




自主學習計畫

Python 暨 APCS 進修

新竹女子高級中學 一年12班 19號 張旻嫻



計畫說明

自主學習計畫說明：

本次自主計畫主題為【Python暨APCS進修】，透過觀看網路上的線上課程，並且透過網路上的解題系統練習，以自學Python。

期望目標：

- 透過**APCS檢定檢測**學習成果：在2024年1月8號的檢定中取得**觀念及實作三級分**
- 每週完成特定課程
- 參加模擬測驗以評估進度



大學程式設計先修檢測成績證明

張時臻
新竹市國立新竹女中
准考證號：112029026

身分證號：0200737633
檢測日期：2023年06月04日

科目	原始總分	級別
程式設計觀念題	56	第三級
程式設計實作題	160	第三級

檢測成績級別說明

程式設計觀念題 檢測人數3043人			程式設計實作題 檢測人數3008人			
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*	說明
五	90~100	2.0	五	350~400	0.3	具備常見資料結構與基礎演算程序運用能力
四	70~89	32.0	四	250~349	2.5	具備程式設計與基礎資料結構運用能力
三	50~69	55.1	三	150~249	25.9	具備基礎程式設計與基礎資料結構運用能力
二	30~49	22.0	二	50~149	22.5	具備基礎程式設計能力
一	0~29	8.8	一	0~49	48.8	尚未具備基礎程式設計能力

* 該次檢測人數百分比（四捨五入取數到小數第一位）

Page: 1/1

申請日期：2023年06月21日

說明：2023 年 六月再次應試時成績

※同學可以視需要使用

中華民國 112 年 7 月 28 日

圖一 APCS觀念與實作三級成績證明(最下方有完整大圖)

學習動機：

在如今的大時代下，軟體可以與各領域跨領域結合，不論是再生醫、金融等等領域，程式設計都是不可或缺的技能。擁有程式設計這項跨域技能不僅可以提升自己邏輯思維和解決問題的能力，更可以提升自己的競爭力。還可以將工作自動化以提升效率，加上我對於程式語言也十分有興趣，所以選擇了這個主題。

學習方式概述：

每週觀看一堂線上課程，並在結束後以線上解題系統的題目檢測自己，透過一邊學習一邊實作的方式，累積經驗、鞏固所學、增加熟練度。

心得反思與檢討

自主學習之路：從計畫到實踐的挑戰與收穫

在自主學習計畫中，我最終達到實作及觀念三級的目標，雖然是在預設目標的下一場。而這次的自主學習計畫在縝密的安排下，也每週都有按著計劃持續進行。原本預期一月的考試當天早上因為身體抱恙，觀念題僅獲得一級的成績，再加上我自己花在觀念題的時間也比較少。在APCS的觀念題考試中是以C++語言作為題目，而我一直以來練習的皆為Python，這也讓我對C++感到生疏，因此導致一月考試的失利。

觀念題的部分是我忽略了，因此我在下一次考試前認真的研讀了C++，寫了考古題。這讓我發現程式語言都是能夠融會貫通的，只要瞭解基本架構學任何程式語言都能更快上手。

Debug的啟示：如何讓程式碼更清晰、更有效率

在學習的過程中我發現我常常需要花大把時間在debug，把時間花在了在茫茫程式碼中尋找錯誤，這也讓我發現我在思考及寫程式上不夠有條理。因此後來我會漸漸要求自己先把大問題分割成小問題，再針對每個小問題去寫程式。在撰寫每個小部分都應該先獨力測試過功能正常，最後在合併時才不會一下子出錯，卻找不到問題在哪。還有就是要善用註解功能

，隨著撰寫的程式越長，這項功能就越重要，這樣才能讓你看懂每部分在做甚麼。

並且我也發現我有時候在解決問題時常會繞一大圈，雖然最終都能達到一樣的效果，但對於電腦就需要更多的空間以及時間去跑我的程式。這方面能夠靠著未來去學習更多演算法，例如：DP、BFS 等去改進。

