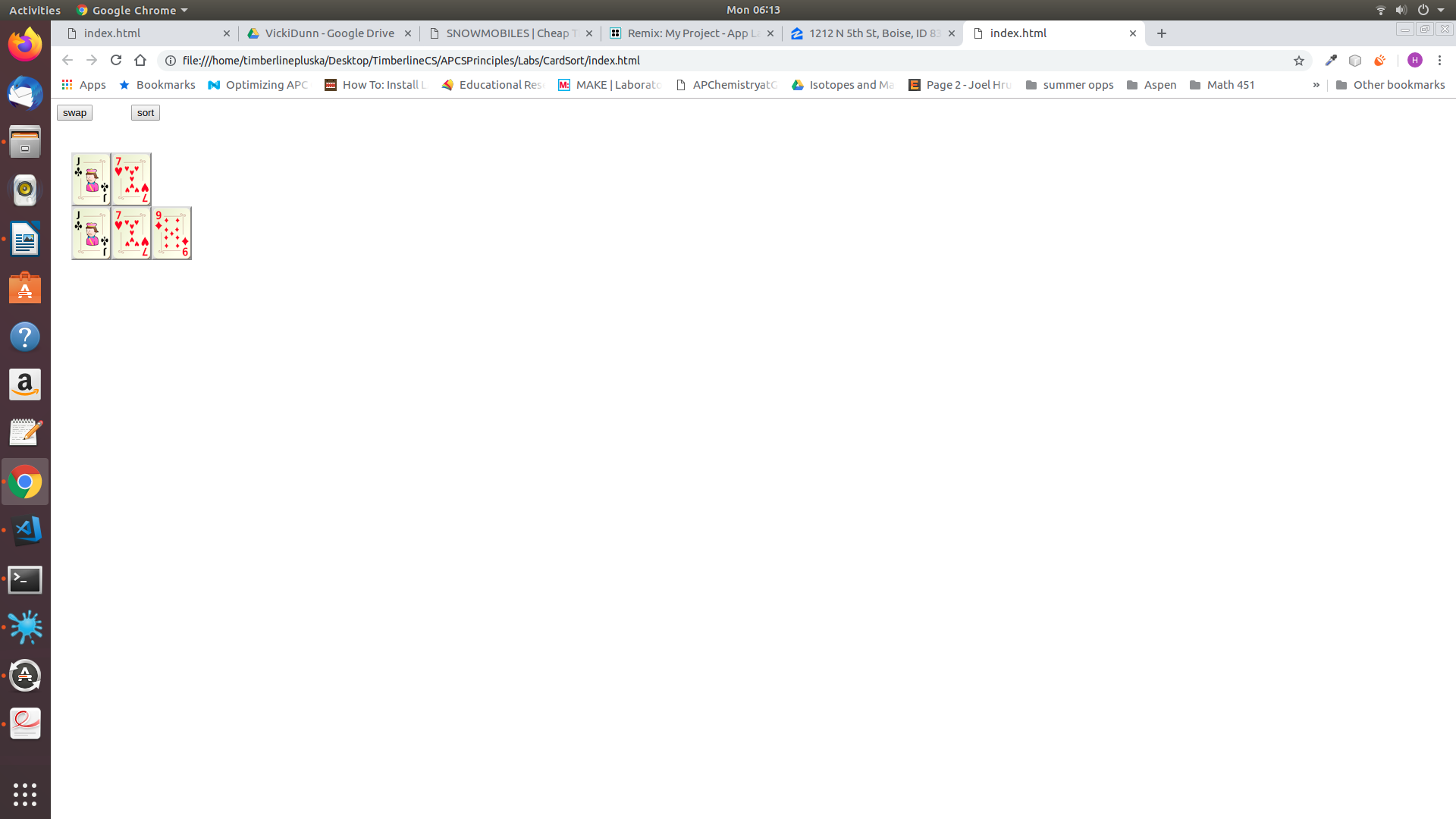
Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Brainstorm a program**

Challenge 1

Consider two cards A and B as defined below. Each card represents a card in a standard 52 deck of cards.



