



**X816/77/11**

**Computing Science**

Duration — 2 hours 30 minutes

---

**Total marks — 70**

**SECTION 1 — 15 marks**

Attempt ALL questions.

**SECTION 2 — 55 marks**

Attempt ALL questions.

**You may use a calculator.**

Write your answers clearly in the answer booklet provided. In the answer booklet you must clearly identify the question number you are attempting.

Use **blue** or **black** ink.

Before leaving the examination room you must give your answer booklet to the Invigilator; if you do not, you may lose all the marks for this paper.



\* X 8 1 6 7 7 1 1 \*

## SECTION 1 — 15 marks

Attempt ALL questions

1. A dynamic website can display statistical information based on historical weather data collected from a weather station.

Weather					
year	month	maxTemp	minTemp	rainfall	sunHours
2018	1	6	1.2	175.1	39.7
2018	2	6.4	0.1	67.6	94.5
2018	3	7.2	1.3	70.4	87.4
2018	4	12	5.2	70.1	116
2018	5	17.7	7.8	27.6	232.8
2018	6	20.9	10.5	59	252.4
2018	7	21.6	12.7	60	218.8
2018	8	18.1	11.5	81.7	110.6
2018	9	15.5	8.9	100.6	114.9
2018	10	13	6.2	104.9	94.1
2018	11	10.1	4.7	148.1	46.6
2018	12	12	-3.1	98.3	28

Draw the output generated by the following server-side code.

2

```

$sql = "SELECT month, sunHours FROM Weather WHERE sunHours
BETWEEN 200 AND 250";
$result = mysqli_query($conn, $sql);
echo "<table><tr><th> Month </th><th> Hours of Sunshine
</th></tr>";
while($row = mysqli_fetch_array($result)) {
    echo "<tr>
        <td>" . $row['month'] . " </td>
        <td>" . $row['sunHours'] . " </td>
    </tr>";
}
echo "</table>";

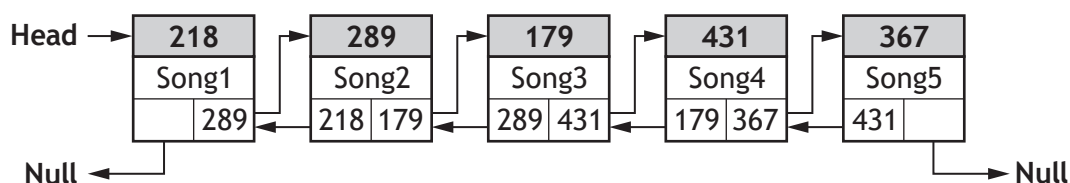
```

2. A DJ app allows users to create a playlist. The playlist is implemented using a double linked list.

The app provides an ADD SONG option.

When users select the ADD SONG option, a new node is added to the end of the playlist.

An example playlist is shown below. The head of this double linked list points to the node stored at memory location 218.



Describe the operations carried out on the double linked list when selecting the ADD SONG option.

3

3. The structure of a relational database is shown below.

Item	Order	OrderItem	Customer
<u>itemID</u>	<u>orderID</u>	<u>orderID *</u>	<u>customerNo</u>
itemName	orderDate	<u>itemID *</u>	customerName
price	customerNo *	<u>date</u>	email
stock		quantity	password

A sample record from the Order table is shown below.

orderID	orderDate	customerNo
245427	2021-05-04	SHG46

The following restrictions apply to the Order table.

