Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER				

7816195114

COMPUTER SCIENCE **0478/22**

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

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 record the number of strokes played by each player in a round of golf and decide who is the winning player.

The object of the game of golf is to hit a small ball into a series of small holes using a golf club. A score is kept of the number of times a player needs to hit the ball to get it in a hole; this is the number of strokes for the hole. The total score for a round of golf is the total number of strokes taken for each hole. The player with the least number of strokes is the winner.

A golf course consists of 9 or 18 holes. A round of golf is completed when all the holes have been played. There can be 2, 3 or 4 players taking part in a round of golf. The number of strokes that an experienced golfer would take to complete a round is called par. The scores for the round are displayed in relation to the par score. For example, if par for an 18-hole course was 72, a score of 80 would be 8 over par and a score of 70 would be 2 under par.

Write and test a program or programs to score a round of golf:

- 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 round

Each player's scores for the round are to be stored in an array with a separate score for each hole. The scores in these arrays are to be set to zero before a round starts.

The following data is to be input and stored:

- the number of players taking part in the round
- the names of the players
- the number of holes to be played: 9 or 18
- the par for the course.

The number of players, their names, the number of holes to be played and the par for the course are all displayed to be checked before the round starts so that any errors seen can be corrected.

Task 2 – scoring the round

For each hole played, each player's name is displayed on the screen and they are asked to enter the number of strokes they played for that hole. Each player must enter the number of strokes twice to verify their score. A player can choose to see the total number of strokes that they have played so far in the round.

Task 3 – deciding the winning player

At the end of the round, display the name and final score for each player relative to par. The winner is identified; display their name and the winning score. There are also options that display:

- every player's score for each hole
- the player's name and hole number of any score of one for a hole (hole-in-one)
- the average score for the round
- the average score for a hole.

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

(a) 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 why the constant was used rather than a variable and explain why the variable was used rather than a constant.

Constant name .....

Value .....

Why a constant was used .....

.....

.....

Variable name .....

Data type .....

Why a variable was used .....

.....

.....
[6]

(b) Describe how your program set the scores to zero before each round for **Task 1**.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Write an algorithm for **Task 2**, using pseudocode, programming statements or a flowchart.

[6]

(d) Explain how your program completed part of **Task 3** to identify the winner and to display their name and the winning score relative to par. All programming statements that you include must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Section B starts on page 8

Section B

2 An algorithm has been written to:

- input the ages of 100 students
- count and output the number of students aged 7 and under 12
- count and output the number of students aged 12 and under 18
- count and output the number of students aged 18 and over.

(a) Complete the pseudocode algorithm:

```
01  Count7to12 ← 0
02  Count12to18 ← 0
03  CountOver18 ← 0
04  FOR Student ← 1 TO ..........................................
05      OUTPUT "Please enter student's age in years "
06      INPUT Age
07      IF Age >= 7 ..............................................
08        THEN
09          Count7to12 ← Count7to12 + 1
10      ENDIF
11      IF Age >= 12 AND Age < 18
12        THEN
13          Count12to18 ← ......................................
14      ENDIF
15      IF Age >= 18
16        THEN
17          CountOver18 ← CountOver18 + 1
18      ENDIF
19  NEXT Student
20  OUTPUT "There are ", Count7to12, " students aged 7 and under 12."
21  OUTPUT "There are ", Count12to18, " students aged 12 and under 18."
22  OUTPUT "There are ", ................, " students aged 18 and over."
```

[4]

(b) Write the extra pseudocode statements that are needed to count and output the number of students under the age of 7. Use the variable `CountUnder7`; assume `CountUnder7` has already been set to zero.

.....

.....

.....

.....

.....

.....

..... [4]

3 **Four** validation checks and **five** descriptions are shown.

