

## 2. Written Responses

Submit one PDF file in which you respond directly to each prompt. **Clearly label your responses 2a–2d in order. Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.**

2a. Provide a written response or audio narration in your video that:

- identifies the programming language;
- identifies the purpose of your program; and
- explains what the video illustrates.

The program is a study guide app. The program was created in App Inventor 2. The purpose of this app is so that the user can put in whatever terms and definitions they desire and study off them later on flashcards. At first, the user puts in the desired term and definition in the two textboxes and then click submit allowing them to put in different terms and definitions after. When they are finally done, they click the study button which takes away the terms and definitions and just shows the term as a flashcard. Then to see what the definition is, they can click the flash card. The video shows the user plugging in values and later studying them as if there were real flashcards.

*(Must not exceed 150 words)*

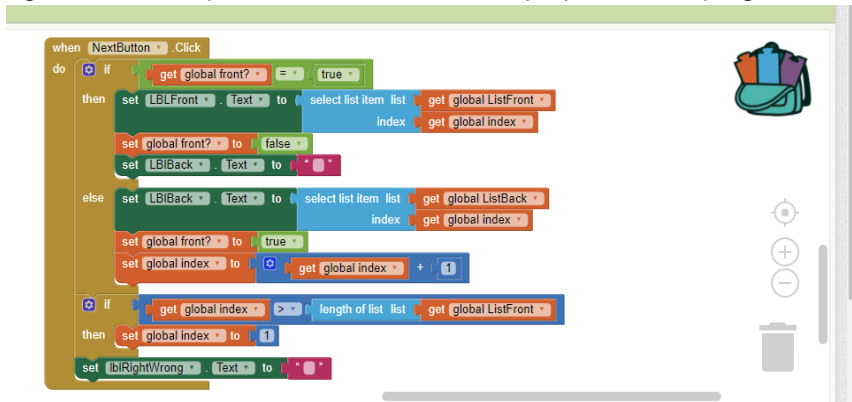
2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development.

There were many problems that arose while coding the program. One of the early problems encountered was deciding how I should set up my study guide. For example, I could have chosen to do flashcards along with doing multiple choice. However, I felt that the flash cards would be more effective and efficient way of creating this app. Furthermore, when making this study guide app, I felt that there needed to be something else that could have made the study guide more useful for the reader. Originally, there were just flashcards, but I felt there was something else that could be done. So I included another button that allowed the user to type in the definition as the word was being given. This was a major development addition as it is more effective for the user to write the information than by just looking at cards. This is also more effective for memorization.

*(Must not exceed 200 words)*

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within

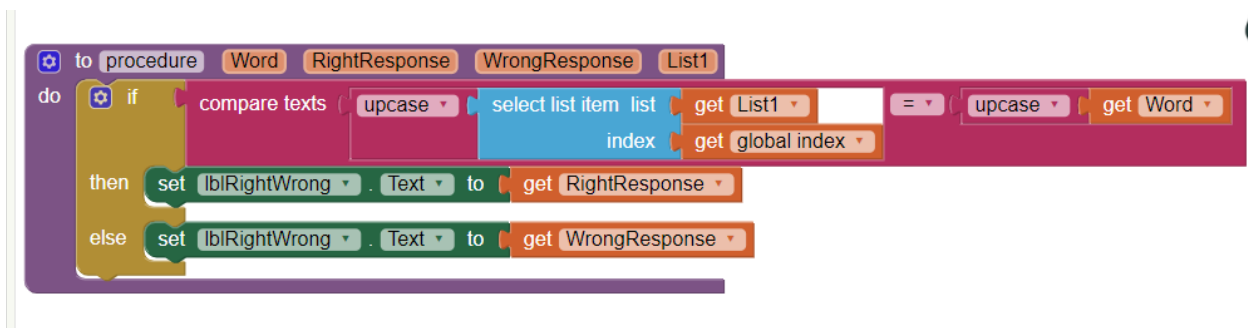
your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program.



This particular algorithm is essential to the program because it allows the user to go to the next flashcard. When the next button is clicked, it displays the label front text and then doesn't show the back text. The index also determined which flashcard that you are on, in which you keep going to the next term as the next button is clicked. Furthermore, if the index is bigger than the number of items in the list then it restarts back to 1, or the first item in the list. This is the same for the other algorithm as they both use an index. One of the independent algorithms makes so that user can type in the term as the other algorithm is displaying the definition as a flashcard. Together as a combination, this makes it so that the user has a study guide environment in which they can type in the necessary term to the definition and then be able to go to the next set of terms.

*(Must not exceed 200 words)*

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program.



This particular abstraction is used to determine if the word you type in, is correct. This abstraction uses mathematical concepts by determining if the word you type in and the actual term, are equal. If they do happen to be equal, then this will be shown in the "IbIWriteWrong." Furthermore, this abstraction uses logical concepts by determining if the word the user types in is true, then it will be displayed as correct through the Write Wrong label. However if the word the user types in is a false word, then it will show that it is incorrect through the Write Wrong label. By creating this abstraction it makes the general coding clearer and easier to read as it is already being used once.

*(Must not exceed 200 words)*

### 3. Program Code

Capture and paste your entire program code in this section.

- Mark with an **oval** the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.
- Mark with a **rectangle** the segment of program code that represents an abstraction you developed.
- Include comments or acknowledgments for program code that has been written by som