- A unique orderID is automatically generated by the system whenever a new order is received
- When customers open an account with the company, they are allocated a unique customerNo that consists of 5 characters
- All orders must have an orderDate

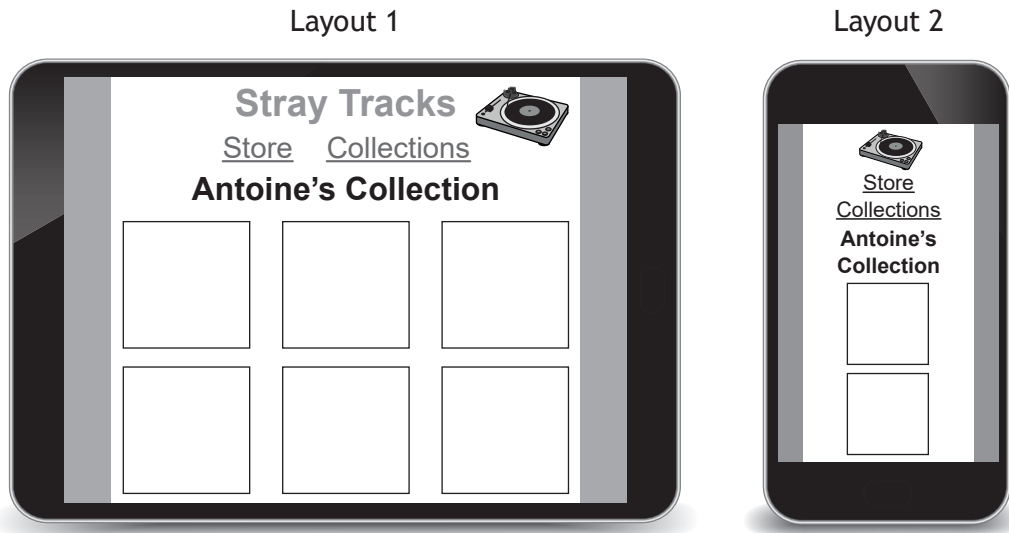
Write the SQL statement used to create the structure of the Order table.

3

[Turn over

4. A new website called Stray Tracks allows music fans to store and share details of their music collections with other users of the website.

When browsing a user’s music collection, a gallery of album covers is displayed. The gallery layout is controlled using a media query.



Some of the incomplete CSS and HTML code used to produce these layouts is shown below.

**CSS**

```

...
body{margin:auto;}
nav {height:35px;}
h1 {display:inline; font-size:28px;}
nav ul li {width:150px; height:20px; font-size:16pt;
float:left;}
@media screen and (max-width: 460px) {
    nav {height:70px;}
    nav ul li {width:100%; font-size:8pt;}
    ...
}

```

**HTML**

```

<header><h1 id="mainHeading"> Stray Tracks </h1></header>
<nav><ul><li> <a href="store.html"> Store </a></li>
<li><a href="collections.html"> Collections </a></li></ul>
</nav>

```

- (a) Describe when the media query would be used to generate Layout 2. 1
- (b) Explain why the `nav height` value is larger in the media query than in the rule `nav {height:35px;}`. 1
- (c) Explain, making reference to the main heading, Stray Tracks, why the CSS is incomplete. 1

5. A program is used to process the details of 5000 marathon runners. Sample runner details are shown below.

runnerName	runnerNumber	timeMins
Ryan Smith	2461	242
Graham Philips	1892	233
Ruth Sturrock	3781	207
Elizabeth Ferry	196	275

This program processes the details using an array of records based on the following record structure and array declaration.

```
RECORD Runner IS {STRING runnerName, INTEGER runnerNumber,
INTEGER timeMins}
```

```
DECLARE athletes AS ARRAY OF Runner INITIALLY [{runnerName =
"name", runnerNumber = 0, timeMins = 0}] * 5000
```

The incomplete pseudocode below uses the bubble sort algorithm to sort the `athletes` array into ascending order of time.

