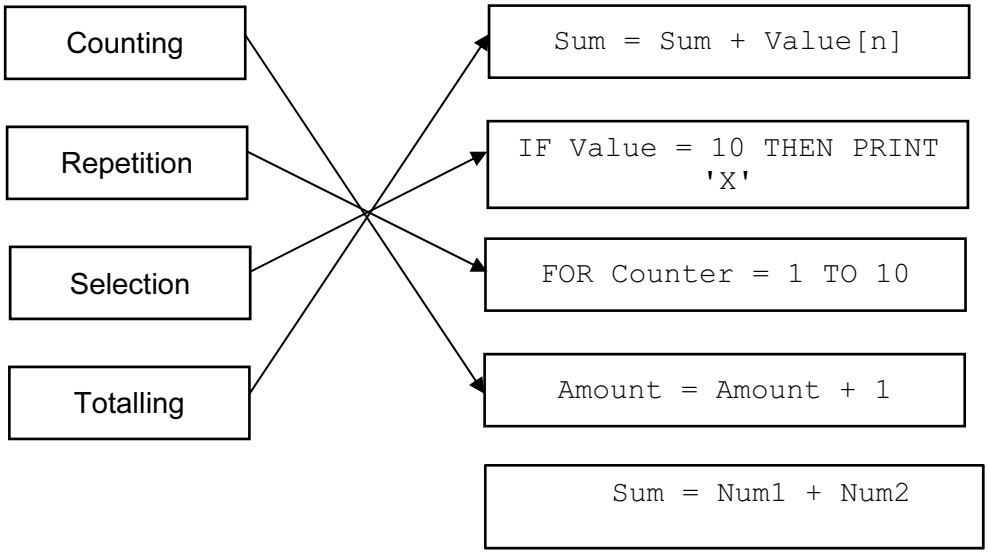
HANDOUT FOR CHAPTER 12

ARRAYS

Marking Scheme

4 1 mark for each correct line, two lines from one box not allowed



[4]

5 **(a)** **1** mark for `FOR … TO … NEXT` **1** mark for `INPUT`

```
FOR Count ← 1 TO 1000
  INPUT A[Count]
NEXT (Count)
```

[2]

(b) 4 marks
– initialisation
– start of loop
– update loop counter
– end of loop

Example1

```
Count ← 1                          (1 mark)
REPEAT                             (1 mark)
  INPUT A[Count]
  Count ← Count + 1                (1 mark)
UNTIL Count > 1000                 (1 mark)
```

Example2

```
Count ← 0                          (1 mark)
WHILE Count < 1000                 (1 mark)
  DO
  Count ← Count + 1                (1 mark)
  INPUT A[Count]
ENDWHILE                           (1 mark)
```

[4]

Section B

2 **(i)** **1** mark for each change

Change variable name in every instance as needs to be meaningful e.g. `Large`
Set this variable to a low value
line 5: change comparison from < to > [3]

(ii) **3** marks maximum, 1 mark for each change correctly included.

```
1  Large = 0
2  Counter = 0
3  REPEAT
4    INPUT Num
5    IF Num > Large THEN Large = Num
6    Counter = Counter + 1
7  UNTIL Counter = 10
8  PRINT Large
```

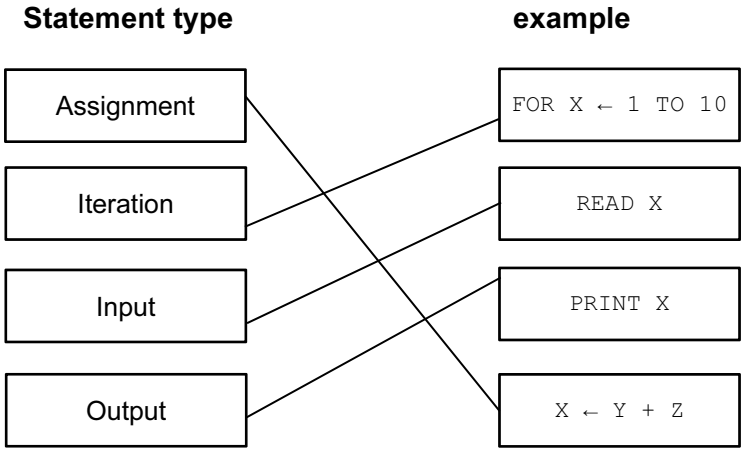
[3]