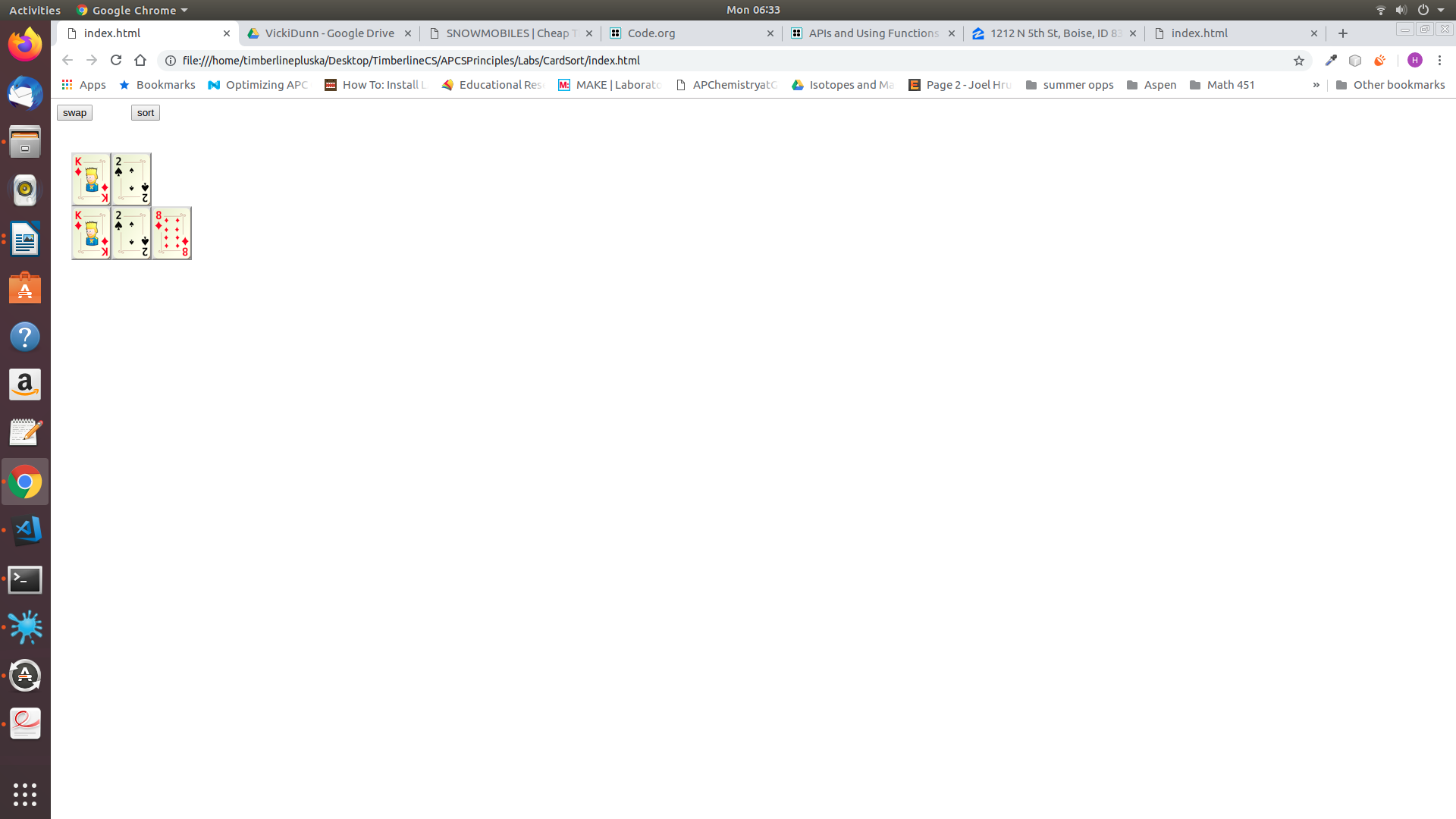
var a = jackOfClubs;

var b = sevenOfHearts;

Brainstorm a program that can swap the values of the cards. That is var a becomes the sevenOfHearts and card b becomes the jackOfClubs.

