

工作桌編號 \_\_\_\_\_ 選手姓名 \_\_\_\_\_ 代表學校 \_\_\_\_\_ 總分 \_\_\_\_\_

試卷說明：

1. 請將寫好之程式原始檔依題號命名存檔，第一題取：選手姓名\_Q1，第二題取：選手姓名\_Q2，依序命名存檔，並存於 C 碟之選手姓名\_Contest 目錄。
2. 競賽時間 4 小時。

**試題一：利用最小平方誤差法求穿過任意三點之直線方程式(25 分)：**

**說明：**根據最小平方誤差(least square error)理論，穿過  $n$  個點的直線方程式  $y=mx+c$ ，其中的斜率  $m$  與截距  $c$ ，其解寫成矩陣形式可表示成

$$\begin{bmatrix} m \\ c \end{bmatrix} = (A^T A)^{-1} A^T K, \text{ 其中 } A^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \text{ 且 } K = [y_1 \quad y_2 \quad \dots \quad y_n]^T$$

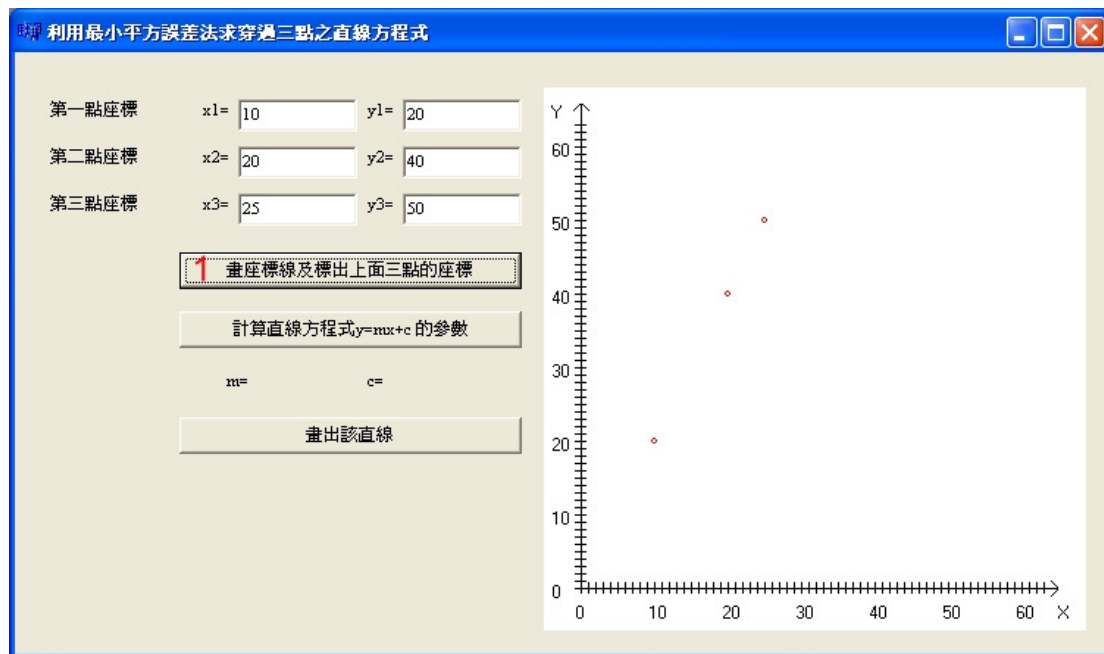
為了簡化我們的問題，假設輸入只有三個點，分別是  $(x_1, y_1)$ ,  $(x_2, y_2)$ ，及  $(x_3, y_3)$ ，則斜率  $m$  與截距  $c$  的解可寫成