3 **(i)** Name type – string
Gender type – char/string
Status type – char/string
Fee type – real
Team member type – Boolean [5]

(ii) Data Structure – several Arrays ……
…….Reason – to simplify programming/ make programs shorter/index can be used to identify the same member across the arrays etc.

[2]

4 **1** mark for each correct line, maximum **3** (zero correct **0**, one correct **1**, two correct **2**, three or four correct **3**), each box must have only one connection.



[3]

5 – data structure (one—dimensional) array …….
– …… reason to simplify programming/ make programs shorter, etc. [2]

6 – `IF (… THEN … ELSE … ENDIF)`

– `CASE (… OF … OTHERWISE … ENDCASE)` [2]

7 **(a)** – 7 [1]

(b) – Brochure Number……..
– ….. Uniquely identifies each record/each Brochure Number different/no duplicates [2]

(c) – Number of Seats – number/integer
– Price in $ – currency/real [2]

(d) **1** mark for each correct result, **1** mark for the results in descending order of price

– Recliner sofa 1,200 RS23
– Recliner chair 600 RC01 [3]

Question	Answer	Marks
5(a)	Any **two** from: - Loop with 300 repetitions (starting at 1) / Loops from 1 to 300 - Values input/stored (in consecutive/different locations) in an array (at position I) - Increases the loop counter/I value by 1 (and returns to the start of the loop)	**2**

5(b)	Any **one** from:
	`REPEAT (… UNTIL)`
	`WHILE (… DO … ENDWHILE)`
		1

5(c)	- Prompt and input number (1)
	- Checking the input number is between 0 and 100 - both limits (1)
	- Correct error message (1)
	
	Many correct algorithms. This is an example only.
	
	`OUTPUT "Enter a number between 0 and 100 "`
	`INPUT Number`
	`IF Number < 0 OR Number > 100`
	`  THEN`
	`  OUTPUT "The number you have entered is outside the specified range"`
	`ENDIF`
		3

Question	Answer	Marks
6		**5**

HighF	HighC	TempF	OUTPUT
–100	–100		
		68	
68	18	46	
68	18	50	
68	18	86	
86	27	65	
86	27	50	
86	27	40	
86	27	30	
86	27	–1	The highest temperature is, 86 Fahrenheit, 27 Celsius.

(1 Mark) (1 Mark) (1 Mark) (2 Marks – see below)

The literal correct output is “The highest temperature is, 86 Fahrenheit, 27 Celsius.”
1 mark for values 86 and 27, 1 mark for correct output words, spacing and punctuation.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(a)</td><td>
<table>
<tr><th>Digit(1)</th><th>Digit(2)</th><th>Digit(3)</th><th>Digit(4)</th><th>Digit(5)</th><th>Digit(6)</th><th>Digit(7)</th><th>Digit(8)</th><th>Sum</th><th>OUTPUT</th></tr>
<tr><td>5</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>6</td><td>44</td><td>GTIN-8</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>57012346</td></tr>
</table>
<table>
<tr><th>Digit(1)</th><th>Digit(2)</th><th>Digit(3)</th><th>Digit(4)</th><th>Digit(5)</th><th>Digit(6)</th><th>Digit(7)</th><th>Digit(8)</th><th>Sum</th><th>OUTPUT</th></tr>
<tr><td>4</td><td>3</td><td>1</td><td>0</td><td>2</td><td>3</td><td>1</td><td>0</td><td>30</td><td>GTIN-8</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>43102310</td></tr>
</table>
One mark for data entry – both sets of digits 1–7
One mark for both Digit(8)
One mark for each Sum (max Two)
One mark for both OUTPUT
</td><td>5</td></tr>
<tr><td>3(b)</td><td>
Any three from
1 Change first loop to 8 iterations
2 Check that the input <code>Digit(8)</code> is equal to the calculated <code>Digit(8)</code>…
3 … if equal output check digit correct
4 … otherwise output check digit incorrect

