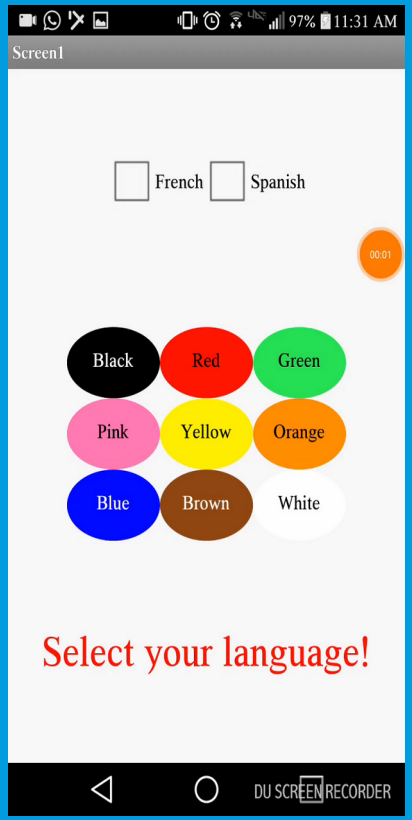
# Sample A

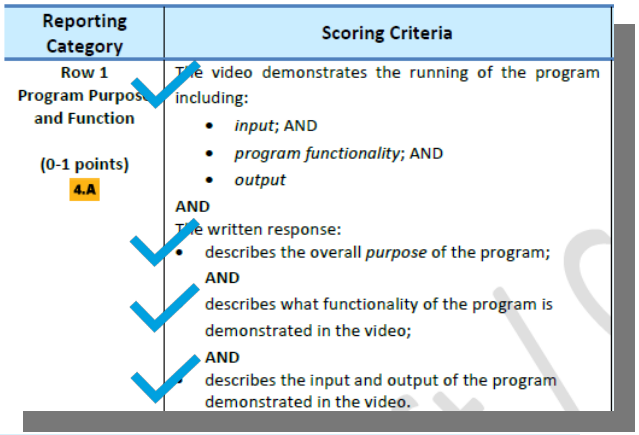
- Walk through the scoring of a sample with the participants. We are walking through Sample A.
- Watch the video.

| Reporting Category | Scoring Criteria |
|---|---|
| Row 1<br>Program Purpose and Function<br><br>(0-1 points)<br>4.A | The video demonstrates the running of the program including:<br>• *input*; AND<br>• *program functionality*; AND<br>• *output*<br>AND<br>The written response:<br>• describes the overall *purpose* of the program;<br>AND<br>• describes what functionality of the program is demonstrated in the video;<br>AND<br>• describes the input and output of the program demonstrated in the video. |



# Sample A

- Read through the response for 3a



| Reporting Category | Scoring Criteria |
|---|---|
| Row 1<br>Program Purpose and Function<br><br>(0-1 points)<br>4.A | The video demonstrates the running of the program including:<br>• *input*; AND<br>• *program functionality*; AND<br>• *output*<br>**AND**<br>The written response:<br>• describes the overall *purpose* of the program;<br>**AND**<br>describes what functionality of the program is demonstrated in the video;<br>**AND**<br>describes the input and output of the program demonstrated in the video. |

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. 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.

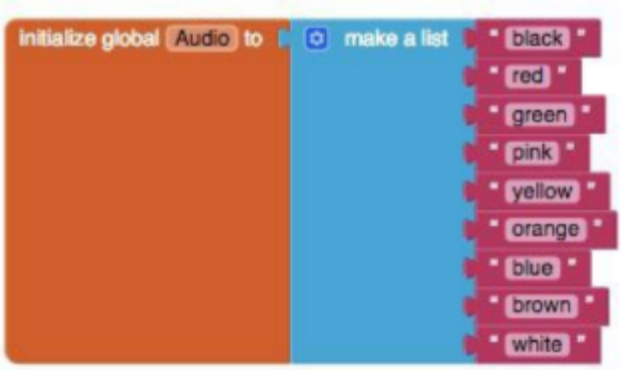
# Create Performance Task Scoring Guidelines *cont.*

