Cambridge O Level

**Cambridge Assessment International Education**
Cambridge Ordinary Level

| CANDIDATE NAME | |
|---|---|

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

*5738214450*

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

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

This document consists of **13** printed pages and **3** blank pages.

DC (CE/CT) 189393
© UCLES 2019

Cambridge Assessment International Education

## 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**

A company supplies concrete slabs for paving. The slabs are made to order in batches of 20; all slabs in a batch are identical. Customers can order from a small range of standard sizes and colours. All measurements are given in millimetres. The price is calculated at \$0.05 for a volume of 100 000 $mm^3$ of grey concrete; red and green concrete are charged at 10% more. Customers can choose their own colours; a custom colour has an initial set up cost of \$5 then 15% more than the price for grey.

| Colour of slab |
|---|
| Grey |
| Red |
| Green |
| Custom |

| Depth of slab |
|---|
| 38 |
| 45 |

| Shapes | Sizes for each shape |
|---|---|
| Square | 600 × 600 or 450 × 450 |
| Rectangular | 600 × 700 or 600 × 450 |
| Round | Diameter 300 or 450 |

Write and test a program or programs for the concrete slab 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** – Price for a batch of 20 slabs

Using the information above set up a routine that allows a customer to choose the concrete slab they require and calculate a price for a batch of 20 slabs. The details of the slab chosen and the price for a batch of 20 should be displayed on the screen.

**Task 2** – Customer places an order

Using the information from TASK 1, the customer places an order for the number of slabs they require. Orders for fewer than 20 slabs or more than 100 slabs are not accepted; orders that are not a multiple of 20 are rounded up to the next multiple of 20 slabs. Display the order price and the number of slabs to be produced.

**Task 3** – Flexible pricing

The cost of concrete is variable. The cost for 100 000 $mm^3$ of grey concrete can be input and two grades are available; basic at the cost input and best at 7% more. Use a copy of your program for TASK 1 to develop TASK 3 to input the cost and grade of concrete before calculating the price for 20 slabs.

© UCLES 2019

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

**(i)** Describe the data structures that you have used to store the data for the concrete slabs in **Task 1**, include the name, data type and use for each data structure.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

**(ii)** State **one** variable that you have used in **Task 3**.
Give the data type for the variable. State what it is used for.

Variable name .....

Data type .....

Use .....

.....
[3]

**(b)** Explain how you calculated the volume of a concrete slab in **Task 1**.

.....

.....

.....

.....

.....

..... [3]

**(c)** Write an algorithm for **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. Assume that **Task 1** has been completed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

**(d)** Explain how you changed your program for **Task 1** to meet the requirements for **Task 3**. Include and fully explain any altered or additional programming statements.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

## Section B

**2** An algorithm has been written in pseudocode to select a random number using the function `RandInt(n)`, which returns a whole number between 1 and the argument n. The algorithm then allows the user to guess the number.

```
Number ← RandInt(100)
TotalTry ← 1
REPEAT
  PRINT "Enter your guess now, it must be a whole number"
  INPUT Guess
  IF TotalTry > Number
    THEN
      PRINT "Too large try again"
  ENDIF
  IF Guess > Number
    THEN
      PRINT "Too small try again"
  ENDIF
  TotalTry ← Guess + 1
UNTIL Guess <> Number
TotalTry ← TotalTry - 1
PRINT "Number of guesses ", TotalTry
```

