Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER				

1649546159

COMPUTER SCIENCE **0478/23**

Paper 2 Problem-solving and Programming **October/November 2022**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- **Do not attempt Tasks 1, 2 and 3** in the copy of the pre-release material on page 2; these are for information only.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- Calculators must **not** be used in this paper.

INFORMATION

- The total mark for this paper is 50.
- The number of marks for each question or part question is shown in brackets [ ].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Any blank pages are indicated.

Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the following tasks before the examination to answer Question 1.

Pre-release Material

A program is needed for a window cleaning company to store details of the services provided to their customers for each cleaning job.

The data stored will be used to show the customers exactly what services they are paying for (provide an itemised bill). The data will allow the company to find out which services are used most often and least often.

The services available to customers are:

Service	Cost
basic window clean outside, only one floor, up to five windows	$10.00
additional windows up to and including five	$5.00
two floors	10% extra
three floors	15% extra
inside as well	25% extra
polish all windows cleaned	5% extra
special solar panel clean	$20.00

The total cost of the customer's bill is calculated in the same order as the rows of the table.
For example, a customer who has six windows over three floors is charged $10 + $5 with 15% extra, giving a total bill of $17.25.

Write and test a program or programs for the window cleaning company:

- Your program or programs must include appropriate prompts for the entry of data. Data must be validated on entry.
- All outputs, including error messages, need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – setting up the system to store the customer and service details

Set up your program to:
- store a customer name and address in a single dimensional array at the next position
- use the array index of this position as the unique number of the itemised bill
- store any service required at the same position in the appropriate service array; use an array for each service.

Task 2 – storing the services for a customer and displaying the itemised bill

Extend your program to:
- display the services and the cost of each one
- input and store a customer's details and the services they require
- calculate the total cost
- display the itemised bill including the total cost and the unique number of the bill
- repeat as required.

Task 3 – providing statistics about the services

Excluding the basic window clean and the additional windows, find the most popular service and the least popular service. For these two services, display the service name, whether it was the most or least popular and the number of times each service was used as a percentage of the total bills stored.

1 All variables, constants and other identifiers must have meaningful names.

(a) (i) Identify **one** constant that you could have used for **Task 1**.

State the value that would be assigned to the constant.

Give a reason why a constant was used rather than a variable.

Constant name .....

Value .....

Reason .....

.....

.....

[3]

(ii) Describe the array that you set up in **Task 1** to record the customer details.

Include the name, data type and sample data for the array in your description.

.....

.....

.....

.....

.....

..... [3]

(b) Explain how your program allowed only two floors or three floors to be chosen but **not** both.

.....

.....

.....

.....

.....

..... [3]

(c) Write an algorithm for the part of **Task 2** that calculates the total cost of the bill.
Assume that the customer details and services required have already been input.

Use pseudocode, programming statements or a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

(d) Explain how your program completed **Task 3**.

Any programming statements that you include in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

Section B starts on page 8.

Section B

2 An algorithm has been written to:

- set all 50 elements of the array `Reading[1:50]` to zero
- input values between 35 and 50 inclusive
- end the process when an input of –1 is made or 50 valid numbers have been entered
- reject all other values
- count the number of times each valid value is input
- output the number of times each value has been input, starting with the lowest value.

(a) Complete the pseudocode algorithm:

```
01  FOR Count ← 1 TO .....
02    Reading[Count] ← 0
03  NEXT Count
04  Count ← 1
05  OUTPUT "Please enter next reading "
06  INPUT Value
07  REPEAT
08    IF Value < 35 OR .....
09      THEN
10        OUTPUT "Reading out of range"
11      ELSE
12        Reading[Value] ← .....
13    Count = Count + 1
14    ENDIF
15    IF Count <= 50
16      THEN
17        OUTPUT "Please enter next reading "
18       .....
19    ENDIF
20  UNTIL Value = -1 OR Count > 50
21  Count ← 35
22  REPEAT
23    OUTPUT "There are ", .....,
             " readings of ", Count
24    Count ← .....
25  UNTIL Count > 50
```

[6]

(b) Describe how the algorithm could be changed to output the number of times each value has been input, starting with the highest value.

