

# 韌體實驗第9題

**Ting-Yu Lin, Adjunct Lecturer (林庭宇)**

**Office: AMOS Lab E517**

**Email: [tonylin0413@gmail.com](mailto:tonylin0413@gmail.com)**



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



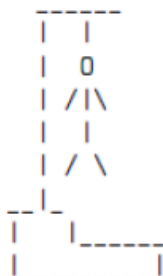
**BIDAS Technology  
GrounTruth Workshop**  
Computer Vision  
Image Processing  
Edge Computing

# 第九題

## 淡江大學電機工程學系 106 學年度軟體實驗題目

### 9. 【創子手遊戲/Hangman Judge】

遊戲規則：電腦想一個單字讓你猜，你每次可以猜一個字母。如果單字裡有那個字母，所有該字母會顯示出來；如果沒有那個字母，則電腦會在一幅「創子手」畫上填一筆。這幅畫一共需要 7 筆就能完成，因此你最多只能錯 6 次。注意，猜一個已經猜過的字母也算錯。



在本題中，你的任務是編寫一個「裁判」程式，輸入單字和玩家的猜測，判斷玩家贏了 (You win.)、輸了 (You lose.)、還是放棄了 (You chickened out.)。每組輸入資料包含 3 行，第一行是遊戲編號 (-1 為輸入結束標記)，第二行是電腦想的單字，第三行是玩家的猜測。後兩行保證只含小寫字母。

輸入：

```
1
cheese
chese
2
cheese
abcdefg
3
cheese
abcdefgij
-1
```

輸出 (註)：

```
Round 1
You win.
Round 2
You chickened out.
Round 3
You lose.
```



# 第九題

在本題中，你的任務是編寫一個「裁判」程式，輸入單字和玩家的猜測，判斷玩家贏了（You win.）、輸了（You lose.）、還是放棄了（You chickened out.）。每組輸入資料包含 3 行，第一行是遊戲編號（-1 為輸入結束標記），第二行是電腦想的單字，第三行是玩家的猜測。後兩行保證只含小寫字母。

輸入：

輸入回合 1  
電腦單字 cheese  
玩家猜測 chese  
2  
cheese  
abcdefg  
3  
cheese  
abcdefghij  
-1

輸出（註）：

Round 1  
You win.  
Round 2  
You chickened out.  
Round 3  
You lose.

- 猜過再猜也算錯，所以將猜過的令為空白字元。
- 將所有字元都猜對，**玩家贏了**。

次數	電腦單字	玩家猜測	結果
1	cheese	c	猜對
2	空heese	h	猜對
3	空空eese	e	猜對
4	空空空空s空	s	猜對

# 第九題

在本題中，你的任務是編寫一個「裁判」程式，輸入單字和玩家的猜測，判斷玩家贏了（You win.）、輸了（You lose.）、還是放棄了（You chickened out.）。每組輸入資料包含 3 行，第一行是遊戲編號（-1 為輸入結束標記），第二行是電腦想的單字，第三行是玩家的猜測。後兩行保證只含小寫字母。

輸入：

```
1
cheese
chese
3
cheese
abcdefgij
-1
```

輸入回合 2

電腦單字 cheese

玩家猜測 abcdefg

輸出（註）：

```
Round 1
You win.
Round 2
You chickened out.
Round 3
You lose.
```

- 猜過再猜也算錯，所以將猜過的令為空白字元。
- 錯7次才算輸，沒有繼續猜，所以**玩家放棄**了。

次數	電腦單字	玩家猜測	結果
1	cheese	a	醜1
2	cheese	b	醜2
3	cheese	c	醜3
4	cheese	d	醜4
5	cheese	e	猜對
6	ch空空s空	f	醜5
7	ch空空s空	g	醜6

# 第九題

在本題中，你的任務是編寫一個「裁判」程式，輸入單字和玩家的猜測，判斷玩家贏了（You win.）、輸了（You lose.）、還是放棄了（You chickened out.）。每組輸入資料包含 3 行，第一行是遊戲編號（-1 為輸入結束標記），第二行是電腦想的單字，第三行是玩家的猜測。後兩行保證只含小寫字母。

輸入：

1  
cheese  
chese  
2  
cheese  
abcdefg

輸入回合 3

電腦單字 cheese

玩家猜測 abcdefgij

-1

輸出（註）：

Round 1  
You win.  
Round 2  
You chickened out.  
Round 3  
You lose.

