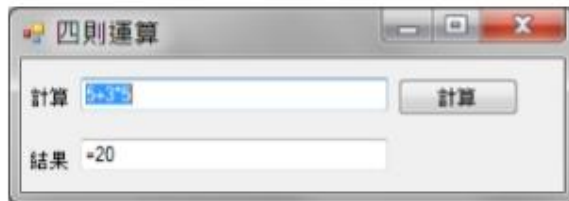


試題二、四則運算器(17 分)

設計一個四則運算器，可以輸入四則運算式並計算結果，要符合（）內優先計算及先乘除後加減的原則

不用考慮大數計算，如果結果為循環小數的話，四捨五入取到小數第四位

畫面設計：



輸入範例 1： $5+3*5$

輸出範例 1：20

輸入範例 2： $(5+3)*5$

輸出範例 2：40

輸入範例 3： $1/2+1/4$

輸出範例 3：0.75

輸入範例 4： $1/7$

輸出範例 4：0.1429

試題一、貓抓老鼠 (17 分)

有一隻很懶惰的貓，想要在最省力的情況下，捕捉到所有老鼠，請計算貓咪該如何在最省步數的情況下，捕捉到所有的老鼠。

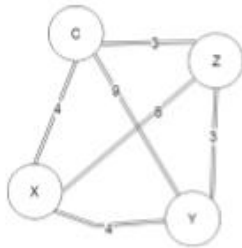
輸入說明

- 貓咪只會上下左右移動
- 0 代表可以行走的路，1 代表不能行走，C 代表貓咪，XYZ 代表老鼠， 同時間可能存在 1-3 個老鼠，老鼠被捕抓後的格子才能行走，整體範圍為 $M \times N$ ， $0 < M \leq 20$ ， $0 < N \leq 30$ 。

輸入範例：

```
00000100
0C000100
0110Z000
01110000
0X100000
001Y0000
00000000
```

將貓跟老鼠的距離列出



可以得到最短的走法是 $C \rightarrow Z \rightarrow Y \rightarrow X$

輸出範例：

3Z3Y4X

輸出說明：

- 依序輸出步數和目標
- 有其中一個獵物走不到的話，就輸出無解
- 3Z3Y4X 代表走3步到Z、再走3步到Y、再走4步到X

輸入範例 2：

0000Y100

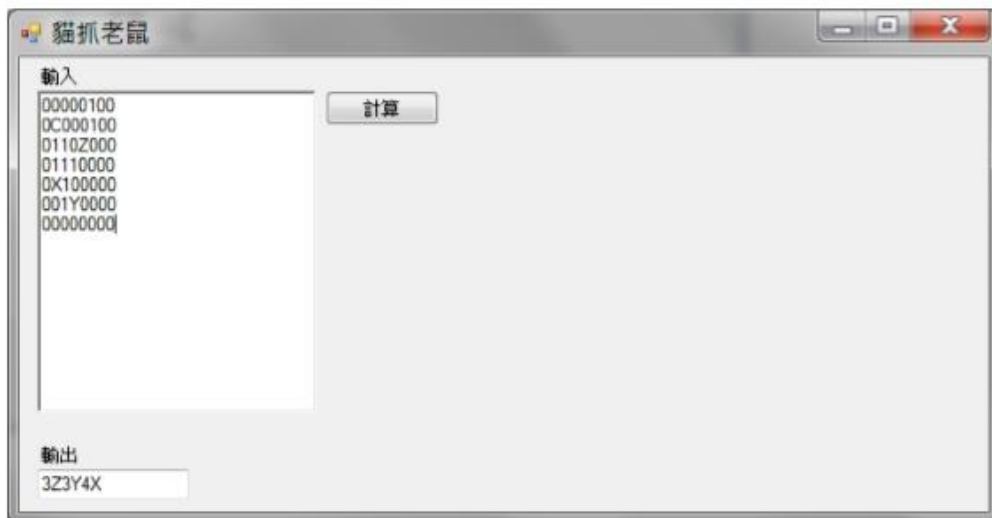
0C011100

0111X000

輸出範例2：

無解

畫面設計：



試題二、圖像隱碼術(Steganography) (17 分)

一個 24 位元的點陣圖中的每個像素都有三個顏色(紅綠藍)，各使用 8 個位元來表示。如果我們只考慮紅色的話，就有 255 種不同的數值來表示深淺不同的紅色。而像 11111111 和 11111110 這兩個值所表示的紅色，人眼幾乎無法區分。因此，這個最低有效位(LSB)就可以用來存儲顏色之外的訊息，而且在某種程度上幾乎是檢測不到的。