$$\left( \sum_{i=1}^3 x_i = x_1 + x_2 + x_3 \right)$$

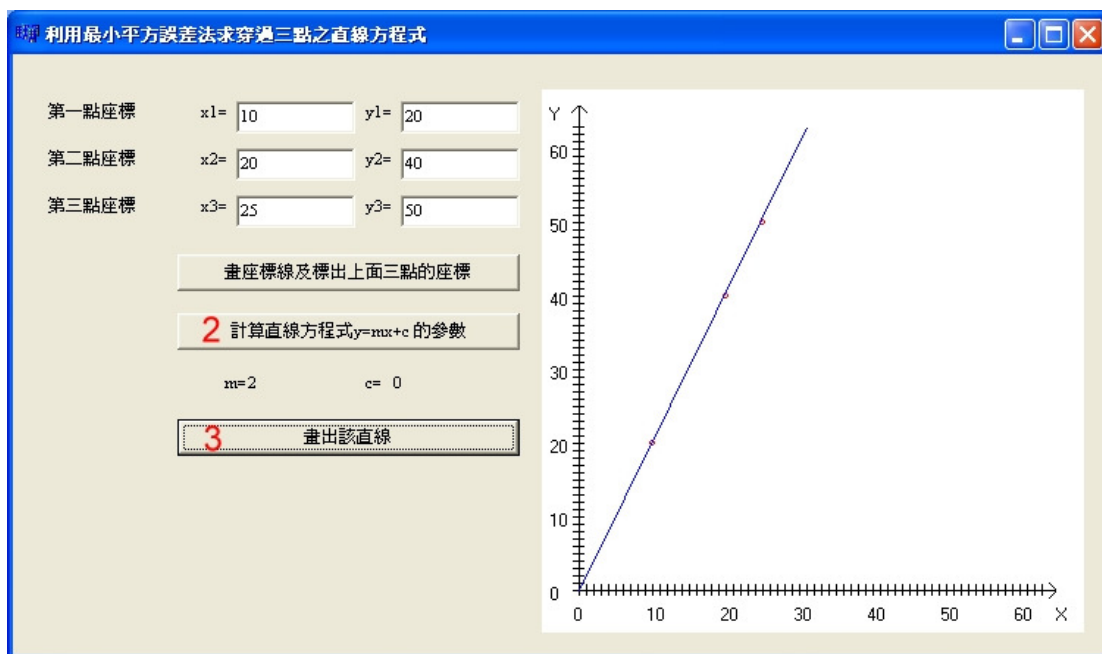
$$m = \frac{3\left(\sum_{i=1}^3 x_i y_i\right) - \left(\sum_{i=1}^3 x_i\right)\left(\sum_{i=1}^3 y_i\right)}{3\left(\sum_{i=1}^3 x_i^2\right) - \left(\sum_{i=1}^3 x_i\right)^2}, \quad c = \frac{\left(\sum_{i=1}^3 x_i^2\right)\left(\sum_{i=1}^3 y_i\right) - \left(\sum_{i=1}^3 x_i\right)\left(\sum_{i=1}^3 x_i y_i\right)}{3\left(\sum_{i=1}^3 x_i^2\right) - \left(\sum_{i=1}^3 x_i\right)^2}$$

