JavaScript Exercise

1. We have an object storing salaries of our team

let salaries = {

John: 100,

Ann: 160,

Pete: 130

}

Write the code to sum all salaries and store in the variable sum. Should be 390 in the example above.

1. Create a function multiplyNumeric(obj) that multiplies all numeric properties of obj by 2

// before the call

let menu = {

width: 200,

height: 300,

title: "My menu"

};

multiplyNumeric(menu);

// after the call

menu = {

width: 400,

height: 600,

title: "My menu"

};

Please note that multiplyNumeric does not need to return anything. It should modify the object in-place

1. Write a function checkEmailId(str) that returns true if str contains '@' and ‘.’, otherwise false. Make sure '@' must come before '.' and there must be some characters between '@' and '.'

The function must be case-insensitive:

1. Create a function truncate(str, maxlength) that checks the length of the str and, if it exceeds maxlength – replaces the end of str with the ellipsis character "…", to make its length equal to maxlength.

The result of the function should be the truncated (if needed) string.

truncate("What I'd like to tell on this topic is:", 20) = "What I'd like to te…"

truncate("Hi everyone!", 20) = "Hi everyone!"

1. Let’s try 5 array operations.

Create an array styles with items “James” and “Brennie”.

Append “Robert” to the end.

Replace the value in the middle by “Calvin”. Your code for finding the middle value should work for any arrays with odd length.

Remove the first value of the array and show it.

Prepend Rose and Regal to the array.

James, Brennie

James, Brennie, Robert

James, Calvin, Robert

Calvin, Robert

Rose, Regal, Calvin, Robert

1. We are transitioning from people swiping their credit card to using a chip. As part of the transition, some credit card companies have moved all of their customers to the new chip card, some are in the process, and some still have yet to comply. We need to implement a system that verifies the card swipe and determine if we should authorize the transaction using a card swipe or if the customer needs to use the chip feature of their card.

We need you to implement the validateCards() function. We will pass in two lists of strings. We are doing batch authorization so the first parameter is a list of the swiped card number. The second is a list of card prefixes that cannot be processed if the card is swiped.

The function should return a JSON array of credit card objects that contains: Card (String): the card number isValid(boolean): a modified Luhn check (described below) isAllowed (boolean): a check to verify that the number doesn’t start with any of the card prefixes.

The modified Luhn check verifies that the card was swiped correctly. We do his by calculating the check digit and comparing it to the last number on the card. We calculate the check digit by taking all but the last digit the card number, take the int value of each individual character, double it, add them together, then divide by 10 and take the reminder. That value is the calculated check digit. If this matches the check digit (last number in the string), then the card number is valid.

Below is an example Let’s say we have a card number: 6724843711060148. The last number is 8. We then have the string 672484371106014. We now take each character and convert it to a number: |6|7|2|4|8|4|3|7|1|1|0|6|0|1|4|

Then we double each number: |12|14|4|8|16|8|6|14|2|2|0|12|0|2|8|

Then we sum them up and get 108. We divide by 10 and the remainder is 8. The calculated check digit from our card matches the last number, so we set isValid to true. If it were any other number, we would set it to false.

For the full example, we would have the input parameter of: cardsToValidate (List of one credit card number): 6724843711060148 And bannedPrefixes(string): 11, 3434, 67453, 9 The result would be a JSON array with one credit card object{ [{card:”6724843711060148”, isValid:true, isAllowed:true}]

*[Hint: we can use JSON.stringify() to convert a JavaScript object into a JSON string]*