1. set `n = 4999`
2. start conditional loop
3. set `swapped = false`
4. start fixed loop `i = 0` to `n-1`
5. if \_\_\_\_\_ > \_\_\_\_\_ then
6. swap runner details
7. set `swapped = true`
8. end if
9. end fixed loop
10. set `n = n - 1`
11. end conditional loop when `swapped = false`

- (a) Rewrite line 5 of the algorithm to complete the condition. 2
- (b) Explain why the variable `n` has been used in this sort. 2

[END OF SECTION 1]

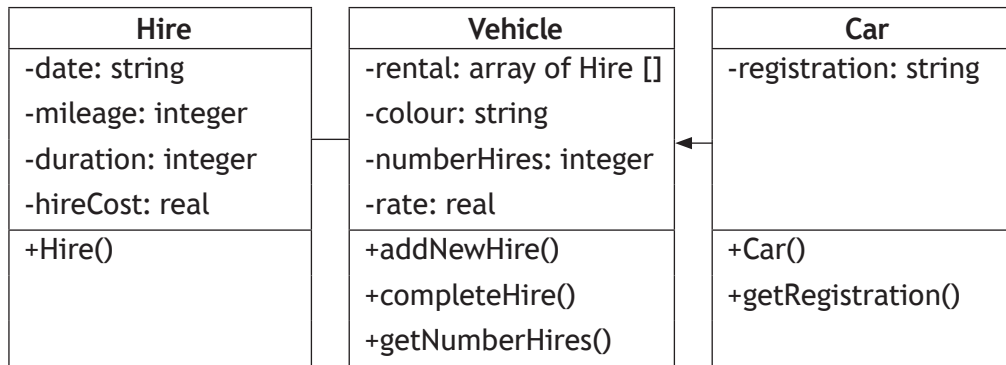
[Turn over

## SECTION 2 — 55 marks

### Attempt ALL questions

6. GlasgowGreen uses an object-oriented program to track the use of hire vehicles across the city.

A simplified UML class diagram for the program is shown below.



Some of the object-oriented code used in the program is shown below.

```
Line 1  CLASS Vehicle IS { ARRAY OF Hire rental,  STRING colour,
        INTEGER numberHires, REAL rate }
Line 2      METHODS
Line 3      PROCEDURE addNewHire (STRING date)
Line 4          SET THIS.rental[THIS.numberHires].date TO date
Line 5          SET THIS.rental[THIS.numberHires].mileage TO 0
Line 6          SET THIS.rental[THIS.numberHires].duration TO 0
Line 7          SET THIS.rental[THIS.numberHires].hireCost TO 0.0
Line 8      END PROCEDURE

Line 9      PROCEDURE completeHire (INTEGER mileage, INTEGER
        duration)
Line 10         SET THIS.rental[THIS.numberHires].duration TO
        duration
Line 11         SET THIS.rental[THIS.numberHires].mileage TO mileage
Line 12         SET THIS.rental[THIS.numberHires].hireCost TO
        THIS.rate * THIS.rental[THIS.numberHires].mileage
Line 13         SET THIS.numberHires TO THIS.numberHires + 1
Line 14      END PROCEDURE

...
Line 23  END CLASS
```

6. (continued)

```

Line 24 CLASS Car INHERITS Vehicle WITH { STRING registration }
Line 25     METHODS
Line 26     CONSTRUCTOR ( STRING registration, STRING colour )
Line 27         DECLARE THIS.registration INITIALLY registration
Line 28         DECLARE THIS.rate INITIALLY 7.25
Line 29         DECLARE THIS.rental AS ARRAY OF Hire INITIALLY[]
Line 30         DECLARE THIS.colour INITIALLY colour
Line 31         DECLARE THIS.numberHires INITIALLY 0
Line 32     END CONSTRUCTOR
...

```

A `Car` object is instantiated using the following statement.

```

DECLARE car1 AS Car INITIALLY Car ("ABC123", "Red")

```

- (a) State what is stored in all of the instance variables of the `car1` object upon instantiation. 1
- (b) Using appropriate object-oriented terminology, explain why the following statement would be invalid. 2

```

SET car1.registration TO "DEF456"

```

- (c) Vehicle hire across the city has proved to be very popular and GlasgowGreen has decided to expand the business by introducing pedal bikes for hire.
 

