

4.16 Recursion

Name: _____ Class: _____ Date: _____

Total: 9 marks

KEY WORDS base case 基准情形 • recursion 递归 • recursive case 递归情形

Objective

Build the skills to answer exam questions on **recursion**.

You must be able to:

- understand that a **recursive** 递归 method calls itself
- identify the **base case** 基准情形 that stops the recursion
- identify the **recursive case** 递归情形 that moves toward the base case

1 Worked examples

Study these first. Each one shows the method for a question type used later.

■ Recursion

A **recursive** method calls **itself**. It needs:

- a **base case** that stops the recursion, and
- a **recursive case** that moves the problem toward the base case.

```
int fact(int n) {  
    if (n <= 1) return 1;      // base case  
    return n * fact(n - 1);    // recursive case  
}
```

Without a base case, the recursion never stops.

■ Unwinding the calls

fact(3) calls **fact(2)**, which calls **fact(1)**. The base case returns 1, then the waiting calls finish in reverse: **fact(2) = 2 * 1 = 2**, then **fact(3) = 3 * 2 = 6**. Each call pauses until the one below it returns.

2 Practice

2.1 Define a recursive method. [1]

2.2 State the purpose of a base case. [1]

2.3 For the `fact` example above, state the value of `fact(3)`. [2]

3 Exam-style questions

3.1 A recursive method is one that [1]

- | | |
|---|---|
| A | B |
| C | D |
- A** never returns
 - B** calls itself
 - C** always uses a loop
 - D** must be `static`

3.2 The part that stops the recursion is the [1]

- | | |
|---|---|
| A | B |
| C | D |
- A** recursive case
 - B** base case
 - C** parameter
 - D** return type

3.3 `int f(int n) { if (n == 0) return 0; return n + f(n - 1); }`

(a) State `f(0)`. [1]

(b) State `f(3)`. [1]

(c) Name the case where `n == 0`. [1]