| Reporting Category | Scoring Criteria | Decision Rules |
|---|---|---|
| **Row 2**<br>**Data Abstraction**<br><br>**(0-1 points)**<br>**3.B** | The written response:<br>• includes two *program code segments*:<br>  • one that shows how *data has been stored in this list* (or other *collection type*); and<br>  • one that shows the data in this same *list being used* as part of fulfilling the program's purpose;<br>**AND**<br>• identifies the name of the variable representing the list being used in this response;<br>**AND**<br>• describes what the data contained in this list is representing in the program. | **Consider ONLY written response 3b when scoring this point.**<br><br>The written response must include two clearly distinguishable program code segments, but these segments may be disjointed code segments or two parts of a contiguous code segment.<br><br>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.<br><br>**Do NOT award a point if any one or more of the following is true:**<br>• the use of the list is trivial and does not assist in fulfilling the program's purpose. |

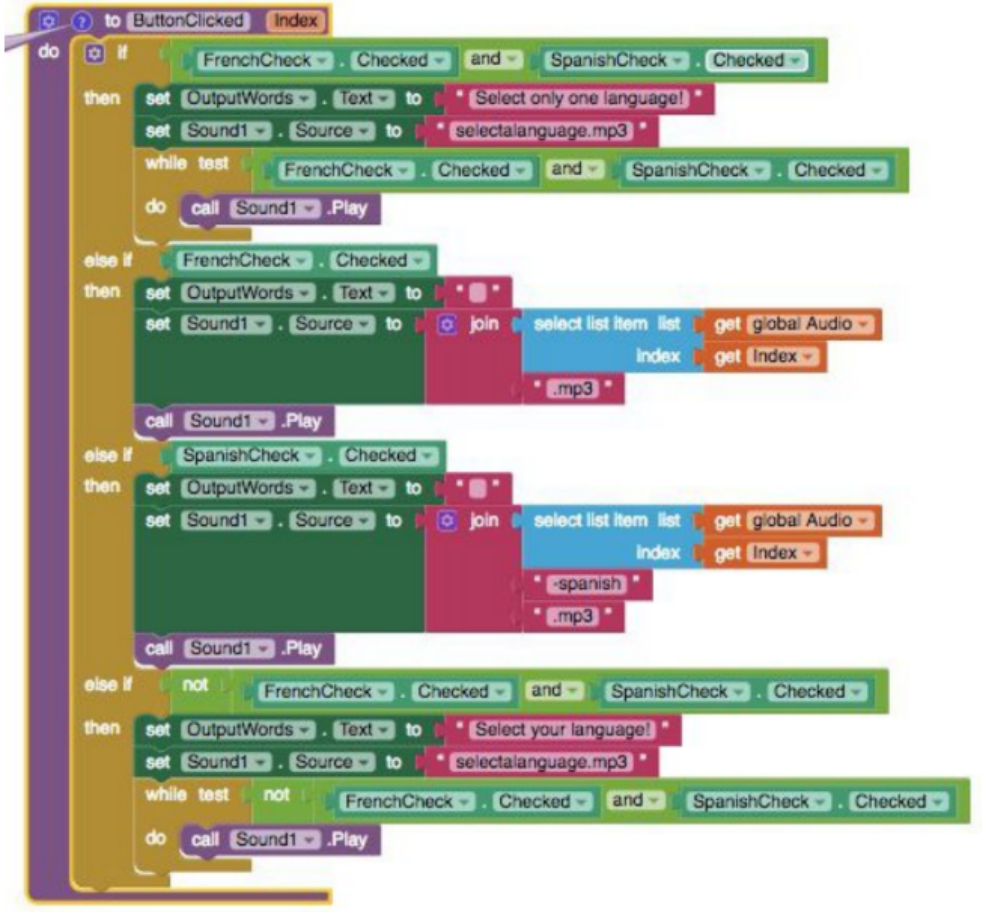


- The following are found in the terminology:
  - Program Code Segment
  - List
  - Data has been stored in this list
  - Collection Type
  - List being used

