Cambridge Assessment International Education

# Cambridge IGCSE™

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| CANDIDATE NAME | | | | | | | | | |
| CENTRE NUMBER | | | | | | CANDIDATE NUMBER | | | | |

*8130505114*

**COMPUTER SCIENCE** **0478/22**

Paper 2 Problem-solving and Programming **October/November 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 **12** pages.


© 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 following tasks before the examination to answer Question 1.

**Pre-release material**

An integrated transport system has been designed to reduce the need for privately owned vehicles. A vehicle is booked to take a passenger from home to a start station, from where they will travel to an end station. A vehicle at the end station will take the passenger to their destination. Each stage of the journey has a price code to represent the distance travelled. The prices for each stage are shown:

| Home to start station | |
|---|---|
| **Code** | **Price ($)** |
| C1 | 1.50 |
| C2 | 3.00 |
| C3 | 4.50 |
| C4 | 6.00 |
| C5 | 8.00 |

| Start station to end station | |
|---|---|
| **Code** | **Price ($)** |
| M1 | 5.75 |
| M2 | 12.50 |
| M3 | 22.25 |
| M4 | 34.50 |
| M5 | 45.00 |

| End station to destination | |
|---|---|
| **Code** | **Price ($)** |
| F1 | 1.50 |
| F2 | 3.00 |
| F3 | 4.50 |
| F4 | 6.00 |
| F5 | 8.00 |

To book a journey, a passenger will enter a code for each stage and the start time of their journey. The total price is calculated by adding together the price for each of the three stages. The total price will be reduced by 40% when the start time of the journey is after 10:00.

Write and test a program or programs for the integrated transport booking system.
- 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** – setting up the booking system
Write a program to set up arrays to record the following:
- codes and prices for each of the three stages
- passenger accounts that include a unique passenger account number and name
- bookings that include a unique passenger account number, a start time of the journey, a code for each stage of the journey, and a unique booking number for the journey.

Store the data for the code and price for each stage.

**Task 2** – using the booking system
Extend **Task 1** to achieve the following:
- Allow passengers to open an account by generating a unique passenger account number and storing it along with their name in the arrays.
- Allow passengers to make a booking by first entering their unique passenger account number, the start time of their journey, and a code for each stage of their journey. Check if the passenger account number already exists.
- Generate a unique booking number for the journey.
- Calculate the total price of the journey, without any discount, and store the journey details.

**Task 3** – applying a discount and checking the entry
Extend **Task 2** to check the start time of the journey and if it is after 10:00, apply a 40% discount to the total price.
Display the total price and booking details for the passenger to check, and allow them to either confirm the details are correct or start again.

© UCLES 2021

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

**(a) (i)** Identify **one** variable you could have used for **Task 2** and state its use.

Variable .....

Use .....

.....

.....
[2]

**(ii)** Describe the arrays you could have used in **Task 1**. Include the name, data type, use and sample data for each array.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

**(b)** Describe how you could validate the data entry for the input of the codes for the different stages of the journey in **Task 2**.

.....

.....

.....

.....

.....

..... [3]

**(c)** Write an algorithm to show how your program carries out **Task 3**, using either pseudocode, programming statements or a flowchart. Assume **Tasks 1** and **2** have already been completed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

[6]

**(d)** Explain how your program could be changed to count and store the number of bookings made by each passenger. Then, after ten bookings have been made by a passenger, apply an additional 10% discount to every future booking. Any programming statements used in your answer must be fully explained.

........................................................................................................................................ [4]

## Section B

**2** Tick (✓) one box in each row to identify if the statement is about validation, verification or neither.

| Statement | Validation (✓) | Verification (✓) | Neither (✓) |
|---|---|---|---|
| a check where data is re-entered to make sure no errors have been introduced during data entry | | | |
| an automatic check to make sure the data entered has the correct number of characters | | | |
| a check to make sure the data entered is sensible | | | |
| a check to make sure the data entered is correct | | | |

[3]

**3** A program checks that the data entered is between 1 and 100 inclusive.

Identify **one** piece of normal, extreme and erroneous test data for this program, and give a reason for each.

Normal test data .....

Reason .....

.....

.....

Extreme test data .....

Reason .....

.....

.....

Erroneous test data .....

Reason .....

.....

.....

[6]

**4** The pseudocode algorithm should work as a calculator and output the result.

```
1   Continue ← 1
2   WHILE Continue = 0
3     OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /"
4     INPUT Operator
5     OUTPUT "Enter the first value"
6     INPUT Value1
7     OUTPUT "Enter the second value"
8     OUTPUT Value2
9     IF Operator
10      1: Answer ← Value1 + Value2
11      2: Answer ← Value1 - Value2
12      3: Answer ← Value1 * Value2
13      4: Answer ← Value1 / Value2
14    ENDCASE
15    OUTPUT "The answer is ", Value1
16    OUTPUT "Do you wish to enter more values (Yes or No)?"
17    INPUT MoreValues
18    IF MoreValues = "No"
19      THEN
20        Continue ← 1
21    ENDIF
22  UNTIL Continue = 0
```

**(a)** Find the **five** errors in the pseudocode and suggest a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

Error 5 .....

Correction .....

.....

