```
// Write a JavaScript function to get the greatest common divisor (gcd) of two
integers.
function gcd_two_numbers(x, y) {
 if (typeof x !== "number" || typeof y !== "number") return false;
 x = Math.abs(x);
 y = Math.abs(y);
 while (y) {
   var t = y;
   y = x % y;
   x = t;
 }
 return x;
}

console.log("Greatest common Divisor: ", gcd_two_numbers(12, 144));
console.log("Greatest common Divisor: ", gcd_two_numbers(95, 35));
// ----------------------------------

// Write a Program for grading students
// A = 90-100, B = 80-89, C = 70-79, D = 60-69, F = 0-59
// Bonus points for returning emoji's representing grades!

function grades(num) {
 if (num >= 90 && num <= 100) {
   return "Grade is: A 🤓";
 }
 if (num >= 80 && num <= 89) {
   return "Grade is B 🤩";
 }
 if (num >= 70 && num <= 79) {
   return "Grade is C 😕";
 }
 if (num >= 60 && num <= 69) {
   return "Grade is D 😫";
 }
 if (num >= 0 && num <= 59) {
   return "Grade is F 🤬";
 } else {
   return "Did you give me a number 0-100?";
 }
}

console.log(grades(85));
// ----------------------------------

// Write a JavaScript program to determine whether a given year is a leap year in the
Gregorian calendar.

function leapyear(year) {
 return year % 100 === 0 ? year % 400 === 0 : year % 4 === 0;
}
console.log(leapyear(2016));
console.log(leapyear(2020));
console.log(leapyear(1754));
console.log(leapyear(1800));
console.log(leapyear(100));

// ----------------------------------

// Write a program for translating words into pig latin
// 1. For words that begin with consonants sounds, all letters before the inital vowel
are placed
// at the end of the word sequence. Then "ay" is addded (ex: "what" = "atwhay", "me" =
"emay")
// 2. When words bein with consonant clusters, the clusters should be moved to the end
of the word sequence and "ay"
// is affixed (ex: "glove" = "oveglay").
// 3. For words that begin with vowel sounds, simply add "way" to the end of the word
(ex. "explain", "explainway")

function pigLatin(str) {
 str = str.toLowerCase();
 const vowels = ["a", "e", "i", "o", "u"];
 let vowelIndex = 0;

 if (vowels.includes(str[0])) {
   return str + "way";
 } else {
   for (let char of str) {
     if (vowels.includes(char)) {
       vowelIndex = str.indexOf(char);
       break;
     }
   }
   return str.slice(vowelIndex) + str.slice(0, vowelIndex) + "ay";
 }
}

console.log(pigLatin("water"));

// ----------------------------------

// Given an integer, n, perform the following conditional actions:
// If n is odd, print Weird
// If n is even and in the inclusive range of 2 to 5, print Not Weird
// If n  is even and in the inclusive range of 6 to 20, print Weird
// If n is even and greater than 20, print Not Weird
// print whether or not n is weird.
// Sample n = 3; output = "WEIRD"

function main(n) {
 if (n % 2 !== 0) {
   console.log("Weird");
 } else if (n % 2 === 0 && n >= 2 && n <= 5) {
   console.log("Not Weird");
 } else if (n % 2 === 0 && n >= 6 && n <= 20) {
   console.log("Weird");
 } else if (n % 2 === 0 && n > 20) {
   console.log("Not Weird");
 }
}

// ----------------------------------
// In a given array of numbers, one element shows up once and the others each show up
twice.
// Find the number that only appears once
```

```
function lonelyNumber(numbers) {
 let appearances = {};

 for (let num of numbers) {
   if (appearances.hasOwnProperty(num)) {
     delete appearances[num];
   } else {
     appearances[num] = true;
   }
 }

 return parseInt(Object.keys(appearances)[0]);
}

// -----------------------------------
// We're provided a positive integer num. Can you write a method to repeatedly add all
of its digits until the result  has only one digit?
//  Example: start with 49; 4+9 = 13; 1+3 = 4; We would return 4!

function sumDigits(num) {
 while (num > 9) {
   const arr = String(num).split("");
   num = arr.reduce((sum, item) => {
     return sum + Number(item);
   }, 0);
 }
 return num;
}

console.log("SumDigits result is: ", sumDigits(49));

// -----------------------------------
// Bubble Sort - use an array with length of 8
// Compare the first item to the second item.
// If the first item should be after the second item, swap them.
// Compare the second item to the third item.
// If the second item should be after the third item, swap them.
// Continue until the end of the data set is reached.
```

```
function bubbleSort(arr) {
 const loop = arr.length; //loop length
 //loop for loop length
 for (let i = 0; i < loop; i++) {
   //cycle through arr items
   for (let j = 0; j < loop; j++) {
     // compare adjacent items
     if (arr[j] > arr[j + 1]) {
       let temp = arr[j];
       arr[j] = arr[j + 1];
       arr[j + 1] = temp;
     }
     //   console.log(arr);
   }
 }
 return arr;
}
console.log("Bubble sort is: ", bubbleSort([7, 4, 45, 35, 19, 28, 101, 83]));

// ---------------------------------
// Remove duplicates from this array [1,5,7,8,4,3,1,6,9,8,14]
function removeDuplicates(data) {
 let unique = [];
 data.forEach((element) => {
   if (!unique.includes(element)) {
     unique.push(element);
   }
 });
 return unique;
}
var arr = [1, 5, 7, 8, 4, 3, 1, 6, 9, 8, 14];
console.log(removeDuplicates(arr));

// ---------------------------------
// Check to see if a word is a palindrome (if you reverse a word and it is the same
word
// (mom, dad, bob, racecar, etc)

function isPalindrome(str) {
 let i;
 let len = str.length;
 for (i = 0; i < len / 2; i++) {
   if (str[i] !== str[len - 1 - i]) return false;
 }
 return true;
}

// ----------------------------------
// Return the length of the longest word in this sentence: “Those who can imagine
anything, can create the impossible.”
function findLongestWordLength(str) {
 let maxVal = 0;

 const wordArr = str.split(" ");

 for (let i = 0; i < wordArr.length; i++) {
   let word = wordArr[i];
   if (word.length > maxVal) {
     maxVal = word.length;
   }
 }
 return maxVal;
}
```