未來展望：從演算法優化到生活應用

在自主學習計畫完成後，我希望能夠去了解更多的程式語言，並且更重要的是我希望能夠將程式設計應用於實際生活中，例如：網頁設計。學習創建個人網站，將軟體能力應用於生活化的地方。預計參加學校在高三時開創的基本網站程式設計課程進一步學習。

寫程式不只是寫程式，還能寫出更有條理的人生！

透過本次自主學習，我不只精進了我的**程式語言能力**，也提升了我**解決問題的能力**。我能夠更有條理地去剖析問題，將複雜的問題拆解成小部分，並用有條理的方法解決。這項能力不僅能應用於程式設計中，也能應用於生活當中。

```

▶ #2次AC
#測資:
## 0 1 2 3 4 5 6 7 8 9
##list1=[4,7,2,9,6,0,8,1,5,3]
##list2=[0,2,1]
##follow=['T','T','T','T','T','T','T','T','T','T']
|
n=int(input())
list1=list(map(int, input().split()))
follow=[]
for a in range(n):
    follow.append("T")

#follow
def Follow(idx, follow):
    follow[idx]="F"
    return follow
#print(Follow(1, follow))
#plus
def plus(follow, list1, i, list0):
    if follow[i]=="T":
        list0.append(list1[i])
    return list0
#print(plus(follow, list1, 0))
#move
def Move(n, follow, list1):
    list0=[]
    x=0
    # !!! 如果有n筆資料就跑n次, 所以Runtime Error
    for i in range(n):
        if follow[i]=="T":
            while True:
                list0=plus(follow, list1, i, list0)
                #print(f"list0={list0}")
                follow=Follow(i, follow)
                #print(f"follow={follow}")
                i=list0[-1]
                if follow[i]=="F":
                    list0=[]
                    x+=1
                    break
            return x
#print(Move(10, follow, list1))
#print(Move(3, follow, list2))
print(Move(n, follow, list1), end="")

```

圖二 Practice08 程式碼

```

#2次AC
#題目要求最後一行無空白應注意!!!
#測資:
#list0=[[1, 2, 3], [4, 5, 6]]
#list0=[[1, 1], [3, 1], [1, 2]]
#list0=[[3, 3], [2, 1], [1, 2]]
#listt=[1, 0, 0]
#listt=[0, 1]
#listt=[1, 0]

#roll 翻轉
def roll(list0):
    list0.reverse()
    #print(f"list0={list0}")
    return list0

#flip 旋轉
def flip(list0):
    list_=[]
    for a in range(int(len(list0[0]))):
        list_.append([])
    for i in list0:
        #print(f"i={i}")
        n=0
        for t in i:
            #print(f"t={t}")
            list_[n].append(t)
            n+=1

    list_.reverse()
    return list_
#print(flip(list0))
#judge
def judge(listt, list0):
    for i in listt:
        if i==0:
            list0=flip(list0)
            #print(f"list0={list0}")

        elif i==1:
            list0=roll(list0)
            #print(f"list0={list0}")

    return list0

R, C, M=map(int, input().split())
list0=[]
for i in range(R):
    list1=list(map(int, input().split()))
    list0.append(list1)
#print(list0)
listt=list(map(int, input().split()))
listt.reverse()
#print(listt)

```

```

list0=judge(listt, list0)
#print(list0)
print(f' {len(list0)} {len(list0[0])}')
for p in list0:
    n=0
    for q in p:
        n+=1
        if n==len(list0[0]):
            print(q, end=" ")
        else:
            print(q, end=" ")
    print("")