A second subclass of the `Vehicle` class called `Bike` must be added to the program.

  - (i) Explain why the use of subclasses with inheritance will reduce the time needed to add the `Bike` class to the program. 1
  - (ii) The `Bike` class will have one additional private property, `bikeID`, which is a string value, and three additional public methods called `Bike()`, `completeHire()` and `getBikeID()`.
 

Complete the UML class diagram to include the `Bike` class, showing all properties and methods. You need only display the name of each class in the existing UML class diagram. 2

[Turn over

## 6. (c) (continued)

(iii) The `completeHire()` method of the `Bike` class is shown below.

```

Line 57      OVERRIDE PROCEDURE completeHire (INTEGER mileage,
              INTEGER duration)
Line 58          SET THIS.rental[THIS.numberHires].duration TO
              duration
Line 59          SET THIS.rental[THIS.numberHires].mileage TO
              mileage
Line 60          SET THIS.rental[THIS.numberHires].hireCost TO
              THIS.rate * THIS.rental[THIS.numberHires].
              duration
Line 61          SET THIS.numberHires TO THIS.numberHires + 1
Line 62      END PROCEDURE

```

Explain the use made of polymorphism in this situation.

2

(d) The variable `bikeArray` is an array of `Bike` objects. This array stores the details of the 125 bikes owned by `GlasgowGreen`.

`GlasgowGreen` wants to know the ID of the bike that has been hired most often.

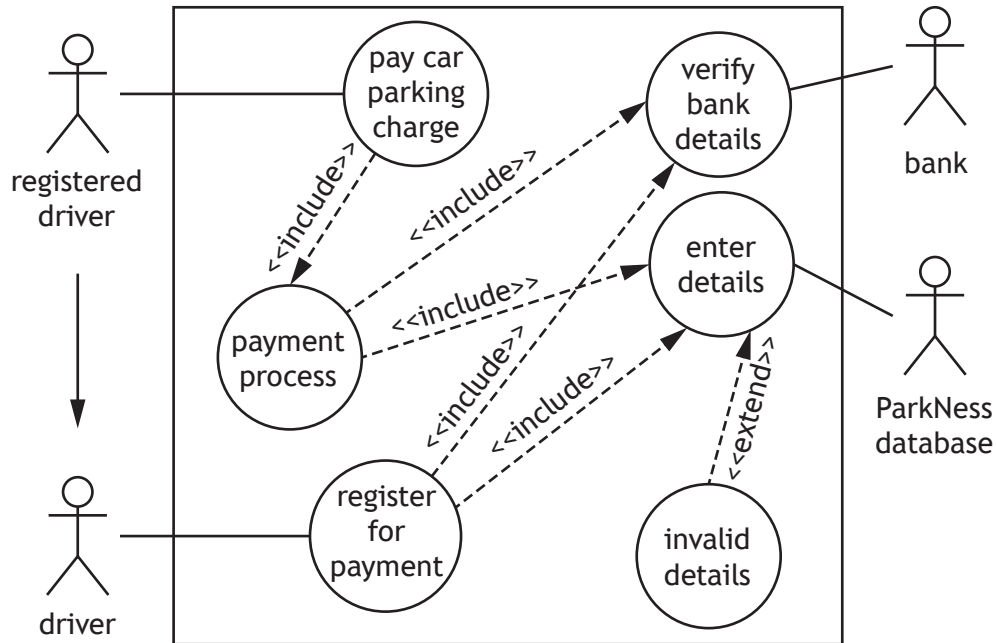
Using a programming language of your choice, write the code to find and display the ID of the bike that has been hired most often.

3

7. ParkNess manages six car parks in Inverness city centre.

Drivers can use the ParkNess app to pay for their parking by providing the licence plate of their car and details of their bank payment card.

The UML use case diagram below illustrates the functionality of the ParkNess app.



