Create PT Prep ('21-'22)

Create PT Review the Task

Resources

Create PT 20-21 College Board Sample A - Score: 6/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**1**	**1**	**1**	**1**	**1**	**1**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

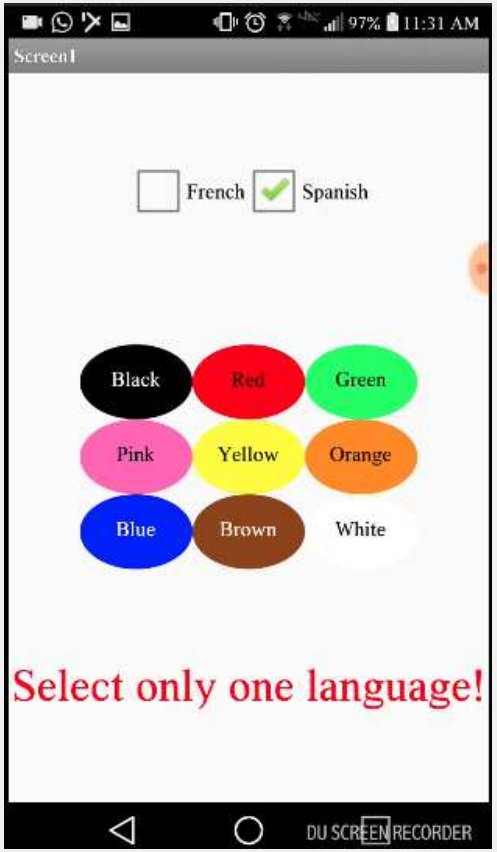
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

This program was created in MIT App Inventor to address the issue of learning new languages. Here it teaches the user how to say different colors, where the user inputs what language it wants to hear, either Spanish or French, and then taps on a color, prompting the program to output the audio for that certain color.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A, CRD-2B

● The video demonstrates the running of the program including:
○ input
○ program functionality
○ output
AND
● The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

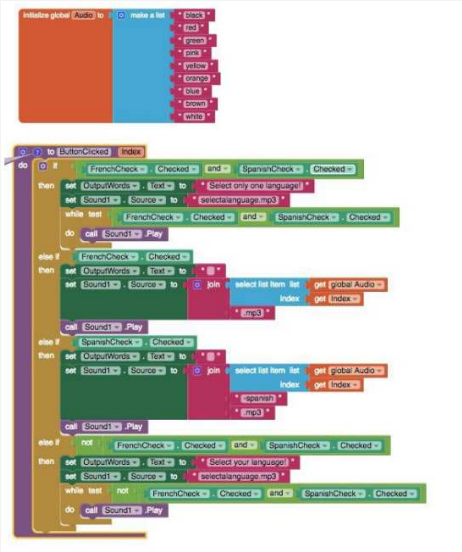
Do NOT award a point if the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="3">The response earned the point for this row, meeting all six criteria.
● The video demonstrates the program receiving user color selection in both French and Spanish as input and producing as output the associated audio response of that color’s pronunciation in the selected language. This satisfies the first three criteria for the video.
● The program’s purpose is to “address the issue of learning new languages.”
● The functionality demonstrated in the video is “where the user inputs what language it wants to hear, either Spanish or French, and then taps on a color, prompting the program to output the audio for that certain color.”
● The input and output demonstrated in the video are described as, “In the video, it shows an example of the user clicking on the Spanish checkbox and playing the audio for red and blue.”</td></tr>
</table>

This allows users to quickly learn how to say colors in another language through interaction and output of audio. In the video, it shows an example of the user clicking on the Spanish checkbox and playing the audio for red and blue. The user can hear what it sounds like, and thus learn how to say it correctly. If the user accidentally inputs no language or both, the program will catch the error and notify the user.	

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

Student Response	**Scoring Guidelines**



The data contained in the list Audio is the list of available color names. It represents all the colors a user can pick for the program in English. These are used to create the corresponding Spanish or French audio files based on which language is selected. When a button is pressed, it will get the English color word from the index of the color in the list. Here, the language the user has chosen does not matter. The program will then create the audio file name for the Spanish or French audio based on what language the user has check marked by manipulating the text (for example, adding"-spanish" to the end if they selected spanish) and then adding ".mp3" to the end

Row and Task	Decision Rules
Row 2 - Response 3b **Data Abstraction** **3.B, AAP-1.C** The written response: ● includes two program segments: ○ one that shows how data has been stored in this list (or other collection type) ○ one that shows the data in this same list being used as part of fulfilling the program's purpose. ● identifies the name of the variable representing the list being used in this response ● describes what the data contained in this list is representing in the program.	**Consider ONLY written response 3b when scoring this point.** **Requirements for program code segments:** ● The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment. ● If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned. **Do NOT award a point if the following is true:** ● The use of the list is trivial and does not assist in fulfilling the program's purpose.
The response earned the point for this row, meeting all three criteria. ● Two distinct code segments are provided: one showing storage of data in a list named Audio; and a second one showing the use of Audio to process output audio to fulfill the program's purpose. ● The name of the list is identified as Audio. ● The response states that the data "represents all the colors a user can pick for the program in English. These are used to create the corresponding Spanish or French audio files based on which language is selected."	
Row 3 - Response 3b **Managing Complexity** **3.C, AAP-3.C** The written response: ● includes a program code segment that shows a list being used to manage complexity in the program.	**Consider ONLY written response 3b when scoring this point.** **Responses that do not earn row 2, may still earn this row.** **Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected

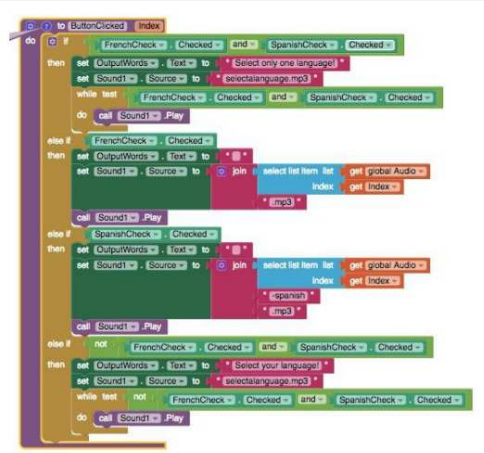
in order to call the correct audio file. For the program to function without lists in general, each button will have to call the individual audio file, meaning we would need to have buttons for every color and language combination making the code and user interface more complex.	• explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	without explicitly identifying the code segments containing the list). • The written response does not name the selected list (or other collection type). • The use of the list is irrelevant or not used in the program. • The explanation does not apply to the selected list. • The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. • The solution without the list is implausible, inaccurate, or inconsistent with the program. • The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. • The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response earned the point for this row, meeting both criteria. • The response includes a program code segment that shows the audio list being used to manage complexity in the program, because the list access and index enable the correct setting of a filename for audio file output. • The response explains how the list Audio manages complexity in the program. It states, "When a button is pressed, it will get the English color word from the index of the color in the list…. The program will then create the audio file name for the Spanish or French audio based on what language the user has check marked by manipulating the text (for example, adding '-spanish' to the end if they selected spanish [sic]) and then adding '.mp3' to the end in order to call the correct audio file." The response also states that the use of lists manages complexity in the project, explaining that "without lists in general, each button will have to call the individual audio file, meaning we would need to have buttons for every color and language combination making the code and user interface more complex."	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="3"> 

This procedure helps to contribute to the overall functionality of the program by selecting the correct audio file based on what language and color the user has selected. Since this algorithm is needed every time a user presses a button, the procedure helps the overall efficiency of the program by having the code located in one location that the program repeatedly calls on. When a user presses a certain color button, the button returns an index pertaining to the main audio list of colors. Then, the procedure takes the index of the color as a parameter and selects the respective list</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>Row 4 - Response 3c

Procedural Abstraction

3.B, AAP-3.C

The written response:
● includes two program code segments:
○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure.
○ one showing where the student-developed procedure is being called.

● describes what the identified procedure does and how it contributes to the overall functionality of the program.</td><td>Consider ONLY written response 3c when scoring this point.

Requirements for program code segments:
● The procedure must be student developed, but could be developed collaboratively with a partner.
● If multiple procedures are included, use the first procedure to determine whether the point is earned.

Do NOT award a point if any one or more of the following is true:
● the code segment is an event handler; OR
● the code segment consisting of the procedure is not included in the written response section; OR
● the written response describes what the procedure does independently without relating it to the overall function of the program.</td></tr>
<tr><td colspan="2">The response earned the point for this row, meeting both criteria.
● The response includes a student-developed procedure ButtonClicked that contains a parameter index. The parameter index is used in the procedure. Additionally, the response includes an example call to the procedure ButtonClicked that passes the argument “3” to the parameter.
● The response describes the purpose of ButtonClicked by stating that it selects “the correct audio file based on what language and color the user has selected.” The response describes how ButtonClicked contributes to the overall functionality of the program by stating it “helps the overall efficiency of the program by having the code located in one location that the program repeatedly calls on.”</td></tr>
<tr><td>Row 5 - Response 3c

Algorithm Implementation</td><td>Consider ONLY written response 3c when scoring this point.

