# Sentinel-Controlled Loop: Averaging Numbers

**The goal for this exercise** is to make sure that you can use sentinel-controlled loops.

Your textbook may explain what is meant by ‘sentinel-controlled loop’. If it doesn’t, you may find it hard to locate this term on the Internet, since it’s a not-quite standard word. Basically, you can think of loops as falling into different categories – for example, a loop that prints out all the numbers from 1 up to (and including) 10 serves as a good example of a ‘counting loop’.

‘Counting loops’ are defined both by the fact that they count through a range of numbers, and (more importantly) by the fact that *when the loop begins to execute, you know how many times the loop will be run.* You might not know at compile time (for example, if your program asks the user how many numbers to print out), but by the time the loop starts, the number of iterations (of times through the loop) is known (continuing the example – by the time the loop starts, you know how many times the user wants you to go through the loop). In C#, the **for** loop syntax was created to make counting loops quick and easy to write.

In contrast, a ‘sentinel-controlled loop’ is a loop that **doesn’t know when it will exit when the loop starts executing.** Maybe your code asks a database for a set of records, and then uses a loop similar to:

**while( there is another record )**

**do something with that record.**

In this case, the loop doesn’t actually know how many times it’s going to run at the time at which the loop starts. (In this particular example, it might be possible to re-write this as a counting loop by asking the database how many records there are, but regardless - the exact loop, as written above, doesn’t know how many records there are). In C#, sentinel-controlled loops are typically written as **while** (or maybe **do…while**) loops

**What you need to do for this exercise:**

1. In the starter project, fill in the Sentinel\_Controlled\_Loop. RunExercise method, so that it will compute the average of a series of integers entered by the user. The user will indicate that he is finished entering numbers by entering a –1 (the -1 will NOT be included in the average). An example transcript of what your program might look like follows. User input is in **bold**.

Welcome to NumberAverager!

What is your next number? (Type -1 to exit) **10**

What is your next number? (Type -1 to exit) **10**

What is your next number? (Type -1 to exit) **20**

What is your next number? (Type -1 to exit) **0**

What is your next number? (Type -1 to exit) **-10**

What is your next number? (Type -1 to exit) **-1**

The average of your numbers is: 6