

作業研究期中專題

管理科學系 仲音綺



109700011

目錄

1. LP Model 解釋
2. Code 解釋
3. Excel 解釋
   * 每月profit分析
   * 產量分析
     + - 同種商品在不同工廠的產量分析
       - 不同種商品在相同工廠的產量分析
   * 銷量分析
     + - 每種商品在各個月份的銷量分析 (附銷量最好的產品)
       - 原因可能性分析
   * Backorder 分析
   * 存貨分析
4. 總結
   * 遇到的問題及解決方式
   * 學習到的辦法（Debug）

|  |  |
| --- | --- |
| **Sets** | |
|  | The set of factories, |
|  | The set of products, |
|  | The set of months during the planning horizon, |
| **Parameters** | |
|  | The installed capacity for factory , |
|  | Per unit sales price of product , |
|  | The holding cost per unit of product per month, |
|  | The backlog cost per unit of product per month, |
|  | The production cost per unit of product , |
|  | The demand of product in month , |
| **Decision Variables** | |
|  | The production quantity for project *j* using factory *i* in month *t*, |
|  | The sales quantity of product in month , |
|  | The backlog quantity of product in month , |
|  | The inventory of product in month, |

**Linear Programming Model:**

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  |  | (1) |
|  |  | (2) |
|  |  | (3) |
|  |  | (4)  (5) |
|  |  | (6) |
|  |  | (7) |
|  |  | (8) |

**Explanations of the LP model:**

For example, constraint (2) ensures that the total production quantity cannot exceed the installed capacity at each factory.

1. 銷貨收入 – 生產成本 – 存貨成本 – backorder成本 = 最大化利潤
2. 限制式解釋
   1. 設定好當月每個產品該有的存貨量
3. 限制產量不能超過每個工廠的最大產能
4. 以*t’*為當月，從*t’*以前包含當月的總銷售量要小於總需求量，因為最多就是賣需求量
5. 限制當月銷售量最大極限，不能超過前期存貨跟累積在所有工廠的產量
6. 設定當月backorder應該要有的量，以*t’*為當月，當月backorder的量要等於，從*t’*以前包含當月的總需求量減總銷售量。
7. 第一個月的存貨量為0
8. 第一個月的存貨量為0
9. 產量、銷售量、backorder、存貨量皆為0

EXCEL 分析

1. Profit 分析

除第一月份的profit相對較高以外，其餘都比較平均，我認為這有可能是因為第一個月份沒有backorder成本的緣故，至於存貨成本，由之後的分析可得知，只有三月份會有存貨成本，並且之於其他成本來說，可以說是少之又少，所以我認為是這backorder占較大的因素。

1. 產量分析
   * **同種商品在不同工廠的產量分析**

C工廠為：941467

D工廠為：23193

E工廠為：233104

A工廠為：1077924

D工廠為：28763

E工廠為：245332

A工廠為：626

D工廠為：92

E工廠為：563

A工廠為：230

D工廠為：0

E工廠為：869

比較特別的是GAN在D工廠完全沒有生產。

A工廠為：673626

D工廠為：160589

E工廠為：166629

B工廠為：22819

D工廠為：5136

F工廠為：5217

B工廠為：22512

D工廠為：8204

F工廠為：19299

D工廠為：187345

F工廠為：209585

D工廠為：8326

F工廠為：9568

D工廠為：10192

F工廠為：10156

C工廠為：17564

D工廠為：1391

* + **不同商品在同一工廠的產量比例**

A工廠很明顯地基本上都是在製造DIS，其餘產品基本上不代工，並且量非常之少，由此可以推測其於兩樣商品有可能是臨時接的訂單，以少量去賣。

B工廠以生產HIV，可看出HIV為其的主要產品，其餘兩個產品也是穩定生產中，由此可推測此工廠叫不會接臨時訂單。

C工廠可看出其主要產品為BCD，SEN有可能也只是臨時接單所造成。

D工廠為全能型工廠，每個產品都能生產，但還是可以看出產品是以LOG為主，接著是MIX及HIV，其餘的產品大約都是少量生產。我的推測是D工廠是可能有所製造這些產品的所有機器及人力，但是除LOG、MIX、HIV以外，其餘產品上述資源的有限，所以才不能大量生產。

E工廠主要是生產BCD及DIS兩樣產品，GAD及GAN的量少之又少，但值得一提的是GAN在第二年位於E工廠的產量為0，我猜測E工廠生產GAN的CP值應該極低，才會導致這個情況。

F工廠的型態我認為跟D工廠的型態較相近，都是屬於有多種類但卻少量的機器及相關知識背景的人力。

1. **銷量分析**

由上圖可知，BCD的總體銷量呈現一個斜率為負的趨勢，在七月分及隔年五月分別有銷量的最高點和最低點，這有可能是其商品對於時間敏感度較高，可能隨著時間有性質相同卻更優良的產品出現，所以才導致其銷量逐漸下降。

此種類的商品最為暢銷的是BCD00050，總銷量為1646。

由上圖可知，DIS的每月銷良呈現正斜率的趨勢，分別在七月及隔年五月有銷量最低點和最高點，以上兩點情況恰恰跟BCD相反，所以我認為DIS跟BCD互為競爭性產品。

此種類的商品最為暢銷的是DIS01128，總銷量為1608。

由上圖可知，GAD每月銷量呈現正斜率的趨勢，分別在十一月及十二月有銷量最低點及最高點，而每個月的銷量極度不穩定，我認為是因為GAD種類的商品只有一個，所以對於賣家而言，只有賣或不賣，並沒有同種類的其他產品可以分擔風險；再者，根據上圖，我們很難看到銷量有呈現連續成長或連續下降，這個我推測對於這個商品來說，客戶群可能非常窄小並且產品的使用期限可能超過一個月，對於這些買家，前幾個月買過，那這個月就不會在買。

由上圖可知，GAN的每月銷量趨勢呈現水平的狀態，分別在五月及隔年九月有最低點和最高點，這種銷量趨勢我推估其客戶群應該範圍不大，並且此產品使用期限可能以月的正負幾天為單位，才會呈現如此平穩的狀態。

此種類的商品最為暢銷的是GAN01128，總銷量為1281。

由上圖可知，HV的每月銷量呈現正斜率的趨勢，隔年六月及七月分別為銷量的最低點及最高點。

此種類的商品最為暢銷的是HV00050，總銷量為2442。

由上圖可知，GAN的每月銷量趨勢呈現水平的狀態，分別隔年五月及十月有最高點和最低點，這種銷量趨勢我推估其客戶群應該範圍不大，並且此產品使用期限可能以月的正負幾天為單位，才會呈現如此平穩的狀態。值得注意的是，可能因為氣候因素，每年11月都會有一個local minimum，所以我推測此種產品可能不是當下寒冷的天氣。