Responses that do not earn row 4 may still earn this row.</td></tr>
</table>

<table>
<tr>
<td rowspan="2">element, which is a string that contains the certain color. Then, depending on the language, the procedure will append a language identifier (for example, “-spanish”) and then add “.mp3”. It will then use this to call a certain color from the database of audio files that is named accordingly. The procedure is able to take parameters and inputs from the user and then output them as such by constructing certain audio file names and then pulling them from the database to play. This action demonstrates selection, and sequencing is when the procedure is able to order tasks accordingly and call from a list. Iteration is used when the procedure senses that there are no languages or both languages selected, and thus repeats the audio of “Select a language!” until the user does so.</td>
<td>2.B, AAP-2.H, AAP-2.K

The written response:
● includes a student-developed algorithm that includes:
○ sequencing
○ selection
○ iteration

● explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.</td>
<td>Requirements for program code segments:
● The algorithm being described can utilize existing language functionality or library calls.
● An algorithm that contains selection and iteration, also contains sequencing.
● An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point.
● Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned.
● If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included.

Do NOT award a point if any one or more of the following is true:
● The response only describes what the selected algorithm does without explaining how it does it.
● The description of the algorithm does not match the included program code.
● The code segment consisting of the selected algorithm is not included in the written response.
● The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm).
● The use of either the selection or the iteration is trivial and does not affect the outcome of the program.</td>
</tr>
<tr>
<td colspan="2">The response earned the point for this row, meeting both criteria.
● The student-developed algorithm within procedure ButtonClicked includes sequencing, selection (if...then statement), and iteration (while test...do).
● The response explains how the algorithm works. It states that it “takes the index of the color as a parameter and selects the respective list element, which is a string that contains the certain color. Then depending on the language, the procedure will append a language identifier (for example, ‘- spanish’) and then add ‘.mp3.’” The response goes on to describe that iteration is used “when the procedure senses that there are no languages or both languages selected, and thus repeats the audio of ‘Select a language!’ until the user does.”</td>
</tr>
</table>

3d. Provide a written response that:

- 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; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">The test cases are based on the conditions of what language is selected and what button for what color is pressed, which is represented by the parameter “index”. We would want to check the program for both Spanish and French. For the language Spanish, we could select one of the colors, for example Orange, and the program should give us the correct translation for Spanish. For the language French, we could select one of the colors, for example Orange, and the program should give us the correct translation for French. To further test the program, we should select another color, for example green, and the program should give us the correct translation for the pre-selected language. Each of these test cases executes different parts of the algorithm, going by the condition of which checkboxes are checked for which languages.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

Testing

4.C, CRD-2.J

The written response:
describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.
describes the condition(s) being tested by each call to the procedure.
identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
A procedure is not identified in written response 3c or the procedure does not have a parameter.
The written response for 3d does not apply to the procedure in 3c.
The two calls cause the same segment of code in the algorithm to execute even if the result is different.
The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
The identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="2">The response earned the point for this row, meeting all three criteria.
The response describes two calls to the procedure: one for Spanish, with the color orange; and one for French, with the color orange.
The response describes the conditions as being whether the user has selected Spanish or French. “For the language Spanish, we could select one of the colors, for example Orange [sic],” and “For the language French, we could select one of the colors, for example Orange [sic].”
The response describes the results being tested as the “correct translation for Spanish” and the “correct translation for French.”</td></tr>
</table>

Create PT 20-21 College Board Sample C - Score: 4/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**1**	**1**	**0**	**1**	**0**	**1**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

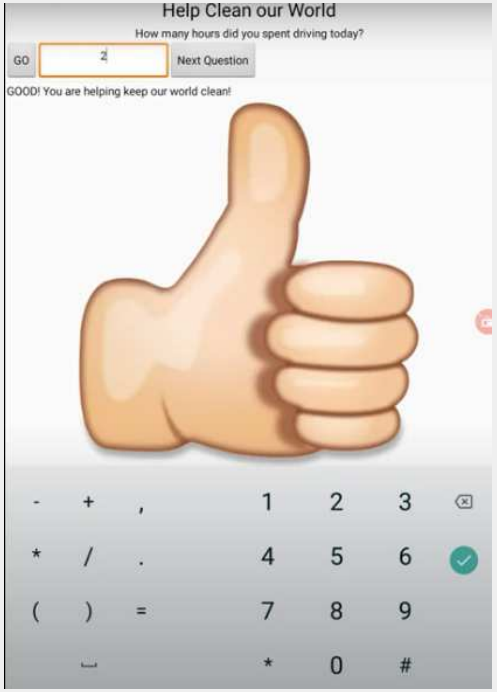
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

My app is intended to lessen the effects of global warming and to decrease the amount of non recyclable waste we produce. In the video, you can see the question label that displays a random list item from “global questions list”. Once the user reads the question, they answer it using the text box. Depending on the users input, the program will produce various outputs, which it pulls from “global solutions list”. If the program decides the user spends too much time driving, using their AC, or</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A, CRD-2B

● The video demonstrates the running of the program including:
○ input
○ program functionality
○ output
AND
● The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

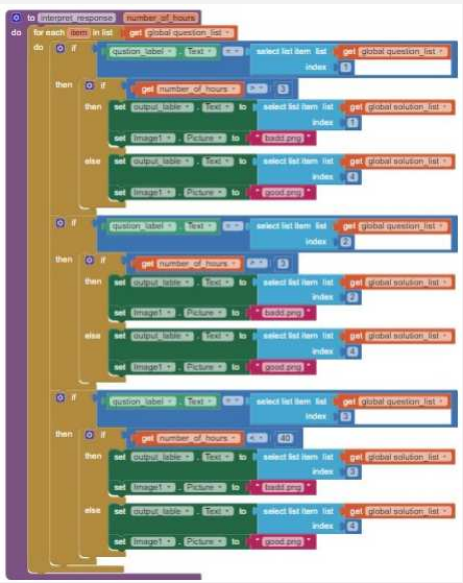
Do NOT award a point if the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="3">The response earned the point for this row, meeting all six criteria.
● The video demonstrates the running of the program, including input (user answering questions), functionality (processing of user input), and output (statement and visual “thumbs-up” or “thumbsdown”). This satisfies the first three criteria for the video.
● The response describes the program’s overall purpose as being “to lessen the effects of global warming and to decrease the amount of non recyclable [sic] waste we produce.”
● The response describes the functionality as follows: “displays a random list item from ‘global questions list’.”
● The response describes the input and output as “Once the user reads the question, they answer it using the text box. Depending on the users [sic] input, the program will produce various outputs, which it pulls from ‘global solutions list,’” and, “Also, a thumbs down image is displayed, unless the input indicates they are being good with their emissions. In that case, a thumbs up is displayed and ‘output label’ tells the user they are doing good and to keep it up.”</td></tr>
</table>

doesn't recycle enough, they will notify them by displaying an output on "output label" that tells them what they can do to lessen their carbon footprint. Also, a thumbs down image is displayed, unless the input indicates they are being good with their emissions. In that case, a thumbs up is displayed and "output label" tells the user they are doing good and to keep it up.	

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

Student Response	**Scoring Guidelines**	
	**Row and Task**	**Decision Rules**
	Row 2 - Response 3b **Data Abstraction** **3.B, AAP-1.C** The written response: ● includes two program segments: ○ one that shows how data has been stored in this list (or other collection type) ○ one that shows the data in this same list being used as part of fulfilling the program's purpose.	**Consider ONLY written response 3b when scoring this point.** **Requirements for program code segments:** ● The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment. ● If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned. **Do NOT award a point if the following is true:** ● The use of the list is trivial and does not assist in fulfilling the program's purpose.

<table>
<tr>
<td rowspan="3"> 

The data in “questions_list” are questions that the program displays for the user. The questions are then interpreted by the user and the user gives the program its input through the text box. The procedure check through “questions list” to see which element in the list matches the question that is displayed on “question label” so that it can provided the right “output label” based on the “number of hours” that was input. The use of the questions_list manages complexity in my program, because the program would be more complicated if I had to type the question into the if statements.</td>
<td>

identifies the name of the variable representing the list being used in this response
describes what the data contained in this list is representing in the program.

</td>
<td></td>
</tr>
<tr>
<td colspan="2">The response earned the point for this row, meeting all three criteria.

The response includes program code segments for initialization of two named lists, solution_list and question_list, as well as a code segment showing how the data in both lists are processed as a part of fulfilling the program’s purpose of questioning the user and evaluating responses.
The response identifies the list to be considered as question_list, so this is the list that was used to determine the score.
The response describes the data in question_list to be “questions that the program displays for the user.”

</td>
</tr>
<tr>
<td>
Row 3 - Response 3b

Managing Complexity

3.C, AAP-3.C

The written response:

includes a program code segment that shows a list being used to manage complexity in the program.
explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.

</td>
<td>
Consider ONLY written response 3b when scoring this point.

Responses that do not earn row 2, may still earn this row.

Do NOT award a point if any one or more of the following is true:

The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list).
The written response does not name the selected list (or other collection type).
The use of the list is irrelevant or not used in the program.
The explanation does not apply to the selected list.
The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program.
The solution without the list is implausible, inaccurate, or inconsistent with the program.
The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier.

