

选择结构

Ref Pseudocode

IF、THEN、ELSE、ENDIF

选择 (selection) 根据一个条件 (condition) 运行这一组语句或那一组。

```
IF <condition>
  THEN
    <statements>
ENDIF
```

```
DECLARE Mark : INTEGER
Mark ← 72
IF Mark >= 50
  THEN
    OUTPUT "Pass"
ENDIF
```

加上 ELSE 处理另一种情况。两个分支 (branch) 中恰好运行一个 —— 绝不会两个都运行, 也绝不会一个都不运行。

```
DECLARE Mark : INTEGER
Mark ← 41
IF Mark >= 50
  THEN
    OUTPUT "Pass"
  ELSE
    OUTPUT "Fail"
ENDIF
```

排版是学生最常丢分的地方, 所以把它当作一个图形记下来:

- IF 和它的条件写在一行, **条件后面什么都不写**。
- THEN 和 ELSE 各占一行, 缩进**两个**空格。
- 语句缩进**四个**空格。
- ENDIF 回到它自己 IF 所在的列。

嵌套

IF 里面再放 IF, 可以处理超过两种结果。每一个都需要自己的 ENDIF。

```
DECLARE Mark : INTEGER
Mark ← 85
IF Mark >= 80
    THEN
        OUTPUT "Distinction"
    ELSE
        IF Mark >= 50
            THEN
                OUTPUT "Pass"
            ELSE
                OUTPUT "Fail"
            ENDIF
        ENDIF
    ENDIF
```

常见错误

- 忘记 ENDIF。每个 IF 都要闭合, 少一个就是语法错误 (syntax error), 评分者一眼就能看出。
- 把 IF Mark >= 50 THEN OUTPUT "Pass" 写成一行。**两份考纲都没有这种写法; 请写成三行。**
- 把 IF Mark = 50 THEN 写成 IF Mark ← 50。箭头是赋值, = 才是比较。

CASE OF

当一个变量要与若干**单个值**逐一比较时, CASE OF 比一叠嵌套的 IF 清楚得多。

```
CASE OF <identifier>
    <value 1> : <statement>
    <value 2> : <statement>
    OTHERWISE <statement>
ENDCASE
```

```

DECLARE Choice : INTEGER
Choice ← 2
CASE OF Choice
    1 : OUTPUT "You chose north"
    2 : OUTPUT "You chose south"
    3 : OUTPUT "You chose east"
    4 : OUTPUT "You chose west"
    OTHERWISE OUTPUT "That is not a direction"
ENDCASE

```

- 每个分支是一个**值**、一个冒号, 然后是要做的事。
- OTHERWISE 兜住所列值没有覆盖到的一切。它是可选的, 写上是好习惯 —— 用户输入意料之外的东西时, 走的就是它。
- 第一个匹配的分支运行, 其余的跳过。

一个范围, 和一个字符

```

DECLARE Grade : CHAR
Grade ← 'B'
CASE OF Grade
    'A' : OUTPUT "Excellent"
    'B' : OUTPUT "Good"
    'C' : OUTPUT "Satisfactory"
    OTHERWISE OUTPUT "Unclassified"
ENDCASE

```

常见错误

- 用 CASE OF 处理**条件**。它把一个变量与若干值比较; Mark >= 50 需要 IF。
- 忘记 ENDCASE。
- 省略 OTHERWISE, 然后困惑于意料之外的值为什么毫无反应。