Zachary Bell

CS1401

Ms. Roy

Lab 7

**Activity 1**

Pseudo Code **closest(int[] A)**:

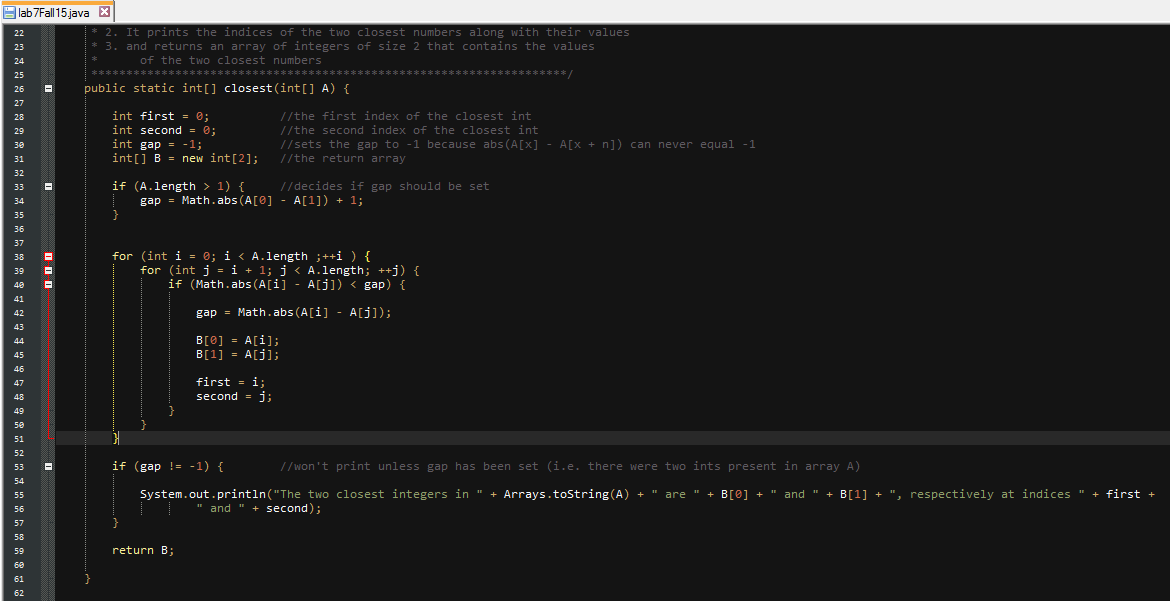
1. Set gap to |A[0] – A[1]|
2. For (i = 0; i < A.length; ++i)
   1. For (j = i + 1; j < A.length; ++j)
      1. If (|A[i] – A[j]| < gap)
         1. gap <- |A[i] – A[j]|
         2. B[0] <- A[i]
         3. B[1] <- A[j]
         4. first <- i
         5. second <- j
3. Give “The two closest integers in “ + A + “are” + B[0] + “and” + B[1] + “, respectively at indices” + first + “and” + second
4. Return B

Pseudo Code **prefix(String str1, String str2):**

1. minLength <- min( length of str1, length of str2)
2. for (i = 0; i < min( length of str1, length of str2); ++i)
   1. if (charAt str1 (i) is equal to charAt str2 (i))
      1. - -minLength
      2. if minLength is equal to 0
         1. isPrefix <- true
3. give isPrefix

**Activity 2**

For my white box test I will be attempting to test out independent control paths, the logical decisions, and limits of the loops in my program.



The first possible point of conflict in my method is an if-statement (logical decision). The two possible true and false values are reached with either A.length being greater than 1 or less than or equal to one. Since an array can’t be any smaller than 0 the only relevant array lengths to test are 0, 1, and an array greater than 1. A print statement for array B is added at the end. When the test is run with a length of 0 the method doesn’t print or return two values “closest” to one another and only returns an array B with uninitialized values. When the test is run with a length of 1 the program only prompts the user for 1 number and returns a blank array. When run with a length of greater than 1 the program returns expected results.

The next point in the method to test is the for loops. Since the method sets the boundaries of the loop based on the length of the array we would need to test if the first two values of the array and if the last two values of the array would be displayed. We’d also need to test to see what would happen if two different sets of values had the same gap. Given the array [1, 2, 4, 6, 8] the 1 and 2 should be returned. The test passes and [1, 2] is returned. Now we can test [2, 4, 6, 8, 9] which should return [8, 9]. This test also passes and [8, 9] are returned. Now we can test a small array of 2 numbers [1, 2]. The array returns [1, 2] as we expected. Since negative numbers don’t impact the bounds of the loop it’s not necessary to test them.

Inside the for loop is another logical decision. We’ve already tested a few ideal values and it works but given that it will be false if the absolute value of two numbers is greater than what the gap is set at we can test the array with two identical values to check it. [2, 3, 5, 2, 8] returns [2, 2] which is what we wanted it to return. Again, since it is taking an absolute value we don’t have to worry about negatives but just in case [-2, 3, 4, 5, -2] returns [-2, -2]. For identical cases [2, 3, 2, 5, 6, 7, 9, 7] we want the first instance of a “closest” value to be returns. When run the array returns [2, 2] which is what we wanted it to return. A more intricate program could later be made that would either return [2, 2, 7, 7] or at least let the user know there are two equal “closest” values.

Finally, we have another logical decision if-statement at the end. We’ve actually already tested this with arrays shorter than 2 and it wouldn’t print its contents unless the array is greater than 1 so we know it works.