

Computing Science
2021-2022

perfectpapers

Higher PAPER A

1. 8-bit two's complement is used to represented numbers.

(a) Convert the following 8-bit two's complement number into denary.

1000 1010

(b) State the range of numbers that can be represented by 8-bit two's complement.

-118 (1 mark)

-128 (1 mark) to 127 (1 mark)

2. Explain, referring to a programming language of your choice, how floating point data in a program is converted to an integer.

Visual Basic/Python: `int(realValue)` **(1 mark)** returns the integer part of the parameter `realValue`. It does not round up or down. **(1 mark)**

JavaScript/Java:

`Math.round(realValue)` **(1 mark)** returns an integer based on the `realValue` parameter. It will round up or down as needed **(1 mark)**.

**Need both the function and the parameter for 1 mark,
1 mark for explanation of function operation**

3. Describe one problem that can occur when using global variables in a program.

- Conflicting use of variable within a function/procedure for another purpose.
- Difficulty reusing/sharing code when in a multi-programmer development team.
- Other valid.

1 mark for any one bullet, max 1 mark

4. Tracking cookies are a feature of modern web browsing.

Describe two potential security risks related to using tracking cookies.

- Potential use by hackers to hijack browser session.
- Potential use by hackers to compromise stored cookies including passwords (or other security settings).
- Potential to allow profiling of user behaviour by monitoring activity across multiple web sites.
- Potential to compromise private information held in cookies, such as user profile information.

1 mark each bullet, max 2 marks

5. Software for a security entry system stores the codes used to access a building, in an array.

AB23X	HYS67	N8X9S	EJSWS	MNJAG	88HSB
--------------	--------------	--------------	--------------	--------------	--------------

A function is used to validate codes entered and returns a message with the security permission linked to the code.

```
Line 57  FUNCTION securityCheck (ARRAY OF STRING codes, STRING userCode) RETURNS STRING
Line 58          FOR EACH code FROM codes DO
Line 59          IF userCode = code THEN
Line 60          SET message TO "Allow access"
Line 61          ELSE
Line 62          SET message TO "No access"
Line 63          END IF
Line 64          END FOR
Line 65          RETURN message
Line 66  END FUNCTION
```

The program does not work correctly.

(a) Identify the error in the program.

A FOR loop is used in the code so, even after a valid code is found (line 59 is true) **(1 mark)**

the program will change the message with the next code which will not be a match **(1 mark)**

(b) Rewrite part of the code above so the function works as expected.

Either:

FUNCTION securityCheck (ARRAY OF STRING
codes, STRING userCode)

RETURNS STRING

FOR EACH code FROM codes DO

IF userCode = code THEN

SET message TO "Allow access"

RETURN message

ELSE

SET message TO "No access"

END IF

END FOR

RETURN message

END FUNCTION

2 marks for use of RETURN message in correct position within IF statement.

OR

FUNCTION securityCheck (ARRAY OF STRING
codes, STRING userCode)

RETURNS STRING

Counter = 0

REPEAT

Counter = Counter + 1

IF userCode = codes[counter] THEN

SET message TO "Allow access"

ELSE

SET message TO "No access"

END IF

UNTIL message = "Allow Access" OR counter =
<size of codes>

RETURN message

END FUNCTION

1 mark for conditional loop

1 mark for using index/counter with array

6. A computer program deals cards to players. The program makes use of a record structure to hold the details of the cards.

Each card has a suit, a name, a value and a colour. All face cards (Jack, Queen and King) have a value of 10, the Ace has a value of 1 and all other cards, 2 to 10 have their face value e.g. 10 has a value of 10. Hearts and diamonds are red cards, spades and clubs are black cards.



For example, the Jack of Hearts has these values:

Suit: Hearts

Name: Jack

Value: 10

Colour: Red

(a)(i) Using a programming language of your choice, create an efficient record structure to store this information.

```
RECORD card IS {  
    STRING suit,  
    STRING name,  
    INTEGER value,  
    BOOLEAN colour  
}
```

1 mark for use of valid record structure

1 mark for data types/names for suit, name and value

1 mark for use of BOOLEAN for colour (as it has only two states)

Max 3 marks

(ii) There are 52 cards in a deck.
Using a programming language of your choice, define a variable to store all the cards in a deck.

```
DECLARE deck AS ARRAY OF card  
INITIALLY {} * 52
```

1 mark for array of record

1 mark for array size / initialisation

Max 2 marks

(b) The program sets up the deck when the program starts.

