

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

The purpose is to display some of my favorite song lyrics and movie lines for the user.

3.a.ii.

The video demonstrates the user clicking on each of the buttons and the correct information being output.

3.a.iii.

The input shown is the user clicking one of the buttons on screen. The output, a list of either song lyrics or movie lines, is displayed as text.

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

```

1 //The sources of the song lyric and movie quotes are included with the quote by citing the Song name/Artist or Movie Title
2 var quoteList = getColumn("quoteList", "quote");
3 var quoteType = getColumn("quoteList", "type");
4 var type;
5 onEvent("movieButton", "click", function() {
6     type = "movie line";
7     filter(type);

```

3.b.ii.

```

17 for (var i = 0; i < quoteList.length; i++) {
18     if (quoteType[i] == userChoice) {
19         appendItem(filteredList, quoteList[i]);
20         output = output + quoteList[i] + "\n";
21     }
22 }
23 setText("outputText", output);
}

```

3.b.iii.

The name of the list being processed is quoteType.

3.b.iv.

The data in this list stores the type of each of my quotes (either song lyric or movie line). This information is used to filter my list and sort the user's choice into an empty list to be output.

3.b.v.

My list makes it easier to sort the quotes depending on the type of quote. Without my list differentiating the two options, all of the quotes would be printed out together when either of the buttons are pressed. This list also makes it possible to easily add additional movie lines or song lyrics to be printed to the data set without complicating the code at all.

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

```

13 function filter(userChoice) {
14     var filteredList = [];
15     var output = "";
16     for (var i = 0; i < quoteList.length; i++) {
17         if (quoteType[i] == userChoice) {
18             appendItem(filteredList, quoteList[i]);
19             output = output + quoteList[i] + "\n";
20         }
21     }
22     setText("outputText", output);
23 }

```

3.c.ii.

```

5 onEvent("movieButton", "click", function() {
6     type = "movie line";
7     filter(type);
8 });
9 onEvent("songButton", "click", function() {
10    type = "song lyric";
11    filter(type);
12 });
13

```

3.c.iii.

The filter function contributes to the overall functionality by filtering one-by-one through my list in order to output a filtered list that matches the user input.

3.c.iv.

The task my function accomplishes is filtering my quotes, sorting them into an empty list, and then outputting that list. My function begins by creating an empty list called `filteredList` (line 14) which will be used to collect the desired quote type. I also created a blank variable called `output` (line 15) which is used in tandem with `filteredList` to create a neat display when the list is output. In order to filter the list, a for loop is used (line 16) which traverses `list quoteType`. The if statement on the next line (line 17) checks to see if the quote type matches the user selection. If they do match, the item is appended to our empty list, `filteredList` (line 18). Next, on line 19, my variable `output` is defined, which includes adding a line break for better display on the user interface. The output is then displayed onto the screen by using `setText` on line 22.

3 d.

3.d.i.

First call:

```
filter("movie line");
```

Second call:

```
filter("song lyric");
```

3 d.ii.

Condition(s) tested by first call:

The argument passed through the filter function in this case would be "movie line". An if loop in my procedure tests to see whether the user choice calls for movie lines or song lyrics.

Condition(s) tested by second call:

The argument passed through the filter function in this case would be "song lyric". An if loop in my procedure tests to see whether the user choice calls for movie lines or song lyrics.

3.d.iii.

Results of the first call:

Movie lines are selected and added to my empty list named `filteredList`. This list is then output into the text box on the screen.

Results of the second call:

Song lyrics are selected and added to my empty list named `filteredList`. This list is then output into the text box of the screen.