- (a) (i) Describe the relationship between the actors 'driver' and 'registered driver'. 1
- (ii) Describe the use of the 'extend' relationship between the 'enter details' and 'invalid details' use cases. 1

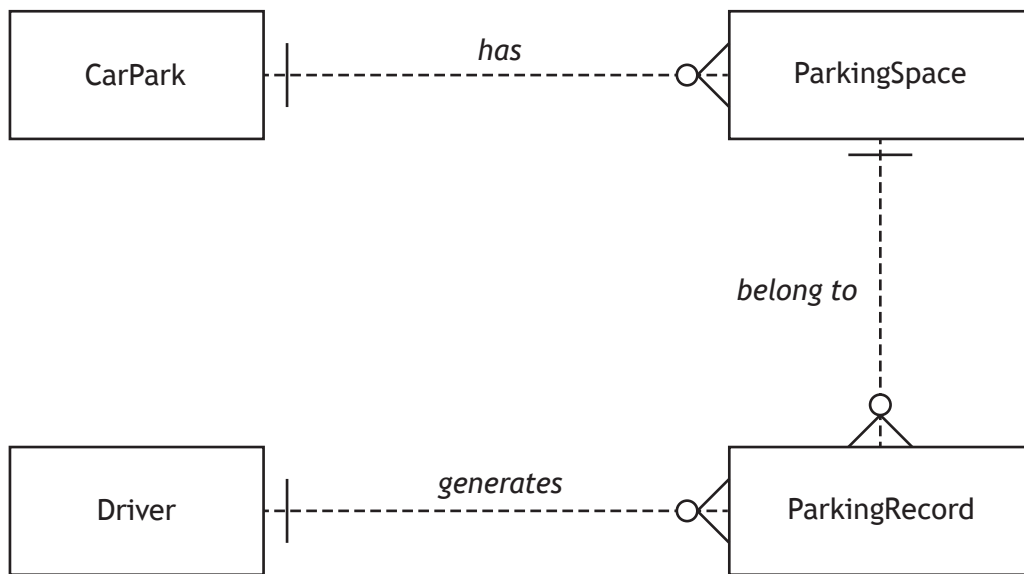
[Turn over

7. (continued)

ParkNess makes use of Automatic Number Plate Recognition to record the licence plates of parked cars. Each parking space has a sensor that records whether it is occupied. The records for every parking space in the car parks are stored in a relational database. The structure of the database is shown below.

CarPark	ParkingSpace	ParkingRecord	Driver
<u>carParkID</u>	<u>spaceID</u>	licencePlate	<u>driverID</u>
carParkName	carParkID *	<u>entryDate</u>	email
capacity	occupied	<u>entryTime</u>	paymentCardNumber
		departureTime	expiry
		<u>spaceID *</u>	
		cost	
		<u>driverID *</u>	

(b) The entity-relationship diagram for this database is shown below.



- (i) State whether each entity in this entity-relationship diagram is strong or weak. 2
- (ii) Describe the relationship participation between the `ParkingSpace` and `ParkingRecord` entities. 2

## 7. (continued)

Sample data to be stored in each table of the database is shown below.

CarPark		
carParkID	carParkName	capacity
cp1	Rose Street	95
cp2	Telford Street	100
cp3	Bank Street	150
cp4	Clachnaharry Road	80
cp5	Ness Walk	64
cp6	Stadium Road	144

ParkingSpace		
spaceID	carParkID	occupied
33687	cp4	0
41722	cp2	1
...	...	...

*Note that a value of 0 in the `occupied` field indicates that the space is unoccupied.*

Driver			
driverID	email	paymentCardNumber	expiry
145	stevenm@inverness.co.uk	1111222233334444	08/21
158	daveb@ness.com	2222111133334444	09/21
193	melanieph@north.com	3333444411112222	10/21
...	...	...	...