Complete the missing steps in this code to setup the deck using your record structure and array from (a)(i) and (a)(ii)

Line 62 SET deck(index).suit TO suits(suit)

(1 mark)

Line 63 SET deck(index).name TO names(name)

(1 mark)

Line 65 SET deck(index).colour TO true

Line 67 SET deck(index).colour TO false

(1 mark for both)

Accept alternatives that use string, don't penalise error in (a)(i) carried forward.

Line 70 SET deck(index).value TO 10 **(1 mark)**

Line 72 SET deck(index).value TO name **(1 mark)**

(c) During testing of the program, the programmer makes use of breakpoints and watchpoints.

Explain the difference between a watchpoint and a breakpoint.

A breakpoint stops execution at a particular position in the code. **(1 mark)**

A watchpoint tracks a variable for when a particular value is set **(1 mark)** and then stops the execution at the position the value was set **(1 mark)**

(d) The card program currently runs on this computer.



Eight Core Processor

32-bit Data Bus

128 Mb Cache Memory

2800 Mhz Clock Speed

(i) Explain how changing the width of the data bus from 32-bit to 64-bit would improve the performance of the computer.

Increasing the width of the databus increases the amount of data that can be transferred between memory and processor **(1 mark)** in each operation **(1 mark)**

(ii) State what is meant by cache memory.

Cache memory is faster access memory within the processor **(1 mark)**

which can be loaded with data from main memory to reduce the need to access main memory **(1 mark)** increasing processor performance.

Also accept responses which compare the relative speeds to access main memory / cache.

(iii) Explain why increasing the size of cache memory may not improve the performance of a program.

A cache only improves processor performance if the data the processors wishes to retrieve is held in the cache **(1 mark)**.

If the cache is frequently “missed” then the processor will frequently need to load data from main memory, reducing the performance of the program **(1 mark)**

7. Code breaker is a simple puzzle game and an app developer wants to develop a program to play the game with a user. The game creates a random code with four letters: A, B, C and D. The code is four characters long and can contain any combination of the four letters.

A user is allowed 10 guesses to match the code. If the user enters a letter that appears in the code but in another position, their feedback is question mark, if the user enters a letter in the position correct, they get a tick in their feedback. If the letter isn't in the code, they get a cross. Some examples of this feedback are shown below.

Code	Guess	Feedback
DDCA	ABCD	?X✓?
DDCA	AACD	??✓?
DDCA	DDCA	✓✓✓✓

(a) During the analysis stage boundaries are identified.

State two boundaries for this task.

- Only letters A,B,C and D are used to create the code
- Users have 10 guesses to get the correct code
- Feedback is ? for appears in code, X if does not appear and ü if in the correct position.
- A random code is generated from the available four characters.

1 mark for each bullet, max 2 marks

(b) The characters in the program can be represented as either ASCII or Unicode.

Describe an advantage of using Unicode over ASCII, referring to the number of bits used to represent a character in each format.

Unicode can represent a greater range of characters **(1 mark)**

Unicode uses 16 bits for representation, whereas extended ASCII uses 8-bit (ASCII uses 7-bit). **(1 mark)**

(c) The app developer updates the app so that the scores of users are uploaded to a server.

Describe how digital certificates are used as part of secure encrypted transfer of data.

Data is encrypted using a public key when sent **(1 mark)**

and is decrypted using the private key when it is received **(1 mark)**

(d) An algorithm is used to find the user with the highest score for the app.

```
Line 72      SET counter TO 1
Line 73      SET maximum TO scores[counter]
Line 74      SET counter TO counter + 1
Line 75      WHILE counter <= LENGTH(scores)
Line 76          IF testscore[counter] > maximum THEN
Line 77              SET maximum TO testscore[counter]
Line 78              SET counter TO counter + 1
Line 79          END IF
Line 80      END WHILE
```

(i) Complete this trace table for the first four values in the array scores. The first one, has been done for you.

scores [2716, 3737, 8009, 2028, 3340]

Line	counter	scores[counter]	maximum
73	1	2716	2716
79	3	3737	3737
79	4	8009	8009
79	4	8009	8009

1 mark for rows 2 and 3 complete,
1 mark for last row correct.

(ii) Explain the error in this algorithm and how it can be resolved.

The counter to loop through the array is only incremented when a new maximum is set.

