

## 全國高級中等學校 108 學年度工業類科學生技藝競賽電腦軟體設計

### 壹、試卷說明：

1. 請將寫好之程式原始檔依題號命名資料夾存檔，第一題取姓名\_Q1，第二題取姓名\_Q2，依序命名存檔，並存於 C 碟之資料夾” 姓名\_Contest” 中。

2. 競賽時間 4 小時。

3 將程式及編譯成執行檔儲存在 C 碟之資料夾姓名\_Contest。

貳、評分說明:本試卷共六題，每題配分不一。

1. 每題評分只有對與錯兩種，對則給滿分，錯則不給分(即以零分計算)。

2. 每解答完一題上傳(程式及執行檔)，評審人員將針對該題進行測試，若解題正確則回應正確，若解題錯誤則扣該題一分至該題零分為止，答錯之題目可繼續作答。(注意：程式輸出的資料字型大小至少 12)

### 試題一：自動販賣機設計 (16 分)

說明: 1.設計一飲料販賣系統程式，模擬投幣式販賣機功能，所販售之飲料有 4 種: Cola, PEPSO, Diet Cola, Diet PEPSO，消費者可依照自己的喜好選擇各種飲料。運作方式是需先投幣儲值，每次投幣先選幣別如 5 元、10 元或 50 元，再按下“投幣鍵”，積累幣值會顯示在“存款 balance\$”，每次只能投一硬幣但可重複投，例如 10 元投幣兩次如圖 1-1 所示，若儲值不足以購買任何一種飲料，可選擇按下“退款鍵”退還存款，輸出“退還 20 元”資訊及清除存款為“0.0”，如圖 1-2 所示。



圖 1-1



圖 1-2

2. 若儲值足以購買任何一種飲料，如投幣投 10 元兩次及 50 元一次，“存款 balance\$” 顯示儲值 70.0，如圖 1-3 所示。用滑鼠點選 4 種飲料任一種，如點選 Cola，存款更新為“35.0”及輸出“送出 Cola”資訊，如圖 1-4 所示。若再點選 PEPSO，存款更新為“5.0”及輸出“送出 PEPSO”資訊。若再點選任何一種飲料，但是存款不足以購滿任何一種飲料，此時狀態不作任何變動，此時可選擇再投幣增加存款值再購滿飲料，或選擇按下“退款鍵”退還存款，此時狀態是輸出信息為“送出 Cola, 送出 PEPSO 退還 5 元”及存款更新為“0.0”如圖 1-5 所示。

3. 若“存款 balance\$” 的儲值 0.0，輸出欄尚有舊資訊，可按下“退款鍵”清除舊資訊，如圖 1-6 所示。

4. 程式欲結束執行可自行設計一“結束”鍵，來執行結束。



圖 1-3



圖 1-4



圖 1-5

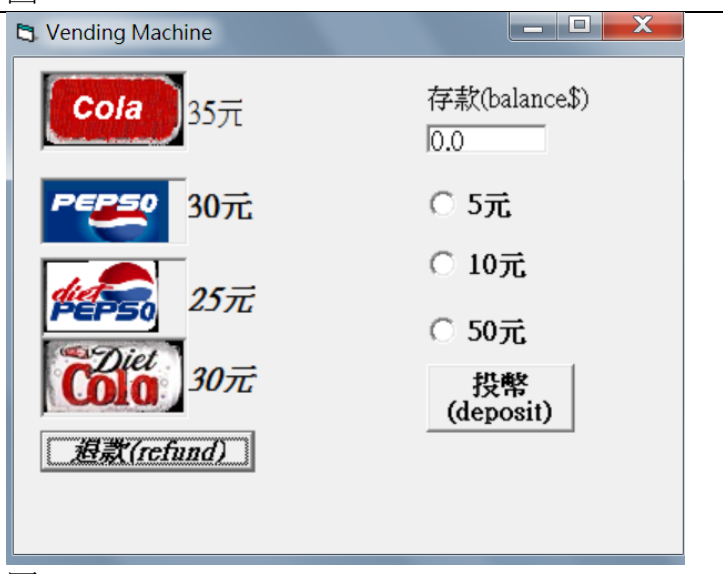


圖 1-6

## 試題二：股價分析程式(17 分)

- **題目說明：**某日股價分析包括股票的代號、名稱、前交易日收盤價、開盤價、最高價、最低價、收盤價、成交量、漲跌價、漲停板、跌停板。
- **專有名詞：**
  - 前交易日收盤價**---指前一個交易日的收盤價格。
  - 開盤價**---指每天成交中最先的一筆成交的價格。
  - 最高價**---指當天股票成交的各種不同價格是最高的成交價格。
  - 最低價**---指當天成交的不同價格中最低成交價格。
  - 收盤價**---指每天成交中最後的一筆股票的價格，也就是收盤價格。
  - 成交量**---指當天成交的股票數量。
  - 漲跌價**---每天的 "收盤價" 減去 "前交易日收盤價" 的價格。在公告牌上用 "+" 或 "-" 號表示。
  - 漲跌幅(%)**---為 (當日收盤價 - 前交易日收盤價) / 前交易日收盤價，以%顯示。
  - 漲停價**---為穩定股價，股票漲幅度不得超過前交易日收盤價的 10%之價格。
  - 跌停價**---為穩定股價，股票跌幅度不得超過前交易日收盤價的 10%之價格。
  - 股票升降單位**---採 6 個級距，

