

## Exercise #08

- You don't need to turn in your homework, but you should practice all problems because they may probably appear in the later exam. 作業自己練習不用交，之後考試可能會出現類似題目
- To ask a question, please provide the code you wrote. It doesn't matter if you write it incompletely, but you need to write a version with your own ideas. 有問題可以 email 跟老師討論，但請附上自己寫的程式碼

### Problem 1. *Valid Palindrome*

- Given a non-empty string *s*, you may delete at most one character. Judge whether you can make it a palindrome.

■ **Note:** The string will only contain lowercase characters a-z. The maximum length of the string is 50.

Example 1:  
Input: "aba"  
Output: True

Example 2:  
Input: "abca"  
Output: True  
Explanation: You could delete the character 'c'.

### Problem 2. *matrix addition*

- Write a function **matrix\_add(a, b)** to add two 3x3 matrices.
- Define two 3x3 int matrices and give the values in main()
- Then, pass these two matrices to function **matrix\_add(a, b)**
- **matrix\_add()** add two matrices like  $a = a + b$
- Main() outputs the result.
- 寫一個程式，main定義二個3x3 int陣列a, b，主程式輸入二陣列值，傳給function，function執行陣列相加( $a = a + b$ )，再由主程式輸出arrayA

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \end{bmatrix}$$

### Problem 3. *matrix transpose*

- Write a function **transpose(A, B)** to transpose a matrix A to B (轉置矩陣)

$$\begin{bmatrix} 2 & 3 & 5 \\ 1 & 6 & 7 \\ 2 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1 & 2 \\ 3 & 6 & 7 \\ 5 & 8 & 9 \end{bmatrix}$$

**Problem 4.**

- main() takes charge of input and output. Function processes the string content. String length < 10
- Write a function stradd(a, b, c, size); perform the operation  $c = a + b$ 
  - void stradd(char a[], char b[], char c[], int sz);
  - a="abc", b="123" → c="abc123"

**Problem 5. Partition Array Into Three Parts With Equal Sum**

- Write a function three\_parts(int A[], int size). Given an array A of integers, return true if and only if we can partition the array into three non-empty parts with equal sums.
- Formally, we can partition the array if we can find indexes  $i+1 < j$  with  $(A[0] + A[1] + \dots + A[i] == A[i+1] + A[i+2] + \dots + A[j-1] == A[j] + A[j+1] + \dots + A[A.length - 1])$
- Example 1:
  - Input: [0,2,1,-6,6,-7,9,1,2,0,1] Output: true
  - Explanation:  $0 + 2 + 1 = -6 + 6 - 7 + 9 + 1 = 2 + 0 + 1$
- Example 2:
  - Input: [0,2,1,-6,6,7,9,-1,2,0,1] Output: false
- Example 3:
  - Input: [3,3,6,5,-2,2,5,1,-9,4] Output: true
  - Explanation:  $3 + 3 = 6 = 5 - 2 + 2 + 5 + 1 - 9 + 4$