學習報告

APCS 大學程式設計先修檢測

在這個 APCS 測驗中，我取得了 6 分（觀念題 3 分、實作題 3 分）。這對不會演算法與對 C 語言不熟悉的我而言，其實已經是不錯的成績了，但在這次考試後，我才意識到演算法究竟有多重要，也讓我認知道了還有許多東西是我需要學習的。

大學程式設計先修檢測成績證明

蔡淳宇
臺北市市立陽明高中
准考證號：114019652

身分證號：[redacted]
檢測日期：2025年01月05日

科目	原始總分	級別
程式設計觀念題	68	第三級
程式設計實作題	205	第三級

檢測成績級別說明

程式設計觀念題 檢測人數2729人			程式設計實作題 檢測人數2739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	5.6	五	350～400	0.1	具備常見資料結構與基礎演算程序運用能力
四	70～89	34.3	四	250～349	1.9	具備程式設計與基礎資料結構運用能力
三	50～69	32.6	三	150～249	43.6	具備基礎程式設計與基礎資料結構運用能力
二	30～49	19.4	二	50～149	27.6	具備基礎程式設計能力
一	0～29	8.1	一	0～49	26.9	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：2025年08月05日

學習報告

APCS 大學程式設計先修檢測
參加動機和訓練過程

我自高中自學程式後，就被學校同學與老師推薦去參加了 APCS，而我之所以會參加，也是想藉著這樣一個免費的測驗，看看自己的程式能力到底如何。我第一次參加時，程式能力並不是很好，因此實作題一題都沒解出來，只拿到了 1 級分。經過了這一次的經驗，我開始到 ZeroJudge 等平台上練習實作題的部分，並去網路上找一些觀念題的教學。經過一段時間的學習，我再度參加了 APCS 測驗，並取得了總分 6 級分的成績。雖然已取得明顯進步，但我仍然認為自己還有許多進步空間，我希望未來我可以繼續學習如演算法和資料結構等程式技術，努力將自己的實力提升到能達到實作 5 級分的程度。

▲ ZeroJudge 練習紀錄

學習報告

APCS 大學程式設計先修檢測 解題語言與實作例題

由於我從高中開始就專注在學習 Python 了，因此我選擇使用 Python 作為我寫實作題時用的語言，Python 作為目前全球最紅的程式語言，雖然執行效率不如傳統編譯型語言來得高，但它以其簡單、美觀與強大的功能而聞名，它也是目前多數人在開發和研究 AI 時會用的主要語言。我之所以選擇它來解題，也是因為它對於字串控制語法與部分內建函式庫都較 C/C++ 等語言方便。

以下是一個程式的題目範例與解答，我先將輸入的算式全部翻轉過來之後，用一個迴圈迭代所有算式，由於 Python 對遞迴效率不佳，我使用 stack 來存放與處理未計算的數值，以此處理不同函數的運算處理及計算優先順序。

內容

給一個運算式，運算式的內容由數字、＋、＊和 某個函式 $f()$ 所組成，除了函式 $f()$ 以外不會有額外的括號。請將此運算式依照 **先加後乘** 的方式運算。函式 $f(x_1, x_2, x_3, x_4, \dots)$ 定義為從這個不定長度的參數 $x_1, x_2, x_3, x_4, \dots$ 中的最大值扣掉最小值。例如 $f(3,6,2) = 6 - 2 = 4$、$f(3) = 0$。

▲ ZeroJudge j607.（取自 2023 年 1 月 APCS 實作第三題）

```python
def high(i): #最高層的運算：(*) > (+)。"處理乘法鏈"
    v, i = mid(i)
    while i < len(inp) and inp[i] == '*':
        n, i = mid(i + 1)
        v *= n
    return (v, i)

def mid(i): #加法層級運算：(+) 比 (*) 晚處理。"處理加法鏈"
    v, i = low(i)
    while i < len(inp) and inp[i] == '+':
        n, i = low(i + 1)
        v += n
    return (v, i)

def low(i): #最基礎的運算：數字直接回傳、f(...) 進括號並 call high()。"處理最小單位"
    if inp[i].isdigit():
        n = 0
        while i < len(inp) and inp[i].isdigit():
            n = n * 10 + int(inp[i])
            i += 1
        return (n, i)
    elif inp[i] == 'f':
        i += 2 #跳過 "f("
        args = []
        while True:
            v, i = high(i)
            args.append(v)
            if inp[i] == ',':
                i += 1
            else:
                break
        i += 1 #跳過 ")"
        return (max(args)-min(args), i)
    else:
        raise ValueError("Unexpected character: " + inp[i])

inp = input().strip()
ans, _ = high(0)
print(ans)
```

▲ 我的 Python 解法（含註釋）