每股市價若是 未滿 10 元者，股價升降單位為 0.01 元，  
 每股市價若是 10 元至未滿 50 元者，股價升降單位為 0.05 元、  
 每股市價若是 50 元至未滿 100 元者，股價升降單位為 0.1 元、  
 每股市價若是 100 元至未滿 500 元者，股價升降單位為 0.5 元、  
 每股市價若是 500 元至未滿 1000 元者，股價升降單位為 1 元、  
 每股市價若是 1000 元以上者，股價升降單位為 5 元。

● **分析範例：**以下單位均為元，有一支股票，前一交易日收盤價為 16.35，

■ 若當日收盤價為 16.4，則漲跌價為+0.05(=16.4-16.35)，漲跌幅為 0.31%(=0.05/16.35)。

■ 該支股票的漲停價及跌停價計算，該股的升降單位為 0.05，前交易日收盤價的 10% 之價格為 1.635，有 32.7(=1.635/0.05) 個升降單位，取整數得 32 個升降單位，

故該支股票的漲停價為 17.95(16.35+32\*0.05)，故該支股票的跌停價為 14.75(16.35-32\*0.05)。

■ 該支股票的當日收盤價為漲停價，該支股票為漲停，該支股票的當日收盤價為跌停價，該支股票為跌停。

● **設計說明：**

請撰寫一支程式，能讀入 10809.txt 檔案以便進行以下股價分析。

	A	B	C	D	E	F	G	H	I	J	K	L
1	代號	名稱	前交易日收盤價	開盤價	最高價	最低價	收盤價	成交量	漲跌價	漲跌幅	漲停板	跌停板
2	*4968	立積	155	160	170.5	159	170.5	15950	+15.50	10.00%	漲停	
3	*3581	博磊	15.05	15.1	16.55	15.1	16.55	613	+1.50	9.97%	漲停	
4	*2504	國產	8.46	9.3	9.3	9.3	9.3	12272	+0.84	9.93%	漲停	
5	*3588	通嘉	23.7	23.7	26.05	23.7	26.05	1621	+2.35	9.92%	漲停	
6	*2429	銘旺科	18.75	18.75	20.6	18.75	20.6	120	+1.85	9.87%	漲停	
7	*3041	揚智	11.7	11.9	12.85	11.9	12.85	47101	+1.15	9.83%	漲停	
8	*2337	旺宏	32.1	32.6	33.8	32.2	33.8	155939	+1.70	5.30%		
9	*2505	國揚	12.5	12.5	13.5	12.5	13	1523	+0.50	4.00%		
10	*6202	盛群	65.2	65.2	66.8	65.2	66.8	767	+1.60	2.45%		
11	*2317	鴻海	73.5	73.8	74.5	73.8	74.5	33275	+1.00	1.36%		
12	*2331	精英	12.85	12.9	13.25	12.85	13	1665	+0.15	1.17%		
13	*1101	台泥	39.8	39.8	40.2	39.65	40.2	13091	+0.40	1.01%		
14	*3679	新至陞	47.5	47.5	47.9	47.4	47.8	7	+0.30	0.63%		
15	*2423	固緯	25.85	26	26.15	25.8	26	142	+0.15	0.58%		
16	*2498	宏達電	37.5	37.6	38.15	37.3	37.7	6580	+0.20	0.53%		
17	*2332	友訊	12.1	12.15	12.2	12.05	12.15	2024	+0.05	0.41%		
18	*1102	亞泥	43.6	43.65	43.9	43.5	43.65	7163	+0.05	0.11%		
19	*2427	三商電	10.55	10.55	10.55	10.5	10.55	164	+0.00	0.00%		
20	*3008	大立光	4300	4320	4350	4290	4300	897	+0.00	0.00%		
21	*2330	台積電	265	266	266.5	264	264	43868	-1.00	-0.38%		
22	*9962	有益	10.75	10.75	10.75	10.65	10.7	23	-0.05	-0.47%		
23	*2324	仁寶	18.2	18.15	18.2	18.05	18.05	13706	-0.15	-0.82%		
24	*2308	台達電	148.5	148.5	148.5	145.5	145.5	8759	-3.00	-2.02%		
25	*6154	順發	16.1	15.1	15.1	14.5	14.5	201	-1.60	-9.94%		跌停
26	*3678	聯享	8.52	8.45	8.45	7.67	7.67	107000	-0.85	-9.98%		跌停
27	*8087	華鎂鑫	8.29	7.78	8.05	7.47	7.46	12	-0.83	-10.01%		