Or

1 Change first loop to 8 iterations
2 Put all 8 digits through the algorithm to calculate <code>Sum</code> …
3 … if <code>MOD(Sum,10)</code> is equal to zero, check digit correct
4 … otherwise output check digit incorrect
</td><td>3</td></tr>
</table>

Question	Answer	Marks
	Section B	
2(a)	**One** mark for description **one** mark for example e.g. To test if the data entered is possible / reasonable … … A range check tests that data entered fits within specified values. Allow any correct validation check as an example	**2**
2(b)	**One** mark for description **one** mark for example e.g. To test if the data input is the same as the data that was intended to be input … … A double entry check expects each item of data to be entered twice and compares both entries to check they are the same. Allow any correct verification check as an example	**2**

Question	Answer	Marks
3	**One** mark for each correct answer Counter = 0 FOR Count = 1 TO 30 Total = Total + Number NEXT Count	**4**

Question	Answer	Marks
3(a)	**One** mark for each correct validation check (max **two**) • Range • Length • Type • Check Digit **One** mark for each correct related purpose (max **two**) e.g. • To make sure the data entered falls within a specific set of values • To make sure the data entered is no longer than specified • To make sure the data entered follows rules related to whether it is numbers of letters • To make sure an identification code entered is genuine or possible	**4**
3(b)	**One** mark for correct verification check (max **one**) • Double (data) entry • Visual check	**1**
3(c)	Any **two** correct statements (max **two**) e.g. • Validation checks if the data entered is possible/it cannot check if data has been entered correctly. • Verification checks if the data entered matches the data submitted for entry/ it does not check if data matches set criteria.	**2**

Question	Answer	Marks
4(a)	Any **two** correct statements (max **two**) e.g. • The value of the variable Count begins as 0 … • … and is incremented by 1 before it is tested by the loop condition • Count will never be 0 at the end of the loop	**2**

Question	Answer	Marks

4(b)	
	`Count ← 0`
	`REPEAT`
	`  INPUT Number`
	`  IF Number >= 100`
	`  THEN`
	`  Values[Count] ← Number`
	`  ENDIF`
	`Count ← Count + 1`
	`UNTIL Count = 50`
	
	One mark – separate INPUT statement
	One mark – IF statement attempted
	One mark – IF statement completely correct
	One mark – termination of loop updated
		4

4(c)	Any **two** correct statements (max **two**) e.g. • Alter the IFstatement/add a second IF statement/comparison that's already there … • … so that additional criteria set an upper limit of <=200	**2**

Question	Answer	Marks
5	**One** mark for explanation of dimension **One** mark for explanation of index **One** mark for inclusion of an example The dimension is the number of indexes required to access an element. The index is the position of the element in an array For example A[25] is the 25th element of a one-dimensional array.	**3**

Question	Answer	Marks
6(a)	**One** mark for field and **one** mark for reason Field Juice code Reason only unique identifier	**2**
6(b)	<table><tr><td>Field:</td><td>Fruit 1</td><td>Fruit 2</td><td>Size</td><td>Stock level</td></tr><tr><td>Table:</td><td>JUICE</td><td>JUICE</td><td>JUICE</td><td>JUICE</td></tr><tr><td>Sort:</td><td></td><td></td><td></td><td></td></tr><tr><td>Show:</td><td>☐</td><td>☐</td><td>☑</td><td>☑</td></tr><tr><td>Criteria:</td><td>="Apple"</td><td>="Apple"</td><td></td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> One mark for each correct column	**4**