输入、输出与运算符

Ref As Pseudocode

INPUT 与 OUTPUT

INPUT

INPUT 从用户那里读取一个值，并把它存进一个变量。

```
INPUT <identifier>
```

标识符必须是一个变量，也可以是数组的一个元素或记录的一个字段。

```
DECLARE Age : INTEGER
OUTPUT "How old are you?"
INPUT Age
OUTPUT "Next year you will be ", Age + 1
```

点击 “ 运行 ” 时，页面会先让你输入数据。

- 先显示一条**提示信息**（prompt）：用一条 OUTPUT 告诉用户该输入什么。评分标准常常为此给分。
- 一条 INPUT 只读取**一个**变量。要读两个值，就写两行 INPUT。

OUTPUT

OUTPUT 在屏幕上显示一个或多个值。多个值之间用**逗号**分隔。

```
OUTPUT <value(s)>
```

```
DECLARE Lives : INTEGER
DECLARE Score : INTEGER
Lives ← 3
Score ← 1250
OUTPUT Score
OUTPUT "You have ", Lives, " lives left"
```

这些值在同一行上一个接一个地输出，中间不会自动加任何东西。所以 "You have " 的末尾留了一个空格。

常见错误

- INPUT "Enter age", Age。INPUT 不会显示消息，应该先写一行 OUTPUT。
- OUTPUT "Total is Total"。引号里的单词只是普通文字。要把变量写在引号外面：OUTPUT "Total is ", Total。
- 用 PRINT 代替 OUTPUT。PRINT 是 Python 的写法，不是剑桥伪代码。

算术运算、DIV 与 MOD

■ 算术运算符

运算符	含义	示例	结果
+	加	7 + 2	9
-	减	7 - 2	5
*	乘	7 * 2	14
/	除	7 / 2	3.5
DIV	整数除法	7 DIV 2	3
MOD	除法后的余数	7 MOD 2	1

- / 的结果总是 REAL，即使两个数都是整数：10 / 2 等于 5.0。
- 和数学一样，先算 * 和 /，再算 + 和 -。用**括号**让运算顺序一目了然：(A + B) / 2。

```
DECLARE A : INTEGER
DECLARE B : INTEGER
A ← 10
B ← 4
OUTPUT A + B * 2
OUTPUT (A + B) * 2
OUTPUT A / B
```

输出依次是 18、28 和 2.5。

■ DIV 与 MOD

DIV 给出**商**（quotient）：一个数里面能完整地包含另一个数多少次。MOD 给出**余数**（remainder）：剩下的部分。在 9618 中，这两个运算符都写在两个数**中间**，就像 + 一样。

```
DECLARE Minutes : INTEGER
Minutes ← 135
OUTPUT Minutes DIV 60, " hours and ", Minutes MOD 60, " minutes"
```

135 分钟就是 2 小时（$2 \times 60 = 120$），还剩 15 分钟。

下面三种用法你会反复遇到：

- **判断奇偶**：偶数的 Number MOD 2 = 0 为 TRUE。
- **取最后一位数字**：对正整数，Number MOD 10 给出它的个位数。
- **去掉最后一位数字**：Number DIV 10。所以 472 DIV 10 等于 47。

```
DECLARE Number : INTEGER
DECLARE DigitSum : INTEGER
Number ← 472
DigitSum ← 0
WHILE Number > 0
   DigitSum ← DigitSum + (Number MOD 10)
   Number ← Number DIV 10
ENDWHILE
OUTPUT DigitSum
```

这段程序把各位数字相加：$4 + 7 + 2$，输出 13。

常见错误

- DIV(7, 2) 或 MOD(7, 2)。这是 **IGCSE** 的写法。在 9618 中要写 7 DIV 2。
- 需要整数时却用了 /，例如用作数组下标。7 / 2 等于 3.5，不是合法的下标；应该用 7 DIV 2。

关系运算符与逻辑运算符

■ 关系运算符

关系运算符（relational operator）比较两个值，结果永远是 BOOLEAN：TRUE 或 FALSE。

运算符	含义
=	等于
<>	不等于
<	小于
>	大于
<=	小于或等于
>=	大于或等于

```
DECLARE Mark : INTEGER
Mark ← 65
OUTPUT Mark >= 50
OUTPUT Mark = 100
OUTPUT Mark <> 0
```

输出依次是 TRUE、FALSE 和 TRUE。

■ 逻辑运算符

逻辑运算符（logic operators）AND、OR 和 NOT 用来连接或改变布尔值。

表达式	何时为 TRUE
A AND B	A 和 B **都**为 TRUE
A OR B	A 和 B **至少一个**为 TRUE
NOT A	A 为 FALSE

判断一个范围要用 AND 连接**两个**完整的比较：

```
DECLARE Age : INTEGER
DECLARE HasTicket : BOOLEAN
Age ← 15
HasTicket ← FALSE
OUTPUT Age >= 13 AND Age <= 19
OUTPUT Age < 12 OR HasTicket
OUTPUT NOT HasTicket
```

输出依次是 TRUE、FALSE 和 TRUE。

AND 和 OR 混在一起时，要加**括号**让意思清楚：`(Day = "Sat" OR Day = "Sun") AND NOT Raining`。

■ 练习

1. 29 DIV 4 等于多少？29 MOD 4 等于多少？

2. 写出一个表达式，当 Temp 在 18 到 25 这个范围**之外**时为 TRUE。

<details> <summary> 参考答案 </summary>

1. 29 DIV 4 等于 7，29 MOD 4 等于 1，因为 $4 \times 7 = 28$，还剩 1。
2. Temp < 18 OR Temp > 25。（也可以写 NOT (Temp >= 18 AND Temp <= 25)。）

```
DECLARE Temp : INTEGER
Temp ← 27
OUTPUT 29 DIV 4, " ", 29 MOD 4
OUTPUT Temp < 18 OR Temp > 25
```

</details>

常见错误

- IF 13 <= Age <= 19。每个比较都要有自己的变量：Age >= 13 AND Age <= 19。
- Age >= 13 AND <= 19。后半部分缺了左边的值，要再写一次 Age。
- 用 != 或 == 表示” 不等于” 和” 等于”。那是 Python 和 C 的写法；伪代码用 <> 和 =。