試設計一個程式，可用來計算該直線方程式，並且畫出其圖形。

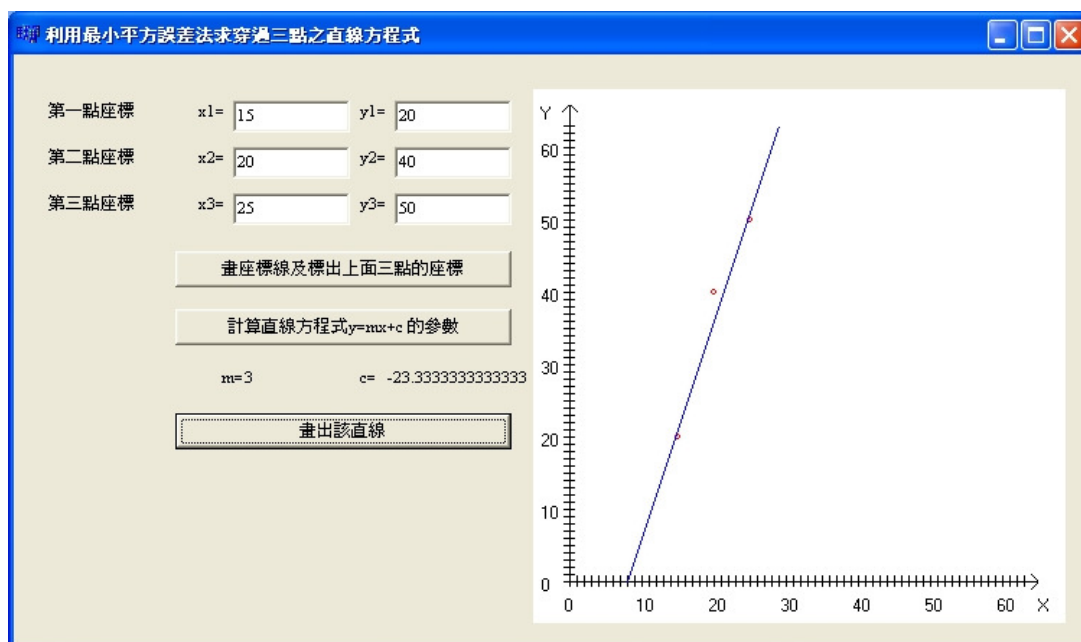
1. 假設三個點的座標，其值  $(x_1, y_1)$ ,  $(x_2, y_2)$ ，及  $(x_3, y_3)$  皆落在  $(0,0)-(63,63)$  之間，超過該範圍則不處理，則當按下"1"畫座標線及標出上面三點的座標鈕時，可以顯示如圖一右邊之圖形。(10 分)(註：請採用卡式座標系統，原點  $(0,0)$  的座標值在左下角)
2. 如圖二所示，按下"2"計算直線方程式  $y=mx+c$  的參數鈕時，可計算出正確的斜率  $m$  與截距  $c$  值。(10 分)
3. 如圖二所示，按下"3"畫出該直線鈕時，可畫出該直線。(5 分)
4. 圖三所示為另外一組座標值所得到的結果，可作為驗證程式是否正確之用。



圖一



圖二



圖三

試題二：資訊隱藏在 BMP 圖片中之程式設計( 25 分)：

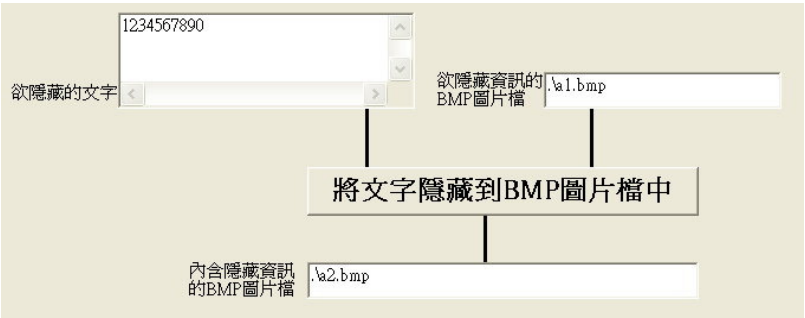
說明：普遍被使用的 BMP (Bit Mapped)圖形檔，是 Microsoft 公司爲了 Windows 所發展的一種影像檔案。其中全彩的格式，每一個影像點(Pixel)是由 B(blue)、G(green)、R(red)共 3Bytes 所組成，每 Byte 的 bit 0 的值影響影像點最不顯著。根據格式，圖檔的第 64Byte 起存放影像點的資料。因此可以把資訊隱藏在 BMP 圖片檔中第 64Byte 起的 bit 0 中，這樣的改變一般眼睛是無法查覺出來。

