Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

0442216027

COMPUTER SCIENCE **2210/23**

Paper 2 Problem-solving and Programming **October/November 2020**

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. Blank pages are indicated.

DC (KS) 205000
© UCLES 2020

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 baguette ordering service allows customers to order filled baguettes. There are two sizes of baguette: 30 cm and 15 cm. Baguettes are available as white, brown or seeded bread. Baguettes have one filling and can have up to three salad items added.

Filling and salad choices are:

Filling	Salad
Beef Chicken Cheese Egg Tuna Turkey	Lettuce Tomato Sweetcorn Cucumber Peppers

Customers choose their baguette options. They then confirm their order, alter their choices or choose not to proceed.

Each day the ordering service calculates the number of each:
- size of baguette sold
- type of bread (white, brown or seeded) sold
- filling sold.

Write and test a program or programs for the baguette ordering service.
- 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 – Ordering.

Customers are given choices on size, type of bread, filling and salad items as shown. Only valid choices can be accepted. The customer is asked to confirm their order, alter their choices or choose not to proceed. If the customer confirms their order they are given a unique order number. Display the baguette ordered and the order number.

Task 2 – Recording the choices.

Extend TASK 1 to record totals for the size, types of bread and fillings sold that day and calculate the total number of baguettes sold that day.

Task 3 – Finding the most and least popular baguette fillings.

Using your results from TASK 2, display the most popular and least popular fillings as a percentage of the total number of baguettes sold that day.

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

(a) Identify **one** constant and identify **one** variable that you could have used for **Task 1**. Give the value that would be assigned to the constant. Give the data type for the variable. State what each one could be used for.

Constant .....

Value .....

Use .....

.....

Variable .....

Data type .....

Use .....

.....

[6]

(b) **(i)** Write an algorithm to allow a customer to choose the filling and salad items for their baguette (part of **Task 1**), using **either** pseudocode, programming statements **or** a flowchart. Your algorithm must only include this part of **Task 1**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Filing:

① Create a file and save 10 values in integer to the file. Values must be input from user.

```
DECLARE Value, i : INTEGER.

OPENFILE "values.txt" FOR WRITE
FOR i ← 1 TO 10
    INPUT Value
    WRITEFILE "values.txt", Value
NEXT i
CLOSEFILE "values.txt"
```

What is a single change, to append more values to existing file, is required?

We need a different mode to open file in it. That should not be WRITE.

Q. OUTPUT "Ten" Lines saved in a file "XYZ.txt"

```
OPENFILE "XYZ.txt" FOR READ
FOR Count ← 1 TO 10
    READFILE "XYZ.txt", myLine
    OUTPUT myLine
NEXT
CLOSEFILE "XYZ.txt"
```

① WRITING TO THE FILE

② READING FROM THE FILE [5]

(ii) Explain how your algorithm in **part (b)(i)** ensured that only valid choices were accepted for the filling.

..... [3]

(c) Explain how you would need to change your program for **Task 1** if there were three sizes of baguette to choose from (15 cm, 20 cm and 30 cm).

..... [2]

(d) Explain how your program completed **Task 3**. Assume that **Task 2** has been completed. Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

NAND/NOR LOGIC

Section B starts on Page 8.

Section B

2 An algorithm has been written in pseudocode to check the temperature readings taken from a freezer are within the range –18 degrees to –25 degrees inclusive.

The algorithm counts the number of times that the temperature reading is below –25 degrees and the number of times that the temperature reading is above –18 degrees.

An engineer is called if there are more than 10 temperature readings below –25 degrees.

An alarm sounds if there are more than 5 temperature readings above –18 degrees.

```
01  TooHot ← 0
02  TooCold ← 1000
03  REPEAT
04      OUTPUT "Please enter temperature"
05      INPUT Temperature
06      IF Temperature < -25
07        THEN
08          TooCold ← TooCold - 1
09      ENDIF
10      IF Temperature > -18
11        THEN
12          TooHot ← TooHot + 1
13      ENDIF
14  UNTIL TooHot > 5 OR TooCold > 10
15  IF TooHot < 5
16    THEN
17      INPUT "Alarm!!"
18  ENDIF
19  IF TooCold > 10
20    THEN
21      OUTPUT "Call the Engineer"
22  ENDIF
```

(a) Give the line number(s) from the algorithm of:

an assignment statement .....

a loop .....

a counting statement .....

a selection statement .....

[4]

(b) Give line numbers where the **four** errors are to be found in the pseudocode. Suggest a correction for each error.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

Error 4 line number .....

Correction .....

.....
[4]

(c) Explain how you could extend the algorithm to count the number of times the temperature readings are within the range –18 degrees to –25 degrees inclusive.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

3 Four programming concepts and five descriptions are shown.

Draw a line to connect each **Programming concept** to its correct **Description**. Not all Descriptions will be connected to a Programming concept.

Programming concept	**Description**
Validation	A subroutine that does not have to return a value
Verification	An automatic check to ensure that data input is reasonable and sensible
Procedure	A subroutine that always returns a value
Function	An overview of a program or subroutine
	A check to ensure that data input matches the original

Structure Diagram.

[4]

```
//Main Program
IF MySize = Default
  THEN
    CALL DefaultLine()
  ELSE
    CALL Line(MySize)
ENDIF

PROCEDURE DefaultLine()
   CALL LINE(60)
ENDPROCEDURE

PROCEDURE Line(Size : INTEGER)
   DECLARE Length : INTEGER
   FOR Length ← 1 TO Size
     OUTPUT '-'
   NEXT Length
ENDPROCEDURE
```

Question 4 starts on Page 12.

4 This flowchart represents an algorithm that allows the input of two numbers and performs a calculation.

The predefined function MOD finds the remainder from integer division for example `X ← 8 MOD 5` gives `X` a value of `3`.

(a) Complete a trace table for this set of input data:
11, 4, 6, 2, 3, 9, 3, 2, 2, 6, 0, 0, 1, 1

Y	Z	A	OUTPUT

[4]

(b) Explain the purpose of this algorithm.

.....

.....

.....

..... [2]

5 A marine wildlife rescue centre uses a database table, MARINE, to keep records of its creatures.

	Creature	Class	Quantity	Ready for release	Offspring
1	Manta Ray	Fish	3	Y	N
2	Short-tailed Albatross	Bird	4	Y	N
3	Emperor Penguin	Bird	50	Y	Y
4	Bluefin Tuna	Fish	2	N	N
5	Manatee	Mammal	4	Y	N
6	Hawksbill Turtle	Reptile	10	Y	Y
7	Hammerhead Shark	Fish	3	Y	N
8	Yellow-eyed Penguin	Bird	4	Y	N
9	Kemp's Ridley Sea Turtle	Reptile	1	Y	N

(a) State how many fields and how many records are shown in this table.

Number of fields 5

Number of records ..... 9 [2]

(b) Show the output that would be given by this query-by-example. QBE → NO more in syllabus

Field:	Creature	Class	Ready for release
Table:	MARINE	MARINE	MARINE
Sort:			
Show:	✓	☐	✓
Criteria:		="Bird"	
or:			

.....

.....

..... [2]

(c) Complete the query-by-example grid to display the creatures, in ascending order of quantity, that have no offspring and are ready for release. Display only the creature field.

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

[4]

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 2020

Cambridge O Level

Cambridge Assessment International Education
Cambridge Ordinary Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

4050284988

COMPUTER SCIENCE **2210/21**

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.

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

© 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

The local high school uses buses to transport students to school. There are six bus routes labelled A to F. You have conducted a survey to analyse the punctuality statistics of these buses over a four-week period. The data from the survey are shown in the table:

	Punctuality table					
Day	**Bus A**	**Bus B**	**Bus C**	**Bus D**	**Bus E**	**Bus F**
Mon1	0	0	2	1	–1	0
Tue1	0	1	0	0	–1	–5
Wed1	0	0	–1	0	–1	–5
Thu1	2	0	–1	0	–2	–5
Fri1	2	1	–2	0	–4	–4
Mon2	4	2	–2	0	–10	–3
Tue2	0	0	–3	0	–2	–5
Wed2	3	0	–1	0	0	0
Thu2	4	0	0	0	0	0
Fri2	–2	0	0	0	0	0
Mon3	–5	1	–2	2	0	0
Tue3	0	0	0	0	1	–2
Wed3	0	0	1	0	2	–3
Thu3	3	0	1	0	–3	1
Fri3	4	2	1	0	1	1
Mon4	–1	0	1	0	1	1
Tue4	8	0	–1	0	3	0
Wed4	1	1	–1	0	–1	0
Thu4	1	0	2	0	0	–2
Fri4	–2	0	–2	0	0	–5

Positive numbers represent minutes early, negative numbers represent minutes late and 0 represents the bus having been on time.

Write and test a program or programs for the local high school.

- 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 data storage.
Using arrays set up a system to enable data for each bus route to be entered covering each day of a four-week period. It must be possible to enter the data supplied or your own set of data, using suitable prompts as necessary.

Task 2 – Working out the statistics.
Extend your program so that the following statistics for the four-week period may be calculated and output:
- the number of late arrivals for each bus route
- the average number of minutes late for each bus route
- the bus route with the highest number of days on which it was late
- the average number of minutes late for each bus route, using only data from days on which it was late

All the results should be displayed with appropriate annotation.

Task 3 – Checking specific days.
Extend the program as follows:
- Allow the user to input a specific day, for example Fri3, to be used for analysis of data.
- Find and display how many buses were late on this particular day.
- For each late bus, display the route label and how late the bus was on this particular day.

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

(i) State **one** array you used for **Task 1**. State the data type and purpose of this array.

Array .....

Data type .....

Purpose .....

.....

.....
[3]

(ii) State **one** variable you used for **Task 2** and **one** variable you used for **Task 3**. In each case, state the data type and purpose of the variable.

Task 2 variable name .....

Data type .....

Purpose .....

.....

.....

Task 3 variable name .....

Data type .....

Purpose .....

.....

.....
[6]

(b) Write an algorithm to show how you calculated and output the number of late arrivals and the average number of minutes late for each bus route (part of **Task 2**), using **either** pseudocode, programming statements **or** a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(c) Explain how your program uses the input in **Task 3** to only find the data for that specific day (part of Task 3). Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(d) Explain how you would alter your program in **Task 1** to allow you to choose the number of weeks to enter data on bus arrival times.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [2]

Section B

2 Describe each of the following data types used in programming. In each case, give an example of a piece of data to illustrate your answer. Each example must be different.

Char

Any letter, digit or symbol, that can be printed, over the keyboard.
E.g: 'a', '@', '#', '9', 'D'

String

Sequence of characters
E.g: "Zafar"

Boolean

A value that can only be TRUE or FALSE
isVisited = FALSE.

[6]

3 **(a)** Give an example of a conditional statement using pseudocode.

[2]

(b) Describe the purpose of a conditional statement.

[2]

Question 4 starts on page 10.

4 This section of program code may be used as a validation check.

```
PRINT "Input a value between 0 and 100 inclusive"
INPUT Value
WHILE Value < 0 OR Value > 100
  PRINT "Invalid value, try again"
  INPUT Value
ENDWHILE
PRINT "Accepted: ", Value
```

boundry boundry
-1 0 1 — 99 100 101
Normal
Extreme
Abnormal

(a) Give a name for this type of validation check.

Range Check ..... [1]

(b) Describe what is happening in this validation check.

..... [2]

abnormal boundry

(c) Complete the trace table for this program code using the test data: 200, 300, –1, 50, 60

Normal

Value	OUTPUT
200	Invalid value, try again
300	Invalid value, try again.
-1	" " " "
50	Accepted 50
60	Accepted 60

[3]

(d) Draw a flowchart to represent this section of program code.

[5]

5 The table, BEVERAGES, shows the number of calories in 100 ml of a range of popular beverages. It also shows the availability of these drinks in a can, a small bottle and a large bottle.

BevNo	BevName	Calories	Can	Small Bottle	Large Bottle
Bev01	Cola	40	Yes	Yes	Yes
Bev02	Lime	45	Yes	No	Yes
Bev03	Energy Drink 1	52	Yes	Yes	No
Bev04	Energy Drink 2	43	Yes	No	No
Bev05	Mango	47	Yes	No	Yes
Bev06	Lemon Iced Tea	38	Yes	No	Yes
Bev07	Lemonade	58	Yes	Yes	Yes
Bev08	Orange Juice	46	Yes	Yes	No
Bev12	Apple Juice	50	Yes	Yes	No
Bev15	Chocolate Milk	83	Yes	Yes	No

(a) Give a reason for choosing BevNo as the primary key for this table.

.....

..... [1]

(b) State the number of records shown in the table BEVERAGES.

..... [1]

© UCLES 2019

(c) List the output that would be given by this query-by-example.

Field:	BevNo	BevName	Can	Small Bottle	Large Bottle	
Table:	BEVERAGES	BEVERAGES	BEVERAGES	BEVERAGES	BEVERAGES	
Sort:		Descending				
Show:	☑	☑	☐	☐	☐	☐
Criteria:			= "Yes"	= "Yes"	= "Yes"	
or:						

.....

.....

.....