# Sample A

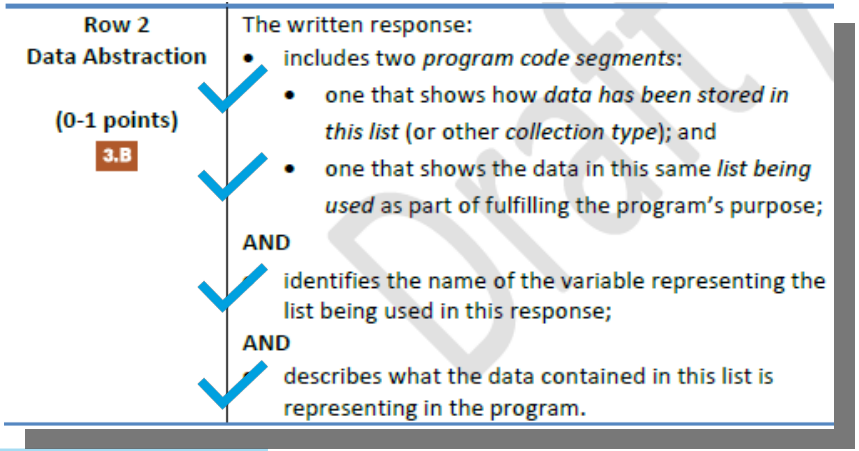



| | |
|---|---|
| **Row 2**<br>**Data Abstraction**<br><br>**(0-1 points)**<br>**3.B** | The written response:<br>• includes two *program code segments*:<br>✓ one that shows how *data has been stored in this list* (or other *collection type*); and<br>✓ • one that shows the data in this same *list being used* as part of fulfilling the program's purpose;<br>**AND**<br>• identifies the name of the variable representing the list being used in this response;<br>**AND**<br>• describes what the data contained in this list is representing in the program. |



# Sample A

- Read through the response for 3b.



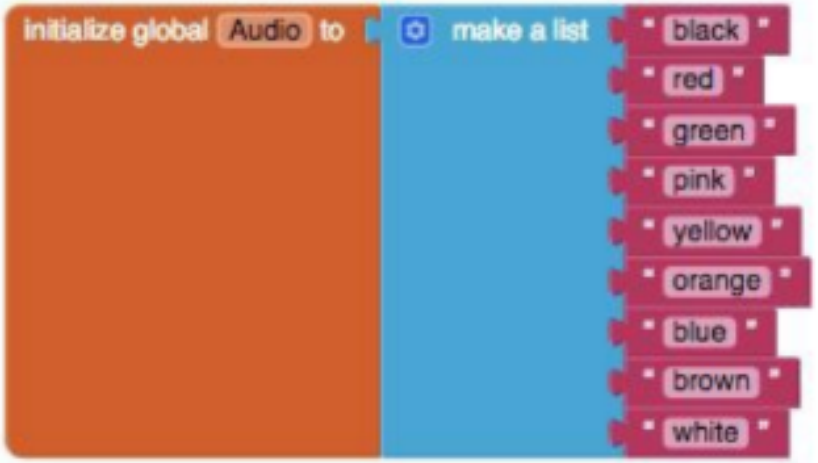
| | |
|---|---|
| **Row 2**<br>**Data Abstraction**<br><br>**(0-1 points)**<br>**3.B** | The written response:<br>• includes two *program code segments*:<br>  • one that shows how *data has been stored in this list* (or other *collection type*); and<br>  • one that shows the data in this same *list being used* as part of fulfilling the program's purpose;<br>**AND**<br>identifies the name of the variable representing the list being used in this response;<br>**AND**<br>describes what the data contained in this list is representing in the program. |

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 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.

