

3a. Provide a written response that:

(Approx. 150 words, for all subparts of 3a combined)

- i. Describes the overall purpose of the program;

The purpose of the Word Game Helper app is to assist users in finding words to help in various games like scrabble or crossword puzzles.

- ii. Describes what functionality the video illustrates;

The video shows how the user selects the length of the word and the first letter from different dropdowns.

- iii. Describes the input and output of the program shown in the video

The user inputs a value to the program using the dropdowns. The output, which is a list of words that meet the chosen conditions, is displayed on the screen.

3b. Capture and paste two program code segments you developed during the administration of this task which contain a list (or other collection type) being used to manage complexity in your program.

- i. The first program code segment must show how data has been stored in the list.

```
1 | var wordList = getColumn("words", "Word");
```

- ii. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

```

15 function filter(len, letter){
16     showElement("waitingImage");
17     filteredWordList = [];
18     setText("output", "");
19
20     for(var i=0; i<wordList.length; i++){
21         if(wordList[i].length == len && wordList[i].substring(0,1)==letter){
22             appendItem(filteredWordList, wordList[i]);
23         }
24     }
25
26     if(filteredWordList.length == 0){
27         appendItem(filteredWordList, "No Options Available");
28     }
29
30     hideElement("waitingImage");
31     setText("output", filteredWordList.join(", "));
32 }

```

Then, provide a written response that does all three of the following:

- iii. Identifies the name of the list being processed in this response

On line 1, a list called wordList collects all of the words from the "words" database and stores them as a list. Each word is stored as a string.

- iv. Identifies what the data contained in the list is representing in your program

The wordList is used in the program to show users suggested words of various lengths starting with a given letter. For example, if the user selects a length of 2 and the letter "b", the filter function is called in which the wordList is traversed. Every element that is a length of two and start with a "b" is added to a new filtered list. This list is what is outputted to the user.

- v. Explains how the selected list manages complexity in your program by explaining why your program code could not be written, or how it would be written differently, if you did not use the list

The wordList manages complexity because it allows any number of words to be stored in the list instead of using individual variables for each word, which would then be checked one by one to see if it met the requirements. The program would extend from 32 lines long to thousands of lines long to account for all of these extra variables. If words are added or removed to the dataset this list pulls from, nothing will need to change about the code. The list allows the program to work for any number of words since the filter function will traverse the entire list of words no matter its length.

3c. Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure which implements an algorithm used in your program and a call to that procedure.

(Approx. 200 words, for all subparts of 3c combined, excluding program code)

- i. This first program code segment must be a student-developed procedure that:
 - Defines the procedure's name and return type (if necessary)
 - Contains and uses one or more parameters that have an effect on the functionality of the procedure; and
 - Implements an algorithm that includes sequencing, selection and iteration.

```
15 function filter(len, letter){
16   showElement("waitingImage");
17   filteredWordList = [];
18   setText("output", "");
19
20   for(var i=0; i<wordList.length; i++){
21     if(wordList[i].length == len && wordList[i].substring(0,1)==letter){
22       appendItem(filteredWordList, wordList[i]);
23     }
24   }
25
26   if(filteredWordList.length == 0){
27     appendItem(filteredWordList, "No Options Available");
28   }
29
30   hideElement("waitingImage");
31   setText("output", filteredWordList.join(", "));
32 }
```

- ii. The second program code segment must show where the student-developed procedure is being called in your program

```
4 filter(getNumber("lengthDropdown"), getText("letterDropdown").toLowerCase());
```

Then, provide a written response that does both of the following:

- iii. Describes in general what the selected procedure does and how it contributes to the overall functionality of the program

The filter function is necessary in order for the program to output a filtered word list to users which meets their chosen requirements.

- iv. Explains in detailed steps how the algorithm implemented in the selected procedure accomplishes its task. Your explanation must be detailed enough for someone else to recreate it.

Lines 16 and 30 show and hide an image to let to the user know the program is working. Once the list has been filtered, the image is hidden. To filter the list, a for loop is used (line 20) which traverses wordList. The if statement on Line 21 checks to see if the element at the index is the required length and starts with with the required letter. If it does, the element is added to the filteredWordList (line 22). After the traversal of the list is finished, if the filteredWordList is empty, a string is added to it to let the user know that there are no options available (line 27). Finally, in line 31 the filtered list is displayed to the user, with all the elements joined together with a comma in between each one.

3d. Provide a written response that does all three of the following:

(Approx. 200 words, for all subparts of 3d combined, excluding program code)

- i. Describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute;

First call:

Let's suppose the user selects from the dropdowns a length of 3 and the letter "d".

Second call:

Another example would be if the function was called with the arguments 1 and "b".

- ii. Describes what condition(s) is being tested by each call to the procedure

Condition(s) tested by the first call:

The arguments passed through the filter function would be 3 and "d" for the parameters *len* and *letter*. In the for loop on line 21, there is an if statement which checks each element in wordList to see if it has a length of 3 and starts with the letter "d". For example, when the element containing "dog" is examined, the conditions (length of 3, first letter "d") is met and therefore the code segment inside of the if statement (line 22) runs and "dog" is added to the filtered list. The for loop continues running checking other elements.

Condition(s) tested by the second call:

In this case, again the for loop on line 21 is used to traverse the wordList. Each element is checked. When "a" is examined, the conditions (length of 1, first letter "b") is not met.

Therefore 21-23 are skipped and the for loop continues on to the next round.

iii. Identifies the result of each call.

Result of the first call:

After the for loop finishes running, any words that had a length of 3 and started with the letter "d" have been added to the filtered list which is displayed to the user.

Result of the second call:

Ultimately, no word is found with these conditions, and therefore the filtered list is blank until lines 26-28 where a string is added to let the user know no word was found that met the conditions.