Cambridge IGCSE

Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

CANDIDATE NAME

CENTRE NUMBER | | | | | |

CANDIDATE NUMBER | | | | |

1551726642

COMPUTER SCIENCE **0478/22**

Paper 2 Problem-solving and Programming **February/March 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 syllabus is regulated for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

This document consists of **14** printed pages and **2** blank pages.

DC (LK/CB) 162269/1

Cambridge Assessment International Education

[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

A pizza ordering service allows customers to design their own pizza. There are three sizes: small, medium and large. A pizza can have a thick or thin base. All pizzas come with tomato and cheese toppings as standard and there are six additional types of topping available:

- Pepperoni
- Chicken
- Extra cheese
- Mushrooms
- Spinach
- Olives

Pizzas always come with tomato and cheese toppings as standard, and can have up to three additional toppings. Customers need to be able to design their own pizza and then confirm or change it. Records are kept showing the number of pizzas sold for each base and size. The number of sales for each additional topping is also recorded.

Write and test a program or programs for the pizza ordering service.

- 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 – Design your pizza.

The customer is given choices of size, base and additional toppings (number and type) as stated above. Only valid choices can be accepted. The customer is asked to confirm their order or alter their choices or not proceed. If the customer confirms their order they are given a unique order number.

TASK 2 – Record the choices.

Extend TASK 1 to record totals for the choices made for ordered pizzas only and calculate the total number of pizzas ordered.

TASK 3 – Find the most and least popular additional pizza toppings.

Using your results from TASK 2, display the most popular and least popular additional toppings as a percentage of the total number of additional toppings ordered.

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

State **one** constant and **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. Explain what each one could be used for.

Constant name .....

Value .....

Use .....

.....

Variable name .....

Data type .....

Use .....

.....

[6]

(b) Explain how you would need to change your program for **Task 1** if there were three bases to choose from (thick, thin and extra crispy).

.....

.....

.....

..... [2]

(c) (i) Write an algorithm for choosing the additional toppings in **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 (c)(i)** only allowed valid choices for the additional pizza toppings.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Section B

2 **(a)** An algorithm has been written in pseudocode to input 50 numbers and total only the positive numbers.

```
Count ← 1
Total ← Count
REPEAT
  INPUT Number
  IF Number <> 0
  THEN
    Total ← Total + Count
  ENDIF
  Count ← Count + 1
UNTIL Count < 50
PRINT Total
```

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 only total numbers greater than 0 and less than 20.

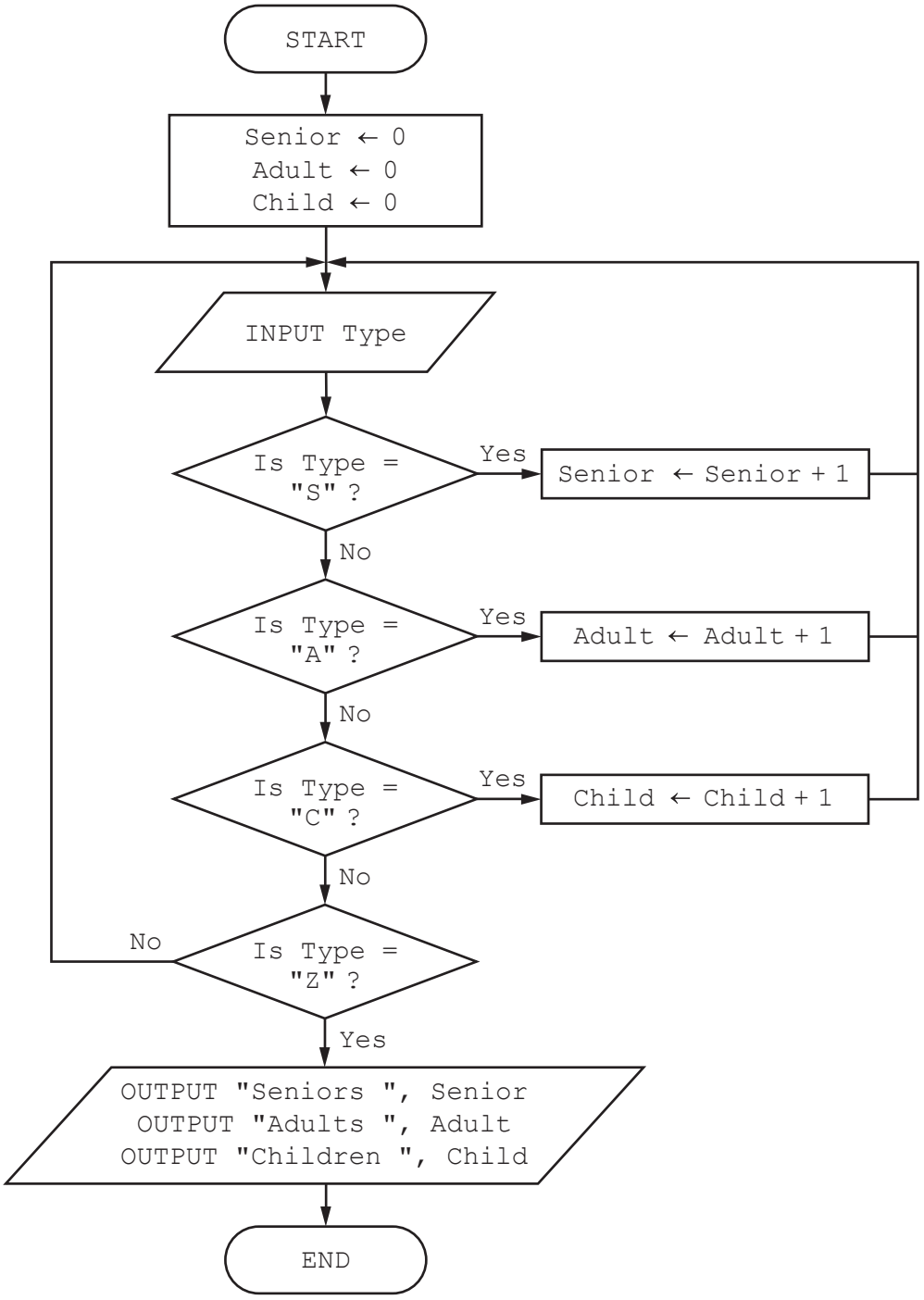
.....