1 mark

Swap lines 78 and 79 to move the counter increment so that it happens outside of the IF statement **(1 mark)**

8. Forest Green make six different flavours of ice cream and use the following program to count the orders for the day. The orders are stored in an array called orders.

orders	Chocolate	Mint	IronBrew	...	Vanilla	Mint
--------	-----------	------	----------	-----	---------	------

```
Line 37      ...
Line 38      SET MintOrders TO CountOrders(orders,"Mint")
Line 39      SET ChocOrders TO CountOrders(orders,"Chocolate")
Line 40      SET IronBewOrders TO CountOrders(orders,"IronBrew")
Line 41      ...

Line 88      FUNCTION CountOrder(ARRAY OF STRING orders, STRING icecream)
              RETURNS INTEGER
Line 89          SET count TO 0
Line 90          FOR EACH order FROM orders DO
Line 91              IF order = icecream THEN
Line 92                  count = count + 1
Line 93              END IF
Line 94          END FOR
Line 95          RETURN count
Line 96      END FUNCTION
```

(a) Describe how the parameters on line 38 are used. Your answer should identify the formal and actual parameters.

(b) Explain why the code for this program is inefficient.

The array orders and the string “Mint” are passed into the function. Orders is looped through to allow the values equal to the string “Mint” to be counted **(1 mark)**

The formal parameters are orders and icecream at Line 88 **(1 mark)**

The actual parameters are orders and “Mint” at Line 38. **(1 mark)**

Each time a count is made, the function has to loop through all the values in orders **(1 mark)**

There are individual counters for each icecream, which refer to specific flavours, if the flavours change, then the program will need to be amended **(1 mark)**

(c) A programmer has plans to use two parallel 1D-Arrays to store the six ice cream flavours being counted and the count for each one.

orderKey	Chocolate	Mint	IronBrew	...
orderCount	7	3	10	...

The programmer designs this algorithm for the initial design of this part of the program.

- 3.1 add each unique icrecream flavour in orders to the orderKey array
- 3.2 initialise orderCount for every element in orderKey
- 3.3 count each order, updating the correct element in orderCount
- 3.4 display the results

(i) Using a design technique of your choice, complete a design for step 3.1 above. Part of the design has been completed for you.

```
SET orderKeyIndex TO 1 #index for orderKey array  
SET orderIndex To 1  #index for order array
```

```
WHILE orderKeyIndex <= 6 AND orderIndex <= Length(orders) DO
```

```
    SET checkIndex TO 0  #index to use with orderKey  
    SET keyFound TO False
```

```
    WHILE NOT keyFound AND checkIndex < orderKeyIndex DO  
        IF orderKey[checkIndex] = orders[orderIndex] THEN  
            SET keyFound TO True  
        ELSE  
            SET checkIndex TO checkIndex + 1  
        END IF  
    END WHILE
```

```
IF keyFound = False THEN  
    SET orderKey[orderKeyIndex] TO orders[orderIndex]  
    SET orderKeyIndex TO orderKeyIndex + 1  
END IF  
SET orderIndex TO orderIndex + 1  
END WHILE
```

1 mark for conditional loop
1 mark for complex condition in loop
1 mark for IF statement to detect key matching order value
1 mark for setting keyFound to true
1 mark for incrementing checkIndex

(ii) Using a design technique of your choice, complete a design for step 3.3.

FOR EACH order FROM orders DO

SET orderKeyIndex TO 0

WHILE NOT(orderKey[orderKeyIndex] = order) DO

orderKeyIndex = orderKeyIndex + 1

END WHILE

orderCount[orderKeyIndex] = orderCount[orderKeyIndex] + 1

END FOR

1 mark for FOR loop for each order

1 mark for use of orderKeyIndex, initialise and increment

1 mark for detecting when the orderKey is found

1 mark for incrementing the count for each order

9. A database consists of the following three tables.

Session		
SessionID	SessionTime	SessionDay
...	...	...
2981	10:00 – 10:50	Thursday
2999	11:00 – 11:50	Thursday
3002	12:30 – 13:20	Thursday
3003	16:00 – 17:00	Friday
3005	09:00 – 11:00	Friday
...	...	...

Space		
SpaceID	SpaceName	Equipment
...	...	...
002	Weights Room	Kettlebells
009	Weights Room	Sandbells
012	Spin Studio 1	Cycles
078	Spin Studio 2	Cycles
102	Gym 1	Treadmill
107	Gym 2	Elliptical
109	HIIT Space	Barbells
...	...	...