..... [3]

(d) Complete the query-by-example grid to output a list showing just the names and primary keys of all the beverages with a calorie count greater than 45. The list should be in alphabetical order of names.

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

[4]

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

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER					

2180258245

COMPUTER SCIENCE **2210/22**

Paper 2 Algorithms, Programming and Logic **May/June 2023**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- 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 75.
- 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.

DC (PQ) 338137/1
© UCLES 2023

1 Tick (✓) **one** box to identify the first stage of the program development life cycle.

A Analysis ☐

B Coding ☐

C Design ☐

D Testing ☐

[1]

2 **Four** logic gates and **five** standard symbols for logic gates are shown.

Draw **one** line to link each logic gate to its standard symbol. **Not** all standard symbols will be used.

[4]

3 Identify **three** different ways that the design of a solution to a problem can be presented.

1 Flowcharts

2 pseudocodes

3 structure digram

[3]

4 A program needs to make sure the value input for a measurement meets the following rules:

- the value is a positive number
- a value is always input
- the value is less than 1000.

(a) Describe the validation checks that the programmer would need to use.

.....

.....

.....

.....

.....

..... [3]

(b) The program needs editing to include a double entry check for the value input.

(i) State why this check needs to be included.

.....

..... [1]

(ii) The input value needs to be stored in the variable `Measurement`
Write pseudocode to perform the double entry check until a successful input is made.

.....

.....

.....

.....

.....

.....

.....

..... [3]

5 Due to an issue with Question 5, the question has been removed from the question paper.

6 State **three** different features of a high-level programming language that a programmer could use to make sure that their program will be easier to understand by another programmer.
Give an example for each feature.

Feature 1 .....

.....

Example .....

.....

Feature 2 .....

.....

Example .....

.....

Feature 3 .....

.....

Example .....

.....

[6]

© UCLES 2023

BLANK PAGE

7 An algorithm has been written in pseudocode to calculate a check digit for a four-digit number. The algorithm then outputs the five-digit number including the check digit.
The algorithm stops when –1 is input as the fourth digit.

```
01 Flag ← FALSE
02 REPEAT
03     Total ← 0
04     FOR Counter ← 1 TO 4
05       OUTPUT "Enter a digit ", Counter
06       INPUT Number[Counter]
07       Total ← Total + Number * Counter
08       IF Number[Counter] = 0
09         THEN
10           Flag ← TRUE
11       ENDIF
12     NEXT Counter
13     IF NOT Flag
14       THEN
15         Number[5] ← MOD(Total, 10)
16         FOR Counter ← 0 TO 5
17             OUTPUT Number[Counter]
18         NEXT
19     ENDIF
20 UNTIL Flag
```

(a) Give the line number(s) for the statements showing:

Totalling .....

Count-controlled loop .....

Post-condition loop .....

[3]

(b) Identify the **three** errors in the pseudocode and suggest a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

[3]

(c) The algorithm does **not** check that each input is a single digit.
Identify the place in the algorithm where this check should occur.
Write pseudocode for this check.
Your pseudocode must make sure that the input is a single digit and checks for `-1`

Place in algorithm .....

Pseudocode .....

.....

.....

.....

.....

.....

[4]

8 Consider this logic expression.

$2^3 = 8$ 0 7

X = (**A** OR **B**) AND (NOT **B** AND **C**)

Complete the truth table for this logic expression.

	4	2	1		
	A	**B**	**C**	**Working space** (A+B) (B'C)	**X**
0	0	0	0	0 0	0
1	0	0	1	0 1	0
2	0	1	0	1 0	0
3	0	1	1	1 0	0
4	1	0	0	1 0	0
5	1	0	1	1 1	1
6	1	1	0	1 0	0
7	1	1	1	1 0	0

[4]

9 This flowchart represents an algorithm.

(a) The array `X[1:5]` used in the flowchart contains this data:

X[1]	X[2]	X[3]	X[4]	X[5]
10	1	5	7	11

Complete the trace table by using the data given in the array.

F	C	X[1]	X[2]	X[3]	X[4]	X[5]	T
		10	1	5	7	11	

[5]

(b) Describe what the algorithm represented by the flowchart is doing.

.....

.....

.....

..... [2]

10 A music streaming service has a new database table named `Songs` to store details of songs available for streaming. The table contains the fields:

- PK `SongNumber` – the catalogue number, for example AG123 Alphanumeric / Text
- `Title` – the title of the song Text
- `Author` – the name of the song writer(s) Text
- `Singer` – the name of the singer(s) Text
- `Genre` – the type of music, for example rock Text
- `Minutes` – the length of the song in minutes, for example 3.75 REAL
- `Recorded` – the date the song was recorded. Date/Time

(a) Identify the field that will be the most appropriate primary key for this table.

— SongNumber — [1]

(b) Complete the table to identify the most appropriate data type for the fields in `Songs`

Field	Data type
`SongNumber`	
`Title`	
`Recorded`	
`Minutes`	

[2]

(c) Explain the purpose of the structured query language (SQL) statements.

```
① SUM (Minutes) FROM Songs WHERE Genre = "rock";
② COUNT (Title) FROM Songs WHERE Genre = "rock";
```

① Calculates the total length of all songs in minutes that belong to the "rock" genre. It adds up the values in the "Minutes" field for all records where the genre is "rock".

② Counts the no. of songs that belong to genre "rock". It returns the total no. of records where genre is "rock". [3]

11 The variables `P` and `Q` are used to store data in a program. `P` stores a string. `Q` stores a character.

(a) Write pseudocode statements to declare the variables `P` and `Q`, store `"The world"` in `P` and store `'W'` in `Q`

```
DECLARE P : STRING
DECLARE Q : CHAR
P ← "The world"
Q ← 'W'
```

[2]

(b) Write a pseudocode algorithm to:

- convert `P` to upper case
- find the position of `Q` in the string `P` (the first character in this string is in position 1)
- store the position of `Q` in the variable `Position`

```
P ← UCASE(P)
Position ← 0
FOR i ← 1 TO LENGTH(P)
    IF SUBSTRING(P, i, 1) = Q Then Position ← i
NEXT i
```

P	Q	i	LENGTH(P)
The world	W	1	9
THE WORLD			

i	SUBSTRING(P,i,1)
1	T
2	H
3	E
4	
5	W

Position

[4]

(c) Give the value of `Position` after the algorithm has been executed with the data in **question 11(a)**.

5

[1]

12 A two-dimensional (2D) array `Account[]` contains account holders' names and passwords for a banking program.

A 2D array `AccDetails[]` has three columns containing the following details:

- column one stores the balance – the amount of money in the account, for example 250.00
- column two stores the overdraft limit – the maximum total amount an account holder can borrow from the bank after the account balance reaches 0.00, for example 100.00
- column three stores the withdrawal limit – the amount of money that can be withdrawn at one time, for example 200.00

The amount of money in a bank account can be negative (overdrawn) but **not** by more than the overdraft limit.
For example, an account with an overdraft limit of 100.00 must have a balance that is greater than or equal to –100.00

Suitable error messages must be displayed if a withdrawal cannot take place, for example if the overdraft limit or the size of withdrawal is exceeded.

The bank account ID gives the index of each account holder's data held in the two arrays.
For example, account ID 20's details would be held in:
`Account[20,1]` and `Account[20,2]`
`AccDetails[20,1]` `AccDetails[20,2]` and `AccDetails[20,3]`

The variable `Size` contains the number of accounts.

The arrays and variable `Size` have already been set up and the data stored.

Write a program that meets the following requirements:

- checks the account ID exists and the name and password entered by the account holder match the name and password stored in `Account[]` before any action can take place
- displays a menu showing the four actions available for the account holder to choose from:
    1. display balance
    2. withdraw money
    3. deposit money
    4. exit
- allows an action to be chosen and completed. Each action is completed by a procedure with a parameter of the account ID.

You must use pseudocode or program code **and** add comments to explain how your code works. All inputs and outputs must contain suitable messages.

You only need to declare any local arrays and local variables that you use.

You do **not** need to declare and initialise the data in the global arrays `Account[]` and `AccDetails[]` and the variable `Size`

.....

.....

.....

.....

.....

.....

..... [15]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2023

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

0258869424

COMPUTER SCIENCE **2210/21**

Paper 2 Algorithms, Programming and Logic **May/June 2023**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- 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 75.
- 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.

DC (DE) 326841
© UCLES 2023

1 **(a)** **Four** descriptions of stages in the program development life cycle are shown.

Draw **one** line to link each description to its most appropriate program development life cycle stage.

Not all program development life cycle stages will be used.

Program development life cycle description	**Program development life cycle stage**
develop an algorithm to solve the problem by using structure diagrams, flowcharts or pseudocode	analysis
detect and fix the errors in the program	coding
identify the problem and its requirements	design
write and implement the instructions to solve the problem	evaluation
	testing

[4]

(b) Identify **three** of the component parts after a problem has been decomposed.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

2 Tick (✓) **one** box to show the name of the data structure used to store a collection of data of the same data type.

A Array ☐

B Constant ☐

C Function ☐

D Variable ☐

[1]

3 **(a)** Describe what is meant by data validation.

.....

.....

.....

..... [2]

(b) A validation check is used to make sure that any value that is input is an integer between 30 and 200 inclusive.

Give **one** example of each type of test data to check that the validation check is working as intended. Each example of test data must be different.

Give a reason for each of your choices of test data.

Normal test data .....

Reason .....

.....

Abnormal test data .....

Reason .....

.....

Extreme test data .....

Reason .....

.....

[6]

4 Explain the purpose of the library routines `DIV` and `ROUND`

`DIV` .....

.....

.....

.....

`ROUND` .....

.....

.....

.....

[4]

5 An algorithm has been written in pseudocode to allow some numbers to be input. All the positive numbers that are input are totalled and this total is output at the end.
An input of `0` stops the algorithm.

```
Exit ← 1
WHILE Exit <> 0 DO
    INPUT Number
    IF Number < 0
      THEN
        Total ← Total + Number
      ELSE
        IF Number = 0
          THEN
            Exit ← 1
        ENDIF
    ENDIF
ENDIF
OUTPUT "The total value of your numbers is ", Number
```

(a) Identify 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) Describe how you could change the corrected algorithm to record and output how many positive numbers have been included in the final total.

You do **not** need to rewrite the algorithm.

.....

.....

.....

.....

.....

.....

.....

..... [4]

6 State **two** features that should be included to create a maintainable program.

Give a reason why each feature should be used.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

7 The flowchart represents an algorithm.

The table represents the two-dimensional (2D) array `Word[]` which stores the first half of the phonetic alphabet used for radio transmission. For example, `Word[10,1]` is 'J'.

Index	1	2
1	A	Alpha
2	B	Bravo
3	C	Charlie
4	D	Delta
5	E	Echo
6	F	Foxtrot
7	G	Golf
8	H	Hotel
9	I	India
10	J	Juliet
11	K	Kilo
12	L	Lima
13	M	Mike

(a) Complete the trace table for the algorithm by using the input data: F, Y, D, N

Pointer	Letter	Choice	OUTPUT

[4]

(b) Identify the type of algorithm used.

.....

..... [1]

(c) Describe **one** problem that could occur with this algorithm if an invalid character was input.

.....

.....

.....

..... [2]

BLANK PAGE

8 The function `LENGTH(Phrase)` calculates the length of a string `Phrase`

(a) Write the pseudocode statements to:

- store the string `"The beginning is the most important part"` in `Phrase`
- calculate and output the length of the string
- output the string in upper case.

Phrase ← "~~~~~~~~~~~~~~~~~~~~"

OUTPUT LENGTH(Phrase)

OUTPUT UCASE(Phrase)

[3]

(b) Write the output your pseudocode should produce.

40

THE BEGINNING IS THE ~~~~ PART

[2]

9 Consider this logic expression.

Z = (NOT **A** OR **B**) AND (**B** XOR **C**)

(a) Draw a logic circuit for this logic expression.

Each logic gate must have a maximum of **two** inputs.

Do **not** simplify this logic expression.

[4]

(b) Complete the truth table from the given logic expression.

A A'	B	C	Working space (A'+B)	Working space (B⊕C)	Z
0 1	0	0	1	0	0
0 1	0	1	1	1	1
0 1	1	0	1	1	1
0 1	1	1	1	0	0
1 0	0	0	0	0	0
1 0	0	1	0	1	0
1 0	1	0	1	1	1
1 0	1	1	1	0	0

[4]

10 A database table called `TVRange` shows the main features and prices of a range of televisions.

Text	INT	Boolean	Boolean	Boolean	Real.
TVCode	**ScreenSize**	**Satellite**	**SmartTV**	**SoundBar**	**Price$**
TV90SaSmSd	90	YES	YES	YES	9750.00
TV75SaSmSd	75	YES	YES	YES	8500.00
TV75SaSd	75	YES	NO	YES	8000.00
TV65SaSmSd	65	YES	YES	YES	6000.00
TV65SmSd	65	NO	YES	YES	5000.00
TV65SaSd	65	YES	NO	YES	5000.00
TV55SaSmSd	55	YES	YES	YES	4000.00
TV55SaSd	55	YES	NO	YES	3500.00
TV55SmSd	55	NO	YES	YES	3500.00
TV50SaSmSd	50	YES	YES	YES	2500.00
TV50Sa	50	YES	NO	NO	1750.00
TV50Sm	50	NO	YES	NO	1750.00
TV40Sa	40	YES	NO	NO	1200.00
TV40	40	NO	NO	NO	950.00
TV32	32	NO	NO	NO	650.00

(a) Give the name of the field that is most suitable to be the primary key.

