Cambridge International Examinations
Cambridge International General Certificate of Secondary Education

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **0478/22**

Paper 2 Problem-solving and Programming **February/March 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.

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 syllabus is approved for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

This document consists of **11** printed pages and **1** blank page.

DC (LK/CGW) 148974/2

CAMBRIDGE International Examinations

[Turn over

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

Students in a school are allowed to choose extra subjects each year. Students provide the school administrator with their names and their subject choices. Places in subject groups are allocated on a 'first come, first served' basis. There are two classes of 30 students and they can each choose **two** extra subjects from:

- Physics
- Chemistry
- History
- Geography
- Computer Science

The maximum group size for each subject choice is 20 students and the minimum group size is 10 students. If more than 20 students choose a subject then that subject can be split into two groups. Each subject can have no more than two groups. If less than 10 students choose a subject then it is not available that year. A program is required to show a summary of the number of students who have chosen each subject, identify subject group sizes, produce subject group lists and identify problems.

Write and test a program or programs for the school administrator.

- 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 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 – Data entry and number of students who have chosen each subject.

The school administrator enters the data for each student. Write a program for TASK 1 to store this data then calculate and output the number of students who have chosen each subject.

TASK 2 – Output subject group lists and identify problems.

Using your results from TASK 1, allocate students to subject groups. Print out list(s) of student names for each viable subject group. Identify any subjects that are over or undersubscribed, identify the students who have been allocated to one subject group only and those who have not been allocated to any group. Print out this information.

TASK 3 – Identify spare places in subject groups.

Using your results from TASK 2, print out the number of spare places for each subject. Any group that has fewer than 20 students has spare places. Calculate the total number of spare places and the total number of unallocated student choices. Show whether the number of spare places available is enough to cover the unallocated choices.

1 **(a)** All identifiers should have meaningful names.

(i) State the name and data structure that you have used to record student names in **Task 1**.

Name .....

Data structure .....
[2]

(ii) State the name of **one** constant and the name of **one** variable that you could have used in your programmed solution.
State the value that would be assigned to the constant. State the data type for the variable. Explain what each one would be used for in your programmed solution.

Constant name .....

Value .....

Use .....

.....

Variable name .....

Data type .....

Use .....

.....
[6]

(b) Explain how you would change your program for **Task 2** if the maximum group size for each subject is increased to 25.

.....

.....

.....

.....

.....

.....[2]

(c) Write an algorithm to complete **Task 1**, using **either** pseudocode, programming statements **or** a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

(d) Explain how your program calculates the total number of spare places in **Task 3**. Any programming statements shown in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

Section B

2 An algorithm has been written in pseudocode to input some numbers and print out any numbers that are greater than or equal to `100`. The number `999` stops the algorithm.

```
INPUT Number
WHILE NUMBERS <> 999 DO
  IF Number > 100 THEN PRINT Number ENDIF
ENDWHILE
PRINT Number
```

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

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

(b) Show, using pseudocode, how you would change the corrected algorithm to print out any numbers between `100` and `200` inclusive.

.....

.....

.....

.....