</td>
</tr>
</table>

		● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response DOES NOT earn the point for this row. The response does not meet either of the criteria. ● The procedure, interpret_response, shows the list question_list being used; however, the value of each index in the list that is being stored in item is never used, making the list irrelevant. Instead, the list access and processing have been hard-coded based on list index number and do not manage complexity in the program as written, since the code has not been made easier to maintain and changes to the size of the list would require significant modifications to the code. ● The response states, “The use of the question_list manages complexity in my program, because the program would be more complicated if I had to type the question into the if statements.” However, the code only uses lists to replace the question strings in a hard-coded manner, so the use of the list is irrelevant. Additionally, changes to the size of the list (i.e., the number of questions) would necessitate significant modifications to the code.	

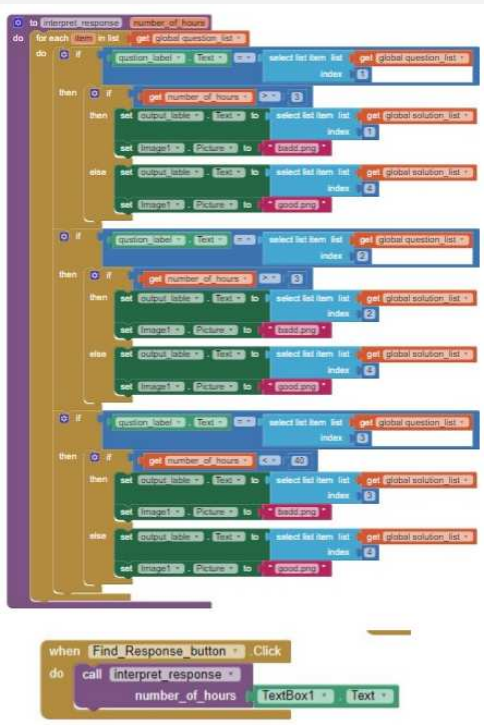
3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	**Scoring Guidelines**

<table>
<tr>
<td rowspan="3"> 

The procedure “interpret_response” has a parameter and selects the response for the user based on the question and the user input. Without it, my program wouldn’t function at all. So, the program displays the question, the user interprets it and then feeds the program data, which is their response to the question. The data that the user inputs to the program is the text from the text box. That text is the parameter for procedure “interpret response”, in the procedure, the user's input is called “number of hours”. Procedure “interpret response” processes the</td>
<td>Row and Task</td>
<td>Decision Rules</td>
</tr>
<tr>
<td>Row 4 - Response 3c

Procedural Abstraction

3.B, AAP-3.C

The written response:
● includes two program code segments:
○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure.
○ one showing where the student-developed procedure is being called.

● describes what the identified procedure does and how it contributes to the overall functionality of the program.</td>
<td>Consider ONLY written response 3c when scoring this point.

Requirements for program code segments:
● The procedure must be student developed, but could be developed collaboratively with a partner.
● If multiple procedures are included, use the first procedure to determine whether the point is earned.

Do NOT award a point if any one or more of the following is true:
● the code segment is an event handler; OR
● the code segment consisting of the procedure is not included in the written response section; OR
● the written response describes what the procedure does independently without relating it to the overall function of the program.</td>
</tr>
<tr>
<td colspan="2">The response earned the point for this row, meeting both criteria.
● The response includes a student-developed procedure, interpret_response, which has a parameter, number_of_hours, that affects the functionality of the procedure. The response provides a code segment showing a call to interpret_response from the Find_Response_button.Click event.
● The response describes what the procedure does: it “processes the ‘number of hours’ and formulates an output that it will pull from the list ‘solution list’ based on the question and the number of hours,” and it “uses a loop that checks what question is displayed to the user, so it can understand the parameter in the context of what question is being asked.”</td>
</tr>
<tr>
<td>Row 5 - Response 3c

Algorithm Implementation

2.B, AAP-2.H, AAP-2.K</td>
<td>Consider ONLY written response 3c when scoring this point.

Responses that do not earn row 4 may still earn this row.

Requirements for program code segments:</td>
</tr>
</table>

"number of hours" and formulates an output that it will pull from the list "solution list" based on the question and the number of hours. The procedure uses a loop that checks what question is displayed to the user, so it can understand the parameter in the context of what question is being asked. After analyzing the question that is displayed and the "number of hours", the procedure will pull different strings from a second list, "solutions list". If the user indicates that they are harming the environment through the "number of hours", the procedure will pull an output from the second list that notifies the user that they are doing harm to the environment, and give them ways to lessen their effects. Also, an image property is set to thumbs down. If "number of hours" indicates they are being good to the environment, the program will notify them and congratulate them on their safe living. Also, the image property is set to a thumbs up.	The written response: ● includes a student-developed algorithm that includes: ○ sequencing ○ selection ○ iteration ● explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.	● The algorithm being described can utilize existing language functionality or library calls. ● An algorithm that contains selection and iteration, also contains sequencing. ● An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point. ● Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned. ● If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included. **Do NOT award a point if any one or more of the following is true:** ● The response only describes what the selected algorithm does without explaining how it does it. ● The description of the algorithm does not match the included program code. ● The code segment consisting of the selected algorithm is not included in the written response. ● The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm). ● The use of either the selection or the iteration is trivial and does not affect the outcome of the program.
	The response DOES NOT earn the point for this row. The response met only one of the two criteria. ● The response includes a program code segment of a student-developed algorithm found in the body of the interpret_response procedure. This algorithm appears to include sequencing, selection (if, then), and iteration (for each and do); however, the iteration is trivial, as the value of item is never used and the outcome is the same whether this code iterates one time or many times. ● The response explains how the algorithm sequence works using "a loop that checks what question is displayed to the user, so it can understand the parameter in the context of what question is being asked. After analyzing the question that is displayed and the 'number of hours', the procedure will pull different strings from a second list, 'solutions list…. Also, an image property is set" based on the number of hours indicated so that the user receives a string and visual output based on processing of the data input.	

3d. Provide a written response that:

- 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; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">If the question label is equal to the second element in “question list” and the parameter “number of hours” (equal to the text box text) is 6, then the second element in “solutions list” will be displayed on “output label” and image property will be set to badd.png. In this scenario, the question asked to the user asks how many hours their AC was on during that day. The user inputs 6 to the text box, indicating that their AC was on for 6 hours. This is where I wanted a message to display to the user that they are using their AC too much and offer them alternatives. So, I created a list with solutions and that message is the second element in “solution list”, so that is why the program pulls the second element from “solution list” and displays it on “output label” when the user inputs 6 for the question at index 2 in “questions list.
Another scenario, if the first element in “question list” is equal to the string displayed in “question label” and the text input by the user, or “number of hours”, is 1, then the fourth element in “solution list” is pulled and displayed on the output label and the</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

Testing

4.C, CRD-2.J

The written response:
● describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.

● describes the condition(s) being tested by each call to the procedure.

● identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
● A procedure is not identified in written response 3c or the procedure does not have a parameter.
● The written response for 3d does not apply to the procedure in 3c.
● The two calls cause the same segment of code in the algorithm to execute even if the result is different.
● The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
● The identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td></td><td colspan="2">The response earned the point for this row, meeting all three criteria.
● The response describes two calls to the interpret_response procedure. The first call asks, “how many hours their AC was on,” where the “number of hours” parameter is “6.” The second calls asks, “the question ... how long the user had spent driving on that day,” where the “number of hours” parameter is “1.”
● The response describes the conditions as “[when to] display to the user that they are using their AC too much” or “when they aren’t emitting too many fossil fuels [sic].”
● The response states that the result of the first call will “display to the user that they are using their AC too much and offer them alternatives,” and that the result of the second call “will display element 4 from ‘solutions list’ that notifies the user that they are doing a good job and that they aren’t emitting too many</td></tr>
</table>

image is set to “good.png”. In the context of my program, this scenario would mean the question asks how long the user had spent driving on that day. The program recognizes this is the question being asked and since the parameter is less than 3, then the program will display element 4 from “solutions list” that notifies the user that they are doing a good job and that they aren’t emitting too many fossil fuels. The user will also be encouraged to keep it up and a thumbs up image will pop up.	fossil fuels [sic].”

