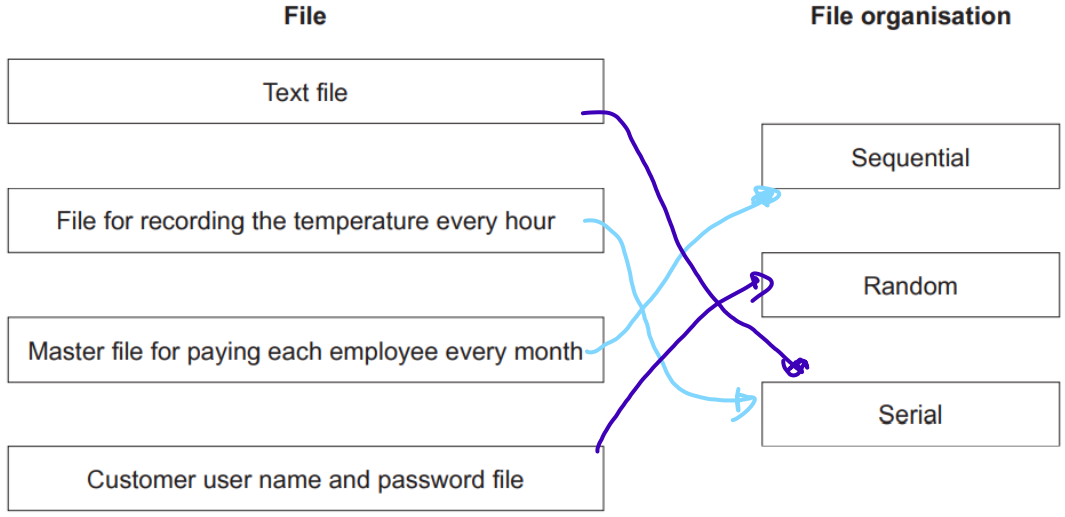
# File Organization And Access

## 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.




[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.

As there is no need to sort the data the data would be appended on basis of time. to allow data to be read in the same order

[2]

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

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

earliest temp access first each temp is accessed in order until the final reading

[2]

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

Method ~~Direct Hash~~ Random organ-

Description The data is stored by ~~pro~~ being processed by an hashing algo to produced a home loc if free data copied, not free the data is copied through overflow or the file is full

On a key field

[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.

To allow for fast access to data when searching

[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 | 4 |
| 689998 | 89,998 |
| 102139 | 2139 |

[1]

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

Minimum value of `RecordKey`: 4 0

Maximum value of `RecordKey`: 89998 99 999

[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 .....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]

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

.....for security purpose if the file is hacked.....
.....then encrypted PIN can't be used.....
.....only encrypted PIN transferred.....
.....[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.

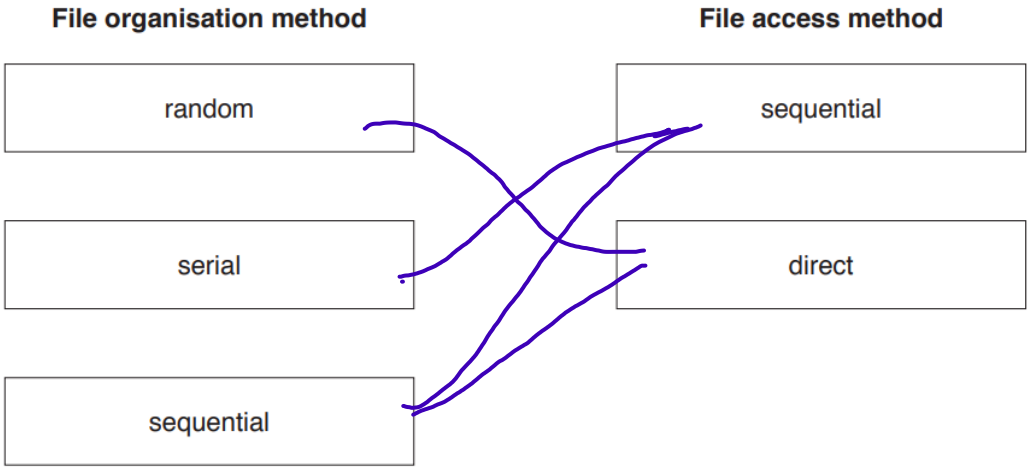
1. Customer ID is read from card.
2. Customer enters PIN.
3. Customer PIN is .....encrypted through an encryption algo.....
4. .....The customer ID is hashkey.....
5. Customer record is located in file.
6. .....The encrypted key is compared to new.....
7. If match then transaction can proceed.