- 顯示 "漲跌價" 最高的 3 支股票的名稱及其漲跌價。
- 顯示 "漲跌價" 最低的 3 支股票的名稱及其漲跌價。
- 顯示 "漲跌幅" 最高的 3 支股票的名稱及其漲跌幅%。
- 顯示 "漲跌幅" 最低的 3 支股票的名稱及其漲跌幅%。
- 顯示 所有漲停股票的名稱及其漲停價。
- 顯示 所有跌停股票的名稱及其跌停價。

<<注意：評判時會以另外的檔案測試>>

● 程式執行結果

```
D:\093教材\1081教學\工科競賽>cd D:\108工科軟體競賽
```

```
D:\108工科軟體競賽>股價分析程式.exe
```

```
讀入 ./10809.txt 檔
```

```
漲跌價最高之3支股票(漲價元): 立積(15.50) 通嘉(2.35) 銘旺科(1.85)
```

```
漲跌價最低之3支股票(跌價元): 聯享(-0.85) 台積電(-1.00) 順發(-1.60)
```

```
漲跌幅最高之3支股票(漲幅%): 立積(10.00%) 博磊(9.97%) 國產(9.93%)
```

```
漲跌幅最低之3支股票(跌幅%): 順發(-9.94%) 聯享(-9.98%) 華鎂鑫(-10.01%)
```

```
所有漲停股票(漲停價元): 立積(15.5) 通嘉(2.35) 銘旺科(1.85) 博磊(1.5) 揚智(1.15) 國產(0.84)
```

```
所有跌停股票(跌停價元): 聯享(-0.85) 順發(-1.6)
```

```
D:\108工科軟體競賽>
```

### 試題三：解密碼演算法 (16 分)

有一個簡易密碼演算法可以將一段長度為 5 的字串，編碼成 5 個 32 bits 十六進制的數值，例如：輸入 apple，將會得到 068afa4e 01871c0c 005af200 00160ecc 000553e6。相反的，給你 5 個 32 bits 十六進制的數值，請你解答原來的字串，也就是說給你 068afa4e 01871c0c 005af200 00160ecc 000553e6，你必須解密碼得到原來的字串 apple。

#### Algorithm: 簡易密碼演算法

**Input:** M[0..4]: 五個字元

**Output:** 五個 32 bits 的十六進制數值

/\*0x 開頭表示該數值為 16 進制的數值\*/

1.  $h0 \leftarrow 0xabcd, h1 \leftarrow 0xcdef, h2 \leftarrow 0x2266, h3 \leftarrow 0xceed, h4 \leftarrow 0xaccd;$
2.  $a \leftarrow h0, b \leftarrow h1, c \leftarrow h2, d \leftarrow h3, e \leftarrow h4;$
3. for  $i \leftarrow 0$  to 4 do
4.      $word[i] = M[i] - ' ';$  //減空白字元
5.      $f \leftarrow b + c; k \leftarrow 0x5a82;$
6.      $temp \leftarrow 4 * a + f + e + k + word[i];$
7.      $e \leftarrow d; d \leftarrow c; c \leftarrow b; b \leftarrow a; a \leftarrow temp;$
8. end for
9.  $h0 \leftarrow h0 + a; h1 \leftarrow h1 + b; h2 \leftarrow h2 + c; h3 \leftarrow h3 + d; h4 \leftarrow h4 + e;$
10. //輸出 5 個 32bits 的結果  
     $printf("%08x \%08x \%08x \%08x \%08x\n", h0, h1, h2, h3, h4);$

#### 注意事項：

1. 所有變數都是 32bits 無號數
2. 輸入字串 M 長度為 5
3. 第四行將每一個字元 M[i] 透過減掉空白字元，轉換成整數數值，並儲存在 word[i] 裡面

#### 1. 輸入說明：

輸入含有多組測試資料。每組測試資料為 5 個 32 bits 十六進制的數值，數值之間以空白隔開。請參考範例輸入。

#### 2. 範例輸入

```
068afa4e 01871c0c 005af200 00160ecc 000553e6
```

```
068b0621 01871ecd 005af2a6 00160ef5 000553ef
```

```
068b0bfc 01872028 005af2f4 00160f07 000553f5
```