ParkingRecord						
licencePlate	entryDate	entryTime	departureTime	spaceID	cost	driverID
SM68 JIX	2021-04-08	08:55	15:18	33687	10.50	145
SS16 PVM	2021-04-09	11:12	19:50	54256	13.50	158
DB19 SJS	2021-04-07	18:08	20:19	12346	4.50	193
SM68 JIX	2021-04-09	09:12	16:22	36027	12.00	145
SJ19 DBM	2021-04-07	09:43	17:02	12346	12.00	127
MI17 BSS	2021-04-09	07:22	18:17	65431	13.50	149
GC15 SNE	2021-04-08	11:10	13:18	12346	4.50	137
...	...	...	...	...	...	...

(c) Describe the output produced from the following query.

1

```
SELECT spaceID
FROM ParkingSpace
WHERE NOT carParkID = "cp4"
AND occupied = 0;
```

[Turn over

## 7. (continued)

- (d) (i) The SQL query below is used to display details of those drivers who parked in a ParkNess car park at least 10 times during April 2021.

```
SELECT email, licencePlate, COUNT(*) AS [Number of Occasions]
FROM Driver, ParkingRecord
WHERE Driver.driverID = ParkingRecord.driverID
AND entryDate BETWEEN "2021-04-01" AND "2021-04-30"
GROUP BY email, licencePlate
_____ ;
```

Write the missing condition needed to complete this query.

1

- (ii) Andrew works near the Rose Street and Bank Street car parks.

The SQL query below is used to show the name of these two car parks along with the number of available spaces in each car park.

```
SELECT carParkName, capacity - COUNT(*) AS [Spaces Available]
FROM CarPark, ParkingSpace
WHERE CarPark.carParkID = ParkingSpace.carParkID
AND _____
GROUP BY carParkName;
```

Using the `IN` operator, write a condition needed to complete the `AND` clause of this query.

1

7. (continued)

- (e) ParkNess uses a SQL query to display details of individual parking spaces from all car parks other than the Ness Walk car park.

Each parking space displayed by the query has a total income which is more than twice the income generated from all the parking spaces in the Ness Walk car park that were used between 14 and 26 April 2021 inclusive.

- (i) Complete the design of this query by writing the missing parts labelled A, B and C of the query design below.

3

Field(s) / Calculation(s)		spaceID, carParkName, totalIncome = sum(cost)		
Table(s)		CarPark, ParkingSpace, ParkingRecord		
Search Criteria		A		
Grouping		spaceID, carParkName		
Having	B	Inner Query	Field(s) / Calculation(s)	C
			Table(s)	ParkingSpace, ParkingRecord, CarPark
			Search criteria	entryDate between 2021-04-14 and 2021-04-26
			AND	carParkName = "Ness Walk"
Sort Order		spaceID ASC		

- (ii) Once the query has been executed, the 267 sorted results are imported into a program for further analysis.

The data is read into an array of records with the structure shown below.

```
RECORD space IS {INTEGER spaceID, STRING carParkName,
REAL totalIncome}
DECLARE spaceList AS ARRAY OF space INITIALLY
[0, "name", 0.0] * 267
```

Use pseudocode to design a binary search algorithm that will prompt the user to enter the ID of a space, search the spaceList array and display the total income for that space.

4

[Turn over

8. Green’s Autos website allows customers to search for details of cars. The page below shows the results of one such search.

Car Sales

**Green’s Autos**

**Yotoya Rocrolla**



**£12,999**

To request a test drive for the car being viewed, the customer clicks the ‘Request a test drive’ button. This loads a new page with the online form shown below. The customer then provides each of the details in the form.

Car Sales

**Green’s Autos**

**Request a test drive**

Name

Email

Date

Make Yotoya

Model Rocrolla

- (a) (i) Write the HTML `<form>` statement used to submit the form details securely to the server-side script called `requestConfirm.php`.

## 8. (a) (continued)

- (ii) Write the HTML statement used to generate the part of the form shown below.

