我们从 strIn 中随机选取一个字符置于临时字符串 str1 之后。（本例的临时字符串的作用为暂时存储乱序的加密单词），然后将其从 strIn 中移除，这样该字符就不会被多次使用。重复上述过程直到 strIn 为空字符串。过程 Randomize 如图 8-11 所示。

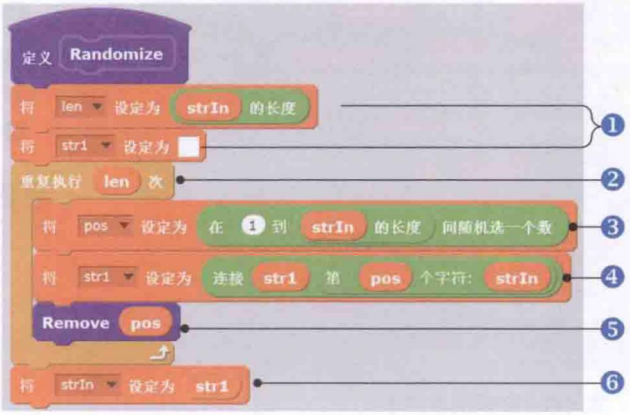

图 8-11：过程 Randomize

过程首先设置变量 len 为输入字符串 strIn 的长度，然后设置临时字符串 str1 为空字符串 ❶。脚本进入**重复执行**生成乱序的加密单词 str1❷，循环次数等于输入字符串的长度。每轮迭代时，程序从 strIn 中随机选取一个位置 ❸，将该位置对应的字符加入到 str1 之后 ❹。注意，在步骤 ❸ 中我们使用了 **strIn 的长度**，这是因为每轮迭代其长度都不相同。随后脚本调用过程 Remove 删除在 strIn 中随机选取的字符 ❺。当循环结束时，脚本将 strIn 的值设定为乱序排列的 str1❻。

过程 Remove 避免了在加密单词时重复使用相同的字母，其脚本如图 8-12 所示。它可以移除字符串 strIn 中 charPos 位置上的字符。

该过程同样使用了临时字符串 str2。脚本首先将 str2 设置为空字符串，设定循环计数器 *n* 等于 1，以获得 strIn 的第一个字符 ❶。过程进入**重复执行**生成输出字符串 ❷。如果我们希望保留当前字符，则将该字符加入 str2 之后 ❸。随后循环计数器 *n* 增加 1，使得下一轮迭代访问 strIn 的下一个字符 ❹。循环结束后，脚本设置 strIn 的值为移除字符的 str2❺。