

109 學年度第一學期 資訊科技概論 學習歷程認證項目

● 內容大要（可點選下方標題連結）：

一、國際運算思維挑戰賽 205/300 分(全國 PR82)	----- P. 1
二、英打成績 28 字/分鐘	----- P. 2
三、課堂程式練習	----- P. 3
四、期末上機考測驗	----- P. 5
五、本學期課堂省思	----- P. 6

一、Bebras 國際運算思維挑戰賽，獲得 205/300 分(全國 PR82)

說明：於 109 年 11 月中旬參加國際運算思維挑戰賽

省思：

在本學期的資訊科技概論課程中，我們學習了基礎程式語言。我覺得在這之中，「運算思維」也很重要。在許多時候並不是不會程式語法，而是邏輯觀念不正確，導致書寫時「語意錯誤」。這正是「運算思維」的重要性。

在進行這個測驗時，需要使用大量的邏輯思考——解開一個個複雜的題目。許多測驗並不是真的很困難，而是需要專心並且靜下心地想一遍。也能利用資訊科技概論課程中所學，更快速地解開問題。這對於正式寫程式時很有幫助，能讓我們更快了解應該如何將腦袋裡的想像化為平面的文字。

平常可以多加思考每個程式應有的邏輯順序，例如應該將某段程式擺在何處，程式才能正常運作，並在練習打程式時多方嘗試，不要害怕錯誤，可以讓自己對於邏輯思考更加精進，即使未來興趣與資訊無關依然受用！

[↩ 回目錄](#)

證明文件：



臺北市 市立內湖高中 許恩齊 同學 (學號：10900334)

參加 109學年度第一學期 國際運算思維挑戰賽

九、十年級 組 獲得 205/300 分 (全國 PR 82)


李忠諫教授
國立臺灣師範大學資訊工程學系

證書日期：2020/12/23
下載日期：2020/12/22

二、英打成績，成績為 28 字/分鐘

說明：以 typing.com 所提供之工具練習與測驗

省思：

剛開學，測驗打字初始值時，我覺得十分挫折，因為我只拿到 19WPM，但沒想到在期末再次測試，我竟然進步到 28WPM。在這之中，每次上課我也不厭其煩地練習打字，在一次次的練習中，不斷往前邁進。

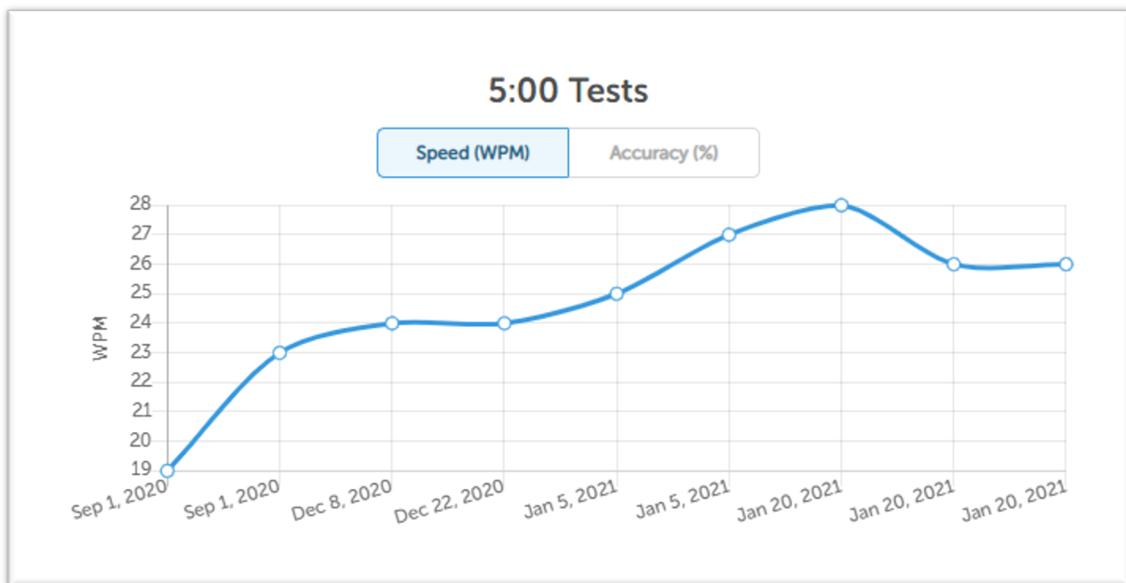
我覺得英打很重要，畢竟在現在競爭強烈的社會中，常常需要英文能力。況且現在也是資訊發展快速的時代，資訊能力必然重要，能在英打中有更快的速度，也許就更有優勢。

在英打中，也有些辦法可以加強自己的速度，例如更熟悉每個單字，讓自己可以背下它們；也可以更加強自己的語感，這樣就能夠對於文法更有把握，不需要一個個看單字；也可以多熟悉鍵盤上的配置，讓自己不需要不斷低頭看鍵盤，提升打字的效率。

