

Debug HW9

第一題

```
43 printf("您在TEST1.txt想尋找數量的字元為: ");
44 scanf("%c",&ch);
45 if((ch>=65 && ch<=90 )||(ch>=97 && ch<=122))
46 {
47     if(ch>=97) //將小寫視為大寫
48     {
49         ch=ch-32;
50     }
51     //else //wrong1
52     //{
53         num1=0; //wrong2 要初始化num1
54         for(i=0; i<1000; i++) //計算輸入字元出現次數
55         {
56             if(ch==data[i] || (ch+32)==data[i])
57             {
58                 num1++;
59             }
60         }
61     }
62     printf("該字元'%c'在該文件中數量為: %d\n\n",ch,num1); //wrong3 ch若原本為小寫，已被改為大寫
63 }
64 else
65 {
66     printf("Wrong input\n\n");
67     //num1=0; //wrong2
```

本區塊共有三個錯誤(1~3)

- Wrong1 計算的執行不能放在 else 中，否則若 ch 為小寫，將其在 if 中改成大寫後，就會跳過 else 中的計算。整體來說「判斷大小寫」和「計算個數」並非一組假如...否則...的關係，所以這邊的修正方式是將 else 去掉。
- Wrong2 重複計算，在新的一次計算開始前，要將計數的變數歸零，否則每次計算的結果會累加。修正方式是在 for 迴圈開始前將 num1 歸零。
- Wrong3 ch 因方便計算，已經一律被改為大寫，所以印出來的皆是大寫。更改方式是在更改前，便以另一個變數另存一次 ch 的初值，在印出數量時，引入的變數是這個存的變數。

第二題

```
76 //fp_out=fopen("output2.txt","w"); //wrong4 應該在回圈內開關
77 printf("每行的字元數: ");
78 scanf("%d",&num_in);
79 while(num_in!=-1) //判別最後斷行數
80 {
81     if(num_in>0){
82         num_out=num_in;
83         printf("更改完成!\n");
84     }
85     else
86     {
87         printf("輸入錯誤，在試一次\n");
88     }
89     /* printf("每行的字元數: ");
90     scanf("%d",&num_in); */ //wrong5 移到下面執行，否則會停止
91     fp_out=fopen("output2.txt","w"); //wrong6 因為是一個整體，所以應該全部括起來
92     i=0; //wrong4
93     while((data[i]=fgetc(fp_in))!=EOF) { //寫入output2.txt
94         fputc(data[i],fp_out); //wrong7 要將(data[i]=fgetc(fp_in))優先，再判斷是否EOF
95         i++;
96         if((i%num_out) == 0)
97         {
98             fprintf(fp_out,"\n");
99         }
100     }
```

本區塊共有四個錯誤(4~7)

- Wrong4 錯誤在於我們是要每輸入一次每行的字元數，就要立時更改，而每關一次檔案，更改

值才會存入檔案，故要將 `fopen` 和 `fclose` 放置於 `while` 迴圈中。

Wrong5 迴圈中的輸入值不能放置在中間，否則迴圈執行到一半就會被中斷。應更改為將下一次的輸入放置在迴圈中的最下面。

Wrong6 輸出「更改完成」、「輸入錯誤，再試一次」應和下面執行輸出並斷行至檔案的程式在一個迴圈下執行。

Wrong7 應該是判斷 `data[i]=fgetc(fp_in)` 是否有效，所以須將判斷的整體括起來，改成 `(data[i]=fgetc(fp_in))!=EOF` 才是有效的判斷式。

```
100 }  
101 fclose(fp_out); //wrong4  
102 rewind(fp_in); //wrong8 讀檔回到最開始  
103 printf("每行的字元數: ");  
104 scanf("%d",&num_in);  
105 } //wrong6 應全部括起來  
106  
107 printf("結束修改\n");  
108 fclose(fp_in);  
109 //fclose(fp_out); //wrong4 應在回圈內關閉檔案，才能及時儲存  
110  
111 return 0;  
112 }
```

Wrong8 因為必須重複讀檔，所以需要回到檔案的最開始，因此需要加上 `rewind` 函式。