- 猜過再猜也算錯，所以將猜過的令為空白字元。
- 猜錯七次，**玩家輸了**。

次數	電腦單字	玩家猜測	結果
1	cheese	a	醜1
2	cheese	b	醜2
3	cheese	c	醜3
4	cheese	d	醜4
5	cheese	e	猜對
6	ch空空s空	f	醜5
7	ch空空s空	g	醜6
8	ch空空s空	i	醜7
9	ch空空s空	j	已輸

題目  
解析

程式  
詳解

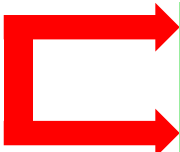
輸出  
結果

## 第九題

➤ 題目解析：

- (1)統計電腦出題與玩家猜測字串長度。
- (2)依玩家猜測字母順序比對，猜過再猜算猜錯。


# 第九題



```

1  #define _CRT_SECURE_NO_WARNINGS
2  //[創子手遊戲/Hangman Judge](1/2)
3  #define IN "P09IN.txt"
4  #define OUT "P09OUT.txt"
5  //*****
6  #include <iostream>
7  #include <ctime>
8  using namespace std;
9  void redir(void);
10
11 //*****
12 /* Work Space*/
13 #include <string>
14 //*****
15
16 int main(void) {
17     redir(); //redirection
18

```



```

59
60     freopen("CON", "r", stdin); //取消重新導向
61     freopen("CON", "w", stdout);
62     printf("Time used = %.2f\n", (double)clock() / CLK_TCK);
63     system("pause");
64     return 0; //the end...
65 }
66
67 void redir(void) {
68     freopen(IN, "r", stdin);
69     freopen(OUT, "w", stdout);
70 }

```

## 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 <string>，宣告及使用字串。

# 第九題

```

19  //*****
20  /* Work Space*/
21  int rnd; //不要用round命名,以免和函數round()混淆
22  string s1, s2;
23  int win, lose, left, chance, i, j, bad;
24
25  while (scanf("%d", &rnd) == 1 && rnd != -1) {
26      cin >> s1 >> s2;
27      win = lose = 0;
28      left = s1.length();
29      chance = 7;
30
31      for (i = 0; i < s2.length(); i++) {
32          bad = 1;
33          for (j = 0; j < s1.length(); j++) {
34              if (s1[j] == s2[i]) {
35                  left--;
36                  //猜一個已經猜過的字母也算錯
37                  s1[j] = ' ';
38                  bad = 0;
39              }
40          }
41          if (bad) chance--;
42          if (!chance) lose = 1;
43          if (!left) win = 1;
44          if (win || lose) break;
45      }

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束


(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘猜測字母數(left)、可猜錯的次數(chance)

(4) 判斷猜測結果

(5) 顯示結果



# 第九題



```

19  //*****
20  /* Work Space*/
21  int rnd; //不要用round命名,以免和函數round()混淆
22  string s1, s2;
23  int win, lose, left, chance, i, j, bad;
24
25  while (scanf("%d", &rnd) == 1 && rnd != -1) {
26      cin >> s1 >> s2;
27      win = lose = 0;
28      left = s1.length();
29      chance = 7;
30
31      for (i = 0; i < s2.length(); i++) {
32          bad = 1;
33          for (j = 0; j < s1.length(); j++) {
34              if (s1[j] == s2[i]) {
35                  left--;
36                  //猜一個已經猜過的字母也算錯
37                  s1[j] = ' ';
38                  bad = 0;
39              }
40          }
41          if (bad) chance--;
42          if (!chance) lose = 1;
43          if (!left) win = 1;
44          if (win || lose) break;
45      }

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束

(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘猜測字母數(left)、可猜錯的次數(chance)


(4) 判斷猜測結果

(5) 顯示結果

● Line 21-23:

- ✓ 宣告**整數**變數**rnd**(回合數)。
- ✓ 宣告**字串**變數**s1**(電腦字串)、**s2**(玩家字串)。
- ✓ 宣告**整數**變數**win**(猜贏標籤)、**lose**(猜輸標籤)、**left**(剩餘猜測字母數量)、**chance**(剩餘猜測次數)、**bad**(是否猜對字母標籤)。

# 第九題



```

19  //*****
20  /* Work Space*/
21  int rnd; //不要用round命名,以免和函數round()混淆
22  string s1, s2;
23  int win, lose, left, chance, i, j, bad;
24
25  while (scanf("%d", &rnd) == 1 && rnd != -1) {
26      cin >> s1 >> s2;
27      win = lose = 0;
28      left = s1.length();
29      chance = 7;
30
31      for (i = 0; i < s2.length(); i++) {
32          bad = 1;
33          for (j = 0; j < s1.length(); j++) {
34              if (s1[j] == s2[i]) {
35                  left--;
36                  //猜一個已經猜過的字母也算錯
37                  s1[j] = ' ';
38                  bad = 0;
39              }
40          }
41          if (bad) chance--;
42          if (!chance) lose = 1;
43          if (!left) win = 1;
44          if (win || lose) break;
45      }

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束

(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘猜測字母數(left)、可猜錯的次數(chance)

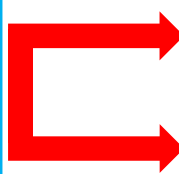
(4) 判斷猜測結果

(5) 顯示結果

● Line 25:

✓ 設定while迴圈，scanf回合數至變數rnd成立時會回傳1，迴圈條件設定為scanf成功 &&(邏輯運算子且) rnd不等於-1。

# 第九題