我們可以利用紅色像素和綠色像素來作訊息傳遞，紅色像素代表是否有資料，綠色像素代表資料內容，紅色像素的 LSB 如果為 1 代表綠色像素 LSB 是資料，如果紅色像素的 LSB 是 0，代表已經沒有資料，可以中斷取樣，像素取樣的順序由左而右，由上而下。

以下是範例的前九個像素資料，(X 代表 0 或 1 均可)

| 紅色像素 | 綠色像素 | 藍色像素 |
|---------|---------|---------|
| XXXXXX1 | XXXXXX0 | XXXXXXX |
| XXXXXX1 | XXXXXX1 | XXXXXXX |
| XXXXXX1 | XXXXXX0 | XXXXXXX |
| XXXXXX1 | XXXXXX0 | XXXXXXX |
| XXXXXX1 | XXXXXX0 | XXXXXXX |
| XXXXXX1 | XXXXXX0 | XXXXXXX |
| XXXXXX1 | XXXXXX0 | XXXXXXX |
| XXXXXX1 | XXXXXX1 | XXXXXXX |
| XXXXXX0 | XXXXXXX | XXXXXXX |

取得資料 01000001，轉為文字後變得到 A

請設計一個程式可以進行圖像隱碼儲存和取得：

- 點擊選擇檔案載入，可以開啟系統對話框，選擇一個檔案後載入到下方畫面上
- 點擊儲存隱碼，會將下方文字方塊內容存入圖中，然後開啟系統對話框，選擇一個檔案位置儲存
- 點擊解讀隱碼，會將畫面上的圖片中含有的隱碼資訊顯示在下方文字方塊中

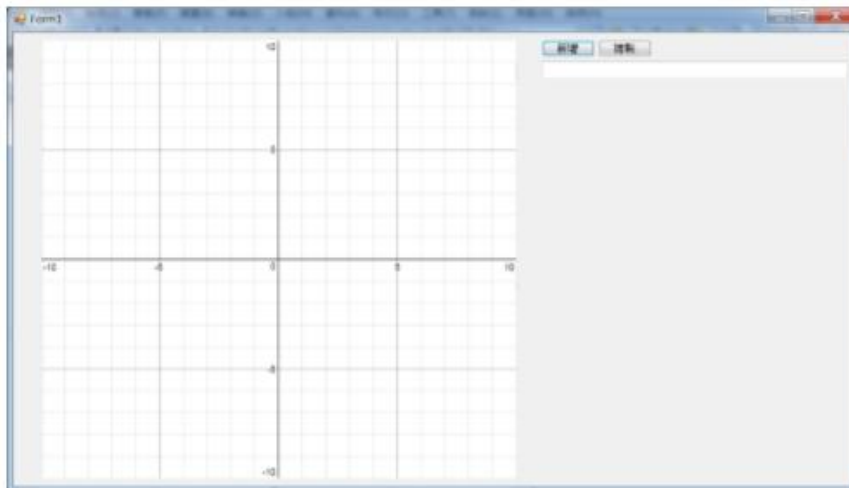
畫面設計：



試題三、數學函數繪圖器 (17 分)

請設計一個數學函數繪圖器，可以輸入多個函數繪製在同一個畫面上，每個函數會自動產生一個顏色，繪圖區域大小 10*10，每個函數可以輸入 實數 + - * / ^ sin cos x y =，其中 ^ 代表次方，畫面說明如下：

- 點擊新增，可以新增一個函數輸入框，最多可以輸入 20 個
- 點擊繪製，會將所有函數繪圖在左方繪圖區中
- 繪圖區需要有主格線(每 5)跟次格線(每 1)，並且標示 0 5 10 的數字

