Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **0478/22**

Paper 2 Algorithms, Programming and Logic **May/June 2025**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- 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 75.
- 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.

DC (WW/JG) 341855/2

[Turn over

1 Tick (✓) **one** box to complete the description of abstraction.

Abstraction is part of the analysis stage of the development life cycle. Abstraction means

A testing the solution to the problem. ☐

B writing a program using program code. ☐

C defining the sub-systems that make up the problem. ☐

D removing details from the problem that are not relevant. ☐

[1]

2 A fence measuring 2 metres in height and 20 metres in length needs painting. The area of the fence needs to be calculated so that enough paint for the job is purchased.

After the task is decomposed, there are three component parts: input, process and output.

Three component parts and **four** example descriptions are shown.

Draw **one** line from each component part to the most appropriate relevant description.
Not all descriptions will be used.

Component part	Description
process	the height of the fence the length of the fence
output	The amount of paint required is 40 square metres.
input	the colour of the paint
	The area of fence is calculated as 40 square metres.

[3]

3 A range check is used to validate that the input temperature is an integer between –100 and 0, inclusive.

Suggest suitable normal, abnormal, extreme and boundary test data to test if this validation check is working correctly.

Normal test data .....

Abnormal test data .....

Extreme test data .....

Boundary test data .....

[4]

4 **(a)** State **one** reason for using a verification check when inputting data.

.....

..... [1]

(b) Write the pseudocode for an algorithm to perform a double entry verification check, using the following criteria:

- Input 10 numbers and store them in the array `Numbers[]`
- Input the same 10 numbers again; as each number is input, compare it with its corresponding number in the array `Numbers[]`
  - if the numbers match, move on to the next number
  - if the numbers do not match, display both numbers and ask for the number to be re-entered
  - store the re-entered number in the corresponding array location and move on to the next number.
- When all the numbers have been checked, display a message to say the check has been completed.

You do **not** need to declare any arrays or variables for this algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

5 This pseudocode algorithm should allow 500 names of people and their heights to be entered and stored in two one-dimensional (1D) arrays. The name of a person and the height of that person are in the same index in both the `Names[]` and the `Heights[]` arrays.

The algorithm performs as follows:

- at the start, the following data is entered:
  - name of person
  - height of person in cm
- the data is stored in the relevant arrays
- the heights of the people are totalled
- the shortest person is found
- at the end, the following is output:
  - name of shortest person
  - height of shortest person
  - average height of all 500 people.

```
01 DECLARE Names : ARRAY[1:500] OF REAL
02 DECLARE Heights : ARRAY[1:500] OF REAL
03 DECLARE Shortest : REAL
04 DECLARE Total : REAL
05 DECLARE Counter : INTEGER
06 DECLARE Index : INTEGER
07 Shortest ← 500
08 Index ← 0
09 Total ← 100
10 FOR Counter ← 1 TO 500
11     INPUT Names[Index]
12     INPUT Heights[Counter]
13     Total ← Counter + Heights[Counter]
14     IF Heights[Counter] < Shortest
15       THEN
16         Shortest ← Heights[Counter]
17         Index ← Counter
18     ENDIF
19 NEXT Counter
20 OUTPUT "The shortest height is ", Heights
21 OUTPUT "The shortest person is ", Names[Index]
22 OUTPUT "The average height is ", Total / 500
```

(a) Identify the line numbers of **five** errors in the pseudocode and 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 .....

.....

Error 5 line number .....

Correction .....

.....

[5]

(b) Write the pseudocode statement that will output the average height rounded to **one** decimal place.

.....

..... [2]

(c) Explain how you could change the corrected algorithm so that it will also:

- find the height of the tallest person.
- output the height of the tallest person at the end of the algorithm.

Any program statements used must be fully explained.

.....

.....

.....

.....

.....

.....

.....

..... [4]

6 This pseudocode is an algorithm.

```
DECLARE Word : STRING
DECLARE V1 : INTEGER
DECLARE V2 : INTEGER
DECLARE Number : INTEGER
DECLARE Index : INTEGER
DECLARE L1 : CHAR
DECLARE L2 : CHAR
DECLARE Continue : BOOLEAN
INPUT Number
FOR Index ← 1 TO Number
    INPUT Word
    Continue ← TRUE
    V1 ← 1
    V2 ← LENGTH(Word)
    WHILE Continue = TRUE AND V1 < V2 DO
        L1 ← SUBSTRING(Word, V1, 1)
        L2 ← SUBSTRING(Word, V2, 1)
        IF L1 <> L2
          THEN
            Continue ← FALSE
          ELSE
            V1 ← V1 + 1
            V2 ← V2 - 1
        ENDIF
    ENDWHILE
    IF Continue = TRUE
      THEN
        OUTPUT "Successful"
      ELSE
        OUTPUT "NOT successful"
    ENDIF
NEXT Index
```

(a) Complete the trace table for the algorithm, using the input data:

2, RACECAR, TREAT

Number	Index	Word	Continue	V1	V2	L1	L2	OUTPUT

[6]

(b) State the purpose of the algorithm on page **6**.

