File Organization And Access

COMPUTER SCIENCE 9618 PAPER 3

File Organization And Access

Refers to the way data is stored and accessed in a file

We have three different types of File Organization

Serial Sequential Random

Serial File Organization

Describe Serial File Organization

- Data is stored in chronological order (Arranged in order of time)
- Data is added at the end of the life which also makes easier to add / append the data in a file
- Allows the data to be read in order they are taken
- No key fields need to be used.

When will we use Serial File Organization

- Chronological order does matter
- Easier to append at the end of the file
- Reorganization and resorting is not required
- Small file so easy to search

Exam Style Question

(b) (i) The weather station records how the outside temperature changes over a period of time. The system will read the temperature once every hour, over a period of 100 days.

The temperature readings are automatically stored in a file. No other data are stored.

Explain why the weather station has decided to use serial organisation for the file.

.....

.....

.....

..... [2]

5(b)(i)	1 mark per bullet point to **max 2** • So the readings are stored in **chronological** order • Easy to add / append each new reading to the end of the file // no further processing is required • Allows the readings to be read in the order that they were taken • Readings do not need to be given further identification as to date / time // no key field needs to be added	**2**

Sequential File Organization

Sequential file organization stores records in a specific, ordered sequence, typically based on a key field like a customer ID or a date or alphabetical order

Sequential file organization is like a library shelf where books are arranged in alphabetical order by the author's last name. If you're looking for a book by a specific author, you can go directly to the section of the shelf where books by that author would be, making it quicker to find what you need compared to a random stack of books.

Describe Sequential File Organization

- Data is stored one after another and need to be accessed one after another
- Sequential files are stored with ordered records and stored in the order of the key field
- In sequential files new records are inserted in the correct position

Where will we use sequential file organization ?

- **Unique field (So can be used as index)**
- **Sorted in a pre decided order**

Difference between serial and sequential file organization

Serial file organization is like a stack of receipts you throw into a drawer as you receive them. There's no particular order, so if you want to find a specific receipt, you have to look through each one until you find it.

Sequential file organization is like a library shelf where books are arranged in alphabetical order

Note : The above can be used as example while writing your answer but you have to add description of each aswell.

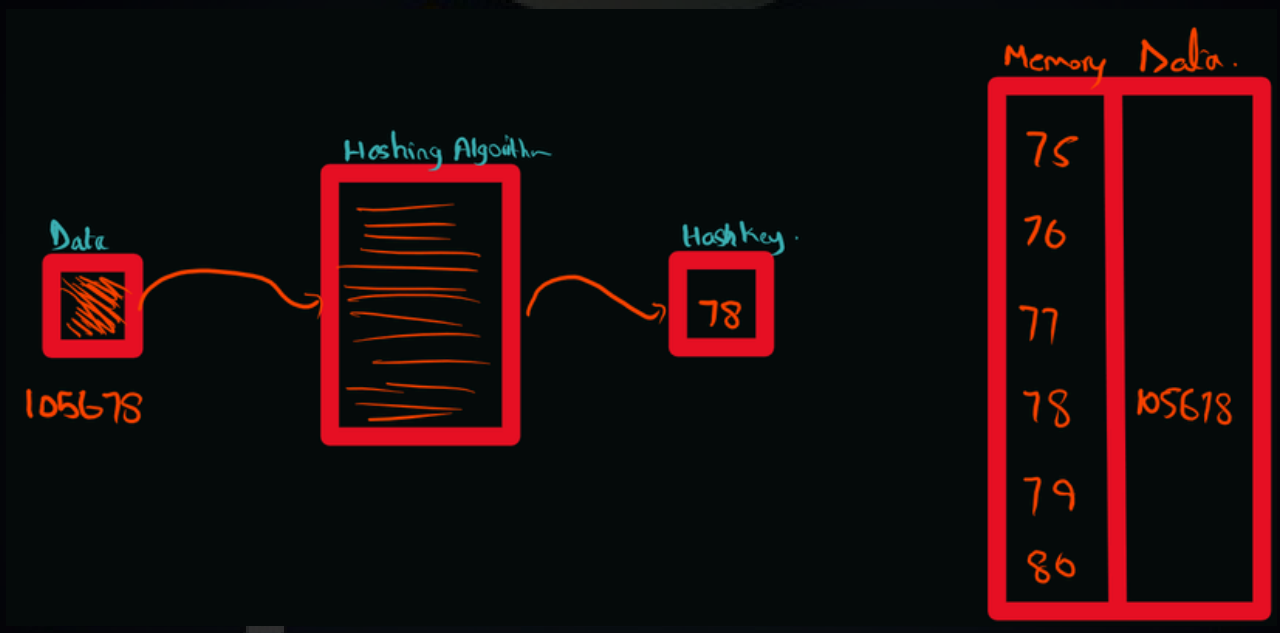
5(a)	**One** mark for each correct marking point **(Max 4)** • In both serial and sequential files records are stored one after the other … • … and need to be accessed one after the other • Serial files are stored in chronological order • Sequential files are stored with ordered records • … and stored in the order of the key field • In serial files, new records are added in the next available space / records are appended to the file • In sequential files, new records are inserted in the correct position.	**4**

