

韌體實驗第8題

Ting-Yu Lin, Adjunct Lecturer (林庭宇)

Office: AMOS Lab E517

Email: tonylin0413@gmail.com



AMOS Lab.
Advanced Mixed-Operation System Lab.
Dept. of Electrical and Computer Engineering,
Tamkang University, Taiwan



**BIDAS Technology
GrounTruth Workshop**
Computer Vision
Image Processing
Edge Computing

第八題

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B->A, C->B, ..., Z->Y, A->Z)，得到 VICTORIOUS。

輸入兩個字串，輸出 YES 或者 NO。

輸入 (註):

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出:

YES

NO

YES

YES

NO

第八題

題目有誤

沒有重排
也沒有映射
只有統計

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：
~~JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B→A, C→B, ..., Z→Y, A→Z)，得到 VICTORIOUS。~~

輸入兩個字串，輸出 YES 或者 NO。

輸入（註）：

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出：

YES

NO

YES

YES

NO

第八題

題目有誤

沒有重排
也沒有映射
只有統計

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：
~~JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B→A, C→B, ..., Z→Y, A→Z)，得到 VICTORIOUS。~~

輸入兩個字串，輸出 YES 或者 NO。

輸入（註）：

JWPUDJSTVP
VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出：

YES

NO

YES

YES

NO

第八題

題目有誤

沒有重排
也沒有映射
只有統計

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：
~~JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B→A, C→B, ..., Z→Y, A→Z)，得到 VICTORIOUS。~~

輸入兩個字串，輸出 YES 或者 NO。

輸入（註）：

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出：

YES

NO

YES

YES

NO

第八題

題目有誤

沒有重排
也沒有映射
只有統計

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：
~~JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B→A, C→B, ..., Z→Y, A→Z)，得到 VICTORIOUS。~~

輸入兩個字串，輸出 YES 或者 NO。

輸入（註）：

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出：

YES

NO

YES

YES

NO

第八題

題目有誤

沒有重排
也沒有映射
只有統計

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：
~~JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B→A, C→B, ..., Z→Y, A→Z)，得到 VICTORIOUS。~~

輸入兩個字串，輸出 YES 或者 NO。

輸入（註）：

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出：

YES

NO

YES

YES

NO

第八題

題目有誤

沒有重排
也沒有映射
只有統計

8. 【古老的密碼/Ancient Cipher】

給定兩個長度相同且不超過 100 的字串，判斷是否能把其中一個字串各個字母重排，然後對 26 個字母做一個一一映射，使得兩個字串相同。例如：
~~JWPUDJSTVP 重排後可以得到 WJDUPSJPVT，然後把每個字母映射到它前一個字母 (B→A, C→B, ..., Z→Y, A→Z)，得到 VICTORIOUS。~~

輸入兩個字串，輸出 YES 或者 NO。

輸入（註）：

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出：

YES

NO

YES

YES

NO

第八題

J W P U D J S T V P

0 1 0 2 0 2 0 1 1 1 1 1

→ ... D ... J ... P ... S T U V W

統計

→ 2 2 1 1 1 1 1 1 0 0 0 0

排序

↕ 依序比對

V I C T O R I O U S

0 1 0 2 0 2 0 1 1 1 1 1

2 2 1 1 1 1 1 1 0 0 0 0

... C ... I ... O ... R S T U V

輸入 (註):

JWPUDJSTVP
VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出:

YES

NO

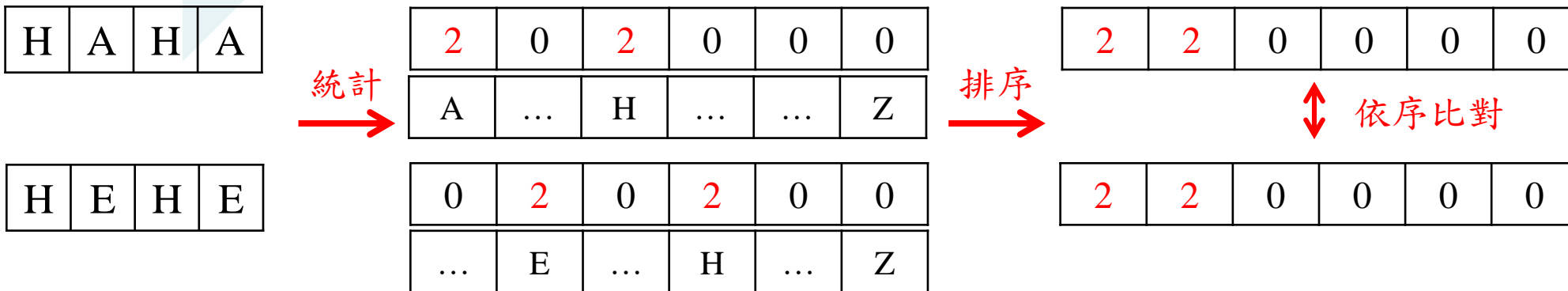
YES

YES

NO

YES !

第八題



輸入 (註):

JWPUDJSTVP

VICTORIOUS

MAMA

ROME

HAHA

HEHE

AAA

AAA

NEERCISTHEBEST

SECRETMESSAGES

輸出:

YES

NO

YES

YES

NO

YES !

第八題

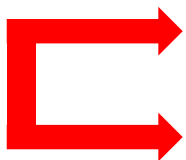
➤ 題目解析：

(1) 沒有重排、也沒有映射、只有統計

(2) 解題要點：統計、排序、比對



第八題



```

1  #define _CRT_SECURE_NO_WARNINGS
2  //[古老的密碼/Ancient Cipher](1/2)
3  #define IN "P08IN.txt"
4  #define OUT "P08OUT.txt"
5  //*****
6  #include <iostream>
7  #include <ctime>
8  using namespace std;
9  void redir(void);
10
11 //*****
12 /* Work Space*/
13 #include <algorithm>
14 #include <string>
15 //*****
16
17 int main(void) {
18     redir(); //redirection

```

```

49
50     freopen("CON", "r", stdin); //取消重新導向
51     freopen("CON", "w", stdout);
52     printf("Time used = %.2f\n", (double)clock() / CLK_TCK);
53     system("pause");
54     return 0; //the end...
55 }
56
57 void redir(void) {
58     freopen(IN, "r", stdin);
59     freopen(OUT, "w", stdout);
60 }
61 //*****

```

需要修改或自行撰寫的部分

- 初行加入crt secure no warnings
- 記得修改輸入和輸出txt檔案名稱
- #define為C++直接定義，變數型態會依照定義自動判定
- 額外include內建函式
#include <algorithm>，sort排序函式使用。
#include <string>，memset函式使用。

第八題

```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****

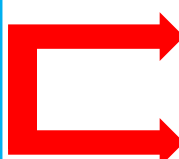
```

需要修改或自行撰寫的部分

➤ 解題思考：

- (1) 變數宣告
- (2) while迴圈輸入欲比對字串
- (3) 迴圈內初始化、統計、排序、比對、輸出

第八題



```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****
    
```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告


(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 22-24:

- ✓ 宣告 **字元陣列** stringA和stringB，大小為101(依題意)，100+1=101為終止字元。
- ✓ 宣告 **整數型態陣列** countA[26]和countB[26]做**A-Z字母**統計與排序使用。
- ✓ 宣告 **整數**變數len為存放**字串長度**。
- ✓ 宣告**布林**變數ans。

第八題



```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****
    
```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告


(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 26-27:

- ✓ while迴圈內使用cin輸入預比對字串stringA，cin輸入字串成功會回傳1(true)，cin輸入遇到EOF(檔案結尾)會轉換為0(false)。
- ✓ 確認stringA預比對字串能成功輸入後，再輸入比對字串stringB。

第八題



```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****
    
```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 28-29:

✓ void *memset(void *str, int c, size_t n)

✓ 從*str指向的位置填充整數型態c，填充的範圍為n。

✓ 將countA與countB填充0，填充範圍為sizeof(A)與sizeof(B)。

✓ <https://reurl.cc/ynbyx8>

Lib. Functions (1/2)