State the reason for this choice.

Field TVCode

Reason It uniquely identifies each record.

[2]

(b) The database uses the data types:

- text
- character
- Boolean
- integer
- real
- date/time.

Complete the table to show the most appropriate data type for each field.
Each data type must be different.

Field	Data type
TVCode	
ScreenSize	
SmartTV	
Price$	

[2]

(c) Complete the structured query language (SQL) query to return the television (TV) code, screen size and price of all Smart TVs in the database table.

```
SELECT TVCode, ....ScreenSize....., ....Price$.....
....FROM..... TVRange
WHERE SmartTV = ....TRUE.....;
```

[4]

Yes.

11 A one-dimensional (1D) array `Days[]` contains the names of the days of the week. A two-dimensional (2D) array `Readings[]` is used to store 24 temperature readings, taken once an hour, for each of the seven days of the week. A 1D array `AverageTemp[]` is used to store the average temperature for each day of the week.

The position of any day's data is the same in all three arrays. For example, if Wednesday is in index 4 of `Days[]`, Wednesday's temperature readings are in index 4 of `Readings[]` and Wednesday's average temperature is in index 4 of `AverageTemp[]`

The temperature readings are in Celsius to one decimal place. Temperatures can only be from –20.0 °C to +50.0 °C inclusive.

Write a program that meets the following requirements:

- input and validate the hourly temperatures for one week
- calculate and store the average temperature for each day of the week
- calculate the average temperature for the whole week
- convert all the average temperatures from Celsius to Fahrenheit by using the formula Fahrenheit = Celsius * 9 / 5 + 32
- output the average temperature in Celsius and in Fahrenheit for each day
- output the overall average temperature in Celsius and in Fahrenheit for the whole week.

You must use pseudocode or program code **and** add comments to explain how your code works.

You do **not** need to declare any arrays, variables or constants; you may assume that this has already been done.

All inputs and outputs must contain suitable messages.

All data output must be rounded to one decimal place.

You will need to initialise and populate the array `Days[]` at the start of the program.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [15]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2023

Cambridge O Level

Cambridge Assessment International Education
Cambridge Ordinary Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

3559578 20

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.

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

DC (LEG) 193951/1
© 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

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.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

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

.....

.....

.....

.....

.....

..... [3]

(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]

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`
`  B ← B + 1`
`NEXT A`
		✓

`CASE A OF`
`  100: B ← A`
`  200: C ← A`
`ENDCASE`
	✓	

`IF A > 100`
`THEN`
`  B ← A`
`ENDIF`
	✓	

`REPEAT`
`  A ← B * 10`
`UNTIL A > 100`
		✓

[4]

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

Validation is the process of checking whether data meets certain predefined criteria & rules. It ensures that the data entered is sensible, reasonable, and within acceptable Boundaries.

Example - Checking if a date entered is in correct format E.g: DD-MM-YYYY.
- Ensuring a text field is not left empty.

Verification is the process of ensuring that data accurately reflects the intended input or specifications. It involves cross-checking of data from credible source to conform its correctness and consistency.

Example Double entry: data in complete entered twice & computer checks if both entries are the same.
Visual check: after entry data is checked against the data entry form visually.

[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]

(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]

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

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

0182972462

COMPUTER SCIENCE **2210/21**

Paper 2 Problem-solving and Programming **May/June 2020**

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. Blank pages are indicated.

DC (JP) 198770
© UCLES 2020

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 shop sells a range of mobile devices, SIM cards and accessories as shown in the table:

Category	Item code	Description	Price ($)
Phone	BPCM	Compact	29.99
Phone	BPSH	Clam Shell	49.99
Phone	RPSS	RoboPhone – 5-inch screen and 64 GB memory	199.99
Phone	RPLL	RoboPhone – 6-inch screen and 256 GB memory	499.99
Phone	YPLS	Y-Phone Standard – 6-inch screen and 64 GB memory	549.99
Phone	YPLL	Y-Phone Deluxe – 6-inch screen and 256 GB memory	649.99
Tablet	RTMS	RoboTab – 8-inch screen and 64 GB memory	149.99
Tablet	RTLM	RoboTab – 10-inch screen and 128 GB memory	299.99
Tablet	YTLM	Y-Tab Standard – 10-inch screen and 128 GB memory	499.99
Tablet	YTLL	Y-Tab Deluxe – 10-inch screen and 256 GB memory	599.99
SIM card	SMNO	SIM Free (no SIM card purchased)	0.00
SIM card	SMPG	Pay As You Go (SIM card purchased)	9.99
Case	CSST	Standard	0.00
Case	CSLX	Luxury	50.00
Charger	CGCR	Car	19.99
Charger	CGHM	Home	15.99

Write and test a program or programs for this shop.

- 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 arrays, 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.
Write a program to:
- use appropriate data structures to store the item code, description and price information for the mobile devices, SIM cards and accessories
- allow the customer to choose a specific phone or tablet
- allow phone customers to choose whether the phone will be SIM Free or Pay As You Go
- allow the customer to choose a standard or luxury case
- allow the customer to choose the chargers required (none, one or both may be purchased)
- calculate the total price of this transaction
- output a list of the items purchased and the total price.

Task 2 – Allow a customer to order multiple mobile devices.
Extend **Task 1** to:
- offer the customer the opportunity to purchase an additional mobile device
- if required, perform bulleted steps 2 to 7 of **Task 1** for each additional mobile device and calculate a running total for the customer
- once no further devices are required, output the total the customer will need to pay.

Task 3 – Offering discounts.
Extend the program to allow a discount of 10% off the price of every additional phone or tablet purchased.
Output the new total the customer will need to pay and the amount of money saved.

© UCLES 2020

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

(a) **(i)** Identify **two** arrays you could have used for **Task 1** and, in each case, state its purpose.

Array 1 .....

Purpose .....

.....

.....

Array 2 .....

Purpose .....

.....

.....
[4]

(ii) Identify **two** variables you could have used for **Task 1** and, in each case, state its purpose.

Variable 1 .....

Purpose .....

.....

.....

Variable 2 .....

Purpose .....

.....

.....
[4]

(b) Explain why the item code data could **not** be stored as a real data type and identify the most suitable data type for the item code data.

.....

.....

.....

.....

.....

..... [2]

(c) Write an algorithm to show how you input your choice of mobile device and SIM card (part of **Task 1**), using **either** pseudocode, programming statements **or** a flowchart. It is **not** necessary to show initialisation or setting up of arrays containing product details.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(d) Explain how your program completes **Task 3**. Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

Section B starts on page 8.

Section B

2 Tick (✓) **one** box in each row to identify if the statement about structure diagrams is true or false.

Statement	True (✓)	False (✓)
A structure diagram is a piece of code that is available throughout the structure of a program.		
A structure diagram shows the hierarchy of a system.		
A structure diagram is another name for an array.		
A structure diagram shows the relationship between different components of a system.		

[2]

3 Programs can perform validation and verification checks when data is entered.

(a) Give the names of **two** different validation checks and state the purpose of each one.

Check 1 Range Check

Purpose Check that the value entered is b/w upper & lower value.

Check 2 Presence Checks

Purpose Check that some data is entered.

[4]

(b) Give the name of **one** verification check.

..... [1]

(c) Describe the difference between validation and verification.

..... [2]

4 The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
  INPUT Values[Count]
  Count ← Count + 1
UNTIL Count = 0
```

(a) Explain why the algorithm will never end.

.....

.....

.....

.....

..... [2]

(b) Re-write the original pseudocode so that it terminates correctly **and** also prevents numbers below 100 from being stored in the array `Values[ ]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

.....

.....

.....

..... [2]

5 The flowchart represents an algorithm.

The predefined function `DIV` gives the value of the result of integer division, for example, `y ← 9 DIV 4` gives `y` a value of `2`

An input value of –1 ends the algorithm.

(a) Complete the trace table for the input data:

50, 33, 18, 15, 30, –1, 45, 12, 90, 6

`Value`	`Calc1`	`Calc2`	**OUTPUT**

[4]

(b) Describe the purpose of the algorithm.

.....

.....

..... [2]

6 A garden centre sells garden tools and stores details of these in a database table named TOOLS. **Code** is the primary key in the TOOLS table.

Code	Description	Price ($)	Quantity_Stock	Quantity_Ordered
GFLG	Garden Fork	50.00	1	50
GSLG	Garden Spade	50.00	11	0
GHLG	Garden Hoe	45.00	8	0
HFSM	Hand Fork	9.99	42	0
HSSM	Hand Spade	9.99	40	0
HWSM	Hand Weeder	9.99	11	0
HS20	Hose (20 metres)	45.00	10	0
HS35	Hose (35 metres)	60.00	2	0
HS50	Hose (50 metres)	75.00	20	60
YBLG	Yard Brush	24.99	100	0
LMHD	Lawn Mower	99.99	5	0
LMBT	Lawn Mower (Battery)	249.99	7	0
LMPT	Lawn Mower (Petrol)	349.99	10	25
TRBT	Edge Trimmer (Battery)	79.99	15	0
TRPT	Edge Trimmer (Petrol)	59.99	20	0
SHSM	Shears	40.00	40	0
HCSM	Hedge Clippers	40.00	45	0

(a) State the purpose of the primary key in the TOOLS table.

.....

.....

..... [1]

(b) List the output from the data shown in the table TOOLS that would be given by this query-by-example.

Field:	Code	Description	Price ($)	Quantity_Stock	Quantity_Ordered
Table:	TOOLS	TOOLS	TOOLS	TOOLS	TOOLS
Sort:					Descending
Show:	✓	✓	☐	☐	✓
Criteria:			>40	>0	>0
or:					

.....

.....

.....

.....

.....

..... [3]

(c) Complete the query-by-example grid to output the tools where the quantity in stock is below 25. Only show the Code, Description and Quantity_Stock fields in ascending order of Code.

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

[3]

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 2020

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

9119756575

COMPUTER SCIENCE **2210/21**

Paper 2 Problem-solving and Programming **May/June 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 **12** pages.

DC (EF) 311448
© UCLES 2022

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

Friends of Seaview Pier is an organisation devoted to the restoration and upkeep of a pier in the town. A pier is a wooden structure that provides a walkway over the sea. The pier requires regular maintenance and the friends of the pier need to raise money for this purpose.

Members of Friends of Seaview Pier each pay $75 per year, as a contribution to the pier's running costs. This entitles them to free admission to the pier throughout the year. They can also volunteer to help run the pier, by working at the pier entrance gate, working in the gift shop, or painting and decorating.

To provide additional income, the pier's wooden planks can be sponsored. A brass plaque, which contains a short message of the sponsor's choice, is fitted to a plank on the pier, for a donation of $200.

Write and test a program or programs for the Friends of Seaview Pier:
- 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 – becoming a member of Friends of Seaview Pier

Set up a system to enable people to become members of Friends of Seaview Pier and for each new member enter:
- their first name and last name
- whether or **not** they wish to work as a volunteer
  - if they choose to volunteer, identify the area from:
    - the pier entrance gate
    - the gift shop
    - painting and decorating
- the date of joining
- whether or **not** they have paid the $75 fee.

All of this information needs to be stored using suitable data structures.

Task 2 – using the membership data

Extend the program in **Task 1** so that a list of the first and last names of members can be output in any of the following categories:
- Members who have chosen to work as volunteers.
- Volunteers who would like to work at the pier entrance gate.
- Volunteers who would like to work in the gift shop.
- Volunteers who would like to help with painting and decorating tasks.
- Members whose membership has expired (they have **not** re-joined this year).
- Members who have **not** yet paid their $75 fee.

Task 3 – sponsoring a wooden plank

Add an additional option to the program in **Task 1** to enable the pier's wooden planks to be sponsored. Separate data structures should be used to store the names of the individuals and the short messages they would like to have written on their brass plaque. An output would display everything that was input for the sponsor to confirm. If errors are found, the program should allow data to be re-entered. Once complete, the data is stored and the sponsor is charged $200.

© UCLES 2022

1 **(a)** Describe the data structures you could have used in **Task 1**. Your description should include types of data structure, names used for data structures, their uses and examples of sample data.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(b) Explain how you could change your program in **Task 1** to total all the money collected from new members who have paid.

.....

.....

.....

.....

.....

..... [3]

(c) Describe how data input in **Task 1** could be validated to find out if a new member wants to work as a volunteer.

.....

.....

.....

.....

.....

..... [3]

(d) Write an algorithm to show how your program completes **Task 3** assuming the option to sponsor a wooden plank has been chosen, using pseudocode, programming statements or a flowchart. Details completed in **Task 1** are **not** required.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(e) Explain how your program allows any one of the member or volunteer lists to be selected and displayed (part of **Task 2**). Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Section B

2 Tick (✓) **one** box in each row to identify the most appropriate data type for each description. Only **one** tick (✓) per column.

Description	Data type				
	Boolean	**Char**	**Integer**	**Real**	**String**
a single character from the keyboard					
multiple characters from the keyboard					
only one of two possible values					
only whole numbers					
any number					

[4]

3 Give **one** piece of normal test data and **one** piece of erroneous test data that could be used to validate the input of an email address.

State the reason for your choice in each case.

Normal test data .....

.....

Reason .....

.....

.....

.....

Erroneous test data .....

.....

Reason .....

.....

.....

.....

[4]

4 The flowchart shows an algorithm that should allow 60 test results to be entered into the variable `Score`. Each test result is checked to see if it is 50 or more. If it is, the test result is assigned to the `Pass` array. Otherwise, it is assigned to the `Fail` array.

(a) Complete this flowchart:

[6]

(b) Write a pseudocode routine that will check that each test result entered into the algorithm is between 0 and 100 inclusive.

