

一〇四學年度 第一學期 期末上機考試
 計算機概論實習 資訊一甲

共四題，第 n 題請使用檔案名稱 “**fxam_dxxxxxxx_n.c**”，“**xxxxxxx**” 為你的學號。同時，請在每個程式原始碼的第一行，以**註解方式**寫上你的**學號及姓名**。未使用規定之檔名或未寫上學號及姓名者，本次成績將被扣 **10%** 的分數。當你作答完畢後，請將**程式的原始碼 (source code, 即 fxam_dxxxxxxx_n.c 檔)** 上傳到教師機台。

1. (25 points) 你可使用 **fxam_skeleton_1.c**，並將此程式架構改名為 **fxam_dxxxxxxx_1.c**。假設一個整數函數 $f(n)$ 的定義如下：

$$f(n) = \begin{cases} n^2 & \text{if } n \text{ is even} \\ n^3 & \text{if } n \text{ is odd} \end{cases}$$

如果 n 是一個大於或等於 0 的整數，數列 $f(0), f(1), f(2), \dots, f(n)$ 的和稱為「偶平方-奇三方和」(sum-of-even-square-odd-cube)，簡稱為 **sesoc**，寫一個 C 語言的遞迴副程式 **int sesoc(int n)** 以計算數列 $f(0), f(1), f(2), \dots, f(n)$ 的「偶平方-奇三方的和」。請上傳 **fxam_dxxxxxxx_1.c**，程式執行範例：

```

D:\課程\A1-計算機概論\Exam\LabExam\Au104\final\fxam...
Enter integer n (stop when n<0): 0
The sum-of-even-square-odd-cube of 0: 0

Enter integer n (stop when n<0): 1
The sum-of-even-square-odd-cube of 1: 1

Enter integer n (stop when n<0): 2
The sum-of-even-square-odd-cube of 2: 5

Enter integer n (stop when n<0): 3
The sum-of-even-square-odd-cube of 3: 32

Enter integer n (stop when n<0): 4
The sum-of-even-square-odd-cube of 4: 48

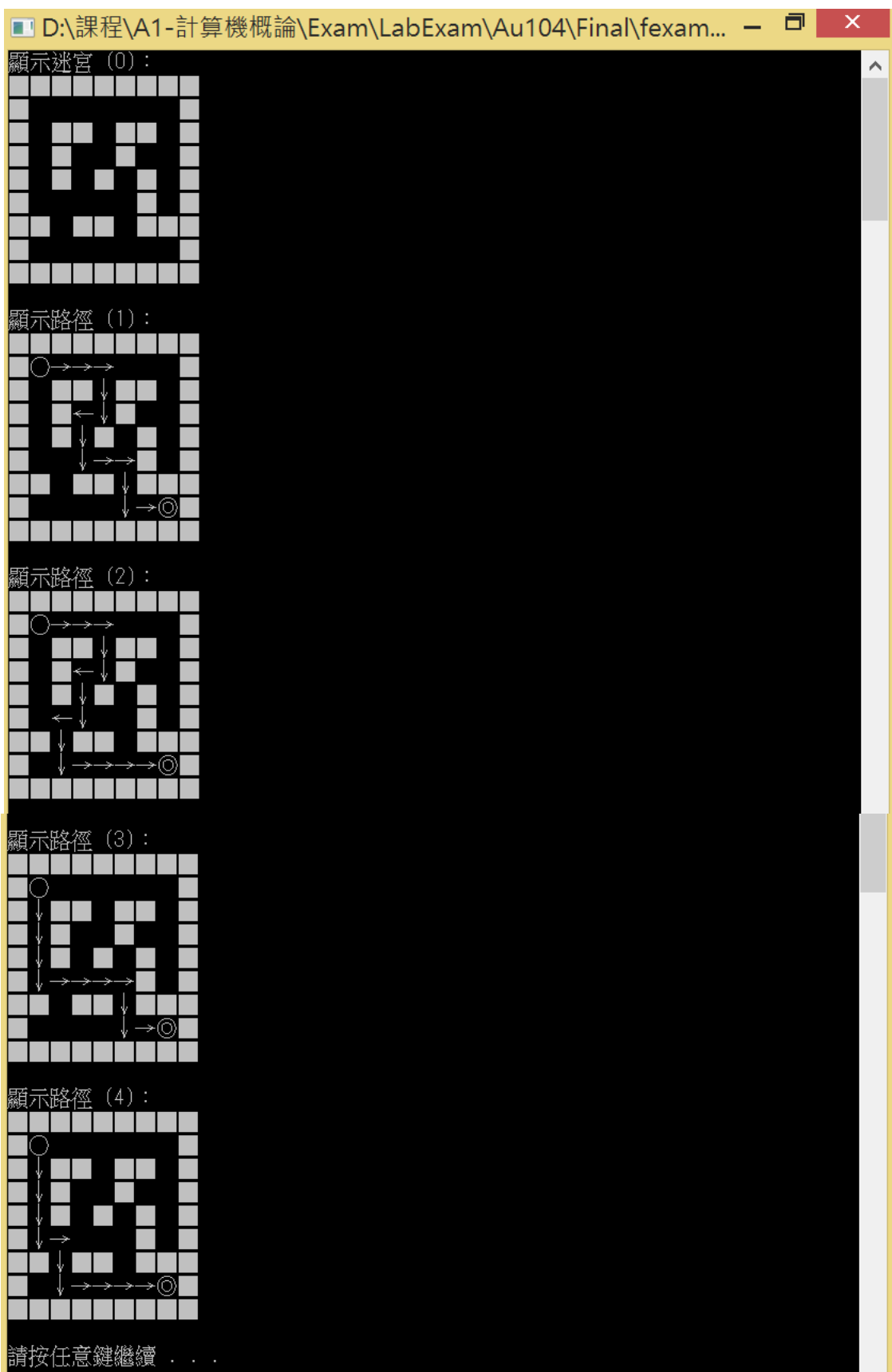
Enter integer n (stop when n<0): 5
The sum-of-even-square-odd-cube of 5: 173

Enter integer n (stop when n<0): 6
The sum-of-even-square-odd-cube of 6: 209

Enter integer n (stop when n<0): -1
請按任意鍵繼續 . . . █
  