Booking		
Client	SessionID	SpaceID
...	...	...
Miss E Smith	2981	009
Mr R Evans	2981	012
Mr M Paul	2981	107
Miss E Smith	3005	002
Mr R Evans	3005	107
...	...	...

(a) State the compound primary key from the database above.

(b) Draw an entity relationship diagram for this database. Your answer should show the entity names and cardinality. You do not need to show attributes in your diagram.

SessionID, SpaceID

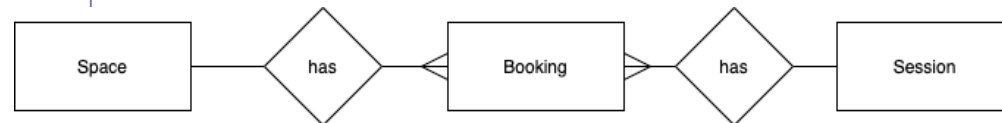
1 mark for each part of compound key

Max 2 marks

1 mark for all entities and named relationships

1 mark for correct relationship types

Max 2 marks



10. A database table is shown below.

Commodity				
product	area	buy	sell	profit
Mineral Extractors	Sherones	429	798	369
Domestic Appliances	Sherones	437	701	264
Liquor	Guguroro	587	841	254
Water Purifiers	Guguroro	182	401	219
Clothing	Guguroro	229	445	216
Basic Medicines	Setesuyara	195	316	121
Food Cartridges	Setesuyara	45	142	97
Power Generators	Setesuyara	437	529	92
Hydrogen Fuel	Setesuyara	94	107	13
Biowaste	Setesuyara	14	15	1

Complete the table below showing the expected output from this SQL statement.

```
SELECT area, MAX(profit) AS "Trade Income"  
FROM Commodity  
WHERE buy < 430  
GROUP BY area;
```

area	Trade Income
Sherones	219
Guguroro	121
Setesuyara	369

1 mark for area

1 mark for trade income

Max 2 marks

11. An orchestra uses a database to hold records of musicians, instruments, and performances at concerts.

The musicians can be selected by the instrument or by their seat in the orchestra. A manager can count the number of musicians that are available to play instruments.

(a) State two functional requirements of the relational database.

- Select musicians by instrument or by seat
- Count musicians
- Store musicians, instruments and performances

1 mark each bullet, max 2 marks

(b) Tables from the database are shown below.

performno	musicianno	concertno	instrument	perf type
9	3	1	cello	classical
10	3	4	banjo	jazz
11	4	1	violin	classical
12	4	4	bass	jazz
13	5	4	guitar	jazz
14	5	4	violin	jazz
15	6	1	violin	classical
16	6	1	drums	classical
17	7	5	trumpet	jazz
18	7	5	clarinet	jazz
19	6	2	viola	classical
20	4	1	guitar	classical
...	...	...	...	...

concertno	concert venue	con date
1	Bridgewater Hall	06/01/2021
2	Bridgewater Hall	08/05/2021
3	Usher Hall	03/06/2021
4	Assembly Rooms	20/09/2021
5	Festspiel Haus	21/02/2021
...	...	...

musicianno	name	born	seat
3	Helen Smyth	07/08/1978	1st
4	Harriet Smithson	08/03/1975	1st
5	James First	11/11/1981	2nd
6	Theo Mengel	13/03/1988	2nd
7	Sue Little	31/10/1974	3rd
...	...	...	...

(i) Design a query to display all the musician details and the number of concerts that each musician, has or will play, in a field called "Total Concerts". Musicians sometimes play more than one instrument at a concert.

Field(s) and calculation(s)	
Table(s)	musician, performance
Search criteria	
Grouping	
Sort order	

- 1 mark for musician fields (ideally with wildcard)
- 1 mark for COUNT of concertno
- 1 mark for alias AS "Total Concerts"
- 1 mark for group by musician

Field(s) and calculation(s)	musician.*, COUNT(concertno) AS "Total Concerts"
Table(s)	musician, performance
Search criteria	
Grouping	musician
Sort order	

(ii) Each musician is paid a fee of £100 per instrument, for each concert. Design a query to calculate the fees due for each concert for jazz performances by 1st seat musicians.

Field(s) and calculation(s)	
Table(s)	musician, performance, concert
Search criteria	
Grouping	
Sort order	

1 mark for calculation of fees (* 100)

1 mark for musician.seat = "1st"

1 mark for use of AND and second condition ("Jazz" performance).

