# Quiz App Search Project Short Handout

| Getting StartedOpen App Inventor with your finished Quiz App from the last lesson.Add a keyword search capability to your app. For example, if the user types in NASA and clicks on the search button, you should find the question or answer with the word NASA in it and show that question. This will be a linear search through the parallel question and answer lists using a loop. |  |
| --- | --- |

# Designing the User Interface (UI)

| **UI Component**  | **Name** | **Properties** |
| --- | --- | --- |
| Horizontal Arrang. | HorizontalArrangement1 | Put at top of screen |
| TextBox | SearchText | Put in Horizontal Arrangement |
| Button | SearchButton | Put in Horizontal Arrangement |

# Coding the App

|  **Event Handlers** | **Algorithms** |
| --- | --- |
| SearchButton.Click | Call a procedure **searchProcedure** that you will make below with the argument **SearchText.Text** |

#

| **Procedural Abstraction** | **Algorithms** |
| --- | --- |
| displayQuestion() | Move the code that displays a question from Screen1.Initialize and NextButton.Click into this new procedure displayQuestions(). It will also be called from the searchProcedure below. |
| searchProcedure(keyword) | -Create a procedure with a parameter keyword-Loop through the answersList using a “for each number” loop where number goes from 1 to the length of the questionList. -If the selected list item from questionList at index number contains the keyword or if the selected list item from the answerList at index number contains the keyword, then set index variable to that number and call displayQuestion(). |

# Hints

The search procedure will do a search with a loop that steps through a list like below using a loop counter i for the index of the loop. In this loop, the counter variable *number* has been renamed to i.



Inside the loop, you will need an if statement and contains text blocks from the Text drawer to test if the search word typed in by the user is in each question or answer.



# Testing the App

|  **Inputs** | **Expected Outputs** | **Actual Outputs** |
| --- | --- | --- |
| Turing (typed into search textbox) | Turing image and question displays | ? |
| xyz | The question does not change. | ? |

# Iterative Design: Enhancements

* Does your app find both “Turing” and “turing”? You can convert both the keyword and the questions and answers to the same case using the upcase block in the Text drawer .
* What happens when your app does not find the keyword? Probably nothing. You can have your app print out “Not found!” using a Notifier.ShowAlert. However, you have to go through the whole loop checking every question and answer before you can say the keyword was not found. In these situations, it’s useful to have a boolean flag variable, for example one called *found* that is set to false. If you find the keyword, you can set this variable found to true. After the loop, you can test the found variable to decide whether to print out “Not found!”.