字符串与内置函数

Ref As Pseudocode

用 & 连接字符串

连接（concatenate）是指把字符串首尾相接。在 9618 中，连接运算符是 `&`。

```
DECLARE FirstName : STRING
DECLARE LastName : STRING
DECLARE FullName : STRING
FirstName ← "Summer"
LastName ← "Pudding"
FullName ← FirstName & " " & LastName
OUTPUT FullName
OUTPUT LENGTH(FullName)
```

- `&` 生成**一个新的字符串**。你可以把它存起来、传给函数，或者输出它。
- 各部分之间不会自动加任何东西。要空格，就自己连接一个 `" "`。
- `&` 用来连接字符串和字符。要连接数字，先用 `NUM_TO_STR` 把它变成字符串（见 7.4 节）。

■ 在循环中拼出一个字符串

从空字符串开始，每一遍往后面加一点：

```
DECLARE Stars : STRING
DECLARE i : INTEGER
Stars ← ""
FOR i ← 1 TO 5
   Stars ← Stars & "*"
NEXT i
OUTPUT Stars
```

■ 用 & 还是逗号?

- 在 `OUTPUT` 中，逗号把各个值依次输出：`OUTPUT "Age: ", Age`。这些值可以是任何类型。
- `&` 生成一个新的**字符串值**，例如用来存进变量或写入文件。

常见错误

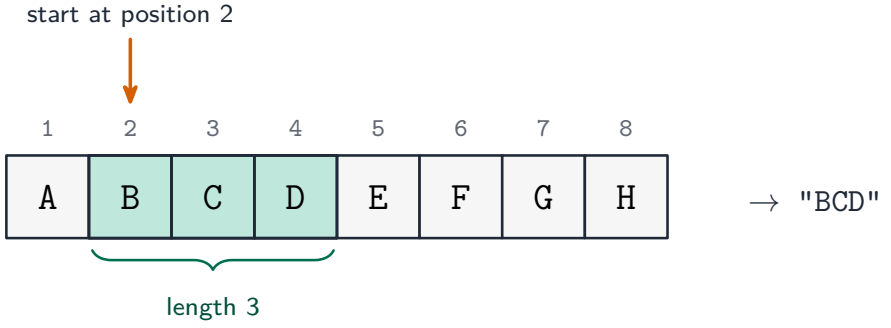
- `"Total: " & 25`。25 是数字；应写 `"Total: " & NUM_TO_STR(25)`。
- 像 Python 那样用 `+` 连接字符串。伪代码用 `&`。

LEFT、RIGHT、MID 与 LENGTH

考试中，你可以使用的字符串函数**总会印在随卷附带的插页（insert）里**。但你仍然要知道它们的用法，因为考试时必须又快又准地使用它们。字符串中的位置从 **1** 开始。

函数	返回	示例	结果
`LENGTH(S)`	S 中的字符个数	`LENGTH("Happy Days")`	`10`
`LEFT(S, x)`	最左边的 x 个字符	`LEFT("ABCDEFGH", 3)`	`"ABC"`
`RIGHT(S, x)`	最右边的 x 个字符	`RIGHT("ABCDEFGH", 3)`	`"FGH"`
`MID(S, x, y)`	从第 x 个位置开始的 y 个字符	`MID("ABCDEFGH", 2, 3)`	`"BCD"`

- `LENGTH` 也会计算空格和标点。
- `MID` 的第三个值是**长度**，不是结束位置。`MID("ABCDEFGH", 2, 3)` 从 B 开始，取 3 个字符。
- **子串**（substring）是字符串的一部分。`LEFT`、`RIGHT` 和 `MID` 返回的都是子串。



MID(“ABCDEFGH”, 2, 3)：从第 *2* 个位置开始，取 *3* 个字符

exercises.lol exercises.com.cn

■ 逐个访问字符

MID(S, i, 1) 就是第 i 个位置上的字符。用一个从 1 到 LENGTH(S) 的循环，就能访问所有字符。几乎每道字符串题都会用到这种写法。

```
DECLARE Word : STRING
DECLARE i : INTEGER
DECLARE Vowels : INTEGER
DECLARE Letter : CHAR
Word ← "EDUCATION"
Vowels ← 0
FOR i ← 1 TO LENGTH(Word)
   Letter ← MID(Word, i, 1)
   IF Letter = 'A' OR Letter = 'E' OR Letter = 'I' OR Letter = 'O' OR
   ↪  Letter = 'U' THEN
      Vowels ← Vowels + 1
   ENDIF
NEXT i
OUTPUT Word, " has ", Vowels, " vowels"
```

■ 把字符串倒过来

```
DECLARE Word : STRING
DECLARE Reversed : STRING
DECLARE i : INTEGER
Word ← "STRESSED"
Reversed ← ""
FOR i ← LENGTH(Word) TO 1 STEP -1
   Reversed ← Reversed & MID(Word, i, 1)
NEXT i
OUTPUT Reversed
```

常见错误

- 从位置 0 开始。第一个字符在位置 **1**。
- 把 MID(S, 2, 5) 理解为” 从 2 到 5”。它的意思是” 从位置 2 开始取 5 个字符”。
- 写 SUBSTRING(S, 2, 3)。那是 IGCSE 的函数名；9618 中的函数是 MID。
- 要取的字符比实际的多：LEFT("Hi", 5) 会出错。