```

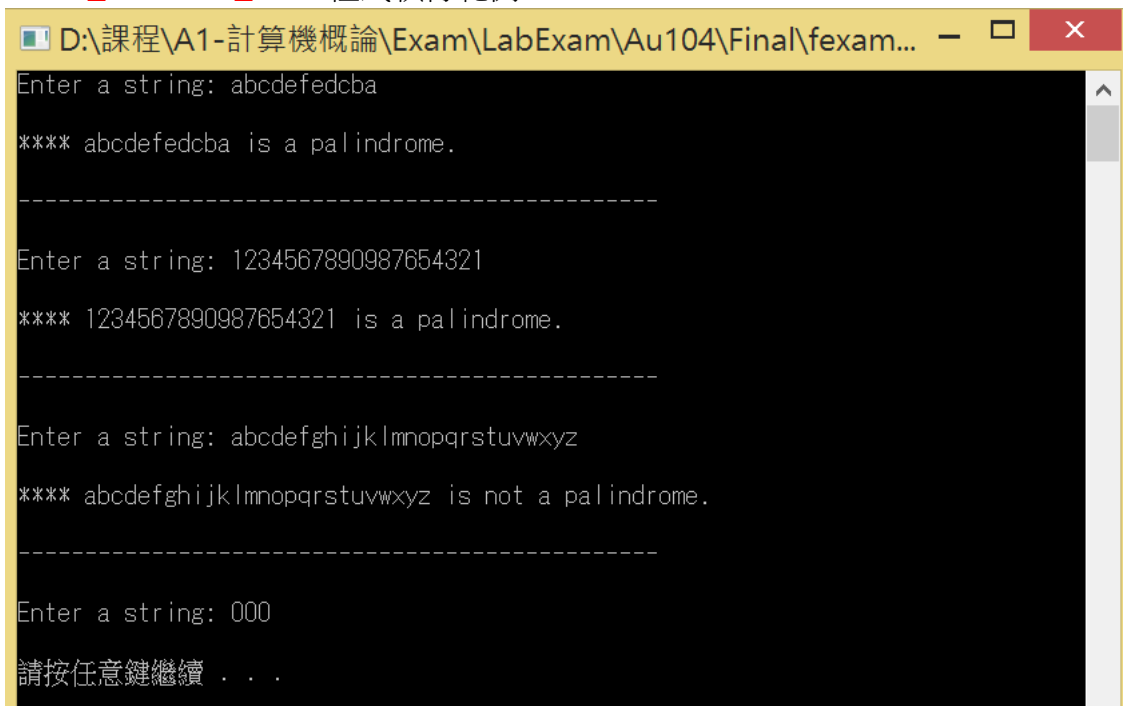
(續下頁)