此種類的商品最為暢銷的是LOG02990，總銷量為1620。

由上圖可知，MEM每月銷量呈現相對平緩的負斜率關係，隔年四月及五月分別為銷量最高點及最低點。

此種類的商品最為暢銷的是MEM03024，總銷量為1522。

由上圖可知，MIX每月銷量呈現負斜率的趨勢，並且每月銷量漲跌幅極大，其原因可能是客群不穩定，抑或產品時效性低，有些買家可能會選擇囤貨。

此種類的商品最為暢銷的是MIX03033，總銷量為1641。

由上圖可知，NVM的每月銷量呈現負斜率的趨勢，我認為此商品面臨與BCD一樣的困境，有其他競爭性商品爭奪其客戶群，才會導致銷量逐月降低並且預估會持續呈現負成長的關係。

此種類的商品最為暢銷的是NVM03371，總銷量為1329。

由上圖可知，ROM每月銷量呈現平穩的正斜率趨勢，其在七月分及隔年二月份分別為銷量最低點及最高點。

此種類的商品最為暢銷的是ROM03388，總銷量為1489。

由上圖可知，SEN的每月銷量基本上是呈現水平狀態，但是有小幅下降的趨勢，其在七月及隔年九月有銷量最高點和最低點。

此種類的商品最為暢銷的是SEN03406，總銷量為1475。

我們可以看出，七月分是許多商品銷售最大值或最小值，這或許是因為這些商品其實存在不完全互補品的可能。

1. **Backorder分析**

**BCD、GAN、GAD、HV、LOG、MIX、SEN的每月backorder皆為0。**

**由以上四張backorder分析折線圖可知，大部分的backorder都是從5-7月開始，這跟前面銷量分析的結果一樣，可以很合理推測5-7是這些產品開始是和使用的時期，所以需求變多，backorder隨之增加。**

1. **存貨分析**

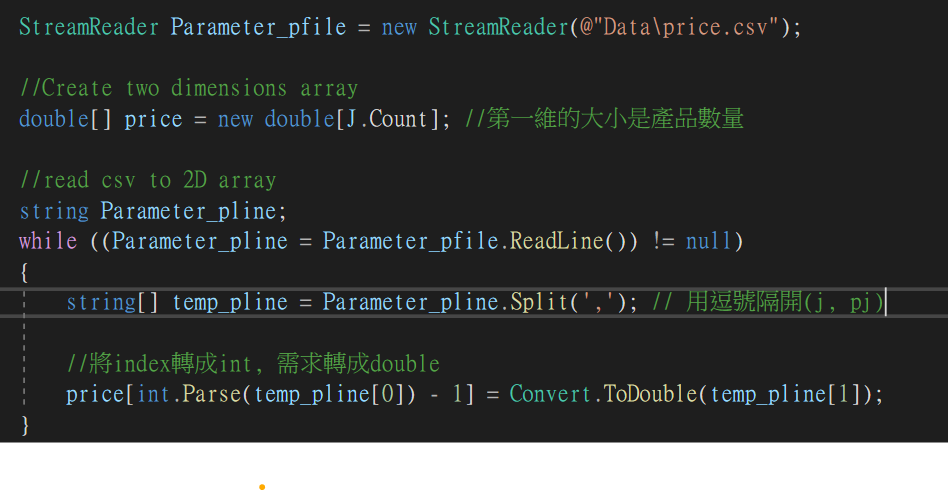