Note : Above is the right answer but make sure to add the examples as well while writing.

Random File Organization

Random (or direct) file organization stores records in any order but provides a direct access method to retrieve them. Each record is associated with a unique key or identifier, and the system uses a Hashing Algorithm or some direct access method to determine the exact location of the record on the storage device.

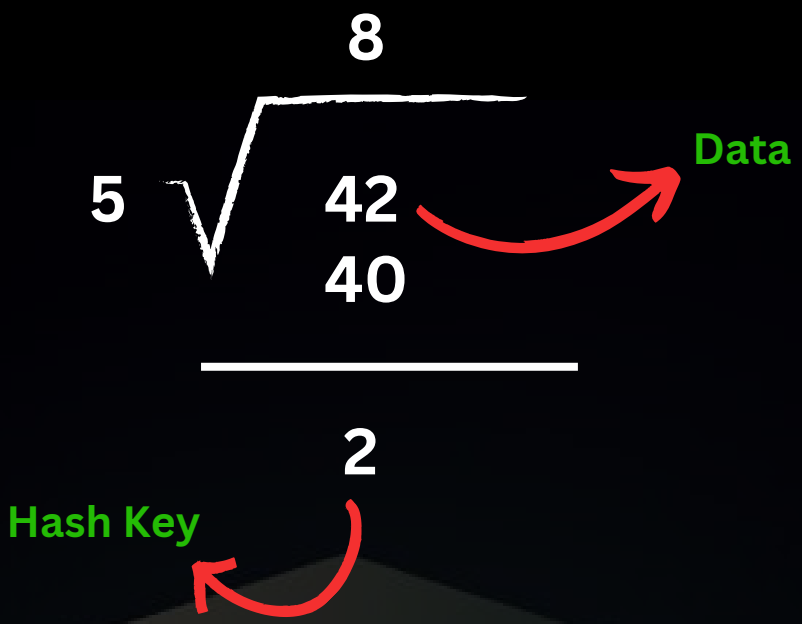
Hashing Algorithm : Is basically a mathematical function or steps to find a unique key knows as hash to access data.



We have various hashing algorithms and some of them are Folding, Truncation and Modular Division

Modular Division

Hash Key = Data MOD table size



Exam Style Question

(b) A hashing algorithm is used to calculate storage locations for records in a random access file. The algorithm calculates hash values using the function modulus 5.

The function modulus gives the remainder after integer division.
For example, 3003 modulus 5 = 3, so the record key 3003 gives a hash value of 3.

Complete the table to show the remaining hash values.

Record key	Hash value
3003	3
1029	
7630	

[2]

2(b)	**One** mark for each correct hash value (**Max 2**) **Record key** / **Hash value** 3003 / 3 1029 / 4 7630 / 0	**2**

4 A bank has 95 000 customers. Each customer has a unique ID.

When a customer uses an Automated Teller Machine (ATM) to obtain cash, their current balance is checked. The balance is stored in a file which has the following fields:

- the customer ID (6-digit number in the range 100000 to 999999)
- an encrypted PIN
- the current balance

The file can store a maximum of 100 000 records.

(a) Give a reason why a random organisation would be appropriate for this file.

.....

.....[1]

(b) An algorithm for inserting a new record in this file uses the following hash function:

```
RecordKey ← CustomerID MOD 100000
```

where `RecordKey` is the record position in the file.

(i) Complete the table to show the values generated by the hash function for the given customer IDs.

CustomerID	RecordKey
802139	2139
700004	
689998	
102139	

[1]

(ii) State the range of possible values for `RecordKey`.