在英打練習的過程，我發現「練習不會背叛你」的道理，做任何事情都能利用練習，提升自己。

[↩ 回目錄](#)

證明文件：



三、課堂程式練習

A. 標題：BMI 計算機

說明：國人時常飲食不正常，設計此計算機，要求使用者輸入身高、體重，就能計算出 BMI 以及相對應的身體狀況。利用變數、printf、scanf、if else。

設計過程：一開始的設計只是單純讓使用者依序輸入身高體重，但經過思考後決定改為系統判斷。第一版本（附圖 3-A-1）只會計算出 BMI；第二版本（附圖 3-A-2）會自動將小數取至第二位，並利用 if else 判斷 BMI 值，告訴使用者該數值之健康狀況為何。

[回目錄](#)

```

1 #include<stdio.h>
2
3 int main(){
4
5     float n1, n2, BMI;
6     printf("請用兩行輸入身高(公分)、體重(公斤), \n");
7     printf("不限先後順序! \n");
8     scanf("%f", &n1);
9     scanf("%f", &n2);
10    if (n1>n2){
11        BMI=n2/( n1*0.01* n1*0.01);
12    }
13    else {
14        BMI=n1/( n2*0.01* n2*0.01);
15    }
16    printf("我的BMI是%f", BMI );
17    return 0;
18 }

```

▲附圖 3-A-1

```

1 #include<stdio.h>
2
3 int main(){
4
5     float n1, n2, BMI;
6     printf("請用兩行輸入身高(公分)、體重(公斤), \n");
7     printf("不限先後順序! \n");
8     scanf("%f", &n1);
9     scanf("%f", &n2);
10    if (n1>n2){
11        BMI=n2/( n1*0.01* n1*0.01);
12    }
13    else {
14        BMI=n1/( n2*0.01* n2*0.01);
15    }
16    printf("我的BMI是%.2f\n", BMI );
17    if(BMI<18.5){
18        printf("體重過輕啦!");
19    }
20    else if(BMI<24){
21        printf("體重適中!");
22    }
23    else if(BMI<27){
24        printf("有點過重喔!");
25    }
26    else if(BMI<30){
27        printf("輕度肥胖啦!");
28    }
29    else if(BMI<35){
30        printf("中度肥胖啦!");
31    }
32    else{
33        printf("重度肥胖啦!");
34    }
35    return 0;
36 }
37

```

▲附圖 3-A-2

B. 標題：成績登記

說明：使用者可依序輸入多個成績，最後系統完整呈現並計算平均。利用陣列、for 迴圈、\t 水平定位。

設計過程：初學陣列時還不太了解，藉由這個程式更了解它。在寫這個程式的最後輸出時，我忽然想到課本上寫的\t 水平定位，就加入其中了，目的是讓使用者在實際使用時能更簡單明瞭。

[↩ 回目錄](#)

```

1 #include<stdio.h>
2
3 int main()
4 {
5     int G[7];
6     printf("請輸入六組成績：國文、英文、數學、歷史、地理、公民\n");
7     int i;
8     for(i=1;i<7;i++)
9     {
10         scanf("%d",&G[i]);
11     }
12     printf("國文\t英文\t數學\t歷史\t地理\t公民\n");
13     for(i=1;i<7;i++)
14     {
15         printf("%d\t",G[i]);
16     }
17     printf("\n平均%.2f", (G[1]+G[2]+G[3]+G[4]+G[5]+G[6])/6.0);
18     return 0;
19 }

```

▲附圖 3-B-1

C. 標題：終極密碼

說明：每回合系統自動選定一題目號碼，使用者每猜一次，系統就會顯示答案應該再大/小一點，直至猜中。利用 while 迴圈、if else。

設計過程：剛開始設計這個遊戲時特別手足無措，因為不曉得從何下手。但仔細拆解遊戲結構後，發現其實沒那麼困難。也一步步加入範圍提示、大小提示、隨機取數等（附圖 3-B-1）。其中，我建立了變數 l 及 r，分別存取當前範圍的最小值及最大值，讓系統可以依照使用者輸入之值改變範圍。