2. (25 points) 你可使用 `fexam_skeleton_2.c`, 並將此程式架構改名為 `fexam_dxxxxxx_2.c`。程式架構 `fexam_skeleton_2.c` 是一個老鼠走迷宮 (mouse maze) 的進化版, 迷宮的表示改成字元陣列 (character array), 'w' 代表牆壁, '.' 代表可行走的路徑; 迷宮路徑從 `maze[1][1]` 開始, 至 `maze[7][7]` 結束, 我們使用 's' 代表開始的位置, 'e' 代表結束的位置; 在路徑的行走中, 我們使用 'r', 'd', 'l', 'u' 分別表示「右」、「下」、「左」、「上」的方向。使用 C 語言的 **switch-case 控制結構 (control structure)** 完成 `print_maze` 的副程式, 使用以下文字符號表示迷宮方格的內容: 牆壁: ■、開始位置: ○、結束位置: ◎、往右方向: →、往下方向: ↓、往左方向: ←、往上方向: ↑、其他: " "。請上傳 `fexam_dxxxxxx_2.c`, 程式執行範例: (請見下頁)



(續下頁)

3. (25 points) 你可使用 `fexam_skeleton_3.c`, 並將此程式架構改名為 `fexam_dxxxxxxx_3.c`。程式架構 `fexam_skeleton_3.c` 是一個迴文 (palindrome) 的程式架構, 所謂是迴文一個從前面讀和從後面讀都是一樣的字串。例如, `abcdedcba`、`vollov`、`945626549` 都是迴文; 但是 `abcd`、`123456123456`、`abcdefghijklmdefcba` 都不是迴文。已知在函式庫 (library) `<string.h>` 中 副程式 `int strlen(char *str);` 回傳字串 `str` 的長度 (length), 使用 `while` 迴圈完成檢查迴文的副程式 `int isPalindrome(char *str)`。不可使用堆疊和其他資料結構。請上傳 `fexam_dxxxxxxx_3.c`, 程式執行範例:



```
D:\課程\A1-計算機概論\Exam\LabExam\Au104\Final\fexam...
Enter a string: abcdefedcba
**** abcdefedcba is a palindrome.
-----
Enter a string: 1234567890987654321
**** 1234567890987654321 is a palindrome.
-----
Enter a string: abcdefghijklmopqrstuvwxyz
**** abcdefghijklmopqrstuvwxyz is not a palindrome.
-----
Enter a string: 000
請按任意鍵繼續 . . .
```

4. (25 points) 你可使用 `fexam_skeleton_4.c`, 並將此程式架構改名為 `fexam_dxxxxxxx_4.c`, 建立一個專案 (project) 並取名為 `fexam_dxxxxxxx_4.dev`。假設有 n 個大小不同的圓盤和三根柱子, 依大小順序, 大的圓盤在小的圓盤之下, 開始時放置於第一根柱子。河內塔 (tower of Hanoi) 的問題就是將圓盤從第一根柱子搬到第三根柱子, 每次只能移動一個圓盤, 且可使用第二根柱子作為中途放置的柱子; 但是在整個移動過程中, 較大的盤子都必須在較小的盤子之下。使用堆疊 (stack) 運算寫一個河內塔的專案程式, 輸出圓盤移動的步驟, 每一步驟輸出三根柱子自下而上的即時狀態。完成河內塔的副程式:

```
void hanoi(int n, stack *sour, stack *aux, stack *dest,  
           stack *ppeg1, stack *ppeg2, stack *ppeg3);
```

堆疊運算可使用 `hanoi_stack_array.h` 和 `hanoi_stack_array.c` 檔案。請上傳 `fexam_dxxxxxxx_4.dev`、`hanoi_stack_array.h`、`hanoi_stack_array.c`、`fexam_dxxxxxxx_4.c` 四個檔案, 程式執行範例:(請見下頁)

```
D:\課程\A1-計算機概論\Exam\LabExam\Au104\Final\fexam... - [X]
Please enter the number of disks: 3

Initial pegs:
Peg A:  3  2  1
Peg B:
Peg C:

*****
>>>> Step 1: Move disk 1 from Peg A to Peg C

Peg A:  3  2
Peg B:
Peg C:  1

*****
>>>> Step 2: Move disk 2 from Peg A to Peg B

Peg A:  3
Peg B:  2
Peg C:  1

*****
>>>> Step 3: Move disk 1 from Peg C to Peg B

Peg A:  3
Peg B:  2  1
Peg C:

*****
>>>> Step 4: Move disk 3 from Peg A to Peg C

Peg A:
Peg B:  2  1
Peg C:  3

*****
>>>> Step 5: Move disk 1 from Peg B to Peg A

Peg A:  1
Peg B:  2
Peg C:  3

*****
>>>> Step 6: Move disk 2 from Peg B to Peg C

Peg A:  1
Peg B:
Peg C:  3  2

*****
>>>> Step 7: Move disk 1 from Peg A to Peg C

Peg A:
Peg B:
Peg C:  3  2  1

Finish!!!

請按任意鍵繼續 . . .
```