

3 a.**3.a.i.**

The overall purpose of the program is to display the user some specific information on female legislators based on the category - Female Senate Democrats or State - they choose.

3.a.ii.

The functionality the video illustrates is by taking in two choices the user selects and providing some information based on that. For example, if the user chooses the button labeled State - Most Representative on the homepage and then chooses a category such as Female Senate Republicans, the program will display Minnesota.

3.a.iii.

The user will choose between three buttons (State - Most Representative, State - Least Representative, and Most Representative in a State - Senate vs House) which are inputs. Depending on the button the user chooses, the program will take them to the corresponding screen, which is the output. Then the user will select something in the dropdown, which is another input, and the answer is the output.

3 b.**3.b.i.**

```
var states = getColumn(▼ "Female State Legislators", ▼ "State");
```

3.b.ii.

```
function findMost() {
  var enter = getText("pickCategory");
  var category = getColumn("Female State Legislators", enter);
  var max = 0;
  var mostName;
  for(var i = 0; i < category.length; i++){
    if(category[i] > max) {
      max = category[i];
      mostName = states[i];
    }
  }
  setText("results2", ("The state with the most " + enter) + " is " + mostName);
}
```

3.b.iii.

The name of the list being processed in this response is called Female State Legislators. This list contains all of the information in order to provide the user with the answer they are looking for based on what they input into the app.

3.b.iv.

The Female State Legislators represent the list from which the answers are deduced. The answers are based on what the user chooses throughout the app. For example, if the user selects State - Most Representatives and in the dropdown box chooses Female State Independents, then the answer the user will receive in the text area is

The state with the most Female State Independents is Alaska

. In other words, the name of the list, Female State Legislators, contains the data used to provide the user with the necessary information.

3.b.v.

The Female State Legislators list manages complexity because it allows for the user to choose between many different categories and states while also controlling what the user inputs into the app. It also ensures that the user does not choose a category or a state that the program cannot provide an answer for. If the program did not contain the list, then the app would be much more complex because it would have to take into account all of the possibilities the user can input, which can even be a thousand.

3 c.**3.c.i.**

```
function housevssenate(input) {
  var stateNames = getColumn("Female State Legislators", "State");
  var senateTotals = getColumn("Female State Legislators", "Female Senate Total");
  var houseTotals = getColumn("Female State Legislators", "Female House Total");
  var cumulativeSenate = 0;
  var cumulativeHouse = 0;
  for (var i = 0; i < stateNames.length; i++) {
    if (stateNames[i] == input) {
      cumulativeSenate = cumulativeSenate + senateTotals[i];
      cumulativeHouse = cumulativeHouse + houseTotals[i];
    }
    if (cumulativeSenate > cumulativeHouse) {
      setText("stateResults", "Senate");
    } else if (cumulativeSenate < cumulativeHouse) {
      setText("stateResults", "House");
    } else {
      setText("stateResults", "Equal Representation");
    }
  }
}
```

3.c.ii.

```
onEvent (▼ "pickastate", ▼ "change", function() {
  var input = getText (▼ "pickastate");
  housevssenate (input);
});
```

3.c.iii.

The selected procedure states whether the House or the Senate has more female representatives from 1981 to 2019 in the list. It takes into account the state the user selects in order to give the result.

3.c.iv.

For the program output that the House or the Senate has more female representation from 1981 to 2019 in the list for a particular state, the program first has to choose three columns in the list: State, Female Senate Total, and Female House Total; these three columns started the function with a parameter named input because the user has to input a state. Two variables were created that stated the value for the Female Senate Total and Female House Total were starting out as zero. The program then includes a for loop. This ensures that the name of the state chosen by the user fits the program's appropriate length. Then, the if statement which is in the for loop adds up the Female Senate Total and Female House Total columns separately. After this, the if else-if else statement decides whether the sum of the Female Senate Total column is greater than the Female House Total column and vice versa. Based on this, the app tells the user that the Senate or House has more female representation or that there is equal representation.

3 d.**3.d.i.**

First call:

The user could select Arizona as the state in the dropdown to see whether the House or the Senate had more female representation between 1981 and 2019.

Second call:

To find whether the House or the Senate had more female representation between 1981 and 2019 in a particular state, the user could select Delaware as the state in the dropdown.

3 d.ii.**Condition(s) tested by first call:**

When the user selects Arizona as the state, the program will add up the numbers in the Female Senate Total and Female House Total columns and compare the two.

Condition(s) tested by second call:

After the state, Delaware, is selected by the user, the program will find two columns: Female Senate Total and Female Total columns. Then, it will add the values in the two columns and see which one is higher/lower.

3.d.iii.**Results of the first call:**

After the program adds the two columns, it will realize the number in the Female House Total is larger than the Female Senate Total, so in the text area, the word

House

will display for the user to show that there is more female representation in the House versus the Senate for that particular state in between 1981 and 2019.

Results of the second call:

When the program finds the sum of the two columns, it will come to the conclusion that both of the values in the columns are equal, and it will display

Equal Representation for the user.