A-LEVEL COMPUTER SCIENCE • HOLIDAY HOMEWORK 假期作业

10 Data Types and Structures – Part 2

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)		
push	入栈			
pop	出栈			
overflow	溢出			
underflow	下溢			
enqueue	入队			
dequeue	出队			
circular array	循环数组			
end of file	文件结束			

Recall

In **Part 1** you met data types, records, arrays (1-D and 2-D), files, and three abstract data types — the stack, queue and linked list. Part 2 looks at the operations on a stack and a queue, and at reading and writing files.

New ideas

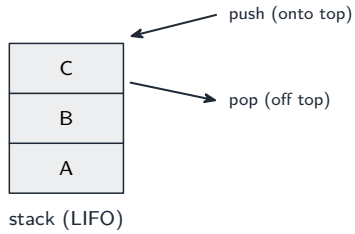
Read these first. Each idea has a short worked example. Then do the Practice.

1 Stack operations (LIFO)

A stack is **last-in, first-out**. Two operations:

- **push** 入栈 — add an item to the top;
- **pop** 出栈 — remove the item from the top.

Trying to push onto a full stack is **overflow** 溢出; trying to pop an empty stack is **underflow** 下溢. (Uses: undo history, function return addresses.)



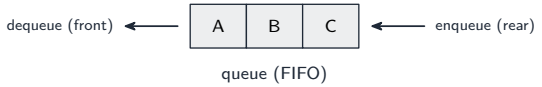
Worked example. Start empty; `push(5)`, `push(8)` → the top is 8. `pop()` removes and returns 8, leaving 5 on the stack.

2 Queue operations (FIFO)

A queue is **first-in, first-out**. Two operations:

- **enqueue** 入队 — add an item at the rear;
- **dequeue** 出队 — remove the item from the front.

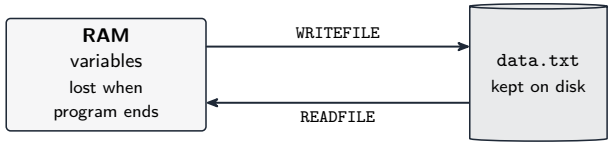
A **circular array** 循环数组 lets the front and rear pointers wrap back to the start of the array, so no space is wasted. (Uses: print queues, scheduling.)



Worked example. `enqueue(5)`, `enqueue(8)` → the front is 5. `dequeue()` removes and returns 5, leaving 8.

3 Handling files

Data in variables is lost when the program ends, so to keep it you write to a **file**. Open it, read or write, then close it; test for the **end of file** 文件结束 (EOF) while reading.




```
OPENFILE "data.txt" FOR READ
WHILE NOT EOF("data.txt") DO
    READFILE "data.txt", Line
    OUTPUT Line
ENDWHILE
CLOSEFILE "data.txt"
```

Worked example. The loop above prints every line of `data.txt`; the `NOT EOF` test stops it cleanly at the end instead of trying to read past the last line.

Watch out: check for EOF *before* each read — reading past the end of a file causes an error.

Practice

10.1 Items A, B, C are pushed onto a stack in that order, then one item is popped. State which item is removed. [1]

10.2 Items A, B, C are enqueued in that order, then one is dequeued. State which item is removed. [1]

10.3 State what is meant by stack (a) overflow and (b) underflow. [2]

10.4 State why a queue is usually stored in a circular array. [2]

10.5 State why a program must close a file after writing to it. [1]

10.6 Explain *why* a stack (not a queue) is used to remember where to return after each function call. [2]

10.7 Challenge. Items $1, 2, 3, 4$ are pushed onto a stack in that order. Three items are then popped and immediately enqueued onto a queue, in the order they were popped. One item is then dequeued. State which item comes out. [2]