

11.2 Constructs

Name: _____ Class: _____ Date: _____ **Total: 40 marks**

KEY WORDS selection 选择 • trace table 追踪表 • constructs 结构 • nested 嵌套

Objective

Build the skills to answer exam questions on **constructs** 结构 — selection (IF, CASE) and the three kinds of loop.

You must be able to:

- write IF...ELSE, **nested** IF, and CASE structures
- write a **count-controlled** (FOR), **pre-condition** (WHILE) and **post-condition** (REPEAT) loop
- **justify** which loop best suits a problem

1 Worked examples

Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

■ Selection: IF and CASE

```
IF Age >= 18 THEN
  OUTPUT "Adult"
ELSE
  OUTPUT "Minor"
ENDIF
```

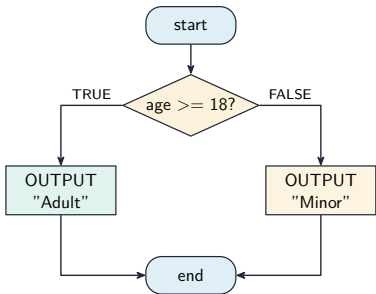
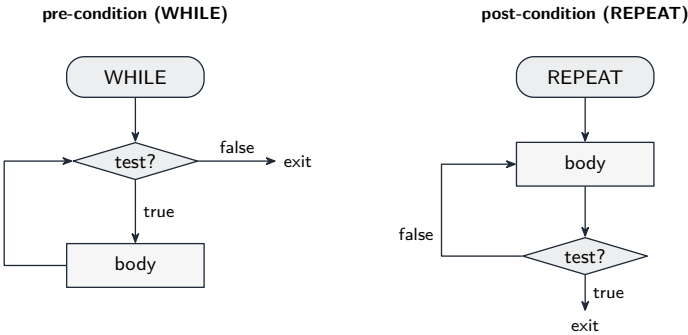
For one value against **several** options, a CASE is cleaner than deep nested IFs:

```
CASE OF Grade
  "A": OUTPUT "Excellent"
  "B": OUTPUT "Good"
  OTHERWISE: OUTPUT "Try again"
ENDCASE
```

■ The three loops

```
FOR i ← 1 TO 10           // count-controlled - known repeats
  OUTPUT i
NEXT i
```

WHILE tests **before** the body (may run **0** times); REPEAT tests **after** (runs **at least once**):



Selection (if-else) is the other core construct

```
WHILE Total < 100 DO      REPEAT
  INPUT n                INPUT Password
  Total ← Total + n      UNTIL Password = "OPEN"
ENDWHILE
```

■ Choosing a loop

- repeats **known** in advance → FOR
- may need **zero** passes → WHILE
- needs **at least one** pass → REPEAT

■ Tracing a loop (trace table)

A **trace table** 追踪表 dry-runs code by hand: one column per variable (plus OUTPUT), one row each time a value changes. Trace this for $N = 3$:

```
Total ← 0
FOR i ← 1 TO N
    Total ← Total + i
NEXT i
OUTPUT Total
```

i	Total	OUTPUT
	0	
1	1	
2	3	
3	6	6

2 Practice

Now apply the methods above.

2.1 Write an IF...ELSE that outputs "Pass" when Mark ≥ 50 , otherwise "Fail". [2]

2.2 Rewrite this nested IF as a CASE structure. [3]

```
IF Grade = "A" THEN
    OUTPUT "Excellent"
ELSE
    IF Grade = "B" THEN
        OUTPUT "Good"
    ELSE
        OUTPUT "Try again"
    ENDIF
ENDIF
```

2.3 State which loop may run **zero** times and which runs **at least once**. [2]

2.4 Write a FOR loop that outputs the **even** numbers from 2 to 20. [2]

3 Exam-style questions

3.1 A menu uses a number Choice. Write a CASE structure: $1 \rightarrow \text{"New"}$, $2 \rightarrow \text{"Open"}$, $3 \rightarrow \text{"Save"}$, otherwise $\rightarrow \text{"Invalid"}$. [4]

3.2 A program must keep asking for a password until the user types "OPEN". Write the loop using the **most suitable** construct, and **justify** your choice. [3]

3.3 Write a **pre-condition** loop that reads numbers and adds them to a **Total**, stopping once **Total** reaches 100. [3]

3.4 Complete the **trace table** for this algorithm when the input **X** is 5. [4]

```
A ← 1
B ← 1
WHILE A < X DO
    B ← B * 2
    A ← A + 1
ENDWHILE
OUTPUT B
```

A	B	OUTPUT
1	1	

3.5 A program keeps the time of day in three global integer variables **HH**, **MM** and **SS**.

(a) A procedure **Tick()** is called once every second and must advance the time correctly, rolling over at 60 seconds, 60 minutes and 24 hours. **Write** pseudocode for **Tick()**. [6]

(b) **Complete** the trace table by dry running your procedure from the starting time shown, for **three** calls. [3]

call	HH	MM	SS
start	10	59	58
1	_____	_____	_____
2	_____	_____	_____
3	_____	_____	_____

(c) **Explain** why the three IF statements must be **nested** or tested in the right order, giving what would go wrong otherwise. [3]

(d) The variables are **global**. **Explain** one drawback of that, and **state** how you would restructure `Tick()` to avoid it. [3]

(e) **State** which loop construct you would use to run the clock until the user stops it, and **justify** your choice. [2]