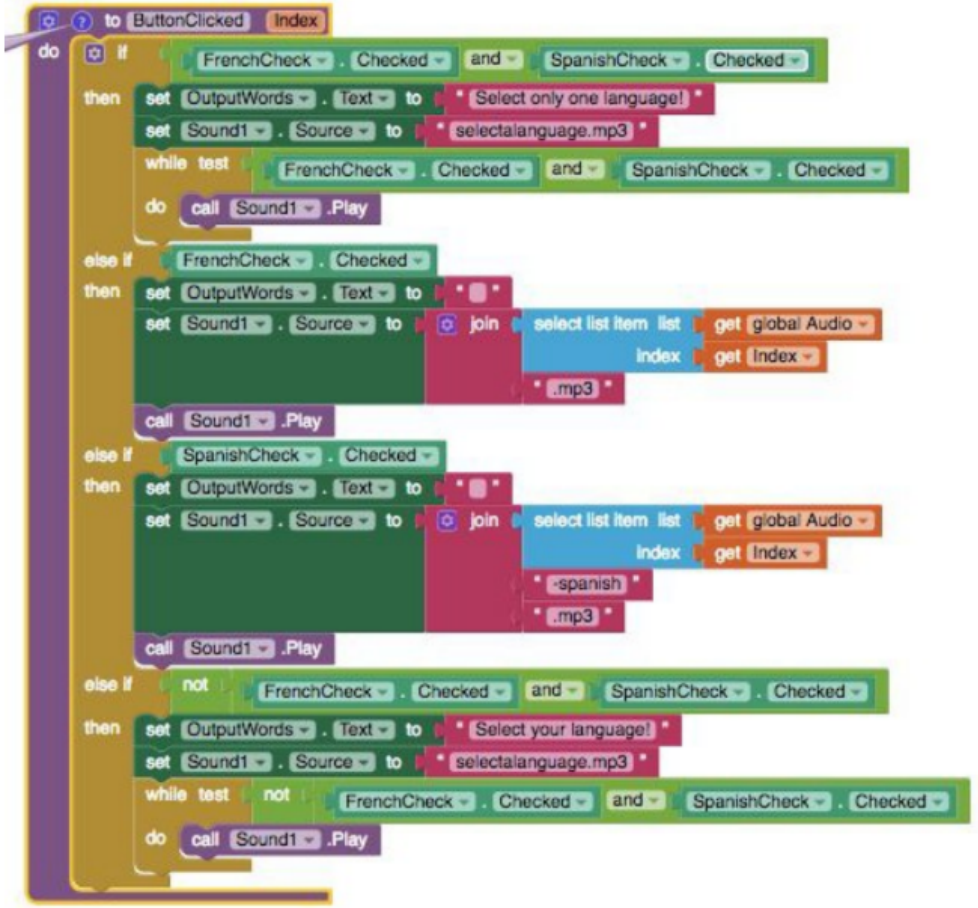
# Create Performance Task Scoring Guidelines *cont.*

| Reporting Category | Scoring Criteria | Decision Rules |
|---|---|---|
| **Row 3**<br>**Managing Complexity**<br><br>**(0-1 points)**<br>**3.C** | The written response:<br>• includes a program code segment that shows a list being used to manage complexity in the program;<br>**AND**<br>• 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.**<br><br>Responses that do not earn row 2, may still earn this row.<br><br>**Do NOT award a point if any one or more of the following is true:**<br>• 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.); OR<br>• the written response does not name the selected list (or other collection type); OR<br>• the use of the list is irrelevant or not used in the program; OR<br>• the explanation does not apply to the selected list; OR<br>• the explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program; OR<br>• the solution without the list is implausible, inaccurate, or inconsistent with the program; OR<br>• 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; OR<br>• 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. |

# Sample A

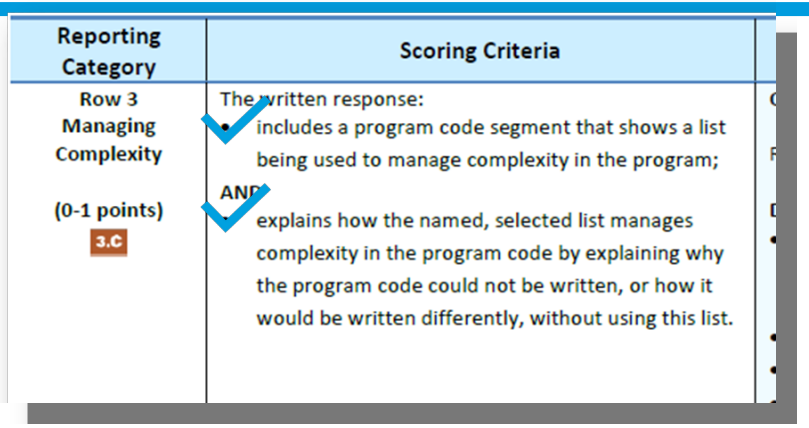



| Reporting Category | Scoring Criteria |
|---|---|
| Row 3 Managing Complexity<br><br>(0-1 points)<br>3.C | The written response:<br>• includes a program code segment that shows a list being used to manage complexity in the program;<br>AND<br>• explains how the named, selected list manages complexity 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. |