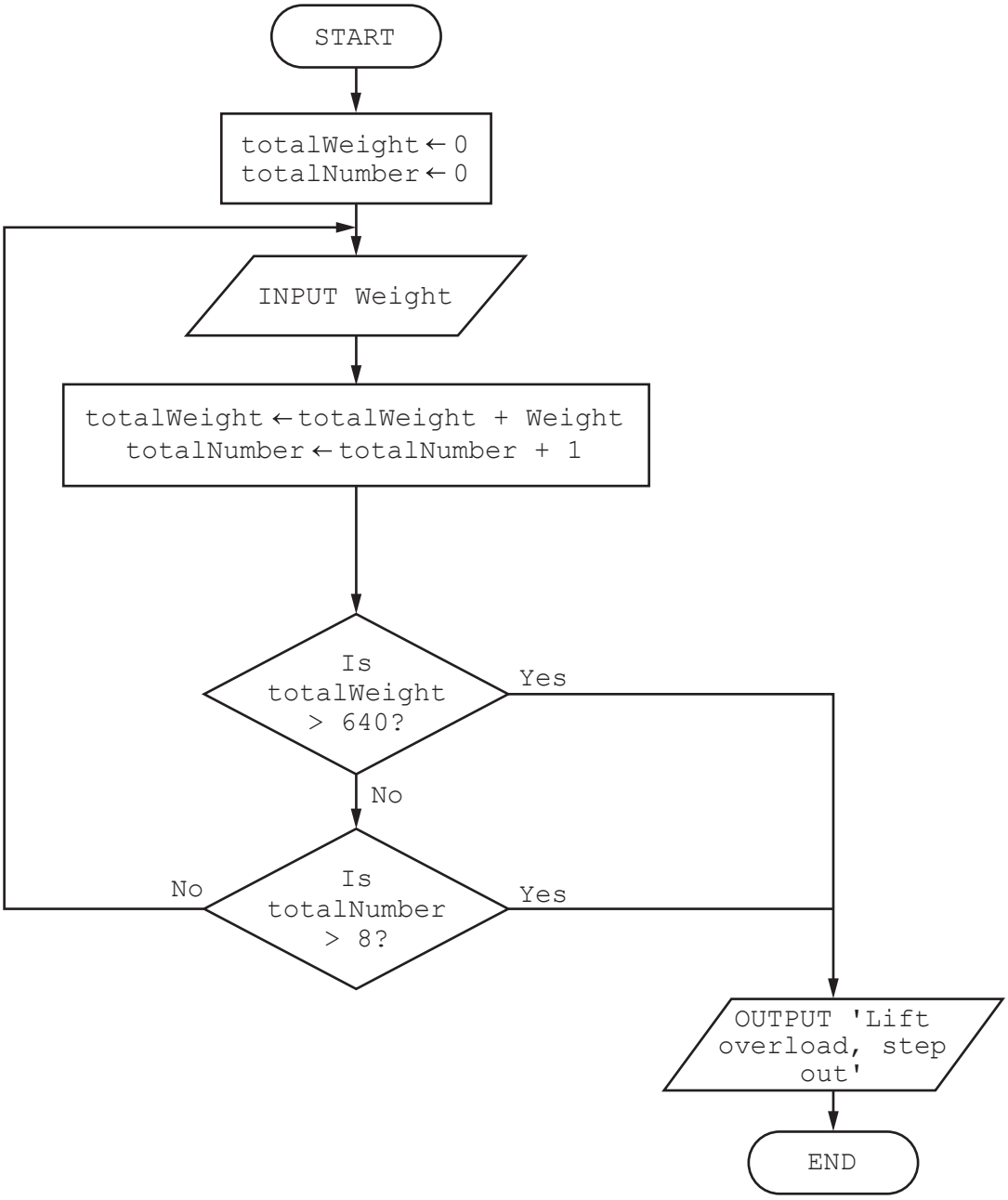
.....

.....

.....

.....[2]

3 This flowchart inputs the weight in kilograms of a passenger stepping into a lift. The lift can take a maximum of eight passengers or a maximum weight of 640 kilograms.



Complete the trace table for the passenger input data:

50, 70, 65, 100, 95, 50, 55, 85, 70, 75

Weight	totalWeight	totalNumber	OUTPUT

[4]

4 A program checks if the weight of a baby is at least 2 kilograms.

Give, with reasons, **two** different values of test data that could be used for the baby's weight. Each reason must be different.

Value 1 .....

Reason .....

.....

Value 2 .....

Reason .....

.....

[4]

5 Explain the difference between the programming concepts of **sequence** and **selection**. Include an example of a programming statement for each concept in your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

6 A database table, JEWEL, is used to keep a record of jewellery for sale in a shop. Each item of jewellery can be made of silver, platinum or gold metal. The shop stocks rings, bracelets and necklaces. The number in stock and the price is also stored.

(a) Identify the **four** fields required for the database. Give each field a suitable name and data type. Explain why you chose the data type for each field.

Field 1 Name ..... Data type .....

Explanation .....

.....

Field 2 Name ..... Data type .....

Explanation .....

.....

Field 3 Name ..... Data type .....

Explanation .....

.....

Field 4 Name ..... Data type .....

Explanation .....

.....

[8]

(b) Explain why none of these fields could be used as a primary key.

.....

.....[1]

(c) Using the query-by-example grid below, write a query to identify the silver bracelets. Only display the number in stock and the price.

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

[3]

BLANK PAGE