.....

..... [1]

(c) Explain how the algorithm knows when all the letters in the word have been checked so that it can stop checking.

.....

.....

.....

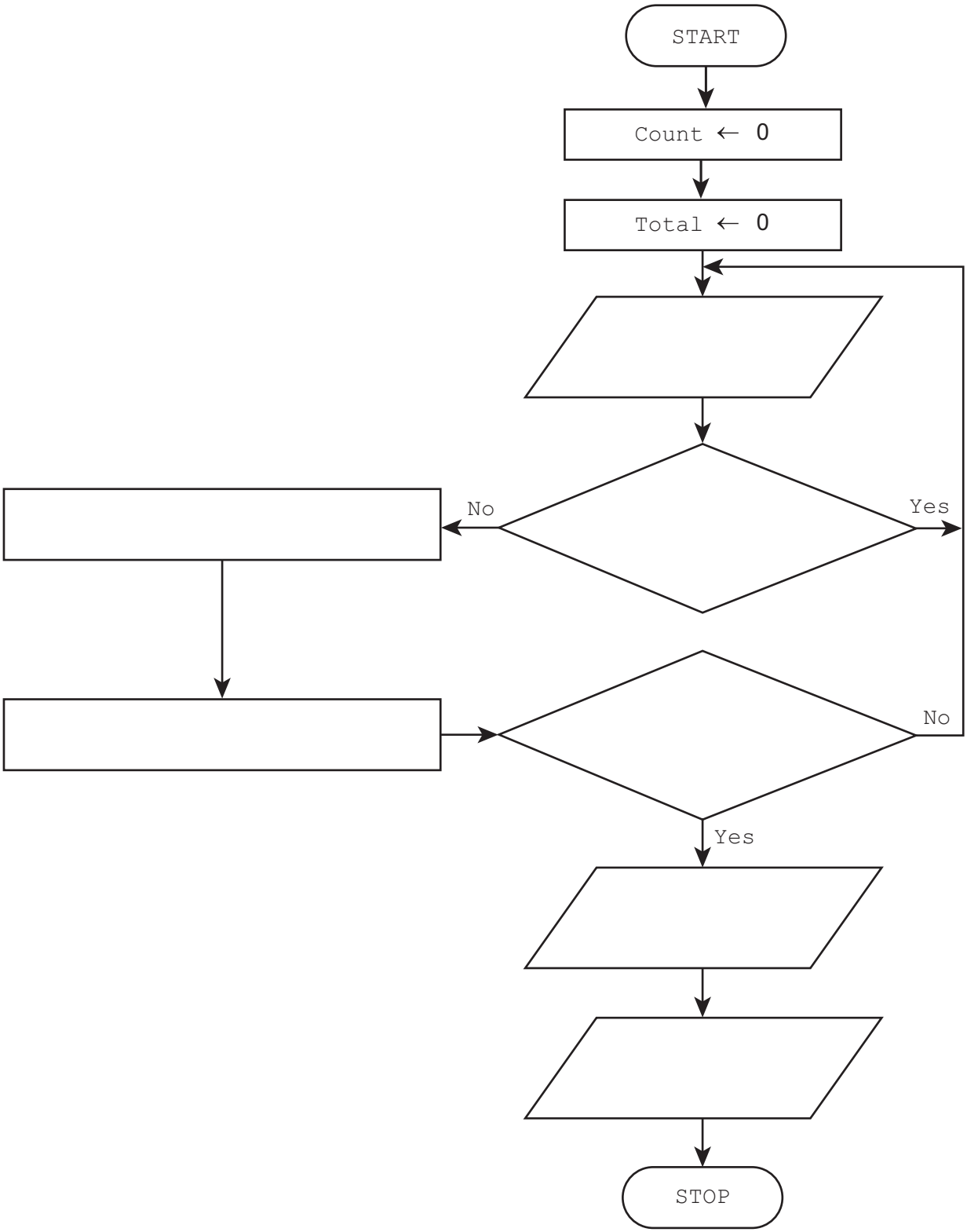
.....