Minimum value of `RecordKey`: .....

Maximum value of `RecordKey`: .....

[2]

Memory

Index	Data
2136	
2137	
2138	

Memory consist of two things location index and data on that location. Location is what we get by the hashing algorithm and we place the data on that location.

Collision In Hashing

Index	Data
2136	
2137	
2138	

What is meant by collision ?

- A collision is when the two values in the key filed for two records pass through a hashing algorithm and result in the same hash value
- So the location identified by the hashing algorithm may already be in use and the two records cannot occupy the same address

Methods To Deal With Collision

1) Linear Probing

Imagine you are placing books on a bookshelf. Each book has a designated spot. However, if that spot is already occupied by another book, you simply move to the next available spot on the shelf and place your book there.

Index	Data
2136	
2137	
2138	

- A process of collision resolution is used
- start at the original hashed storage space
- go through the following spaces in a linear fashion
- and store the data item in the first available slot

Process Of Finding The Record If Linear Probing Is Used

- If collision is identified search linearly from where you are (closed hash) until the matching record key is found
- if not found record is not in the file

2) Overflow Area

Consider the bookshelf again, but this time, if the spot is taken, you place the book in a separate, overflow section of the bookshelf. You then continue checking the overflow section in a linear fashion until you find an empty spot.

Index	Data
2136	
2137	
2138	

Index	Data
3000	
3001	
3002	

- Search the overflow area
- go through the following spaces in a linear fashion
- and store the data item in the first available slot

Process Of Finding The Record If Overflow Area Is Used

- If collision is identified then search the overflow area linearly (open hash) until the matching record key is found
- if not found record is not in the file

3) Chaining

If a spot on the shelf is occupied, instead of moving the book, you attach it to the existing book with a clip. This creates a chain of books at that spot

Index	Data
2136	
2137	
2138	

- Each storage space holds a reference to a chain of items
- which can be searched individually
- The data item is stored in the first available space in this chain

Process Of Finding The Record If Chaining Is Used

- If collision is identified then search the data by linearly searching in the chain.
- if not found record is not in the file

Closed Hash And Open Hash

In closed hashing, all elements are stored within the hash table itself. (Linear Probing)

In open hashing, collisions are handled by storing the elements outside the hash table (Overflow and Chaining)

Describe Random File Organization

- Record location is calculated
- Using a hashing algorithm on a key field
- Records are stored in no particular order within the file
- If a record can not be stored due to collision
- Then subsequent location is searched known as closed hash
- or an overflow area is searched known as open hash
- updates to the file can be carried out directly

Sequential Access And Direct Access

Sequential Access : Imagine you're looking for a specific book in a library, but the books are arranged in a long row without any clear order, like a line of books stacked one after the other. To find the book you need, you would have to start at the beginning and check each book in sequence until you find the one you're looking for. This is how sequential access works

Direct Access : Imagine having a personal locker at a gym. Each locker has a unique number, and you have a key that corresponds exactly to your locker. When you want to retrieve something from your locker, you don't need to search through all the other lockers one by one. Instead, you can go directly to your specific locker

Outline the process of sequential access

- Starts at the beginning of the file and earliest reading is accessed first
- checks records linearly and each successive reading is read
- until the desired record is found or end of the file is reached.

Outline the process of Random Access

- Generate a hash key using a hashing algorithm
- Calculate the exact record location in the random access file using the hash value.
- Directly access the record determined by the hash value.
- Retrieve the desired record directly without scanning through other records.

Draw lines to link each file organisation method to its appropriate file access method(s).

File organisation method	File access method
random	sequential
serial	direct
sequential	

[4]

File Organization And Access

Question 1

2 The diagram shows four files and three methods of file organisation.

Draw **one** line to match each file with its most appropriate method of file organisation.

File	File organisation
Text file	Sequential
File for recording the temperature every hour	Random
Master file for paying each employee every month	Serial
Customer user name and password file	

[4]

Question 2

(b) (i) The weather station records how the outside temperature changes over a period of time. The system will read the temperature once every hour, over a period of 100 days.

The temperature readings are automatically stored in a file. No other data are stored.

Explain why the weather station has decided to use serial organisation for the file.

.....

.....

.....

..... [2]

(ii) Serial files can be accessed using sequential access.

Explain how sequential access could be used for the temperature readings file.

.....

.....

.....

