2.11 1D and 2D Array Processing: Writing and Tracing Programs: Test Pack

Total Marks

OCR J277 · Calculators not allowed · about 20 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 A gym class records the number of press-ups completed by 5 students in a 1D array called pressUps, indexed from 0: pressUps = [12, 20, 8, 15, 17]

(a) State the value stored at pressUps[0]. **(1)**

(b) State the index of the element that holds the value 15. **(1)**

(c) State the number of elements in the array pressUps. **(1)**

(Total for Question 1 is 3 marks)

---

2 The pseudocode below is intended to input 5 scores from the keyboard into a 1D array called scores, but one line is missing.

```
scores <- [0, 0, 0, 0, 0]
FOR index <- 0 TO 4
    _____________________
NEXT index
```

(Total for Question 2 is 2 marks)

© Revision Library | revisionlibrary.co.uk

3 The array answers stores a student's answers to a 5 question multiple-choice quiz: answers = ["A", "C", "B", "A", "D"]
Write pseudocode to count how many of the 5 answers are "A", storing the result in a variable called countA.

(Total for Question 3 is 3 marks)

---

4 A tutor records monthly expenses, in pounds, for 3 categories over 4 months in a 2D array called expenses, where the first index is the category (0 to 2) and the second index is the month (0 to 3):
expenses = [[80, 95, 70, 105], [40, 45, 38, 42], [120, 110, 130, 140]]
Write a complete pseudocode algorithm that calculates the average monthly expense for each category (i.e. each row's total divided by 4), and stores the three results in a new 1D array called averages.

(Total for Question 4 is 8 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 2.11 1D and 2D Array Processing: Writing and Tracing Programs: Test Pack

Question 1

(a) 12, cao **B1**

(a) Answer: 12

(b) 3, cao **B1**

(b) Answer: 3

(c) 5, cao **B1**

(c) Answer: 5

Question 2

INPUT used to read a value from the keyboard, oe **M1**

scores[index] <- INPUT("Enter a score: ") or equivalent, correctly assigning the input to the current array element **A1**

Answer: scores[index] <- INPUT("Enter a score: ")

Question 3

countA <- 0 initialised before the loop, and a FOR loop over all 5 indices, 0 TO 4, oe **M1**

correct IF condition, answers[index] = "A", oe **M1**

countA <- countA + 1 correctly placed inside the IF, loop/IF correctly closed **A1**

Answer:

```
countA <- 0
FOR index <- 0 TO 4
    IF answers[index] = "A" THEN
        countA <- countA + 1
    ENDIF
NEXT index
OUTPUT countA
```

Question 4

averages <- [0, 0, 0] (or an empty array later appended to) correctly declared before the loop, oe **M1**

outer FOR loop over the category index, 0 TO 2, oe **M1**

rowTotal (or equivalent) correctly re-initialised to 0 at the start of each category, inside the outer loop, oe **M1**

inner FOR loop over the month index, 0 TO 3, correctly nested inside the outer loop, oe **M1**

rowTotal <- rowTotal + expenses[category][month] correctly placed inside the inner loop **A1**

inner loop correctly closed before the average is calculated, oe **M1**

averages[category] <- rowTotal / 4 correctly calculated and stored after the inner loop, before the outer loop moves to the next category **A1**

outer loop correctly closed and averages output/available after both loops finish, with an overall structure that would run without error **B1**

Answer:

```
averages <- [0, 0, 0]
FOR category <- 0 TO 2
    rowTotal <- 0
    FOR month <- 0 TO 3
        rowTotal <- rowTotal + expenses[category][mon
    NEXT month
    averages[category] <- rowTotal / 4
NEXT category
OUTPUT averages
```


© Revision Library | revisionlibrary.co.uk