

Name _____ Period _____ Role (Circle one) Programmer/Driver

Name _____ Period _____ Role (Circle one) Programmer/Driver

for-each loops

Your Tasks (Mark these off as you go)

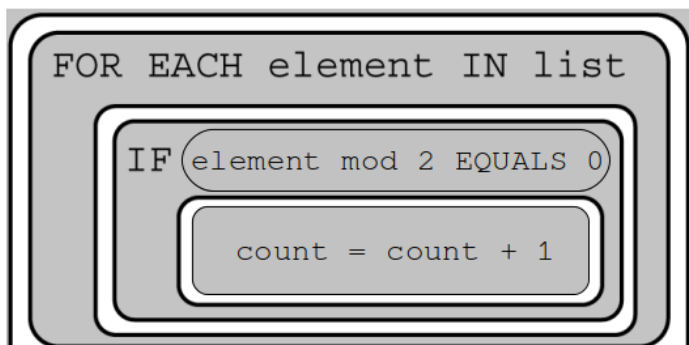
- ☐ Interpret *for-each* loop pseudocode
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Write a *for-each* loop
- ☐ Brainstorm a program
- ☐ Receive credit for the group portion of this lab

□ Interpret *for-each* statement pseudocode

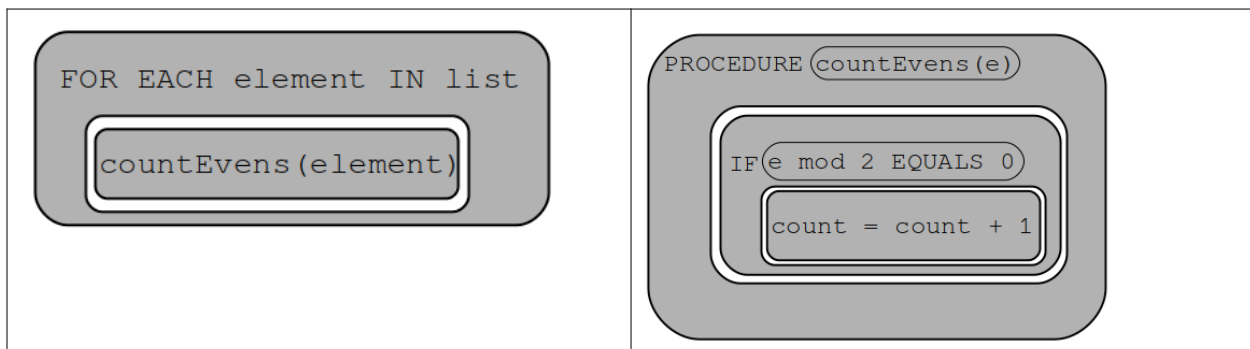
A *for-each* loop allows us to easily access each element in an array. Then once the element is accessed we can perform a function on the element. The diagram below illustrates the components of a *for-each* loop.

FOR EACH element IN list doSomething(element)	FOR EACH – key word for accessing an element IN – key word for referencing 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.

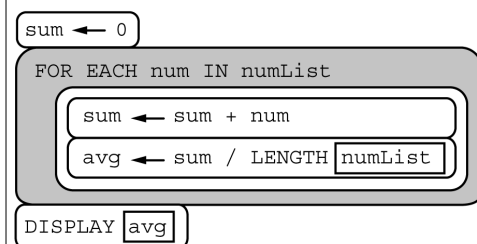
```
PROCEDURE IsHot (temperatureList)
{
  total ← 0
  counter ← 0
  FOR EACH temperature IN temperatureList
  {
    IF (< MISSING CODE A > )
    {
      counter ← counter + 1
    }
    total ← total + 1
  }
  RETURN (< MISSING CODE B > )
}
```

(A) Which of the following can be used to replace <MISSING CODE A> so that the procedure works as intended?