```

圖三 Practice09 程式碼

國立新竹女中彈性學習時間學生自主學習成果表

2025/04/01 18:06

申請人	張綺嫻	班級/座號	1 年 12 班 19 號
申請學期	11101	申請時數	19
共學同學	無		
計畫名稱	Python 暨 APCS 進修		
學習類型	線上課程	對應學科屬性	資訊
設備需求	筆電		
指導教師			
自主學習內容概述	因為對程式語言頗感興趣，所以決定嘗試考檢定。過程採取一邊學習一邊累積經驗，及實際操作，增加熟練度。		
預期效益	今年 10/23 APCS 淺試一下 希望在明年 1/8 前考到 3 級		
與十二年國教核心素養之關聯	B2 科技資訊與媒體素養		
成果展示	不同意於校內學習平台提供自主學習成果與資料給其他同學參考		

週次	日期	自學內容	檢核進度	學習心得	自學場地
1	111/08/30(二)	基本 Python 複習 APCS 考試環境了解	完全達標	大概了解過 APCS 考試的模式，並學習基礎 Python。之後會在程式解題系統上找對應概念的題目練習。	
2	111/09/06(二)	APCS 應考重點-後測 Adv01-01. 成績指標 Adv01-02. 三角形辨別 Adv01-03. 線段覆蓋長度-小測資	完全達標	基本了解了 APCS 檢測應考時應該注意的重點，尤其最大的不同便是 APCS 是採用後測的機制，必須要一次做到對，沒有邊測邊 debug 的機會	
3	111/09/13(二)	IDE Adv02-01. 最大和 Adv02-02. 秘密差 Adv02-03. 邏輯運算子 (Logic Operators) Adv02-04. 籃球比賽	完全達標	這次耗時了較久才完成進度，尤其是秘密差這個題目卡了特別久，最後參考了網路上的解法之後，經過自我消化才完成了這一題。	
4	111/09/20(二)	pop()會造成 object 移位	完全達標	這次的練習題中使用了兩種寫法，也只	

		記憶體操作會拖累效能 Adv03-01. 定時 K 彈-小測資 Adv03-02. 定時 K 彈-大測資		有第二種寫法是用在大測資上，而第一種若套用大測資上也會造成超出時間限制的現象，者也讓我體會到不同寫法即便結果相同，也會造成極細微的差距。	
5	111/09/27(二)	觀念題 考試策略 觀念題難在哪	稍有落後	主要學習 APCS 在觀念題上主要考些甚麼，其主要是考 C 語言。雖說學會一個程式語言後再學習其他稱是語言可以比較容易的融會貫通，但畢竟 C++ 的規定比較多且複雜，解題及學習的過程也有特定的挑戰。	
6	111/10/04(二)	實作三級應考策略 Adv05-01. 魔王迷宮-第一子題(50 分) Adv05-02. 魔王迷宮-第二子題(50 分) 創造雙重迴圈 data 刪除一個 list 裡面特定 object List Comprehension 序列推導式	完全達標	考試著重策略，在沒有絕對把握下，不應該一開始就把整題做完，死科到底，應該逐步進行。在這次練習中亦學習到 python 許多不易發現的細節，只有在錯誤中不斷地去 debug、實踐，所學方能真正地被吸收。	
8	111/10/18(二)	Test-driven development "魔王迷宮"解題 "線段重疊問題"演算法 Multi-dimensional list sort Adv06-01. 線段覆蓋長度-大測資	完全達標	由上次的魔王迷宮提檢討，我重新梳理過自己寫程式的步驟，讓自己更有條理、系統性地完成，我也學會了拆分問題解決，大大地提升效率。靠自己的力量完成題目，過程雖然燒腦，但我很享受完成後帶來的成就感。	
9	111/10/25(二)	複習一下運算思維 切割 function	完全達標	學會了如何透過例子一步步推倒題	

