MARK SCHEME for the May/June 2015 series

0478 COMPUTER SCIENCE

0478/22	Paper 2 (Written), maximum raw mark 50

Section A

1 (a) (i) **Many correct answers, they must be meaningful. This is an example only.**
– `PupilName[1:30]`
or `PupilName[0:29]`
or `PupilName[30]`
or `PupilName[29]`
or `PupilName[]` [1]

(ii) **Many correct answers, they must be meaningful. This is an example only.**
– `StartWeight[1:30]`
or `StartWeight[0:29]`
or `StartWeight[30]`
or `StartWeight[29]`
or `StartWeight[]` [1]

(iii) Answers, must match (i) and (ii) above and the upper bound should have been changed from 30 to 600 or 29 to 599 or no change if not used.
– `StartWeight[1:600]` or `StartWeight[600]`
– `PupilName[1:600]` or `PupilName[600]` [1]

(b) any **four** from
– prompt for entry of final weight that includes pupil's name
– input final weight
– validation check for final weight
– calculation of difference in weight
– ……using the initial weight stored in the array
– store difference in weight

(Max 4 marks)

– loop for 600 pupils

(1 mark) [5]

sample algorithm:

```
FOR Count ← 1 TO 600
    REPEAT
        PRINT 'Please enter weight for ', PupilName[Count]
        INPUT FinalWeight
    UNTIL FinalWeight < 120 AND FinalWeight > 20
    WeightDifference[Count] ← FinalWeight - StartWeight[Count]
NEXT Count
```

(c) (i) any **two** from
– check that the weights are within a given range
– check that the weights are numeric
– check that the weights are given to one decimal point
– character/type check on name
– length check on name

[2]

(ii) **1** mark for the data and **1** mark for the matching reason.
There are many possible correct answers this is an example only.
Weight 1   – 35.2
Reason   – normal data that should be accepted

Weight 2   – twenty
Reason   – abnormal data that should be rejected   [4]

(d) Maximum 6 marks **in total** for question part
Explanation (max 6)
– loop 30 or 600 times to check each difference in weight
– check for a difference in weight
– less than -2.5 (final weight – start weight) or greater than 2.5 (start weight – final weight)
– …If so output pupil's name
– …if so output difference in weight
– …if so output message that it is a fall in weight

Sample algorithm (max 4)
```
FOR Count ← 1 TO 30
    IF WeightDifference [Count] < -2.5
     THEN PRINT PupilName[Count], 'The weight loss was ',
        WeightDifference [Count]
    ENDIF
NEXT Count
```

If pseudocode or programming only and no explanation, then maximum 4 marks   [6]

Section B

2 1 mark for each error identified + suggested correction
Line 1 or Large =9999:   this should read **Large = 0**
Line 3 or WHILE: this should read **WHILE Counter < 30**
line 6 or IF: this should read **IF Num > Large THEN Large = Num**
line 7 or Counter =…: this should read **Counter = Counter + 1**   [4]

3 **(a)**

Trace table set 1

A	B	C	D	E	F	Total	Check	Output
5	2	4	3	1	5	38	5	Accept

←--------------------------------(1 mark)----------------------→←----------(1 mark)------→

Trace table set 2

A	B	C	D	E	F	Total	Check	Output
3	2	1	0	7	3	45	1	Reject

←--------------------------------(1 mark)----------------------→←----------(1 mark)------→

[4]

(b) – (modulo 11) check digit calculation   [1]

(c) **1** mark for identifying the problem, **2** marks for the solution
   Problem   – doesn't deal correctly with remainder 10/a check digit of X
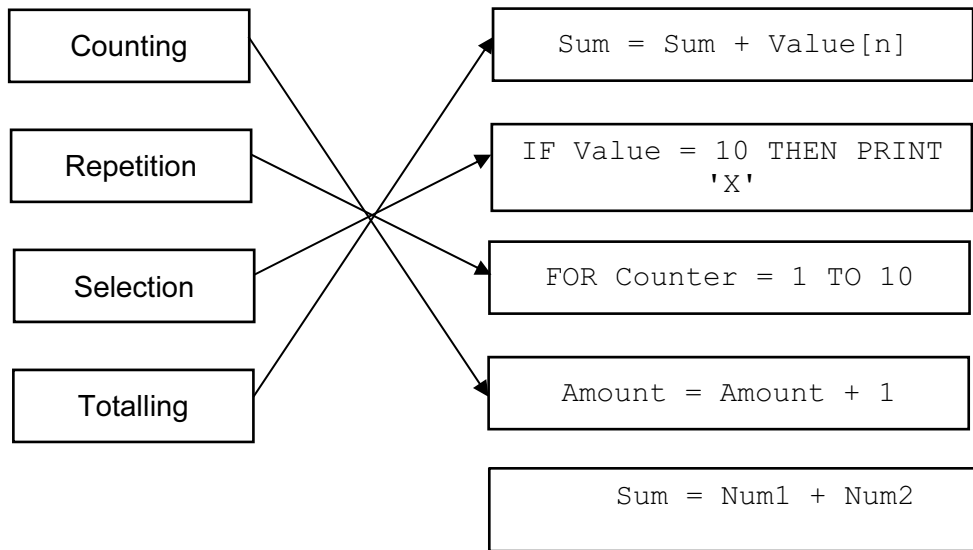   **Solution**   – check Z for X as a final digit
                         – have a special case where check = 10
                         – accept where Check = 10 and F = X   [3]

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

Counting		Sum = Sum + Value[n]
Repetition		IF Value = 10 THEN PRINT 'X'
Selection		FOR Counter = 1 TO 10
Totalling		Amount = Amount + 1
		Sum = Num1 + Num2

[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]

6 (a) – 7 [1]

   (b) – Class ID
       – Uniquely identifies each student [2]

   (c) Diana Abur, Paul Smith
       – both names
       – ……. correct order [2]

   (d)

Field:	Student Name	Maths	English
Table:	MARKS	MARKS	MARKS
Sort:			
Show:	☑	☐	☐
Criteria:		<40	<40
or:			

            (1 mark)   (1 mark)   (1 mark)   [3]