

# Examples of using a while loop

## 1. Summing numbers using a while loop

Let's write a program that reads numbers from the keyboard (as long as the user enters something that looks like a number) and then displays their sum on the screen. Here's how the code of such a program would look (we're only showing the code inside the `main` method).

Code	Explanation
<pre>Scanner console = new Scanner(System.in); int sum = 0; while (console.hasNextInt()) {     int x = console.nextInt();     sum = sum + x; } System.out.println(sum);</pre>	<p>Create a <code>Scanner</code> object to read data from the console.</p> <p>We will store the sum of the numbers in the <code>sum</code> variable.</p> <p>As long as numbers are entered from the console</p> <p>read the next number into the <code>x</code> variable.</p> <p>Add <code>x</code> to the sum of numbers (the <code>sum</code> variable).</p> <p>Display the calculated sum on the screen.</p>

## 2. Finding the maximum number using a while loop

Our second program will also read numbers from the keyboard (as long as the user enters something number-like), but now we want to display the largest of the numbers entered. Here's how the code of such a program would look (we're only showing the code inside the `main` method).

Here's an interesting point: if all the numbers entered from the keyboard are negative, then the program will display 0. Which is incorrect.

As a result, the initial value of the max variable should be as small as possible.

### Option 1:

You can set it equal to `-2,000,000,000` (negative two billion). This isn't a bad start.

### Option 2:

Assign the smallest possible `int` value. There's a special constant for this: `Integer.MIN_VALUE`;

### Option 3:

Better yet, initialize max with the first number entered. This is the best option. But this will only work if the task conditions require the user to enter at least one number.