About Debugging

2018 P2Q2

2 The following algorithm calculates the average mark for a group of students.

```
01 FOR Counter ← 1 TO NumberOfStudents
02    Total ← 0
03    INPUT Mark
04    Total ← Total + Mark
05 ENDFOR
06 Average ← Total / NumberOfStudents
07 OUTPUT Average
```

(a) There is an error in this algorithm causing an incorrect result. Describe the error and explain the change required to correct this error. [3]

(b) State the name of this type of error. [1]

(c) The lowest mark in the exam is 0 and the highest is 100. Give an example from each of the appropriate test cases which could be used to test the algorithm. [4]

(d) Name and describe a suitable technique that could be used to manually identify errors in the algorithm. [2]

2   The following algorithm calculates the average mark for a group of students.

```
01  FOR Counter ← 1 TO NumberOfStudents
02      Total ← 0
03      INPUT Mark
04      Total ← Total + Mark
05  ENDFOR
06  Average ← Total / NumberOfStudents
07  OUTPUT Average
```

(a)   There is an error in this algorithm causing an incorrect result. Describe the error and explain the change required to correct this error.   [3]

Cambridge Markers' Report:

(a) Almost all candidates were able to describe and explain the changes required to correct the error.

Suggested Solution :

At line 2, Total should not be reset to 0. It should be declared before the for loop. So Total is not added up but contains the mark for the last student when the for loop finished execution.

(b) State the name of this type of error. [1]

The lowest mark in the exam is 0 and the highest is 100. Give an example from each of the appropriate test cases which could be used to test the algorithm. [4]

Cambridge Markers' Report:

(c) Successful responses made reference to Normal, Abnormal, Erroneous and Boundary data, and included suitable examples of each.

Suggested Solution :

Normal
INPUT 50 60 70 80 90
Expected OUTPUT 70

Boundary
INPUT 0
Expected OUTPUT 0
INPUT 100 100 100 100
Expected OUTPUT 100
INPUT 0 0 0
Expected OUTPUT 0

Invalid
INPUT
Expected OUTPUT 0

(d) Name and describe a suitable technique that could be used to manually identify errors in the algorithm. [2]

Cambridge Markers' Report:

(d) The majority of candidates had not read the question carefully and missed the word 'manually'. Their responses suggested error checks performed by, for example, using breakpoints, which are not regarded as manual checks.

Suggested Solution :

Use desk-checking as questions stated manually.

- Desk-Checking – manual check of each line with its variable content and conditions. Use a trace table.

- Use of print statement to display Total

Not considered as manual checks:
Use a debugger
Use of "break points"

About Debugging

2017 P2Q5(c)

5 The following grid shows the initial state of a popular puzzle.

	8		9					
						7	8	9
2				(4)	5	6		
		1	2	3				
6								4
				1	9	8		
		4	3	2				8
7	6	5						
					7		1	

The aim of the puzzle is to fill the whole grid so that every row, every column and every 3 × 3 mini-grid contains a number between 1 and 9. No number should be repeated in any row, column or 3 × 3 mini-grid.

A software company is creating an online version of the puzzle. A programmer is asked to create the puzzle software.

(c) During the testing of the puzzle software, several errors are discovered.

Describe **two** debugging techniques that could be used to locate these errors. [4]

Comments:

• Note that there is a difference between testing and debugging.

Cambridge Markers' Report:

• A minority of candidates were able to identify a possible debugging technique. Testing techniques rather than debugging techniques were often given

(c) During the testing of the puzzle software, several errors are discovered.

Describe **two** debugging techniques that could be used to locate these errors. [4]

Suggested Solution:

Technique 1: use of print or output statements to output expected results.

Technique 2: call a step-over or step-into: parsing control into a specific function to do a step by step trace (eg Python Tutor)

Technique 3: use breakpoints to perform a step-by-step trace of the code.