(1.) **Biggie Size** - Given an array, write a function that changes all positive numbers in the array to the string "big". Example: makeItBig([-1,3,5,-5]) returns that same array, changed to [-1, "big", "big", -5].

```
function makeItBig(arr){

for(var i = 0; i<arr.length; i++){
  if(arr[i]>0){
    arr[i]="big"
  }
}
console.log(arr);
}
makeItBig([-1,3,5,-5]);
```

(2.) **Print Low, Return High** - Create a function that takes in an array of numbers. The function should **print** the lowest value in the array, and **return** the highest value in the array.

```
function printLow(x){

  max = x[0];
  min = x[0];

   for(var i = 1; i<x.length; i++){
     if(x[i]<min){
       min = x[i];

     }
     if(x[i]>max){
       max = x[i];

     }

  }
  console.log(min);
  return max;
}

printLow([1,3,-2,7, -1,7]);
```

(3.) **Print One, Return Another** - Build a function that takes in an array of numbers. The function should **print** the *second-to-last* value in the array, and **return** the *first odd* value in the array.

```
function x(arr){
  for(var i=0; i<arr.length; i++){
    if(arr[i] % 2 ===1){
      break;
    }
  }
  console.log(arr[arr.length-2]);
  return arr[i];
}

x([0,2,3,4,6,5]);
```

(5.) **Count Positives** - Given an array of numbers, create a function to replace the last value with the number of positive values found in the array. Example, countPositives([-1,1,1,1]) changes the original array to [-1,1,1,3] and returns it.

```
function countDouble(arr, sum){
    for(var i=0; i<arr.length; i++){
      if(arr[i]>0){
        sum += arr[i];
      }
    }
  arr[arr.length-1] = sum;
  return arr;
}
countDouble([-1,1,1,1], 0);
```

(6.) **Evens and Odds** - Create a function that accepts an array. Every time that array has three odd values in a row, print "That's odd!". Every time the array has three evens in a row, print "Even more so!".

```
function evenOdd(arr){

  countOdd=0;
  countEven=0;

  for(var i=0; i<arr.length; i++){
    if(arr[i] % 2 ==1){
      countOdd++;
      countEven=0;
    }
    else{
      countEven++;
      countOdd=0;
    }
    if(countEven==3){
      console.log("Even more so")
    }
    if(countOdd==3){
      console.log("That's odd")
    }
  }
  return arr;
}

evenOdd([1,3,4,6,8]);
```

(7.) **Increment the Seconds** - Given an array of numbers **arr**, add 1 to every other element, specifically those whose index is odd (arr[1], arr[3], arr[5], etc). Afterward, console.log each array value and return **arr**.

```
function second(arr){
  var newArr = [];
  for(var i = 0; i<=arr.length; i++){
    if(i % 2 ===1){
      newArr.push(i);
    }
  }
  for(var j=0; j<arr.length; j++){
    arr[newArr[j]]++;
    console.log(arr[j]);
  }

  return arr;
}

console.log(second([1,1,3,3,5,5,7,7,9,9,11,11]));
```

(8.) **Previous Lengths** - You are passed an array (similar to saying 'takes in an array' or 'given an array') containing **strings**. Working within that same array, replace each string with a number - the *length* of the string at the previous array index - and return the array. For example, previousLengths(["hello", "dojo", "awesome"]) should return ["hello", 5, 4]. **Hint:** Can for loops only go forward?

```
function prevLength(arr){
  for(var i = arr.length-1; i>0; i--){
    arr[i] = arr[i-1].length;
  }
  return arr;
}

prevLength(["hello", "dojo", "awesome"]);
```

(9.) **Add Seven** - Build a function that accepts an array. Return a **new** array with all the values of the original, but add 7 to each. Do not alter the original array. Example, addSeven([1,2,3]) should return [8,9,10] in a new array.

```
function addSeven(arr){
  var newArr = [];
  for(var i = 0; i<arr.length; i++){
    newArr.push(arr[i]+7);
  }
  return newArr;
}
addSeven([1,2,3]);
```

(10.) **Reverse Array** - Given an array, write a function that reverses its values, in-place. Example: reverse([3,1,6,4,2]) returns the same array, but now contains values reversed like so... [2,4,6,1,3]. Do this **without** creating an empty temporary array. (Hint: you'll need to swap values).

```
function reverseArr(arr){
  var start = 0;
  var end= arr.length-1;
  while(start<end){
    var temp=arr[start];
    arr[start] = arr[end];
    arr[end] = temp;
    start++;
    end--;
  }
  return arr;
}

reverseArr([3,1,6,4,2]);
```

(11.) **Outlook: Negative** - Given an array, **create and return a new one** containing all the values of the original array, but make them all negative (not simply multiplied by -1). Given [1,-3,5], return [-1,-3,-5].

```
function allNegatives(arr){
  var newArr = [];
  for(var i = 0; i<arr.length; i++){
    if(arr[i]>0){
      newArr.push(arr[i]*-1);
    }
    else{
      newArr.push(arr[i]);
    }
  }
  return newArr;
}

allNegatives([1,-3,5,5,6,-9]);
```

(12.) **Always Hungry -** Create a function that accepts an array, and prints "yummy" each time one of the values is equal to "food". If no array values are "food", then print "I'm hungry" once.

```
function hunger(arr){
  var countHungry = 0;
  var countFood = 0;

  for(var i = 0; i<arr.length; i++){
    if(arr[i] == "food"){
      countFood++;
    }
    if(countFood>0){
      console.log("Yummy");
      break;
    }
    else if(arr[i] !=="food"){
      countHungry++;
    }
    if(countHungry>=arr.length){
      console.log("I'm hungry");
    }
  }
}
hunger(["one","two","fire","brick","food"]);
```

(13.) **Swap Toward the Center -** Given an array, swap the first and last values, third and third-to-last values, etc. Example: swapTowardCenter([true,42,"Ada",2,"pizza"]) turns the array into ["pizza", 42, "Ada", 2, true]. swapTowardCenter([1,2,3,4,5,6]) turns the array into [6,2,4,3,5,1]. No need to return the array this time.

```
function swapTowardCenter(arr){
  var first = 0;
  var last = arr.length-1;
  var third = 2
  var thirdLast=arr.length-3

  for(var i = 0; i<arr.length; i++){
    if(i === 0){
      var temp= arr[first];
      arr[first] = arr[last];
      arr[last] = temp;
    }
    if(i == 3){
      var hold= arr[third];
      arr[third] = arr[thirdLast];
      arr[thirdLast] = hold;
    }
  }
}
swapTowardCenter([true,42,"Ada",2,"pizza"]);
```

(14.)**Scale the Array -** Given an array **arr** and a number **num**, multiply all values in the array **arr** by the number **num**, and return the changed array **arr**. For example, scaleArray([1,2,3], 3) should return [3,6,9].

```
function scaleArray(arr,num){
  for(var i = 0; i<arr.length; i++){
    arr[i] = arr[i]*num;
  }
  return arr;
}

scaleArray([1,2,3], 3);
```