# Homework 3: Writing and interpreting algorithms Answers

1. The numbers 15, 73, 29, 66, 35, 11, 43, 21 held in array a[n] of length n are to be sorted using a bubble sort. The array index starts at 1.

 The algorithm is given below.

 n = length(a)

 REPEAT

 flag = FALSE

 FOR count = 1 to n-1

 IF a[count] > a[count + 1] THEN

 temp = a[count]

 a[count] = a[count + 1]

 a[count + 1] = temp

 flag = TRUE

 ENDIF

 ENDFOR

 n = n-1

 UNTIL n = 1 OR flag = FALSE

 (a) Show the state of the array after each pass through the data. [3]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **73** | **29** | **66** | **35** | **11** | **43** | **21** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **29** | **66** | **35** | **11** | **43** | **21** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **29** | **35** | **11** | **43** | **21** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **29** | **11** | **35** | **21** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **11** | **29** | **21** | **35** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | **15** | **21** | **29** | **35** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | **15** | **21** | **29** | **35** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

 (b) What is the purpose of the variable **flag** in the algorithm? [1]

 To indicate when there have been no swaps in a pass, therefore the list is sorted.

 (c) How many times will the REPEAT loop be performed when sorting the following numbers? Explain your answer. [2]

 32, 11, 15, 21, 29, 35, 43, 66, 73

 Twice. Once to swap 32 and move into the correct position. Once to pass through cleanly without making any swaps.

 (d) Name **two** circumstances in which a Bubble sort will execute in about the same time as a much more efficient sort algorithm such as the recursive Merge sort. [2]

 For a very small data set, and if the list is virtually sorted already.

2. A factory uses barcodes which represent 6 digits, with the last digit being a check digit.

 For example, 253217 is a valid barcode.

 The check digit is calculated as follows:

* assign weights of 1, 3, 1, 3, 1 to the first five digits
* multiply each digit by its weight
* add the weighted numbers together
* divide the total by 10
* the remainder is the check digit

 (a) One of the following barcodes is invalid. Which one is invalid, and what should the check digit be?

 (i) 365125 invalid – should be 1

 (ii) 212557 [2]

 (b) Write a pseudocode algorithm which

* inputs 100 barcodes, each input as six individual random digits
* recalculates the check digit for each one and compares it with the one input
* outputs the number of barcodes which were input correctly [5]

(see program **check barcodes.py** – this is given below.)

 subroutine to calculate barcode held as separate digits in array **code**

Assume array index starts at 0

SUB calculateCheckDigit (code):

 x 🡨 code[0] + code[2] + code[4] + (code[1] + code[3])\*3

 x 🡨 x mod 10

 return x

ENDSUB

#main program

totalCorrect 🡨 0

FOR count = 0 to 99

 barcode 🡨 [] #initialise array of barcode digits

 FOR n = 0 to 5

 digit 🡨 random(0,9)

 barcode[n] 🡨 digit

 END FOR

 checkdigit 🡨 calculateCheckDigit(barcode)

 IF checkdigit = barcode[5] THEN

 totalCorrect 🡨 totalCorrect + 1

 OUTPUT(barcode)

 END IF

END FOR

OUTPUT ("Total correct barcodes ",totalCorrect)

 [Total 15 marks]