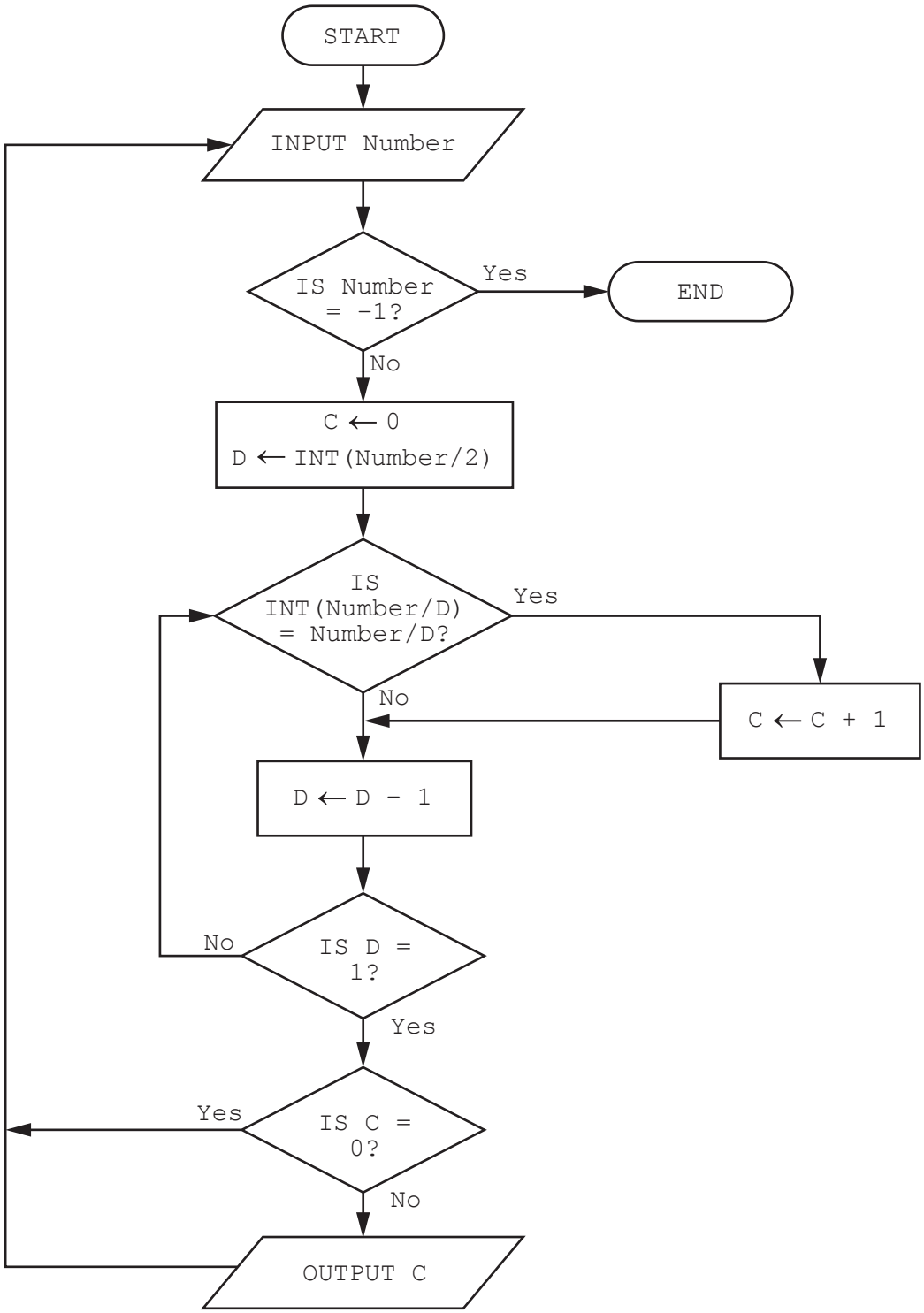
Draw a line from each validation check to the most appropriate description.

Validation check	Description
length check	checks that the data input is between two values
check digit	checks that the data input is an integer
range check	checks that the data input has three digits
type check	checks that the data has been input
	checks that the data input has the correct digits

[4]

4 This flowchart inputs a whole number. The function `INT` returns the integer value of a number. For example, `INT (7.5)` is `7`

An input of `-1` ends the routine.



(a) Complete the trace table for the given algorithm using this input data:
7, 6, 5, 4, –1, 12, 34

Number	C	D	OUTPUT

[6]

(b) Describe the purpose of this algorithm.

.....

.....

.....

..... [2]

(c) **(i)** Describe the problem that occurs if a whole number smaller than 4 and **not** equal to –1 is input.

.....

.....

.....

..... [2]

(ii) Explain how to change the flowchart to prevent this problem occurring.

.....

.....

.....

..... [2]

5 A database table, FLOWER, is used to keep a record of the type of flowers available to make up a bouquet.

FlowerID	Type	Colour	Style	Fragrance
CN001	Carnation	Pink	Stem	Y
CN002	Carnation	Red	Stem	N
CN103	Carnation	White	Stem	N
CN104	Carnation	Yellow	Stem	Y
CN105	Carnation	Pink	Spray	Y
CN106	Carnation	Red	Spray	N
CN107	Carnation	White	Spray	N
CN108	Carnation	Yellow	Spray	Y
RE101	Rose	Pink	Stem	Y
RE102	Rose	Red	Stem	Y
RE103	Rose	White	Stem	N
RE104	Rose	Yellow	Stem	Y
RE105	Rose	Orange	Spray	Y
RE106	Rose	Peach	Spray	N
LY101	Lily	White	Spray	Y

A query-by-example has been written to display just the type, style and colour of all flowers that have no fragrance.

Field:	FlowerID	Fragrance	Style	Colour
Table:	FLOWER	FLOWER	FLOWER	FLOWER
Sort:				
Show:	☑	☑	☑	☑
Criteria:		= Y		
or:				

Explain why the query-by-example is incorrect and write a correct query-by-example.

Explanation .....

.....

.....

.....

.....

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

[6]

BLANK PAGE

BLANK PAGE

BLANK PAGE