Create PT 20-21 College Board Sample F - Score: 3/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**1**	**1**	**0**	**0**	**1**	**0**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

The video shows the user typing in their username, throwing the dart, getting their score, and then their high score being displayed. One input into the program is the user's name. Another input is the coordinates of the mouse click. The user's name is turned into an output when it is displayed in the high scores list. The input of the coordinates of the mouse is shown as output in the end as a score. The program entertains children by giving them multiple different games to play.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

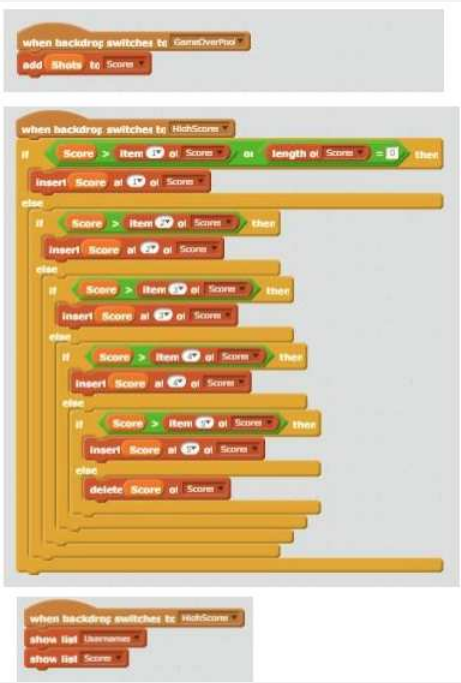
4.A, CRD-2B

● The video demonstrates the running of the program including:
○ input
○ program functionality
○ output
AND
● The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

Do NOT award a point if the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row, meeting all six criteria.
● The video demonstrates the running of the program, including game play input, such as entering the user's name, selecting a game, and clicking where darts are thrown; functionality of gameplay; and output of the score for each dart thrown, as well as a high score list. This satisfies the first three criteria for the video.
● The response describes the overall purpose of creating a program that "entertains children by giving them multiple different games to play."
● The response describes the functionality demonstrated as, "The video shows the user typing in their username, throwing the dart, getting their score, and then their high score being displayed."
● The response describes the input and output shown in the video as, "One input into the program is the user's name," and, "The user's name is turned into an output when it is displayed in the high scores list."</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

In our program, we made a list to show our high scores called “scores” and usernames called “usernames” at the end of the program. The program collects the username based on what the user types in as their name in the beginning and the scores are</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>Row 2 - Response 3b

Data Abstraction

3.B, AAP-1.C

The written response:
● includes two program segments:
○ one that shows how data has been stored in this list (or other collection type)
○ one that shows the data in this same list being used as part of fulfilling the program’s purpose.

● identifies the name of the variable representing the list being used in this response

● describes what the data contained in this list is representing in the program.</td><td>Consider ONLY written response 3b when scoring this point.

Requirements for program code segments:
● The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.
● If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

Do NOT award a point if the following is true:
● The use of the list is trivial and does not assist in fulfilling the program’s purpose.</td></tr>
<tr><td colspan="3">The response earned the point for this row, meeting all three of the criteria.
● The first two code segments presented are used to compute the score. The response includes program code segments showing how data have been stored in the list Scores, as well as how another list, HighScores, uses and sorts Scores to identify the highest-scoring player.
● The name of the list used in the response is Scores.
● The response explains that Scores is “a list to show our high scores called ‘scores,’” as well as “data in the scores list is an integer [sic].”</td></tr>
</table>

collected through the program collecting the user's results. The data in the username list is in the form of a string. The data in the scores list is an integer. The program code segment that I selected shows how the list "scores" is being used in the program. The list manages the complexity of the program by reducing the amount of lines needed. It reduces the amount of lines because it makes it so we don't have to display the scores over and over again each time the game is played. Instead, we can just add it to the list so we can put the scores in order and print them all at one time.	**Row 3 - Response 3b** **Managing Complexity** **3.C, AAP-3.C** The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**Consider ONLY written response 3b when scoring this point.** **Responses that do not earn row 2, may still earn this row.** **Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response DOES NOT earn the point for this row. The response does not meet either of the criteria. ● The code segment demonstrates the use of list Scores; however, the use of the list does not manage complexity in the program. If the length of the list grows beyond 5, significant modifications would need to be made for the functionality to be maintained. ● The response states that the list "reduces the amount of lines because it makes it so we don't have to display the scores over and over again each time the game is played." However, it does not explain how the list Scores manage complexity by explaining how the program would be written different without using the list. Additionally, the response does not explain precisely how the program would be more complex if a list was not used to maintain high scores. In fact, in the program code given, use of individual variables instead of a list would lead to essentially the same program.	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 
</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>Row 4 - Response 3c

Procedural Abstraction

3.B, AAP-3.C

The written response:
● includes two program code segments:
○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure.
○ one showing where the student-developed procedure is being called.

● describes what the identified procedure does and how it contributes to the overall functionality of the program.</td><td>Consider ONLY written response 3c when scoring this point.

Requirements for program code segments:
● The procedure must be student developed, but could be developed collaboratively with a partner.
● If multiple procedures are included, use the first procedure to determine whether the point is earned.

Do NOT award a point if any one or more of the following is true:
● the code segment is an event handler; OR
● the code segment consisting of the procedure is not included in the written response section; OR
● the written response describes what the procedure does independently without relating it to the overall function of the program.</td></tr>
<tr><td colspan="2">The response DOES NOT earn the point for this row. The response met only one of the two criteria.
● The response <u>does not include a student-developed procedure</u> that uses at least one parameter. The procedure backdrop switches is a built-in event. Additionally, there is no code segment included showing a call to backdrop switches.</td></tr>
</table>

<table>
<tr>
<td rowspan="2"> 

The algorithm shown above throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart bases on what color the tip of the dart is touching. It does this by first, showing you how many darts you have left. Then, it only lets you throw the dart if there are 3 darts left because it is the first dart. When the mouse is clicked, the dart goes to where the pointer is and then its y valueis CSP 2020 Sample F changed randomly. One is taken away from darts left and then the program runs a loop that assigns the dart different scores based on which color the tip of the dart is touching. The algorithm uses sequencing inside of the first if statement when it has to first, wait, then throw, then take away a dart. The algorithm uses selection by having a lot of if</td>
<td colspan="2">• The response does describe what the event does: it “throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart bases [sic] on what color the tip of the dart is touching.”</td>
</tr>
<tr>
<td>Row 5 - Response 3c

Algorithm Implementation

2.B, AAP-2.H, AAP-2.K

The written response:
• includes a student-developed algorithm that includes:
  ○ sequencing
  ○ selection
  ○ iteration

• explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.</td>
<td>Consider ONLY written response 3c when scoring this point.

Responses that do not earn row 4 may still earn this row.

Requirements for program code segments:
• The algorithm being described can utilize existing language functionality or library calls.
• An algorithm that contains selection and iteration, also contains sequencing.
• An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point.
• Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned.
• If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included.

Do NOT award a point if any one or more of the following is true:
• The response only describes what the selected algorithm does without explaining how it does it.
• The description of the algorithm does not match the included program code.
• The code segment consisting of the selected algorithm is not included in the written response.
• The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm).
• The use of either the selection or the iteration is trivial and does not affect the outcome of the program.</td>
</tr>
</table>

statements. These if statements tell the program what to do if the dart is touching certain colors. The algorithm uses iteration by using a forever loop to score the darts.	**The response earned the point for this row, meeting both criteria.** ● The response includes a program code segment of a student-developed algorithm that includes sequencing, selection (if...then), and iteration (forever loop). Because the forever loop is used purposefully to record where the dart is touching, it can be used to satisfy this requirement. ● The response explains in detail how the algorithm works, including that it "throws a dart at the dart board at a random strength/height drop. Then, the algorithm scores the dart bases [sic] on what color the tip of the dart is touching. It does this by first, showing you how many darts you have left. Then, it only lets you throw the dart if there are 3 darts left because it is the first dart. When the mouse is clicked, the dart goes to where the pointer is and then its y value is changed randomly. One is taken away from darts left and then the program runs a loop that assigns the dart different scores based on which color the tip of the dart is touching."

3d. Provide a written response that:

- 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; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
One test case I ran was clicking mouse to throw all three of the darts. By doing this, I was testing the ability of the code to throw and score the darts. I expected the dart to be thrown and for its y value to be changed randomly. I also expected the dart to be scored correctly base on what color the tip is touching. When I tested this, I found that it worked correctly. Another test case that I ran was that I threw all of the darts off of the board. By doing this I was testing how the program would score a dart if the tip of the dart is touching a color that has no score to go along with it. I was expecting the dart to have a score of zero, but when I tested it, the program would assign it the score of whatever color	**Row 6 - Response 3d** **Testing** **4.C, CRD-2.J** The written response: ● describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute. ● describes the condition(s) being tested by each call to the procedure. ● identifies the result of each call.	**Consider ONLY written response 3d when scoring this point.** **Responses that do not earn row 4 may still earn this row.** **Do NOT award a point if any one or more of the following is true:** ● A procedure is not identified in written response 3c or the procedure does not have a parameter. ● The written response for 3d does not apply to the procedure in 3c. ● The two calls cause the same segment of code in the algorithm to execute even if the result is different. ● The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program. ● The identified results of either call are

<table>
<tr><td rowspan="2">it is closest to. I fixed this by adding an if, else statement to the forever loop that assigns the background color a score.</td><td></td><td>implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="2">The response DOES NOT earn the point for this row. The response does not meet any of the criteria.
The response does not describe two calls to the backdrop switches procedure using different arguments. Instead, the response describes two paths in the program code based on user input during execution of the code rather than two calls to the procedure from another part of the program code that lead to different behaviors.</td></tr>
</table>

Create PT 20-21 College Board Sample G - Score: 2/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**0**	**1**	**1**	**0**	**0**	**0**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

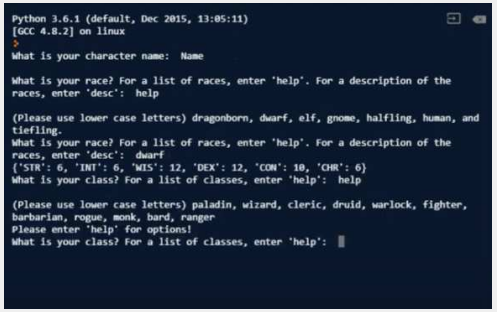
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