[↩ 回目錄](#)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main()
6 {
7     int q,y,l=1,r=100;
8     srand(time(NULL)); /*以目前時間抽取隨機數*/
9     q=(rand()%100)+1; /*取除100的餘數*/
10    printf("終極密碼開始囉~答案在1~100中！\n請輸入你的答案！\n");
11    scanf("%d",&y);
12    while (y!=q)
13    {
14        if (y<q)
15        {
16            l=y;
17            printf("不是啦~再大一點！\n答案在%d~%d\n再猜一次吧！\n",l,r);
18            scanf("%d",&y);
19        }
20        else
21        {
22            r=y;
23            printf("不是啦~再小一點！\n答案在%d~%d\n再猜一次吧！\n",l,r);
24            scanf("%d",&y);
25        }
26    }
27    printf("答對了~答案就是%d!",q);
28    return 0;
29 }

```

▲附圖 3-C-1

四、期末上機考測驗

A. 標題：簡易計算機（下圖左，附圖 4-A-1）

[↩回目錄](#)

總幻想著上數學課能使用計算機，現在趕快利用資訊課寫一個簡易計算機吧！
請設計程式以讀入2個整數，依序輸出a+b、a-b、a*b、a/b。

輸入格式
輸入僅一行，包含2個整數，兩數之間有一空白(2正整數皆小於30000)。

輸出格式
輸出共有四行，
第一行計算出 a 加 b；
第二行計算出 a 減 b；
第三行計算出 a 乘 b；
第四行計算出 a 除 b(取到小數第二位)。

提示
以scanf讀入兩數
以printf輸出四行(特別注意整數除法)

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int a,b;
7     scanf("%d %d",&a,&b);
8     printf("%d + %d = %d\n",a,b,a+b);
9     printf("%d - %d = %d\n",a,b,a-b);
10    printf("%d * %d = %d\n",a,b,a*b);
11    printf("%d / %d = %.2f\n",a,b,(float)a/b);
12    return 0;
13 }
14

```

▲附圖 4-A-1

每次發考卷老師都會唸出最高分的成績，請設計一個程式能幫老師計算平均、找到最高分及該生座號。

輸入格式
輸入1行，3個成績(各數之間有一空白)。

輸出格式
輸出3行，第一行為最高分之成績
第二行為最高分之座號
第三行為平均值(取到小數點第二位)

提示
1.可使用for或while迴圈條件設定好(執行3次)
2.可使用min存最高分數、max存最低分數、avg計算出平均值

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int s[3],min,max,i;
7     float avg;
8     scanf("%d %d %d",&s[0],&s[1],&s[2]);
9     max=s[0];
10    min=s[0];
11    avg=(float)(s[0]+s[1]+s[2])/3;
12    for(i=0;i<3;i++){
13        if(max<s[i]){
14            max=s[i];
15        }
16        if(min>s[i]){
17            min=s[i];
18        }
19    }
20    printf("最高分:%d分\n",max);
21    printf("最低分:%d分\n",min);
22    printf("平均值: %.2f分",avg);
23    return 0;
24 }

```

▲附圖 4-B-1

B. 標題：找出最大值（上圖右，附圖 4-B-1）

設計過程：這個題目當時對我有點棘手，我不知道該如何讓多個數字比較大小。

但我後來忽然想起以前上過的演算法——選擇排序法，我就利用其原理，一一比較每個數字。這個程式我利用了陣列、輸入輸出，並將迴圈與 if 結合，使過程更快速。希望未來能嘗試快速排序法。

[↩回目錄](#)

C. 標題：算出加總

設計過程：寫這個題目時我想到數學課所學，並且將之融入，更快的完成題目！

這個程式中除了用到基本的輸入、出，也使用了迴圈結構，讓程式更簡單。

[↩回目錄](#)

試著寫一個程式，能計算出n1~n2的加總。

問題描述
試著寫一個程式，能計算出n1~n2的加總。

輸入格式
輸入1行，為2個整數n1、n2(範圍±300)，且n1≤n2。

輸出格式
輸出1行，為n1~n2的總和

提示
1.可設三個變數分別為n1、n2、sum
2.可使用for或while迴圈，並將條件設定好每回合都將數值加進sum中

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int n1,n2,sum=0,a;
7     scanf("%d %d",&n1,&n2);
8     for(a=n1;a<n2+1;a++){
9         sum=sum+a;
10    }
11    printf("%d",sum);
12    return 0;
13 }
14

```

▲附圖 4-C-1

五、本學期課堂省思

在這學期的資訊科技概論課程中，從一開始的程式基本架構，到最後自己完成一個簡單的程式，實在收穫許多。

我覺得許多時候我們不應該只是封閉地念課本內容，我們亦應將之融入生活。這堂課就能充分展現這點。生活中也有許多地方可以用到我們學到的演算法、邏輯思考模式等，儘管未來興趣有可能與資訊牽扯不上，但也都對我們有很深的幫助。

這堂課絕對不僅讓我學到寫程式。在程式設計時，我們需要按部就班地思考，將頭腦複雜的想像化為簡單的文字。遇到問題可以先拆解，變成小問題，一一解決，最後整件事都能完成。這些思考的方式也能應用在危機處理、人際交流等多種日常活動上，故資訊課程絕對不單單是一堂普通的課，更是一堂讓我更能輕鬆應付生活狀況的課程。

[↩ 回目錄](#)