

A-Level Computer Science

Name: _____ Score: _____

1. Why are user-defined types needed? Select **all** that apply.

- ☐ they group values of different types that describe one thing
- ☐ the declaration says what the variable is, not just how it is stored
- ☐ they make the program run faster
- ☐ they restrict a field to its legal values, so the compiler catches more mistakes

Tick every correct option.

2. A serial file stores records:

- ☐ in the order they were added, with no sorting
- ☐ in random memory addresses
- ☐ sorted by a key field
- ☐ at positions computed from a key

3. A hash function:

- ☐ turns a record's key into a storage address
- ☐ encrypts the record
- ☐ compresses the data
- ☐ sorts the file by key

4. A floating-point number is stored as:

- ☐ a string of digits
- ☐ a mantissa multiplied by 2 raised to an exponent
- ☐ a denary fraction
- ☐ a single integer

5. An enumerated type holds:

- ☐ a memory address
- ☐ a value from a fixed list of named constants
- ☐ a group of different fields
- ☐ any string at all

6. A random (direct-access) file places each record:

- ☐ in alphabetical order only

- ☐ at a position computed from its key, giving very fast direct lookup
- ☐ always at the end
- ☐ in arrival order

7. What makes a hash function a good one? Select **all** that apply.

- ☐ the same key always gives the same address
- ☐ it never produces a collision
- ☐ it is fast to compute
- ☐ it spreads keys evenly over the slots

Tick every correct option.

8. Written as a normalised fraction times a power of two, $2.5 = 0.101 \times 2^n$ (the fraction in binary). The exponent n is _____.

9. A pointer stores the memory address of another variable, and dereferencing it (e.g. $\text{p}^{\wedge}$) reaches the variable it points to.

- ☐ True ☐ False

10. A sequential file can be read by sequential access and also by direct access.

- ☐ True ☐ False

A-Level Computer Science

1. **they group values of different types that describe one thing; the declaration says what the variable is, not just how it is stored; they restrict a field to its legal values, so the compiler catches more mistakes**

Restriction, grouping and self-documentation. Speed is not the reason; correctness and clarity are.

2. **in the order they were added, with no sorting**

Serial files keep records in arrival order – fast to append, slow to search. Sequential files are sorted by key.

3. **turns a record's key into a storage address**

A hash function maps a key to an address, enabling near-instant direct lookup.

4. **a mantissa multiplied by 2 raised to an exponent**

$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$ – binary scientific notation, with both parts stored in two's complement.

5. **a value from a fixed list of named constants**

An enumerated type restricts a variable to one of a fixed set of named values (e.g. days of the week).

6. **at a position computed from its key, giving very fast direct lookup**

A key is converted (often by a hash) to a position, so a single-key lookup is very fast.

7. **the same key always gives the same address; it is fast to compute; it spreads keys evenly over the slots**

Fast, deterministic and evenly spread. No realistic hash avoids collisions entirely, which is why every design includes a resolution strategy.

8. **2**

2.5 is 10.1 in binary; sliding the point two places left gives 0.101, so the exponent is 2 and the mantissa is '01010000'.

9. **True**

Pointers (with NULL marking "points to nothing") are how dynamic structures like linked lists and trees are built.

10. **True**

Sequential access serves serial and sequential files; direct access serves sequential and random files. Only a serial file is limited to one method.