The program my partner and I wrote addresses the issue of making an RPG character. Though there are many RPG character creators out there, they can be complicated and difficult to use. Our program is very straightforward with lots of room for customizability in order to allow the easy creation of an RPG character for either a beginner or a seasoned veteran.</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A, CRD-2B

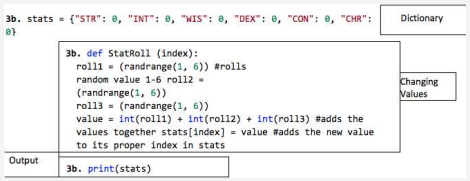
• The video demonstrates the running of the program including:
  ○ input
  ○ program functionality
  ○ output
AND
• The written response:
  ○ describes the overall purpose of the program.
  ○ describes what functionality of the program is demonstrated in the video
  ○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

Do NOT award a point if the following is true:
• the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response DOES NOT the point for this row. The response met only four of the six criteria.
• The video demonstrates the running of the program to collect input of answers based on a series of output in the form of displayed role-playing game questions and prompts for character creation. This satisfies the first three criteria for the video.
• The response states that the program's purpose is to address "the issue of making an RPG character" in a way that is "very straightforward with lots of room for customizability in order to allow the easy creation of an RPG character for either a beginner or a seasoned veteran."
• The response does not describe the functionality demonstrated in the video.
• The response does not describe the input and output of the program demonstrated in the video.</td></tr>
</table>

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

second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="4"> 

The data in the list “stats” represents the stats of the player’s character, those being Strength, Intelligence, Wisdom, Dexterity, Constitution, and Charisma. This list manages complexity by allowing all of the stats of the character to be in one area, so they can be easily changed when bonuses arise, certain armor or magic alters them, etc. Without this list, each stat would be its own separate variable. Furthermore, this dictionary will make it much easier to export everything to a document, as all the player stats are in one central location, and it’ll take much less formatting.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 2 - Response 3b

Data Abstraction

3.B, AAP-1.C

The written response:
● includes two program segments:
○ one that shows how data has been stored in this list (or other collection type)
○ one that shows the data in this same list being used as part of fulfilling the program’s purpose.

● identifies the name of the variable representing the list being used in this response

● describes what the data contained in this list is representing in the program.</td><td>Consider ONLY written response 3b when scoring this point.

Requirements for program code segments:
● The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.
● If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

Do NOT award a point if the following is true:
● The use of the list is trivial and does not assist in fulfilling the program’s purpose.</td></tr>
<tr><td colspan="2">The response earned the point for this row, meeting all three criteria.
● The response provides program code segments showing the list stats being initialized and being used in the StatRoll procedure.
● The response provides that the list is named stats.
● The response describes the data in the list as “the stats of the player’s character, those being Strength, Intelligence, Wisdom, Dexterity, Constitution, and Charisma.”</td></tr>
<tr><td>Row 3 - Response 3b</td><td>Consider ONLY written response 3b when scoring this</td></tr>
</table>

	Managing Complexity **3.C, AAP-3.C** The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**point.** **Responses that do not earn row 2, may still earn this row.** **Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response earned the point for this row, meeting both criteria. ● The response provides a list, stats, which is being used to manage complexity in the program. ● The response explains how stats manages complexity in the program "by allowing all of the stats of the character to be in one area, so they can be easily changed when bonuses arise, certain armor or magic alters them, etc. Without this list, each stat would be its own separate variable. Furthermore, this dictionary will make it much easier to export everything to a document, as all the player stats are in one central location, and it'll take much less formatting."	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="4">

```
looper1 = 1
race = race.lower()
while looper1 == 1:
      if race == "dragonborn": PostRollValue("STR", 2)
        PostRollValue("CHR", 1) print()
        looper1 = 2
      elif race == "dwarf": PostRollValue("CON", 2)
        looper1 = 2
      elif race == "elf": PostRollValue("DEX", 2)
        looper1 = 2
      elif race == "gnome": PostRollValue("INT", 2)
        looper1 = 2
      elif race == "halfling": PostRollValue("DEX", 2)
        looper1 = 2
      elif race == "human": PostRollValue("STR", 1)
        PostRollValue("INT", 1)
        PostRollValue("WIS", 1)
        PostRollValue("DEX", 1)
        PostRollValue("CON", 1)
        PostRollValue("CHR", 1)
        looper1 = 2
      elif race == "tiefling": PostRollValue("INT", 1)
        PostRollValue("CHR", 2)
        looper1 = 2
```

An algorithm that we have developed that includes sequencing, selection, and iteration is the race selection algorithm. This algorithm loops until a proper race is entered, and using if / elif statements, it selects which race was entered. It then uses sequencing to append values to the stats list based on that race. This allows the character to choose the race they want to play, and get its bonuses.

</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 4 - Response 3c

Procedural Abstraction

3.B, AAP-3.C

The written response:
includes two program code segments:
one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure.
one showing where the student-developed procedure is being called.
describes what the identified procedure does and how it contributes to the overall functionality of the program.</td><td>Consider ONLY written response 3c when scoring this point.

Requirements for program code segments:
The procedure must be student developed, but could be developed collaboratively with a partner.
If multiple procedures are included, use the first procedure to determine whether the point is earned.
Do NOT award a point if any one or more of the following is true:
the code segment is an event handler; OR
the code segment consisting of the procedure is not included in the written response section; OR
the written response describes what the procedure does independently without relating it to the overall function of the program.</td></tr>
<tr><td colspan="2">The response DOES NOT earn the point for this row. The response does not meet any of the criteria.
The program code segment provided in response 3c is not a student-developed procedure. While there is a student-developed procedure with parameter in response 3b, program code or answers provided in one prompt cannot be considered when scoring another prompt.</td></tr>
<tr><td>Row 5 - Response 3c</td><td>Consider ONLY written response 3c when scoring this point.</td></tr>
</table>

	Algorithm Implementation **2.B, AAP-2.H, AAP-2.K** The written response: ● includes a student-developed algorithm that includes: ○ sequencing ○ selection ○ iteration ● explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.	**Responses that do not earn row 4 may still earn this row.** **Requirements for program code segments:** ● The algorithm being described can utilize existing language functionality or library calls. ● An algorithm that contains selection and iteration, also contains sequencing. ● An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point. ● Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned. ● If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included. **Do NOT award a point if any one or more of the following is true:** ● The response only describes what the selected algorithm does without explaining how it does it. ● The description of the algorithm does not match the included program code. ● The code segment consisting of the selected algorithm is not included in the written response. ● The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm). ● The use of either the selection or the iteration is trivial and does not affect the outcome of the program.
	The response DOES NOT the point for this row. The response does not meet either of the criteria. ● The response includes a program code segment of a student-developed algorithm that appears to include sequencing, selection (if…elif statement), and iteration (while loop). Even though this algorithm is not included in a procedure with a parameter, it would still be considered. <u>However, because the loop control variable looper1 is initialized to 1 and will be changed to 2 after the first iteration causing the loop to stop, this loop is trivial and the code segment will function the same regardless</u>.	

	● The response generally explains the functionality as the “algorithm loops until a proper race is entered, and using if/elif statements, it selects which race was entered. It then uses sequencing to append values to the stats list based on that race.” However, the response does not explain how the algorithm works in enough detail to enable someone to recreate it.

3d. Provide a written response that:

- 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; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">To test this program, many different test cases were used. We tried entering numbers, as well as differently capitalized words, and in all cases we found we had made the program foolproof. One algorithm we tested was the race selection, in which we misspelled words and entered numbers. However, with our loop and .lower, the program worked fine. We also tried entering things other than lawful, neutral, and chaotic in the alignment selection, but it worked perfectly as well.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

Testing

4.C, CRD-2.J

The written response:
● describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.

● describes the condition(s) being tested by each call to the procedure.

● identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
● A procedure is not identified in written response 3c or the procedure does not have a parameter.
● The written response for 3d does not apply to the procedure in 3c.
● The two calls cause the same segment of code in the algorithm to execute even if the result is different.
● The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
● The identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="2">The response DOES NOT the point for this row.
● Since there is not a procedure with parameters identified in written response 3c, this response does not provide two specific test cases to a procedure with different paths and outcomes. The response does not meet any of the three criteria.</td></tr>
</table>

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.

3a. Provide a written response that:
(Approx. 150 words, for all subparts of 3a combined)

i. Describes the overall purpose of the program;

Purpose of the program is to help the user decide whether or not to do something

ii. Describes what functionality the video illustrates;

When the user clicks on the screen, the magic 8 ball appears to make a recommendation and the icons on the screen change to represent if it's a positive, neutral, or negative response.

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

Input is the user clicking on the screen. Output is the text displayed on the magic 8 ball and the different icons.

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

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

```
var answers = ["Yes, absolutely", "I have a good feeling about this", "Why not!",
    "Maybe", "I'm not sure", "Ask me again",
    "Don't even think about it!", "Are you kidding?", "You are crazy!"];
```

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.

```
// selects a random index
// sets the text of the magic 8 ball to the answer stored at the random index
// play a sound
// calls the function to set the images
function updateScreen(){
  index = randomNumber(0, answers.length-1);
  setText("answerOutput", answers[index]);
  playSound("sound://category_pop/bubble_pop_cluster_2.mp3");
  setImages(index);
}
```

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

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

Name of list = answers

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

List of strings which store responses randomly chosen to display on the screen.