**只有第三個月有存貨成本，分別是：**

**BCD00888有59個存貨、DIS01478有33個存貨、DIS02128有55個存貨、HV02273有56個存貨。**

**造成會有存貨的原因有可能是製造簡易，又或是棄單之類的可能性。**

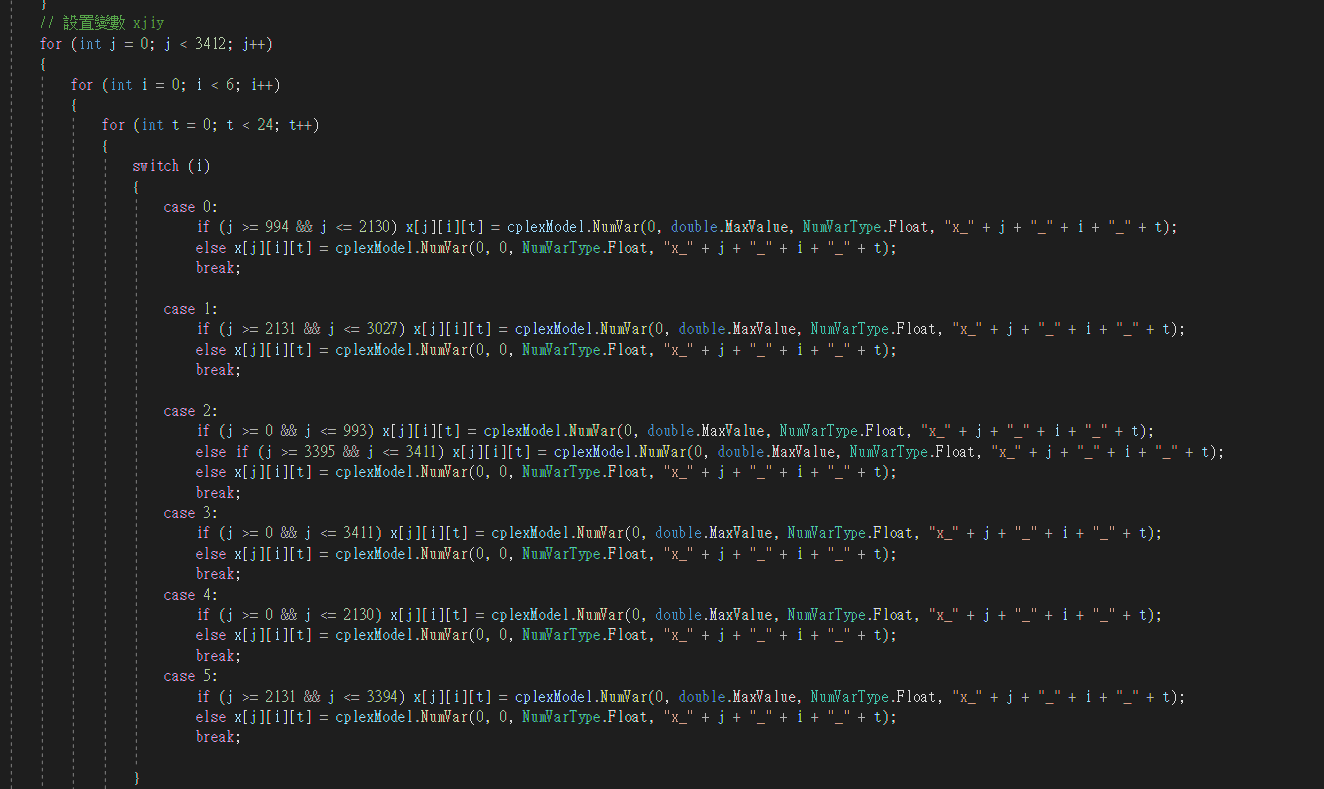
Code 分析

首先，我先輸入csv檔：



上圖是我輸入每種產品的價錢，其跟demand 不同的是，demand 是二維，而price只有一維而已，所以在輸入的時候，只需要在維度那邊輸入進產品數量就好，無需管時間。

接著，設定變數：



上圖為設定產量變數（xjit）：

Case1 ：A工廠生產（i = 0）只能生產產品編號尾數995~2131（DIS, GAD, GAN）的商品

Case2 ：B工廠生產（i = 1）只能生產產品編號尾數2132~3028（HV, LOG, MEM）的商品

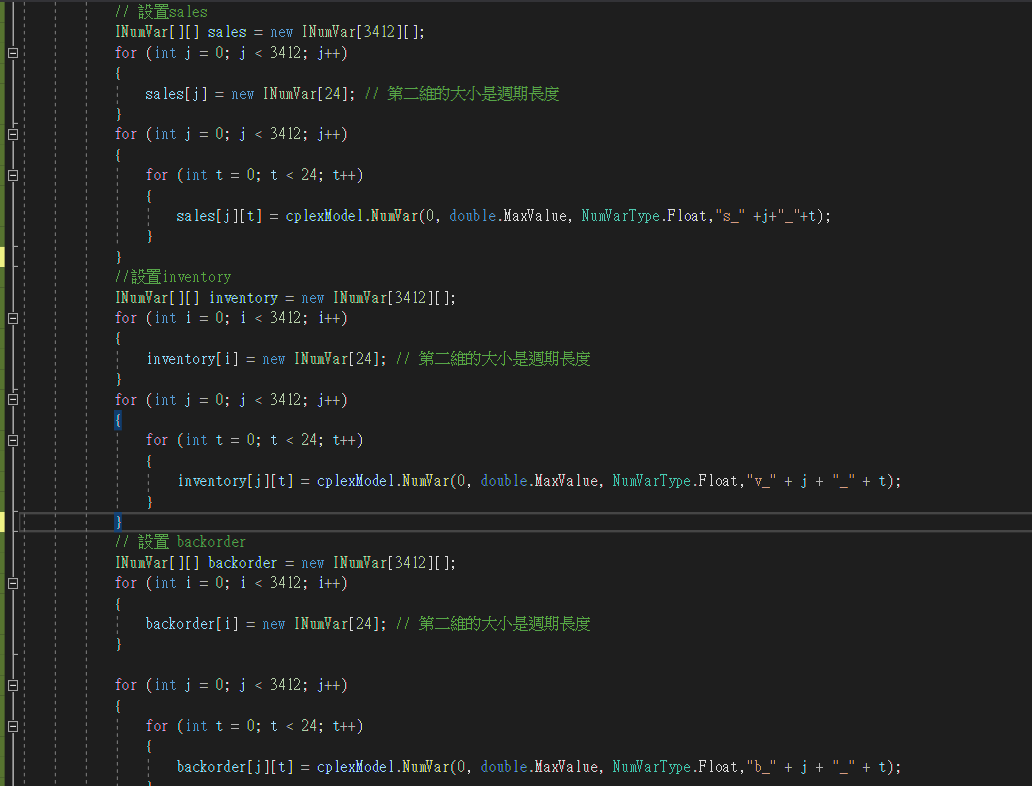
Case3 ：C工廠生產（i = 2）只能生產產品編號尾數1~994（BCD）及3396~3412（SEN）的商品

Case4 ：D工廠生產（i = 3）能生產所有種類的商品

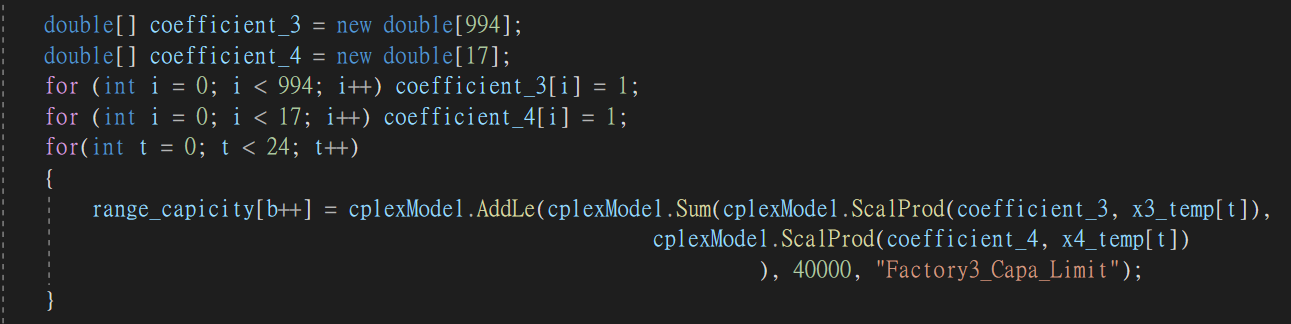
Case5 ：E工廠生產（i = 4）只能生產產品編號尾數1~2131（BCD, DIS, GAD, GAN）的商品

Case6 ：F工廠生產（i = 5）只能生產產品編號尾數2132~3395（HV, LOG, MEM, MIX, NVM, ROM, SEN）的商品

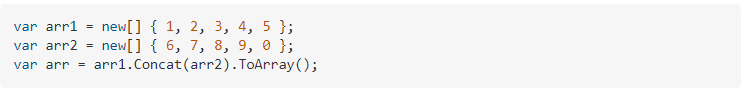
以及 sales, inventory, backorder的變數，NumVar後面的的頭兩個數是在設定變數的上界下界。