.....

.....

.....

.....

.....

..... [3]

3 A check digit is to be used to validate an identification number on input. The identification number contains five digits and the check digit.
The check digit is calculated by adding up the first five digits, dividing by 10 and taking the remainder.
For example, `5 + 1 + 2 + 4 + 3` divided by `10` gives a remainder of `5` so the six-digit identification number would be `512435`

(a) (i) Calculate the check digit for `69321`

.....

..... [1]

Working space .....

.....

.....

(ii) State which of these identification numbers have incorrect check digits.

A `123455`
B `691400`
C `722855`
D `231200`

.....

..... [2]

Working space .....

.....

.....

(b) (i) Describe an input error that would **not** be found using this check digit.

.....

.....

.....

..... [2]

(ii) Describe a more suitable algorithm to calculate the check digit for this identification number.

.....

.....

.....

..... [2]

(c) Identify **two** other validation checks that could be used when inputting this identification number.

1 .....

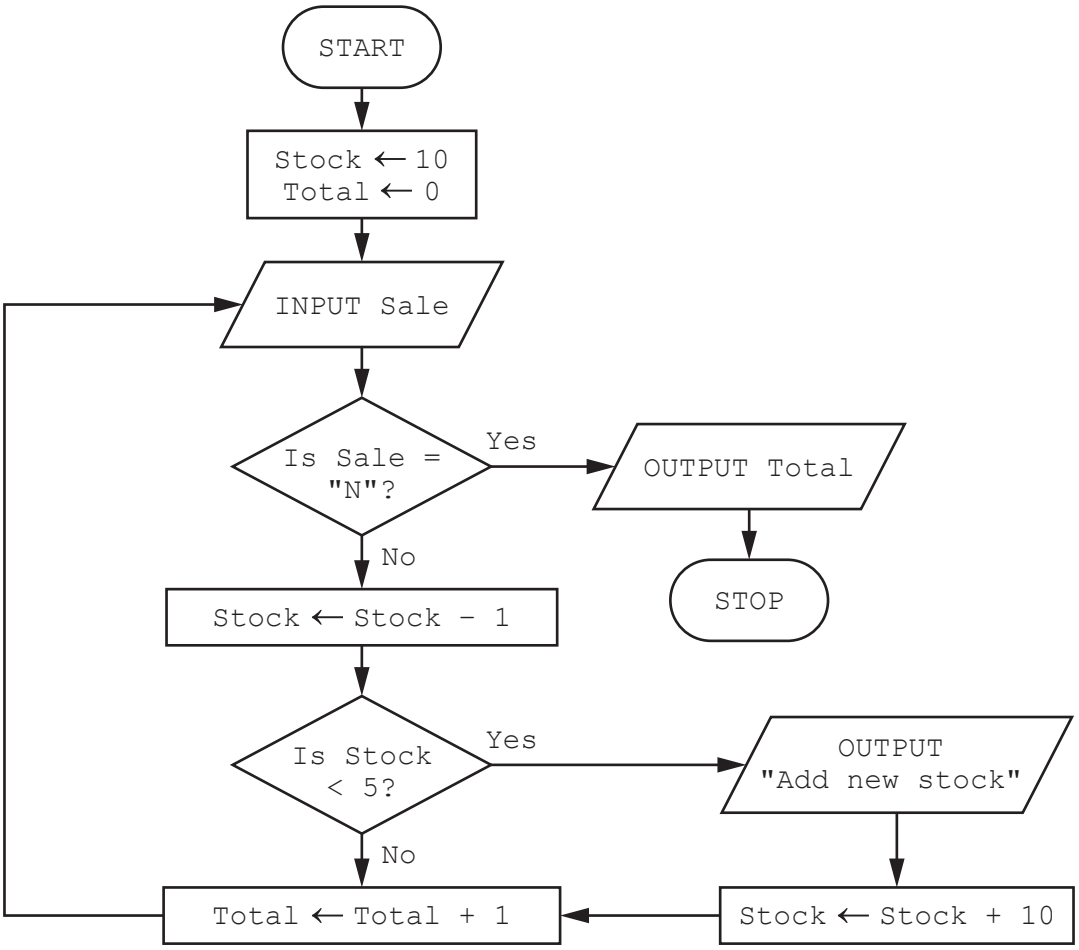
.....

