

Title

Exploring floating point precision

Problem Description

In this problem we will be exploring floating point precision and the creation of good test cases. Create two variables of type float with names x and y. Have the user enter values to store in x and y. Output the sum of the two variables.

The real fun lies in trying to make test cases that break the code. If you enter relatively small numbers, like 10 and 20, you will probably end up with the correct answer (30, in our case). Come up with test cases that allow you to test the following:

- 1) What happens if x is very large and y is very small?
- 2) Enter values of x and y until you find out how large x can be and how small y can be before the result is not what you expect.
- 3) How many nonzero digits can you enter for both x and y before the output is not what you expect? For example, if $x = 123456789$ and $y = 1$, I would expect the result to be 123456790. Instead, my computer outputs 1.23457e+008.

Note: you'll see the notation 1.23457e+008 a lot. This is the computer's way of saying that the answer is 1.23457×10^8 .

Testing

Since your goal is to come up with your own test cases, we will not provide you with any here.

Time Target

Since this problem relies on exploration, it's best not to limit your time. Thoroughly test your code, try many different combinations, and try to get a feel for the behaviour of the computer as it works with different types of numbers.

Section

Variables and Fundamentals