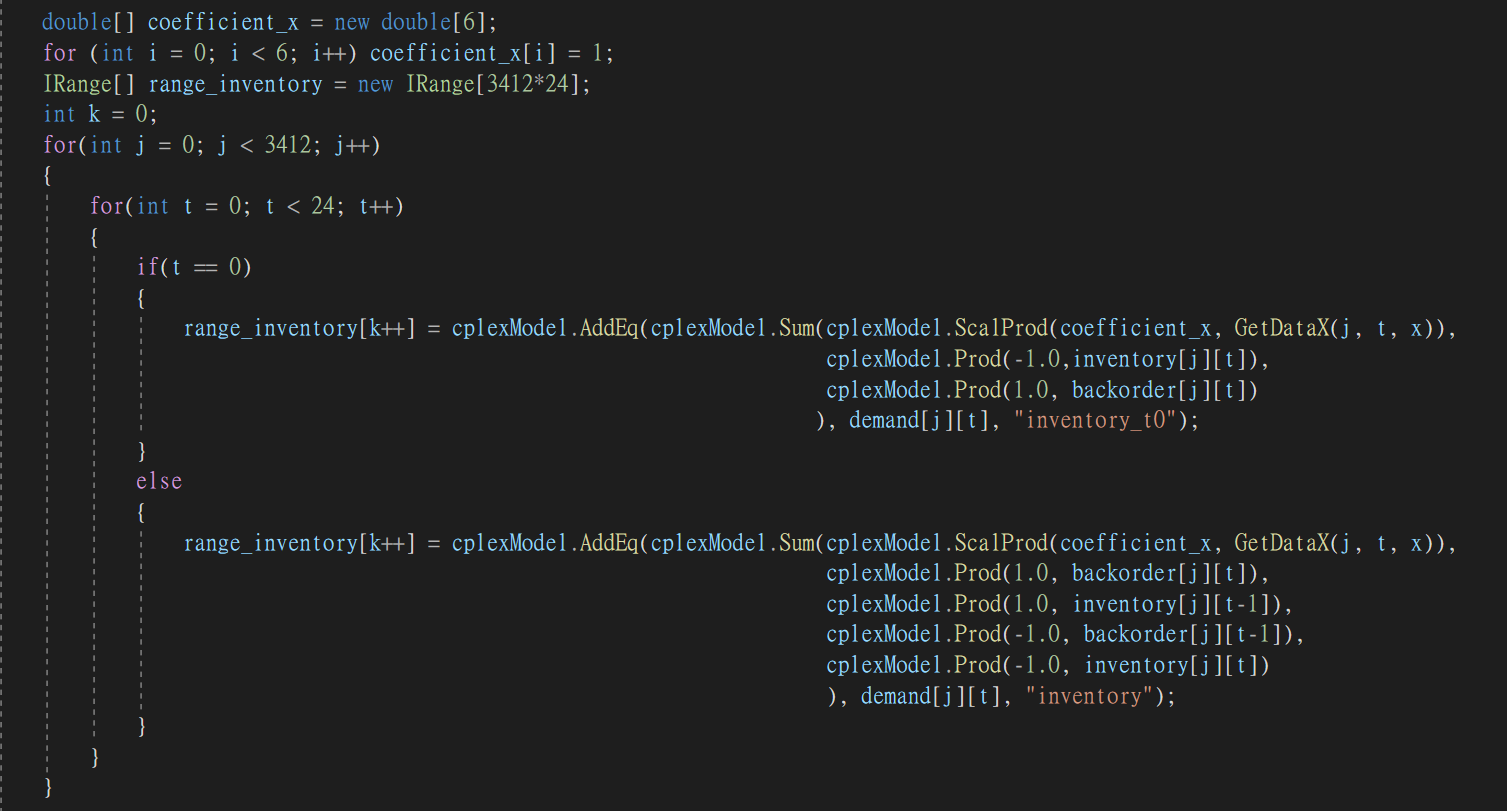
接著設定限制式，其中比較特別的有C工廠是需要兩種變數相加在一起，如下圖：



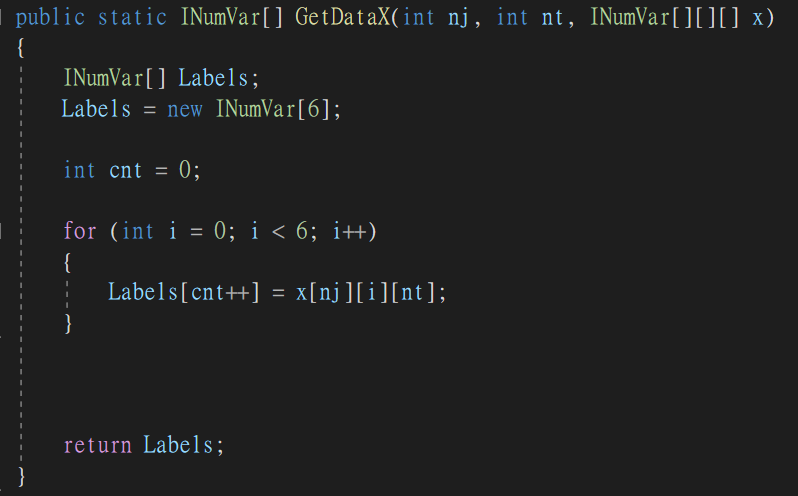
其實也有不用Sum，直接用一個ScalProd去設置，可以combine兩個array，如下圖：

