Cambridge
O Level

**Cambridge Assessment International Education**
Cambridge Ordinary Level

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

*3555957820*

**COMPUTER SCIENCE** **2210/22**

Paper 2 Problem-solving and Programming **May/June 2019**

**1 hour 45 minutes**

Candidates answer on the Question Paper.

No Additional Materials are required.

No calculators allowed.

**READ THESE INSTRUCTIONS FIRST**

Write your centre number, candidate number and name in the spaces at the top of this page.
Write in dark blue or black pen.
You may use an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.

**DO NOT ATTEMPT TASKS 1, 2 AND 3** in the pre-release material; these are for information only.

You are advised to spend no more than **40 minutes** on **Section A** (Question 1).

No marks will be awarded for using brand names of software packages or hardware.

Any businesses described in this paper are entirely fictitious.

At the end of the examination, fasten all your work securely together.

The number of marks is given in brackets [ ] at the end of each question or part question.

The maximum number of marks is 50.



DC (LEG) 180906
© UCLES 2019

## 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 auction company has an interactive auction board at their sale rooms, which allows buyers to place bids at any time during the auction. Before the auction starts, the sellers place their items in the sale room with a unique number attached to each item (item number). The following details about each item need to be set up on the interactive auction board system: item number, number of bids, description and reserve price. The number of bids is initially set to zero.

During the auction, buyers can look at the items in the sale room and then place a bid on the interactive auction board at the sale room. Each buyer is given a unique number for identification (buyer number). All the buyer needs to do is enter their buyer number, the item number and their bid. Their bid must be greater than any existing bids.

At the end of the auction, the company checks all the items and marks those that have bids greater than the reserve as sold. Any items sold will incur a fee of 10% of the final bid to be paid to the auction company.

Write and test a program or programs for the auction company.

- 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** – Auction set up.

For every item in the auction the item number, description and the reserve price should be recorded. The number of bids is set to zero. There must be at least 10 items in the auction.

**Task 2** – Buyer bids.

A buyer should be able to find an item and view the item number, description and the current highest bid. A buyer can then enter their buyer number and bid, which must be higher than any previously recorded bids. Every time a new bid is recorded the number of bids for that item is increased by one. Buyers can bid for an item many times and they can bid for many items.

**Task 3** – At the end of the auction.

Using the results from TASK 2, identify items that have reached their reserve price, mark them as sold, calculate 10% of the final bid as the auction company fee and add this to the total fee for all sold items. Display this total fee. Display the item number and final bid for all the items with bids that have not reached their reserve price. Display the item number of any items that have received no bids. Display the number of items sold, the number of items that did not meet the reserve price and the number of items with no bids.

© UCLES 2019

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

Describe the data structures you have used in **Task 1** to record the items for sale. Include some sample data for each data structure you have described.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

**(b)** Explain how your program for **Task 1** ensures the item number is unique.

.....

.....

.....

..... [2]

**(c)** Write an algorithm for the part of **Task 2** that allows the buyer to add a new bid, using **either** pseudocode, programming statements **or** a flowchart. Assume that **Task 1** has been completed and that the item details have already been found.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

© UCLES 2019

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

**(d)** Explain how your program for **Task 2** checks that a new bid is higher than previous bids for an item.

.....

.....

.....

.....

.....

..... [3]

© UCLES 2019

**(e)** Explain how your program identifies the items that have reached their reserve price, then calculates and displays the total auction company fee for all sold items as part of **Task 3**. Any programming statements used in your answer must be fully explained.

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

........................................................................................................................................

.................................................................................................................................. [5]

© UCLES 2019

## Section B

**2** **(a)** An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ← 1
INPUT Number
High ← Number
Low ← Count
REPEAT
  INPUT Number
  IF Number > High
    THEN
      High ← Number
  ENDIF
  IF Number > Low
    THEN
      Low ← Number
  ENDIF
  Count ← Count + 1
UNTIL Count > 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1.....

Correction .....

.....

Error 2.....

Correction .....

.....

Error 3.....

Correction .....

.....

Error 4.....

Correction .....

.....

[4]

**(b)** Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable `Total`.

.....

.....

.....

.....