```
DECLARE testResult : INTEGER
INPUT "Enter test result:", testResult
IF testResult >=0 AND testResult <=100
    THEN
        OUTPUT "The test result is valid."
    ELSE
        OUTPUT "The test result is invalid."
ENDIF
```

Range Check

[4]

5 The pseudocode represents an algorithm.

The pre-defined function `DIV` gives the value of the result of integer division.
For example, `Y = 9 DIV 4` gives the value `Y = 2`

The pre-defined function `MOD` gives the value of the remainder of integer division.
For example, `R = 9 MOD 4` gives the value `R = 1`

```
First ← 0
Last ← 0
INPUT Limit
FOR Counter ← 1 TO Limit
  INPUT Value
  IF Value >= 100
    THEN
      IF Value < 1000
        THEN
          First ← Value DIV 100
          Last ← Value MOD 10
            IF First = Last
              THEN
                OUTPUT Value
            ENDIF
      ENDIF
  ENDIF
NEXT Counter
```

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

8, 66, 606, 6226, 8448, 642, 747, 77, 121

Counter	Value	First	Last	Limit	OUTPUT

[5]

(b) Describe the purpose of the algorithm.

.....

.....

.....

..... [2]

6 A computer game shop records its stock levels in a database table called GAMES. The fields used in the stock table are shown.

Name	Description
GameID	primary key
GameName	the name of each game
AgeRestriction	the minimum age at which a person is allowed to play each game
GamePrice	the selling price for each game
NumberStock	the quantity of each game currently in stock
OnOrder	whether or **not** each game is on order from the suppliers
DateLastOrdered	the date the most recent order for each game was placed
GameDescription	a summary of the contents and purpose of each game

(a) State the number of fields that are in the table GAMES.

..... [1]

(b) State **one** important fact that must be true for a field to be a primary key.

.....

..... [1]

(c) Complete the query-by-example grid to output all the games that have no stock and that are on order with the supplier. Display only the GameID, GameName and GamePrice fields in alphabetical order of the name of the game.

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

[3]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2022

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

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

3022701867

COMPUTER SCIENCE **2210/22**

Paper 2 Problem-solving and Programming **May/June 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 **12** pages.

DC (LK) 311524
© UCLES 2022

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 to allow a Wildlife Park to sell tickets. A booking consists of one or more tickets for the same day(s) and can be made up to a week in advance. A booking can be made for a visit of one day or two consecutive days. A booking can have extra attractions included. A booking will be valid for the day(s) chosen only.

Ticket type	Cost for one day	Cost for two days
one adult	$20.00	$30.00
one child (an adult may bring up to two children)	$12.00	$18.00
one senior	$16.00	$24.00
family ticket (up to two adults or seniors, and three children)	$60.00	$90.00
groups of six people or more, price per person	$15.00	$22.50

Extra attraction	Cost per person
lion feeding	$2.50
penguin feeding	$2.00
evening barbecue (two-day tickets only)	$5.00

Write and test a program or programs for the Wildlife Park:
- 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 – displaying the ticket options and the extra attractions available
Set up your program to:
- display the options, attractions and prices for one-day tickets
- display the options, attractions and prices for two-day tickets
- show the days available for booking; assume that there are tickets available for any valid day.

Task 2 – process a booking
Extend your program for **Task 1** to:
- input the tickets and extra attractions required, then calculate the total cost of the booking
- allocate a unique booking number
- display the booking details, including the total cost and the unique booking number
- repeat as required.

Task 3 – ensuring each booking is the best value
Check that the total for each booking gives the best value and offer an alternative if this is **not** the case. For example, buying two family tickets is better than a group ticket for a group of 10 that includes four adults and six children.

© UCLES 2022

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

(a) **(i)** Identify **one** constant that you could have used for **Task 1**.
Give the value and use of the constant.

Constant .....

Value .....

Use .....

.....
[3]

(ii) Identify **one** variable that you could have used for **Task 2**.
Give the data type and use of the variable.

Variable .....

Data type .....

Use .....

.....
[3]

(b) Explain how your program showed the days available for booking in **Task 1**.

.....

.....

.....

.....

.....

..... [3]

(c) Explain how your program made sure that each booking number allocated in **Task 2** was unique.

.....

.....

.....

..... [2]

(d) Write an algorithm for the part of **Task 2** that inputs the tickets and extra attractions required then calculates the total cost of the booking.

Assume that the booking is for a single day.

Use pseudocode, programming statements or a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(e) Explain how your program decides when a family ticket is better value in **Task 3**.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

Section B starts on page 8

Section B

2 An algorithm allows a user to input their password and checks that there are at least eight characters in the password. Then, the user is asked to re-input the password to check that both inputs are the same. The user is allowed three attempts at inputting a password of the correct length and a matching pair of passwords. The pre-defined function `LEN(X)` returns the number of characters in the string, `X`

```
Attempt ← 0
REPEAT
  PassCheck ← TRUE
  OUTPUT "Please enter your password "
  INPUT Password
  IF LEN(Password) < 8
    THEN
      PassCheck ← TRUE
    ELSE
      OUTPUT "Please re-enter your password "
      INPUT Password2
      IF Password <> Password
         THEN
         PassCheck ← FALSE
      ENDIF
  ENDIF
  Attempt ← Attempt + 1
UNTIL PassCheck OR Attempt <> 3
IF PassCheck
  THEN
   OUTPUT "Password success"
  ELSE
    OUTPUT "Password fail"
ENDIF
```

(a) Identify the **three** errors in the pseudocode and suggest a correction to remove each error.

Error 1 .....

Correction .....

Error 2 .....

Correction .....

Error 3 .....

Correction .....

[3]

(b) The algorithm includes **two** types of check on the data input.
Identify and describe each type of check.

Type of check 1 .....

Description .....

.....

Type of check 2 .....

Description .....

.....
[4]

(c) Give **two** sets of test data for this algorithm and a reason for choosing each set.

Each set of test data and its reason must be different.

Set 1 .....

Reason .....

.....

Set 2 .....

Reason .....

.....
[4]

3 **(a)** Describe a one-dimensional array. Include an example of an array declaration.

.....

.....

.....

.....

.....

..... [3]

(b) Explain how indexing could be used to search for a value stored in a one-dimensional array.

.....

.....

.....

..... [2]

4 This algorithm checks the temperature of hot food being served to customers.

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

75, 78, 84, 87, 91, 80, 75, 70, 65, 62, –1, 20

Counter	Hot	Cold	Serve	Temp	Error	OUTPUT

[7]

(b) State how the final output from the algorithm could be improved.

.....

..... [1]

(c) Identify the process in the algorithm that is **not** required.

.....

..... [1]

5 A database table, NURSE, is used to keep a record of disposable items worn by veterinary nurses.

This is part of the table:

ItemNumber	Description	SingleUse	Uses	StockLevel	ReorderLevel
DIG1	Glove (pair)	Y	1	500	800
DIA1	Apron	Y	1	700	800
DIM5	Hair net	Y	1	650	500
DIA2	Apron	N	5	25	100
DIS4	Suit	N	3	70	50
DIV9	Shoe cover (pair)	Y	1	400	250

(a) Complete this query-by-example grid to display only the item number and the description of single use items, where the stock level is below the reorder level.

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

[4]

(b) Give a reason why the field SingleUse is **not** required in the table NURSE.

.....

..... [1]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2022

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

COMPUTER SCIENCE **2210/23**

Paper 2 Algorithms, Programming and Logic **October/November 2023**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- 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 75.
- 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.

DC (CE/CT) 333541
© UCLES 2023

1 Tick (✓) **one** box to show which term is an example of a verification check.

A Double entry check ☐

B Format check ☐

C Length check ☐

D Presence check ☐

[1]

2 Tick (✓) **one** box to show which library routine returns the remainder of a division.

A DIV ☐

B MOD ☐

C RANDOM ☐

D ROUND ☐

[1]

3 **(a)** **Four** pseudocode descriptions and **five** pseudocode keywords are shown.

Draw **one** line to link each pseudocode description to the most appropriate pseudocode keyword. **Not** all pseudocode keywords will be used.

Pseudocode description	**Pseudocode keyword**
stores data in a file	OUTPUT
retrieves data from a file	WRITE
displays data on a screen	READ
enters data from a keyboard	OPEN
	INPUT

[4]

(b) Give **two** reasons for storing data in a file.

1 Persistance

2 Data Sharing

[2]

4 A programmer is writing a data entry program for booking theatre seats.
The programmer needs the program to accept only whole numbers that are greater than or equal to one and less than or equal to six.

Type check to allow only digits (0—9).

(a) Give the names of **two** validation checks that are required for this program.

1 Format Check to see no one enters a dot (.).

2 >= 1 AND <= 6
Range Check (1—6).

[2]

(b) Complete this pseudocode to perform your **two** validation checks, using your answers given in **(a)**:

```
OUTPUT "Please enter the number of seats you want to book "

INPUT Seats
WHILE Seats < 1 OR Seats > 6 OR Seats <> INT(Seats)
    OUTPUT "Invalid input. Please enter a whole number
            between 1 and 6"
    INPUT Seats
ENDWHILE
OUTPUT "You have successfully booked ", Seats, " seats."
```

[5]

(c) Give **one** item of test data to use when testing this program.
State the reason for your choice of test data.

Test data .....

Reason for choice .....

.....

[2]

BLANK PAGE

5 An algorithm has been written in pseudocode to check if a new password is in a list of previously used passwords `OldList[]`
If the password is **not** found, the new password will be stored at the end of the list to replace `"XXXX"` already stored there.

```
OUTPUT "Enter your new password "
INPUT NewPassword
Posn ← 1
Found ← FALSE
REPEAT
    IF Password = OldList[Posn]
      THEN
        Found ← TRUE
      ELSE Posn ← Posn + 1
    ENDIF
UNTIL Found AND OldList[Posn] = "XXXX"
IF Found
  THEN
    OUTPUT "Password has been used before"
  ELSE
    INPUT "New password accepted"
    OldList[Posn] ← NewPassword
ENDIF
```

(a) Identify the **three** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

[3]

(b) Complete this flowchart for the corrected algorithm:

[6]

6 There are **three** descriptions of logic gates. Each logic gate has two inputs **A** and **B** with one output **X**.
Identify each logic gate.
Complete a truth table for each logic gate.

(a) The only time the output is 1 is when both inputs are 1.

Logic gate .....AND.....

Complete the truth table for this description.

	A	B	X
0	0	0	0
1	0	1	0
2	1	0	0
	1	1	1

[2]

(b) The output is 1 when both inputs are different.

Logic gate .....XOR.....

Complete the truth table for this description.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

[2]

(c) The only time the output is 1 is when both inputs are 0.

Logic gate .....NOR.....

Complete the truth table for this description.

(A+B)	A	B	X
~~0~~ 1	0	0	1
~~1~~ 0	0	1	0
~~1~~ 0	1	0	0
~~1~~ 0	1	1	0

[2]

(d) Consider this logic expression:

X = (NOT **A** OR NOT **B**) OR NOT **C**

Draw a logic circuit for this logic expression. Each logic gate must have a maximum of **two** inputs. Do **not** attempt to simplify this logic expression.

[5]

7 A program uses both local variables and global variables.

Describe **two** differences between local variables and global variables.

Difference 1 - Local variables are declared inside the subroutine
- Global variables are declared outside the subroutine

Difference 2 - Local variables can only be accessible to the subroutine in which they are declared
- Global variables are available (and shared) to all subroutines equally in whole program.

[4]

8 This is an algorithm to find if a batch of parts has been manufactured successfully.

(a) Complete the trace table using this data:
Y, Y, Y, N, Y, Y, Y, Y, N, Y, Y, Y, Y

Accept	Reject	PartOK	Error	OUTPUT

[5]

(b) Describe how the algorithm should be changed to accept 'Y' or 'y' for a successfully manufactured part.

.....

.....

.....

.....

.....

..... [3]

9 A sanctuary for pheasants has set up a new database table called `PheasantList` to store details of the different species of bird at the sanctuary. Part of this table is given, showing: species, description, number of birds at the sanctuary, if the birds are breeding or **not**, and number of young born this year.

Species	Description	NumberBirds	Breeding	Young
Edwards	blue-black with white tail	5	Yes	0
Japanese green	dark green with pale grey tail	2	Yes	2
Reeves	golden, white and red scaled plumage	4	Yes	1
Crawfords Kalij	glossy blue-black plumage	4	No	0
Crested fireback	blue-black with black tail	3	No	0
True silver	white laced top half and black lower half	7	Yes	1
Siamese fireback	grey plumage with crimson legs and feet	5	No	0
Mikado	iridescent plumage with white striped wings	3	Yes	4
Red junglefowl	many colours	2	Yes	0
Himalayan monal	many colours with metallic green crest	3	Yes	2
White eared	white with ear tufts	5	Yes	3
Brown eared	brown with ear tufts	9	Yes	1
Ring necked	long tail with white ring neck	2	Yes	2
Golden	rainbow coloured	3	Yes	4

(a) State the number of records and fields in this part of the database table.

Records ....14....

Fields ....05....

[2]

(b) **(i)** Give the name of a field that could be used for the primary key.

....Species.... [1]

(ii) Explain why the sanctuary might decide **not** to use the field in **(b)(i)** as the primary key.

....Species names may not be unique.....

.... [1]

(iii) A new field `SpeciesID` is added to the database table.
This field contains a six-character code, for example Ph0001.

Give a reason why this field would be a better primary key.

....SpeciesID ensures uniqueness for each record.....

.... [1]

(c) Write the output that would be given by this structured query language (SQL) statement:

