**Most Wanted User Stories**

**100 points**

**Goal:** You have been contracted to build a prototype for a person search for a top-secret government project. You have been given access to an array of objects representing individuals. The prototype should just use window.prompt and window.alert for the User Interface (UI). All results should be printed through the window.alert and window.prompt. Although this isn’t typical in production, you may use only two files for this project, an HTML file and a JS file for the application.

**Technologies:** JavaScript

**User stories:**

**(5 points):** As a developer, I want to make consistent commits with good, descriptive messages.

**(5 points):** As a developer, I want to run validation on any user input, ensuring that a user is re-prompted when they provide invalid input.

**(10 points):** As a user, I want to be able to search for someone based on a single criterion. (You should be able to find and return a list of people who match the search)

**(20 points):** As a user, I want to be able to search for someone based on 2-5 criteria. (I.e if you search for Gender: male and Eye Color: blue, you should get back a list of people who match the search)

trait search two

**(15 points):** As a user, I want to be able to look up someone’s information after I find them with the program (display values for the various traits of the found person).

To Do:

In each search, save search results to a variable, and pass into this method.

**(25 points):** As a user, I want to be able look up someone’s descendants after I find them with the program (display the names of the descendants), **using recursion**.

**(20 points):** As a user, I want to be able look up someone’s immediate family members after I find them with the program (display the names of the family members and their relation to the found person. Parents, spouse, and siblings).

function findFamilyMembers(id, people){

let personsId == person.id === "(id)"; // firstname//lastname/etc

let personsSpouse = person.currentSpouse; (id number)

let personsParents = person.parents; (number)

let personsSiblings == (find people where( person.parents === personsParents)

let spouseOfPerson = people.currentSpouse.includes(id == personId);

let siblingsOfPerson = people.parents.filter(parents === personsParents);

let parentsOfPerson = people.id.filter(id === personsParents);

//display

displayPersonsFamily(person, spouseOfPerson, siblingsOfPerson, parentsOfPerson, people)

}

function displayPersonsFamily(person, spouseOfPerson, siblingsOfPerson, parentsOfPerson, people){

// print all of the information about a person:

// height, weight, age, name, occupation, eye color.

let personInfo = "First Name: " + person.firstName + "\n";

personInfo += "Last Name: " + person.lastName + "\n";

let spouseInfo = "Spouse First Name: " + spouseOfPerson.firstName + "\n";

spouseInfo += "Spouse Last Name: " + spouseOfPerson.lastName + "\n";

let siblingsCount = siblingsOfPerson.count;

let siblingsCountSt = siblingsCount.tostring();

let parentCount = parentsOfPerson.count;

let parentsCountSt = parentCount.tostring();

var siblingInfo;

var paren

switch(siblingsCountSt){

case "1":

let siblingInfo1 = "Sibling First Name: " + siblingOfPerson.firstName + "\n";

siblingInfo1 += "Last Name: " + siblingOfPerson.lastName + "\n";

var siblingInfo = siblingInfo;

break;

case "2":

let siblingInfo2 = "Sibling First Name: " + siblingOfPerson.atindex(0).firstName + "\n";

siblingInfo2 += "Sibling Last Name: " + siblingOfPerson.atindex(0).lastName + "\n";

siblingInfo2 += "Sibling First Name: " + siblingOfPerson.atindex(1).firstName + "\n";

siblingInfo2 += "Sibling Last Name: " + siblingOfPerson.atindex(1).lastName + "\n";

siblingInfo = siblingInfo2;

break;

case "3":

let siblingInfo3 = "Sibling First Name: " + siblingOfPerson.atindex(0).firstName + "\n";

siblingInfo3 += "Sibling Last Name: " + siblingOfPerson.atindex(0).lastName + "\n";

siblingInfo3 += "Sibling First Name: " + siblingOfPerson.atindex(1).firstName + "\n";

siblingInfo3 += "Sibling Last Name: " + siblingOfPerson.atindex(1).lastName + "\n";

siblingInfo3 += "Sibling First Name: " + siblingOfPerson.atindex(2).firstName + "\n";

siblingInfo3 += "Sibling Last Name: " + siblingOfPerson.atindex(2).lastName + "\n";

siblingInfo = siblingInfo3;

break;

}

switch(parentCountSt){

case "1":

let parentInfo1 = "Parent First Name: " + parentsOfPerson.firstName + "\n";

parentInfo1 += "Last Name: " + parentsOfPerson.lastName + "\n";

var parentInfo = parentInfo1;

break;

case "2":

let parentInfo2 = "Parent First Name: " + parentsOfPerson.atindex(0).firstName + "\n";

parentInfo2 += "Parent Last Name: " + parentsOfPerson.atindex(0).lastName + "\n";

parentInfo2 += "parent First Name: " + parentsOfPerson.atindex(1).firstName + "\n";

parentInfo2 += "Sibling Last Name: " + parentsOfPerson.atindex(1).lastName + "\n";

parentInfo = parentInfo2;

break

}

}

alert(personInfo, spouseInfo, sinblingInfo, parentInfo );

}