2 .....

.....

[2]

4 This algorithm makes sure that there are enough wheelbarrows in stock.



(a) Complete the trace table for the algorithm using this input data:

"Y", "Y", "Y", "Y", "Y", "Y", "N"

Stock	Total	Sale	OUTPUT

[4]

(b) Explain how you could extend the algorithm to allow for the sale of more than one wheelbarrow at a time.

.....

.....

.....

.....

.....

.....

..... [3]

5 A database table, MUSEUM, is used to keep a record of items in the museum. The table has these fields:

- ItemCode – code for each type of item, for example ART0005
- Description – brief description of each item, for example gold coin
- InStore – whether the item is in store or **not**, for example Y
- Century – century when item made, for example 18
- Country – country of origin, for example China.

(a) Identify which field you would choose for the primary key.

State a reason for your choice.

Field .....

Reason .....

.....

[2]

(b) Complete the query-by-example grid to display only the description and the country of origin for those items **not** in store.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

[3]

BLANK PAGE

BLANK PAGE

Cambridge Assessment International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/23**

Paper 2 Problem-solving and Programming **October/November 2022**

MARK SCHEME

Maximum Mark: 50

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the October/November 2022 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

[Turn over

PUBLISHED

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Please note the following further points:

The words in **bold** in the mark scheme are important text that needs to be present, or some notion of it needs to be present. It does not have to be the exact word, but something close to the meaning.

If a word is underlined, this **exact** word must be present.

A single forward slash means this is an alternative word. A double forward slash means that this is an alternative mark point.

Ellipsis (…) on the end of one-mark point and the start of the next means that the candidate **cannot** get the second mark point without being awarded the first one. If a mark point has an ellipsis at the beginning, but there is no ellipsis on the mark point before it, then this is just a follow-on sentence and **can** be awarded **without** the previous mark point.

Question	Answer	Marks
	Section A	
1(a)(i)	Many correct answers, the name used must be meaningful. The name given is an example only. **One** mark per mark point, max **three** Constant name `BasicClean` Value `10.00` Reason this number will not change whilst the program is running	**3**
1(a)(ii)	Description Many correct answers, the name used must be meaningful. The name is an example only. **One** mark per mark point, max **three** MP1 name e.g. `NameAddress` MP2 appropriate data type e.g. string MP3 sample data for the array e.g. Mrs Singh, Park View	**3**
1(b)	**One** mark per mark point, max **three** MP1 use a conditional statement // an example of conditional statement used // use an input statement with validation// an example of input statement used MP2 any working method – for example, to only store `True` in the three floors array if the two floors array has not been set to `True` MP3 output an appropriate error message when an error condition has occurred // choose appropriate routes through the program depending upon the choice of floors	**3**

Question	Answer	Marks

1(c)	**One** mark per mark point, max **six**
	
	MP1 initialise bill total to 10
	MP1 check for additional windows
	MP2 … and add in amount to total if required
	MP3 check for 2 floors **or** 3 floors
	MP4 … multiply by 1.1 **or** 1.15 as required
	MP5 check for inside and multiply by 1.25 if required
	MP6 check for polish and multiply by 1.05 if required
	MP7 MP2 to MP7 completed in correct order
	MP8 check for solar panels and add 20.00 if required
	
	Sample
	`Total ← 10.00`
	`ExtraWindowsCost ←INT((NumberExtraWindows[Bill] −1) / 5) * 5.00`
	`Total ← Total + ExtraWindowsCost`
	`CASE NumberOfFloors[Bill] OF // check number of floors`
	`  2 : Total ← Total * 1.1`
	`  3 : Total ← Total * 1.15`
	`ENDCASE`
	`IF Inside[Bill] // check for cleaning inside windows`
	`  THEN`
	`    Total ← Total * 1.25`
	`ENDIF`
	`IF Polish[Bill] // check for polishing windows`
	`   THEN`
	`    Total ← Total * 1.05`
	`ENDIF`
	`IF Solar[Bill] // check for cleaning solar panels`
	`  THEN`
	`    Total ← Total + 20.00`
	`ENDIF`
		6

