HANDOUT FOR CHAPTER 12

ARRAYS

Past questions

1 **(a)** Write an algorithm, using pseudocode and a `FOR … TO … NEXT` loop structure, to input 1000 numbers into an array.

.....

.....

.....

.....

.....

.....[2]

(b) Rewrite your algorithm using another loop structure.

.....

.....

.....

.....

.....

.....

.....[4]

2 A program will be written to store information about members of a swimming club. The following membership details will be recorded:

- Name
- Gender
- Status:
  Senior
  Junior
- Fee
- Team member (Yes or No)

(i) Choose a suitable data type for each of the membership details to be recorded.

Membership details	Data type
Name	
Gender	
Status	
Fee	
Team member	

[5]

(ii) The swimming club has 50 members.

State the data structure that would be most suitable to use and give a reason for your choice.

Data structure .....

Reason .....

.....[2]

3 A programmer writes a program to store a patient's temperature every hour for a day.

State the data structure that would be most suitable to use and give the reason for your choice.

Data structure .....

Reason.....

.....[2]

4 **(a)** Describe the purpose of each statement in this algorithm.

```
FOR I ← 1 TO 300
  INPUT Name[I]
NEXT I
```

.....

.....

.....

.....

.....[2]

(b) Identify, using pseudocode, another loop structure that the algorithm in **part (a)** could have used.

.....

.....[1]

5 The global trade item number (GTIN-8) barcode has seven digits and a check digit.
This pseudocode algorithm inputs seven digits and calculates the eighth digit, then outputs the GTIN-8.
DIV(X,Y), finds the number of divides in division for example **DIV(23,10)** is

2. **MOD(X,Y)**, finds the remainder in division for example **MOD(23,10)** is 3.

```
FOR Count ← 1 TO 7
   INPUT Number
   Digit(Count) ← Number
NEXT
Sum ← (Digit(1)+Digit(3)+Digit(5)+Digit(7))*3+Digit(2)+Digit(4)+Digit(6) IF
MOD(Sum,10) <> 0
   THEN Digit(8) ← DIV(Sum,10)*10 + 10 - Sum
   ELSE Digit(8) ← 0
ENDIF
OUTPUT "GTIN-8"
FOR Count ← 1 TO 8
   OUTPUT Digit(Count)
NEXT
```

(a) Complete the trace table for the input data: 5, 7, 0, 1, 2, 3, 4

Digit(1)	Digit(2)	Digit(3)	Digit(4)	Digit(5)	Digit(6)	Digit(7)	Digit(8)	Sum	OUTPUT

Complete the trace table for the input data: 4, 3, 1, 0, 2, 3, 1

Digit(1)	Digit(2)	Digit(3)	Digit(4)	Digit(5)	Digit(6)	Digit(7)	Digit(8)	Sum	OUTPUT

[5]

(b) Explain how you would change the algorithm to input eight digits (seven digits and the check digit) and output if the check digit entered is correct or not.

.....

.....

.....

.....

.....

.....

..... [3]

6 This section of program code reads the contents of the array, totals the numbers and prints out the sum and average of the numbers. Assume the array is full.

Complete the **four** missing items by writing them in the spaces provided in this code.

```
1   Numbers[1:30]
2   Total = 0
3   ..... = 0
4   FOR Count = 1 TO .....
5   Number = Numbers[Count]
6   Total = ..... + Number
7   Counter = Counter + 1
8   _____                                    Count
9   PRINT 'The sum of the numbers you entered is ', Number
10  PRINT 'The average of the numbers you entered is ', Number / Counter
```

[4]

6 The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
  INPUT Values[Count]
  Count ← Count + 1
UNTIL Count = 0
```

(a) Explain why the algorithm will never end.

.....

.....

.....

.....

..... [2]

(b) Re-write the original pseudocode so that it terminates correctly **and** also prevents numbers below 100 from being stored in the array `Values[ ]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

.....

.....

.....

..... [2]

7 Arrays are data structures used in programming. Explain what is meant by the terms dimension and index in an array. Use examples of arrays in your explanations.

Dimension .....

.....

.....

.....

.....

Index .....

.....

.....

.....

.....

[3]