```

19  //*****
20  /* Work Space*/
21  int rnd; //不要用round命名,以免和函數round()混淆
22  string s1, s2;
23  int win, lose, left, chance, i, j, bad;
24
25  while (scanf("%d", &rnd) == 1 && rnd != -1) {
26      cin >> s1 >> s2;
27      win = lose = 0;
28      left = s1.length();
29      chance = 7;
30
31      for (i = 0; i < s2.length(); i++) {
32          bad = 1;
33          for (j = 0; j < s1.length(); j++) {
34              if (s1[j] == s2[i]) {
35                  left--;
36                  //猜一個已經猜過的字母也算錯
37                  s1[j] = ' ';
38                  bad = 0;
39              }
40          }
41          if (bad) chance--;
42          if (!chance) lose = 1;
43          if (!left) win = 1;
44          if (win || lose) break;
45      }

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束

(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘猜測字母數(left)、可猜錯的次數(chance)

(4) 判斷猜測結果

(5) 顯示結果

● Line 26-29:

- ✓ cin輸入電腦與玩家的字串。
- ✓ 設定初始輸(lose)和贏(win)的旗標(flag)皆為0。
- ✓ 使用s1.length()設定left為初始剩餘猜測字母數量。
- ✓ chance為可猜錯的次數(七次)。

# 第九題

```

19  //*****
20  /* Work Space*/
21  int rnd; //不要用round命名,以免和函數round()混淆
22  string s1, s2;
23  int win, lose, left, chance, i, j, bad;
24
25  while (scanf("%d", &rnd) == 1 && rnd != -1) {
26      cin >> s1 >> s2;
27      win = lose = 0;
28      left = s1.length();
29      chance = 7;
30
31      for (i = 0; i < s2.length(); i++) {
32          bad = 1;
33          for (j = 0; j < s1.length(); j++) {
34              if (s1[j] == s2[i]) {
35                  left--;
36                  //猜一個已經猜過的字母也算錯
37                  s1[j] = ' ';
38                  bad = 0;
39              }
40          }
41          if (bad) chance--;
42          if (!chance) lose = 1;
43          if (!left) win = 1;
44          if (win || lose) break;
45      }

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束

(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘猜測字母數(left)、可猜錯的次數(chance)

(4) 判斷猜測結果

(5) 顯示結果

● Line 31-40:

- ✓ for外層迴圈為**玩家猜測**，bad預設為1代表預設**玩家**這次猜的字母**猜錯**。
- ✓ for內層迴圈為使用if判斷電腦答案與玩家猜測的字母是否相同，如果相同則left--(將剩餘猜測字母數量-1)，令電腦答案的此字母為**空字元**，bad = 0表示**猜對**。



# 第九題

```

19  //*****
20  /* Work Space*/
21  int rnd; //不要用round命名,以免和函數round()混淆
22  string s1, s2;
23  int win, lose, left, chance, i, j, bad;
24
25  while (scanf("%d", &rnd) == 1 && rnd != -1) {
26      cin >> s1 >> s2;
27      win = lose = 0;
28      left = s1.length();
29      chance = 7;
30
31      for (i = 0; i < s2.length(); i++) {
32          bad = 1;
33          for (j = 0; j < s1.length(); j++) {
34              if (s1[j] == s2[i]) {
35                  left--;
36                  //猜一個已經猜過的字母也算錯
37                  s1[j] = ' ';
38                  bad = 0;
39              }
40          }
41          if (bad) chance--;
42          if (!chance) lose = 1;
43          if (!left) win = 1;
44          if (win || lose) break;
45      }

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束

(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘猜測字母數(left)、可猜錯的次數(chance)

(4) 判斷猜測結果

(5) 顯示結果

● Line 41-44:

✓ 判斷猜測結果

✓ 如果bad為1代表沒有猜對字母，if判斷式成立，所以chance--將可以猜錯的次數減1。

✓ 如果chance為0，if判斷式成立，則玩家輸了。

✓ 如果left為0，if判斷式成立，則玩家贏了。

✓ 如果win || lose (邏輯運算子或)為true則跳出迴圈，表示玩家贏或輸了13

# 第九題

```

47     printf("Round %d\n", rnd);
48     if (win) {
49         cout << "You win.\n";
50     }
51     else if (lose) {
52         cout << "You lose.\n";
53     }
54     else {
55         cout << "You chickened out.\n";
56     }
57 }
58 //*****

```

## 需要修改或自行撰寫的部分

➤ 解題思考：

(1) 變數宣告

(2) while迴圈輸入回合數，如果是-1為結束

(3) 輸入電腦與玩家字串、設定輸贏放棄標籤(flag)、剩餘  
猜測字母數(left)、可猜錯的次數(chance)

(4) 判斷猜測結果

(5) 顯示結果

● Line 47-56:

✓ 顯示第幾回合。

✓ 顯示猜測結果贏、輸或是放棄了。

題目  
解析

程式  
詳解

輸出  
結果

## 第九題

P09IN.txt - 記事本	P09OUT.txt - 記事本
檔案(F) 編輯(E) 格式(O)	檔案(F) 編輯(E) 格式(O)
1	Round 1
cheese	You win.
chese	Round 2
2	You chickened out.
cheese	Round 3
abcdefg	You lose.
3	
cheese	
abcdefghijkl	
-1	



# Q&A