```
SELECT Species, Description

FROM PheasantList

WHERE NumberBirds > 6;
```

.....

..... [2]

(d) Complete this SQL statement to display all the species of pheasant where the birds are breeding and there were no young born this year:

SELECT Species

FROM PheasantList

WHERE Breeding = "Yes" AND Young = 0

..... ;

[4]

10 A weather station takes temperature readings once an hour for a week. These temperatures are stored in a two-dimensional (2D) array `Temperatures[]`
Each column contains 24 readings for a single day. The first temperature is recorded at 00:00 and the final temperature at 23:00. There are seven columns, one for each day of the week, starting with Monday and ending with Sunday.

The variables `MaxDay`, `MinDay` and `AvDay` are used to store the maximum, minimum, and average temperatures for a day. The variables `MaxWeek`, `MinWeek` and `AvWeek` are used to store the maximum, minimum, and average temperatures for the week.

The array has already been set up and the data stored.

Write a program that meets the following requirements:

- finds the maximum and minimum temperatures for each day
- calculates the average temperature for each day
- outputs for each day:
  - name of the day, for example Monday
  - maximum temperature
  - minimum temperature
  - average temperature
- finds the maximum and minimum temperatures for the week
- calculates the average temperature for the week
- outputs:
  - maximum temperature for the week
  - minimum temperature for the week
  - average temperature for the week.

All temperatures output must be rounded to two decimal places.

You must use pseudocode or program code **and** add comments to explain how your code works. All inputs and outputs must contain suitable messages.

You do **not** need to declare any arrays or variables; you may assume that this has already been done.
You do **not** need to initialise the data in the array `Temperatures[]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [15]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2023

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

3419194803

COMPUTER SCIENCE **2210/22**

Paper 2 Problem-solving and Programming **October/November 2020**

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. Blank pages are indicated.

DC (KS) 205003
© UCLES 2020

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 online computer shop sells customised personal computers. Every computer sold includes a basic set of components costing $200 and additional items can be added from the table:

Category	Item code	Description	Price ($)
Case	A1	Compact	75.00
Case	A2	Tower	150.00
RAM	B1	8 GB	79.99
RAM	B2	16 GB	149.99
RAM	B3	32 GB	299.99
Main Hard Disk Drive	C1	1 TB HDD	49.99
Main Hard Disk Drive	C2	2 TB HDD	89.99
Main Hard Disk Drive	C3	4 TB HDD	129.99
Solid State Drive	D1	240 GB SSD	59.99
Solid State Drive	D2	480 GB SSD	119.99
Second Hard Disk Drive	E1	1 TB HDD	49.99
Second Hard Disk Drive	E2	2 TB HDD	89.99
Second Hard Disk Drive	E3	4 TB HDD	129.99
Optical Drive	F1	DVD/Blu-Ray Player	50.00
Optical Drive	F2	DVD/Blu-Ray Re-writer	100.00
Operating System	G1	Standard Version	100.00
Operating System	G2	Professional Version	175.00

As well as the basic set of components every computer must include one case, one RAM and one Main Hard Disk Drive from the table.
A computer is supplied with or without an Operating System.

Write and test a program or programs for the online computer shop.

- 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 arrays, 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 and ordering the main items.
Write a program to:

- use arrays to store the item code, description and price
- allow a customer to choose one case, one RAM and one Main Hard Disk Drive
- calculate the price of the computer using the cost of the chosen items and the basic set of components
- store and output the chosen items and the price of the computer.

Task 2 – Ordering additional items.
Extend TASK 1 to:

- allow a customer to choose whether to purchase any items from the other categories – if so, which item(s)
- update the price of the computer
- store and output the additional items and the new price of the computer.

Task 3 – Offering discounts.
Extend TASK 2 to:

- apply a 5% discount to the price of the computer if the customer has bought only one additional item
- apply a 10% discount to the price of the computer if the customer has bought two or more additional items
- output the amount of money saved and the new price of the computer after the discount.

© UCLES 2020

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

(a) **(i)** Identify **one** array you could have used for **Task 1** and state its purpose.

Array .....

Purpose .....

.....

..... [2]

(ii) Identify **one** variable you could have used for **Task 2** and state its purpose.

Variable .....

Purpose .....

.....

..... [2]

(iii) Identify **one** constant you could have used for **Task 3** and state its purpose.

Constant .....

Purpose .....

.....

..... [2]

(b) Explain the benefits of storing Price as a real data type.

.....

.....

.....

.....

.....

.....

..... [2]

(c) Write an algorithm to show how you completed **Task 1**, using **either** pseudocode, programming statements **or** a flowchart. It is not necessary to show initialisation or setting up of arrays in your answer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(d) Explain how your program completes **Task 3**. Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(e) Describe how you could alter your program to allow more than one computer to be bought.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [2]

Section B

2 Tick (✓) **one** box in each row to identify if the statement about subroutines is **true** or **false**.

Statement	true (✓)	false (✓)
A subroutine is called from within a program.	✓	
A subroutine is **not** a complete program.	✓	
A subroutine is a self-contained piece of code.	✓	
A subroutine must return a value to the code from which it was called.		✓

Only if it is a function.

[2]

3 This pseudocode algorithm is used as a validation check.

```
PRINT "Input a number from 1 to 5000"
REPEAT
  INPUT Number
  IF Number < 1 OR Number > 5000
    THEN
      PRINT "Invalid number, please try again"
  ENDIF
UNTIL Number >= 1 AND Number <= 5000
PRINT Number, " is within the correct range"
```

Identify **three** different types of test data. For each type, give an example of the test data you would use to test this algorithm and state a reason for your choice of test.

Type of test data 1 .....

Test data .....

Reason .....

.....

Type of test data 2 .....

Test data .....

Reason .....

.....

Type of test data 3 .....

Test data .....

Reason .....

.....

[6]

4 This pseudocode algorithm allows 5000 numbers to be entered and stored in an array called `Number`.

```
FOR Count ← 1 TO 5000
  INPUT Number[Count]
NEXT Count
```

Extend and re-write the algorithm using pseudocode to also count and output how many of the numbers stored in the array are greater than 500, using the variable `Higher`. Only output `Higher` once with an appropriate message.

```
DECLARE Number : ARRAY [1 : 5000] OF INTEGER
DECLARE Count : INTEGER

DECLARE Higher : INTEGER

Higher ← 0

FOR Count ← 1 TO 5000
    OUTPUT "Enter Number:"
    INPUT Number[Count]

    IF Number[Count] > 500 THEN Higher←Higher+1
NEXT Count

OUTPUT "The number of values > 500 = ", Higher
```

[6]

5 This pseudocode represents an algorithm.

Bubble Sort

```
REPEAT
  Flag ← 0
  FOR Count ← 0 to 3
    IF Num[Count] < Num[Count + 1]
      THEN
        Store ← Num[Count]
        Num[Count] ← Num[Count + 1]
        Num[Count + 1] ← Store
        Flag ← 1
    ENDIF
  NEXT Count
UNTIL Flag = 0
```

0	45
1	56
2	30
3	12
4	15

Num[]

(a) The contents of the array at the start of the algorithm are:

Num[0]	Num[1]	Num[2]	Num[3]	Num[4]
45	56	30	12	15

Complete the trace table for the algorithm using the data given in the array.

Flag	Count	Num[0]	Num[1]	Num[2]	Num[3]	Num[4]	Store
0	0	45	56	30	12	15	45
1		56	45				
	1						
	2						
1	3				15	12	12
		56	45	30	15	12	
0	0						
	1						
	2						
	3						

[5]

(b) Describe the purpose of the algorithm.

The purpose is to sort the array Num[] in descending order.

[2]

6 Draw a flowchart symbol to represent each of the following:

Input/Output	Decision

[2]

Bubble Sort.

Question 7 starts on Page 12.

0	45
1	56
2	30
3	12
4	15

Num[]

0	45	30	12
✓ 1	30	12	15
✓ 2	12	15	30
✓ 3	15	45	45
4	56	56	56

Temp
45
56

```
For j ← 1 TO 3
  FOR i ← 0 TO 3
    IF Num[i] > Num[i+1]
    THEN
      Temp←Num[i]
      Num[i] ← Num[i+1]
      Num[i+1] ← Temp
    ENDIF
  NEXT i
NEXT j
```

Asc: >
Desc: <

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.

7 The table AUDIOPARTS stores the part number, description, cost and quantity in stock of the items sold by a music shop.

PartNum	Description	Cost	Quantity
A01	Compact Amplifier Case	50.00	15
A02	Deluxe Amplifier Case	75.00	1
A03	Amplifier Standard	79.99	48
A04	Amplifier Midrange	149.99	50
A05	Amplifier Megablaster	299.99	48
S01	Tweeter	59.99	10
S02	Midrange Woofer	99.99	0
S03	Subwoofer	139.99	16
S04	Tower Speaker Basic	159.99	25
S05	Tower Speaker Skyscraper	219.99	9
S06	Centre Speaker	149.99	25
S07	Soundbar	89.99	2
S20	Soundbar	129.99	0
S21	Ceiling Surround Speaker	75.00	15
S22	Ceiling Full Range Speaker	100.00	1
S25	Surround Speaker	100.00	60
T19	Speaker Stands (Pair)	75.00	60

(a) State the number of records in the table AUDIOPARTS

..... [1]

(b) Identify the field that is most suitable to be a primary key and give a reason for your choice.

Fieldname .....

Reason .....

.....

.....
[2]

(c) Complete the query-by-example grid to show the items where the quantity in stock is fewer than 10. Show all the fields from the database table in descending order of cost.

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

[4]

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

8250423928

COMPUTER SCIENCE **2210/21**

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.

DC (LO) 214473
© 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

A system is required to record and count votes for candidates in school council elections. The voting system will allow for one representative to be elected from a tutor group. The school has between 28 and 35 students in each tutor group, five year groups named Year 7 to Year 11, and there are six tutor groups in each year group. Tutor group names are their year group followed by a single letter e.g. 7A, 7B, etc.

All students are allowed to vote in the system. Each student may only vote once for a representative from their tutor group in the election.

Write and test a program or programs for the voting 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 voting system to allow a tutor group to elect a representative.
Write a program to:

- allow the tutor to enter the name of the tutor group
- allow the tutor to enter the number of students in the tutor group
- allow the tutor to enter the number of candidates in the election; maximum of four candidates
- allow the tutor to enter the names of the candidates and store them in a suitable data structure
- allow each student to input their vote or to abstain
- count the votes for each candidate and student abstentions.

When all students have voted, display the name of the tutor group, the votes for each candidate and the name of the candidate who has won the election. If there is a tie for first place, display all candidates with the equal highest number of votes.

Task 2 – Checking that students only vote once.
Each student is given a unique voter number by their teacher.
Extend **Task 1** to achieve the following:

- Allow students to enter their unique voter number before casting their vote.
- Check whether the student has already voted:
  - if so, supply a suitable message and do **not** allow them to vote.
  - if not, store the unique voter number, but **not** their vote, in a suitable data structure, and add their vote to the relevant candidate count or abstention.

Task 3 – Showing statistics and dealing with a tie.
Extend **Task 2** to achieve the following:

- Calculate the percentage of the votes that each candidate received from the number of votes cast, excluding abstentions.
- Display the name of each candidate, the number of votes and the percentage of votes they received from the number of votes cast, excluding abstentions.
- Display the total number of votes cast in the election and the number of abstentions.
- In the event of a tie, allow the election to be immediately run again, with only the tied candidates as candidates, and all the students from the tutor group voting again.

© UCLES 2021

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

(i) Identify **one** constant you could have used for **Task 1**, give the value that would be assigned to it and its use.

Constant .....

Value .....

Use .....

.....

.....
[3]

(ii) Identify **one** variable and **one** array you could have used for **Task 1**.
Explain the use of each one.

Variable .....

Use .....

.....

.....

Array .....

Use .....

.....

.....
[4]

(b) Explain how you should change your program in **Task 1** to allow a tutor to enter up to eight candidates for the election.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Write an algorithm using pseudocode, programming statements **or** a flowchart to show how your program completes these parts of **Task 2**:

- Allows students to enter their unique voter number before casting their vote.
- Checks whether the student has already voted:
  - if so, supplies a suitable message and does **not** allow them to vote.
  - if not, stores the unique voter number, but **not** their vote, in a suitable data structure.

It is **not** necessary to show parts completed in **Task 1**, including counting of votes for each candidate.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(d) Explain how your program completes these parts of **Task 3**:

- Calculate the percentage of the votes that each candidate received from the number of votes cast, excluding abstentions.
- Display the name of each candidate, the number of votes and the percentage of votes they received from the number of votes cast, excluding abstentions.
- Display the total number of votes cast in the election and the number of abstentions.

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 **both**.

Statement	Validation (✓)	Verification (✓)	Both (✓)
Entering the data twice to check if both entries are the same.			✓
Automatically checking that only numeric data has been entered.	✓		
Checking data entered into a computer system before it is stored or processed.	✓		
Visually checking that no errors have been introduced during data entry.		✓	

[3]

3 Name and describe the most appropriate programming data type for each of the examples of data given. Each data type must be different.

Data: `37`

Data type name .....

Data type description .....

.....

.....

Data: `Cambridge2021`

Data type name .....

Data type description .....

.....

.....

Data: `47.86`

Data type name .....

Data type description .....

.....

.....

[6]

4 The pseudocode algorithm shown has been written by a teacher to enter marks for the students in her class and then to apply some simple processing.