..... [2]

(iii) Name **and** describe a method of file organisation other than serial or sequential.

Method .....

Description .....

.....

.....

.....

.....

.....

[4]

Question 3

4 A bank has 95 000 customers. Each customer has a unique ID.

When a customer uses an Automated Teller Machine (ATM) to obtain cash, their current balance is checked. The balance is stored in a file which has the following fields:

- the customer ID (6-digit number in the range 100000 to 999999)
- an encrypted PIN
- the current balance

The file can store a maximum of 100 000 records.

(a) Give a reason why a random organisation would be appropriate for this file.

.....

.....[1]

(b) An algorithm for inserting a new record in this file uses the following hash function:

```
RecordKey ← CustomerID MOD 100000
```

where `RecordKey` is the record position in the file.

(i) Complete the table to show the values generated by the hash function for the given customer IDs.

CustomerID	RecordKey
802139	2139
700004	
689998	
102139	

[1]

(ii) State the range of possible values for `RecordKey`.

Minimum value of `RecordKey`: .....

Maximum value of `RecordKey`: .....

[2]

(iii) A procedure is written to insert a new record into the file.

Complete the algorithm for this procedure.

```
PROCEDURE InsertRecord(CustomerID : INTEGER)
   RecordKey ← CustomerID MOD 100000
   Success ← FALSE
   // Find position for new record and insert it
   REPEAT
      IF record at position RecordKey is .....
         THEN
            Insert new record at position RecordKey
            Success ← TRUE
         ELSE
            IF RecordKey = .....
               THEN
                  RecordKey ← .....
               ELSE
                  RecordKey ← ..... + 1
            ENDIF
      ENDIF
   UNTIL Success = TRUE
ENDPROCEDURE
```

[4]

(c) **(i)** Explain why an encrypted version of the PIN is stored in the file.

.....

.....

.....

.....[2]

(ii) A customer attempts to withdraw cash from an ATM. An algorithm is used to check if the customer has entered the correct PIN.

Complete the algorithm.

1. Customer ID is read from card.
2. Customer enters PIN.
3. Customer PIN is .....
4. .....
5. Customer record is located in file.
6. .....
7. If match then transaction can proceed.

[3]

Question 4

4 (a) Three file organisation methods and two file access methods are shown below.

Draw lines to link each file organisation method to its appropriate file access method(s).

File organisation method	File access method
random	sequential
serial	direct
sequential	

[4]

(b) An energy company supplies electricity to a large number of customers. Each customer has a meter that records the amount of electricity used. Customers submit meter readings using their online account.

The company's computer system stores data about its customers.

This data includes:

- account number
- personal data (name, address, telephone number)
- meter readings
- username and encrypted password.

The computer system uses three files:

File	Content	Use
A	Account number and meter readings for the current month.	Each time a customer submits their reading, a new record is added to the file.
B	Customer's personal data.	At the end of the month to create a statement that shows the electricity supplied and the total cost.
C	Usernames and encrypted passwords.	When customers log in to their accounts to submit meter readings.

For each of the files A, B and C, state an appropriate file organisation method for the use given in the table.

All three file organisation methods must be different.

Justify your choice.

(i) File A organisation .....

Justification .....

.....

.....

.....[3]

(ii) File B organisation .....

Justification .....

.....

.....

.....[3]

(iii) File C organisation .....

Justification .....

.....

.....

.....[3]

Question 5

4 **(a)** Three file organisation methods and two file access methods are shown below.

Draw lines to link each file organisation method to its appropriate file access method or methods.

File organisation method	File access method
serial	direct
sequential	sequential
random	

[4]

(b) A bank has a very large number of customers. The bank stores data for each customer. This includes:

- unique customer number
- personal data (name, address, telephone number)
- transactions

The bank computer system makes use of three files:

- A – a file that stores customer personal data. This file is used at the end of each month for the production of the monthly statement.
- B – a file that stores encrypted personal identification numbers (PINs) for customer bank cards. This file is accessed when the customer attempts to withdraw cash at a cash machine (ATM).
- C – a file that stores all customer transaction records for the current month. Every time the customer makes a transaction, a new record is created.

For each of the files A, B and C, state an appropriate method of organisation. Justify your choice.

(i) File A organisation .....

Justification .....

.....

.....

.....[3]

(ii) File B organisation .....

Justification .....

.....

.....

.....[3]

(iii) File C organisation .....

