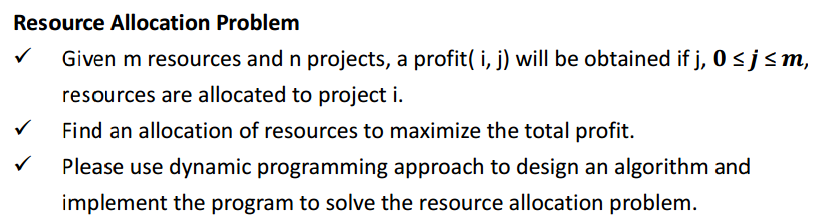
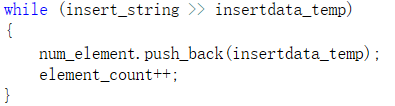
# Report

**實作目標**

**設計過程**

1. 環境：VS2015
2. 讀取檔案操作：

因為沒有規定profit table的大小，還有測資的數量。因此我通過判斷讀取的檔案的所含的數字的數量來判斷是在輸入profit table還是在輸入測資：

上面的程式碼我將當前讀入的行中包括的每個數字讀入一個vector中，同時用element\_count記錄這行有多少個數字。

根據element\_count的大小判斷接下來要進行什麼操作：

If（element\_count>1）

{

進行profit table的構建。

}

If（element\_count==1）

{

進行resource allocation的計算。

}

If（element\_count==0）

{

什麼也不做

}

1. Resource allocation：

Resource allocation可以利用dynamic programming的方法：

1. 進行如下假設：

Resource總數為7

階段：k=1,2,3,4 //表示考試科目的數量

決策變數：xk（k=1,2,3,4） //表示分配到每個階段k的resource

狀態變數：sk //剩餘未分配的resource

效果變數:Pi（xi） //分配x個resource給階段i產生的profit

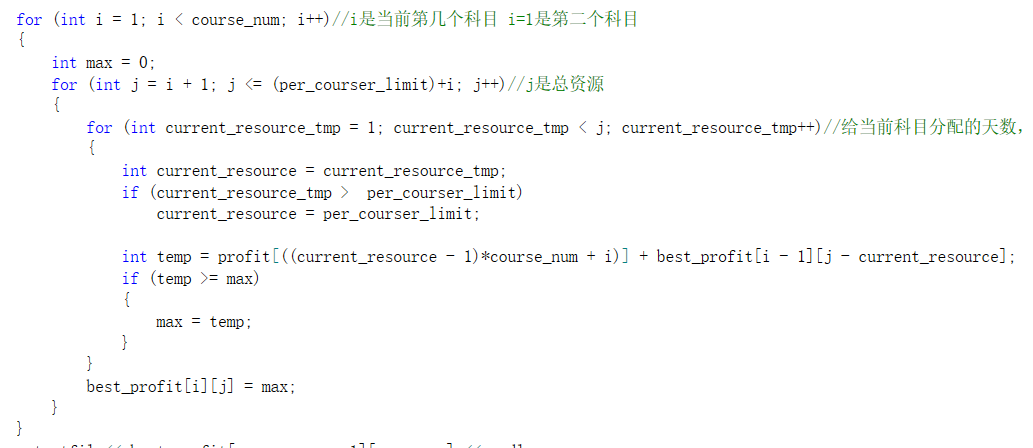
目標：在x1+x2+x3+x4=7, 且x1, x2, x3, x4>=1的條件下，選擇x1, x2, x3, x4達到

max[P­1(X1)+P2(X2)+P3(X3)+P4(X4)]的效果。

1. 分析：

每一個決策的最佳方案可以建立在更小的決策的最佳方案的基礎上得到。

例如說resource為7，courses為4，現在已知在courses為3的情況下的最佳allocation，則為了得到增加了一個course之後的最佳allocation，只需要判斷分配給這多出來的course的resource分別為1,2,3,4並且把剩餘的resource給其它3個course的狀況下哪一個是最佳方案即可。

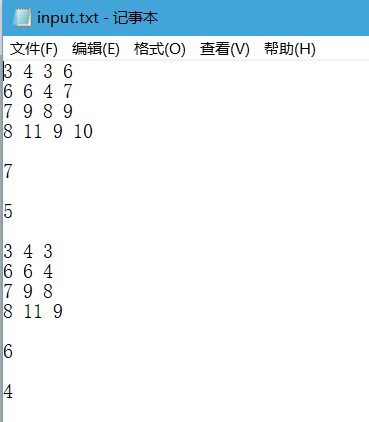
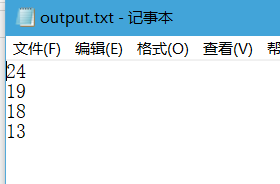


判斷剩餘resource給剩餘course哪個是最佳方案可以通過建立一個table進行查詢操作來得到解決：

我建立了一個二维数组，best\_profit，用來存儲最佳allocation，當得到一個新的最佳allocation的時候加入這個二维数组：

best\_profit[i][j] = max;

**實驗結果**

**遇到的問題及解決方法：**

**問題：**

1. Vector出現index overflow。
2. 計算結果不正確。

**解決辦法：**

1. 對profit table進行查詢的時候要注意不要超過index限制，為了通用性把具體的resource數還有course數分別用一個變數表示：

profit[((current\_resource - 1)\*course\_num + i)]

int resource = num\_element[0];

int course\_num;

int per\_courser\_limit = resource - course\_num + 1;//每門科目最多分配的resource

1. 檢查程式碼是否存在邏輯漏洞。