```
Count ← 0
REPEAT
  INPUT Score[Count]
  IF Score[Count] >= 70
    THEN
      Grade[Count] ← "A"
    ELSE
      IF Score[Count] >= 60
        THEN
          Grade[Count] ← "B"
        ELSE
          IF Score[Count] >= 50
            THEN
              Grade[Count] ← "C"
            ELSE
              IF Score[Count] >= 40
                THEN
                  Grade[Count] ← "D"
                ELSE
                  IF Score[Count] >= 30
                    THEN
                      Grade[Count] ← "E"
                    ELSE
                      Grade[Count] ← "F"
                  ENDIF
              ENDIF
          ENDIF
      ENDIF
  ENDIF
  Count ← Count + 1
UNTIL Count = 30
```

(a) Describe what happens in this algorithm.

1. Count is initialised to 0 and used as index for the 2 arrays.
2. loop to input Scores & assign grades
3. Input scores
4. Grade assignment
5. Increments counter
6. Termination

..... [3]

(b) Write the pseudocode to output the contents of the arrays `Score[]` and `Grade[]` along with suitable messages.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(c) Describe how you could change the algorithm to allow teachers to use it with any size of class.

.....

.....

.....

.....

.....

.....

..... [3]

5 The flowchart represents an algorithm.

The algorithm will terminate if –1 is entered.

(a) Complete the trace table for the input data:

50, 75, 99, 28, 82, 150, –1, 672, 80

Value	Diff1	Diff2	OUTPUT

[4]

(b) Describe the purpose of the algorithm.

.....

.....

.....

..... [2]

6 A library uses a database table, GENRE, to keep a record of the number of books it has in each genre.

ID	GenreName	Total	Available	Loaned	Overdue
ABI	Autobiography	500	250	250	20
BIO	Biography	650	400	250	0
EDU	Education	20200	10000	10200	1250
FAN	Fantasy	1575	500	1075	13
GFI	General Fiction	35253	23520	11733	0
GNF	General Non-Fiction	25200	12020	13180	0
HFI	Historical Fiction	6300	3500	2800	0
HNF	Historical Non-Fiction	8000	1523	6477	0
HUM	Humour	13500	9580	3920	46
MYS	Mystery	26000	13269	12731	0
PFI	Political Fiction	23561	10523	13038	500
PNF	Political Non-Fiction	1823	750	1073	23
REF	Reference	374	374	0	0
ROM	Romance	18269	16800	1469	0
SAT	Satirical	23567	12500	11067	0
SCF	Science Fiction	36025	25000	11025	0
SPO	Sport	45720	32687	13033	3256
THR	Thriller	86000	46859	39141	0

(a) State the reason ID could be used as a primary key in the table GENRE.

.....

..... [1]

(b) State the number of records in the table GENRE.

.....

..... [1]

(c) Complete the query-by-example grid to display any genres with overdue books. Only display the ID, GenreName and Overdue fields in order of the number of books overdue from largest to smallest.

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

[4]

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

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

COMPUTER SCIENCE **2210/22**

Paper 2 Algorithms, Programming and Logic **October/November 2023**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- 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 75.
- 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.

DC (LK) 333552
© UCLES 2023

1 Tick (✓) **one** box to complete the sentence.

Verification is used to make sure that a value entered

A has **not** changed during input. ☐

B is an integer. ☐

C is correct. ☐

D is **not** a string. ☐

[1]

2 A type of validation check is a length check. Another type of validation check is used to make sure that any date entered is in the dd/mm/yyyy style:
dd means day, mm means month and yyyy means year.

(a) State the type of validation check used.

— Format Check — [1]

(b) Give **one** example of normal test data and **one** example of abnormal test data you should use to make sure the check **in part (a)** is working properly.

State a reason for each of your choices of test data.

Normal dd is b/w 1 & 31

Reason .....

Abnormal dd is < 1 or > 31

Reason .....

[4]

(c) Describe how a length check could be used with the date entered.

.....

.....

.....

..... [2]

3 **Four** pseudocode statements and **five** pseudocode uses are shown.

(a) Draw **one** line to link each pseudocode statement to the most appropriate pseudocode use.

Not all pseudocode uses will be required.

Pseudocode statement	Pseudocode use
`CALL Colour(NewColour)`	counting
`Value ← (A1 + A2 + A3) / 3`	finding an average
`Loop1 ← Loop1 + 1`	totalling
`IF Count > 7 THEN X1 ← 0`	using a conditional statement
	using a procedure

[4]

(b) A one-dimensional (1D) array called `Temperatures[]` has 25 elements beginning at index 1. It holds values that range between –20 and 100 inclusive.

Write a pseudocode algorithm using a single loop to find the lowest value in this array and output the result only once.

You do **not** need to declare or populate this array.

```
Lowest ← 1000
FOR Count ← 1 TO 25
    IF Temperatures[Count] < Lowest THEN Lowest ←
                                        Temperatures[Count]
NEXT
OUTPUT "Lowest temperature:", Lowest
```

[4]

4 An algorithm has been written in pseudocode to allow the names of 50 cities and their countries to be entered and stored in a two-dimensional (2D) array. The contents of the array are then output.

```
01 DECLARE City ARRAY[1:50, 1:2] OF BOOLEAN
02 DECLARE Count : INTEGER
03 DECLARE Out : INTEGER
04 Count ← 1
05 IF
06     OUTPUT "Enter the name of the city"
07     INPUT City[Count,2]
08     OUTPUT "Enter the name of the country"
09     INPUT City[Count, 2]
10     Count ← Count + 1
11 UNTIL Count = 50
12 FOR Out ← 1 TO 1
13     OUTPUT "The city ", City[Out, 1], " is in ", City[Out, 2]
14 NEXT Out
```

(a) Identify the **four** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

(b) Describe the changes you should make to the corrected algorithm to allow the name of a country to be input and to display only the stored cities from that country.

You do **not** need to rewrite the algorithm.

- A variable will be declared & initialise for the country name.
- User input will be taken into the country name variable
- A loop from 1 TO 50 will be executed
- City array's second column will be checked inside the loop for matching countries and
- if the country matches in second column, then
- first column holding city of that country will be output.

..... [5]

5 Explain how variables and constants should be used when creating and running a program.

..... [3]

6 The flowchart represents an algorithm that performs a process on groups of values that are input. The algorithm will fail if the first value of any group is 0.

An input of –1 will terminate the algorithm.

(a) Complete the trace table for the input data:

25, 35, 3, 0, 57, 20, 25, 18, 0, –1, 307, 40, 0

Value	Average	Total	Count	OUTPUT

[5]

(b) Describe the purpose of the algorithm.

.....

.....

.....

..... [2]

7 The string operation `SUBSTRING(Quote, Start, Number)` returns a string from `Quote` beginning at position `Start` that is `Number` characters long. The first character in `Quote` is in position 1.

Write pseudocode statements to:

- store the string `"Learning Never Exhausts The Mind"` in `Quote`
- extract and display the words `"The Mind"` from the string
- output the original string in lower case.

Quote ← "Learning Never Exhausts The Mind"
Start ← 25
Number ← 8
OUTPUT SUBSTRING(Quote, 25, 8) / OUTPUT SUBSTRING(Quote, Start, Number)
OUTPUT LCASE (Quote)

[5]

8 Explain why a programmer would use procedures and parameters when writing a program.

[4]

9 Consider the logic expression:

Z = (**A** NAND **B**) OR NOT (**B** XOR **C**)

$(B \oplus C)'$

(a) Draw a logic circuit for this logic expression.

Each logic gate must have a maximum of **two** inputs.

Do **not** simplify this logic expression.

[4]

(b) Complete the truth table from the given logic expression.

Working space

A	B	C	(AB)	(AB)'	(B⊕C)	(B⊕C)'	Z
0	0	0	0	1	0	1	1
0	0	1	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	1	0	1	0	1	1
1	0	0	0	1	0	1	1
1	0	1	0	1	1	0	1
1	1	0	1	0	1	0	0
1	1	1	1	0	0	1	1

OR

[4]

10 A database table called `Horses` stores details about the horses kept at a horse sanctuary.

Code	Breed	BreedOrigin	Gender	Age	Arrived
Text	Text	Text	Character	INTEGER	Date/Time
H002	Arabian	Saudi Arabia	M	5	28/09/2022
H004	Percheron	France	M	5	30/10/2022
H010	Friesian	Netherlands	M	6	15/11/2022
H011	Fjord	Norway	F	4	17/11/2022
H012	Clydesdale	Scotland	M	10	18/11/2022
H015	Arabian	Saudi Arabia	F	5	15/12/2022
H016	Arabian	Saudi Arabia	F	5	15/12/2022
H017	Clydesdale	Scotland	F	4	16/01/2023
H019	Percheron	France	M	3	16/01/2023
H025	Percheron	France	M	7	16/01/2023
H026	Clydesdale	Scotland	F	9	20/01/2023
H030	Clydesdale	Scotland	M	12	20/01/2023
H032	Fjord	Norway	M	3	24/03/2023
H033	Arabian	Saudi Arabia	F	15	27/04/2023
H034	Clydesdale	Scotland	F	4	14/06/2023
H035	Fjord	Norway	M	7	15/06/2023
H036	Friesian	Netherlands	F	15	20/07/2023
H037	Friesian	Netherlands	M	12	20/07/2023

(a) State the number of records in this database table.

18 ..... [1]

(b) Give the name of the field that is most suitable to be the primary key.

State the reason for this choice.

Field Code .....

Reason Uniquely identifies each horse (record) .....

..... [2]

(c) The database only allows the data types:

- Boolean
- character
- date/time
- integer
- real
- text.

Complete the table to show the most appropriate data type for each field.
Each data type must be different.

Field	Data type
Breed	text
Gender	character
Age	integer
Arrived	date/time

[2]

(d) Complete the structured query language (SQL) to return the code and breed of all the horses whose breed originated in Scotland.

SELECT ..... Code, Breed, ??

FROM ....Horses.....

WHERE ....BreedOrigin..... = "Scotland";

[3]

11 A wood flooring company stores the names of up to 100 customers in a one-dimensional (1D) array `Customers[]`. A two-dimensional (2D) array `Quotations[]` stores details of each customer's quotation:

- length of room (one decimal place)
- width of room (one decimal place)
- area of wood required (rounded up to next whole number)
- choice of wood index (whole number)
- price of wood required in dollars (two decimal places).

The floor measurements (room length and room width) are taken in metres. All floors are rectangles and room measurements must be between 1.5 and 10.0 inclusive.

The index of any customer's data is the same in both arrays. For example, a customer named in index 4 of `Customers[]` corresponds to the data in index 4 of `Quotations[]`

The wood choices available are:

Index	Wood type	Price per square metre ($)
1	Laminate	29.99
2	Pine	39.99
3	Oak	54.99

The data are stored in two 1D arrays named `WoodType[]` and `Price[]`. The index of the wood type and price in their arrays share the same index number.

Write a program that meets the following requirements:

- input a new customer's name, room length and room width
- check that each measurement is valid
- output an error message and require the measurement to be re-entered until it is valid
- calculate the area of the room by multiplying together the length of the room and the width of the room
- input the choice of wood and find its price per square metre
- calculate the price of the wood needed
- store all data in the relevant array
- output the customer's quotation to include: the name of the customer, the choice of wood and the calculated price of the wood required
- continue to accept the next customer.

You must use pseudocode or program code **and** add comments to explain how your code works. You do **not** need to declare any arrays or variables; you may assume that this has already been done.

You will need to initialise `WoodType[]` and `Price[]`

All inputs and outputs must contain suitable messages.

.....

.....

.....

.....

.....

..... [15]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2023

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

2950158328

COMPUTER SCIENCE **2210/22**

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.

DC (CJ) 315861
© UCLES 2022

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 organisation has a visitor car park with 20 car parking spaces numbered 1 to 20. Car park spaces can be booked by visitors up to two weeks before the date they are needed, as long as a space is available. Visitors request a car parking space by stating the day in the two-week period in which it is required. They give the licence number of the car to be parked and their name. The next available space, beginning at space 1, is allocated and the given data and booking are stored. A system is required to record the car park bookings.

Write and test a program or programs for the visitor car park booking system to work for a static period of two weeks:

- 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 booking system
Set up suitable data structures to store the car licence numbers and names of visitors who have booked car parking spaces. The data structures should have sufficient capacity to store data for each of the 20 parking spaces for a static period of two weeks. Allow a visitor to request a parking space on any day within the two-week period by entering a number between 1 and 14, inclusive. The system will check that there are spaces available on the day requested, and if so, will ask the visitor to enter their name and car licence number. This data will be stored in the data structures representing the first available parking space for the day requested. The visitor will be told the number of their parking space.

At the end of the two-week period, allow all of the data to be deleted ready for the next two-week period.

Task 2 – adding accessible parking spaces
The visitor car park booking system is to be re-designed to offer accessible parking. Spaces 1 to 5 are named accessible spaces. Spaces 6 to 20 are named general spaces.

Extend your program in **Task 1** so that:

- when a visitor requests a parking space, they are additionally asked if they need an accessible space
  - if so, they are allocated the first available space beginning at space 1 and finishing at space 20
  - if **not**, they are allocated the first available space beginning at space 20 and finishing at space 6.

The system must work so that visitors requiring accessible parking may be allocated any of the 20 spaces, but visitors who do **not** need accessible parking may only be allocated general spaces.

Task 3 – working out car park usage statistics
Extend the program to enable the following statistics to be counted and output on request:

- The number of accessible spaces used on any of the 14 days.
- The number of general spaces used on any of the 14 days.
- The total number of spaces used on any of the 14 days.
- The number of accessible spaces used in the whole 14-day period.
- The number of general spaces used in the whole 14-day period.
- The total number of spaces used in the whole 14-day period.

© UCLES 2022

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

(a) **(i)** Identify **one** constant you could have used for **Task 1** and state its value.

Constant .....

Value .....
[2]

(ii) Identify **one** array you could have used in **Task 1** and describe its use.

Array .....

Use .....

.....

.....
[2]

(b) Describe how your program could validate the input for the day number within the two-week period to make sure an appropriate value is entered (part of **Task 1**).

You must include programming statements as part of your answer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(c) Write an algorithm to show how your program allocates a parking space if the visitor requires accessible parking (part of **Task 2**), including all relevant input and output, using pseudocode, programming statements or a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(d) Explain how your program in **Task 2** could be altered so that bookings can be made for any time over a four-week period instead of the current two-week period. Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(e) Explain how your program records and outputs the number of accessible and general parking spaces used for the two-week period (part of **Task 3**). Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Section B

2 Draw a line to connect each programming concept to the most appropriate description.

Programming concept	Description
	carrying out an action multiple times within a loop structure
counting	
	adding together the numbers in a list of numbers
repetition	
	tracking the number of iterations a program has performed in a loop
selection	
	branching off to take a course of action depending on the answer to a question
sequence	
	a set of statements to be executed in order
totalling	

[4]

3 Describe the use of verification on input of data when entering a list of items in stock into a database. Explain why verification is necessary.

.....

.....

.....

.....

.....

..... [3]

— Range check —

4 Describe **one** type of test data that must be used to test if a program accepts valid input data.

.....

.....

.....

..... [2]

5 This pseudocode should allow 500 marks to be entered into the algorithm. If the mark is 80 or greater it is stored in an array for higher marks. If the mark is less than 80, but greater than or equal to 50 it is stored in an array for middle marks. The remaining marks are stored in an array for lower marks. The results from the algorithm are displayed at the end.