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

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

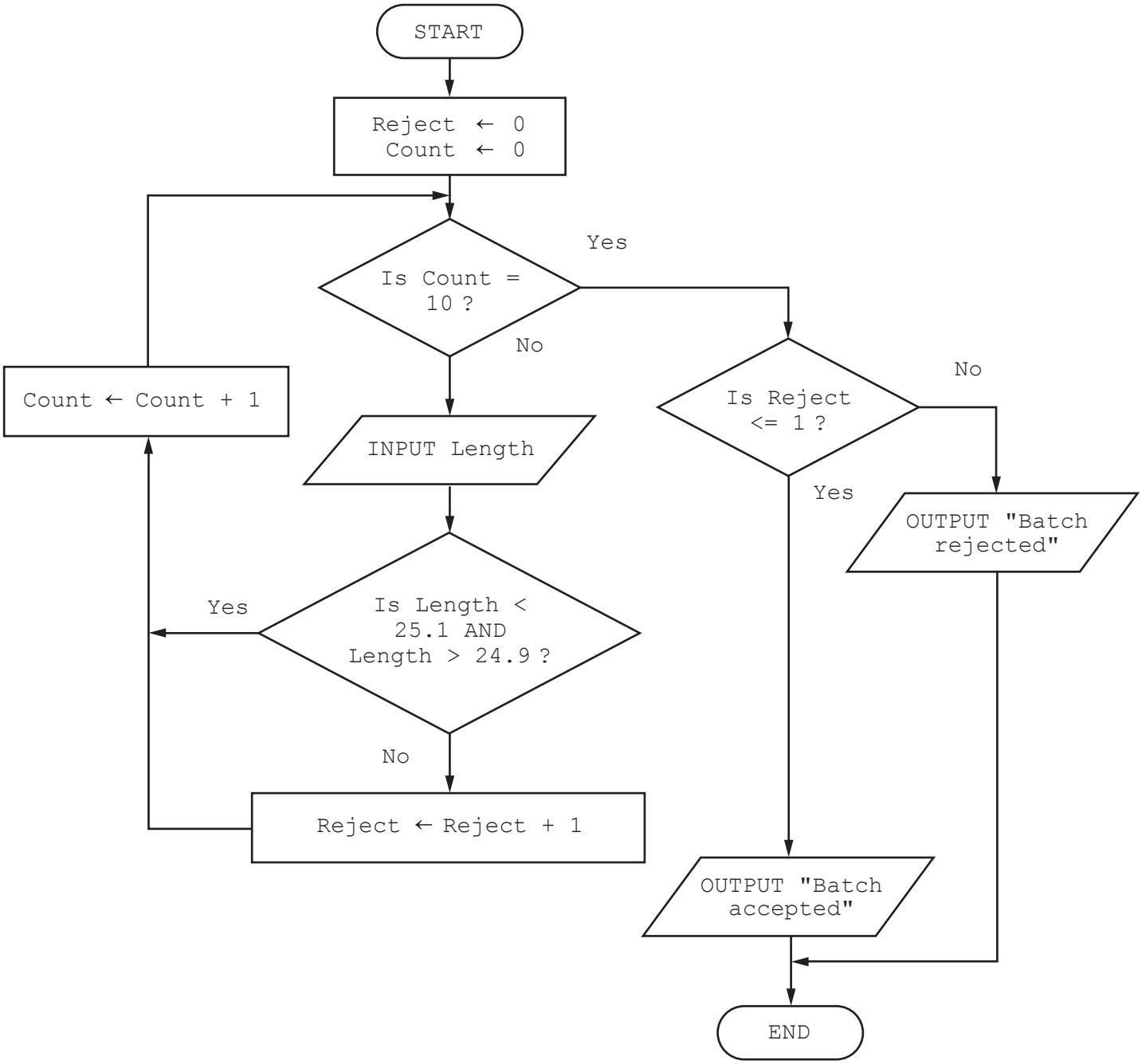
Error 4 .....

Correction .....

.....

[4]

**3** **(a)** The flowchart checks the lengths of a batch of 10 ropes. For the batch to be accepted 90% of the lengths need to be between 24.9 and 25.1 metres.



Complete the trace table for the input data:

24.88, 25.01, 24.98, 25.00, 25.05, 24.99, 24.97, 25.04, 25.19, 25.07

| Reject | Count | Length | OUTPUT |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

[4]

**(b)** **(i)** It has been decided to only reject batches of rope that contain ropes that are too short.

State the change required to the algorithm.

.....

..... [1]

**(ii)** Explain how the algorithm to reject batches could be improved to make it more effective.

.....

.....

.....

..... [2]

**4** **Four** validation checks and **four** descriptions are shown.

Draw a line to connect each validation check to the correct description.

| Validation Check | Description |
|---|---|
| Range check | Checks that some data is entered. |
| Presence check | Checks for a maximum number of characters in the data entered. |
| Length check | Checks that the characters entered are all numbers. |
| Type check | Checks that the value entered is between an upper value and a lower value. |

[3]

**5** A programmer writes a program to weigh baskets of fruit in grams, keeping a total of the weight and counting the number of baskets. The total weight is stored in a variable `Total` and the number of baskets is stored in a variable `BasketCount`.

Explain, including examples of programming statements, how totalling and counting could be used in this program.

Totalling .....

.....

.....

.....

.....

.....

Counting .....

.....

.....

.....

.....

.....

[4]

**6** Explain why constants, variables and arrays are used in programming.

Constants .....

.....

.....

.....

Variables .....

.....

.....

.....

Arrays .....

.....

.....

.....

[6]

**7** A database table, SALES, is used to keep a record of items made and sold by a furniture maker.

| Item number | Order number | Notes | Amount | Status |
|---|---|---|---|---|
| CH001 | 1921 | Smith – six dining chairs | 6 | Delivered |
| TB003 | 1921 | Smith – large table | 1 | In progress |
| CH001 | 1924 | Hue – extra chairs | 4 | In progress |
| CH003 | 1925 | For stock | 2 | Cancelled |
| BN001 | 1927 | Patel – replacement bench | 1 | Not started |
| ST002 | 1931 | Sola – small table | 1 | Delivered |
| CH003 | 1927 | Patel – eight dining chairs with arms | 8 | Not started |
| TB003 | 1927 | Patel – large table | 1 | Not started |

**(a)** Explain why the field **Item number** could not be used as a primary key.

.....

..... [1]

**(b)** A query-by-example has been written to display only the order number and item numbers of any items in progress or not started.

| | | | | |
|---|---|---|---|---|
| Field: | Item number | Order number | Amount | Status |
| Table: | SALES | SALES | SALES | SALES |
| Sort: | | | | |
| Show: | ☐ | ☑ | ☑ | ☐ |
| Criteria: | | | | Not Like "Delivered" |
| or: | | | | |

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

Explanation .....

.....

.....

.....

.....

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

[5]

**BLANK PAGE**

© UCLES 2019

**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 2019