**Homework 2**

You saw the corresponds method used with two arrays of strings. Make a call to corresponds that checks whether the elements in an array of strings have the lengths given in an array of integers. **(1)**

Write a function largest(fun: (Int) => Int, inputs: Seq[Int]) that yields the largest value of a function within a given sequence of inputs. For example, largest(x => 10 \* x - x \* x, 1 to 10) should return 25. Don’t use a loop or recursion. **(2)**

Modify the previous function to return the *input* at which the output is largest. For example, largestAt(fun: (Int) => Int, inputs: Seq[Int])should return 5. Don’t use a loop or recursion. **(3)**

Implement an unless control abstraction that works just like if, but with an inverted condition. Does the ﬁrst parameter need to be a call-by-name parameter? Do you need currying? **(4)**

Write a function values(fun: (Int) => Int, low: Int, high: Int) that yields a collection of function inputs and outputs in a given range. For example, values(x => x \* x, -5, 5) should produce a collection of pairs (-5, 25), (-4, 16), (-3, 9), . . . , (5, 25). **(5)**