.....

..... [2]

Question 3 starts on page 10.

3 This flowchart inputs the type of passenger for a survey: S for Senior, A for Adult, C for Child. All other values are ignored, apart from Z which ends the process.



Complete the trace table for the passenger input data:
S, S, S, A, C, C, C, A, A, A, A, W, S, S, D, C, Z, D, S

Senior	Adult	Child	Type	OUTPUT

[5]

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 X ← 1 TO 10`
`  SUM ← SUM + 1`
`NEXT X`		

`WHILE X > 10 DO`
`  SUM ← SUM + 1`
`  X ← X - 1`
`ENDWHILE`		

`IF X > 10`
`THEN`
`  SUM ← SUM + 1`
`  X ← X - 1`
`ENDIF`		

`REPEAT`
`  SUM ← SUM + 1`
`  X ← X - 1`
`UNTIL X > 10`		

[4]

5 A programmer restricts input values to less than 90 and greater than 60.

(a) State whether this is called validation or verification.

.....

Name the check that needs to be used.

.....

[2]

(b) State **three** different types of test data the programmer would need to use. Give an example of each type and the reason that the programmer chose that test data.

Type 1 .....

Example .....

Reason .....

.....

Type 2 .....

Example .....

Reason .....

.....

Type 3 .....

Example .....

Reason .....

.....

[9]

6 A database table, BIKETYRES, is used to keep a record of tyres for sale in a cycle shop. Tyres are categorised by width and diameter in millimetres, whether they have an inner tube and the type of terrain for which they are designed.

Tyre Code	Width	Diameter	Tube	Terrain	Stock Level
SLTT	23	700	YES	Asphalt	18
MLNT	24	700	NO	Asphalt	23
LLNT	28	700	NO	Asphalt	19
SLTM	23	700	YES	Mixed	22
MLTM	24	700	YES	Mixed	14
LLTM	28	700	YES	Mixed	12
SLTH	23	700	YES	Hard	10
MLTH	24	700	YES	Hard	5
LLNH	28	700	NO	Hard	7
SLNM	23	700	NO	Mixed	12
MLNM	24	700	NO	Mixed	22
LLNM	28	700	NO	Mixed	18
SSNT	23	650	NO	Asphalt	10
MSNT	24	650	NO	Asphalt	8
SSTM	23	650	YES	Mixed	5
MSNM	24	650	NO	Mixed	4

The query-by-example grid below displays the tyre code and the stock level of all 28 mm width tyres suitable for mixed terrain.

Field:	Tyre Code	Stock Level	Width	Terrain
Table:	BIKETYRES	BIKETYRES	BIKETYRES	BIKETYRES
Sort:				
Show:	☑	☑	☐	☐
Criteria:			= 28	= 'Mixed'
or:				

Alter the query to show the tyre code and stock level in ascending order of stock level for all 24 mm asphalt terrain tyres. Write the new query in the following query-by-example grid.