#### 3. 輸出說明:輸出長度為 5 的字串



#### 4. 範例輸出

apple

juice

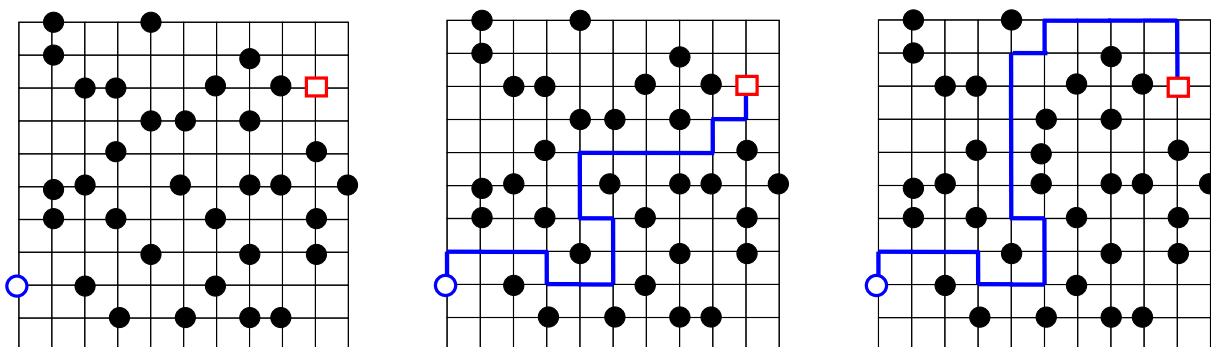
point

A. 評分標準:1 介面不拘，可以輸入測試資料並輸出結果即可。2 考測試資料如上三組，評審會測試其它多組資料，測試資料皆正確，才能通過。

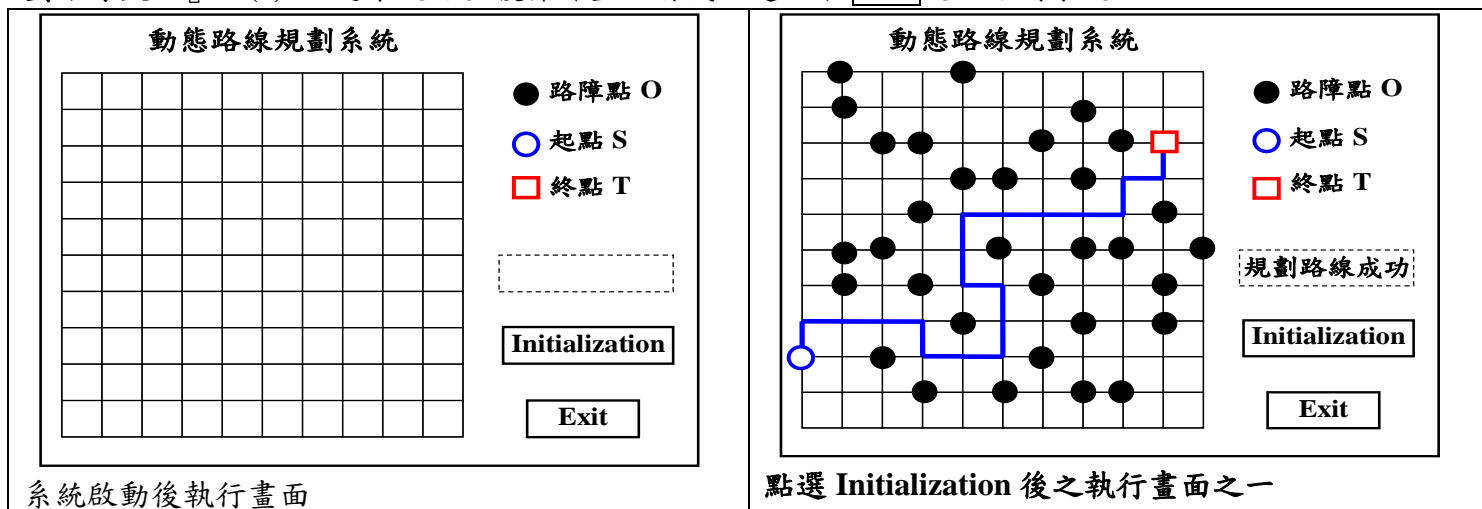
#### 試題四：動態路線規劃系統(Dynamic Route Planning System) (17 分)

說明：(一) 動態路線規劃 (Dynamic route planning) 是人工智慧的基本應用之一，它能有智慧地搜尋起點到終點之最短或最佳的路線，當路障點 O (Obstacle points) 臨時增加或解除時，路線規劃必須要重新調整；當起點 S(Starting point) 或終點 T (Target point) 臨時改變時，路線規劃也必要自動隨著改變。

(二) 如下圖左方所示為 11x11 方陣網格點 (grid point)，實體黑色圓形為路障點 O，空心圓形為路線規劃的起點 S，空心方形為路線規劃的終點 T；如下圖中間所示，從起點 S 到終點 T 且避開所有路障點，可以規劃出一條最短的路線 (shortest path)；如下圖右方所示，當一個路障點移動到新位置時，可從起點 S 到終點 T 重新規劃出一條最短的路線。但也有例外情形，當路障點 O、起點 S 或終點 T 被移動或設定到新位置時，重新規劃路線而可能找不到任何一條可行路線而宣告規劃失敗。



