A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

13.2 File organisation and access

Name: _____ Class: _____ Date: _____ **Total: 14 marks**

KEY WORDS file organisation 文件组织 • hash function 散列函数 • direct access 直接存取 • sequential access 顺序存 • record 记录

Objective

Build the skills to answer exam questions on **file organisation and access** — serial, sequential and random files, and **hashing**.

You must be able to:

- describe **serial**, **sequential** and **random** file organisation
- match **sequential** and **direct** access to a problem
- use a **hash function** and resolve a **collision**

1 Worked examples

Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

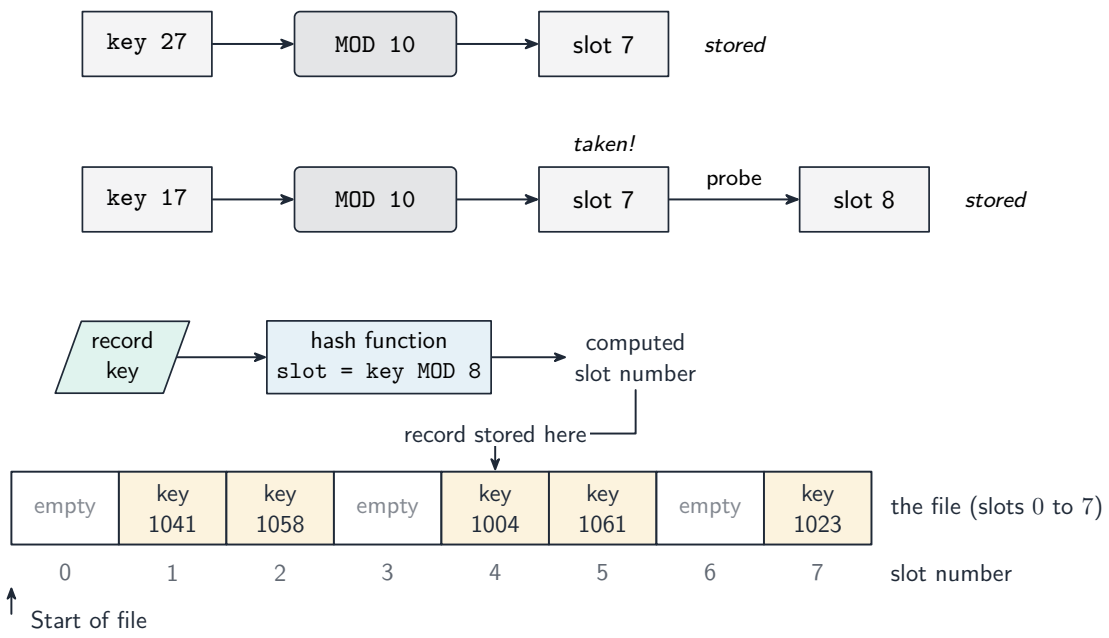
File organisation and access

Organisation	Order of records	Best for	Access
serial	order **added**	logs, audit trails	sequential only
sequential	sorted by a **key**	master files, in-order reports	sequential / direct
random	position from the **key** (hash)	fast single-key lookup	direct

Sequential access reads from start to end; **direct access** jumps straight to a known position.

Hashing

A **hash function** turns a key into an **address**: e.g. `address ← key MOD N`.



Random (direct) file access by address

A **collision** is when two keys hash to the same slot. Resolve it by **linear probing** (try the next free slot) or **chaining** (each slot holds a linked list). Keep the **load factor** (records ÷ slots) below ~70%.

2 Practice

Now apply the methods above.

2.1 State the difference between a **serial** file and a **sequential** file. [2]

2.2 State which file organisation gives the **fastest single-key lookup**. [1]

2.3 State what a **hash function** does. [1]

2.4 State what a **collision** is. [1]

3 Exam-style questions

3.1 A file uses `address ← key MOD 10`. Give the slot number for the keys `45`, `23` and `30`. [3]

3.2 Keys `45` and `35` both hash to slot 5. Explain how **linear probing** stores the second one. [2]

3.3 State **one** advantage and **one** disadvantage of a **random (direct-access)** file. [2]

3.4 A bank must look up an account **quickly** by its account number. State the best file organisation and **justify** your choice. [2]

Solutions

2.1 A **serial** file keeps records in the **order they were added** (unsorted); a **sequential** file keeps them **sorted by a key field**.

2.2 A **random (direct-access)** file.

2.3 It turns a **record key** into an **address/slot** where the record is stored.

2.4 When **two different keys** hash to the **same address/slot**.

3.1 `45 MOD 10 = 5`; `23 MOD 10 = 3`; `30 MOD 10 = 0`.

3.2 `35 MOD 10 = 5`, but slot 5 is **taken** by `45`, so linear probing tries the **next slot** (slot 6), and stores `35` there (the first free slot, wrapping if needed).

3.3 Advantage: very **fast direct lookup** by key. Disadvantage: reading in **key order** is harder, and collisions must be handled (space may be wasted).

3.4 A **random (direct-access)** file using a **hash** of the account number, because it gives near-instant lookup of a single record without searching the whole file.