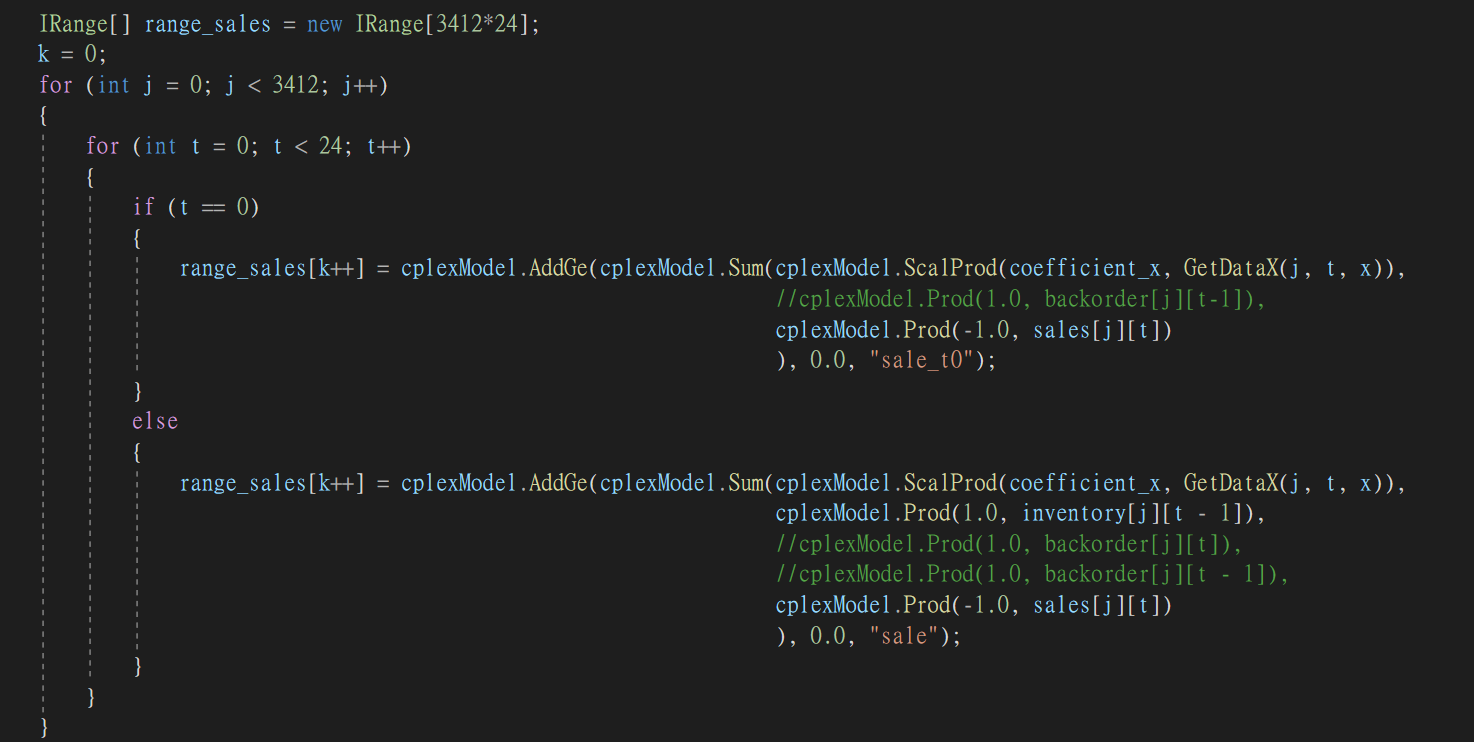


接著以下三條限制式都是需要額外function使得code更加精簡易閱讀

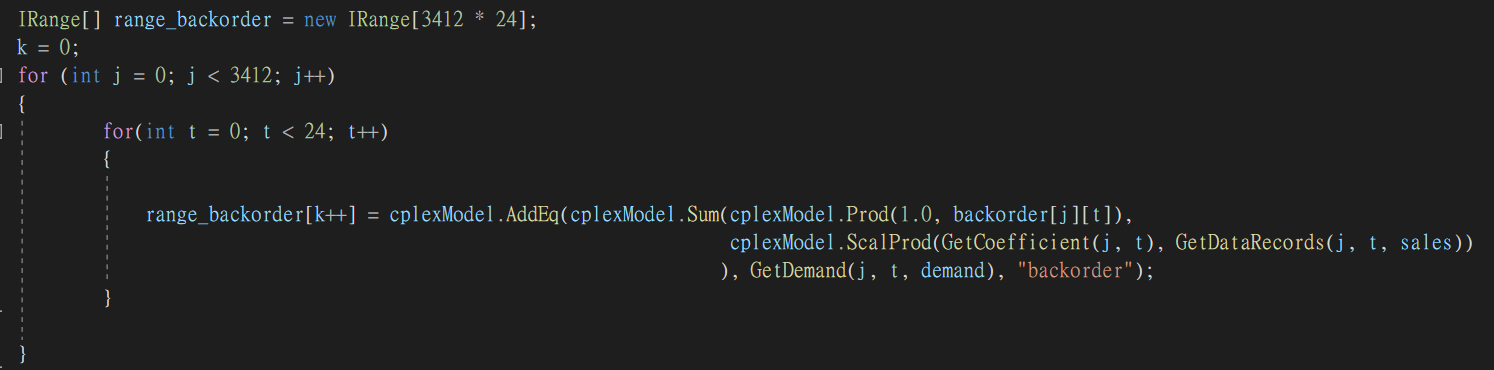


上圖是inventory的限制式，因為( t-1 )的關係，所以我設定( t = 0 ) 與 ( t != 0 )的區別，使其不會有溢位的錯誤，其中GetDataX是我額外設定，因為跟著j, t的改變，x也相對需要壓成一維，但因為有太多限制式要設定，直接硬設太費時，所以便設了一個去回傳一維矩陣，如下：

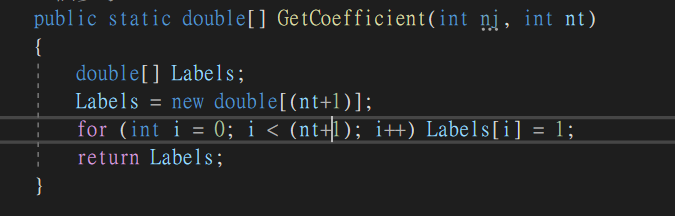




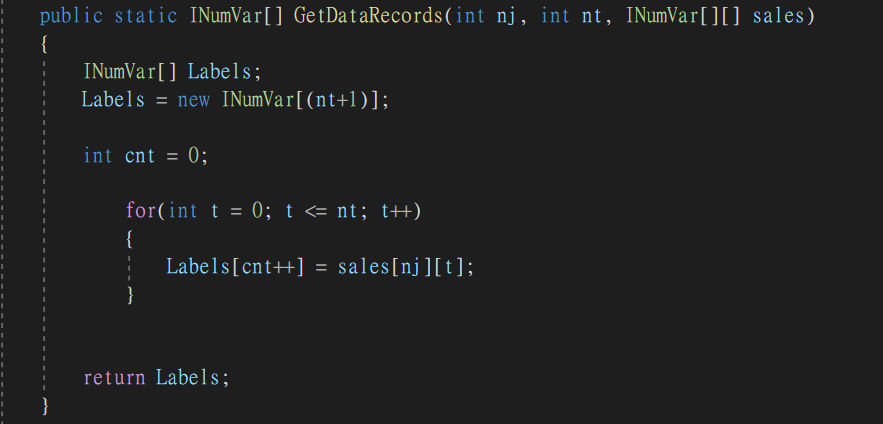
上圖是設置sales的限制式 ，我把s移到與I, s同一側，所以在設置限制式時用的是AddGe。



