Cambridge O Level

Cambridge International Examinations
Cambridge Ordinary Level

CANDIDATE NAME	

CENTRE NUMBER						CANDIDATE NUMBER					

1699026409

COMPUTER SCIENCE **2210/22**

Paper 2 Problem-solving and Programming **October/November 2018**

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 **11** printed pages and **1** blank page.

DC (SC) 173734
© UCLES 2018

CAMBRIDGE International Examinations

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 junior park run event is held every week on a Saturday morning in a local park over a distance of two kilometres. Children between the ages of 4 and 14 inclusive can register to take part. Children register with their name and age. When they register, they are allocated a unique identification number of four digits; the last digit is a check digit. Once registered a child can take part in junior park run events for a year.

For each event, the organisers record the time each child takes to run two kilometres. Their time is stored for every event they complete and the number of runs they have completed is updated by one. If their time is faster than their personal best (PB) time, their PB time is updated. When a child has completed 11 runs, they are awarded a half-marathon wristband. When a child has completed 22 runs, they are awarded a full-marathon wristband.

A program is required to update the children's data, update PB times if necessary, and decide if a wristband is to be awarded. The program also needs to identify the fastest child at this event for each of the age ranges: 4 to 6, 7 to 10 and 11 to 14.

Write and test a program or programs for the park run organiser.

- Your program or programs must include appropriate prompts for the entry of data.
- Error messages and other output need to be set out clearly and be understandable.
- All variables, arrays, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

TASK 1 – Registering to take part.

Write a program to set up arrays to store the data for 20 children. On registration, each child must be allocated a unique identification number of four digits; the last digit is a check digit. The unique identification number, age in years and name for each child is recorded and stored on registration. The PB time and the number of runs are initialised to zero and these values stored on registration. Their PB time is stored as minutes correct to two decimal places.

TASK 2 – Recording the times.

Extend your program to record the unique identification number and to input the start time and finish time for every child completing the junior park run event. Calculate and store the time each child took to complete the run. A registered child does not have to compete in each event. Only one time per child is recorded during an event.

TASK 3 – Updating the children's data and identifying the fastest child for each age range.

Extend your program to update the number of runs and the PB time if necessary for every child completing the junior park run event. Check if any half- or full-marathon wristbands need to be awarded. Output the names and the type of wristbands. Output the names and the times of the fastest child at this event for each of the age ranges 4 to 6, 7 to 10 and 11 to 14.

© UCLES 2018

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

(i) State the name of **one** variable you have used for **Task 3**.
Give the data type for the variable. State what it is used for.

Variable name .....

Data type .....

Use .....
[3]

(ii) Describe the arrays that you have used to store the data for the children in **Task 1**.
Include the name, data type and its use for each array.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(b) Explain how you ensured that each identification number entered in **Task 1** was unique and included a correct check digit.

.....

.....

.....

.....

.....

.....

.....

..... [4]

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) Explain how your program identifies and outputs the names and times of the fastest runner for each age range in **Task 3**. Any programming statements used in your answer must be fully explained. **Do not** include the wristband check or PB time update.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Section B

2 **(a)** Write an algorithm, using pseudocode, to input three different numbers, multiply the two larger numbers together and output the result. Use the variables: `Number1`, `Number2` and `Number3` for your numbers and `Answer` for your result.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(b) Give **two** sets of test data to use with your algorithm in **part (a)** and explain why you chose each set.

Test data set 1 .....

Reason .....

.....

Test data set 2 .....

Reason .....

.....

[4]

3 **Four** programming concepts and **four** descriptions are shown.
Draw a line to connect each programming concept to the most appropriate description.

Programming concept	Description
Library routine	A subroutine that does not have to return a value.
Structure diagram	A standard subroutine that is available for immediate use.
Procedure	A subroutine that always returns a value.
Function	An overview of a program or subroutine.

[3]

4 A programmer wants to test that the readings from 2000 electricity meters are greater than 400 units and less than 900 units. The programmer uses selection and repetition statements as part of the program.
Explain, using programming statements, how selection and repetition could be used in this program.

Selection .....

.....

.....

.....

.....

Repetition .....

.....

.....

.....

.....

.....

.....

