**Advanced Functions: Assignment**

This checkpoint should take you between 30 and 60 minutes. If you spend longer than that on this checkpoint, reach out for help!

**Instructions**

Your goal for this checkpoint is to get the tests to pass. To do so, you will be writing five different functions related to park and user data.

**Dataset**

You will be working with two different sets of data for these functions: **parks** and **users**.

Parks is an array of objects, similar to this:

const parks = [

{

id: 1,

name: "Acadia",

areaInSquareKm: 198.6,

location: { state: "Maine" },

},

{

id: 2,

name: "Canyonlands",

areaInSquareKm: 1366.2,

location: { state: "Utah" },

},

{

id: 3,

name: "Zion",

areaInSquareKm: 595.9,

location: { state: "Utah" },

},

];

Users is an object with a number of keys that represents each user. It looks something like this:

const users = {

"karah.branch3": {

visited: [2],

wishlist: [1, 3],

},

"dwayne.m55": {

visited: [2, 3],

wishlist: [1],

},

};

**getParksByState**

This function takes in an array of parks and a specific state and returns all the parks that are in that state.

getParksByState(parks, "Utah");

*/\**

*[*

*{*

*id: 2,*

*name: "Canyonlands",*

*areaInSquareKm: 1366.2,*

*location: { state: "Utah" },*

*}*

*]*

*\*/*

**getWishlistParksForUser**

This function takes in an array of parks, all users, and a username. It then returns an array of park objects related to the wishlist of the user with that username. Each number in the wishlist refers to the id of one of the parks.

getWishlistParksForUser(parks, users, "karah.branch3");

*/\**

*[*

*{*

*id: 1,*

*name: "Acadia",*

*areaInSquareKm: 198.6,*

*location: { state: "Maine" },*

*}*

*]*

*\*/*

**userHasVisitedAllParksInState**

This function returns a boolean that represents whether or not a user has visited all parks in the parks array from a given state.

userHasVisitedAllParksInState(parks, users, "Utah", "dwayne.m55"); *//> true*

userHasVisitedAllParksInState(parks, users, "Utah", "karah.branch3"); *//> false*

**userHasVisitedParkOnWishlist**

This function takes in the list of users and two usernames. If the first user has visited *any* of the parks represented by the second user's wishlist, return true. Otherwise, return false.

userHasVisitedParkOnWishlist(users, "dwayne.m55", "karah.branch3"); *//> true*

userHasVisitedParkOnWishlist(users, "karah.branch3", "dwayne.m55"); *//> false*

**getUsersForUserWishlist**

This function returns all the usernames who have visited any park on the given user's wishlist.

getUsersForUserWishlist(users, "karah.branch3"); *//> ["dwayne.m55"]*

getUsersForUserWishlist(users, "dwayne.m55"); *//> []*