IIFE = Immediately Invoked Function Expressions

1. Write IIFE that replaces the first and the last element of the given array and prints out its elements.

Input array: [4, 5, 11, 9]  
 Output array: [ 9, 5, 11, 4]

|  |
| --- |
| (function (elements) {  console.log("Initial order: ", elements);  var indexOfFirstElement = 0;  var indexOfLastElement = elements.length - 1;  var tmp = elements[indexOfFirstElement];  elements[indexOfFirstElement] = elements[indexOfLastElement];  elements[indexOfLastElement] = tmp;  console.log("After rotating first and the last one: ", elements);  }([2, 3, 6, 8, 12])); |

1. Write IIFE that calculates the surface area of the given rectangle with sides *a* and *b*.   
   Input: 4 5  
   Output: 20

|  |
| --- |
| var rectangleSurface = (function (a, b) {  return a \* b;  }(3, 6));  console.log("\nRectangle surface: ", rectangleSurface); |

1. Write IIFE that replaces all appearances of the letters *m* or *M* with *\** and returns the number of replacements.

Input: prograMming  
 Output: progra\*\*ing, 2

|  |
| --- |
| var transformedInput = (function (sentence) {  var transformedSentence = "";  var charOccurrence = 0;  for (var charIndex in sentence) {  if (sentence[charIndex].toLocaleLowerCase() !== "m") {  transformedSentence += sentence[charIndex];  continue;  }  transformedSentence += "\*";  charOccurrence++;  }  return transformedSentence + ", " + charOccurrence;  }("prograMming"));  console.log("\nSentence without 'M' and 'm': ", transformedInput); |

1. Write a function with parameters name and surname that returns a function that suggests an email in the form [name.surname@gmail.com](mailto:name.surname@gmail.com).

Input: pera peric  
 Output: [pera.peric@gmail.com](mailto:pera.peric@gmail.com)

|  |
| --- |
| var getEmailSuggester = function (name, surname) {  return function () {  return name + "." + surname + "@gmail.com";  }  };  var suggester = getEmailSuggester("bit", "student");  var suggestedEmail = suggester();  console.log("\nSuggested email: ", suggestedEmail); |

1. Write a function that returns a function that calculates a decimal value of the given octal number.   
   Input: 034  
   Output: 28

|  |
| --- |
| var getOctalToDecimalConverter = function () {  function isOctalNumber(number) {  return number[0] === "0";  }  function convert(number) {  var decimal = 0;  for (var i = number.length - 1; i >= 1; i--) {  decimal += 8 \*\* (number.length - 1 - i) \* parseInt(number[i]);  }  return decimal;  }  return function (value) {  if (!isOctalNumber(value)) {  return 0;  }  return convert(value);  }  };  var octalToDecimalConverter = getOctalToDecimalConverter();  var decimalValue = octalToDecimalConverter("034");  console.log("\nDecimal value: ", decimalValue); |

1. Write a function that checks if a given string is valid password. The password is valid if it is at least 6 characters long and contains at least one digit. The function should receive two callbacks named successCallback and errorCallback that should be called in case password is correct or invalid.   
   Input: JSGuru   
   Output: Your password is invalid!

Input: JSGuru123  
 Output: Your password is cool!

|  |
| --- |
| var passwordChecker = function (password, success, error) {  success = success || function(){};  error = error || function() {};  function isPasswordValid(password) {  if (typeof password !== "string" || password.length < 6) {  return false;  }  for (var i in password) {  if (isNumeric(password[i])) {  return true;  }  }  return false;  }  function isNumeric(n) {  return !isNaN(parseFloat(n)) && isFinite(n);  }  return isPasswordValid(password)  ? success()  : error()  }  passwordChecker(  "sdfsdf123",  function(){ console.log("\nYour password is valid!"); },  function(){ console.log("\nYour password is invalid!"); }  ); |

1. Write a function that filters elements of the given array so that they satisfy a condition given by the callback function.  
   Input: [2, 8, 11, 4, 9, 3], callback function checks if the number is odd  
   Output: [11, 9, 3]

|  |
| --- |
| var filter = function(elements, conditionChecker) {  if (typeof conditionChecker !== "function") {  return elements;  }  var filteredColection = [];  for (var index in elements) {  if (!conditionChecker(elements[index])) {  continue;  }  filteredColection[filteredColection.length] = elements[index];  }  return filteredColection;  }  var filteredArray = filter(  [23, 5, 7, 42, 12],  function(element){ return element >= 20; }  );  console.log("\nFiltered array: ", filteredArray); |