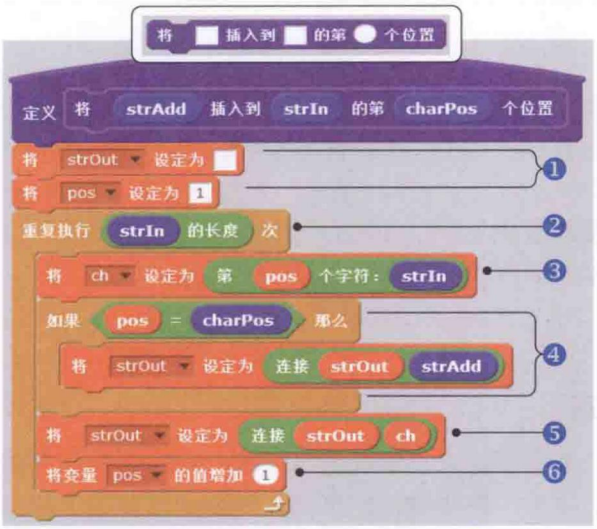

图 8-8：过程 Insert

过程首先初始化 strOut 为空字符串，并设置 pos 为 1，以获得输入字符串的第一个字符 ❶。随后过程进入**重复执行**，将 strIn 中的字符依次加入 strOut 之后 ❷。每次迭代都会设置 ch 的值为 strIn 的 pos 位置的字符 ❸。如果当前位置 pos 与插入位置 charPos 相等，脚本则将 strAdd 加入到 strOut 之后 ❹。当然，无论是否满足相等的条件，ch 都会加入到 strOut 之后 ❺，最后增加 pos 以准备下一轮迭代时访问 strIn 的下一个字符 ❻。

我们已经制作好 Insert 过程，现在来看看如何使用它吧！见图 8-9。

变量 alpha 保存了所有的小写字母，程序将随机选取一个字母插入到正确的单词中，从而形成错误的拼写 ❶。脚本先从事先准备好的单词表中随机抽取一个正确的单词保存到变量 inWord 中 ❷，该单词表的相关知识将在下一章学习，现在只需简单地将其想象为词库，从中可以获得正确的单词。再从 alpha 中随机选取一个字母保存到 randChar❸，最后从 inWord 中随机选取一个插入位置保存到 randPos 中 ❹。随后脚本调用了我们之前编写的过程 Insert 生成错误的单词（strOut）❺。调用完毕后，脚本进入循环要求玩家输入答案 ❻。每次迭代时，脚本都会询问用户的输入 ❼，最后使用**如果…那么**判断正确与否 ❽。如果玩家的回答与原单词 inWord 相等，游戏结束；否则玩家需要再次尝试。