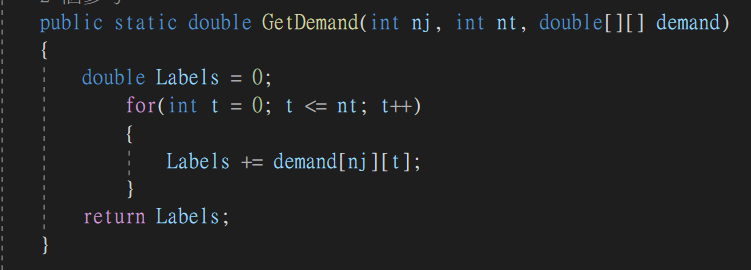
上圖是設定backorder的限制式 ，我把sales移到左邊，讓常數demand在右邊。每一次j, t的改變都要多設一條限制式，並且demand, sales的t也需要連續加起來及壓成一維，其係數及變數都會隨著j, t改變，其中t又更複雜的一點是，同一個產品（j）在每個月都必須從頭加一次，所以我多建幾個function，如下：



上圖是設定coefficient的function



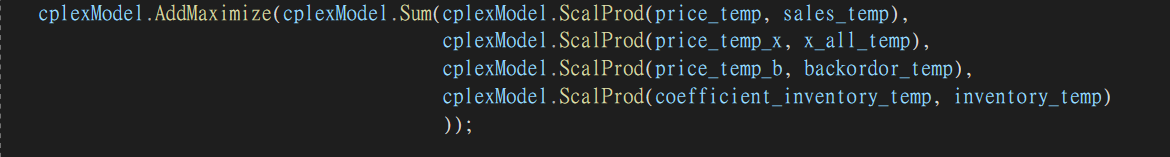
上圖是將sales二維壓成一維，並回傳其array的function



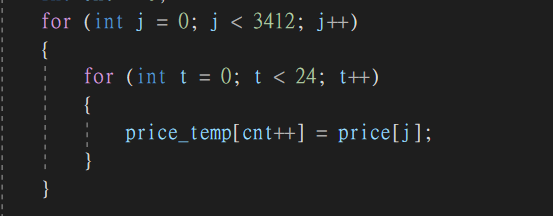
上圖是將同一產品從第一個月到個月的demand全部相加，並回傳其值的function

在我最初版的code有把，這條限制式加進去，但是好像不太影響最後結果，所以我就把其刪掉，簡化code

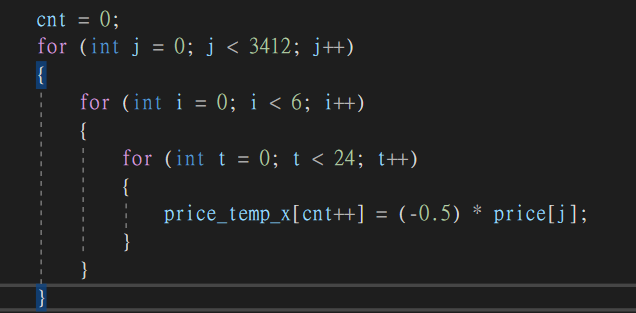
接下來是目標函式最大化



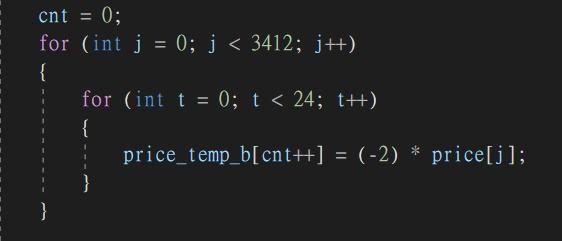
Z = 銷量 – 製造成本 – 存貨成本 – backorder成本



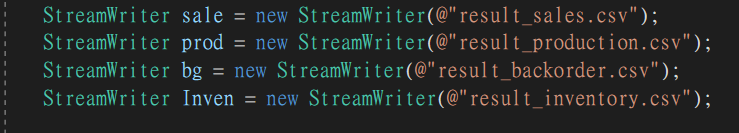
上圖是在把每一個產品價格壓成一維，因為為代表每一個產品都有24個月，所以依照限制式，我需要用兩個迴圈來壓成一維。