		Adv07-01. 數字龍捲風		目，並且在推倒過程中構思所需的 function，再分段撰寫讓思緒及程式都能清晰地知道自己現在做的是甚麼，在這次的作業中也能感受出長進，明顯更加的快速有效率許多。	
10	111/11/01(二)	為什麼總是無法一次 AC 怎麼編周全的測資 Adv08-01. 小群體	完全達標	一次 AC 在檢測中十分重要，在答題過程中若是能先列出測資就能，大大減少一筆一筆重新輸入的麻煩，也能增加測試效率，因而提高正確率，像這次作業的考慮不周產生了 Runtime Error 也錯失一次 AC 的機會。	
11	111/11/08(二)	function 定義不佳以「定時 K 彈」為例，說明 function 定義修正 以「數字龍捲風」為例，說明 function 定義修正 Adv09-01. 矩陣轉換	完全達標	當遇到了更加困難的難題時，就須以更冷靜的心重新梳理，從所需到所求，列出核心 function 在一一撰寫，寫完一個測一個更容易發現錯誤，最後組裝時也更有條理不易出錯，比起一拿到題目就盲目地寫更加有效率。	
12	111/11/15(二)	題目誤解 I/O 加速-stdin.readline() I/O 加速-連續輸入直到 EOF I/O 加速-連續多組直到 EOF Adv10-01. 特殊編碼 Adv10-02. 動線安排	稍有落後	這次學到了 stdin 可以加速 input 和 output 的速度，雖然在一般做的小測資裡可能感覺不出太大的差別，但是放到測資鐘就能感受到明顯的差別，在寫特定的條件時也能比較有效率，可惜在 colab 中不適用	

13	111/11/22(二)	String basic String + String is iterable in / not in Check a String Adv11-01. 交錯字串 (Alternating Strings)	完全達標	string 在考題中也是常出現的，但有許多隱藏的小知識是我今天才學到的，可見程式這條路還長著呢，必須要時時更新，日日學習。	
14	111/11/29(二)	Adv12-01. 完全奇數 解題示範	完全達標	當自己解完題後再參考他人的寫法，也有效的提升了自己的能力，看看別人怎麼寫能讓程式執行更快速、更有邏輯、更簡便。	
15	111/12/06(二)	淺拷貝 深拷貝 function 內操作 list-無效 function 內操作 list-有效 Adv13-01. 人口遷移	完全達標	身拷貝與淺拷貝的差異在於身拷貝將元素一一抽出來拷貝，因此在更動的時候只會改變一方，不同於淺拷貝兩方都會更動，這就像以前學到若是同時產生多個 list 會遇到的問題一樣。	
16	111/12/13(二)	109 年七月考古題 Adv14-01. 購物車 Adv14-02. 骰子	完全達標	現在寫程式與以前相比，正確率已經大大提升了，但是在速度方面仍然需要加強，主要原因可能出在練習不足，打字也不快，或是仍然有些錯誤需要大把時間來 debug，這些都需要多加練習來訓練。	
17	111/12/20(二)	美麗的彩帶 dictionary 邊界差異法 Adv15-01. 美麗的彩帶	完全達標	很多時候只會看到自己第一直覺地解法，然而有時候這樣雖然寫起來比較直觀易懂，但是也讓程式繞了許多個彎，這些就要靠每次解題	

				後看看自己與別人解題的邏輯思維差異，並且學習較有效的辦法。	
18	111/12/27(二)	<p>enumerate</p> <p>110 年一月考古題</p> <p>Adv16-01. 購買力</p> <p>Adv16-02. 流量-第一子題</p> <p>Adv16-03. 流量-第二子題</p>	完全達標	又學習了一個新的、更高效、更快速、更簡便的 enumerate，雖然可能比起原本的寫法，沒那麼直觀，但面對更複雜的題目時就可以應用它，會變得簡便許多。	
19	112/01/03(二)	<p>111 年一月考古題</p> <p>Adv17-01. 程式交易</p> <p>Adv17-02. 贏家預測-第一子題</p> <p>Adv17-03. 贏家預測-第二子題</p>	完全達標	第一題相對第二題會簡單一些，因此當我們直接看到第二題卻沒有想法時，可以透過先完成第一子提的要求，在去擴大的功能以符合第二子題所求，便會簡單許多，畢竟萬丈高樓也是一層層疊起的。	
20	112/01/10(二)	APCS 觀念題模擬考	稍有落後	這是我的最後一堂自主學習了，在接下來便要真正面對考場了，雖然經過練習，我早已與以前相比進步了些，但面對正式來臨時，仍然有點惶惶不安，希望已經經過這麼多次練習也多了幾分把握，正式考時能冷靜面對。	