因此本程式設定以下規格：

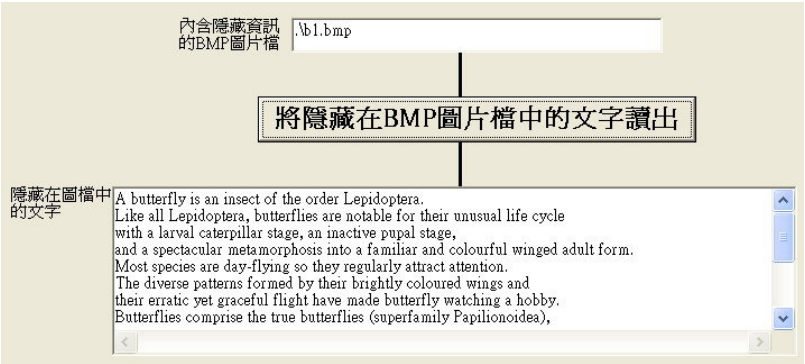
BMP 圖片檔中第 wx~yz Byte 的 bit 0	說明	例如
BMP 圖片檔中第 64Byte 到第 95Byte 的 bit 0 共 32bits	依序存放 0、1、0、1、 …、0、1、0、1 共 32bits，代表該圖片檔有隱藏資訊	第 64Byte 的 bit 0 爲 0、第 65Byte 的 bit 0 爲 1、第 66Byte 的 bit 0 爲 0、第 67Byte 的 bit 0 爲 1、……
BMP 圖片檔中第 96Byte 到第 111Byte 的 bit 0 共 16bits	存放所隱藏資訊的長度，第 96Byte 的 bit 0 爲最不顯著位元(LSB)	若隱藏資訊的長度爲 5Bytes、則第 96Byte 的 bit 0 爲 1、第 97Byte 的 bit 0 爲 0、第 98Byte 的 bit 0 爲 1、第 99~111Byte 的 bit 0 均爲 0
BMP 圖片檔中第 112Byte 起的 bit 0	存放欲隱藏的資訊	若隱藏資訊爲 “A” 1 byte，其 ASCII 值用二進制表示成 01000001 第 112Byte 的 bit 0 爲 1、第 113Byte 的 bit 0 爲 0、第 114Byte 的 bit 0 爲 0、第 115Byte 的 bit 0 爲 0、第 116Byte 的 bit 0 爲 0、第 117Byte 的 bit 0 爲 0、第 118Byte 的 bit 0 爲 1、第 119Byte 的 bit 0 爲 0、

請依上面規格設計一支程式

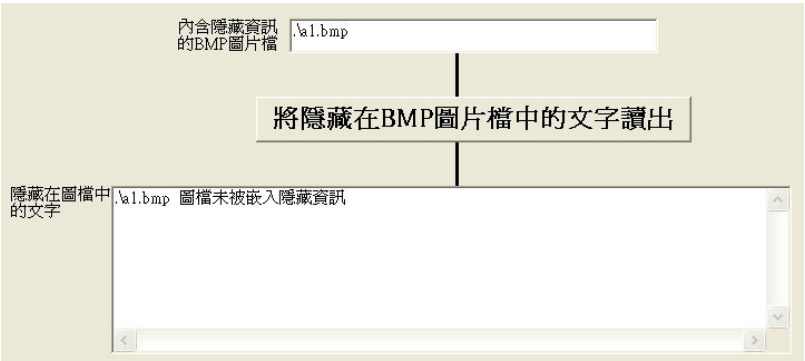
1. 能輸入“欲隱藏的文字”、“欲隱藏資訊的 BMP 圖片檔”檔名、“內含隱藏資訊的 BMP 圖片檔”檔名，及一個按鈕“將文字隱藏到 BMP 圖片檔中”。如右上圖所示。
- 說明：將字串“1234567890”嵌入 a1.bmp 檔，另存檔成 a2.bmp 檔。