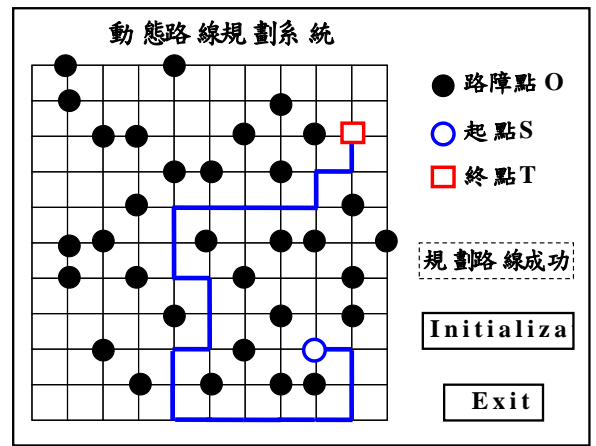
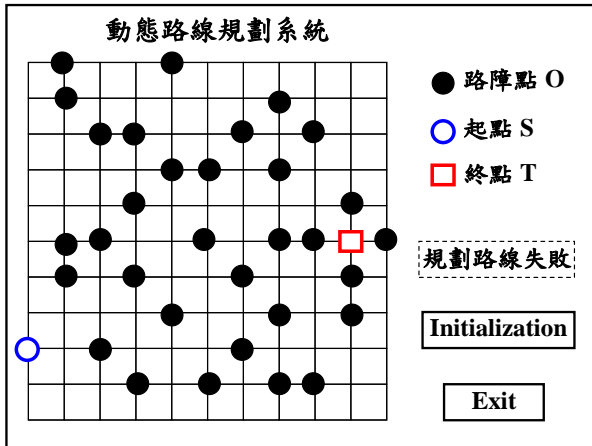
(三) 系統設計：(1)請參考以上陳述方法，設計如下圖左方所示具有 11x11 方陣網格點之「動態路線規劃系統」，每個網格點只限制四個放置狀態之一：空置點、路障點 (實體黑色圓形)、起點 S (空心圓形)、或終點 T (空心方形)，後面三者間絕不允許有任何重疊放置。(2) 每當滑鼠點選一下初始鍵 **Initialization**，系統能自動重置後，並隨機產生 32 個路障點、一個起點 S 及一個終點 T；接著，系統自動從起點 S 到終點 T 且避開所有路障點而規劃出一條最短的路線。如此路線存在則顯示『規劃路線成功』，如下圖右方所示；否則顯示『規劃路線失敗』。(3)每當滑鼠點選一個路障點、起點 S 或終點 T 後，該點即被移除，而可在任一空置點重新放置剛被移除之路障點、起點 S 或終點 T 於新位置後，系統接著自動再從起點 S 到終點 T 且避開所有路障點而重新規劃出一條最短的路線。如此路線存在則顯示『規劃路線成功』；否則顯示『規劃路線失敗』。(4) 上述系統可重複操作直至滑鼠點選一下 **Exit** 鍵而離開系統。



### 範例一

輸入格式：參考右上圖畫面，當滑鼠點選**終點 T**而移動或設定到新位置後。

輸出格式：系統自動隨著再從**起點 S**到**終點 T**且避開一些**路障點**而重新規劃出一條最短的路線，結果為此路線不存在則顯示『**規劃路線失敗**』，如下圖左方所示。



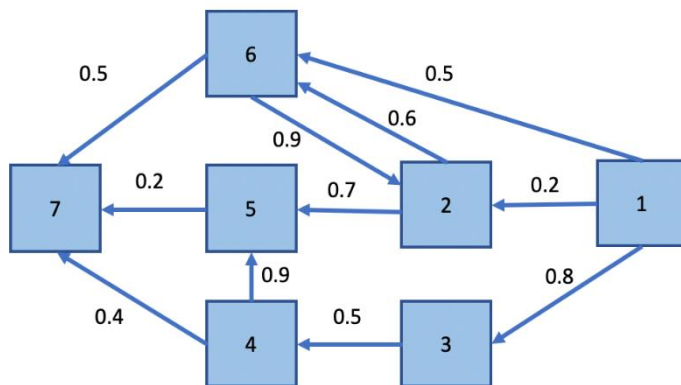
### 範例二

輸入格式：參考右上圖畫面，當滑鼠點選**起點 S**而移動或設定到新位置後。

輸出格式：系統自動隨著再從**起點 S**到**終點 T**且避開一些**路障點**而重新規劃出一條最短的路線，結果為此路線存在則顯示『**規劃路線成功**』，如上圖右方所示。

### 試題五：大地遊戲闖關 (17 分)

說明：小青擔任 T 大迎新宿營大地遊戲 A 小隊隊長，此大地遊戲中，設有許多遊戲關卡，然而關卡和路途嚴峻，可能會遇到無法通行的情況。



上圖是某一大地遊戲 7 個關卡的分布情況，遊戲關卡使用不同數字(其值大於 1 並小於 30)標記，而關卡與關卡之間的數值(大於 0.0)，代表路途上險峻的程度，每條路線僅能通行一次，箭頭代表通行方向，每一個箭頭代表一條路線，例如關卡 6 到關卡 2 之間有兩條路線。