# Sample A

- Read through the response for 3b.



| Reporting Category | Scoring Criteria |
|---|---|
| Row 3 Managing Complexity<br><br>(0-1 points)<br>3.C | The written response:<br>• includes a program code segment that shows a list being used to manage complexity in the program;<br><br>AND<br><br>• 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. |

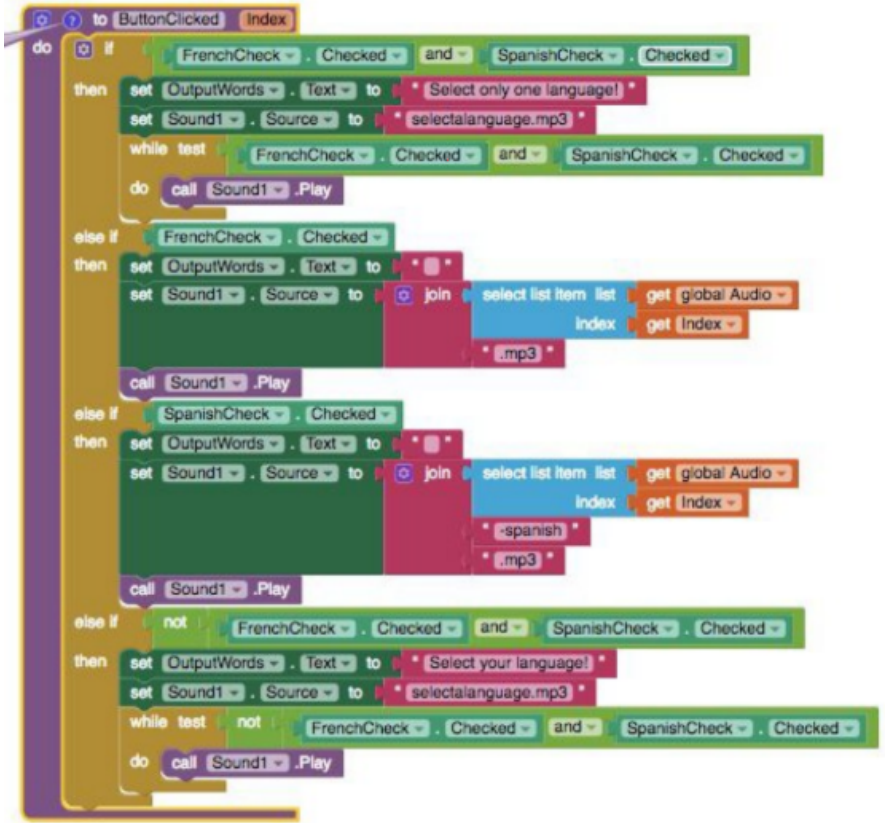
The data contained in the list Audio is the list of available color names. It represents all the colors a user can pick for the program in English. These are used to create the corresponding Spanish or French audio files based on which language is selected. When a button is pressed, it will get the English 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 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.

# Create Performance Task Scoring Guidelines *cont.*

| Reporting Category | Scoring Criteria | Decision Rules |
|---|---|---|
| **Row 4 Procedural Abstraction**<br><br>**(0-1 points)**<br>**3.B** | The written response:<br>• includes two program code segments:<br>  • one of a *student-developed procedure*. The procedure:<br>  - contains at least one *parameter*; and<br>  - the parameter has an effect on the functionality of the procedure;<br>  • one showing where the student-developed procedure is being called;<br>**AND**<br>• 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.**<br><br>The procedure must be student developed, but could be developed collaboratively with a partner.<br><br>If multiple procedures are included, use the first procedure to determine whether the point is earned.<br><br>**Do NOT award a point if any one or more of the following is true:**<br>• the code segment consisting of the procedure is not included in the written responses section; OR<br>• the procedure is a built-in or existing procedure or language structure, such as an event handler or main method, where the student only implements the body of the procedure rather than defining the name, return type (if applicable) and parameters; OR<br>• the written response describes what the procedure does independently without relating it to the overall function of the program. |