[5]

**(b)** The algorithm needs changing to allow only the numbers 1, 2, 3, or 4 to be entered for the input variable `Operator`.

Write the pseudocode to perform this task and state where in the algorithm it would be located.

Pseudocode .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Location in algorithm .....

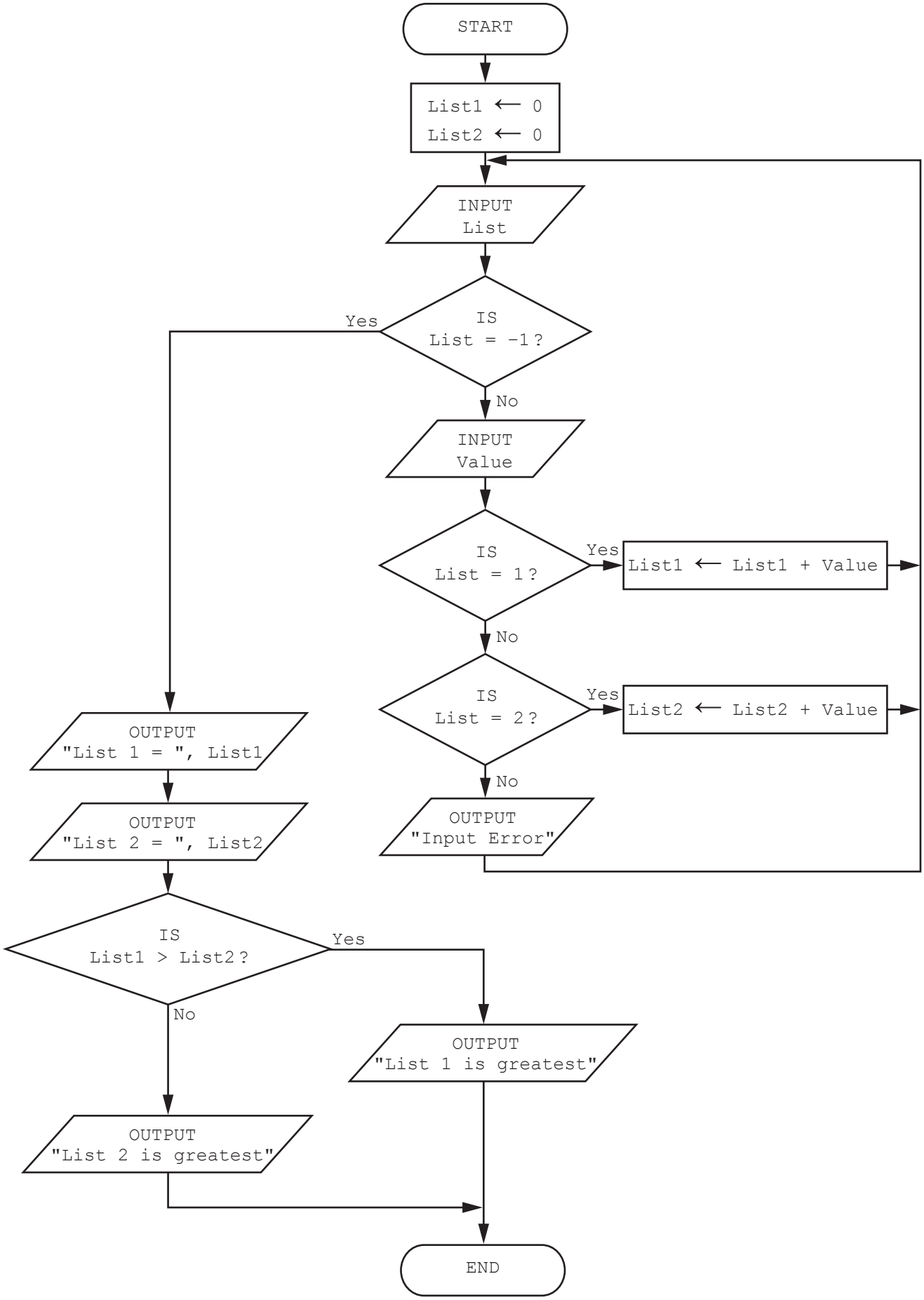
.....

.....

[5]

**5** The flowchart represents an algorithm.

The algorithm will terminate if –1 is entered at the `List` input.



Complete the trace table for the algorithm using this input data:

2, 77, 2, 16, 1, 35, 2, –7, 5, 18, 1, 11, 1, 12, 2, 20, –1, 18

| List | Value | List1 | List2 | OUTPUT |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

[5]

**6** A pet supplier uses the database table, STOCK, to keep records of its products for pets.

The fields are:

| Field name | Description |
|---|---|
| ProductID | code to identify the product |
| ProductName | name of product |
| ProductDescription | information about the product |
| Animal | type of animal the product is for, e.g. cat, bird, horse |
| ProductType | type of product, e.g. food, toy, medicine |
| InStock | whether the product is in stock or **not** |

**(a) (i)** Identify the field that could have a Boolean data type.

..... [1]

**(ii)** Identify the field that should be used as the primary key.

..... [1]

**(b)** Complete the query-by-example grid to output the products intended for a cat that are in stock. Display only the primary key and the name of the products. The output should be sorted by the primary key.

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

[4]

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