2. 能輸入“內含隱藏資訊的 BMP 圖片檔”檔名、顯示“隱藏在圖檔中的文字”、及一個按鈕“將隱藏在 BMP 圖片檔中的文字讀出”，如右中圖所示。



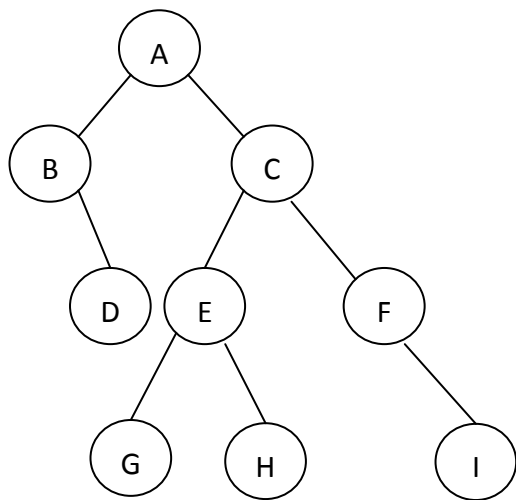
3. 若輸入的 BMP 圖片檔內未含隱藏資訊，若按“將隱藏在 BMP 圖片檔中的文字讀出”按鈕，則將出現該圖檔未被嵌入隱藏資訊，如右下圖所示。



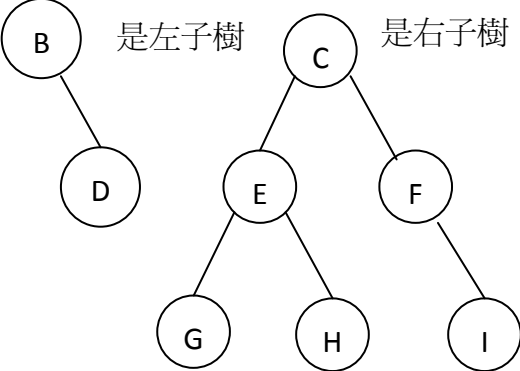
註：a1.bmp 檔未被嵌入任何資訊，  
b1.bmp 檔被嵌入 1025Bytes 文字

**試題三： 繪二元樹( 25 分)：**

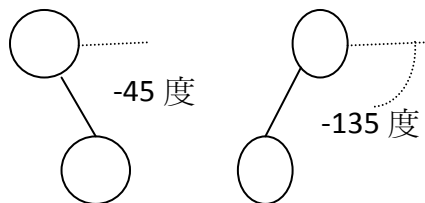
**說明：**一個二元樹是由一個節點集合所構成，這個集合可能是一個空集合，或者裏面有一個節點元素稱為根，然後它連接兩個二元樹（分別稱為左子樹及右子樹）。下面是一個二元樹的例子（其中 B，C 是 A 的子節點，而 A 是 B，C 的親代節點）：