Challenge 2

Now consider three cards with the following values.



var a = kingOfDiamonds;

var b = twoOfSpades;

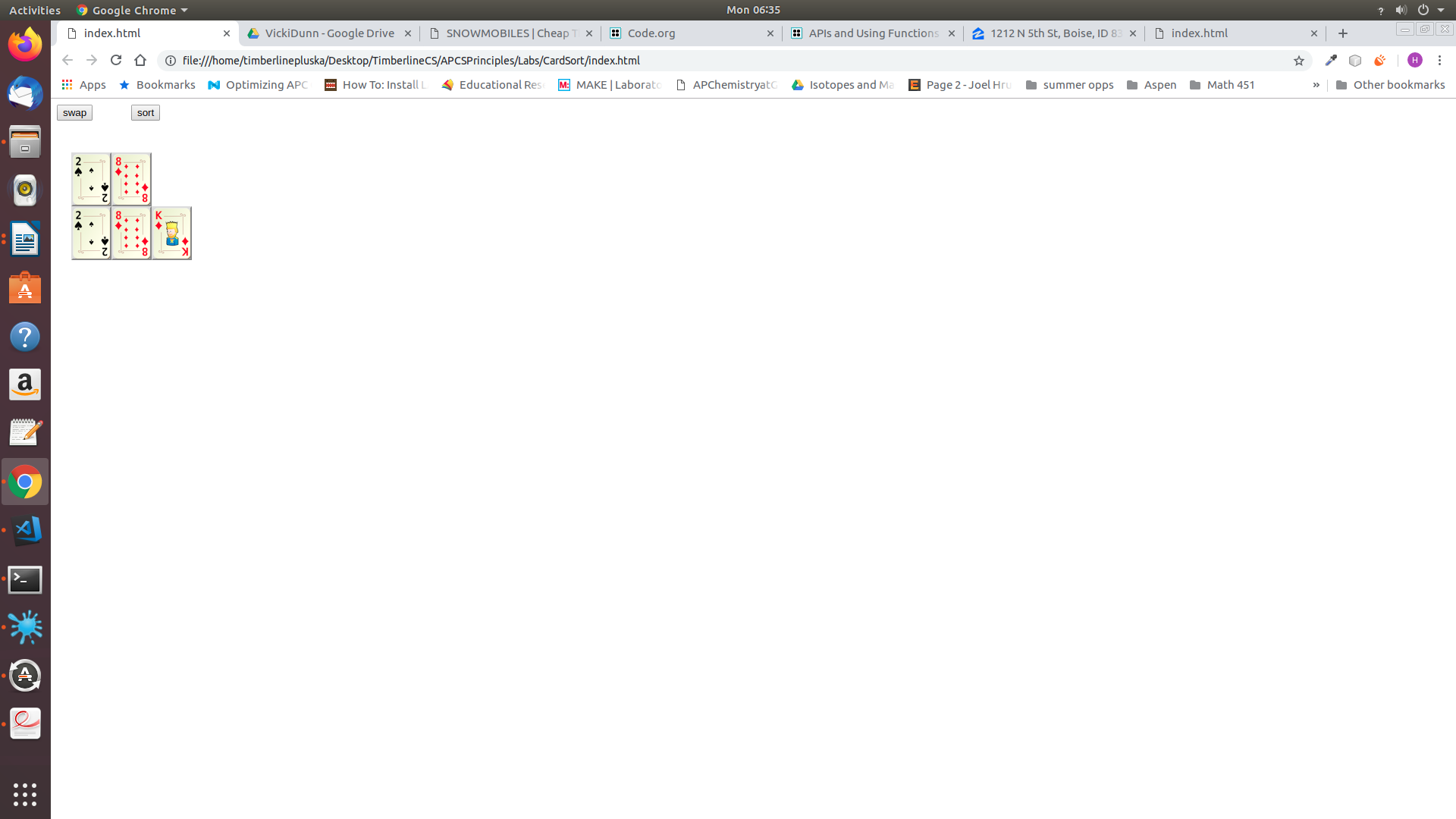
var c = eightOfDiamonds;

a.value = 13;

b.value = 2;

c.value = 8;

Brainstorm a program that could be used to sort the cards, as shown below. Your program must reassign the variables a, b, and c to the correct cards.



var a = twoOfSpades;

var b = eightOfDiamonds;

var c = kingOfDiamonds;

* **Receive Credit for the group portion of this lab**



* Indicate the names of all group members.
* Make sure both you and your partner have completed the above tasks
* Have Ms. Pluska check off the group tasks
* Submit your lab to the needs to be graded folder to receive credit for the group portion of this lab.
* Do not submit your lab until you have Ms. Pluska’s (or her designated TA’s) signature