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.

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**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [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]

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.

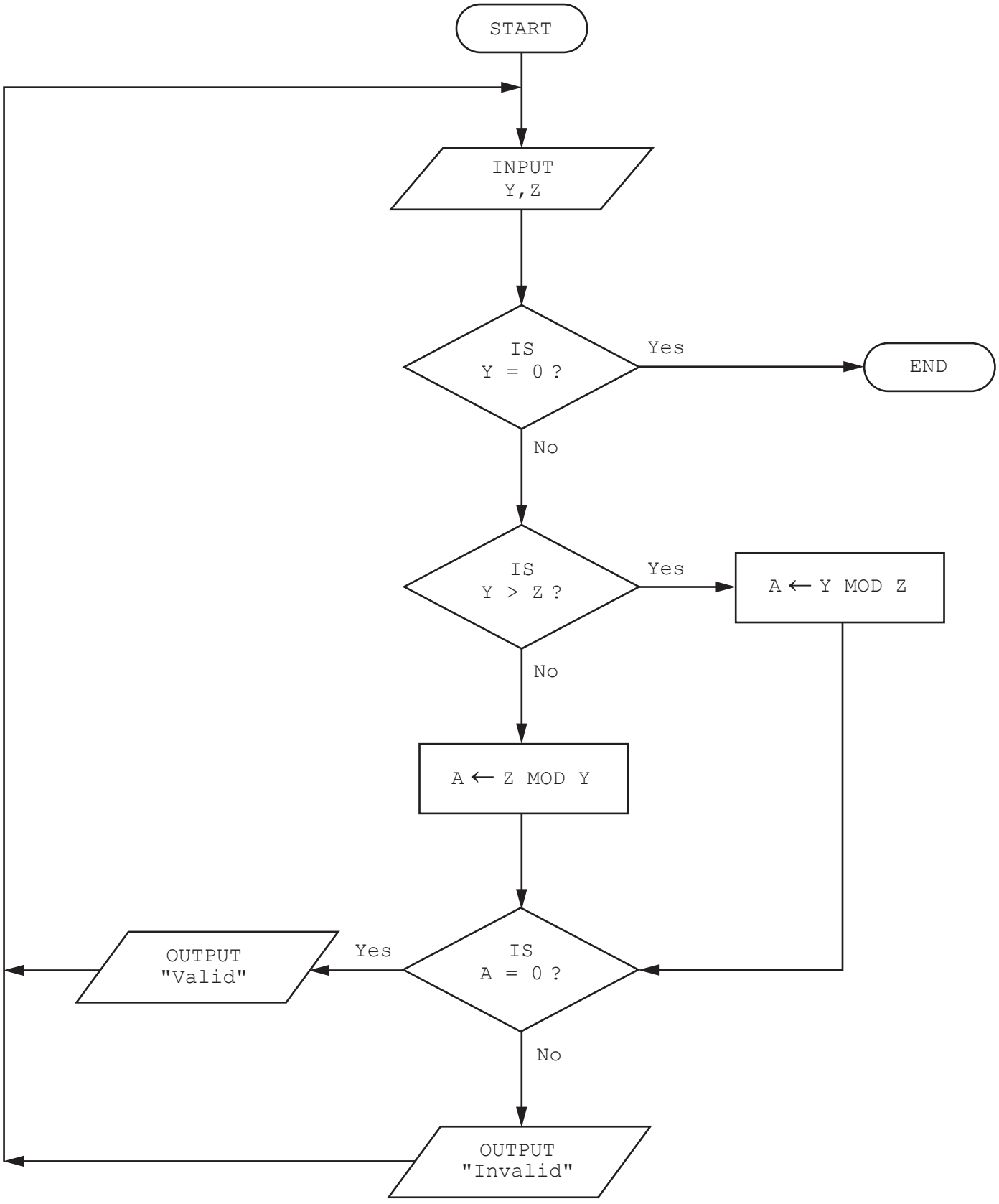
Programming concept	Description
Validation	A subroutine that does not have to return a value
Verification	An automatic check to ensure that data input is reasonable and sensible
Procedure	A subroutine that always returns a value
Function	An overview of a program or subroutine
	A check to ensure that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

[4]

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
Manta Ray	Fish	3	Y	N
Short-tailed Albatross	Bird	4	Y	N
Emperor Penguin	Bird	50	Y	Y
Bluefin Tuna	Fish	2	N	N
Manatee	Mammal	4	Y	N
Hawksbill Turtle	Reptile	10	Y	Y
Hammerhead Shark	Fish	3	Y	N
Yellow-eyed Penguin	Bird	4	Y	N
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 .....

Number of records .....
[2]

(b) Show the output that would be given by this query-by-example.

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