[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).




[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 ..... Serial file organisation

Justification ..... Meter readings submitted over time (stored chronologically)

.....

.....[3]

**(ii)** File B organisation ..... Sequential Organisation

Justification ..... → sorted on Account number (each customer would have a unique ID) Sequential Organisation is suitable for batch processing (high hit rate)

.....[3]

**(iii)** File C organisation ..... Direct access organisation

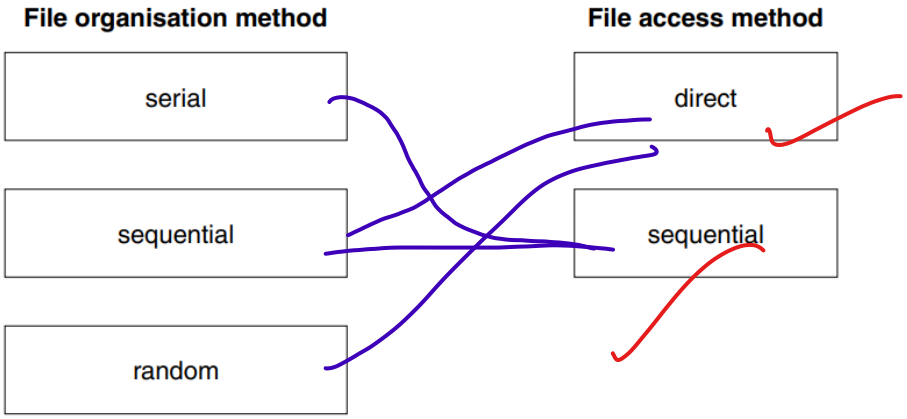
Justification ..... stored on hashed number (from keyfield) → allows for fast indexing & access search. Low hit rate

.....[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.




[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 ..... Sequential Organisation .....

Justification ..... High hit rate suitable for batch process .....
each customer would have a unique ID .....
allowing for sorting. .....
.....[3]

**(ii)** File B organisation ..... Direct Access Organisation (Random) .....

Justification ..... → allows for faster access of data. .....
→ low hit rate. The user wouldn't have to wait .....
that long. no need to search through .....
every records .....[3]

**(iii)** File C organisation ..... Serial Organisation .....

Justification ..... → stored chronologically .....
around everytime a new transaction is made. .....
allowed for faster writing. .....
no need for fields to be sorted .....[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 .....

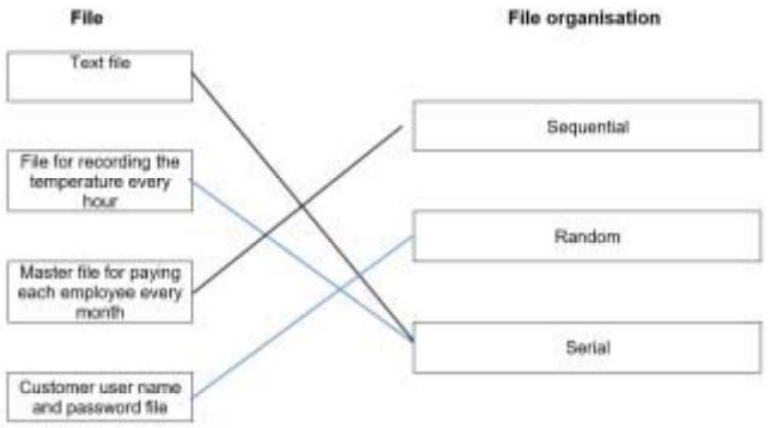
.....