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

[4]

BLANK PAGE

BLANK PAGE

Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/22**

Paper 2 **March 2019**

MARK SCHEME

Maximum Mark: 50

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the March 2019 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

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

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Question	Answer	Marks
	Section A	
1(a)	Many correct answers, they must be meaningful. The names are examples only. Constant name `Max_Extra_Top` Value 3 Use Storing the number of extra toppings a pizza can have Variable name `NextOrderNo` Data type integer Use storing the next order number available	**6**
1(b)	Any **two** from Store an extra type of base Display another option Change the if statement/validation check to include the third option//extra crispy	**2**

Question	Answer	Marks

1(c)(i)	Any **five** from:
	Enter Number of toppings
	Check number of toppings chosen
	Display toppings available
	Provide method of selection
	Only accept correct selections
	Error message if topping not found
	Finish selection
	There are many possible correct answers, this is an example only.
	Sample answer
	`PRINT "Pepperoni 1"`
	`PRINT "Chicken 2"`
	`PRINT "Extra cheese 3"`
	`PRINT "Mushrooms 4"`
	`PRINT "Spinach 5"`
	`PRINT "Olives 6"`
	`REPEAT`
	`  PRINT "How many extra toppings do you want"`
	`  INPUT NoTopping`
	`UNTIL NoTopping >= 0 and NoTopping <=3`
	`WHILE NoTopping >0 DO`
	`  PRINT "Enter Topping ", NoTopping`
	`  INPUT ToppingType`
	`  CASE OF ToppingType`
	`  1: Pepperoni ← Pepperoni + 1`
	`  2: Chicken ← Chicken + 1`
	`  3: ExtraCheese ← ExtraCheese + 1`
	`  4: Mushrooms ← Mushrooms + 1`
	`  5: Spinach ← Spinach + 1`
	`  6: Olives ← Olives + 1`
	`  OTHERWISE: PRINT "Error"`
	`  ENDCASE`
		5

Question	Answer	Marks

1(c)(i)	`  IF ToppingType >=1 AND ToppingType <=6`
	`    THEN`
	`    PizzaTop[NoTopping] ← ToppingType`
	`    NoTopping ← NoTopping - 1`
	`  ENDIF`
	`ENDWHILE`	

1(c)(ii)	Answers must relate to the algorithm provided for (c)(i) Any **three** from: Display the valid toppings e.g. choose from a menu Check input for each topping is valid … method e.g. using a CASE statement / range check Provide a suitable error message for invalid toppings Provide a method to re-input a topping e.g. use of REPEAT…UNTIL	**3**
1(d)	Explanation Any **four** from: How the total of each additional pizza toppings was checked How the largest value was selected How the smallest value was selected How the topping descriptions were recorded for largest/smallest Method used to calculate percentages … … calculation relates to the total number of additional toppings ordered Display results including suitable messages	**4**

Question	Answer	Marks
	Section B	

2(a)	`Total ← Count` should be `Total ← 0`
	
	`Number <> 0` should be `Number > 0`
	
	`Total ← Total + Count` should be `Total ← Total + Number`
	
	`UNTIL Count < 50` should be
	`UNTIL Count > 50, UNTIL Count >= 51, UNTIL Count = 51`
	
	1 mark for each error identified + suggested correction
		4

2(b)	The test should be `IF Number > 0 AND Number <20` One mark for both ends of the range and one mark for the **AND.**	**2**

Question	Answer	Marks
3		**5**

Senior	Adult	Child	Type	OUTPUT
0	0	0		
			S	
1			S	
2			S	
3			A	
	1		C	
		1	C	
		2	C	
		3	A	
	2		A	
	3		A	
	4		A	
	5		W	
			S	
4			S	
5			D	
			C	
		4	Z	Seniors 5
				Adults 5
				Children 4

One mark for each correct column.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4</td><td>
<table>
<tr><th>Statements</th><th>Selection</th><th>Repetition</th></tr>
<tr><td>FOR X ← 1 TO 10
  SUM ← SUM + 1
NEXT X</td><td></td><td>✓</td></tr>
<tr><td>WHILE X > 10 DO
  SUM ← SUM + 1
  X ← X - 1
ENDWHILE</td><td></td><td>✓</td></tr>
<tr><td>IF X > 10
THEN
  SUM ← SUM + 1
  X ← X - 1
ENDIF</td><td>✓</td><td></td></tr>
<tr><td>REPEAT
  SUM ← SUM + 1
  X ← X - 1
UNTIL X > 10</td><td></td><td>✓</td></tr>
</table>
</td><td>4</td></tr>
</table>

Question	Answer	Marks
5(a)	Validation Range check	2

Question	Answer	Marks
5(b)	For each of **three** different data types Data type - 1 mark, Example - 1 mark, Reason - 1 mark There are many possible correct answers, this is an example only. Normal data (1 mark) 65 (1 mark) to show that the program accepts this value (1 mark) Erroneous data (1 mark) seventy (1 mark) to show that the program rejects this value (1 mark) Extreme data (1 mark) 89 (1 mark) to show that the program accepts this value (1 mark)	**9**

Question	Answer					Marks
6	Field:	Tyre Code	Stock Level	Width	Terrain	**4**
	Table:	BIKETYRES	BIKETYRES	BIKETYRES	BIKETYRES	
	Sort:		Ascending			
	Show:	☑	☑	☐	☐	
	Criteria:			= 24	= 'Asphalt'	
	or:					
	One mark for each correct column					