成果說明：文字

在這次的自主學習中，我學到了很多，最終也達到想要的目標。在十月的那一場檢測中我拿到了實作及觀念二級的成績，這也讓我能夠更明白我自己的不足。隔年一月，我的實作拿到了三級，但當天早上因為身體抱恙等緣故，拿了一級，我也明白在觀念題方面我比實作題花的心力更少，得到這樣的結果提醒著我更加努力，因此我在六月時再試了一次那次的考試，終於達到我所望實作及觀念都達到三級。其實我做這個自主學習不光是為了考試，更多的是我喜歡這個過程，我享受每一次寫出來的程式解決問題的成就感，debug 成功的快感，還有看著自己一路走來成長的滿足，從一開始的程式雜亂無章，每有邏輯想到甚麼寫甚麼，也不明白為什麼要有註解功能，到後來分段寫程式，許多註解標明用意，註記錯誤，錯誤變少，效率變高，這些都帶果無限的滿足與快樂。但我也明白我自己的實力仍然只在三級初，想要更往上爬，還需付出更多的心力學習，程式之路漫漫，路上也人才濟濟。我也相信自己的實力，不僅限於此，未來定要更加努力才行。



大學程式設計先修檢測成績證明

張綺嫻
新竹市國立新竹女中
准考證號：112012401

身分證號：0200737633
檢測日期：2023年01月08日

科目	原始總分	級別
程式設計觀念題	20	第一級
程式設計實作題	185	第三級

檢測成績級別說明

程式設計觀念題 檢測人數2636人			程式設計實作題 檢測人數2622人			
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*	說明
五	90~100	0.0	五	350~400	0.4	具備常見資料結構與基礎演算程序運用能力
四	70~89	4.6	四	250~349	3.2	具備程式設計與基礎資料結構運用能力
三	50~69	26.4	三	150~249	32.2	具備基礎程式設計與基礎資料結構運用能力
二	30~49	49.8	二	50~149	26.6	具備基礎程式設計能力
一	0~29	19.1	一	0~49	37.5	尚未具備基礎程式設計能力

* 該次檢測人數百分比（四捨五入取概數到小數第一位）



大學程式設計先修檢測成績證明

張旂嫻

新竹市國立新竹女中

准考證號：112029026

身分證號：0200737633

檢測日期：2023年06月04日

科目	原始總分	級別
程式設計觀念題	56	第三級
程式設計實作題	160	第三級

檢測成績級別說明

程式設計觀念題 檢測人數3043人			程式設計實作題 檢測人數3008人			
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*	說明
五	90~100	2.0	五	350~400	0.3	具備常見資料結構與基礎演算程序運用能力
四	70~89	32.0	四	250~349	2.5	具備程式設計與基礎資料結構運用能力
三	50~69	35.1	三	150~249	25.9	具備基礎程式設計與基礎資料結構運用能力
二	30~49	22.0	二	50~149	22.5	具備基礎程式設計能力
一	0~29	8.8	一	0~49	48.8	尚未具備基礎程式設計能力

* 該次檢測人數百分比（四捨五入取概數到小數第一位）

Page : 1/1

申請日期：2023年06月21日

說明：2023 年 六月再次應試時成績

※同學可以視需要使用

中 華 民 國 112 年 7 月 28 日