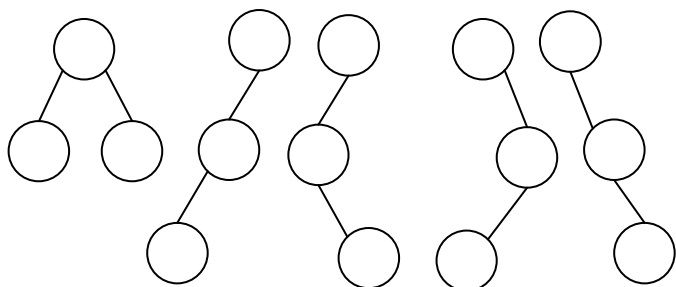
其中  是根， 是左子樹  是右子樹。



給定節點的個數，我們可以產生不同形狀的二元樹。譬如：0 或 1 個節點都只有一種二元樹。2 個節點的形狀有如下：

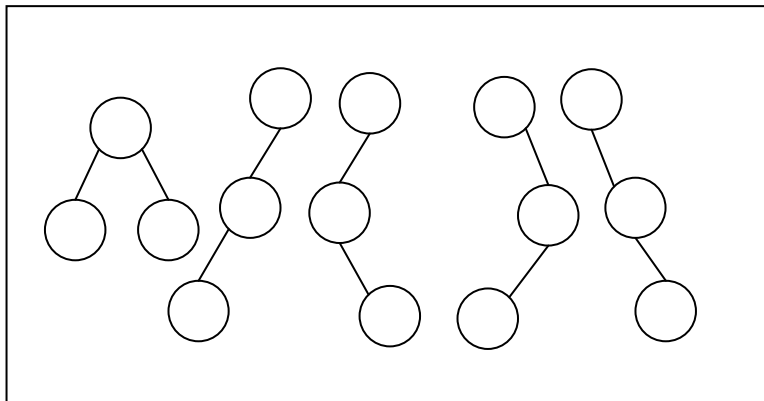


3 個節點的形狀有 5 種如下所示：



範例-- 系統輸入：N=3

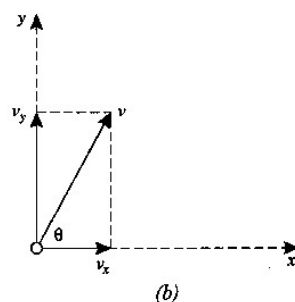
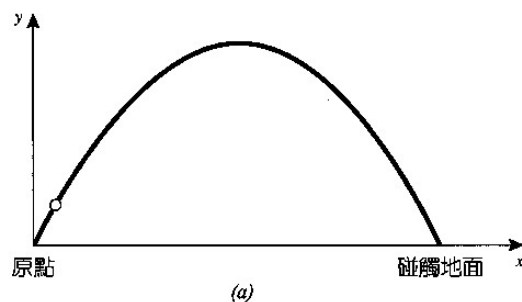
輸出參考畫面如下：



請寫一個程式，輸入  $N$  ( $N = 4$ )，輸出  $M$  個所有形狀的二元樹 ( $M$  是  $N$  個節點的二元樹不同形狀總數)。  
 評分：每對一個二元樹，則得 2 分，最高為 25 分。

#### 試題四：球的飛行

**說明：**在物理中，假如忽略空氣摩擦和地球彎曲的表面，將一顆球從地面拋向空中，則其將依照拋物線的軌跡飛行（如下圖(a)所示）。



在任何時間  $t$ ，球的高度，可以由方程式(1)求出：

$$y(t) = y_0 + v_{y0}t + \frac{1}{2}gt^2 \quad (1)$$

其中  $y_0$  是物體在地面上的初始高度， $v_{y0}$  是物體的初始垂直速度，而  $g$  是地球重力加速度 ( $-9.81 \text{ m/s}^2$ )。當球被拋到空中時，其飛行的水平距離可以時間的函數來表示，如方程式(2)所示：