在給定**起始關卡**和**終點關卡**後，身為小隊長的小青，要評估如何走那些路線?才能夠順利且最快的闖關成功。

**例 1**:從關卡 1(起始)到關卡 4(終點)只有一條路線(1→3→4)，所以可以順利完成闖關(至少有一條路線可以連結起始關卡和終點關卡)，此路途險峻程度為  $0.8 + 0.5 = 1.3$ 。

**例 2**:從關卡 1(起始)到關卡 2(終點)則有兩種路線：(1→6→2)，其路途險峻程度為  $0.5 + 0.6 = 1.1$ ，以及(1→2)，其路途險峻程度為 0.2；小青隊長為了順利且最快的闖關成功，所以，她要選關卡 1 直接到關卡 2，減少路途險峻進而順利闖關成功。

例 3:從關卡 1(起始)到關卡 7(終點)則有七種路線，其中線路 1->6->7，路途險峻程度 1 為最低，所以，小青會選此路線，而不選其他六條路線。

請你(妳)幫小青寫一程式，幫助小青做明智路線抉擇，才能夠順利且最快的闖關成功。

大地遊戲關卡文字檔：

在輸入的第 1 列(Row)為大地遊戲關卡的數目 X；接下來第 2 列到第 X+1 列代表一個 X\*X 陣列，陣列的第 m 列(Row) (即輸入的第 m+1 列)的每一數值代表某一編號為 m 的關卡到其它關卡 (1 至 X) 間存有一條路線直接連接它們並可通行的險峻程度；輸入檔的最後一列 (第 X+2 列) 有兩個不同的數值，分別由左至右代表大地遊戲的起始關卡與終點關卡。任一兩數字之間都用一個空白隔開。以上圖為例，輸入的大地遊戲關卡文字檔內容如下：

```
7
0.0 0.2 0.8 0.0 0.0 0.5 0.0
0.0 0.0 0.0 0.0 0.7 0.6 0.0
0.0 0.0 0.0 0.5 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.9 0.0 0.4
0.0 0.0 0.0 0.0 0.0 0.0 0.2
0.0 0.9 0.0 0.0 0.0 0.0 0.5
0.0 0.0 0.0 0.0 0.0 0.0 0.0
1 4
```

此大地遊戲關卡文字檔內容第一列為 7，表示大地遊戲關卡的數目為 7，接下來第 2 列到第 8 列代表一個 7\*7 陣列，陣列的第 1 列(Row)和第 2 行(Column)的值為 0.2，這表示關卡 1 到關卡 2 路途險峻程度為 0.2；輸入檔的最後一列 (第 9 列) 有兩個不同的數值，分別表示大地遊戲的起始關卡 1 與終點關卡 4。

程式功能要求：

請利用上述說明和範例，寫一個程式，能完成以下功能要求:

- (1) 能讓小青選擇輸入大地遊戲關卡文字檔(1<遊戲關卡數目<30)。
- (2) 能將小青所輸入的大地遊戲關卡文字檔內容顯示出來。
- (3) 能顯示最快的闖關路線，並顯示路途險峻程度。

程式執行例 1：

程式執行例 2：

<pre>請輸入大地遊戲關卡文字檔檔名: ex1.txt 你輸入檔名為 'ex1.txt' 大地遊戲關卡文字檔內容為: 7 0.0 0.2 0.8 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.7 0.6 0.0 0.0 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.9 0.0 0.4 0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.0 0.9 0.0 0.0 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1 4 最快的闖關路線 [1 -&gt; 4]: 1-&gt;3-&gt;4 (路途險峻程度 1.2)</pre>	<pre>請輸入大地遊戲關卡文字檔檔名: ex2.txt 你輸入檔名為 'ex2.txt' 大地遊戲關卡文字檔內容為: 7 0.0 0.2 0.8 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.7 0.6 0.0 0.0 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.9 0.0 0.4 0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.0 0.9 0.0 0.0 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1 2 最快的闖關路線 [1 -&gt; 2]: 1-&gt;2 (路途險峻程度 0.2)</pre>
---	---