Question	Answer	Marks
1(d)	Explanation **One** mark per mark point, max **five** MP1 how the program counted the number of customers using a service MP1 … how the program counted this for all services MP2 … excluding basic windows and additional windows MP3 how the program identified the service with the largest number of customers MP4 how the program identified the service with the smallest number of customers MP5 how the program calculated a percentage of the total number of bills stored… MP6 … for largest and smallest MP7 how the program displayed these percentages with suitable messages Programming statements when used must be explained.	**5**

Question	Answer	Marks
	Section B	
2(a)	**One** mark per mark point in the correct position, max **six** • line 01 `50` • line 08 `Value > 50` • line 12 `Reading[Value] + 1` • line 18 `INPUT Value` • line 23 `Reading[Count]` • line 24 `Count + 1`	**6**

2(b)	**One** mark per place in code **and** action, max **three**
	
	• line 21 set Count to 50 / `Count ← 50`
	• line 24 subtract 1 from Count / `Count ← Count -1`
	• line 25 check for Count equal to 34 / check for Count less than 35 / `UNTIL Count = 34` / `UNTIL Count < 35`
	or
	One mark per place in code **and** action, max **three**
	
	• line 21 set up FOR loop stating at 50 and finishing at 35 / `FOR Count ← 50 TO 35 STEP -1`
	• Remove lines 22 and 24
	• line 25 End FOR loop / `NEXT Count`
	
	Examples
	`21 Count ← 50`
	`22 REPEAT`
	`23 OUTPUT "There are ", Reading[Count], " readings, " of ", Count`
	`24 Count ← Count -1`
	`25 UNTIL Count = 34`
	
	`21 FOR Count ← 50 TO 35 STEP -1`
	`23 OUTPUT "There are ", Reading[Count], " readings, " of ", Count`
	`25 NEXT Count`
		3

Question	Answer	Marks
3(a)(i)	• 1	**1**
3(a)(ii)	**One** mark for each correct letter seen, max **two** • C • D	**2**
3(b)(i)	**One** mark per mark point, max **two** • two or more digits • transposed	**2**
3(b)(ii)	**One** mark per mark point, max **two** • multiply each digit by a different number / its place value • before adding them together and dividing by a number	**2**
3(c)	**One** mark per mark point, max **two** • length check • type check • presence check • format check	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(a)</td><td>One mark for each correct column, max four
<table>
<tr><th>Stock</th><th>Total</th><th>Sale</th><th>OUTPUT</th></tr>
<tr><td>10</td><td>0</td><td></td><td></td></tr>
<tr><td>9</td><td>1</td><td>Y</td><td></td></tr>
<tr><td>8</td><td>2</td><td>Y</td><td></td></tr>
<tr><td>7</td><td>3</td><td>Y</td><td></td></tr>
<tr><td>6</td><td>4</td><td>Y</td><td></td></tr>
<tr><td>5</td><td>5</td><td>Y</td><td></td></tr>
<tr><td>4</td><td>6</td><td>Y</td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>N</td><td>Add new stock</td></tr>
<tr><td></td><td></td><td></td><td>6</td></tr>
</table></td><td>4</td></tr>
<tr><td>4(b)</td><td>One mark per mark point, max three

• input a number / quantity
• … check that the number is less than (or equal to) the number in stock
• … after checking for N
• update <code>Stock</code> by that number / <code>Stock ← Stock – Number</code>
• update <code>Total</code> by that number / <code>Total ← Total + Number</code></td><td>3</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>One mark per mark point, max two

• ItemCode
• uniquely identifies each item</td><td>2</td></tr>
<tr><td>5(b)</td><td>One mark per mark point, max three

• correct rows Field, Table and Sort
• correct row Show
• correct Criteria row

<table>
<tr><td>Field:</td><td>Description</td><td>Country</td><td>InStore</td><td></td></tr>
<tr><td>Table:</td><td>MUSEUM</td><td>MUSEUM</td><td>MUSEUM</td><td></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></td><td></td><td>N</td><td></td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td>3</td></tr>
</table>