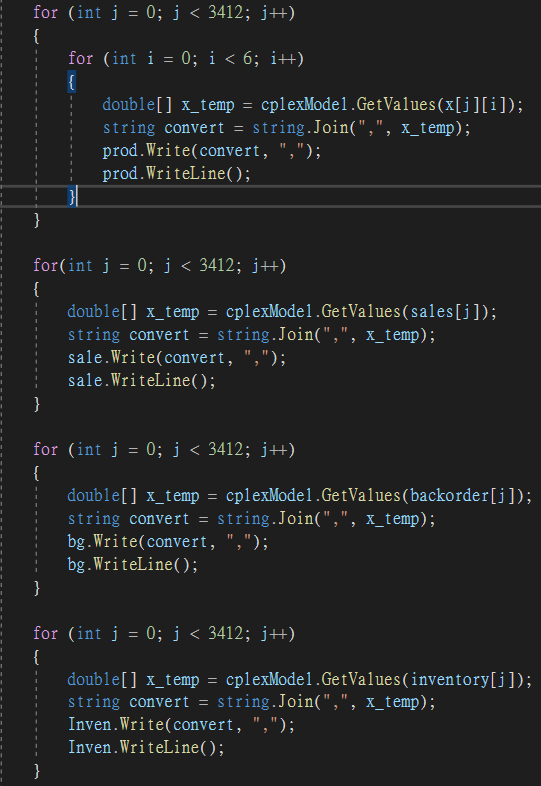
上圖為製造成本的係數壓成一維。



上圖為backorder係數壓成一維。

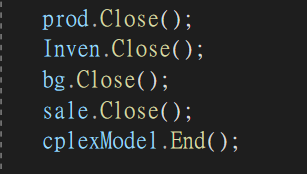


上圖為輸出銷量、產量、number of backorder、number of inventory的csv檔



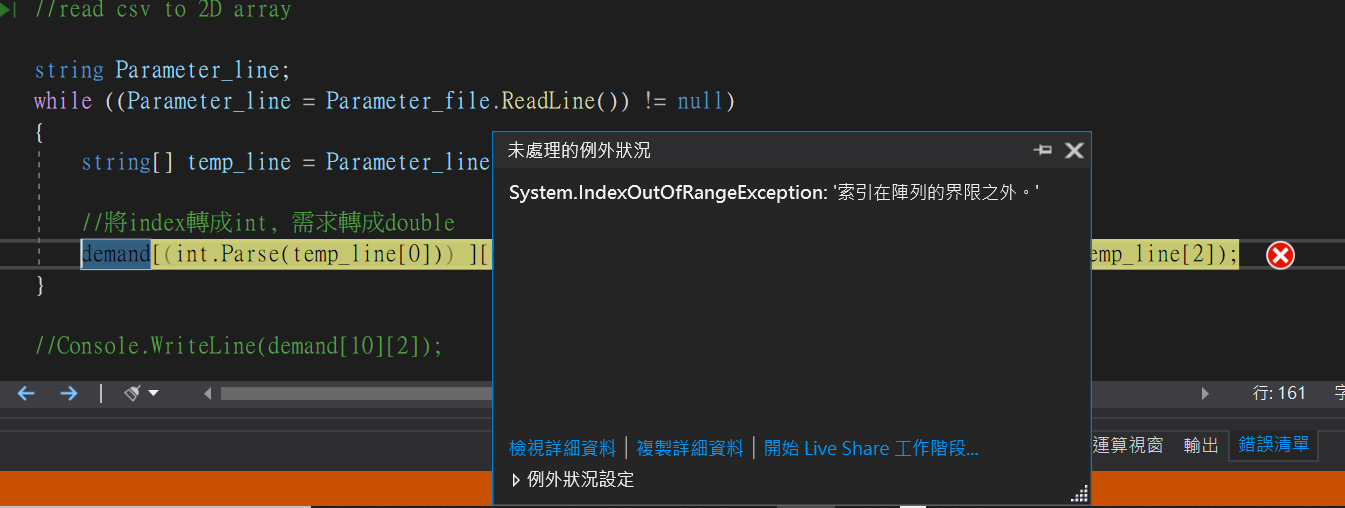
上圖為把各個變數解出來的數值。比較麻煩的是三維（產量）的輸出，因為需要每個商品在不同工廠一次印出24個月的產量，每個數值用逗號隔開。

最後，關閉所有檔案，程式結束。



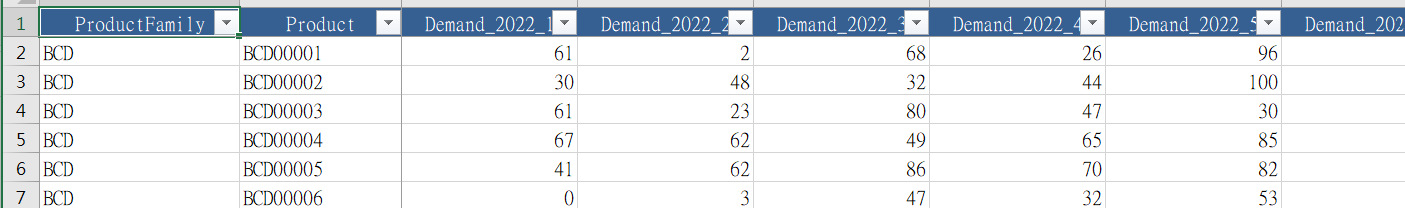
**結語**

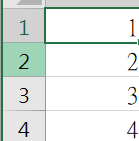
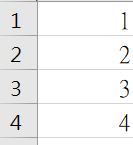
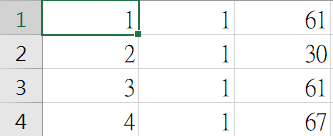
1. 我所遇到的問題—限制式
2. 我所遇到的問題—code
   1. Input問題：在首次input的時候會跑出「索引在陣列的界線之外」，Debug後發現有可能有以下兩點原因



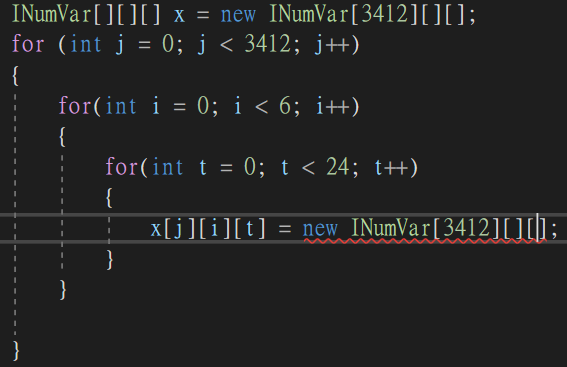
* + 1. Excel 格式錯誤：原本的應該會 -> 月份在最上面那一row，產品會是column，也就代表一個產品會有24個column；

而範例的應該是：

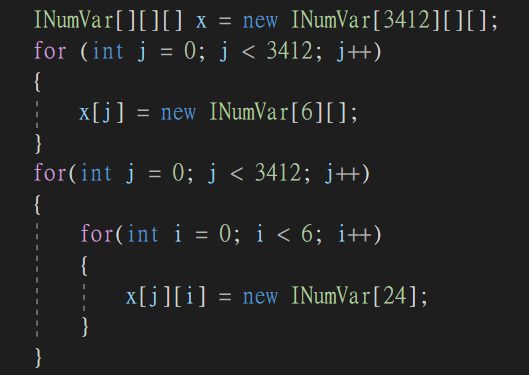
(原本)

(product) (period = t) (djt)

* + 1. Excel 轉換問題：當我把excel轉成上述說明的形式時，我轉換錯誤，使得product的數量有誤，同樣導致上述「索引在陣列的界線之外」的問題。
  1. 設置三維變數：當我要設置三維變數時，最初我設定的方式跟二維一樣。