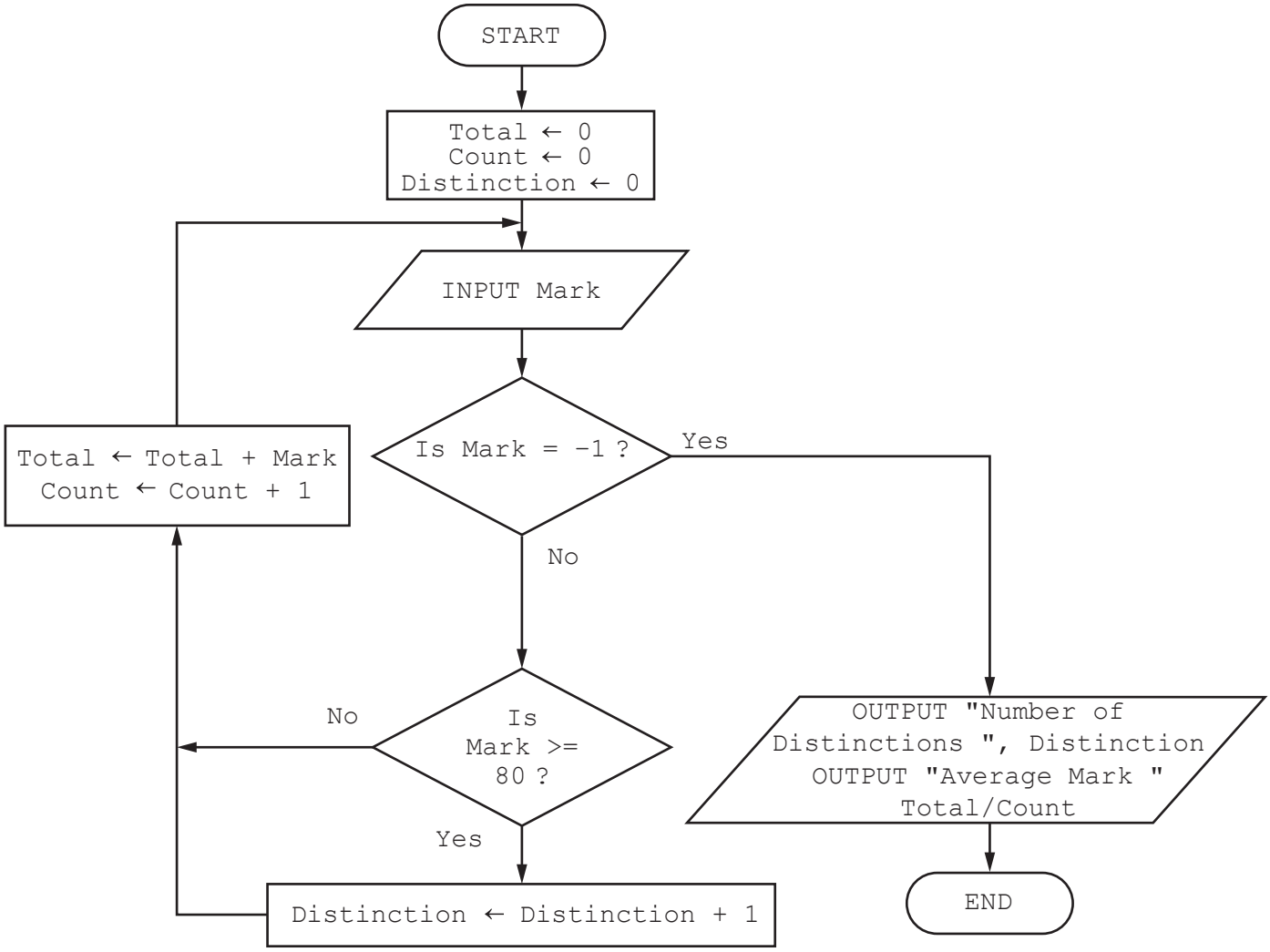
.....

.....

.....

..... [4]

**3** This flowchart inputs the marks gained in an examination. An input of –1 ends the routine.




Complete the trace table for the mark input data: 50, 70, 65, 30, 95, 50, 55, 85, 65, 35, –1, 45

| Total | Count | Distinction | Mark | **OUTPUT** |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

[4]

**4** For each of the **four** groups of statements in the table, place a tick in the correct column to show whether it is an example of **Selection** or **Repetition**.

| Statements | Selection | Repetition |
|---|---|---|
| `FOR A ← 1 TO 100`<br>`  B ← B + 1`<br>`NEXT A` | | |
| `CASE A OF`<br>`  100: B ← A`<br>`  200: C ← A`<br>`ENDCASE` | | |
| `IF A > 100`<br>`THEN`<br>`  B ← A`<br>`ENDIF` | | |
| `REPEAT`<br>`  A ← B * 10`<br>`UNTIL A > 100` | | |

[4]

**5** Explain what is meant by **validation** and **verification**.
Give an example for each one.

Validation .....

.....

.....

.....

Example .....

.....

.....

Verification .....

.....

.....

.....

Example .....

.....

.....

[6]

**6** A database table, FLIGHT, is used to keep a record of flights from a small airfield. Planes can carry passengers, freight or both. Some flights are marked as private and only carry passengers.

| Flight number | Plane | Notes | Departure time | Passengers |
|---|---|---|---|---|
| FN101 | Caravan 1 | Private passenger flight | 08:00 | Y |
| CN101 | Caravan 2 | Freight only | 08:30 | N |
| CN102 | Piper 1 | Freight only | 09:00 | N |
| FN104 | Piper 2 | Passengers only | 09:20 | Y |
| FN105 | Piper 1 | Freight and passengers | 10:00 | Y |
| FN106 | Caravan 1 | Passengers only | 10:30 | Y |
| CN108 | Caravan 2 | Freight only | 08:00 | N |
| CN110 | Lear | Private passenger flight | 08:00 | Y |

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

Field ..... [1]

© UCLES 2019

**(b)** A query-by-example has been written to display just the flight numbers of all planes leaving after 10:00 that only carry passengers.

| | | | | |
|---|---|---|---|---|
| Field: | Flight number | Passengers | Departure time | |
| Table: | FLIGHT | FLIGHT | FLIGHT | |
| Sort: | | | | |
| Show: | ☐ | ☑ | ☐ | ☐ |
| Criteria: | | = Y | = 10:00 | |
| or: | | | | |

Explain why the query-by-example is incorrect, and write a correct query-by-example.

Explanation .....

.....

.....

.....

.....

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

[7]

© UCLES 2019

**BLANK PAGE**

© UCLES 2019

**BLANK PAGE**

© UCLES 2019

**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 2019