1 mark for group by concertno

Field(s) and calculation(s)	concert.*, count(musician.musician no) * 100 AS "Concert Fees"
Table(s)	musician, performance, concert
Search criteria	musician.seat = "1 st " AND performance.perf_type = "Jazz"
Grouping	concert.concertno
Sort order	

(c) The perf_type attribute of performance appears in the performance table but it is actually concerts that are “Jazz”, “Classical” and other values. Evaluate the design of the database in terms of fitness for purpose and describe a solution to improve the database design.

The design of the current solution is not fit for purpose as errors could be introduced because.

- Making changes to the perf_type value requires that multiple rows are amended.
- Deleting all the performances, removes details of the concert type.

1 mark for each bullet, answer needs to refer to issues which would result in a solution that was not fit for purpose.

Improvement:

Move the perf_type attribute to concert entity so that it is dependent on concertno. **(1 mark)**

12. A classic vehicle company supply vehicles to customers across the world. Sample data relating to customer orders is shown below.

customer						
CID	Company	CLastname	CFirstname	Address	City	Country
I03	Atelier graphique	Schmitt	Carine	54, rue Royale	Nantes	France
I12	Signal Gift Stores	King	Jean	8489 Strong St.	Las Vegas	USA
I14	Australian Collectors, Co.	Ferguson	Peter	636 St Kilda Road	Melbourne	Australia
I19	La Rochelle Gifts	Labruno	Janine	67, rue des Cinquante Otages	Nantes	France
I21	Baane Mini Imports	Bergulfsen	Jonas	Erling Skakkes gate 78	Stavern	Norway
I24	Mini Gifts Distributors Ltd.	Nelson	Susan	5677 Strong St.	San Rafael	USA
I25	Havel & Zbyszek Co	Piestrzeniewicz	Zbyszek	ul. Filtrowa 68	Warszawa	Poland
I28	Blauer See Auto, Co.	Keitel	Roland	Lyonerstr. 34	Frankfurt	Germany

prodorder					
orderNo	orderDate	productCode	shippedDate	status	CID
I0103	2020-01-29	3891		On Order	I21
I0104	2020-01-29	1589		On Order	I21
I0278	2021-08-06	3891		On Order	I12
I0309	2021-10-15	1589		In Process	I21
I0315	2021-10-29	3990		In Process	I19
I0325	2021-11-05	1749	2021-11-08	Shipped	I21
I0342	2021-11-24	4473	2021-11-29	Shipped	I14
I0345	2021-11-25	1889	2021-11-26	Shipped	I03
I0346	2021-11-29	1662	2021-11-30	Shipped	I12
I0425	2022-05-31	3891		In Process	I19

product			
productCode	item	inStock	cost
3891	1969 Ford Falcon	3	83050
3990	1970 Plymouth Hemi Cuda	12	31920
4473	1957 Chevy Pickup	13	55700
4675	1969 Dodge Charger	15	58730
1097	1940 Ford Pickup Truck	6	58330
1129	1993 Mazda RX-7	8	83510
1342	1937 Lincoln Berline	18	60620
1367	1936 Mercedes-Benz 500K Special Roadster	18	24260
1589	1965 Aston Martin DB5	19	65960
1662	1980s Black Hawk Helicopter	11	77270
1749	1917 Grand Touring Sedan	6	86700
1889	1948 Porsche 356-A Roadster	18	53900

(a) A total of the cost for all orders which have not been shipped is required.

Pending Orders
412990

Using the same data provided, write the SQL statement that would produce this output.

```
SELECT SUM(cost) AS "Pending Orders"
FROM prodorder, product
WHERE prodorder.productCode =
product.productCode AND
Status<>"Shipped"
```