程式執行例 3：

```

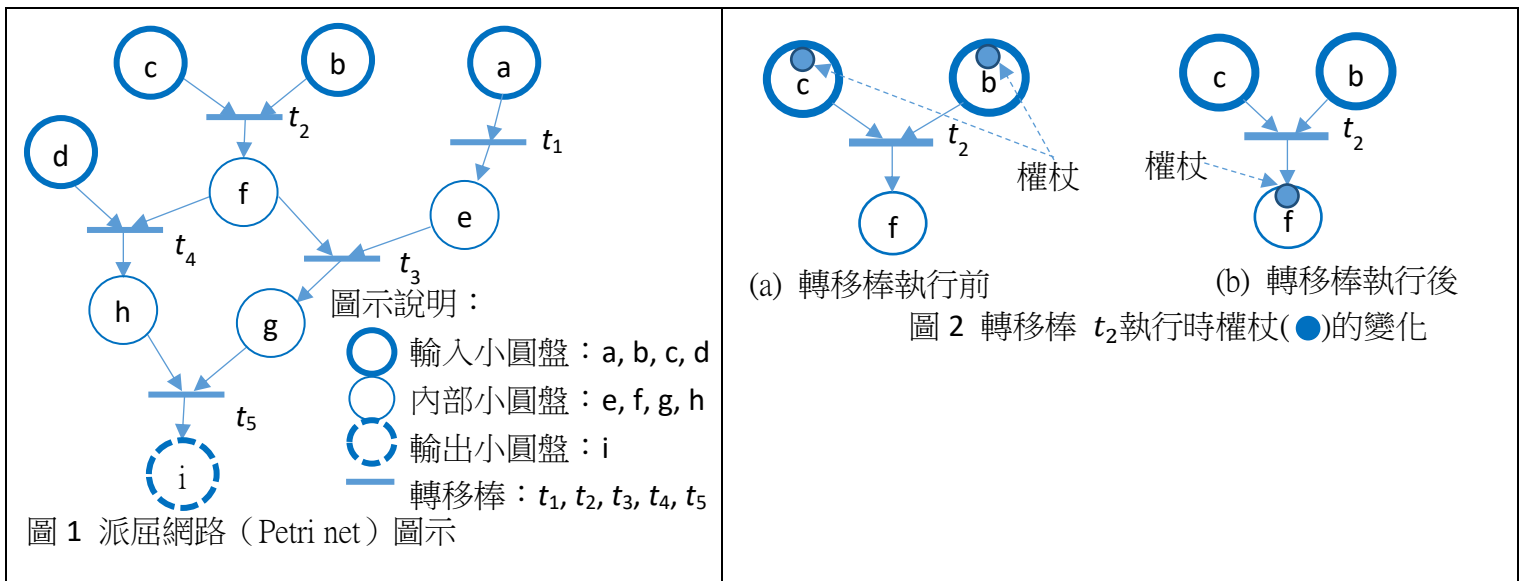
請輸入大地遊戲關卡文字檔檔名: ex3.txt
你輸入檔名為 'ex3.txt'
大地遊戲關卡文字檔內容為:
7
0.0 0.2 0.8 0.0 0.0 0.5 0.0
0.0 0.0 0.0 0.0 0.7 0.6 0.0
0.0 0.0 0.0 0.5 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.9 0.0 0.4
0.0 0.0 0.0 0.0 0.0 0.0 0.2
0.0 0.9 0.0 0.0 0.0 0.0 0.5
0.0 0.0 0.0 0.0 0.0 0.0 0.0
1 7
最快的闖關路線 [1 -> 7]: 1->6->7 (路途險峻程度 1)

```

若妳(你)的程式都完成上述功能和要求，才可以上傳要求檢查功能。

#### 試題六：派屈網路的模擬(17 分)

派屈網路 (Petri net) 被用來模擬操作程序，例如：交通流程、工業製程等。在此，把它簡化如圖 1。它的表示由小圓盤及轉移棒組成。小圓盤分為三類：輸入、內部及輸出等。「輸入小圓盤」只出現在轉移棒的輸入端，「輸出小圓盤」只出現在轉移棒的輸出端，而「內部小圓盤」則在轉移棒的輸入端及輸出端均有，如圖 1。轉移棒有「1-1 轉移棒」及「2-1 轉移棒」，如圖 1 的  $t_1$  及  $t_2$ 。轉移棒在被執行時，小圓盤被放入權杖或被移除權杖；而轉移棒是否執行？端視其輸入端的所有小圓盤是否都被放入權杖？若是，則該轉移棒將被執行，執行時，其輸入端的所有小圓盤的權杖全都被移除，且其輸出端的小圓盤被放入權杖，如圖 2。注意：若輸出端小圓盤的權杖尚未移除，則該轉移棒將不執行。每一個小圓盤只能放入一個權杖，被移除後就空了，只能再等其上之轉移棒滿足條件執行時再放入。之後，依序檢查各個轉移棒是否符合執行條件而予以執行。而且每次只在 1 個「輸入小圓盤」放入權杖，接著執行該派屈網路，可一直重複執行。權杖在圖示時以小黑點表示，如圖 2；在程式執行時，小圓盤有權杖時可用 1 表示，無則以 0 表示之。而我們也可以隨時查看各個小圓盤擺放權杖的情況。為了簡化複雜度，在此限制每個轉移棒的輸入端(上方)連接一個(例如：圖 1 的  $t_1$ )或二個小圓盤(例如：圖 1 的  $t_2$ )，轉移棒的輸出端(下方)只能連接一個小圓盤；小圓盤的上方連接一個轉移棒，小圓盤的下方連接一個或二個轉移棒(例如：圖 1 的  $f$ )。此外，網路不可以形成迴圈。