Justification .....

.....

.....

.....[3]

Question 6

(ii) The programmer decides to store all the data in a file. Initially, data from 27 locations will be stored. More rainfall locations will be added over time and will never exceed 100.

The programmer has to choose between two types of file organisation. The two types are serial and sequential.

Give **two** reasons for choosing serial file organisation.

.....

.....

.....

.....[2]

Question 7

(ii) The programmer decides to store all the data in a file. The number of weather stations could grow to reach 20000, but not all stations will be present at first.

The programmer decides on random organisation for the file.

Describe **three** steps which show how a new weather station record is added to the file.

1 .....

.....

2 .....

.....

3 .....

.....[3]

Question 8

5 **(a)** Compare sequential and serial methods of file organisation.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) State the most suitable method of file access when a record is referenced by a unique address on a disk-type storage medium.

..... [1]

(c) State the most suitable method of file access when a bank stores its data records in ascending order of account number.

..... [1]

Question 9

3 **(a)** A hashing algorithm is used to calculate storage locations for records in a random access file. It calculates hash values by using the function modulus 3.

The function modulus gives the remainder after integer division.
For example, 1030 modulus 3 = 1. Therefore, the record key 1030 gives a hash value of 1.

Complete the table to show the remaining hash values.

Record key	Hash value
1030	1
1050	
1025	

[2]

(b) Describe what happens, in relation to the storage or retrieval of a record in the file, when the calculated hash value is a duplicate of a previously calculated hash value for a different record key.

.....

.....

.....

.....

.....

.....

.....

..... [4]

Question 10

2 **(a)** Describe how records are organised and accessed in a sequential file.

.....

.....

.....

.....

.....

..... [3]

(b) A hashing algorithm is used to calculate storage locations for records in a random access file. The algorithm calculates hash values using the function modulus 5.

The function modulus gives the remainder after integer division.
For example, 3003 modulus 5 = 3, so the record key 3003 gives a hash value of 3.

Complete the table to show the remaining hash values.

Record key	Hash value
3003	3
1029	
7630	

[2]

Question 11

4 **(a)** Describe sequential and random methods of file organisation.

Sequential file organisation .....

.....

.....

.....

.....

Random file organisation .....

.....

.....

.....

.....

[4]

(b) Outline the process of sequential access for serial and sequential files.

.....

.....

.....

..... [2]

Question 12

3 The location of a record in a random file is determined using a hashing algorithm.

A collision may occur during the process of adding a record.

(a) Outline what is meant by the term **collision** in this context.

.....

.....

.....

..... [2]

(b) Explain how a collision can be dealt with when writing records to a random file.

.....

.....

.....

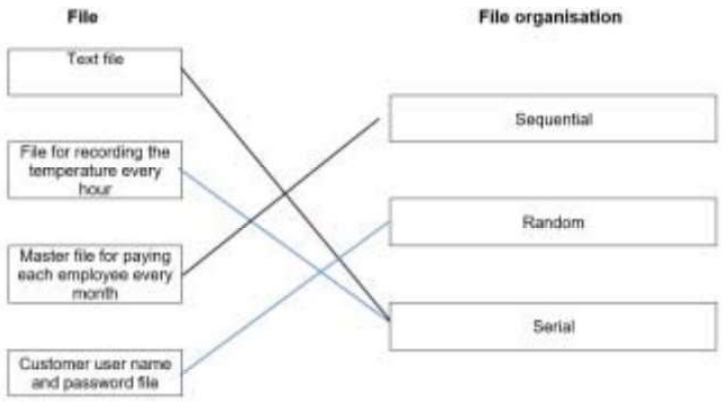
.....

.....

..... [3]

Answer

Answer 1

2	**One** mark for each correct line drawn  	**4**

Answer 2

5(b)(i)	1 mark per bullet point to **max 2** • So the readings are stored in **chronological** order • Easy to add / append each new reading to the end of the file // no further processing is required • Allows the readings to be read in the order that they were taken • Readings do not need to be given further identification as to date / time // no key field needs to be added	**2**
5(b)(ii)	1 mark per bullet point **max 2** • Earliest temperature reading is accessed first • and each successive temperature reading is read (in date / time order) • until the final reading has been accessed	**2**
5(b)(iii)	1 mark for Random 1 mark per bullet point for description to **max 3** • Record locations are calculated • … using a hashing algorithm **on a key field** • If a record cannot be stored / found at that location • … then subsequent locations are searched // closed hash • ... or an overflow area is searched // open hash	**4**

