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.
Calculators must not be used in this paper.

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.
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 Section A Question 1.

Pre-release material

A teacher needs a program to record marks for a class of 30 students who have sat three computer science tests.

Write and test a program for the teacher.

- Your program 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 – Set up arrays

Set-up one dimensional arrays to store:

- Student names
- Student marks for Test 1, Test 2 and Test 3
  - Test 1 is out of 20 marks
  - Test 2 is out of 25 marks
  - Test 3 is out of 35 marks
- Total score for each student

Input and store the names for 30 students. You may assume that the students’ names are unique.

Input and store the students’ marks for Test 1, Test 2 and Test 3. All the marks must be validated on entry and any invalid marks rejected.

TASK 2 – Calculate

Calculate the total score for each student and store in the array.
Calculate the average total score for the whole class.

Output each student’s name followed by their total score.
Output the average total score for the class.

TASK 3 – Select

Select the student with the highest total score and output their name and total score.
1 (a) All variables, constants and other identifiers should have meaningful names.

(i) Declare the array to store the students' names.

(ii) Declare the arrays to store each student's marks and total score.

(b) (i) Show the design of your algorithm to complete Task 1 and Task 2 using pseudocode, programming statements or a flowchart. Do not include any of the validation checks or input prompts in your algorithm.
(ii) Comment on the efficiency of your design.

-------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------


(c) Show two different sets of student data that you could use to check the validation used in Task 1. Explain why you chose each data set.

Set 1
Reason for choice

-------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------

Set 2
Reason for choice

-------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------
(d) (i) Explain how you select the student with the highest score (Task 3). You may include pseudocode or programming statements to help illustrate your explanation.

(ii) How does your program work when there is more than one student having the highest score? Explain using your method given in part (d)(i).
Jatinder uses Internet banking. This pseudocode checks her PIN.

```plaintext
c ← 0
INPUT PIN
x ← PIN
REPEAT
    x ← x/10
    c ← c + 1
UNTIL x < 1
IF c <> 5
    THEN
        PRINT “error in PIN entered”
    ELSE
        PRINT “PIN OK”
ENDIF
```

(a) What value of \( c \) and what message would be output if the following PINs were entered?

- **5 1 0 2 0**
  - Value of \( c \): 5
  - Message: “error in PIN entered”

- **5 1 2 0**
  - Value of \( c \): 4

(b) What type of validation check is being carried out here?

- [1]
3 The flowchart inputs the size of a number of car engines; a value of –1 stops the input.

This information is output: *average engine size* and *number of engines with size > 1.5*

```
START

Engine ← 0
Count ← 0
Number ← 0

INPUT Size

is Size = –1 ?
Yes
Average ← Engine/Number
No

is Size > 1.5 ?
Yes
Count ← Count + 1
No

Number ← Number + 1

Engine ← Engine + Size

OUTPUT
Average, Count

STOP
```
Complete the trace table for the input data.
1.8, 2.0, 1.0, 1.3, 1.0, 2.5, 2.0, 1.3, 1.8, 1.3, –1

<table>
<thead>
<tr>
<th>Engine</th>
<th>Count</th>
<th>Number</th>
<th>Size</th>
<th>Average</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Read this section of program code that inputs twenty (20) numbers and then outputs the largest number input.

```plaintext
1  h = 0
2  c = 0
3  REPEAT
4      READ x
5      IF x > h THEN x = h
6      c = c + 1
7      PRINT h
8  UNTIL c < 20
```

There are **three** errors in this code.

Locate these errors and suggest a corrected piece of code.

1  ----------------------------------------------------------------------------------------------------------------------------------
2  ----------------------------------------------------------------------------------------------------------------------------------
3  ----------------------------------------------------------------------------------------------------------------------------------

[3]
5 A floor turtle uses these instructions.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORWARD $d$</td>
<td>Move $d$ cm forward</td>
</tr>
<tr>
<td>BACKWARD $d$</td>
<td>Move $d$ cm backward</td>
</tr>
<tr>
<td>LEFT $t$</td>
<td>Turn left $t$ degrees</td>
</tr>
<tr>
<td>RIGHT $t$</td>
<td>Turn right $t$ degrees</td>
</tr>
<tr>
<td>REPEAT $n$</td>
<td>Repeat the next set of instructions $n$ times</td>
</tr>
<tr>
<td>ENDREPEAT</td>
<td>End of REPEAT loop</td>
</tr>
<tr>
<td>PENUP</td>
<td>Raise the pen</td>
</tr>
<tr>
<td>PENDOWN</td>
<td>Lower the pen</td>
</tr>
</tbody>
</table>

(Each square in the drawing is 10 cm by 10 cm.)

Start

Finish
Complete the set of instructions to draw the shape (shown in bold lines).

PENDOWN
LEFT 90
REPEAT

..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................
..................................................................

[5]
6 (a) Write an algorithm, using pseudocode or flowchart only, which:

- inputs three numbers
- outputs the largest of the three numbers

(b) Write an algorithm, using pseudocode or flowchart only, which:

- inputs 1000 numbers
- outputs how many of these numbers were whole numbers (integers)
  (You may use INT(x) in your answer, e.g. y = INT(3.8) gives the value y = 3)
(c) Describe, with examples, two sets of test data you would use to test your algorithm.

1. Describe the test data set 1.

2. Describe the test data set 2.

7. A database was set up to show the properties of certain chemical elements. Part of the database is shown below.

<table>
<thead>
<tr>
<th>Name of element</th>
<th>Element symbol</th>
<th>Atomic number</th>
<th>Atomic weight</th>
<th>Melting point (°C)</th>
<th>Boiling point (°C)</th>
<th>State at room temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>oxygen</td>
<td>O</td>
<td>8</td>
<td>16</td>
<td>−218</td>
<td>−183</td>
<td>gas</td>
</tr>
<tr>
<td>iron</td>
<td>Fe</td>
<td>26</td>
<td>56</td>
<td>1538</td>
<td>2861</td>
<td>solid</td>
</tr>
<tr>
<td>mercury</td>
<td>Hg</td>
<td>80</td>
<td>201</td>
<td>−38</td>
<td>356</td>
<td>liquid</td>
</tr>
<tr>
<td>bromine</td>
<td>Br</td>
<td>35</td>
<td>80</td>
<td>−7</td>
<td>59</td>
<td>liquid</td>
</tr>
<tr>
<td>osmium</td>
<td>Os</td>
<td>76</td>
<td>190</td>
<td>3033</td>
<td>5012</td>
<td>solid</td>
</tr>
<tr>
<td>caesium</td>
<td>Cs</td>
<td>55</td>
<td>133</td>
<td>28</td>
<td>671</td>
<td>solid</td>
</tr>
<tr>
<td>gallium</td>
<td>Ga</td>
<td>31</td>
<td>70</td>
<td>30</td>
<td>2204</td>
<td>solid</td>
</tr>
<tr>
<td>argon</td>
<td>Ar</td>
<td>18</td>
<td>40</td>
<td>−189</td>
<td>−186</td>
<td>gas</td>
</tr>
<tr>
<td>silver</td>
<td>Ag</td>
<td>47</td>
<td>108</td>
<td>961</td>
<td>2162</td>
<td>solid</td>
</tr>
</tbody>
</table>

(a) How many fields are in each record?

(b) The following search condition was entered:

\[(\text{Melting point (°C) < 40}) \text{ AND } (\text{Atomic weight > 100})\]

Using **Element symbol** only, which records would be output?

(c) Which field would be best suited as primary key?