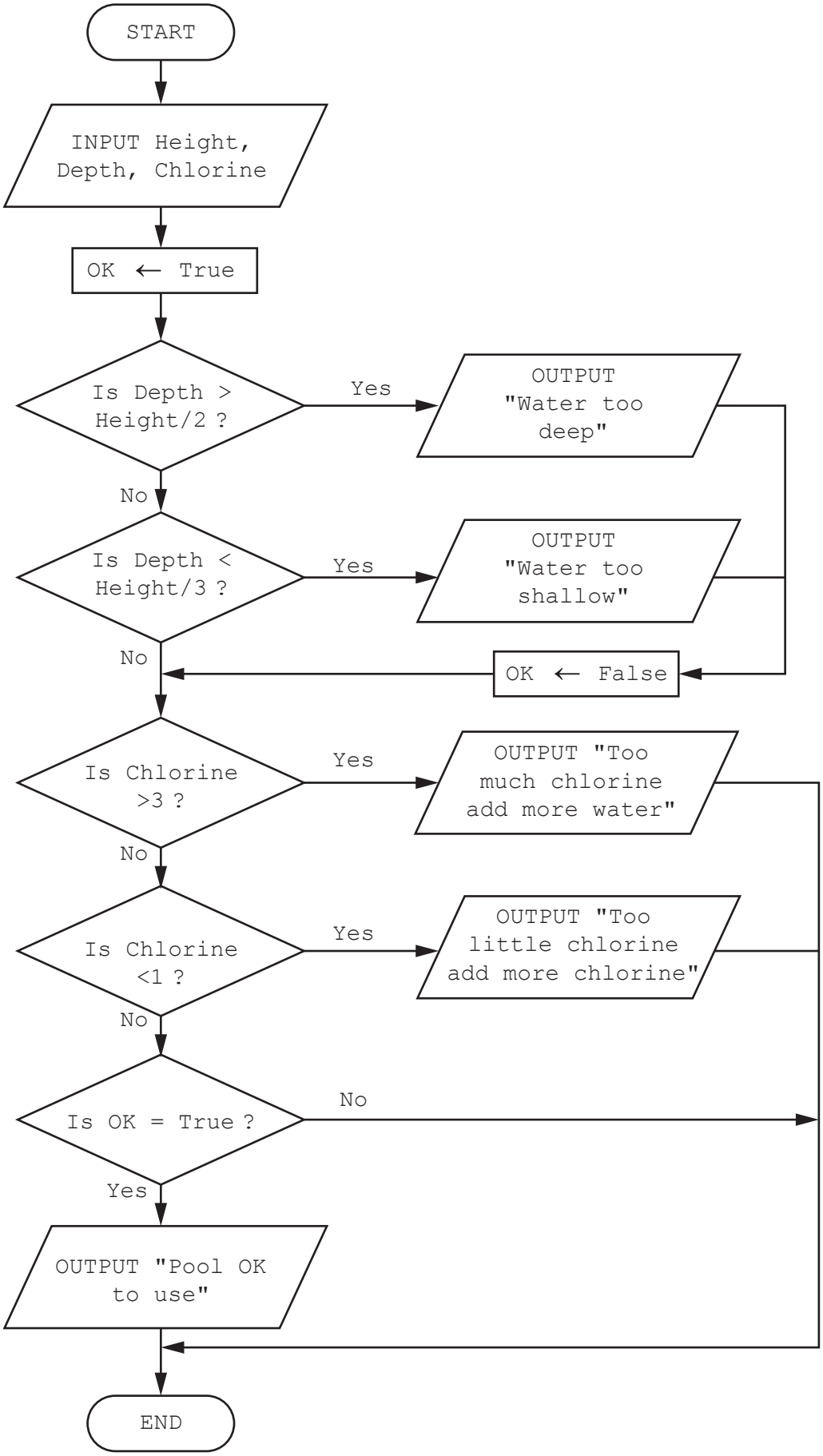
.....

[4]

BLANK PAGE

Question 5 starts on page 10.

5 The flowchart checks the level of chlorine and the depth of water compared to the height of the swimming pool. Error messages are output if a problem is found.



(a) Complete the trace tables for each set of input data.

Input data: **6**, **2.5**, **2**

Height	Depth	Chlorine	OK	OUTPUT

Input data: **4**, **3**, **1.5**

Height	Depth	Chlorine	OK	OUTPUT

Input data: **6**, **3.5**, **4**

Height	Depth	Chlorine	OK	OUTPUT

[6]

(b) Identify a problem with the algorithm that the flowchart represents.

.....

..... [1]

6 A database table, PORTRAIT, is used to keep a record of the portraits available from a photographic studio. Each portrait has a unique reference number PICnnn, where n is a single digit, for example PIC123. The studio keeps a record of the size (for example 20 × 15), the type (black and white or colour), and the price in dollars.

(a) Complete the table to show the most appropriate data type for each of the fields.

Field	Data type
Reference Number	
Size	
Type	
Price in $	

[4]

(b) The results from the query-by-example grid should show the reference number, price, type and size of all portraits under $50. Identify the **three** errors in the query-by-example grid.

Field:	Reference No	Price in $	Type	Size
Table:	PORTRAIT	PORTRAIT	PORTRAIT	PORTRAIT
Sort:				
Show:	✓	✓	☐	✓
Criteria:		>50.00		
or:				

Error 1 .....

.....

Error 2 .....

.....

Error 3 .....

.....

[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 International Examinations Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cie.org.uk after the live examination series.

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

© UCLES 2018