赋值、输入与输出

Ref Pseudocode

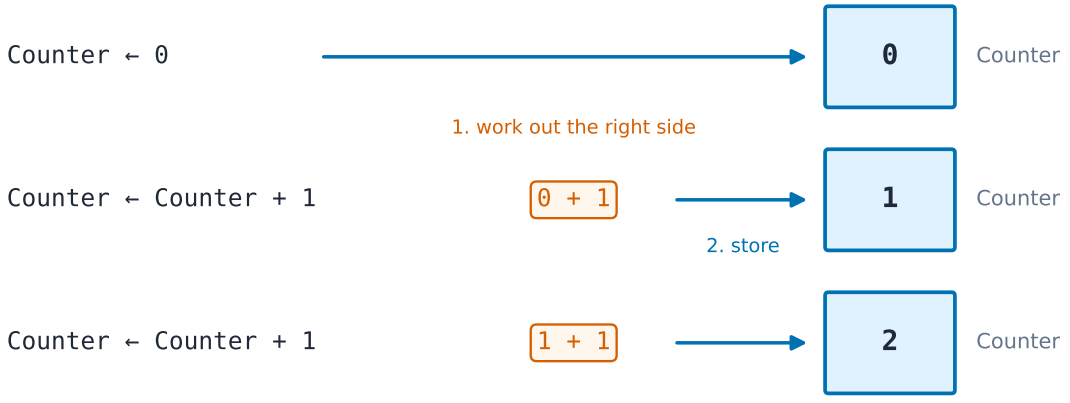
The assignment arrow

赋值赋值运算符是 ←，即左向箭头。它表示取右侧的值放入左侧的框中。右侧先计算，然后存入框中。

```
<identifier> ← <value>
```

```
DECLARE Counter : INTEGER
Counter ← 0
Counter ← Counter + 1
Counter ← Counter + 1
OUTPUT Counter
```

`Counter ← Counter + 1` 作为数学式子看似不可能，但作为指令却非常自然：用当前框中的值计算右侧结果，再将结果存回框中。



右侧先计算，然后存入框中

- 如果你无法输入 ←，请写 `<-`。两者均可接受，此处及考试中均有效。
- 左侧必须是一个**单一标识符**或一个数组元素。绝不能是计算表达式。

常见错误

- `Counter = 0`。在伪代码中，`=` 表示**比较**而非赋值 — 容易混淆。将二者混淆是本论文中最常见的语法错误。
- `Counter + 1 ← Counter`。箭头指向被修改的对象。

INPUT and OUTPUT

INPUT 从用户处读取值存入变量。OUTPUT 显示一个或多个值。

```
DECLARE Name : STRING
DECLARE Age : INTEGER
INPUT Name
INPUT Age
OUTPUT "Hello ", Name
OUTPUT "Next year you will be ", Age + 1
```

- INPUT 恰好需要**一个**变量，且该变量必须已被声明。
- OUTPUT 可接受任意数量的项目，由**逗号**分隔；它们依次打印彼此之间无任何间隔。
- 字面量字符串 literal string 在 OUTPUT 中使用双引号；变量则不需要。

■ 提示用户

单独的 INPUT 在屏幕上不会显示任何内容，因此良好的算法应先输出提示信息：

```
DECLARE Radius : REAL
OUTPUT "Enter the radius: "
INPUT Radius
OUTPUT "The diameter is ", Radius * 2
```



提示先出现，然后是输入的文本，最后是回复

常见错误

- INPUT "Enter a number", X— 提示本身是一个独立的 OUTPUT 语句。
- OUTPUT "Total: " + Total。0478 用逗号连接输出项；+ 属于算术运算，而 & 归属于 A-Level 9618（主题 11）。
- 忘记加引号时：OUTPUT Hello 会将其视为名为 Hello 的变量来查找。