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

Manages complexity because my code would be longer without a list.

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.

```
23  // calls a function to set the images based on the index of the answer
24  // index {number} - the random index selected when the screen is clicked
25  function setImages(index){
26    if(index < 3){
27      styleImages("icon://fa-star", "yellow");
28    } else if (index < 6) {
29      styleImages("icon://fa-question-circle", "orange");
30    } else {
31      styleImages("icon://fa-ban", "red");
32    }
33  }
34
35  // styles all ten images on the screen
36  // icon {string} - icon image
37  // color {string} - icon color
38  function styleImages(icon, color){
39    for(var i=0; i<10; i++){
40      setProperty("outputImage" + i, "icon-color", color);
41      setProperty("outputImage" + i, "image", icon);
42    }
43  }
```

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

```
8  onEvent("playScreen", "click", function(){
9    updateScreen();
10 });
11
12 // selects a random index
13 // sets the text of the magic 8 ball to the answer stored at the random index
14 // play a sound
15 // calls the function to set the images
16 function updateScreen(){
17   index = randomNumber(0, answers.length-1);
18   setText("answerOutput", answers[index]);
19   playSound("sound://category_pop/bubble_pop_cluster_2.mp3");
20   setImages(index);
21 }
```

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 function setImages controls what icons are displayed on the screen after a random response is chosen. This helps the user know if the response was positive or not.

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.

The function setImages works by choosing an image and calling another function to set the color of the image.

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:

Call # 1: setImages(2)

Second call:

Call #2: setImages(7)

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

Condition(s) tested by the first call:

The list is organized so the first three elements are positive responses, the next three are neutral, and the last three are negative. This call is passing through the argument 2. Lines 26-27 check if 2 is less than three.

Condition(s) tested by the second call:

This call is passing through the argument 7. Lines 26-29 are skipped, because 7 not less than 3 and 7 is not less than 6. Lines 30-32 run because it is the final branch of the if else statement.

iii. Identifies the result of each call.

Result of the first call:

2 is less than 3 is true, so the response is positive and the styleImages function is run to set the icon to a yellow star.

Result of the second call:

The response is negative and therefore the styleImages function is run to set the icon to a red crossed out icon.

3a. Provide a written response that:
(Approx. 150 words, for all subparts of 3a combined)

i. Describes the overall purpose of the program;

The Random Dog Picker app displays a random image and name of a dog based on what size is selected.

ii. Describes what functionality the video illustrates;

The video shows several different sizes being chosen and how a different dog shows up each time.

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

The input is the selection in the dropdown, and the output is the image and name of the dog 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.
(Approx. 200 words, for all subparts of 3b combined, excluding program code)

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

```
3 | var dogHeight = getColumn("dogs", "Max Height");
```

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.

```
21  function filter(){
22    // clears the filtered lists
23    filteredDogNames = [];
24    filteredDogImages = [];
25
26    // gets the size from the dropdown
27    var dogSize = getText("sizeDropdown");
28
29    // traverses the dogHeight List
30    // if dogHeight and dogSize meet certain conditions
31    // the corresponding names and images are stored in the filtered lists
32    for(var i=0; i<dogHeight.length; i++){
33      if(dogHeight[i] < 16 && dogSize == "Small"){
34        appendItem(filteredDogNames, dogNames[i]);
35        appendItem(filteredDogImages, dogImages[i]);
36      } else if(dogHeight[i] >= 16 && dogHeight[i] < 24 && dogSize == "Medium")
37        appendItem(filteredDogNames, dogNames[i]);
38        appendItem(filteredDogImages, dogImages[i]);
39      } else if(dogHeight[i] >= 24 && dogSize == "Large") {
40        appendItem(filteredDogNames, dogNames[i]);
41        appendItem(filteredDogImages, dogImages[i]);
42      }
43    }
44
45    // prints the list of dog names that match the value in the dropdown
46    console.log(dogSize + " Dogs:\n" + filteredDogNames);
47  }
```

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

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

The list is filtered based on the size selected.

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

If the size chosen is "Small" only dogs whose height is less than 16 will be randomly chosen to be displayed. If the size is "Medium" then the displayed dog needs to be between 16 and 23. If the size is "Large" then dogs that have a height bigger than 24 may be displayed.

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 list manages complexity because without it, dog heights would all have to be stored in their own individual variables. This would be very confusing, and would add a lot of extra lines.

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.

```
function filter(){
  // clears the filtered lists
  filteredDogNames = [];
  filteredDogImages = [];

  // gets the size from the dropdown
  var dogSize = getText("sizeDropdown");

  // traverses the dogHeight List
  // if dogHeight and dogSize meet certain conditions
  // the corresponding names and images are stored in the filtered lists
  for(var i=0; i<dogHeight.length; i++){
    if(dogHeight[i] < 16 && dogSize == "Small"){
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    } else if(dogHeight[i] >= 16 && dogHeight[i] < 24 && dogSize == "Medium")
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    } else if(dogHeight[i] >= 24 && dogSize == "Large") {
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    }
  }

  // prints the list of dog names that match the value in the dropdown
  console.log(dogSize + " Dogs:\n" + filteredDogNames);
}
```

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

```
55  // sets up the lists and the screen
56  function listSetup(){
57    filter();
58    updateScreen();
59  }
```

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

This function filters the lists to smaller lists from which a random name and image can be picked and then displayed.

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.

To do this, a for loop in lines 31-43 traverses the dogHeight list and an if else if statement checks to see if each element fits into what the user wants. If it does, the element at that index in the dogName and in the dogImage lists is added to the filtered lists.

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:

If the user selects a small dog.

Second call:

If the user selects a medium dog.

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

Condition(s) tested by the first call:

The function filters the lists to only include small dogs.

Condition(s) tested by the second call:

The function filters the list to only include medium dogs.

iii. Identifies the result of each call.

Result of the first call:

One of those is chosen randomly and displayed on the screen.

Result of the second call:

One of those is chosen randomly and displayed on the screen.

Create PT 20-21 Code.org Sample 1 - Score: 6/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 1	**1**	**1**	**1**	**1**	**1**	**1**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

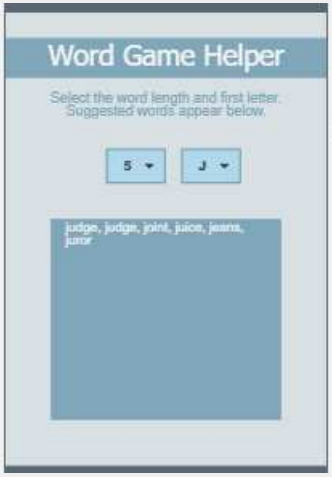
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

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. The video shows how the user selects the length of the word and the first letter from different dropdowns. 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.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A

CRD-2B

● The video demonstrates the running of the program including:
○ input
○ program functionality
○ output

AND
● The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

Do NOT award a point if the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The response describes the purpose of the app is "to assist users in finding words to help in various games…" Input and output are shown in the video and described in the response as dropdowns and "list of words...displayed on the screen."</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and

- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="4">

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

```
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 }
```

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. 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.

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

</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>

Row 2 - Response 3b

Data Abstraction

3.B

AAP-1.C

The written response:
- includes two program segments:
  - one that shows how data has been stored in this list (or other collection type)
  - one that shows the data in this same list being used as part of fulfilling the program's purpose.
- identifies the name of the variable representing the list being used in this response
- describes what the data contained in this list is representing in the program.

</td><td>

Consider ONLY written response 3b when scoring this point.

Requirements for program code segments:
- The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.
- If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

Do NOT award a point if the following is true:
- The use of the list is trivial and does not assist in fulfilling the program's purpose.

</td></tr>
<tr><td colspan="2">

The response earned the point for this row.
The program code shows how the list is created and processed in the two code segments. The name of the list is identified as wordList. The list stores words that are displayed to the user.

</td></tr>
<tr><td>

Row 3 - Response 3b

Managing Complexity

3.C

</td><td>

Consider ONLY written response 3b when scoring this point.

Responses that do not earn row 2, may still earn this row.

</td></tr>
</table>

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.	**AAP-3.C** The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response earned the point for this row. The response explains that the code would be written differently without the list by storing each word individually in its own variable, which would require extra program code. It also explains that the code allows the program to work for any length word list without any changes.	