模擬派屈網路

第一階段：建構資料結構

- 1. (1)鍵入「輸入小圓盤」；(2)鍵入「內部小圓盤」；(3)鍵入「輸出小圓盤」
- 2. (1)鍵入「2-1 轉移棒」； (2)鍵入「1-1 轉移棒」
- 3. 印出資料結構：轉移棒與小圓盤的關係，小圓盤與轉移棒的關係

第二階段：轉移棒的執行，每次只在 1 個「輸入小圓盤」放入權杖，便執行該派屈網路，可一直重複執行

- 1. 將權杖鍵入「輸入小圓盤」的名稱，例如：a。每次只鍵入 1 個，並查看各個小圓盤權杖的情況
- 2. 執行轉移棒，並查看執行後各個小圓盤權杖的情況
- 3. 重複執行前述步驟 1 及 2

例 1：派屈網路如圖 1，執行過程如圖 3。

第一階段：建構資料結構

鍵入「輸入小圓盤」的數目及其名稱:  
4 a b c d  
鍵入「內部小圓盤」的數目及其名稱:  
4 e f g h  
鍵入「輸出小圓盤」的數目及其名稱:  
1 i  
鍵入「2-1轉移棒」的名稱及小圓盤的名稱:  
t2 b c f  
Continue?(1/0):1  
t3 e f g  
Continue?(1/0):1  
t4 d f h  
Continue?(1/0):1  
t5 g h i  
Continue?(1/0):0  
鍵入「1-1轉移棒」的名稱及小圓盤的名稱:  
t1 a e  
Continue?(1/0):0  
轉移棒與小圓盤的關係:t2: b c f t3: e f g t4: d f h t5: g h i t1: a e  
小圓盤與轉移棒的關係:  
a: t1 b: t2 c: t2 d: t4 e: t1 t3 f: t2 t3 t4 g: t3 t5 h: t4 t5 i: t5

圖 3a 在例 1 中建構圖 1 的資料結構

第二階段：轉移棒的執行

鍵入將放權杖的小圓盤名稱 a  
查看各個小圓盤權杖的情況:a:1 b:0 c:0 d:0 e:0 f:0 g:0 h:0 i:0  
執行轉移棒.  
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:1 f:0 g:0 h:0 i:0  
Continue?(1/0):1  
鍵入將放權杖的小圓盤名稱 b  
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:0 e:1 f:0 g:0 h:0 i:0  
執行轉移棒.  
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:0 e:1 f:0 g:0 h:0 i:0  
Continue?(1/0):1  
鍵入將放權杖的小圓盤名稱 c  
查看各個小圓盤權杖的情況:a:0 b:1 c:1 d:0 e:1 f:0 g:0 h:0 i:0  
執行轉移棒.  
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:0 f:0 g:1 h:0 i:0  
Continue?(1/0):1

```

鍵入將放權杖的小圓盤名稱 d
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:1 e:0 f:0 g:1 h:0 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:1 e:0 f:0 g:1 h:0 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 b
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:1 e:0 f:0 g:1 h:0 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:1 e:0 f:0 g:1 h:0 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 c
查看各個小圓盤權杖的情況:a:0 b:1 c:1 d:1 e:0 f:0 g:1 h:0 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:0 f:0 g:0 h:0 i:1
Continue?(1/0):.

```

圖 3b 例 1 之圖 1 轉移棒的執行

例 2：派屈網路如圖 1，執行過程如圖 4。

第一階段：建構資料結構

同圖 3a。

第二階段：轉移棒的執行

```

鍵入將放權杖的小圓盤名稱 b
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:0 e:0 f:0 g:0 h:0 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:0 e:0 f:0 g:0 h:0 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 c
查看各個小圓盤權杖的情況:a:0 b:1 c:1 d:0 e:0 f:0 g:0 h:0 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:0 f:1 g:0 h:0 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 d
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:1 e:0 f:1 g:0 h:0 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:0 f:0 g:0 h:1 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 a
查看各個小圓盤權杖的情況:a:1 b:0 c:0 d:0 e:0 f:0 g:0 h:1 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:1 f:0 g:0 h:1 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 b
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:0 e:1 f:0 g:0 h:1 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:1 c:0 d:0 e:1 f:0 g:0 h:1 i:0
Continue?(1/0):1
鍵入將放權杖的小圓盤名稱 c
查看各個小圓盤權杖的情況:a:0 b:1 c:1 d:0 e:1 f:0 g:0 h:1 i:0
執行轉移棒.
查看各個小圓盤權杖的情況:a:0 b:0 c:0 d:0 e:0 f:0 g:0 h:0 i:1
Continue?(1/0):.

```

圖 4 例 2 之圖 1 轉移棒的執行