程式設計（108-1）
期中專案

題目設計：孔令傑
國立臺灣大學資訊管理學系

說明：本專案題目的類似題目曾經出現在之前幾份作業。為了閱讀上的方便，在這份文件中我們完整地描述了整個問題。你可以利用你之前的想法或程式碼[1]，不過除非你想多看一些範例，否則你不需要回去看那兩個題目的敘述。

1 題目敘述

某工廠想替產品的生產線進行產能規劃，以減少生產成本。我們假設目前工廠只有一條生產線，此生產線由若干個工作站組成，每一個工作站中各有若干台機台。舉例來說，假設此生產線由 2 個工作站組成，則採購原料進來之後，要將原料投入工作站 1，得到工作站 1 產出的在製品（work-in-progress，WIP），再將工作站 1 產出的在製品投入工作站 2，得到完成品（最終產品）。在本專案中，我們希望透過生產方案規劃，在確保最終產品數量滿足當日需求量，平衡每一站的生產量、存貨量與使用量，同時也控制每站每日的生產量不得超過產能上限的條件下，根據輸入的資訊，規劃最佳生產方案。以下我們將依序介紹問題描述以及一個簡單的演算法。

1.1 問題描述

假設目前工廠只有一條生產線，此生產線由 I 個工作站組成，每一個工作站中各有 M_i 台機台。針對每一個工作站，我們假設工作站內的機台具有同質性，也就是說，不同工作站之間雖有不同的機台，但單一工作站內的機台都是相同的。

在工作站 i 中每開一個機台進行生產，都必須付出一定的開機成本 C_i^{ON}；開機後，一台機台每生產一公斤產品的生產成本為 C_i^D 元，且一台機台每日的產能上限為 B_i 公斤，代表機台一天最多能產出 B_i 公斤的產品；在製造過程中，當日投入原料當日就能通過所有個工作站並產出完成品。各工作站的產品製成率為 R_i，製成率 R_i 定義為投入 R_i 公斤的原料會得到一公斤的產出，且 R_i 為 1 到 5 之間的正整數。

若工作站的產品成為存貨，每日每公斤的存貨成本則為 C_i^S 元，由於除了工作站 1 至工作站 I 的產品有庫存成本，原料購入後也有儲存幾日再投入工作站 1 生產的可能，

[1]或許你應該感謝當時你有好好模組化，讓你的某些程式（特別是函數）可以被重新利用。如果你正在咒罵當時的你沒有好好模組化，也請不要介意，因為大部分人都是如此。從今天起好好模組化吧！

我們可以把原料倉庫當作工作站 0，因此庫存成本共會有 $I+1$ 項。另外，在規劃的第一天，每站皆有可能會有一些初始存貨 S_i（以公斤計算），初始存貨皆可投入生產，同理，初始存貨也會有 $I+1$ 項。

已知需規劃的生產時段共有 T 天，規劃的目標為，在每日生產的最終產品數量（也就是生產線最後一個工作站的生產量）滿足當天需求量 D_t（以公斤計算）的條件下，找出能最小化總成本的最佳生產方案。總成本除上段中提到的開機成本 C_i^{ON}、單位生產成本 C_i^D 和存貨成本 C_i^S 外，顯然也必須包含原料成本，我們假設每日原料成本可能不同，以每公斤 P_t 元表示。在最佳生產方案中，必須規劃工廠工作站 1 至工作站 I 的每日生產量 x_{it}（定義為該工作站的產出量而非投入量，以公斤計算）以及每日機台開機數 n_{it}。此外，也必須規劃每日原料採購量 w_t（以公斤計算），

規劃中，首先必須確保最終產品數量滿足當日需求量，也必須平衡每一站的生產量、存貨量與使用量，同時也須控制每站每日的生產量不得超過產能上限（受到你決定的開機數量影響）。請根據輸入的資訊，規劃最佳生產方案。