Answer 3

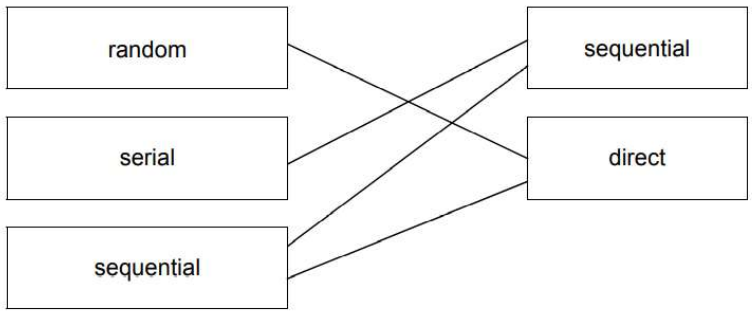
4(a)	Example: Speed of access Just used as a look-up file No need for any serial or sequential processing 1 mark for any valid point	**1**

4(b)(i)	<table><tr><th>CustomerID</th><th>RecordKey</th></tr><tr><td>802139</td><td>2139</td></tr><tr><td>700004</td><td>4</td></tr><tr><td>689998</td><td>89998</td></tr><tr><td>102139</td><td>2139</td></tr></table>	**1**
4(b)(ii)	Minimum value: 0 1 Maximum value: 99999 1	**2**

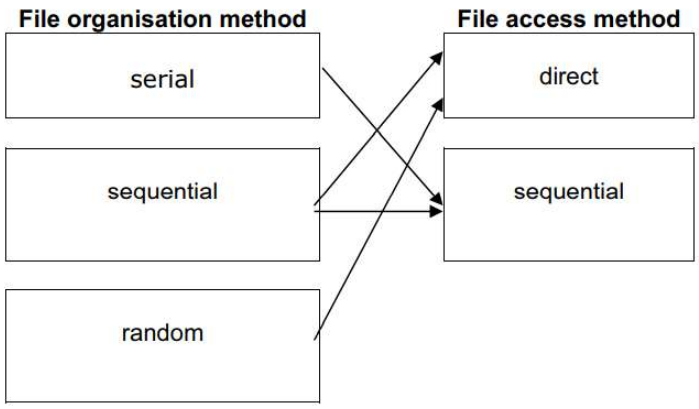
4(b)(iii)	`PROCEDURE InsertRecord(CustomerID : INTEGER)`
	`  RecordKey ← CustomerID MOD 100000`
	`  Success ← FALSE`
	`  // Find position for new record and insert it`
	`  REPEAT`
	`  IF record at position RecordKey is empty`
	`  THEN`
	`  Insert new record at position RecordKey`
	`  Success ← TRUE`
	`  ELSE`
	`  IF RecordKey = 99999`
	`  THEN`
	`  RecordKey ← 0`
	`  ELSE`
	`  RecordKey ← RecordKey + 1`
	`  ENDIF`
	`  ENDIF`
	`  UNTIL Success = TRUE`
	`ENDPROCEDURE`
		4

4(c)(i)	For security If file is hacked then encrypted PIN cannot be used Only encrypted PINs are transmitted and compared 1 mark for any valid point	**Max 2**
4(c)(ii)	1. Customer ID is read from card 2. Customer enters PIN 3. Customer PIN is **encrypted** 4. **Customer ID is hashed** 5. Customer record is located in file 6. **PIN is checked against PIN in record** 7. If match then transaction can proceed	**3**

Answer 4

4(a)	 1 mark for random correct 1 mark for serial correct 2 marks for sequential correct (1 per correct line)	4
4(b)(i)	File A: Serial 1 Meter readings are submitted over time // added to the end of file 1 Stored chronologically 1	3
4(b)(ii)	File B: Sequential 1 Any two points from: Each customer has a unique account number 1 Sorted on Account number 1 High hit rate // Suitable for batch processing monthly statements 1	3
4(b)(iii)	File C: Random 1 Login without waiting // Random organisation allows fastest direct access to required record 1 Low hit rate // Suitable for access to individual records 1	3