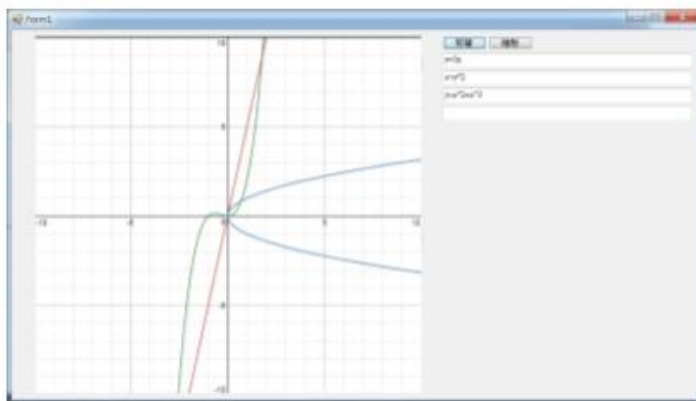


輸入範例 1：

$$y=5x$$

$$x=y^2$$

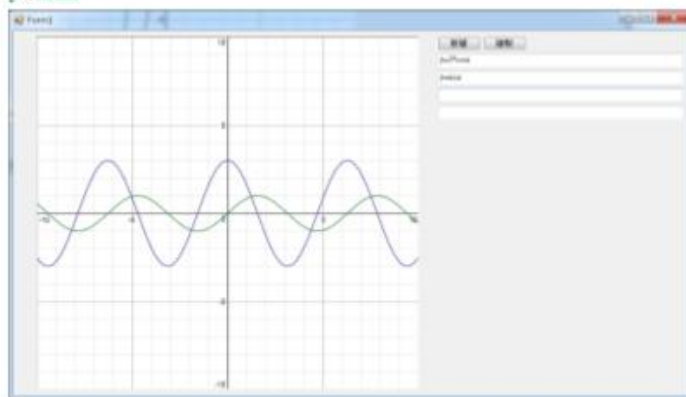
$$y=x^2+x^3$$



輸入範例 2 :

$$y=3*\cos x$$

$$y=\sin x$$



試題四：前序、後序、中序(17分)

說明：

平常所使用的運算式，主要是將運算元放在運算子的兩旁，例如 $a+b/d$ 這樣的式子，這稱之為中序 (Infix) 表示式，對於人類來說，這樣的式子很容易理解，但由於電腦執行指令時是有順序的，遇到中序表示式時，無法直接進行運算，而必須進一步判斷運算的先後順序，所以必須將中序表示式轉換為另一種表示方法。

三種表示方式如下：

- 1、中序式 (infix): 我們平常所看到的寫法，就是屬於中序式，把運算子放在兩個運算元的中間，如 $axb+cx d$
- 2、後序式 (postfix): 將運算子放在兩個運算元之後，因此 $axb+cx d$ 改寫為後序式為， $abxc d x +$
- 3、前序式 (prefix): 將運算子放在兩個運算元之前，因此 $axb+cx d$ 改寫為後序式為， $+xabxc d$

請設計一程式可以輸入任一表示方法後按下轉換可以轉成另外兩種表示方式。

畫面設計：

畫面設計如下面範例，輸入資料皆為正確的表示式，運算元與運算子的總字數不會超過 20。

範例 1：

(a) 初始畫面輸入中序資料

(b) 按下以此轉換，可以得到前序與後序的答案

範例 2：

(a) 初始畫面輸入前序資料

(b) 按下以此轉換，可以得到中序與後序的答案

試題五：排列組合(16 分)

說明：

在排列組合問題中將一組數字進行排列，可以得到不同的數字順序，例如 12 這個數的排列共有：
(1)12、(2)21 二組(由小到大排序)；例如 123 這個數的排列組合順序為：(1)123、(2)132、(3)213、
(4)231、(5)312、(6)321 六組(由小到大排序)；例如 1234 這數的排列組合有 24 組，數列順序如下：

(1)1234(2)1243(3)1324(4)1342(5)1423(6)1432(7)2134(8)2143(9)2314(10)2341(11)2413(12)2431
(13)3124(14)3142(15)3214(16)3241(17)3412(18)3421(19)4123(20)4132(21)4213(22)4231
(23)4312(24)4321

請輸入 3 份資料，第 1 份資料的數字 n 代表有幾筆資料要測試， $2 \leq n \leq 5$ ，第 2 份資料為每筆的測試資料，共有三個正整數。各個數字間以“,” 隔開。 i 的值為 12, 123, 1234, 12345, 123456 其中之一，而 j 和 k 代表 i 值排列組合順序(由小到大排序)的第 j 個和第 k 個值。 j 和 k 不會超出 i 的排列數，請輸出第 j 個和第 k 個值的總合。

[範例 1]

i, j, k : 12, 1, 2 這組測試資料中，12 這個數的排列組合有：(1)12、(2)21 二組(由小到大排序)，第 1 個值為 12；第 2 個值為 21；總合為 $12+21=33$ 。

[範例 2]

i, j, k : 123, 1, 2 這組測試資料中，123 這個數的排列組合有：(1)123、(2)132、(3)213、(4)231、
(5)312、(6)321 六組(由小到大排序)，第 1 個值為 123；第 2 個值為 132；總合為 $123+132=255$ 。

畫面設計：

畫面設計如下，輸入欲測試的資料筆數與資料後可以計算出加總結果。