1.2 一個簡單的演算法

為了協助你進入狀況，這裡我們提供一個簡單（愚蠢?）的演算法。我們首先忽略所有的初始存貨（對，就是這麼愚蠢），並且計算在產能有限下，如果開啟所有機台，每天最多能產出多少完成品。接下來我們計算每天的需求量總和。我們的生產計畫是，從第一天開始，每天都盡全力生產到完成品的產能上限，直到總生產量能滿足總需求量為止。每一天我們都採購剛好夠當日生產量的原料數量，然後剛好用完，因此每一天都不會有原料與在製品的存貨。只有最初數日會有完成品存貨。

以下舉一個簡單的例子。我們假設目前工廠只有一條生產線，此生產線由 3 個工作站組成，每一個工作站中各有 1 台機台，需規劃的生產時段共有 4 天。假設每一工作站的產能上限皆為 150 公斤，每一站的製成率分別為 3、1、2，從第一天到第四天的需求量分別為 30、20、100、60。演算法執行步驟如下：

- 首先，計算每天的需求量總和：$30+20+100+60=210$。
- 接下來，計算每天的完成品產能上限。由於每一站的產能皆為 150 公斤且工作站 1 的製成率為 3，因此，為了讓工作站 1 能生產 150 公斤，需要 450 公斤的原料投入。而工作站 2 的製成率為 1，因此投入 150 公斤的在製品便能產出 150 公斤的在製品。接下來再將 150 公斤的在製品投入到工作站 3 ，由於工作站 3 的製成率為 2，150 公斤的投入只能產出 75 公斤完成品。因此，每日的完成品產能上限為 75 公斤。

- 因此我們的生產計畫如下。從第一天起，每天皆生產 75 公斤，直到生產完 210 公斤為止，因此第三天只需生產 60 公斤。前兩天我們都採購 450 公斤原料，第三天則採購 360 公斤。每一日的剩餘完成品存貨則依序為 45（$75-30$）、100（$75+75-30-20$）、60（$75+75+60-30-20-100$）與 0 公斤。

透過此演算法可以得到一個合乎要求的可行解，但未必是很省成本的作法。這次的期中專案，就是希望大家可以發想並且嘗試不同的演算法，看看能否讓你的演算法可以得到比這個方案更好的解。

2 輸入輸出格式

系統會提供一共 25 組測試資料，每組測試資料裝在一個檔案裡。每個檔案中會有十行：

- 第一行包含兩個整數 I 和 T。
- 第二行至第六行皆包含 I 個整數，第二行依序為 M_1、M_2 直到 M_I，第三行依序為 C_1^{ON}、C_2^{ON} 直到 C_I^{ON}，第四行依序為 C_1^D、C_2^D 直到 C_I^D，第五行依序為 B_1、B_2 直到 B_I，第六行依序為 R_1、R_2 直到 R_I。
- 第七行與第八行皆包含 $I+1$ 個整數，第七行依序為 C_0^S、C_I^S 直到 C_I^S，第八行依序為 S_0、S_1 直到 S_I。
- 第九行及第十行皆包含 T 個整數，第九行依序為 D_1、D_2 直到 D_T，第十行依序為 P_1、P_2 直到 P_T。

以上每一行中，兩兩整數之間皆被一個空白隔開。已知 $1 \le I \le 100$、$1 \le T \le 1000$、$1 \le M_i \le 100$、$1 \le C_i^{ON} \le 1000$、$1 \le C_i^D \le 1000$、$1 \le B_i \le 10000$、$1 \le R_i \le 5$、$1 \le C_i^S \le 1000$、$1 \le S_i \le 10000$、$1 \le D_t \le 10000$、$1 \le P_t \le 10000$。

