READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name in the spaces at the top of the previous 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.

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.
A computer shop will build a computer from components to meet a customer’s requirements. For each request for a computer to be built, an estimate of the cost is produced. The component stock level is checked; if all the components are in stock, a firm order to build the computer can be placed. A program is required to work out the cost of the computer, update the stock levels and provide a daily summary of orders for the shop owner.

Write and test a program or programs for the computer shop owner.

• 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 – Produce an estimate.

Write a program for TASK 1 to calculate the cost of building a computer using these components.

<table>
<thead>
<tr>
<th>COMPONENT</th>
<th>CHOICES</th>
<th>PRICES IN $</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>p3 / p5 / p7</td>
<td>100 / 120 / 200</td>
</tr>
<tr>
<td>RAM</td>
<td>16 GB / 32 GB</td>
<td>75 / 150</td>
</tr>
<tr>
<td>Storage</td>
<td>1 TB / 2 TB</td>
<td>50 / 100</td>
</tr>
<tr>
<td>Screen</td>
<td>19&quot; / 23&quot;</td>
<td>65 / 120</td>
</tr>
<tr>
<td>Case</td>
<td>Mini Tower / Midi Tower</td>
<td>40 / 70</td>
</tr>
<tr>
<td>USB ports</td>
<td>2 ports / 4 ports</td>
<td>10 / 20</td>
</tr>
</tbody>
</table>

The customer makes a choice for each component and an estimate is produced. The estimate must show a unique estimate number, the components chosen and the price of each component. The estimate must also show the total cost of the computer, which is calculated as the sum of the cost of the components chosen plus 20%.

TASK 2 – Place an order.

Using your estimate from TASK 1, check if the components required are in stock. If all the components are in stock then update the stock levels. Add the unique estimate number to the list of order numbers. Add the customer’s details and today’s date to the estimate details to finalise the order. Print two copies of the order, one for the customer and one for the shop.

TASK 3 – Summarise the day’s orders.

Extend TASK 2 to provide an end of day summary showing the number of orders made, the total number of each component sold and the value of the orders.
1 (a) All variables, constants and other identifiers should have meaningful names.

(i) You recorded information for the estimate of the cost of building a computer in TASK 1. Give a data structure that you created for TASK 1, its name, data type and use.

Data structure ______________________________

Name ___________________________________________

Data type _______________________________________

Use ____________________________________________

______________________________________ [4]

(ii) Describe the data structures that you have used in TASK 2 to record the customer details. Include sample data in the description.

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________ [4]
(b) Explain how your program for TASK 1 produces a unique estimate number.

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________ [2]
(c) Write an algorithm for part of TASK 2 to check that the chosen processor and chosen RAM are in stock, using EITHER pseudocode, programming statements OR a flowchart. Assume that TASK 1 has been completed. Do NOT check the other components or produce the order.
(d) Explain how your program completes TASK 3. Any programming statements used in your answer must be fully explained.
2  (a) Write an algorithm to input 1000 numbers. Count how many numbers are positive and how many numbers are zero. Then output the results. Use EITHER pseudocode OR a flowchart.
(b) Give one change you could make to your algorithm to ensure initial testing is more manageable.
QUESTION 3 STARTS ON PAGE 15.
3 The global trade item number (GTIN-8) barcode has seven digits and a check digit. The pseudocode algorithm on page 16 inputs seven digits and calculates the eighth digit, then outputs the GTIN-8.

DIV(X,Y), finds the number of divides in division for example DIV(23,10) is 2.
MOD(X,Y), finds the remainder in division for example MOD(23,10) is 3.
FOR Count ← 1 TO 7
    INPUT Number
    Digit(Count) ← Number
NEXT
Sum ← (Digit(1)+Digit(3)+Digit(5)+Digit(7))*3+Digit(2)+Digit(4)+Digit(6)
IF MOD(Sum,10) <> 0
    THEN Digit(8) ← DIV(Sum,10)*10 + 10 - Sum
    ELSE Digit(8) ← 0
ENDIF
OUTPUT "GTIN-8"
FOR Count ← 1 TO 8
    OUTPUT Digit(Count)
NEXT
(a) Complete the trace table for the input data: 5, 7, 0, 1, 2, 3, 4

