

外层循环的迭代做好了准备。当外层循环进行第二轮迭代时，变量  $S$  的值从之前的 3 重置为 1，内层循环再次执行。因此，角色会说“P2,S1”“P2,S2”和“P2,S3”。后面的过程类似，角色会说“P3,S1”“P3,S2”和“P3,S3”，最后一轮迭代会说“P4,S1”“P4,S2”和“P4,S3”。仔细看看脚本的运行过程，一定要理解循环的嵌套结构。

如果你已经理解了上述内容，我们就用它来解决一个有趣的数学问题：找出三个正整数  $n_1$ 、 $n_2$  和  $n_3$ ，使其满足  $n_1+n_2+n_3=25$ ， $(n_1)^2+(n_2)^2+(n_3)^2=243$ 。由于计算机擅长重复性的工作，因此，我们让它完成迭代所有的数字组合（专业术语称为穷举搜索）这项艰巨的任务。

由第一个条件可知， $n_1$  的取值范围是 1 到 23。因为除  $n_1$  外，还需要两个数字才能加到 25。（你可能发现  $n_1$  甚至不会超过 15。因为  $16^2=256$  已经超过了第二个条件的 243。这里我们先忽略第二个条件，并将  $n_1$  的上限设定为 23。）

第二个数字  $n_2$  的范围是 1 到  $(24-n_1)$ 。为什么？假设  $n_1$  等于 10，那么  $n_2$  最大只能取到 14，因为  $n_3$  最小为 1。 $n_3$  比较简单，只需要对第一个条件移项，便可得到  $n_3=25-(n_1+n_2)$ 。如果在某个  $n_1$ 、 $n_2$  的组合下，三个数字的平方和等于 243，那么任务就完成了。否则尝试其他的  $n_1$ 、 $n_2$  组合。最终的脚本如图 7-16 所示。

NestedLoops2  
.sb2

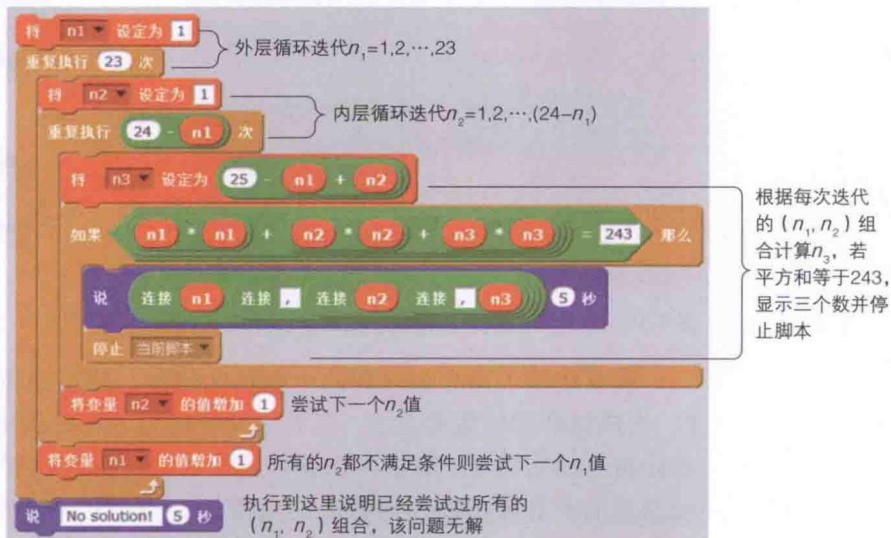


图 7-16：该脚本尝试找到三个正整数，使其和等于 25，平方和等于 243