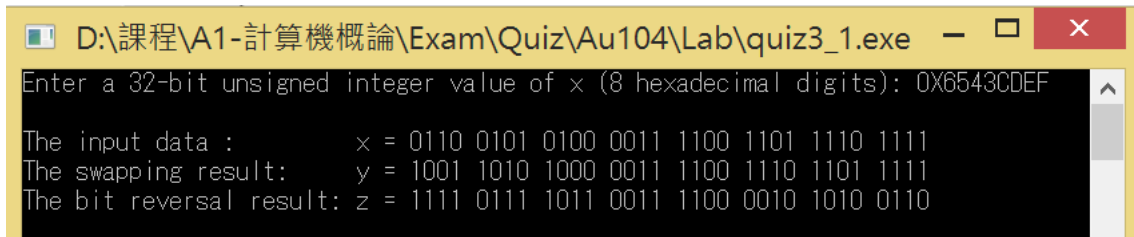


104 學年度 第一學期 資訊一甲
計算機概論實習 第三次平時上機考試

第一題請使用檔案名稱 **quiz3_dxxxxxxx_1.c**”，第二題請使用檔案名稱 **“quiz3_dxxxxxxx_2.c”**。“**dxxxxxxx**” 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, .c 檔)** 上傳到教師機台。

1. (50 points) 你可使用 **quiz3_skeleton_1.c** 的程式架構，並先將其改名為 **quiz3_dxxxxxxx_1.c**。寫一個 C 語言的程式宣告 32 位元三個非負整數 (non-negative integer) 變數 x, y, 及 z。讀入一個八個位數的十六進位非負整數 (non-negative hexadecimal integer) 存入 x，即 $x = X_{31}X_{30}X_{29}X_{28} \dots X_3X_2X_1X_0$ ；將 y 設成 x 兩兩相鄰的位元對調 (bit swap) 的數，即 $y = X_{30}X_{31}X_{28}X_{29} \dots X_2X_3X_0X_1$ ；將 z 設成 x 位元反向順序 (bit reversal) 的數，即 $z = X_0X_1X_2X_3 \dots X_{28}X_{29}X_{30}X_{31}$ 。輸出 x, y, 和 z 最後的二進位值 (binary value)。程式執行範例：



```
D:\課程\A1-計算機概論\Exam\Quiz\Au104\Lab\quiz3_1.exe
Enter a 32-bit unsigned integer value of x (8 hexadecimal digits): 0X6543CDEF
The input data :      x = 0110 0101 0100 0011 1100 1101 1110 1111
The swapping result:  y = 1001 1010 1000 0011 1100 1110 1101 1111
The bit reversal result: z = 1111 0111 1011 0011 1100 0010 1010 0110
```

(續下一頁)

2. (50 points) 你可使用 **quiz3_skeleton_2.c** 的程式架構，並先將其改名為 **quiz3_dxxxxxxx_2.c**。使用 C 語言的邏輯運算 (logical operations) 寫一個十六位元加法器 (16-bit adder) 模擬程式。十六位元加法器之設計請參考文件 **adder.pdf**。本程式中一個十六位元的非負整數 (0 與 65535 之間) 是以 C 語言中的三十二位元整數來代表，且兩個十六位元非負整數的和為十六或十七位元整數。

一個十六位元加法器之設計將使用 16 個單一位元的全加器 (one-bit full adder)，以序列 (sequential) 的方式從最右邊起將兩個單一位元相加，直到最左邊的位元。全加器有三個輸入位元，包括兩個運算位元 (operand bit) 及一個進位位元 (carry bit)；並有兩個輸出位元，包括一個和位元 (sum bit) 和一個進位位元。在起始時，輸入進位位元設定為 0；一個全加器的輸出進位位元成為下一個全加法的輸入進位位元。假設：

A ：第一個輸入運算位元，

B ：第二個輸入運算位元，

C_i ：輸入進位位元，

S ：輸出和位元，

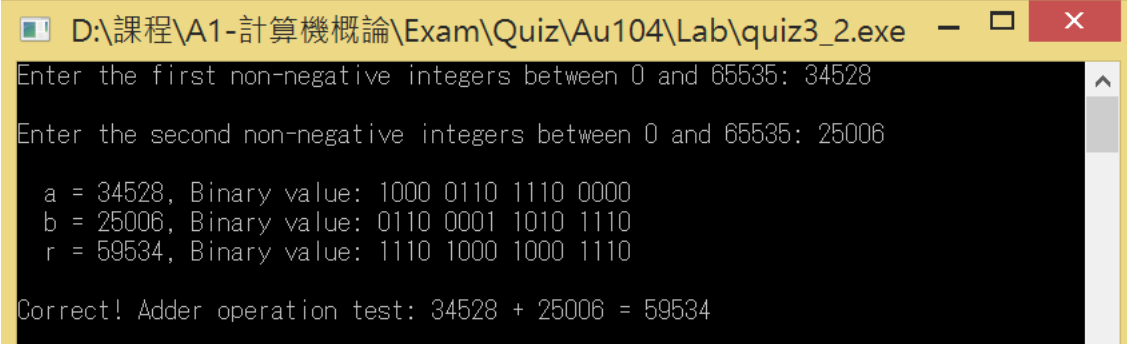
C_o ：輸出進位位元，

則全加器的計算公式如下：

$$S = (A \oplus B) \oplus C_i$$

$$C_o = (A \wedge B) \vee (C_i \wedge (A \oplus B))$$

完成 **quiz3_skeleton_2.c** 中的兩個副程式 **fullAdder** 和 **adder**。程式執行範例：



```
D:\課程\A1-計算機概論\Exam\Quiz\Au104\Lab\quiz3_2.exe
Enter the first non-negative integers between 0 and 65535: 34528
Enter the second non-negative integers between 0 and 65535: 25006
a = 34528, Binary value: 1000 0110 1110 0000
b = 25006, Binary value: 0110 0001 1010 1110
r = 59534, Binary value: 1110 1000 1000 1110
Correct! Adder operation test: 34528 + 25006 = 59534
```