- The following are found in the terminology:
  - Student-developed Procedure / Algorithm
  - Procedure
  - Parameter
  - Arguments

# Sample A

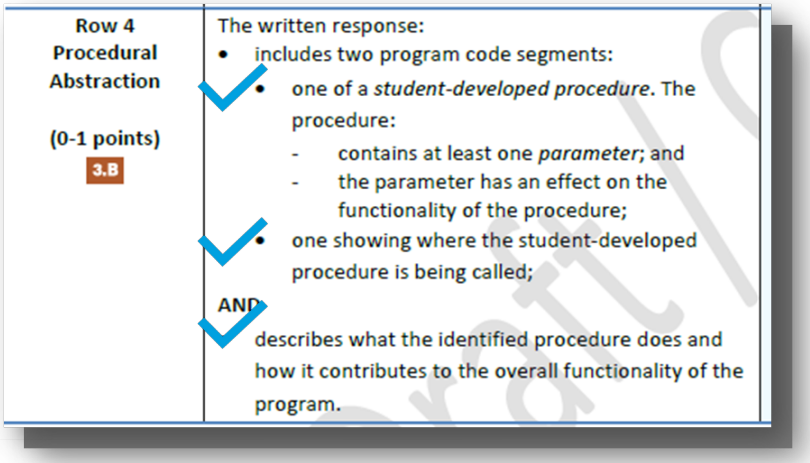




| | |
|---|---|
| **Row 4 Procedural Abstraction**<br><br>**(0-1 points)**<br>**3.B** | The written response:<br>• includes two program code segments:<br>✓ one of a *student-developed procedure*. The procedure:<br>- contains at least one *parameter*; and<br>- the parameter has an effect on the functionality of the procedure;<br>✓ • one showing where the student-developed procedure is being called;<br>**AND**<br>• describes what the identified procedure does and how it contributes to the overall functionality of the program. |

# Sample A

- Read through the response for 3c.



| | |
|---|---|
| **Row 4 Procedural Abstraction**<br><br>**(0-1 points)**<br>**3.B** | The written response:<br>• includes two program code segments:<br>  • one of a *student-developed procedure*. The procedure:<br>  - contains at least one *parameter*; and<br>  - the parameter has an effect on the functionality of the procedure;<br>  • one showing where the student-developed procedure is being called;<br>**AND**<br>describes what the identified procedure does and how it contributes to the overall functionality of the program. |

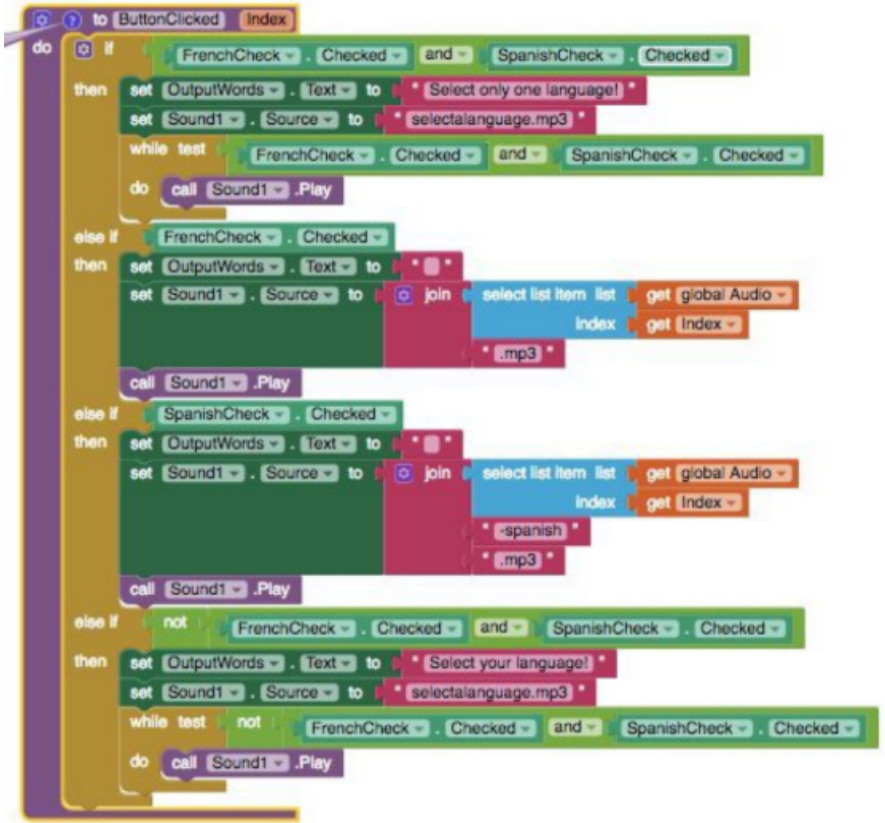
# Create Performance Task Scoring Guidelines *cont.*

