

11.2.1 CASE structures, and choosing between IF and CASE

Name: _____ Class: _____ Date: _____

Total: 39 marks

KEY WORDS selection • CASE structure CASE • nested • pseudocode
• variable

Objective

Learn to write selection with the **CASE structure CASE**, which chooses between many paths by looking at the value of **one** variable. You also learn to change a nested IF into a **CASE** and back, and to say when a **CASE** cannot be used. Sheet 11.2 taught the IF statement; the loops start on sheet 11.2.2.

You must be able to:

- write a **CASE** structure with single values, with a range of values, and with an **OTHERWISE** clause
- follow a **CASE** structure for a given value
- rewrite a nested IF as a **CASE** structure, and a **CASE** structure as a nested IF
- explain when a **CASE** structure is the better choice, and when it cannot be used

1 Worked examples

Study these first. Each one shows a **method** that a question later on this sheet needs.

■ The CASE structure

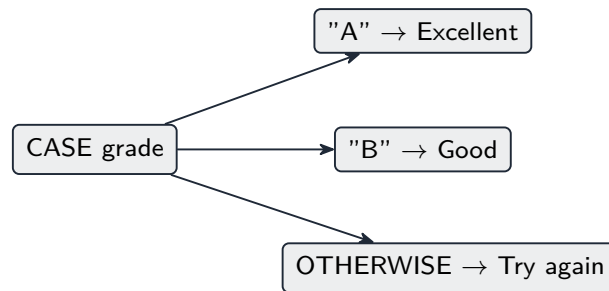
A **CASE** structure tests **one variable** against a list of values, and runs the statement for the value that matches.

```
CASE OF Direction
  'N' : Y ← Y + 1
  'S' : Y ← Y - 1
  'E' : X ← X + 1
  'W' : X ← X - 1
  OTHERWISE : OUTPUT "Not a direction"
ENDCASE
```

- The first line names the variable: **CASE OF Direction**. It ends with **ENDCASE**.
- Each line is a **value**, a **colon**, and the statement to run. A **CHAR** value is in single

quotes, a **STRING** value in double quotes, and a number has no quotes.

- The values are tested **in order**. When one matches, its statement runs, and the structure **ends**: the other values are not tested.
- **OTHERWISE** runs when **no** value matches. It must be the last line. Without it, a value that matches nothing does nothing.



■ A range of values

One line can cover a range, written with **TO**:

```
CASE OF Mark
  70 TO 100 : Grade ← "Merit"
  50 TO 69  : Grade ← "Pass"
  0 TO 49   : Grade ← "Fail"
  OTHERWISE : OUTPUT "The mark is not valid"
ENDCASE
```

■ When CASE cannot be used

A **CASE** looks at **one variable** only. It **cannot** replace an **IF** whose conditions:

- test **different variables** (IF Age > 17 AND HasLicence = TRUE), or
- **compare two variables** with each other (IF Score > HighScore).

For these you need **IF**.

■ From a nested IF to a CASE

A nested IF can become a CASE when **every** condition tests the **same variable** for equality (or a range).

1. The variable that is tested goes on the first line: CASE OF Colour.
2. Each = value becomes one line.
3. The last ELSE becomes OTHERWISE.

```
CASE OF Colour
    "red"    : OUTPUT "Stop"
    "amber"  : OUTPUT "Get ready"
    "green"  : OUTPUT "Go"
    OTHERWISE : OUTPUT "Signal fault"
ENDCASE
```

The CASE version is **shorter, easier to read, and easier to change**: a new colour is one new line, not one more level of nesting. It replaces this nested IF:

```
IF Colour = "red" THEN
    OUTPUT "Stop"
ELSE
    IF Colour = "amber" THEN
        OUTPUT "Get ready"
    ELSE
        IF Colour = "green" THEN
            OUTPUT "Go"
        ELSE
            OUTPUT "Signal fault"
        ENDIF
    ENDIF
ENDIF
```

2 Practice

Now use the methods above. The first questions need one idea each.

2.1 Study this pseudocode.

```
CASE OF Day
  1 : OUTPUT "Monday"
  2 : OUTPUT "Tuesday"
  6 TO 7 : OUTPUT "Weekend"
  OTHERWISE : OUTPUT "Midweek"
ENDCASE
```

M	T	W	T	F	S	S	

State what is output when:

- (a) Day is 2 [1]
- (b) Day is 7 [1]
- (c) Day is 4 [1]

2.2 State the purpose of the OTHERWISE clause in a CASE structure. [1]

OTHERWISE

2.3 A shop sells drinks in three sizes. The CHAR variable **Size** holds 'S', 'M' or 'L'. A small drink costs 1.50, a medium drink 2.00 and a large drink 2.50. Complete the CASE structure so that it stores the correct price. [3]

```
CASE OF Size
  'S' : Price ← .....
  ..... : Price ← 2.00
  'L' : Price ← 2.50
  ..... : OUTPUT "No such size"
ENDCASE
```

S	M	L
1.50	2.00	2.50

2.4 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
```

A

2.5 State whether each IF statement could be rewritten as a CASE structure. Give a reason for each answer.

[4]

(a) IF Size = 'S' THEN ... ELSE IF Size = 'M' THEN ... ELSE IF Size = 'L' THEN ...

IF

CASE

(b) IF Height > 1.2 AND Age >= 10 THEN ...

3 Exam-style questions

3.1 A menu uses a number Choice. Write a CASE structure: 1→"New", 2→"Open", 3→"Save", otherwise→"Invalid".

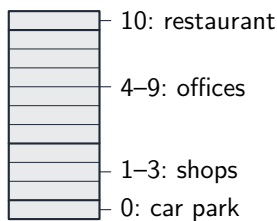
[4]

1 New
2 Open
3 Save

3.2 A lift (elevator) program stores the floor number in the integer variable **Floor**. The building has these floors:

- floor 0 is the car park
- floors 1 to 3 are shops
- floors 4 to 9 are offices
- floor 10 is the restaurant.

Write a **CASE** structure that outputs the use of the floor, and outputs "No such floor" for any other number. [5]



3.3 Study this **CASE** structure.

```
CASE OF Choice
    'A' : Total ← Total + 5
    'B' : Total ← Total + 10
    OTHERWISE : Total ← 0
ENDCASE
```

Choice
Total

(a) Rewrite it as nested **IF** statements. [4]

(b) Give **two** reasons why a programmer may prefer the **CASE** version. [2]

3.4 A calculator program inputs two real numbers, **First** and **Second**, and a character **Operator**, which should be '+', '-', '*' or '/'. It outputs the result of the calculation. If the operator is not one of the four, it outputs "Unknown operator".

(a) Write pseudocode for the program, using a **CASE** structure. Declare your variables.[7]



(b) The program must not divide by zero. **Explain** why this check cannot be written as one more line of the **CASE** structure, and state how it should be written. [3]