



图 2.13 checkNumber()函数传入 Long 型参数

很显然，我们的程序并不支持此类型的参数。

when 语句的基本用法就是这些，但其实 when 语句还有一种不带参数的用法，虽然这种用法可能不太常用，但有的时候却能发挥很强的扩展性。

拿刚才的 `getScore()` 函数举例，如果我们不在 when 语句中传入参数的话，还可以这么写：

```
fun getScore(name: String) = when {
    name == "Tom" -> 86
    name == "Jim" -> 77
    name == "Jack" -> 95
    name == "Lily" -> 100
    else -> 0
}
```

可以看到，这种用法是将判断的表达式完整地写在 when 的结构体当中。注意，Kotlin 中判断字符串或对象是否相等可以直接使用 `==` 关键字，而不用像 Java 那样调用 `equals()` 方法。可能会觉得这种无参数的 when 语句写起来比较冗余，但有些场景必须使用这种写法才能实现。举个例子，假设所有名字以 Tom 开头的人，他的分数都是 86 分，这种场景如果用带参数的 when 语句来写就无法实现，而使用不带参数的 when 语句就可以这样写：

```
fun getScore(name: String) = when {
    name.startsWith("Tom") -> 86
    name == "Jim" -> 77
    name == "Jack" -> 95
    name == "Lily" -> 100
    else -> 0
}
```

现在不管你传入的名字是 Tom 还是 Tommy，只要是以 Tom 开头的名字，他的分数就是 86 分。

通过这一小节的学习，相信你也发现了，Kotlin 中的 when 语句相比于 Java 中的 switch 语句要灵活很多，希望你能多写多练，并熟练掌握它的用法。

### 2.4.3 循环语句

学习完了条件语句之后，接下来我们开始学习 Kotlin 中的循环语句。

熟悉 Java 的人应该都知道，Java 中主要有两种循环语句：while 循环和 for 循环。而 Kotlin 也提供了 while 循环和 for 循环，其中 while 循环不管是在语法还是使用技巧上都和 Java 中的