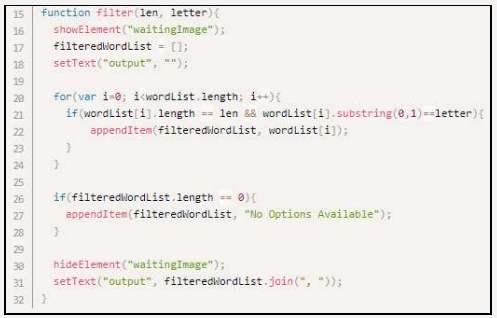
3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	Scoring Guidelines



```
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  }
```

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

Row and Task	Decision Rules
Row 4 - Response 3c **Procedural Abstraction** **3.B, AAP-3.C** The written response: ● includes two program code segments: ○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. ○ one showing where the student-developed procedure is being called. ● describes what the identified procedure does and how it contributes to the overall functionality of the program.	**Consider ONLY written response 3c when scoring this point.** **Requirements for program code segments:** ● The procedure must be student developed, but could be developed collaboratively with a partner. ● If multiple procedures are included, use the first procedure to determine whether the point is earned. **Do NOT award a point if any one or more of the following is true:** ● the code segment is an event handler; OR ● the code segment consisting of the procedure is not included in the written response section; OR ● the written response describes what the procedure does independently without relating it to the overall function of the program.
The response earned the point for this row. The code segment is a procedure (function) with parameters (len, letter). The written response explains what the procedure's purpose in the overall program is: "to output a filtered word list to users which meets their chosen requirements."	
Row 5 - Response 3c **Algorithm Implementation** **2.B, AAP-2.H, AAP-2.K** The written response: ● includes a student-developed algorithm that includes: ○ sequencing	**Consider ONLY written response 3c when scoring this point.** **Responses that do not earn row 4 may still earn this row.** **Requirements for program code segments:** ● The algorithm being described can utilize existing language functionality or library calls. ● An algorithm that contains selection and iteration, also contains sequencing.

	○ selection ○ iteration ● explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.	● An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point. ● Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned. ● If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included. **Do NOT award a point if any one or more of the following is true:** ● The response only describes what the selected algorithm does without explaining how it does it. ● The description of the algorithm does not match the included program code. ● The code segment consisting of the selected algorithm is not included in the written response. ● The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm). ● The use of either the selection or the iteration is trivial and does not affect the outcome of the program.
	The response earned the point for this row. The code segments displayed an algorithm that included: ● sequencing (more than one line inside the procedure) ● selection (an if-statement) ● iteration (a for-loop) The written response explains in detail, line by line, how the algorithm works, which filters a list according to length and the first letter.	

3d. Provide a written response that:

- 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; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">Let's suppose the user selects from the dropdowns a length of 3 and the letter "d". 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. 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.

Another example would be if the function was called with the arguments 1 and "b". 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. 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.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

Testing

4.C, CRD-2.J

The written response:
describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.
describes the condition(s) being tested by each call to the procedure.
identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
A procedure is not identified in written response 3c or the procedure does not have a parameter.
The written response for 3d does not apply to the procedure in 3c.
The two calls cause the same segment of code in the algorithm to execute even if the result is different.
The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
The identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="3">The response earned the point for this row.
The written response clearly explains two different calls to the procedure. Two examples are given with different parameters, which results in different segments of code running. The results of each call are explained "any word that had a length of 3 and started with the letter "d" have been added to the filtered list" and "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."

Code.org commentary: While the student earned the point, we believe a stronger case for the requirement that "each call must pass a different argument causing a different segment of code to execute" would be made if the code included an if-else or if-else-if statement.</td></tr>
</table>

Create PT 20-21 Code.org Sample 2 - Score: 4/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 2	**1**	**1**	**0**	**1**	**0**	**1**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

Purpose of the program is to help the user decide whether or not to do something
When the user clicks on the screen, the magic 8 ball appears to make a recommendation and the icons on the screen change to represent if it's a positive, neutral, or negative response.
Input is the user clicking on the screen. Output is the text displayed on the magic 8 ball and the different icons.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A, CRD-2B

• The video demonstrates the running of the program including:
○ input
○ program functionality
○ output
AND
• The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

Do NOT award a point if the following is true:
• the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The student explains that the purpose of the program is "to help the user decide whether or not to do something." The student goes on to explain the functionality the video illustrates: "user clicks on the screen… magic 8 ball appears to make a recommendation… and icons change." The input "user clicking on the screen" and output "text displayed on the magic 8 ball… and the different icons" are also defined in a separate bullet.</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and

- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="4">

```
var answers = ["Yes, absolutely", "I have a good feeling about this", "Why not!",
    "Maybe", "I'm not sure", "Ask me again",
    "Don't even think about it!", "Are you kidding?", "You are crazy!"];
```

```
// selects a random index
// sets the text of the magic 8 ball to the answer stored at the random index
// play a sound
// calls the function to set the images
function updateScreen(){
  index = randomNumber(0, answers.length-1);
  setText("answerOutput", answers[index]);
  playSound("sound://category_pop/bubble_pop_cluster_2.mp3");
  setImages(index);
}
```

- *Name of list = answers*
- *List of strings which store responses randomly chosen to display on the screen.*
- *Manages complexity because my code would be longer without a list.*

</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>

Row 2 - Response 3b

Data Abstraction

3.B, AAP-1.C

The written response:
- includes two program segments:
  - one that shows how data has been stored in this list (or other collection type)
  - one that shows the data in this same list being used as part of fulfilling the program's purpose.
- identifies the name of the variable representing the list being used in this response
- describes what the data contained in this list is representing in the program.

</td><td>

Consider ONLY written response 3b when scoring this point.

Requirements for program code segments:
- The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.
- If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

Do NOT award a point if the following is true:
- The use of the list is trivial and does not assist in fulfilling the program's purpose.

</td></tr>
<tr><td colspan="2">

The response earned the point for this row.
- The program code shows how the list is created and processed in the two code segments.
- The name of the list is identified as answers.
- The list stores "responses randomly chosen to display on the screen."

</td></tr>
<tr><td>

Row 3 - Response 3b

Managing Complexity

3.C, AAP-3.C

</td><td>

Consider ONLY written response 3b when scoring this point.

Responses that do not earn row 2, may still earn this row.

</td></tr>
</table>

	The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response DOES NOT earn the point for this row. The response explains that the code would be longer without a list. This is not enough information to earn the point. The students should have explained the specifics of how their code would be different and why the code would be longer.	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	**Scoring Guidelines**

```
// calls a function to set the images based on the index of the answer
// index {number} - the random index selected when the screen is clicked
function setImages(index){
  if(index < 3){
    styleImages("icon://fa-star", "yellow");
  } else if (index < 6) {
    styleImages("icon://fa-question-circle", "orange");
  } else {
    styleImages("icon://fa-ban", "red");
  }
}

// styles all ten images on the screen
// icon {string} - icon image
// color {string} - icon color
function styleImages(icon, color){
  for(var i=0; i<10; i++){
    setProperty("outputImage" + i, "icon-color", color);
    setProperty("outputImage" + i, "image", icon);
  }
}
```

- *The function setImages controls what icons are displayed on the screen after a random response is chosen. This helps the user know if the response was positive or not.*
- *The function setImages works by choosing an image and calling another function to set the color of the image.*

Row and Task	Decision Rules
Row 4 - Response 3c **Procedural Abstraction** **3.B, AAP-3.C** The written response: ● includes two program code segments: ○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. ○ one showing where the student-developed procedure is being called. ● describes what the identified procedure does and how it contributes to the overall functionality of the program.	**Consider ONLY written response 3c when scoring this point.** **Requirements for program code segments:** ● The procedure must be student developed, but could be developed collaboratively with a partner. ● If multiple procedures are included, use the first procedure to determine whether the point is earned. **Do NOT award a point if any one or more of the following is true:** ● the code segment is an event handler; OR ● the code segment consisting of the procedure is not included in the written response section; OR ● the written response describes what the procedure does independently without relating it to the overall function of the program.
The response earned the point for this row. ● The code segments is a procedure (function) with a parameter (index). ● The written response explains what the procedure's purpose in the overall program is: "controls what icons are displayed on the screen after a random response is chosen."	
Row 5 - Response 3c **Algorithm Implementation** **2.B, AAP-2.H, AAP-2.K** The written response: ● includes a student-developed algorithm that includes: ○ sequencing	**Consider ONLY written response 3c when scoring this point.** **Responses that do not earn row 4 may still earn this row.** **Requirements for program code segments:** ● The algorithm being described can utilize existing language functionality or library calls. ● An algorithm that contains selection and iteration, also contains sequencing.

<table>
<tr>
<td rowspan="2"></td>
<td>

selection
iteration

explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.

</td>
<td>

An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point.
Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned.
If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included.

Do NOT award a point if any one or more of the following is true:

The response only describes what the selected algorithm does without explaining how it does it.
The description of the algorithm does not match the included program code.
The code segment consisting of the selected algorithm is not included in the written response.
The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm).
The use of either the selection or the iteration is trivial and does not affect the outcome of the program.

</td>
</tr>
<tr>
<td colspan="2">
The response DOES NOT earn the point for this row.
The code segments displayed an algorithm that included:

sequencing (more than one line inside the procedure)
selection (an if-statement)
iteration (a for-loop)

The written response did not explain in detail how the algorithm works. Instead of referring to specific parts of the procedure, the student wrote in general about what the algorithm does instead of HOW it does it.
</td>
</tr>
</table>

3d. Provide a written response that:

- 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; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">

Call # 1: setImages(2)

The list is organized so the first three elements are positive responses, the next three are neutral, and the last three are negative. This call is passing through the argument 2. Lines 26-27 check if 2 is less than three.
2 is less than 3 is true, so the response is positive and the styleImages function is run to set the icon to a yellow star.

Call #2: setImages(7)

This call is passing through the argument 7. Lines 26-29 are skipped, because 7 not less than 3 and 7 is not less than 6. Lines 30-32 run because it is the final branch of the if else statement.
The response is negative and therefore the styleImages function is run to set the icon to a red crossed out icon.