.....

..... [3]

7 The flowchart shows an algorithm that is used to input 50 numbers. Each number is stored in the variable `Value` when it is input. Any number input that is less than or equal to –20 or greater than or equal to 50 is rejected and re-input. The accepted numbers are added together as they are entered and the total and average of these numbers are output at the end of the algorithm.

(a) Complete the flowchart.



[6]

(b) Explain how you would alter the flowchart in **7(a)** so that it will individually store each accepted value as it is entered and then output all the numbers, one after the other, outside of the loop that inputs values.

You may include program statements or flowchart symbols in your answer, but they must be fully explained.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

8 **(a)** Draw the symbol for an XOR logic gate **and** complete the truth table for it.

XOR logic gate symbol:

Truth table:

A	B	X
0	0	
0	1	
1	0	
1	1	

[2]

(b) Consider the logic rules in the following table:

Input	Binary value	Description
A	0	operator **not** present
	1	operator present
B	0	ignition key present
	1	ignition key **not** present
C	0	operator code correct
	1	operator code **not** correct

A crane is a machine with a long arm used to lift heavy objects. The crane will only move:

- if the operator is present and
- either the ignition key is present, or the operator code is correct.

The crane moving is represented by **X = 1**.

(i) Write the logic expression for the crane moving.

X = ... [2]

(ii) Complete the truth table for the crane moving.

A	B	C	Working space	X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

[4]

9 A database table, `CapitalCity`, stores data about capital cities from around the world.

Field	Description
`Identification`	a unique identifier
`Name`	the name of the city
`Country`	the name of the country in which the city is located
`Continent`	the name of the continent in which the city is located
`Population`	the number of people living in the city
`Founded`	the year the city was founded
`AverageIncome`	average income of the city's population, in $, rounded to 2 decimal places
`Subway`	whether or not the city has an underground railway system

(a) Complete the table by writing **one** appropriate field from `CapitalCity` for each data type.

Each field must be different.

Data type	Field
Boolean	
integer	
real	
text	

[2]

(b) Complete the structured query language (SQL) statement to list only the name, country and population of all the cities in `CapitalCity` from the continent of Asia.

```
SELECT .....

..... CapitalCity

WHERE ..... = ..... ;
```

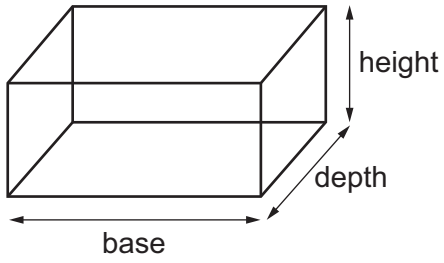
[4]

BLANK PAGE

10 A program is required to allow students to calculate the volume of a number of standard three-dimensional shapes. It will use functions to allow other shapes to be added in the future.

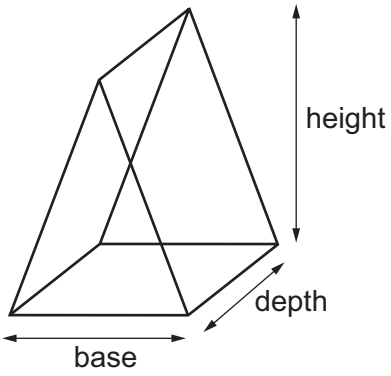
Cuboid

volume = base × depth × height



Triangular prism

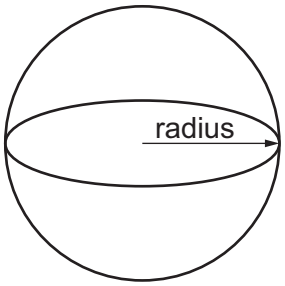
volume = ½ × base × depth × height



Sphere

volume = ⁴⁄₃ × Pi × radius × radius × radius

Pi is a constant with value 3.142



Write a program that meets the following requirements:

- Output a menu to ask the user for an input to indicate the shape for which a volume needs to be calculated, from the three shapes given or if they want to stop the program. The program must validate the input.
- Input the data needed to calculate the volume of the chosen shape, based on the given formulae.
- Create one function for each shape to calculate and return the volume using the given formulae. Each function uses the input data as parameters.
- Output the volume in the main program, rounded to two decimal places.

The program continues until the user inputs the stop option from the menu.

You must use pseudocode or program code **and** add comments to explain how your code works.

All variables and constants used must be declared for this algorithm.

All inputs and outputs must contain suitable messages.

.....

.....

.....

[15]