1

Date	<input type="text" value="yyyy-mm-dd"/>
------	---

- (b) (i) The PHP script `requestConfirm.php` generates a separate 'Confirm Page' of the website shown below. This is used to confirm that the customer's test drive request has been received.

Car Sales
<h2 style="margin: 0;">Green's Autos</h2> <h3 style="margin: 0;">Thank you</h3> <p style="margin: 10px 0;">Dear Melanie Peters</p> <p style="margin: 10px 0;">Thank you for requesting a test drive of a Yotoya Rocrolla.</p> <p style="margin: 10px 0;">We will check availability on 22/06/2021 and send an email to mp27@email.com to confirm your booking.</p> <p style="margin: 10px 0;">Click below to return to the car page.</p> <div style="text-align: center; margin: 10px 0;"> <input type="button" value="Return to car page"/> </div>

Explain the term session variables, making reference to the page above.

2

- (ii) The PHP script `requestConfirm.php` also forms a connection with the secure database server by performing the following tasks.

- (1) Assign the connection details to server-side variables
- (2) Create a connection called `$conn` with the secure database

Write the PHP code required to perform these two tasks. Your code should make use of the connection details below.

- `server name = "serv1a";`
- `username = "root";`
- `password = "AQS20";`
- `database = "TestDrive ";`

2

8. (continued)

(c) (i) Once a connection has been created, details of a customer’s test drive request are assigned to three variables.

- \$customerName
- \$email
- \$startDate

The values in these variables will then be stored in a database table called `ClientTD48`. The structure of this table is shown below.

Attribute	Type	Size	Key	Required	Validation
id	int	4	PK	Y	Auto increment
customerName	varchar	50		Y	
email	varchar	30		Y	
startDate	date			Y	
requestStatus	varchar	12		Y	Restricted choice: one of pending, successful or unsuccessful

A single query is used to

- store the details of a customer’s test drive request in a new record of the `ClientTD48` table
- assign the initial value ‘pending’ to the `requestStatus` field of this new record.

Write the SQL statement for this query.

2

## 8. (c) (continued)

- (ii) The database table called `ClientTD48` stores details of all test drive requests received. Sample data stored in this table is shown below.

clientTD48				
id	customerName	email	startDate	requestStatus
...	...	...	...	...
1732	Norma Banks	nbanks@me.com	2021-06-21	successful
1733	Charlie Babbage	cbab9@tmail.com	2021-06-21	unsuccessful
1734	Sylvan Broad	brs@email.com	2021-06-22	pending
1735	Mo Ali	ma13@me.com	2021-06-22	pending
1736	Joanne Leach	lj12@email.com	2021-06-22	pending
1737	Melanie Peters	mp27@email.com	2021-06-22	pending
1738	Thomas Murtagh	murt@tmail.com	2021-06-22	pending
1739	Marge Smith	ma14@imail.com	2021-06-22	pending
1740	Scott Curran	scus@me.com	2021-06-22	pending
...	...	...	...	...

PHP code is used to display details of all customers who have requested a test drive on 22 June 2021.

Some of the code used to provide this functionality is shown below.

```

...
Line 20  $sql = "SELECT * FROM ClientTD48 WHERE
        startDate = '2021-06-22' ";
Line 21  $result = mysqli_query($conn, $sql);
Line 22  _____
Line 23  echo "Number of test drives requests
        found is " . $rowcount
...

```

This output from this code is shown below.

```

        Number of test drive requests found is 7

```

Using the `mysqli_num_rows()` function, write the missing code required at Line 22 to calculate the total number of customers who have requested a test drive.

8. (c) (continued)

(iii) Line 21 on the previous page generates the variable `$result` which contains the following data.

id	customerName	email	startDate	requestStatus
1734	Sylvan Broad	brs@email.com	2021-06-22	pending
1735	Mo Ali	ma13@me.com	2021-06-22	pending
1736	Joanne Leach	lj12@email.com	2021-06-22	pending
1737	Melanie Peters	mp27@email.com	2021-06-22	pending
1738	Thomas Murtagh	murt@tmail.com	2021-06-22	pending
1739	Marge Smith	ma14@imail.com	2021-06-22	pending
1740	Scott Curran	scus@me.com	2021-06-22	pending

A maximum of five test drives is permitted on any given date.