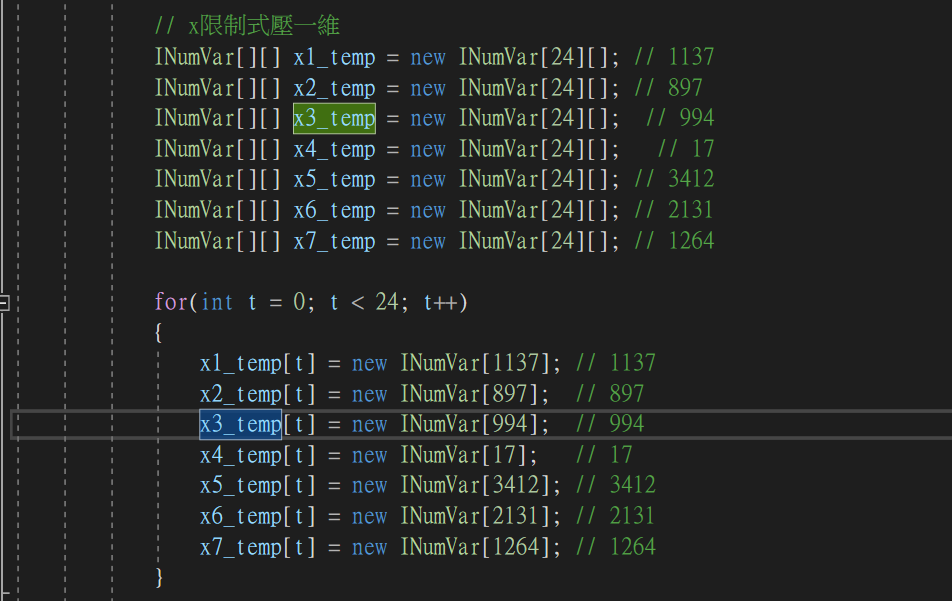


我也曾上網查其他的設定方法如下：

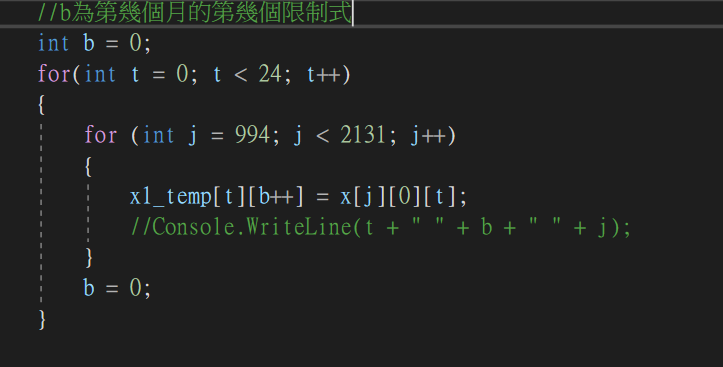
但後來我發現，c#不能像c/c++一樣，一次設定好所有維度的量，得分批也就是設立三個迴圈去一層一層設定，如旁：

* 1. 工廠設定問題（xjit）：起初我沒有我只有設定某個工廠可以生產那些產品，並沒有設定不能生產那些產品，導致跑出來的答案optimal solution會非常小
  2. 因為一行限制式有很多變數所以不能直接用一個一個設，所以目前的方法是用內積的方法來去解決，而其方法只能用一維去內積，所以我先把三維、二維都壓成一維。

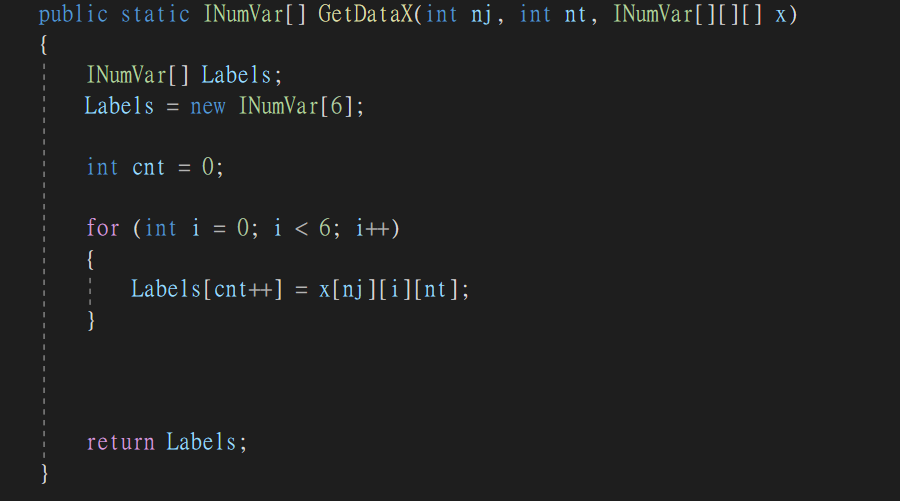
最初因為我不會設定，把一個變數就當成一個限制式，但之後找到錯誤。同時，我意識到因為每個月分都得要設成相同矩陣數量，也是是說每個工廠都要設24個相同但不同月份的陣列，因為我認為硬去一個個設定太過麻煩及冗長，所以設定二維陣列，第一維為月份，第二維為儲存「被壓成一為陣列後的」



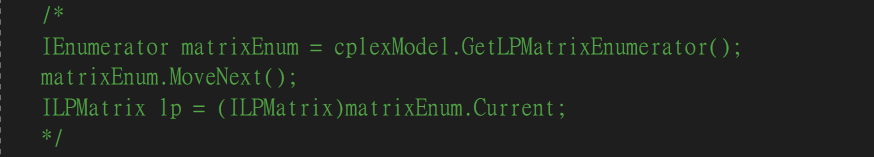
下圖為將三維壓成一維的過程：

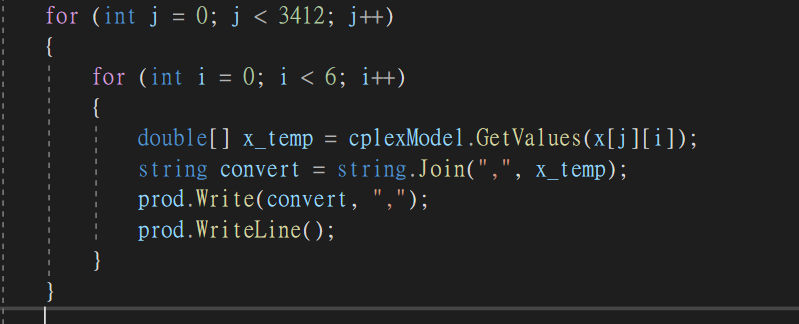


* 1. 因為以前我只有學過c++，所以我並不知道該怎麼設定function，但是上網查之後，發現其實在c#是用public static去設定，例如下圖為三維壓一維的function：



* 1. 下圖為輸出csv檔，在這邊我跟助教提供的code比較不一樣，一開始我想要用LP去解，然後直接getvalue，但是後來發現，因為在這邊設的變數可以直接getvalue，不需要多一個步驟去解出來，但因為變數大部分都是三維跟二維不能直接getvalue，所以只能一層一層去output出來，在這邊我是以產品為單位，一次印出24個月分





1. 我學習到的額外技巧

在做這份報告中，我學習到不止是OR以及coding的技巧，還包括excel的基本運以及如何看lp檔debug，在debug的過程我也更加熟悉設定變數的意涵，在這項作業之前，我是基本上沒有使用過C#、Cplex、excel等等，但我通過此番，我認為我增進的不僅僅是相關技巧的使用，以及如何運用所學去完成一項實際案例，更具有價值的是，學習如何解決問題的能力，當一項作業所需要的能力基本沒有時，我一開始也十分慌張，但是冷靜下來，給自己加個油，我相信關關難過關關過，一步一步地走，總會有完成的一天！