```
01  HighList ← 0
02  MidList ← 0
03  LowList ← 0
04  MarksEntry ← 0
05  REPEAT
06    INPUT Mark
07    IF Mark >= 80
08      THEN
09        Higher[HighList] ← Mark
10        HighList ← HighList + 1
11      ELSE
12        IF Mark >= 50
13          THEN
14            Middle[MidList] ← Mark
15            MidList ← MidList
16          ELSE
17            Lower[HighList] ← Mark
18            LowList ← LowList + 1
19        ENDIF
20    ENDIF
21    MarksEntry ← MarksEntry + 1
22  UNTIL MarksEntry = 500
23  OUTPUT "You entered ", HighList, " higher marks"
24  OUTPUT "You entered ", MidList, " middle marks"
25  OUTPUT "You entered ", LowList, " lower marks"
```

index of Array.

MarksEntry

MarksEntry

MarksEntry

+1

Mark = -1

Higher

(a) Identify 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) The corrected algorithm needs to be changed so that any number of marks may be entered and the algorithm runs until the user tells it to stop.

Write the new pseudocode statements that would be needed to achieve this and state where in the algorithm they would be placed.

To allow we need to replace the 'REPEAT ... UNTIL' loop with a loop that check for stopping conditions. Also the values for high mark, middle mark and low mark should be input before the loop starts, new variables will be required.

lets say a value -1 will be used as rogue value, i.e. to stop the loop as user wants to finish entry.

a check will also be needed to avoid count in lowlist when user enters -1.

[4]

6 This flowchart represents an algorithm to divide three-digit numbers into hundreds, tens and units.

The pre-defined function `DIV` gives the value of the result of integer division, for example `Y = 9 DIV 4` gives the value `Y = 2`

The pre-defined function `MOD` gives the value of the remainder of integer division, for example `R = 9 MOD 4` gives the value `R = 1`

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

97, 876, 4320, 606, 9875, 42, 124

Counter	Number	Hundreds	Temp	Tens	Units	OUTPUT

[5]

7 A school uses a database table, ASSESS, to keep a record of the internal assessments and the number of candidates for each of the subjects in its curriculum.

Text	Text	INT/Number	INT	INT
SubjectCode	**SubjectName**	**Exams**	**Practicals**	**Candidates**
COMP	Computer Science	2	1	200
INFO	Information Technology	1	2	200
MATH	Mathematics	3	0	350
PHYS	Physics	2	1	120
CHEM	Chemistry	2	1	120
BIOL	Biology	2	1	200
GEOG	Geography	2	0	200
HIST	History	2	0	250
GEOL	Geology	2	0	80
PHED	Physical Education	1	2	350
FREN	French	2	2	120
ENGL	English	2	2	350

This database only allows the data types:

- text
- number ✓
- currency
- Boolean.

(a) **(i)** State the most appropriate data type for the fields SubjectCode and Exams.

SubjectCode .....

Exams ..... [1]

(ii) State **one** reason why the Candidates field could **not** be of the Boolean data type.

.....

..... [1]

(b) Show the output given by the query-by-example grid.

Field:	SubjectName	Practicals	Candidates		
Table:	ASSESS	ASSESS	ASSESS		
Sort:	Ascending				
Show:	✓	☐	✓	☐	☐
Criteria:		<1			
or:					

.....

.....

.....

.....

.....

..... [3]

(c) Complete the query-by-example grid to output the subjects with fewer than 150 candidates. Display only the SubjectCode, SubjectName and Candidates fields in order of the number of candidates from largest to smallest.

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

[3]

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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.

© UCLES 2022

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

6434217614

COMPUTER SCIENCE **2210/22**

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.

DC (LO) 214517
© 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).

```
DECLARE LargestNum, SmallestNum, Range : INTEGER.
DECLARE Number, i : INTEGER.
LargestNum ← -1000
SmallestNum ← 1000
FOR i ← 1 TO 500
    REPEAT   // Range check (0 – 999), +ve Number
        OUTPUT "Enter a Number, < 1000:"
        INPUT Number
    UNTIL Number < 1000
    IF Number > LargestNum THEN LargestNum ← Number
    IF Number < SmallestNum THEN SmallestNum ← Number
NEXT i
Range ← LargestNum – SmallestNum

OUTPUT "Largest Number: ", LargestNum
OUTPUT "Smallest Number: ", SmallestNum
OUTPUT "Range:", Range
```

Number >= 0 AND Number < 1000

[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` ..... Condition Statement

`PRINT Number` ..... Output Statement

`Number ← Number + 1` ..... Counting Statement.

[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.

```
FOR i ← 1 TO 20
    dataArray[i] ← 0
NEXT
```

[3]

(b) Explain why you chose this loop structure.

Because array index range 1 to 20 is already known.
So, Count-controlled loop / for loop suits better. [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]

© UCLES 2021

BLANK PAGE

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

Cambridge O Level

Cambridge Assessment International Education
Cambridge Ordinary Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

2143143116

COMPUTER SCIENCE **2210/23**

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 **12** printed pages.

DC (CE) 189390
© 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

You have been asked to write a program to calculate the area of a wall and the cost of the tiles needed to cover it. The program should work for any room with up to four walls being tiled. Tiles are sold in boxes; each box covers one square metre.

Tile description	Price per box
Small black granite	$19.50
Small grey marble	$25.95
Small powder blue	$35.75
Medium sunset yellow	$12.50
Medium berry red	$11.00
Medium glitter purple	$52.95
Large oak wood effect	$65.00
Large black granite	$58.98
Large bamboo effect	$85.00
Extra-large white marble	$62.75

Write and test a program or programs to calculate the area of the walls and the cost of tiles needed to tile a room.

- 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 arrays, 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 and using the system for a single wall.
Set up your program to:
- Store the tile description, price per box and an identification code using a suitable programming technique.
- Display the list of tiles including the identification code, description and price per box.
- Prompt the user to input the height and width of the wall, and the identification code of the tiles they would like.
- Calculate and display the area of the wall, the number of boxes of tiles required (tiles are sold in boxes of complete square metres) and the total cost of the tiles.

Task 2 – Adding additional walls.
Extend your program to:
- Enter the number of walls to be tiled and the identification code of the tiles to be used. Only one identification code to be used for all walls.
- Enter the dimensions for each wall to be tiled.
- Calculate and display the total area of all walls, the number of boxes of tiles required (tiles are sold in boxes of complete square metres) and the total cost of the tiles.

Task 3 – Allowing for waste.
It is likely that some of the tiles will not be useable so it is sensible to allow a percentage for wastage, for example 10%. Alter your program to allow the user to input a percentage to calculate wastage and add this to the total area to be tiled. Calculate and display the revised number of boxes of tiles to be purchased (tiles are sold in boxes of complete square metres) and the total cost of the tiles.

© UCLES 2019

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

Name **two** arrays you could use for **Task 1**. State the data type and purpose of each one.

Array 1 .....

Data type 1 .....

Purpose .....

.....

Array 2 .....

Data type 2 .....

Purpose .....

.....

[4]

(b) Name a variable that you used for **Task 2**. State the data type and purpose of this variable. Explain why you chose to use a variable rather than a constant.

Variable .....

Data type .....

Purpose .....

.....

.....

Reason for use of a variable .....

.....

.....

[4]

(c) Write an algorithm for **Task 1**, using **either** pseudocode, programming statements **or** a flowchart. Assume that the first part of **Task 1**, the storage and display of tile descriptions, prices per box and identification codes has already been done.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(d) Explain how you extended your program to complete **Task 2**. Any programming statements you use in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(e) Explain how you could validate the input for percentage waste allowance in **Task 3**.

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

Section B

2 Describe the use of a subroutine in a program.

1. Code reusability
2. Modularity
3. Simplifies Debugging
4. Maintainability
5. Improved Collaboration

[2]

3 Name the **three** types of loop structure used in pseudocode.

1. Count-Controlled (FOR) Loop.
2. Post-condition (REPEAT) Loop.
3. Pre-Condition (WHILE) loop.

[3]

4 The following pseudocode algorithm uses nested IF statements.

```
IF Response = 1
  THEN
    X ← X + Y
  ELSE
    IF Response = 2
      THEN
        X ← X - Y
      ELSE
        IF Response = 3
          THEN
            X ← X * Y
          ELSE
            IF Response = 4
              THEN
                X ← X / Y
              ELSE
                OUTPUT "No response"
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

```
CASE OF Response
1: X ← X+Y
2: X ← X-Y
3: X ← X*Y
4: X ← X/Y
OTHERWISE: OUTPUT "No response"
END CASE.
```

(a) Name the type of statement demonstrated by the use of `IF … THEN … ELSE … ENDIF`

Conditional Statement.