Server-side code is used to set the `requestStatus` field of the `ClientTD48` table to 'successful' for the first five requests for any given date or 'unsuccessful' for all additional requests.

Using pseudocode, design an algorithm to process the `$result` variable and amend the `requestStatus` field of each relevant record in the `ClientTD48` table appropriately.

3

(d) All stored session data associated with a test drive request should be deleted as soon as the user has completed the session.

State how this can be achieved.

1

[Turn over for next question

**DO NOT WRITE ON THIS PAGE**

9. A block of flats has several software-controlled lifts to access all eight floors of the building. Users of the lifts have complained about waiting times. The developers of the existing control software have been asked to update the software to reduce waiting times for lift users.

(a) (i) Time estimates for the analysis and design stages of this project are listed below.

- Carry out feasibility study — 2 days
- Produce requirements specification — 3 days
- Design of data structures — 2 days
- Design of software logic — 4 days

Both stages of design should take place at the same time.

Draw a Gantt chart for the analysis and design stages of the project.

2

(ii) Explain why economic feasibility must be considered before proceeding with any project.

1

(b) Lifts in the building may be descending, ascending or stationary. In the example below, Lift 1 is ascending, Lift 3 and Lift 4 are descending and Lift 2 is stationary.

	Lift 1	Lift 2	Lift 3	Lift 4
7th Floor				
6th Floor				
5th Floor	A		D	
4th Floor				
3rd Floor				
2nd Floor				D
1st Floor				
Ground		S		

A = Ascending

D = Descending

S = Stationary

One possible implementation of the updated control software would make use of a 2-D array.

(i) Using a programming language of your choice, declare a 2-D array called `lifts` to store the lift details.

2

(ii) Write code to assign the lifts to the four locations indicated above.

1

9. (b) (continued)

- (iii) Code will be used to find the nearest lift for any floor whenever the up or down button is pressed by the user.

The user inputs will be stored in two variables, `floor` and `direction`. Details of these variables are provided in the table below.

Variable Name	Possible Values	Purpose
<code>floor</code>	0 to 7	indicates floor user is on when the button is pressed
<code>direction</code>	UP or DOWN	indicates whether the user wants to go up or down

The software will calculate how far away all suitable lifts are from the user’s floor and store these results in a 1-D array called `distance`. Unsuitable lifts will be assigned the value 99.

Using the lift details in part (b), when a user on the third floor presses the DOWN button the following logic is applied.

Lift	Suitability	Reason	Assigned value
1	Unsuitable	above the user and travelling upwards	99
2	Suitable	three floors below the user and stationary	-3
3	Suitable	two floors above the user and descending	2
4	Unsuitable	below the user and travelling downwards	99

These results would be stored in a 1-D array called `distance`.

<code>distance</code>	99	-3	2	99
-----------------------	----	----	---	----

Using pseudocode, design an algorithm to calculate and store the correct results in the `distance` array for a single user, on any floor, travelling in either direction.

4

- (iv) Using a programming language of your choice, write code that will find and display the number of the nearest lift using the values in the `distance` array.

2

- (c) Describe how final testing of this software could be carried out.

1

[END OF QUESTION PAPER]

[BLANK PAGE]

DO NOT WRITE ON THIS PAGE

[BLANK PAGE]

DO NOT WRITE ON THIS PAGE

[BLANK PAGE]

DO NOT WRITE ON THIS PAGE

*Acknowledgement of copyright*

Question 4 Image is taken from PNGkey. Reproduced under Creative Commons.

Question 8 Editorial Credit: Jose Gil/shutterstock.com