$$x(t) = x_0 + v_{x0}t \quad (2)$$

其中  $x_0$  是球在地面的初始水平位置，而  $v_{x0}$  是球的初始水平速度。

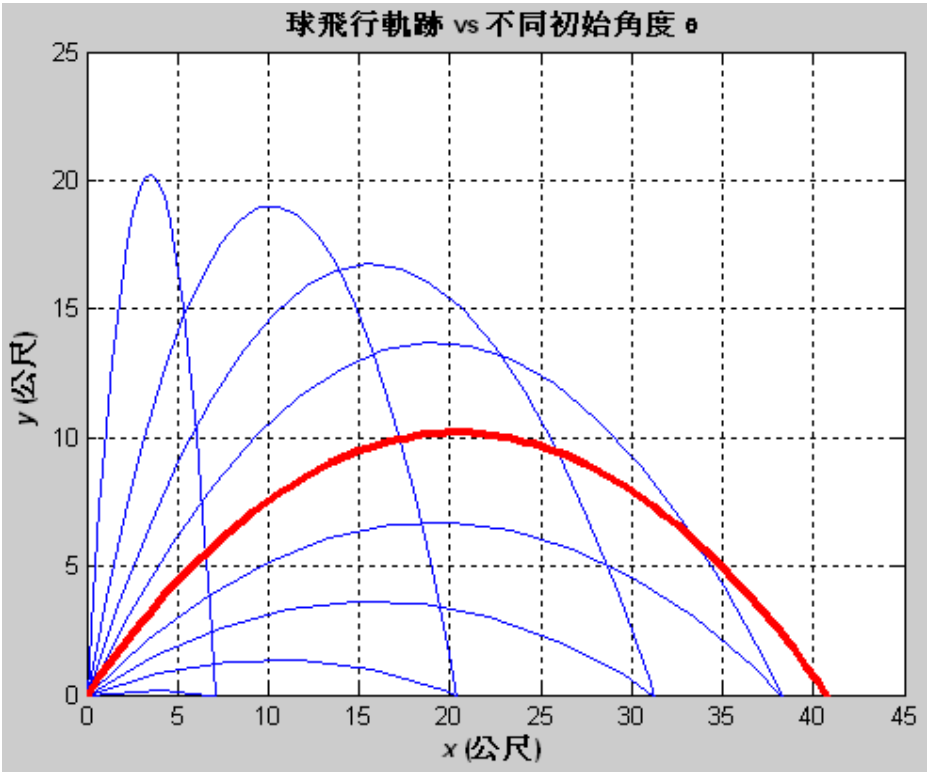
如果將球以相對於地面的角度  $\theta$  與初始速度  $v_0$  拋出，則其初始速度的水平分量和垂直分量分別為（如上圖(b)所示）：

$$v_{x0} = v_0 \cos \theta \quad (3)$$

$$v_{y0} = v_0 \sin \theta \quad (4)$$

假設球以初始角度 $\theta$ 和每秒 20 公尺的初始速度  $v_0$ ，從 $(x_0, y_0) = (0, 0)$  的位置拋出。請寫一個能畫出球飛行軌跡的程式，並且求出球在碰觸地面前，所飛行的水平距離。這個程式必須繪出不同初始角度下，球的飛行軌跡（ $\theta$ 從 5 度變化到 85 度，以 10 度為增量）；也必須計算不同初始角度下的水平飛行距離（ $\theta$ 從 0 度變化到 90 度，以 1 度為增量）。最後，這個程式必須求出最大水平飛行距離時的角度 $\theta$ ，並利用不同顏色的厚實線，來畫出此角度 $\theta$ 的飛行軌道。

執行範例：



角度	水平距離	70	26.2095
0	0.0000	71	25.1034
1	1.4230	72	23.9668
2	2.8443	73	22.8009
3	4.2621	74	21.6073
4	5.6747	75	20.3874
5	7.0805	76	19.1426
6	8.4775	77	17.8745
7	9.8643	78	16.5846
8	11.2390	79	15.2745
9	12.6001	80	13.9458
10	13.9458	81	12.6001
11	15.2745	82	11.2390
12	16.5846	83	9.8643
13	17.8745	84	8.4775
14	19.1426	85	7.0805
15	20.3874	86	5.6747
16	21.6073	87	4.2621
17	22.8009	88	2.8443
18	23.9668	89	1.4230
19	25.1034	90	0.0000
20	26.2095	...	
		最大水平飛行距離是 40.7747 at 45 度.	

**評分：**

1. 程式介面 (2.5 分)。
2. 繪出不同初始角度下，球的飛行軌跡（ $\theta$ 從 5 度變化到 85 度，以 10 度為增量）(10 分)。
3. 計算不同初始角度下的水平飛行距離（ $\theta$ 從 0 度變化到 90 度，以 1 度為增量）(5 分)。
4. 必須求出最大水平飛行距離時的角度 $\theta$ ，並利用不同顏色的厚實線，來畫出此角度 $\theta$ 的飛行軌道(7.5 分)。