..... [1]

(b) Re-write the pseudocode algorithm using a `CASE` statement.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

5 The algorithm performs an operation on the array named `MyData`

`DIV` means integer division, so only the whole number part of the result is returned
e.g. `7 DIV 2` returns a value of `3`

Binary Search

```
First ← 0
Last ← 16
Found ← FALSE
INPUT UserIn
WHILE (First <= Last) AND (Found = FALSE) DO
  Middle ← (First + Last) DIV 2
  IF MyData[Middle] = UserIn
     THEN
       Found ← TRUE
     ELSE
       IF UserIn < MyData[Middle]
         THEN
           Last ← Middle - 1
         ELSE
           First ← Middle + 1
       ENDIF
   ENDIF
 ENDWHILE
 OUTPUT Found
```

7 — DIV
2) 15
14
1 — MOD

15 DIV 2 = 7
15 MOD 2 = 1
DIV(15,2) = 7
MOD(15,2) = 1

This table shows the contents of the array: `MyData` e.g. `MyData[2]` stores the value `5`

First Last

	MyData															
Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]
Value	2	3	5	6	8	10	12	13	14	16	18	20	25	27	29	34	36

(a) Complete the trace table for the input data: 10

First	←	Last	UserIn	Middle	Found = False	OUTPUT
0	True	16	10	8	False T	
0	True	7		3	T	
4	True	7		5	True T	
						TRUE

[6]

(b) Describe the function being performed by the algorithm.

This algorithm is searching for the UserIn by finding the middle value in the array. It updates lower & upper bound every time value doesn't match & recalculates the middle in search of UserIn. (It is binary search).

[2]

6 Draw **four** different flowchart symbols and describe how they are used in a program flowchart.

Flowchart symbol	Description of use
	Decision diagram, used to show use of IF...ENDIF and CASE...ENDCASE statements
	Process box. Shows assignment
	Input/output Shows inputs & outputs of a program.
	Start/Stop. Marks the start or end of the program.

[4]

7 A teacher has decided to use a database table as her mark book for her Computer Science class, which she has called MARKBOOK. For each student, the following data will be recorded: first name, last name, their year 10 test score and their year 11 test score. The class has 32 students.

(a) State the number of fields and records required for this database.

Number of Fields .....

Number of Records .....
[2]

(b) The data in MARKBOOK is stored under category headings: LastName, FirstName, Y10TestScore and Y11TestScore.

State, with a reason, whether any of these headings would be suitable as a primary key.

.....

.....

.....

..... [2]

(c) Complete the query-by-example grid to only display the first name, last name and year 10 test score of each student who achieved 50 or more in their year 10 test. The output should be in test score order with the highest marks at the top of the list.

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 2019

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

2085427188

COMPUTER SCIENCE **2210/22**

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.

DC (CE/FC) 221492
© 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		Start station to end station		End station to destination	
Code	**Price ($)**	**Code**	**Price ($)**	**Code**	**Price ($)**
C1	1.50	M1	5.75	F1	1.50
C2	3.00	M2	12.50	F2	3.00
C3	4.50	M3	22.25	F3	4.50
C4	6.00	M4	34.50	F4	6.00
C5	8.00	M5	45.00	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.

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.

```
Continue ← 1
WHILE Continue = 0
  OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /"
  INPUT Operator
  OUTPUT "Enter the first value"
  INPUT Value1
  OUTPUT "Enter the second value"
  OUTPUT Value2
  IF Operator
    1: Answer ← Value1 + Value2
    2: Answer ← Value1 - Value2
    3: Answer ← Value1 * Value2
    4: Answer ← Value1 / Value2
  ENDCASE
  OUTPUT "The answer is ", Value1
  OUTPUT "Do you wish to enter more values (Yes or No)?"
  INPUT MoreValues
  IF MoreValues = "No"
    THEN
      Continue ← 1
  ENDIF
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 PK INT	code to identify the product
ProductName Text	name of product
ProductDescription Text	information about the product
Animal Text	type of animal the product is for, e.g. cat, bird, horse
ProductType Text	type of product, e.g. food, toy, medicine
InStock Boolean	whether the product is in stock or **not**

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

InStock ..... [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

Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER				

6201582860

COMPUTER SCIENCE **2210/22**

Paper 2 Problem-solving and Programming **May/June 2020**

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. Blank pages are indicated.

DC (JP) 198772
© UCLES 2020

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 car park payment system allows customers to select the number of hours to leave their car in the car park. The customer will get a discount if they enter their frequent parking number correctly. The system calculates and displays the amount the customer must pay. The price of parking, the number of hours the customer can enter, and any discount depend upon the day of the week and the arrival time. The number of hours entered is a whole number. The price per hour is calculated using the price in force at the arrival time. No parking is allowed between Midnight and 08:00.

Day of the week	Arrival time			
	From 08:00 to 15:59		From 16:00 to Midnight	
	Max stay in hours	Price per hour	Hours	Price
Sunday	8	2.00	Up to Midnight	2.00
Monday	2	10.00	Up to Midnight	2.00
Tuesday	2	10.00	Up to Midnight	2.00
Wednesday	2	10.00	Up to Midnight	2.00
Thursday	2	10.00	Up to Midnight	2.00
Friday	2	10.00	Up to Midnight	2.00
Saturday	4	3.00	Up to Midnight	2.00

A frequent parking number can be entered for discounted parking. This number consists of 4 digits and a check digit that is calculated using a modulo 11 check digit calculation. A discount of 50% is available for arrival times from 16:00 to Midnight; the discount is 10% at all other arrival times.

Write and test a program or programs to simulate the car park payment 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 – Calculating the price to park.

A customer inputs the day, the hour of arrival excluding minutes (for example 15:45 would be 15), the number of hours to leave their car, and a frequent parking number if available. If the frequent parking number has an incorrect check digit, then no discount can be applied. The price to park, based on the day, the hour of arrival, the number of hours of parking required and any discount available, is calculated and displayed.

Task 2 – Keeping a total of the payments.

Extend **Task 1** to keep a daily total of payments made for parking. The daily total is zeroed at the start of the day. For the simulation, each customer inputs the amount paid, this must be greater than or equal to the amount displayed. There is no change given so the amount input may exceed the amount displayed. Each customer payment is added to the daily total, and this total is displayed at the end of the day.

Task 3 – Making payments fairer.

Customers have complained that sometimes they are being charged too much if they arrive before 16:00 and depart after 16:00. Extend **Task 1** to calculate the price before 16:00, then add the evening charge. For example, a customer arriving at 14:45 on a Sunday and parking for five hours was previously charged 10.00 and would now be charged 6.00

© UCLES 2020

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

(a) Identify **two** variables that you could have used for **Task 1**. Give the data type and state the use of each variable.

Variable 1 .....

Data type .....

Use .....

.....

Variable 2 .....

Data type .....

Use .....

.....

[4]

(b) Data input by a customer for **Task 1** includes the day and the hour of arrival.

Identify **one** suitable validation check for each input and justify your choice. Your validation checks must be different.

Day of arrival – validation check .....

.....

Justification .....

.....

.....

.....

Hour of arrival – validation check .....

.....

Justification .....

.....

.....

.....

[4]

(c) Explain how your program for **Task 1** ensured that the frequent parking number entered by the customer had a valid check digit.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) Write an algorithm for part of **Task 2** that simulates customer payment and calculating total payments using **either** pseudocode, programming statements **or** a flowchart. Assume that **Task 1** has been completed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(e) Explain how your program completed **Task 3**. Assume that **Task 2** has been completed. Any programming statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Section B

2 Most programming languages include basic data types. Ahmad is describing the basic data types he has used.

State the data type that Ahmad is describing in each sentence.

Choose the data type from this list of programming terms.

Array **Boolean** **Char** **Constant** **Function** **Integer**
Iteration **Procedure** **Real** **String** **Variable**

A number with a fractional part that can be positive or negative and used in calculations

Data type .....Real.....

A whole number that can be positive, negative or zero and used in calculations

Data type .....Integer.....

A single number, symbol or letter

Data type .....Char.....

A sequence of characters

Data type .....String.....

A data type with two values, True or False

Data type .....Boolean.....

[5]

3 **(a)** An algorithm has been written in pseudocode to input the names and marks of 35 students. The algorithm stores the names and marks in two arrays `Name[ ]` and `Mark[ ]`. The highest mark awarded is found and the number of students with that mark is counted. Both of these values are output.

LowestMark ← 100

Highest ← 0 −1000

Lowest ← 1000

```
01  HighestMark ← ~~100~~ 0
02  HighestMarkStudents ← 0  LowestMarkStudent ← 0
03  FOR Count ← 1 TO 35
04      OUTPUT "Please enter student name"
05      INPUT Name[Count]
06      OUTPUT "Please enter student mark"
07      INPUT Mark[Count]
08      IF Mark[Count] = HighestMark
09        THEN                LowestMark
10          HighestMarkStudents ← HighestMarkStudents + 1
11      ENDIF  LowestMarkStudents ← LowestMarkStudents+1
12      IF Mark[Count] > HighestMark
13        THEN  Mark[Count] < LowestMark
14          ~~Mark[Count] ←~~ HighestMark ← Mark[Count]
15          HighestMarkStudents ← 1        LowestMark ← Mark[Count]
16      ENDIF  LowestMarkStudents ← 1
17  NEXT Count
18  OUTPUT "There are ", HighestMarkStudents," with the highest mark of ",
      HighestMark
```

Counter → Count

− → +

Give line numbers where the **four** errors are to be found in the pseudocode. Suggest a correction for each error.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

Error 4 line number .....

Correction .....

.....

[4]

(b) Explain how you could extend the algorithm to also find the lowest mark awarded, count the number of students with that mark, and output both these values.

To extend the algo. to also find the lowest marks awarded, count the nmbr of students with that mark, and output both these values, we:

1) initialise "LowestMark" and "LowestMarkStudent" variables.
2) during the loop, update the "LowestMark" and 'LowestMarkStudents' accordingly.
3) Output both variables alongside others.

```
// Initialize variables
HighestMark ← 0
HighestMarkStudents ← 0
LowestMark ← 100
LowestMarkStudents ← 0

FOR Count ← 1 TO 35
   OUTPUT "Please enter student name"
   INPUT Name[Count]
   OUTPUT "Please enter student mark"
   INPUT Mark[Count]

   // Check for highest mark
   IF Mark[Count] = HighestMark THEN
      HighestMarkStudents ← HighestMarkStudents + 1
   ENDIF

   IF Mark[Count] > HighestMark THEN
      HighestMark ← Mark[Count]
      HighestMarkStudents ← 1
   ENDIF

   // Check for lowest mark
   IF Mark[Count] = LowestMark THEN
      LowestMarkStudents ← LowestMarkStudents + 1
   ENDIF

   IF Mark[Count] < LowestMark THEN
      LowestMark ← Mark[Count]
      LowestMarkStudents ← 1
   ENDIF
NEXT Count

// Output results
OUTPUT "There are ", HighestMarkStudents," students with the highest mark of ", HighestMark
OUTPUT "There are ", LowestMarkStudents," students with the lowest mark of ", LowestMark
```

[6]

4 This flowchart inputs the points won and the points lost when playing a game. The difference between the points won and lost is calculated and depending on the result the player can: move up to the next level, stay at the same level, or move down to the previous level. The flowchart finishes when the input for points won is –1.

(a) Complete a trace table for this set of input data:
5000, 4474, 6055, 2000, 7900, 9800, 3000, 2150, –1, 6700, 7615

PointsWon	PointsLost	Difference	OUTPUT

[3]

(b) The flowchart needs to be changed. When the difference is more than 5000 the output message is 'Fantastic leap up two levels'.

Describe the changes that will need to be made to the flowchart.

.....

.....

.....

.....

.....

.....

.....

..... [3]

5 Arrays are data structures used in programming. Explain what is meant by the terms dimension and index in an array. Use examples of arrays in your explanations.

Dimension refers to the number of indices to specify an element in the array. It defines the shape & structure of the array. Array can be 1D or 2D.

Y[3] ← 90

X[5,6] ← 105

Row Column

Index is the position of an element within the array, which is used to access or modify the element. Index can typically starts from 1 or 0, depending on language / requirement.

[3]

6 A database table, JUICE, is used to keep a record of cartons of fresh fruit juice available for sale.

Juice code (Alphanumeric)	Fruit 1 (Text)	Fruit 2 (Text)	Size (Text)	Volume (ml) (INT)	Stock level (INT)
LMO10	Mango	Orange	Large	1000	18
MOO05	Orange	Orange	Medium	500	8
SAM02	Apple	Mango	Small	200	25
SAA02	Apple	Apple	Small	200	50
SPP02	Pineapple	Pineapple	Small	200	10
MMM05	Mango	Mango	Medium	500	12
LMM10	Mango	Mango	Large	1000	5
MGG05	Guava	Guava	Medium	500	5
SMO02	Mango	Orange	Small	200	7
MOP05	Orange	Pineapple	Medium	500	12
LAA10	Apple	Apple	Large	1000	32
SGO02	Guava	Orange	Small	200	10
LPP10	Pineapple	Pineapple	Large	1000	3
LOO10	Orange	Orange	Large	1000	25
SOO02	Orange	Orange	Small	200	40

(a) Identify a suitable field to use as the primary key. State a reason for your choice.

Field .....Juicecode.....

Reason .....

.....

[2]

(b) Complete the query-by-example grid to display only the stock level and size of all cartons containing only apple juice.

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

[4]

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 2020