Modify your programming exercise #4 to compute annual salary increase and new annual salary in two separate functions, rather than inline. See requirement #6 for details.

Requirements:
1. Include header comments that give the following information:

   - source file name
   - your last name, your first name
   - 110A - your section number
   - Task of program

2. Include step comments for the major logic sections of your program.

3. For each employee, the user must enter the employee's full name and previous annual salary. Use the following test data:

<table>
<thead>
<tr>
<th>Name</th>
<th>Previous annual salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jose Bartolo</td>
<td>100000</td>
</tr>
<tr>
<td>Alice Kellenberger</td>
<td>80000</td>
</tr>
<tr>
<td>Pat Sweeney</td>
<td>76000.34</td>
</tr>
</tbody>
</table>

   end (trailer data)

4. Your employee report should have the following format:

   line 1: employee's name
   line 2: previous annual salary
   line 3: annual salary increase
   line 4: retroactive pay due the employee
   line 5: new annual salary
   line 6: new monthly salary

   Note that each output item is to be printed on a separate line and preceded by an appropriate label. Print one blank line between the data entry and the employee report.

5. Include the code that will format all floating-point output to have a precision of 2.
6. In addition to main(), you should have a function that takes one double argument, previous annual salary, and returns the annual salary increase as a double. This function should use an increase of 7.6% for employees having an annual salary of 80,000 or less and 6.5% for employees having an annual salary that is greater than 80,000.

You should have a second function that takes two double arguments, previous annual salary and annual salary increase, and returns the new annual salary as a double.

7. Use a while loop that will exit when the employee's name is end. The loop exit point should be at the while statement. You must have a priming and duplicate input of the employee's name.

8. Use good programming style; that is, vertically align the matching braces, indent the block of statements, and use whitespace (blank lines in your program) judiciously.