Suppose we have a class:

public class Foo {

  public void first() { print("first"); }

  public void second() { print("second"); }

  public void third() { print("third"); }

}

The same instance of Foo will be passed to three different threads. Thread A will call first(), thread B will call second(), and thread C will call third(). Design a mechanism and modify the program to ensure that second() is executed after first(), and third() is executed after second().

**Example 1:**

**Input:** [1,2,3]

**Output:** "firstsecondthird"

**Explanation:** There are three threads being fired asynchronously. The input [1,2,3] means thread A calls first(), thread B calls second(), and thread C calls third(). "firstsecondthird" is the correct output.

**Example 2:**

**Input:** [1,3,2]

**Output:** "firstsecondthird"

**Explanation:** The input [1,3,2] means thread A calls first(), thread B calls third(), and thread C calls second(). "firstsecondthird" is the correct output.

**Note:**

We do not know how the threads will be scheduled in the operating system, even though the numbers in the input seems to imply the ordering. The input format you see is mainly to ensure our tests' comprehensiveness.