3 .....

.....[3]

# Answer

## Answer 1

| | | |
|---|---|---|
| 2 | **One** mark for each correct line drawn<br>  <br> | **4** |

## Answer 2

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

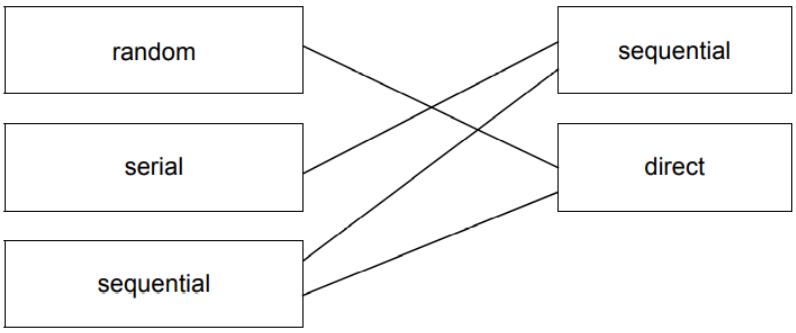
## Answer 3

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

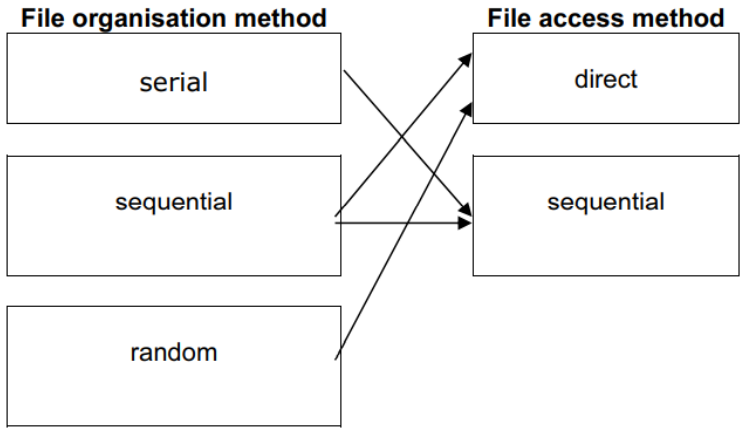
| | | |
|---|---|---|
| 4(b)(i) | <table><tr><th>CustomerID</th><th>RecordKey</th></tr><tr><td>802139</td><td>2139</td></tr><tr><td>700004</td><td>4</td></tr><tr><td>689998</td><td>89998</td></tr><tr><td>102139</td><td>2139</td></tr></table> | **1** |
| 4(b)(ii) | Minimum value: 0 1<br>Maximum value: 99999 1 | **2** |
| 4(b)(iii) | `PROCEDURE InsertRecord(CustomerID : INTEGER)`<br>`  RecordKey ← CustomerID MOD 100000`<br>`  Success ← FALSE`<br>`  // Find position for new record and insert it`<br>`  REPEAT`<br>`  IF record at position RecordKey is` **`empty`**<br>`  THEN`<br>`  Insert new record at position RecordKey`<br>`  Success ← TRUE`<br>`  ELSE`<br>`  IF RecordKey =` **`99999`**<br>`  THEN`<br>`  RecordKey ←` **`0`**<br>`  ELSE`<br>`  RecordKey ←` **`RecordKey`** `+ 1`<br>`  ENDIF`<br>`  ENDIF`<br>`  UNTIL Success = TRUE`<br>`ENDPROCEDURE` | **4** |

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

## Answer 4

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

## Answer 5

| | | |
|---|---|---|
| **4 (a)** | **File organisation method** / **File access method**<br>  <br> | **1**<br>**2**<br>**1** |

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

## Answer 6

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

## Answer 7

| | | |
|---|---|---|
| **(ii)** | `StationID` is hashed to produce home location<br>If home location is free insert record<br>Else use overflow method to find free location | 1<br>1<br>1 |