Experiments

- Ex 1: Integer Average (课后习题5.6)

Lab Objectives

- Using sentinel-controlled repetition with a for loop.

The follow-up question and activity also will give you practice:
- Using counter-controlled repetition with a for loop.

Description of the Problem

Write a program that uses a for statement to calculate and print the average of several integers. Assume the last value read is the sentinel 9999. A typical input sequence might be 10 8 11 7 9 9999 indicating that the program should calculate the average of all the values preceding 9999.

Sample Output

```
Enter integers (9999 to end):
10 8 11 7 9 9999
The average is: 9
```

Problem-Solving Tips

1. When used for sentinel-controlled repetition, a for loop can be written much like a while loop, using the same loop-continuation condition as a while loop.
2. When performing sentinel-controlled repetition, a for loop does not need to increment any counter variable. But it can still initialize a variable if so desired.

Follow-up Question

1. Modify the program to perform counter-controlled repetition with a for loop. Assume that the first integer entered by the user represents the number of subsequent integers that the user will input to be averaged.

Sample output:

```
Enter integers (first integer should be the number of subsequent integers):
5 12 7 4 19 6
The average is: 9.6
```

- Ex 2: Pythagorean Triples (课后习题5.20)

Lab Objectives

- Using counter-controlled repetition.
- Using “brute force” to solve a problem.
• Nesting for loops.
The follow-up questions and activities will also give you practice:
• Using break statements.
• Using continue statements.
• Using long integers.

Description of the Problem
A right triangle can have sides that are all integers. A set of three integer values for the sides of a right triangle is called a Pythagorean triple. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Find all Pythagorean triples for side1, side2 and hypotenuse all no larger than 500. Use a triple-nested for loop that tries all possibilities. This is an example of “brute force computing.” You will learn in more advanced computer-science courses that there are many interesting problems for which there is no known algorithmic approach other than using sheer brute force.

Sample Output

<table>
<thead>
<tr>
<th>Side 1</th>
<th>Side 2</th>
<th>Side 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>319</td>
<td>360</td>
<td>481</td>
</tr>
<tr>
<td>320</td>
<td>336</td>
<td>464</td>
</tr>
<tr>
<td>325</td>
<td>360</td>
<td>485</td>
</tr>
<tr>
<td>340</td>
<td>357</td>
<td>493</td>
</tr>
</tbody>
</table>

A total of 386 triples were found.

Problem-Solving Tips
1. This program does not require any input from the user.
2. This program can take a significant amount of time to run, depending on your computer’s processor speed. If you have a CPU monitor available on your system, it is worth taking a look at it when this program executes.
3. Do not be concerned that you are trying values that do not seem to make sense, such as a 1–1–500 triangle.
4. Remember that brute-force techniques try all possible values.
5. The formula for the Pythagorean Theorem is \( \text{hypotenuse}^2 = (\text{side} \, 1)^2 + (\text{side} \, 2)^2 \).
6. To avoid producing duplicate Pythagorean triples, start the second for loop at \( \text{side} \, 2 = \text{side} \, 1 \) and the third for loop at \( \text{hypotenuse} = \text{side} \, 2 \). This way, when a Pythagorean triple is found, \( \text{side} \, 1 \) will be the shortest side of the triangle and hypotenuse will be the longest side.

Follow-up Question
1. How many times did this program execute the innermost for loop? Add another counter to the program that counts the number of times this loop iterates. Declare a new variable of type long, named loopCounter and initialize it to 0. Then add a statement in the innermost for statement that increments loopCounter by 1. Before exiting the program, print the value of loopCounter. Do the numbers match?

2. Add a break statement to the program inside the innermost for loop. This break statement should be called after the 20th Pythagorean triple is found. Explain what happens to the program after the 20th triple is found. Are all three for loops exited, or just the innermost one? What happens when the break statement is placed inside the middle loop? The outermost loop?

3. Add a continue statement to the program that prevents a Pythagorean triple from being found when side1 is equal to 8. Using your solution to Follow-Up Question 1, calculate how many times this new program executes the innermost for loop. Explain why the continue statement affected the output.

4. Explain why a long variable is used for hypotenuseSquared and sideSquared. Modify the program so that they are both of type short instead of type long. Rerun the program. What happens?