- `void *memset(void *ptr, int value, size_t num);`

- Defined in `<string>`

- Fill block of memory

- `memset`被定義於`<string>`

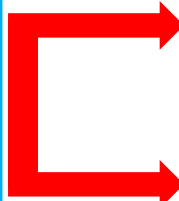
- 填充記憶體的block，設定被`ptr`指向的位置`num` bytes給予`value`值

- Set the first `num` bytes of the block pointed by `ptr` to the specified `value` (interpreted as an `unsigned char`)

- E.g. Clear (reset) the blocks of memory, `countA` and `countB`

```
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
```

第八題



```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****
    
```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 30-34:

- ✓ 使用 `strlen()` 統計 stringA 的字串長度並使用 int 轉換為整數型態變數存入 `len`。
- ✓ `for` 迴圈統計 countA 和 countB 中 A-Z 字母個別的數量。
- ✓ `[stringA[i] - 'A']` 中字元相減相當於 ASCII 碼相減。
- ✓ Ex: `countA['C' - 'A']++` → `countA[67 - 65]++` → `countA[2]++`
- ✓ 此區塊為統計

第八題

```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****

```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 35-38:

✓ sort為<algorithm>中用法，sort(起始位置, 結束位置)。

✓ 排序A[0]...A[25]、B[0]...B[25]。

✓ <https://reurl.cc/3NA10l>

✓ 氣泡排序法由大到小做排序。

✓ 此區塊為排序

Lib. Functions (2/2)

- `void sort(pointerA to the first element, pointerB to the last element);`

- Defined in `<algorithm>`
- Sort the elements in the range `[pointerA, pointerB)` in ascending order
- Overload operator `<`
- E.g.

- `sort`被定義於`<algorithm>`
- 遞增排序元素從小到大，範圍為`[pointerA, pointerB)`
- `pointerA`為第一個元素，`pointerB`為最後一個元素

```
37      sort(countA, countA+26); //sort A[0], A[1], ..., A[25]
38      sort(countB, countB+26); //sort B[0], B[1], ..., B[25]
```


第八題

```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****

```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 39-45:

✓ ans預設為true(YES)。

✓ for迴圈依序比對countA和countB字母數量，如果不相同則ans為false(NO)break跳出迴圈。

✓ 此區塊為比對

第八題

```

20 //*****
21 /* Work Space*/
22 char stringA[101], stringB[101]; //依題意
23 int countA[26], countB[26], len, i;
24 bool ans;
25
26 while (cin >> stringA) {
27     cin >> stringB;
28     memset(countA, 0, sizeof(countA));
29     memset(countB, 0, sizeof(countB));
30     len = int(strlen(stringA));
31     for (i = 0; i < len; i++) {
32         countA[stringA[i] - 'A']++;
33         countB[stringB[i] - 'A']++;
34     }
35     //sort A[0], A[1], ..., A[25]
36     sort(countA, countA + 26);
37     //sort B[0], B[1], ..., B[25]
38     sort(countB, countB + 26);
39     ans = true;
40     for (i = 0; i < 26; i++) {
41         if (countA[i] != countB[i]) {
42             ans = false;
43             break;
44         }
45     }
46     cout << (ans ? "YES" : "NO") << endl;
47 }
48 //*****

```

需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入欲比對字串

(3) 迴圈內初始化、統計、排序、比對、輸出

● Line 46:

✓ 條件運算子(?:)，如果ans為true則輸出"YES"，如果為false則輸出"NO"

✓ 此區塊為輸出

題目
解析

程式
詳解

輸出
結果

第八題

P08IN.txt - 記事本	P08OUT.txt - 記事本
檔案(F) 編輯(E) 格式(O)	檔案(F) 編輯(E) 格式(O)
JWPUDJSTVP	YES
VICTORIOUS	NO
MAMA	YES
ROME	YES
HAHA	NO
HEHE	
AAA	
AAA	
NEERCISTHEBEST	
SECRETMESSAGES	



Q&A