大小写与字符编码

■ 大写与小写

函数	返回	示例	结果
TO_UPPER(x)	把 x 中的字母全部变成大写；x 可以是 STRING 或 CHAR	TO_UPPER("Hello")	"HELLO"
TO_LOWER(x)	把 x 中的字母全部变成小写	TO_LOWER("Hello")	"hello"
UCASE(c)	把一个 CHAR 变成大写	UCASE('h')	'H'
LCASE(c)	把一个 CHAR 变成小写	LCASE('W')	'w'

数字、空格等非字母字符不会改变。Paper 2 插页里印的是 TO_UPPER 和 TO_LOWER，它们可以处理整个字符串。

一个常见用法：**不区分大小写**地比较两个字符串。

```
DECLARE Answer : STRING
Answer ← "yEs"
IF TO_UPPER(Answer) = "YES" THEN
   OUTPUT "You said yes"
ENDIF
```

■ 字符编码：ASC 与 CHR

每个字符都以一个数字的形式存储，这个数字就是它的**字符编码**（character code）。在 ASCII 中，'A' 是 65，'B' 是 66，……'Z' 是 90，'a' 是 97。

函数	返回	示例	结果
ASC(c)	字符 c 的编码	ASC('A')	65
CHR(x)	编码为 x 的字符	CHR(66)	'B'

```
DECLARE Letter : CHAR
DECLARE Code : INTEGER
Letter ← 'C'
Code ← ASC(Letter)
OUTPUT Letter, " has code ", Code
OUTPUT "The next letter is ", CHR(Code + 1)
```

有了编码，下面两件事就很容易：

- **是不是大写字母？** Letter >= 'A' AND Letter <= 'Z' 能这样判断，是因为编码是按顺序排列的。
- **字母移位**（如凯撒密码）：CHR(ASC(Letter) + 3)。

常见错误

- 在 9618 答案中对整个字符串用 UCASE("hello")。UCASE 只处理一个字符；字符串要用 TO_UPPER。

- 忘记 'a' 和 'A' 的编码不同，所以 'a' = 'A' 为 FALSE。

数字与字符串的转换

■ 互相转换

函数	返回	示例	结果
NUM_TO_STR(x)	把数字 x 变成字符串	NUM_TO_STR(87.5)	"87.5"
STR_TO_NUM(x)	把字符串 x 变成数字	STR_TO_NUM("23")	23
IS_NUM(S)	如果 S 是合法的数字，返回 TRUE	IS_NUM("12a")	FALSE

从文件读入或由用户输入的数据，往往是字符串。转换之前先用 IS_NUM 检查，否则程序可能出错。

```
DECLARE Entry : STRING
DECLARE Value : INTEGER
Entry ← "42"
IF IS_NUM(Entry) THEN
   Value ← STR_TO_NUM(Entry)
   OUTPUT "Double: ", Value * 2
ELSE
   OUTPUT "Not a number"
ENDIF
OUTPUT "Code " & NUM_TO_STR(7) & "B"
```

■ INT 与 RAND

函数	返回	示例	结果
INT(x)	x 的整数部分；小数部分被**直接去掉**	INT(27.9)	27
RAND(x)	一个从 0 到 x 的随机 REAL，**不包括** x	RAND(87)	例如 35.43

INT **不是**四舍五入：INT(27.9) 是 27，不是 28。要四舍五入到整数，先加 0.5：INT(X + 0.5)。

要得到一个 1 到 6 之间的随机整数，就像掷骰子：

```
DECLARE Dice : INTEGER
Dice ← INT(RAND(6)) + 1
OUTPUT "You rolled ", Dice
```

RAND(6) 给出一个从 0 到 6（不包括 6）的值。INT 把它变成 0 到 5，再 + 1 就得到 1 到 6。

■ 日期

DATE 类型的值在插页中有自己的函数，例如 DAY(ThisDate)、MONTH(ThisDate)、YEAR(ThisDate) 和 SETDATE(Day, Month, Year)。具体有哪些，请看你那份试卷的插页，上面对每个函数都有说明。

■ 练习

1. 像 "AB1234" 这样的产品编号由两个字母和四个数字组成。写出伪代码：如果最后四个字符是数字，就输出 "Valid"，否则输出 "Invalid"。

2. 输出 "Grace Brewster Hopper" 的首字母缩写，也就是每个单词的第一个字母。

<details> <summary> 参考答案 </summary>

```
DECLARE Code : STRING
Code ← "AB1234"
IF LENGTH(Code) = 6 AND IS_NUM(RIGHT(Code, 4)) THEN
   OUTPUT "Valid"
ELSE
   OUTPUT "Invalid"
ENDIF
```

```
DECLARE FullName : STRING
DECLARE Initials : STRING
DECLARE i : INTEGER
FullName ← "Grace Brewster Hopper"
Initials ← LEFT(FullName, 1)
FOR i ← 2 TO LENGTH(FullName)
   IF MID(FullName, i - 1, 1) = " " THEN
      Initials ← Initials & MID(FullName, i, 1)
   ENDIF
NEXT i
OUTPUT Initials
```

输出 GBH。如果一个字母**前面**的字符是空格，它就是一个新单词的开头。

</details>

常见错误

- 对字符串做算术："5" + 1。要先用 STR_TO_NUM 转换。
- 以为 INT 会四舍五入。它总是直接去掉小数部分。
- 把 RAND(6) 当骰子用。它会给出 3.81 这样的小数，还可能给出 0；应该用 INT(RAND(6)) + 1。