Cambridge Assessment International Education

Cambridge IGCSE®

COMPUTER SCIENCE **0478/02**

Paper 2 Problem-solving and Programming **For examination from 2020**

MARK SCHEME

Maximum Mark: 50

Specimen

This document consists of **6** pages.

Section A

1 (a) (i) Many correct answers, they must be meaningful. This is an example only.

```
StudentNames[1:30]
```
[1]

(ii) Many correct answers, they must be meaningful. This is an example only.

```
StudentMarksTest1[1:30]
StudentMarksTest2[1:30]
StudentMarksTest3[1:30]
```
(1 mark)

```
StudentTotalScore[1:30]
```
(1 mark) [2]

(b) (i)
- outside loop zeroing total for loop (sum in example below)
- loop for all students
- input name and all test scores
- in loop adding a student's total
- storing the total
- inside loop printing student's name and total
- outside loop calculating class average
- printing class average

sample algorithm:

```
Sum ← 0
FOR Count ← 1 TO 30
   INPUT Name
   StudentName[Count] ← Name
   INPUT Mark1, Mark2, Mark3
   StudentMarksTest1[Count] ← Mark1
   StudentMarksTest2[Count] ← Mark2
   StudentMarksTest3[Count] ← Mark3
   Total ← Mark1 + Mark2 + Mark3
   StudentTotalScore[Count] ← Total
   Sum ← Sum + Total
   PRINT StudentName[Count], StudentTotalScore[Count]
NEXT Count
ClassAverage = Sum/30
PRINT ClassAverage
```
[8]

(ii) any relevant comment with regards to efficient code (e.g. single loop) [1]

(c) Many correct answers, these are examples only.
1 mark per data set and reason

Set 1: 20, 25, 35
Reason: valid data to check that data on the upper bound of each range check is accepted

Set 2: 21, 26, 36
Reason: invalid data to check that data above the upper bound of each range check is rejected [2]

(d) (i) Maximum 5 marks **in total** for question part
Maximum 3 marks for algorithm

Description (max 3)
- set variable called HighestScore to zero and variable called BestName to dummy value
- loop 30 times to check each student's total score in turn
- check student's score against HighestScore
- if student's score > HighestScore then
- … replace value in HighestScore by student's score and store student's name in BestName
- output BestName and HighestScore outside the loop

Sample algorithm (max 3):

```
HighestScore ← 0
BestName ←  "xxxx"                                    (1 mark)
FOR Count ← 1 TO 30
   IF StudentTotalScore[Count] > HighestScore         (1 mark)
      THEN
         HighestScore ← StudentTotalScore[Count]
         BestName ← StudentName[Count]                (1 mark)
   ENDIF
NEXT Count                                            (1 mark)
PRINT BestName, HighestScore                          (1 mark)
```

If algorithm or program code only, then maximum 3 marks [5]

(ii) comment on which student(s)' name will be output
e.g. The first student with the highest score will be output [1]

Section B

2 **(a)** 1 mark for value of c and message

51020: value of c: 5
message: PIN OK (1 mark)

5120: value of c: 4
message: error in PIN entered (1 mark) [2]

(b) length check [1]

3

Engine	Count	Number	Size	Average	OUTPUT
0	0	0	1.8		
1.8	1	1	2.0		
3.8	2	2	1.0		
4.8		3	1.3		
6.1		4	1.0		
7.1		5	2.5		
9.6	3	6	2.0		
11.6	4	7	1.3		
12.9		8	1.8		
14.7	5	9	1.3		
16.0		10	–1		
				1.6	
					1.6, 5
(1 mark)	(1 mark)	(1 mark)	(1 mark)	(1 mark)	(1 mark)

[6]

4 1 mark for each error identified + suggested correction

line 5: this should read ***IF x > h THEN h = x***

line 7: ***PRINT h*** should come <u>after the end of the repeat loop</u>

line 8: this should read ***UNTIL c = 20*** or ***UNTIL c >= 20*** or ***UNTIL c > 19*** [3]

5 *PENDOWN*
LEFT 90
REPEAT **3**
FORWARD 30
RIGHT 90
} (1 mark)

ENDREPEAT
FORWARD 10
LEFT 90 OR **PENUP**
} (1 mark)

PENUP OR **LEFT 90**
FORWARD 10
PENDOWN
} (1 mark)

REPEAT 2 OR **REPEAT 3**
FORWARD 20
} (1 mark)

RIGHT 90
ENDREPEAT
FORWARD 20 OR (**LEFT/RIGHT 180)**
(LEFT 90)
} (1 mark)

Alternative answer *for* ***last 2 marks:***

FORWARD 20
RIGHT 90
} (1 mark)

FORWARD 20
RIGHT 90
FORWARD 20
} (1 mark)

Give a mark for each correct group of statements [5]

6 **(a)** marking points:

the way to find and print the largest value a	1 mark
the way to find and print the largest value b	1 mark
the way to find and print the largest value c	1 mark

sample algorithm:

```
INPUT a, b, c
   IF a > b AND a > c THEN PRINT a          (1 mark)
      ELSE IF b > c THEN PRINT b            (1 mark)
         ELSE PRINT c                       (1 mark)
```
[3]

(b) marking points:

loop construct	1 mark
check if number is an *integer*	1 mark
counting the number of integers input	1 mark
output count value (outside the loop)	1 mark

sample algorithm:

```
FOR x ← 1 TO 1000                                   (1 mark)
   INPUT Number
      Difference ← INT(number) - Number             (1 mark)
      IF Difference = 0 THEN Total ← Total + 1      (1 mark)
NEXT x
PRINT total                                         (1 mark)
```
(NOTE: alternative to lines 3 and 4:

```
IF INT(Number) = Number THEN Total ← Total + 1      (2 marks) )
```
[4]

(c) Description of any **two** sets of test data. Many correct answers, these are examples only.

1000 whole numbers to ensure that loop works properly

900 whole numbers and 100 numbers with decimal places to ensure that the routine distinguishes correctly [2]

7 **(a)** 7 [1]

(b) Hg, Cs [2]

(c) Element symbol [1]