Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER			

0337186959

COMPUTER SCIENCE **0478/22**

Paper 2 Problem-solving and Programming **May/June 2021**

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.

© UCLES 2021

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 tasks before the examination to answer Question 1.

Pre-release material

An electric mountain railway makes four return trips every day. In each trip the train goes up the mountain and back down. The train leaves from the foot of the mountain at 09:00, 11:00, 13:00 and 15:00. The train returns from the top of the mountain at 10:00, 12:00, 14:00 and 16:00. Each train has six coaches with eighty seats available in each coach. Passengers can only purchase a return ticket; all tickets must be purchased on the day of travel. The cost is $25 for the journey up and $25 for the journey down. Groups of between ten and eighty passengers inclusive get a free ticket for every tenth passenger, provided they all travel together (every tenth passenger travels free). Passengers must book their return train journey, as well as the departure train journey, when they purchase their ticket. Passengers can return on the next train down the mountain or a later train. The last train from the top of the mountain has two extra coaches on it.

The train times are displayed on a large screen, together with the number of tickets still available for each train. Every time a ticket is booked the display is updated. When a train is full, the word 'Closed' is displayed instead of the number of tickets available.

Write and test a program or programs for the electric mountain railway.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output 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 – Start of the day.
Write a program to set up the screen display for the start of the day. Initialise suitable data structure(s) to total passengers for each train journey and total the money taken for each train journey. Each train journey must be totalled separately. There are four journeys up and four journeys down every day.

Task 2 – Purchasing tickets.
Tickets can be purchased for a single passenger or a group. When making a purchase, check that the number of tickets for the required train journeys up and down the mountain is available. If the tickets are available, calculate the total price including any group discount. Update the screen display and the data for the totals.

Task 3 – End of the day.
Display the number of passengers that travelled on each train journey and the total money taken for each train journey. Calculate and display the total number of passengers and the total amount of money taken for the day. Find and display the train journey with the most passengers that day.

© UCLES 2021

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

(a) Identify and give the data type and use of **one** array that you could have used for **Task 1**.

Array .....

Data type .....

Use .....

[3]

(b) Describe **two** validation checks that could be used when inputting the number of tickets to buy for **Task 2**. For each validation check give one example of normal data and one example of erroneous data.

Validation check 1 .....

.....

.....

Normal data .....

Erroneous data .....

Validation check 2 .....

.....

.....

Normal data .....

Erroneous data .....

[6]

(c) Write an algorithm for the part of **Task 2** that inputs the tickets required, calculates the total price for the ticket(s) purchased by a passenger including a group discount (if applicable) and updates the data structures used for the totals.
Use **either** pseudocode, programming statements **or** a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(d) Explain how your program completed **Task 3**. Include any programming statements that you have used and fully explain the purpose of each statement.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

Section B starts on page 8

Section B

2 **(a)** Write an algorithm in pseudocode to input 500 positive whole numbers. Each number input must be less than 1000. Find and output the largest number input, the smallest number input and the range (difference between the largest number and smallest number).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(b) Describe how the algorithm could be changed to make testing less time-consuming.

.....

.....

.....

..... [2]

3 **(a)** Draw the most appropriate flowchart symbol for each pseudocode statement.

Pseudocode statement	Flowchart symbol
`IF Number = 20`	
`PRINT Number`	
`Number ← Number + 1`	

[3]

(b) State the type of each pseudocode statement. For example, `X ← X + Y` is totalling.

`IF Number = 20` .....

`PRINT Number` .....

`Number ← Number + 1` .....

[3]

4 This algorithm checks passwords.

- Each password must be 8 or more characters in length; the predefined function `Length` returns the number of characters.
- Each password is entered twice, and the two entries must match.
- Either `Accept` or `Reject` is output.
- An input of 999 stops the process.

```
REPEAT
  OUTPUT "Please enter password"
  INPUT Password
  IF Length(Password) >= 8
    THEN
      INPUT PasswordRepeat
      IF Password <> PasswordRepeat
        THEN
          OUTPUT "Reject"
        ELSE
          OUTPUT "Accept"
      ENDIF
    ELSE
      OUTPUT "Reject"
  ENDIF
UNTIL Password = 999
```

(a) Complete the trace table for the algorithm using this input data:
Secret, Secret, VerySecret, VerySecret, Pa55word, Pa55word, 999, 888

Password	PasswordRepeat	OUTPUT

[3]

(b) Explain how the algorithm could be extended to allow three attempts at inputting the matching password. Any pseudocode statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

..... [4]

5 A one-dimensional array `dataArray[1:20]` needs each element set to zero.

(a) Write a pseudocode routine that sets each element to zero. Use the most suitable loop structure.

.....

.....

.....

.....

.....

.....

.....

..... [3]

(b) Explain why you chose this loop structure.

.....

..... [1]

6 A database table, PLANT, is used to keep a record of plants sold by a nursery. The table has these fields:

- NAME – name of plant
- FLOWER – whether the plant flowers (True) or not (False)
- POSITION – shade, partial shade or sun
- SIZE – small, medium or large
- PRICE – price in $
- NUMBERSOLD – how many sold

A query-by-example grid has been completed to display only the price, name and number sold of small plants that do not flower.

Field:	NAME	PRICE	NUMBERSOLD	SIZE	FLOWER	POSITION
Table:	PLANT					
Sort:						
Show:	☑	☑	☑	☐	☐	☐
Criteria:						= "shade"
or:						

Identify the errors in the query-by-example grid.

.....

.....

.....

.....

.....

.....

Rewrite the corrected query-by-example grid.

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

[5]

BLANK PAGE

© UCLES 2021

BLANK PAGE

BLANK PAGE

BLANK PAGE

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

Cambridge Assessment International Education is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which itself is a department of the University of Cambridge.

© UCLES 2021