<table>
<thead>
<tr>
<th>Digit(1)</th>
<th>Digit(2)</th>
<th>Digit(3)</th>
<th>Digit(4)</th>
<th>Digit(5)</th>
<th>Digit(6)</th>
<th>Digit(7)</th>
<th>Digit(8)</th>
<th>Sum</th>
<th>OUTPUT</th>
</tr>
</thead>
</table>

Complete the trace table for the input data: 4, 3, 1, 0, 2, 3, 1

<table>
<thead>
<tr>
<th>Digit(1)</th>
<th>Digit(2)</th>
<th>Digit(3)</th>
<th>Digit(4)</th>
<th>Digit(5)</th>
<th>Digit(6)</th>
<th>Digit(7)</th>
<th>Digit(8)</th>
<th>Sum</th>
<th>OUTPUT</th>
</tr>
</thead>
</table>

[5]
Copy of algorithm

FOR Count ← 1 TO 7
    INPUT Number
    Digit(Count) ← Number
NEXT
Sum ← (Digit(1) + Digit(3) + Digit(5) + Digit(7)) * 3 + Digit(2) + Digit(4) + Digit(6)
IF MOD(Sum, 10) <> 0
    THEN Digit(8) ← DIV(Sum, 10) * 10 + 10 - Sum
    ELSE Digit(8) ← 0
ENDIF
OUTPUT "GTIN-8"
FOR Count ← 1 TO 8
    OUTPUT Digit(Count)
NEXT
(b) Explain how you would change the algorithm to input eight digits (seven digits and the check digit) and output if the check digit entered is correct or not.

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________ [3]
A programmer has written a routine to check that prices are below $10.00. These values are used as test data.

10.00  9.99  ten

Explain why each value was chosen.

10.00

9.99

ten

[3]
5 Explain the difference between the programming concepts of COUNTING and TOTALLING. Include an example of a programming statement for each concept in your explanation.
A database table, PERFORMANCE, is used to keep a record of the performances at a local theatre.

<table>
<thead>
<tr>
<th>Show Number</th>
<th>Type</th>
<th>Title</th>
<th>Date</th>
<th>Sold Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN091</td>
<td>Comedy</td>
<td>An Evening at Home</td>
<td>01 Sept</td>
<td>Yes</td>
</tr>
<tr>
<td>SN102</td>
<td>Drama</td>
<td>Old Places</td>
<td>02 Oct</td>
<td>No</td>
</tr>
<tr>
<td>SN113</td>
<td>Jazz</td>
<td>Acoustic Evening</td>
<td>03 Nov</td>
<td>No</td>
</tr>
<tr>
<td>SN124</td>
<td>Classical</td>
<td>Mozart Evening</td>
<td>04 Dec</td>
<td>Yes</td>
</tr>
<tr>
<td>SN021</td>
<td>Classical</td>
<td>Bach Favourites</td>
<td>01 Feb</td>
<td>Yes</td>
</tr>
<tr>
<td>SN032</td>
<td>Jazz</td>
<td>30 Years of Jazz</td>
<td>02 Mar</td>
<td>Yes</td>
</tr>
<tr>
<td>SN043</td>
<td>Comedy</td>
<td>Street Night</td>
<td>03 Apr</td>
<td>No</td>
</tr>
<tr>
<td>SN054</td>
<td>Comedy</td>
<td>Hoot</td>
<td>04 May</td>
<td>No</td>
</tr>
</tbody>
</table>

(a) State the number of fields and records in the table.

Fields

Records [2]
(b) Give TWO validation checks that could be performed on the Show Number field.

Validation check 1 ____________________________

__________________________________________

Validation check 2 ____________________________

__________________________________________

[2]

(c) Using the query-by-example grid, write a query to identify jazz performances that are not sold out. Only display the date and the title.

<table>
<thead>
<tr>
<th>Field:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Table:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sort:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>or:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[4]