MODULE O ROBUST PROGRAMS · O6

Finding and correcting errors in programs

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 7.8 (with 7.6, 7.7 and 4.2.3)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. Write any corrections using pseudocode. The total for this worksheet is **21 marks**.

1

Total: 4 marks

A programmer writes a program in a high-level language. The program contains **two** errors that break the rules of the language: one on line 8 and one on line 20.

(a) Tick (✓) **one** box in each row to show whether each statement describes how a **compiler** or an **interpreter** would report these errors. **[4]**

Statement	Compiler	Interpreter
The code on lines 1 to 7 is run before any error is reported.		
One error report lists both errors, for the whole code.		
The error on line 20 is only reported after the error on line 8 has been corrected.		
None of the program can be run until both errors are corrected.		

2

Total: 4 marks

Each pseudocode statement in the table contains **one** error. For the string, the first character is in position 1.

(a) Write a corrected pseudocode statement for each one. **[4]**

What the statement should do	Statement with an error	Corrected statement
Add 1 to the value in `Count`	`Count + 1 ← Count`	
Repeat once for each of the 24 hours in a day	`FOR Hour ← 1 TO 23`	
Check that `Length` is from 2 to 5 inclusive	`IF Length >= 2 AND <= 5`	
Store the first 3 characters of `Code` in `Prefix`	`Prefix ← SUBSTRING(Code, 3, 1)`	

3 **Total: 7 marks**

A ferry company weighs the vehicles before they board. This pseudocode algorithm should:
- input the weights of 20 vehicles, in whole kilograms
- only accept a weight from 500 to 4000 inclusive, and ask again until a weight in this range is input
- count the vehicles that weigh more than 3000 kg
- output this count.

```
01 DECLARE Vehicle : INTEGER
02 DECLARE Weight : INTEGER
03 DECLARE Heavy : INTEGER
04 Heavy ← 0
05 FOR Vehicle ← 1 TO 20
06     REPEAT
07         OUTPUT "Enter the vehicle weight in kg (500 to 4000)"
08         INPUT Weight
09     UNTIL Weight >= 500 OR Weight <= 4000
10     IF Weight > 3000
11       THEN
12         Heavy ← Heavy + 1
13     ENDWHILE
14 NEXT Weight
15 OUTPUT "Vehicles over 3000 kg: ", Weight
```

(a) Identify the line numbers of the **four** errors in the pseudocode and suggest a correction for each error. **[4]**

Line number	Correction

(b) Before it is corrected, the error on line 09 lets the algorithm accept weights that should be rejected. **[3]**

Suggest **one** weight that could be used as test data to show this error. State the type of test data and the expected result.

Weight	
Type of test data	
Expected result	

4 **Total: 6 marks**

A weather station stores the rainfall, in whole millimetres, for five days in the array `Rain`. This algorithm should count the number of days with **less than 5** mm of rain.

Rain[1]	Rain[2]	Rain[3]	Rain[4]	Rain[5]
3	8	4	15	2

```
01 Count ← 0
02 FOR Index ← 1 TO 5
03     IF Rain[Index] < 5
04       THEN
05         Count ← Count + Rain[Index]
06     ENDIF
07 NEXT Index
08 OUTPUT "Low rainfall days: ", Count
```

(a) Complete the trace table for the algorithm. You may not need to use all the rows. **[3]**

Index	Count	OUTPUT

(b) State the output the weather station expected from this data. **[1]**

(c) Identify the line number of the error and write a corrected pseudocode statement. **[2]**

Total for this worksheet: 21 marks