

10 Data Types and Structures — Part 1

Name: _____ Class: _____ Date: _____

Vocabulary 词汇

Write each word three times. 把每个单词抄写三遍。

English	中文	Write it 3 times (English)		
data type	数据类型	_____	_____	_____
record	记录	_____	_____	_____
array	数组	_____	_____	_____
element	元素	_____	_____	_____
index	索引	_____	_____	_____
file	文件	_____	_____	_____
abstract data type	抽象数据类型	_____	_____	_____
stack	栈	_____	_____	_____
queue	队列	_____	_____	_____
linked list	链表	_____	_____	_____

Recall

At IGCSE you stored single values in variables. Bring this with you:

- A variable holds one value (a number, a character, ...).
- You worked with lists of items informally.

At AS we store whole collections of data and meet some useful data structures.

New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

■ 1 Data types, records and arrays

- a **data type** 数据类型 says what kind of value is stored (integer, real, char, string, Boolean).
- a **record** 记录 groups several fields of *different* types under one name (like one row of a table).
- an **array** 数组 stores many values of the *same* type in one list; you reach each **element** 元素 by its **index** 索引 (its position). A 2D array is a grid of rows and columns.

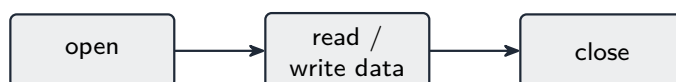
		column index			
		1	2	3	4
row index	1				
	2			99	
	3				

← Grid[2,3] ← 99

Worked example. `scores = [70, 85, 90]` is an array (all integers); a **Student** record holds `name` (string), `age` (integer) and `passed` (Boolean) together in one structure.

■ 2 Files

A **file** 文件 stores data on secondary storage, so it is kept after the program ends. You open a file, read or write data, then close it.



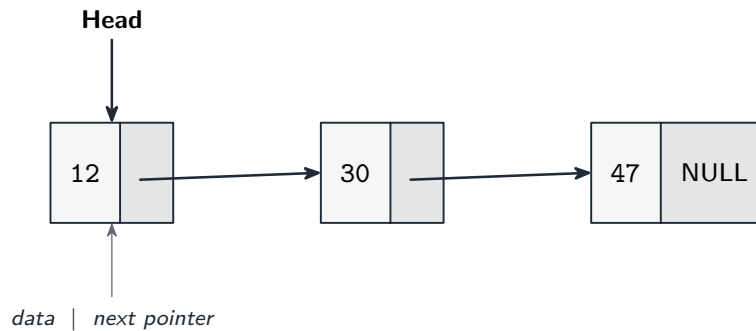
Worked example. A game saves its high-score table to a file when it closes, so the scores are still there the next time you play.

■ 3 Abstract data types: stacks and queues

An **abstract data type** 抽象数据类型 (ADT) is described by what it *does*, not how it is stored:

- a **stack** 栈 is last-in, first-out (LIFO) — like a pile of plates;
- a **queue** 队列 is first-in, first-out (FIFO) — like a line of people.

A **linked list** 链表 stores items where each one points to the next.



Worked example. Pressing “undo” again and again uses a stack — your most recent action is undone first; a printer queue uses a queue — the first job sent is printed first.

Watch out: in many languages an array index starts at 0, not 1 — so `A[0]` is the first element. Always check where the counting begins.

Practice

10.1 Give two examples of a data type. [2]

10.2 An array `A` holds the values 10, 20, 30, 40 in positions 1 to 4. State the value at index 2. [1]

10.3 A stack is LIFO and a queue is FIFO. State which one removes the **most recently added** item first. [1]

10.4 State one difference between an array and a record. [2]

10.5 State one thing a file lets you do that an ordinary variable does not. [1]

10.6 Explain *why* a stack is the right structure for an “undo” feature. [2]

10.7 Challenge. Items A, B, C are pushed onto a stack in that order, then two items are popped. State which two are removed, and in what order. [2]