| Reporting Category | Scoring Criteria | Decision Rules |
|---|---|---|
| **Row 5 Algorithm Implementation**<br><br>**(0-1 points)**<br>**2.B** | The written response:<br>• includes a program code segment of a *student-developed algorithm* that includes:<br>  • *sequencing*; AND<br>  • *selection*; AND<br>  • *iteration*;<br>**AND**<br>• explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. | **Consider ONLY written response 3c when scoring this point.**<br><br>Responses that do not earn row 4 may still earn this row.<br><br>The algorithm being described can utilize existing language functionality or library calls.<br><br>An algorithm that contains selection and iteration, also contains sequencing.<br><br>An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point.<br><br>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.<br><br>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.<br><br>**Do NOT award a point if any one or more of the following is true:**<br>• the response only describes what the selected algorithm does without explaining how it does it; OR<br>• the description of the algorithm does not match the included program code; OR<br>• the code segment consisting of the selected algorithm is not included in the written response; OR<br>• 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); OR<br>• the use of either the selection or the iteration is trivial and does not affect the outcome of the program. |

The following are found in the terminology:

- Sequencing
- Selection
- Iteration

# Sample A




| Reporting Category | Scoring Criteria |
|---|---|
| **Row 5**<br>**Algorithm Implementation**<br><br>**(0-1 points)**<br>**2.B** | The written response:<br>• includes a program code segment of a *student-developed algorithm* that includes:<br>✓ *sequencing*; AND<br>✓ *selection*; AND<br>✓ *iteration*;<br>**AND**<br>• explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. |

# Sample A

- Read through the response for 3c.

| Reporting Category | Scoring Criteria |
|---|---|
| Row 5<br>Algorithm Implementation<br><br>(0-1 points)<br>2.B | The written response:<br>• includes a program code segment of a *student-developed algorithm* that includes:<br>✓ *sequencing*; AND<br>✓ *selection*; AND<br>✓ *iteration*;<br>AND<br>✓ explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. |

# Create Performance Task Scoring Guidelines *cont.*

| Reporting Category | Scoring Criteria | Decision Rules |
|---|---|---|
| **Row 6**<br>**Testing**<br><br>**(0-1 points)**<br>**1.B** | The written response:<br>• 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;<br>**AND**<br>• describes the condition(s) being tested by each call to the procedure;<br>**AND**<br>• identifies the result of each call. | **Consider ONLY the written response for 3d and the selected procedure identified in written response 3c.**<br><br>Responses that do not earn row 4 may still earn this row.<br><br>**Do NOT award a point if any one or more of the following is true:**<br>• a procedure is not identified in written response 3c or the procedure does not have a parameter; OR<br>• the written response for 3d does not apply to the procedure in 3c; OR<br>• the two calls cause the same segment of code in the algorithm to execute even if the result is different; OR<br>• the response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program; OR<br>• the identified results of either call are implausible, inaccurate, or inconsistent with the program. |

# Sample A

- Read through the response for 3d.

| Reporting Category | Scoring Criteria |
|---|---|
| Row 6<br>Testing<br><br>(0-1 points)<br>1.B | The written response:<br>• 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;<br>AND<br>• describes the condition(s) being tested by each call to the procedure;<br>AND<br>• identifies the result of each call. |

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.