Answer 5

4 (a)		1 2 1

(b) (i)	Sequential As all customers get statement … // high hit rate Suitable for batch processing of the records // the records will be processed one after the other File organised using customer's unique ID (as primary key field) // Serial As all customers get statement … // high hit rate Suitable for batch processing of the records // the records will be processed one after the other Order not important	**1** **1** **1** **1** **1** **1** **1** **1** **Max 3**
(ii)	Random Real-time transaction processing Requires fastest access to data No need to search through records	**1** **1** **1** **1** **Max 3**
(iii)	Serial Each new record is appended Transactions are recorded in chronological order File re-organisation not required for each new record // no need for the records to be sorted	**1** **1** **1** **Max 3**

Answer 6

(ii)	• no need to re-sort data every time new data is added • only a small file so searching will require little processing • new records can easily be appended	1 1 1 [max 2]

Answer 7

(ii)	`StationID` is hashed to produce home location If home location is free insert record Else use overflow method to find free location	1 1 1

Answer 8

5(a)	**One** mark for each correct marking point **(Max 4)** • In both serial and sequential files records are stored one after the other … • … and need to be accessed one after the other • Serial files are stored in chronological order • Sequential files are stored with ordered records • … and stored in the order of the key field • In serial files, new records are added in the next available space / records are appended to the file • In sequential files, new records are inserted in the correct position.	**4**
5(b)	Direct (access)	**1**
5(c)	Sequential (access)	**1**

Answer 9

3(a)	**One** mark for each correct hash value (**Max 2**) **Record key** / **Hash value** 1030 / 1 1050 / 0 1025 / 2	**2**
3(b)	**One** mark per mark point (**Max 4**) MP1 A collision occurs when the record key doesn't match the stored record key MP2 … this means the determined storage location has already been used for another record. **If the record is to be stored** MP3 Search the file linearly MP4 … to find the next available storage space (closed hash) MP5 Search the overflow area linearly MP6 … to find next available storage space (open hash) **If the record is to be found** MP7 … search the overflow area linearly (open hash) until the matching record key is found MP8 … search linearly from where you are (closed hash) until the matching record key is found MP9 If not found record is not in file	**4**

Answer 10

<table>
<tr><td>2(a)</td><td>One mark per mark point (Max 3)
MP1 records are stored in a particular order
MP2 the order is determined based on the value in a key field
MP3 records are accessed one after the other
MP4 records can be found by searching from the beginning of the file, record by record,
MP5 … until the required record is found or key field value is exceeded.</td><td>3</td></tr>
<tr><td>2(b)</td><td>One mark for each correct hash value (Max 2)
<table><tr><th>Record key</th><th>Hash value</th></tr><tr><td>3003</td><td>3</td></tr><tr><td>1029</td><td>4</td></tr><tr><td>7630</td><td>0</td></tr></table></td><td>2</td></tr>
</table>

Answer 11

<table>
<tr><td>4(a)</td><td>One mark per mark point – sequential (Max 2)
MP1 Records (in the file) are ordered
MP2 …based on the key field
MP3 A new version (of the file) has to be created to update the file

One mark per mark point – random (Max 2)
MP4 Records are stored in no particular order within the file // There is no sequencing in the placement of the records
MP5 There is a relationship between the key of the record and its location within the file // a hashing algorithm is used to find the location of the record
MP6 Updates to the file can be carried out directly.</td><td>4</td></tr>
<tr><td>4(b)</td><td>One mark per mark point (Max 2)
MP1 Start at the beginning of the file
MP2 …check records linearly
MP3 …until the desired record is found // … processing / updating records as required //… EOF found.</td><td>2</td></tr>
</table>

Answer 12

<table>
<tr><td>3(a)</td><td>One mark per mark point (Max 2)
MP1 A collision is when the two values / data items in the key field for two records (pass through a hashing algorithm and) result in the same hash value
MP2 …so the location identified (by the hashing algorithm) may already be in use // two records cannot occupy the same address.</td><td>2</td></tr>
</table>

3(b)	**One** mark per mark point (**Max 3**) MP1 A process of collision resolution is used MP2 Start at the original hashed storage space MP3 …go through the following spaces in a linear fashion MP4 …and **store the data item** in the first available slot. **OR** MP5 Search the overflow area MP6 …go through the following spaces in a linear fashion MP7 …and **store the data item** in the first available slot. **OR** MP8 Each storage space holds a reference to a collection / chain of items MP9 …which can be searched individually. MP10 The **data item is stored** in the first available space in this chain.	**3**