or

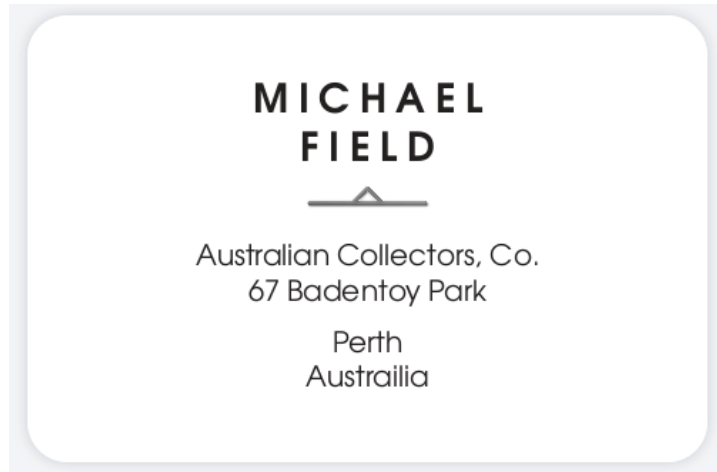
```
SELECT SUM(cost) AS "Pending Orders"
FROM prodorder, product
WHERE prodorder.productCode =
product.productCode AND
NOT(Status="Shipped")
```

1 mark for the use of SUM with Alias

1 mark for JOIN

1 mark for only those not marked as "Shipped"

(b) The details for Australian Collectors, Co. have changed. The new information is shown on this business card.



Write the SQL to amend the details for this customer.

UPDATE customer

SET CLastname="Field", CFirstname = "Michael", Address="67 Badentoy Park", City="Perth"

WHERE CID=114

1 mark for UPDATE customer

1 mark for SET with correct fields/values updated

1 mark for selecting correct customer row

13. HTML 5 elements have been used to define parts of the web page shown below.

Home News Contact About

DEVHELP
Machine intelligence for today

Your application to join our team

Thank you for applying to join our team. Your application details are shown below.

About You

Firstname: Charlotte Lastname: Green
Position Applied For: Lead Programmer

Your Contact Details

Mobile: 07700 900796 Email: charlotte@devhelp.com
Address: 55 York Road EAST LONDON E21 0LD

Qualifications

Bachelor of Science, Honours, First Class - Computing Science
University of Edinburgh

State which elements have been used for the parts labelled A and B.

A - main

B - section

1 mark each, max 2 marks

14. HTML for a menu is shown below.

```
<ul>
  <li><a href="home.html">Index</a></li>
  <li><a href="css.html">Learn CSS</a></li>
  <li><a href="javascript.html">Learn
JavaScript</a></li>
</ul>
```

Two options are available for the styling of the menu.

Option 1

[Index](#)
[Learn CSS](#)
[Learn JavaScript](#)

Option 2

[Index](#) [Learn CSS](#) [Learn JavaScript](#)

Complete the CSS rule for Option 1 and for Option 2

Option 1

display: block; **(1 mark)**

Option 2

Display: inline; **(1 mark)**

15. A web development company makes use of the following document.

Alexi Johnson



Title: **Marketing and Performance Manager**
Decision Maker: **Yes**
Industry: **Software and Technology**
Age: **30**
Skills: **Team Management, Project Planning, Recruitment**

Goals: **Build a great team, improve quality of development projects**
Challenges: **Need to increase number of clients and recommendations**
How we help: **Give Alexi tools to monitor projects and share success with prospective clients**

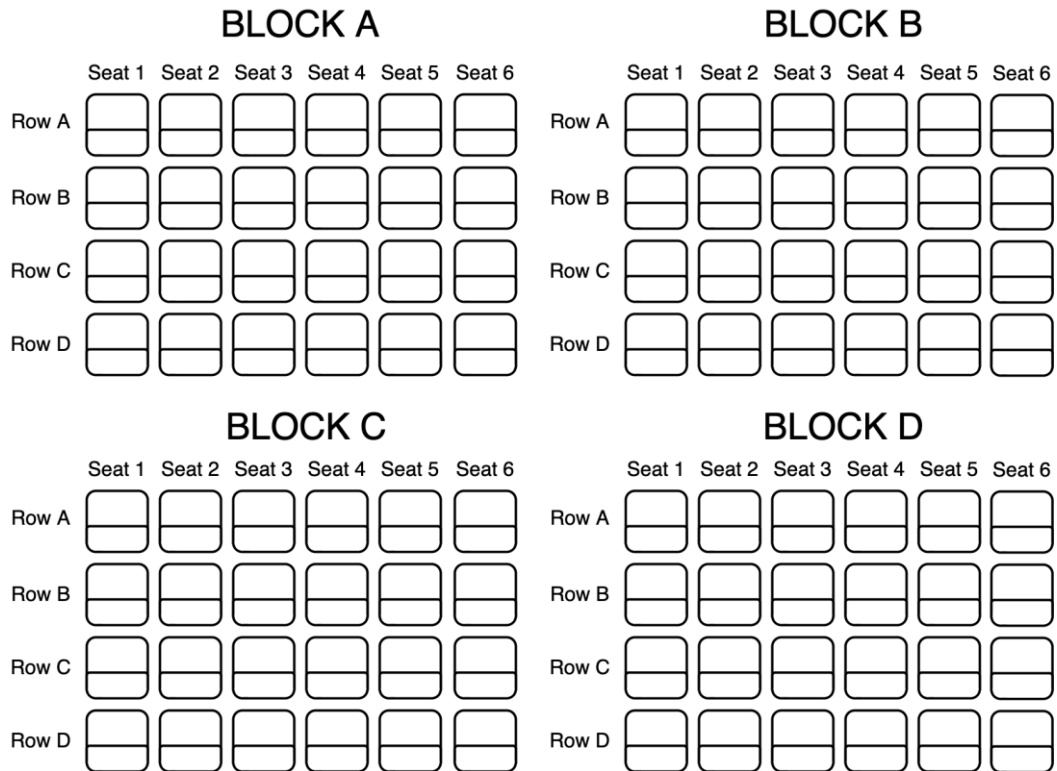
- (a) State the name given to this document.
- (b) Describe how this document is used in the development process.