讀入資料後，請先用 I 行輸出一個 $I \times T$ 的矩陣代表 n_{it}，接著再用 I 行輸出一個 $I \times T$ 的矩陣代表 x_{it}，最後再用一行輸出 T 個數字代表 w_t。在第 1 到第 I 行中，第 i 行依序為 $n_{i,1}$、$n_{i,2}$ 直到 n_{it}。在第 $I+1$ 到第 $2I$ 行中，第 $I+i$ 行依序為 $x_{i,1}$、$x_{i,2}$ 直到 x_{it}。在第 $2I+1$ 行中，依序為 w_1、w_2 直到 w_t。每一行中皆有 T 個數字，兩數字間以一個空白隔開。

舉例來說，如果輸入是

```
3 4
```

```
1 1 1
500 700 900
2 3 4
150 150 150
3 1 2
1 1 2 5
1000 1000 1000 1000
30 20 100 60
10 7 3 2
```

則採用上述的簡單演算法，輸出應該是

```
1 1 1 0
1 1 1 0
1 1 1 0
150 150 120 0
150 150 120 0
75 75 60 0
450 450 360 0
```

PDOGS 會讀取你的輸出、檢查可行性、判定為可行，並且計算你得到的總成本是 **54995**。如果輸入是

```
2 5
10 10
200 600
20 10
2000 2000
3 2
4 10 25
1000 1000 2350
1200 1500 800 2100 1000
10 40 30 10 5
```

則採用上述的簡單演算法，輸出應該是

```
7 0 0 0 0
4 0 0 0 0
```

```
13200 0 0 0 0
6600 0 0 0 0
39600 0 0 0 0
```

PDOGS 會讀取你的輸出、檢查可行性、判定為可行，並且計算你得到的總成本是 **1428550**。

你的.cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你***可以***使用任何方法。

3 評分原則

這一題的其中 75 分會根據程式運算的結果給分。你的程式不需要找出真的能最小化總成本的最佳方案（optimal solution）。只要你的輸出符合規定，且確實是一組可行方案（feasible solution，滿足每日需求量、滿足每日產能限制、原料與在製品是否足以供應生產計畫），就會得到分數。對於每一組輸入，PDOGS 會檢查你的輸出，如果輸出格式不合乎要求或方案不可行，則在該筆測試資料會得到零分；如果合乎要求，且目標式值非負，則對每筆測資，我們依下列公式計分：假設 z 是這組的生產成本、z_1 是上述「一個簡單的演算法」的生產成本、z_0 是所有組的生產成本中最小的，則在這筆測資的得分就是

$$3\left(\frac{z_1 - z}{z_1 - z_0}\right)。$$

寫程式之外，每組還需要合力用中文或英文寫一份書面報告（所謂「寫」，就是用電腦打的意思），以組為單位上傳 PDF 檔至 PDOGS。在報告裡請用文字描述你的演算法（可以用 pseudocode 但不能直接貼 code）、系統的設計（哪個函數做什麼、程式執行的流程等等）、分工方式（誰寫哪個函數、誰負責指揮、誰負責寫書面報告、誰負責買便當等等；當然一個人可以又買便當又寫程式），以及每個人的簡單心得感想。報告***不可以超過八面 A4 紙***。書面報告佔 25 分。這份專案截止後，書面報告才會被批改。

此外，授課團隊會考慮 PDOGS 上的得分以及書面報告上的解法，邀請若干組在合適的上課時間，每組用 10 分鐘跟全班同學介紹自己的解法。被邀請的組可以婉拒上台報告，但上台報告的組可以獲得專案總成績 5 分的額外加分。

4 繳交方式

請修課的同學們自行組成每組三至四人的小組，以組為單位繳交你們的程式和報告。

有兩件事需要注意。首先，系統會以該組內任意一位同學的最後一次上傳得到的分數，做為該組所有人的分數，所以愈傳愈低分是有可能的。其次，原則上 PDOGS 不限制兩次上傳間的時間間隔，但如果有許多組在同個時間大量地上傳執行時間很長的程式，導致 PDOGS 大塞車，屆時我們會對兩次上傳的時間間隔做出限制。

程式的截止時間是 ***12 月 3 日早上八點***，書面報告的截止時間是 ***12 月 5 日早上八點***。