```
class IsLongerThan {

  static ______________ isLongerThan(___________________________________________) {



  }
  public static void main(String[] args) {
    assert isLongerThan("abc", 2) == true;
    assert isLongerThan("abc", 4) == false;
    assert isLongerThan("password", 7) == false;
  }
}
```

IsLongerThan.java

`String.length()`

Returns the number of characters in this string

```
class LetterGrade {
  static String letterGrade(int points) {
    if(points >= 90) {
      return "A";
    }
    else if(points >= 80) {
      return "B";
    }
    else if(points >= 70) {
      return "C";
    }
    else {
      return "F";
    }
  }
  public static void main(String[] args) {
    assert letterGrade(7).equals("F");
    assert letterGrade(95).equals("A");
    assert letterGrade(88).equals("B");
  }
}
```

Letter.java

```
class Phase {
  // Takes a temperature in degrees celsius and
  // returns "water", "steam", or "ice"
  static ______________ phaseOfWater(____________________________) {









  }
  public static void main(String[] args) {
    assert phaseOfWater(50).equals("water");
    assert phaseOfWater(105).equals("steam");
    assert phaseOfWater(-9).equals("ice");
  }
}
```

Phase.java

```
class Practice {

  _________ between(__________________________________________________) {


  }

  __________ abs(_____________________________________) {




  }

  __________ addLengths(________________________________) {


  }

  public static void main(String[] args) {





  }
}
```

Practice.java

Write the following methods and **two assert tests** for each:

`between` that takes three numbers and returns true if the first is in between the other two.

`abs` that takes a number and returns its absolute value.

`addLengths` that takes two strings and returns the sum of their lengths.

```
class Truncate {






  public static void main(String[] args) {



  }
}
```

Truncate.java

Write a method truncate that takes a string and a number n. If the string is length n or shorter, return the string. If it's **longer**, return a new string that's the first n characters of the string concatenated with "..."

`String.substring(int start, int end)`

Returns a new string made from the characters in this string from index `start` (inclusive) to index end (exclusive).