Persona **(1 mark)**

The benefits of personas are that:
they enable developers to consider the end-user's needs and wants,
remind developers that their own views are not necessarily the end-users' views, and
provide an effective communication tool, which facilitates better design decisions.

Any 1 bullet for 1 mark

16. A concert ticket company wish to develop a new web site. They are selling tickets for a VIP area at a concert. The concert seats are available as shown.



Seats are only sold in pairs e.g. A1 with A2, D3 with D4 or B5 with B6. Sales that would create single seat gaps are not allowed e.g. B2 with B3 would not be allowed as it would make B1 a single seat.

When selecting tickets, customers also have to enter the following details:

- Name
- Email Address
- Mobile Number

(a) Using this information, draw a wireframe design for the form on the concert ticket web page.

Annotated wireframe showing all required form elements.

2 marks for name, email address, mobile number,

1 error 1 mark

2 marks for Block, Row, Seat,

1 error, 1 mark

Max 4 marks

Name

Email Address

Mobile Number

Block

Select Block

Row

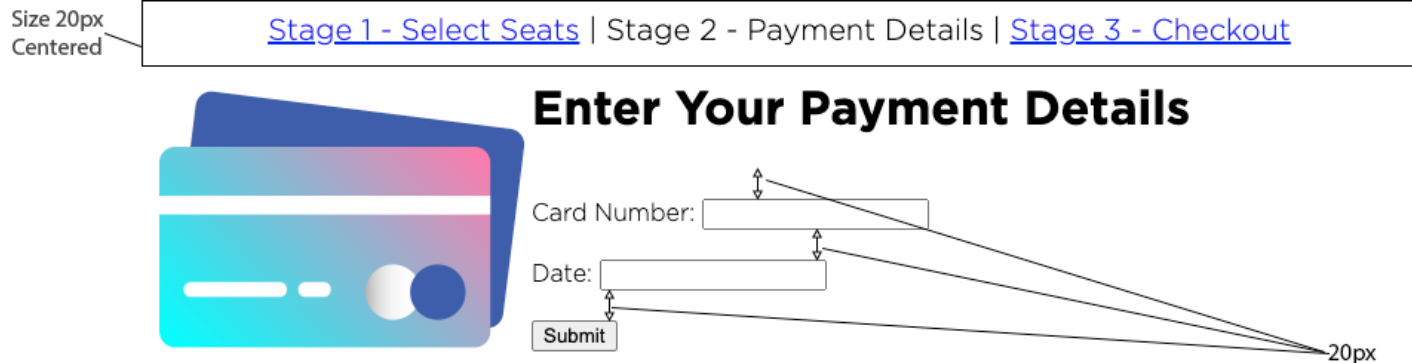
Select Row

Seat

Select Seats

Book Seats

(b) Once a customer has selected their seats, the next stage of the process is to add payment details. An annotated wireframe design for this page and form is shown below.



```
<body>
```

```
<header>
```

```
<span><a href="stage1.php">Stage 1 - Select Seats</a> | </span>
```

```
<span>Stage 2 - Payment Details | </span>
```

```
<span><a href="stage3.php">Stage 3 - Checkout</a></span>
```

```
</header>
```

```
<form action="stage3.php" method="post">
```

```
 </pre>
```

```
<h1>Enter Your Payment Details</h1>
```

```
Card Number:
```

```
<input type="text" name="card" id="cardnumber"><br/>
```

```
Date:
```

```
<input type="text" name="expiry" id="expiry"><br />
```

```
<input type="submit" name="submit" value="Submit">
```

```
</form>
```

```
</body>
```

