1.

const filterOutOdds = (…arguments) => arguments.filter(num => num % 2 === 0 )

2. findMin

Const findMin = (…nums) => Math.min(…nums)

3. mergeObjects

Const mergeObjects = (object1, object2) => ({…object1, …object2})

4. doubleAndReturnArgs

Const doubleAndReturnArgs = (arr, …args) => […arr], …args.map(vals => vals\*2)]

5. Slice and Dice

/\*\* remove a random element in the items array

and return a new array without that item. \*/

const removeRandom = (items) => {

let indexToRemove = Math.floor(Math.random() \* items.length);

let itemsCopy = […items];

itemsCopy.splice(indexToRemove, 1);

return itemsCopy;

}

/\*\* Return a new array with every item in array1 and array2. \*/

const extend = (array1, array2) => {

return […array1, …array2];

}

/\*\* Return a new object with all the keys and values

from obj and a new key/value pair \*/

const addKeyVal = (obj, key, val) => {

let newObject = {…obj};

newObject[key] = val;

return newObject;

}

/\*\* Return a new object with a key removed. \*/

const removeKey = (obj, key) => {

let newObject = {…obj};

delete newObject[key];

return newObject;

}

/\*\* Combine two objects and return a new object. \*/

const combine = (obj1, obj2) => {

return {…obj1, …obj2};

}

/\*\* Return a new object with a modified key and value. \*/

const update = (obj, key, val) => {

let newObject = {…obj} ;

newObject[key] = val;

return newObject;

}