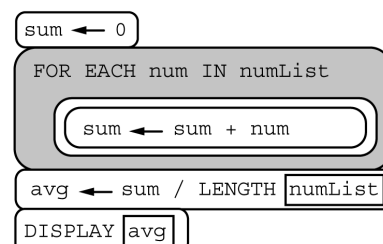
(B) Which of the following can be used to replace <MISSING CODE A> so that the procedure works as intended?

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.

Program I:

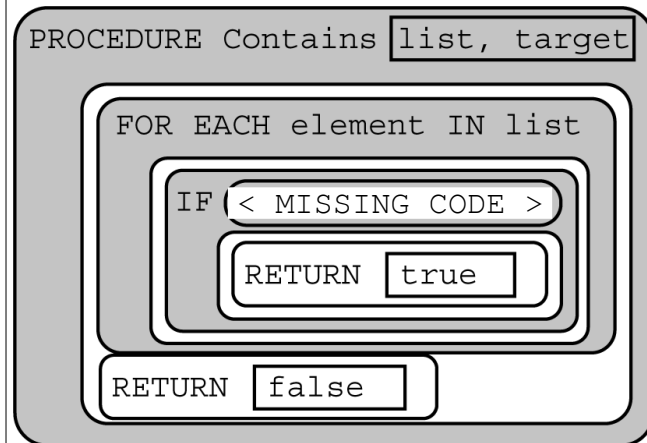


Program II:



- (A) Code segment I displays the correct average, but code segment II does not.
- (B) Code segment II displays the correct average, but code segment I does not.
- (C) Both code segments display the correct average, but code segment I requires more arithmetic operations than code segment II.
- (D) 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.



(A) What should replace < MISSING CODE > for the procedure to work as intended?

(B) 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 name of the array The keyword "forEach"

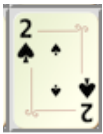

```
array.forEach(function(item){  
    //do something with the item  
});
```

The function you want to perform on each item in the array. Item is the element in the array.

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

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

Consider an array of cards called *dealt*. The value of each card can be accessed using the `.getValue()` function. Likewise, the suit of each card can be accessed using the `.getSuit()` function. How to use `getValue()` and `getSuit()` to access a cards value and suit is illustrated below,

	<pre>Card a = new Card(); a.getValue();//returns 2 a.getSuit();//returns spades</pre>
	<pre>Card b = new Card(); b.getValue();//returns 8 b.getSuit();//returns diamonds</pre>

(A) Write a `forEach` loop that could be used to access each Card in the *dealt* array and print its value and suit to the console.

(B) Write a function that could be used to locate the Card with the lowest value in the *dealt* array and return the Card

(C) Write a function that could be used to locate the number of diamonds in the *dealt* array and return the value.

□ 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

The cards shown below represent an array of cards called *dealt*.

Brainstorm code that could be used to sum the values of all the cards in the *dealt* array and return the total.



Challenge 2

For the dealt array shown above, brainstorm code that could be used to count the number of hearts, spades, clubs, and diamonds. The total number of each suit should be stored in an array called *suits* where index 0 represents the number of hearts, index 1 represents the number of spades, index 2 represents the number of clubs, and index 3 represents the number of diamonds. For example, for the *dealt* array shown above *suits*, would look as follows,

```
var suits = [1, 1, 2, 3]
```

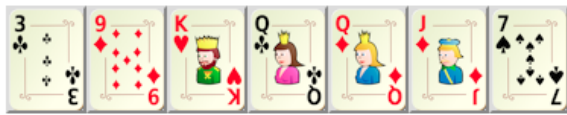
Challenge 3

For the dealt array shown above, brainstorm code that could be used to count the number of cards of a given value. The total number of each value should be stored in an array called *values* where index 1 represents the number of aces, index 2 represents the number of 2's, index 3 represents the number of 3's, etc. For example, for the *dealt* array shown above *values*, would look as follows,

```
var suits = [0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 2, 1]
```

Challenge 4

For the dealt array shown above, brainstorm code that could locate the minimum card and bring it to the front. When your code is ran on the dealt array shown above the 3 of clubs should be moved to the front as shown below,



❑ 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