Write CSS rules so the output will match the wireframe above.

```
body {  
    font-family: Gotham;  
}  
header {  
    font-size: 20px;  
    text-align: center;  
}  
#cardimage {  
    _____  
    _____  
}  
  
form input {  
    _____  
}
```

```
#cardimage {  
    float: left; (1 mark)  
    clear: both; (1 mark)  
}  
  
form input {  
    margin-top: 20px; (1 mark)  
}
```


(c) Explain why the descendant selector form input is used to format the page in (b).

Descendant selector matches all elements that are descendants of the specified element **(1 mark)**

In the code for (b) this applies the rule to all the INPUTs within the FORM. **(1 mark)**

17. An esports web site has a page where team managers can upload their team details to take part in competitions. The page makes use of the following CSS.

```
#sidenav {  
    float: left;  
    background: darkblue;  
    color: whitesmoke;  
    width: 25%;  
    padding: 5%;  
}  
#sidenav a hover {  
    color: white;  
    font-weight: bolder;  
}  
#sidenav a {  
    color: white;  
}  
#detail {  
    float: left;  
    height: 200px;  
    background: lightgray;  
    color: black;  
    width: 55%;  
    padding: 5%;  
}
```

(a) This CSS code is not efficient.
Rewrite this code making use of
grouping and descendant selectors.

```
#sidenav, #detail {  
    float: left;  
    padding: 5%;  
}  
  
#detail {  
    background: lightgray;  
    color: black;  
    width: 55%;  
}  
  
#sidenav {  
    background: darkblue;  
    color: whitesmoke;  
    width: 25%;  
}  
  
#sidenav a {  
    color: white;  
}  
  
#sidenav a hover {  
    font-weight: bolder;  
}
```

Grouping of #sidenav, detail with
Float, padding properties in grouping
(1 mark)

Separate rules for #detail and
#sidenav with unique properties **(1
mark)**

Rule for #sidenav a, (such that hover
inherits color property **(1 mark)**

Separate rule for #sidenav a hover **(1
mark)** which removed color property.

(b) An input field is required for the number of team members. The field is required and can be a minimum of 4 and a maximum of 8.

Write the HTML to ensure that only valid values will be accepted in the form.

(c) Compatibility testing should also be carried out on the website.

State one problem that may be identified when testing compatibility.

```
<input type="number" min="4" max="8" />
```

Number type for 1 mark

Range for (min and max) for 1 mark

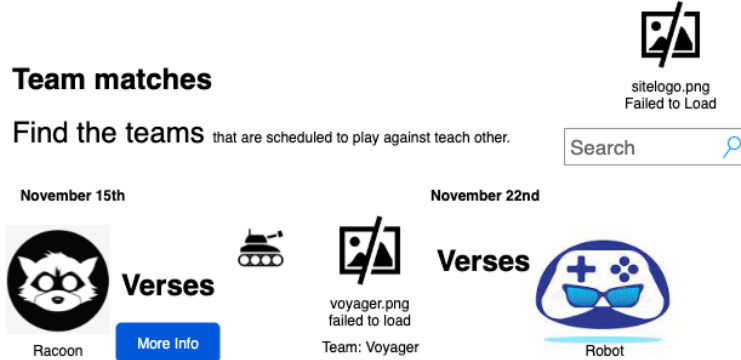
Issues with page display/loading on:

Devices (mobile, tablet, desktop)

Range for browsers

1 mark for any bullet, max 1 mark

(d) Once all the teams have been submitted, team managers have access to a matches page as shown below.



Evaluate this solution in terms of usability.

Issues include:

- No search button
- Images not loaded
- Unclear if buttons are missing or how graphics work.
- Some text missing
- Page displays error messages

1 mark for each bullet, max 2 marks

(e) When a team icon is clicked, a dialogue appears on the page showing more information about a team.

Complete the following code to display this dialogue

```
<div id="racoon" _____="displayInfo('Racoon');">  
    Racoon  
</div>
```

onClick **(1 mark)**