1.

（1）Identify the fault.

应该遍历x数组中的每一个数，for (int i=x.length-1; i > 0; i--)应修改为for (int i=x.length-1; i >= 0; i--)。

（2）If possible, identify a test case that does not execute the fault. (Reachability) 

x = [],y = 2

直接抛出错误，不再往下执行。

（3）If possible, identify a test case that executes the fault, but does not result in an error state.

x = [3,2,5],y = 2，Expected = 1

执行了错误代码，但内部执行状态和所期待的状态是一致的,即执行了fault,但不存在error状态。

（4）If possible identify a test case that results in an error, but not a failure. 

x = [3,3,5],y = 2，Expected = -1

结果虽然是正确的，但是应该遍历第一个数字，即有error，但最终没有导致failure。

2.

（1）Identify the fault.

应该从末位向首位遍历数组，for (int i = 0; i < x.length; i++) 应修改为for (int i=x.length-1; i >= 0; i--)。

（2）If possible, identify a test case that does not execute the fault. (Reachability) 

x = []

直接抛出错误，没有执行错误代码，即没有运行fault。

（3）If possible, identify a test case that executes the fault, but does not result in an error state.

x = [1,1,1]，Expected = -1

执行了错误代码，但内部执行状态和所期待的状态是一致的,即执行了fault,但不存在error状态。

（4）If possible identify a test case that results in an error, but not a failure.

x = [0,1,1]，Expected = 0

结果虽然是正确的，但是应该倒过来遍历，即有error，但最终没有导致failure。