</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

Testing

4.C, CRD-2.J

The written response:
describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.
describes the condition(s) being tested by each call to the procedure.
identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
A procedure is not identified in written response 3c or the procedure does not have a parameter.
The written response for 3d does not apply to the procedure in 3c.
The two calls cause the same segment of code in the algorithm to execute even if the result is different.
The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
The identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The written response clearly explains two different calls to the procedure.
Two examples are given with different parameters, which results in different segments of code running.
The results of each call are explained and different lines of code execute depending on the arguments. In the first call, lines 26-27 are executed and "the styleImages function is run to set the icon to a yellow star" whereas with the second call, lines 26-29 are skipped and lines 30-32 run and "the styleImages function is run to set the icon to a red crossed out icon."</td></tr>
</table>

Create PT 20-21 Code.org Sample 3 - Score: 2/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**1**	**0**	**0**	**0**	**1**	**0**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, visual, or textual) based on input from:
     - the user (including user actions that trigger events); or
     - a device; or
     - a file
   - use of at least one list (or other collection type) to represent a collection of data related to the program's purpose; and
   - development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

   Include comments or acknowledgements for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

   Create a PDF file that contains all your program code (including comments).

2. **Video**
   Your video must demonstrate your program running, including:
   - input to your program; and
   - at least one aspect of the functionality of your program; and
   - output produced by your program.

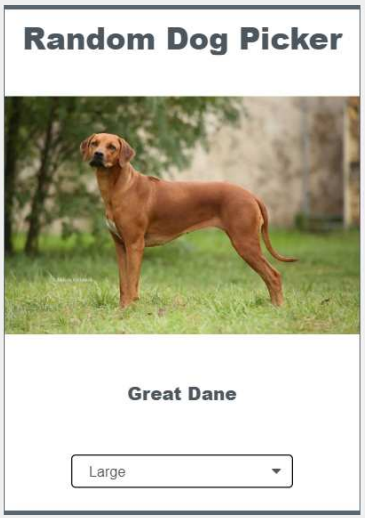
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

The Random Dog Picker app displays a random image and name of a dog based on what size is selected. The video shows several different sizes being chosen and how a different dog shows up each time. The input is the selection in the dropdown, and the output is the image and name of the dog displayed on the screen.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A

CRD-2B

● The video demonstrates the running of the program including:
○ input
○ program functionality
○ output

AND
● The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

Do NOT award a point if the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The student explains that the purpose of the program is to "display[s] a random image and name of a dog based on what size is selected" The student goes on to explain the functionality the video displays: "several different sizes being chosen and… a different dog shows up each time." The input "is the selection in the dropdown" and the output is explained as "the image and name of the dog displayed on the screen."</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and

- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="4">

```
var dogHeight = getColumn("dogs", "Max Height");
```

```
function filter(){
  // clears the filtered lists
  filteredDogNames = [];
  filteredDogImages = [];

  // gets the size from the dropdown
  var dogSize = getText("sizeDropdown");

  // traverses the dogHeight List
  // if dogHeight and dogSize meet certain conditions
  // the corresponding names and images are stored in the filtered lists
  for(var i=0; i<dogHeight.length; i++){
    if(dogHeight[i] < 16 && dogSize == "Small"){
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    } else if(dogHeight[i] >= 16 && dogHeight[i] < 24 && dogSize == "Medium"){
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    } else if(dogHeight[i] >= 24 && dogSize == "Large") {
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    }
  }

  // prints the list of dog names that match the value in the dropdown
  console.log(dogSize + " Dogs:\n" + filteredDogNames);
}
```

The list is filtered based on the size selected. If the size chosen is "Small" only dogs whose height is less than 16 will be randomly chosen to be displayed. If the size is "Medium" then the displayed dog needs to be between 16 and 23. If the size is "Large" then dogs that have a height bigger than 24 may be displayed. The list manages complexity because without it, dog heights would all have to be stored in their own individual variables. This would be very confusing, and would add a lot of extra lines.

</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>

Row 2 - Response 3b

Data Abstraction

3.B

AAP-1.C

The written response:

includes two program segments:

one that shows how data has been stored in this list (or other collection type)
one that shows the data in this same list being used as part of fulfilling the program's purpose.

identifies the name of the variable representing the list being used in this response
describes what the data contained in this list is representing in the program.

</td><td>

Consider ONLY written response 3b when scoring this point.

Requirements for program code segments:

The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.
If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

Do NOT award a point if the following is true:

The use of the list is trivial and does not assist in fulfilling the program's purpose.

</td></tr>
<tr><td colspan="2">The response DOES NOT the point for this row.
The program code shows how the list is created and processed in the two code segments. However, the name of the list is not identified.</td></tr>
<tr><td>

Row 3 - Response 3b

Managing Complexity

3.C

</td><td>

Consider ONLY written response 3b when scoring this point.

Responses that do not earn row 2, may still earn this row.

</td></tr>
</table>

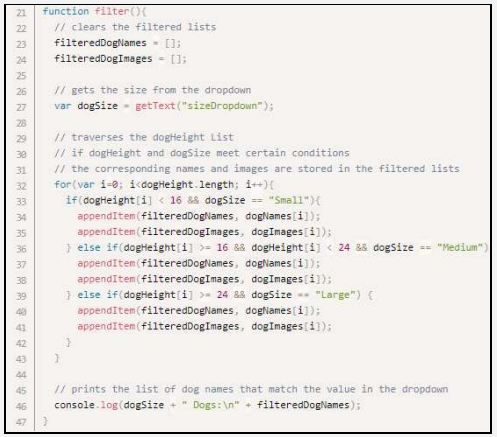
	AAP-3.C The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	The response DOES NOT earn the point for this row. According to the decision rules, the student does not earn the point for this row because "the written response does not name the selected list." The student explains how the list manages complexity, but because the list is not specifically named, the answer is invalid.	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
 *This function filters the lists to smaller lists from which a random name and image can be picked and then displayed. To do this, a for loop in lines 31-43 traverses the dogHeight list and an if else if statement checks to see if each element fits into what the user wants. If it does, the element at that index in the dogName and in the dogImage lists is added to the filtered lists.*	**Row 4 - Response 3c** **Procedural Abstraction** **3.B** **AAP-3.C** The written response: ● includes two program code segments: ○ one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. ○ one showing where the student-developed procedure is being called. ● describes what the identified procedure does and how it contributes to the overall functionality of the program.	**Consider ONLY written response 3c when scoring this point.** **Requirements for program code segments:** ● The procedure must be student developed, but could be developed collaboratively with a partner. ● If multiple procedures are included, use the first procedure to determine whether the point is earned. **Do NOT award a point if any one or more of the following is true:** ● the code segment is an event handler; OR ● the code segment consisting of the procedure is not included in the written response section; OR ● the written response describes what the procedure does independently without relating it to the overall function of the program.
	The response DOES NOT earn the point for this row. The code segments is a procedure (function) but the procedure does not use any parameters. The response explains what the procedure does, but does not receive the point because there is no parameter.	
	Row 5 - Response 3c **Algorithm Implementation** **2.B, AAP-2.H, AAP-2.K** The written response:	**Consider ONLY written response 3c when scoring this point.** **Responses that do not earn row 4 may still earn this row.** **Requirements for program code segments:** ● The algorithm being described can utilize existing

<table>
<tr>
<td></td>
<td>

includes a student-developed algorithm that includes:

sequencing
selection
iteration

explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.

</td>
<td>
language functionality or library calls.

An algorithm that contains selection and iteration, also contains sequencing.
An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point.
Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned.
If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included.

Do NOT award a point if any one or more of the following is true:

The response only describes what the selected algorithm does without explaining how it does it.
The description of the algorithm does not match the included program code.
The code segment consisting of the selected algorithm is not included in the written response.
The algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm).
The use of either the selection or the iteration is trivial and does not affect the outcome of the program.

</td>
</tr>
<tr>
<td></td>
<td colspan="2">
The response earned the point for this row.
The code segments displayed an algorithm that included:

sequencing (more than one line inside the procedure)
selection (an if-statement)
iteration (a for-loop)

The written response explains how the procedure accomplishes its task: "a for loop in lines 31-43 traverses the dogHeight list and an if else if statement checks to see if each element fits into what the user wants."
</td>
</tr>
</table>

3d. Provide a written response that:

- 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; and

- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">If the user selects a small dog, the function filters the lists to only include small dogs and one of those is chosen randomly and displayed on the screen.
If the user selects a medium dog, the function filters the list to only include medium dogs and of of those is chosen randomly and displayed on the screen.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

Testing

4.C, CRD-2.J

The written response:
describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.
describes the condition(s) being tested by each call to the procedure.
identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
A procedure is not identified in written response 3c or the procedure does not have a parameter.
The written response for 3d does not apply to the procedure in 3c.
The two calls cause the same segment of code in the algorithm to execute even if the result is different.
The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
The identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td></td><td colspan="2">The response DID NOT earn the point for this row.
The procedure given in 3c does not have a parameter. Therefore the user is unable to explain two different calls to the function which cause different parts of the code to execute.</td></tr>
</table>