MODULE K ALGORITHM DESIGN · K8

Trace tables and dry-runs

Original practice questions · 21 marks · ~25 minutes · Cambridge 0478 syllabus 7.7 (with 7.3, 7.8)

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. All algorithms are written in Cambridge pseudocode or as flowcharts. In each trace table, start a new row each time a value changes, and record every output, including user prompts, in the OUTPUT column. You may not need to use all the rows. The marks for each part are shown in brackets [ ]. The total for this worksheet is **21 marks**.

1

Total: 2 marks

(a) Tick (✓) **one** box to show which statement describes a dry-run of an algorithm. **[1]**

Statement	Tick (✓)
Running the program on a computer to see if it crashes	
Working through the algorithm by hand, one step at a time, recording what happens	
Checking that data has been copied correctly from its original source	
Rewriting the algorithm in a high-level programming language	

(b) A trace table records the values of the variables at each step of an algorithm. Give **one** other thing that is recorded in a trace table at each step. **[1]**

2

Total: 4 marks

A box office takes group bookings. The number of seats in each booking is entered. Bookings of more than 20 seats are refused. Booking stops when 50 or more seats have been booked.

```
01 Total ← 0
02 Count ← 0
03 REPEAT
04     OUTPUT "Enter seats"
05     INPUT Seats
06     IF Seats > 20
07       THEN
08         OUTPUT "Too many"
09       ELSE
10         Total ← Total + Seats
11         Count ← Count + 1
12     ENDIF
13 UNTIL Total >= 50
14 OUTPUT "Bookings accepted: ", Count
```

(a) Complete the trace table for the algorithm using this input data: **18, 25, 15, 20, 12** **[4]**
You may not need to use all the input data.

Total	Count	Seats	OUTPUT
0	0		

3

Total: 6 marks

A water company stores six river-level readings, in cm, in the array `Reading`. The array is indexed from 1.

Index	1	2	3	4	5	6
Reading	38	45	31	52	29	40

```
01 Low ← Reading[1]
02 Above ← 0
03 FOR Index ← 1 TO 6
04     IF Reading[Index] < Low
05       THEN
06         Low ← Reading[Index]
07     ENDIF
08     IF Reading[Index] > 40
09       THEN
10         Above ← Above + 1
11     ENDIF
12 NEXT Index
13 OUTPUT "Lowest: ", Low
14 OUTPUT "Above limit: ", Above
```

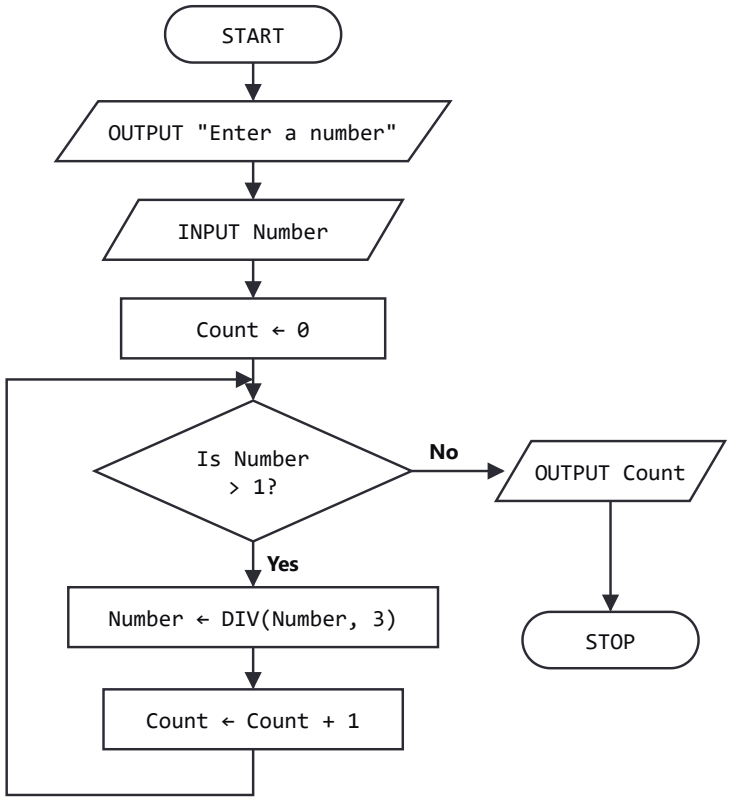
(a) Complete the trace table for the algorithm. The first row has been done for you. **[4]**

Index	Low	Above	OUTPUT
	38	0	

(b) State the purpose of this algorithm. **[2]**

4 **Total: 4 marks**

This flowchart shows an algorithm used by a stock-control system.



(a) Complete the trace table for the algorithm using this input data: **50** **[3]**

Number	Count	OUTPUT

(b) Dry-run the algorithm again using the input data **1**. State the final output. **[1]**

5 **Total: 5 marks**

Four judges at a baking competition each give a score. This algorithm should output the mean (average) of the four scores, but it contains an error.

```
01 FOR Count ← 1 TO 4
02     Total ← 0
03     OUTPUT "Enter score"
04     INPUT Score
05     Total ← Total + Score
06 NEXT Count
07 Average ← Total / 4
08 OUTPUT "Average is ", Average
```

(a) Complete the trace table for the algorithm using this input data: **12, 9, 15, 6** **[3]**

Count	Total	Score	Average	OUTPUT

(b) Identify the line number of the error and suggest a correction so that the algorithm outputs the mean of the four scores. **[2]**

Total for this worksheet: 21 marks