# Problem 1:

# 請寫一可製作萬年曆(Gregorian Solar Calendar)之程式。

1. **須可明顯區分出某一天為星期幾、周末日(weekend)及非周末日(weekdays)；**
2. **當使用者輸入西元年份時(例如2009)，須可顯示出以月份(Month)為單位的整年度月曆表；**
3. **當使用者輸入西元年份及月份時(例如2009/09)，須可顯示出自該月份起，連續三個月的月曆表。**

**Problem 2:** 爬梯子

**五位同學在宿舍趕作業到十點多時，有人提議買宵夜進來吃，大夥兒商議好買的東西與分擔的費用，但要由誰來出門採購呢？為了公平起見，甲乃拿起一張紙劃了五條等長平行線要大家用「爬梯子」的方式決定誰負責採購，方法是每人在這五條平行線間隨意劃上1至4條連接線與相鄰的平行線垂直連結，5個人都劃完後，將中間遮住，只露出平行線兩端，在一端的5個線頭中，任選一個劃上圓圈，另一端再由5人依次簽選一個線頭，最後攤開紙張，由圓圈處的平行線往上走，若碰到連接線就順著轉彎到鄰近的另一條，直到另一端看是誰簽選，就是他”中獎”！**

**試將此一「爬梯子」的方式撰寫成一個程式，初始的狀態如圖5-1，共有五條垂直平行線─A-I, B-J, C-K, D-L, E-M；垂直平行線兩兩之間各有八條水平虛線（共四組1~8編號），代表備選之連接線；每人輪流依序選取四個0~8數字，0代表該組不選，1~8代表圖中相對位置之虛線要劃成實線而產生轉彎的作用，若兩人重覆選取同一位置等同於只選一次；5個人都選完後，找出A, B, C, D, E「爬梯子」對應的是I, J, K, L, M之中那一個。**

**圖5-1 初始的狀態**

**範例：（及示意圖）**

**輸入：**

**0536**

**6332**

**2884**

**0666**

**2302**

**  
圖5-2範例輸入**

**輸出：**

1. **I**
2. **L**
3. **J**
4. **K**
5. **M**

**  
圖5-3 A對應的是I**

# Problem3: 中文數字大寫

**試撰寫一程式，如下範例，輸入最長13位正整數，即輸出其對應之「中文數字大寫」，程式可持續執行，直到輸入0時結束。**

## 範例(加底線者表示user自行輸入)

**Input:**

**1200357**

**Output:**

**壹佰貳拾萬零參佰伍拾柒**

**Input:**

**3000000100253**

**Output:**

**參兆零壹拾萬零貳佰伍拾參**

**Input:**

**700633000**

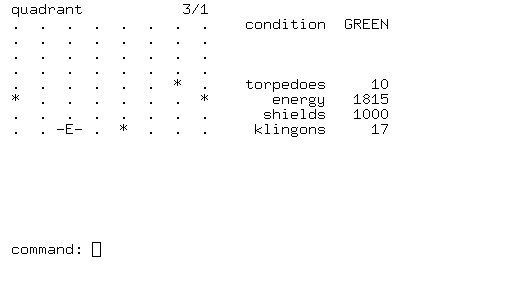
**Output:**

**柒億零陸拾參萬參仟**

**Problem 4:**

# Star Trek

**星際旅行(Star Trek)是民國66年左右，當時的微電腦流行的一個文字式電腦遊戲，遊戲者的任務是指揮企業號(Enterprise)追捕和摧毀來犯的克林貢(klingon)艦隊軍艦。**

**太空宇宙區分成許多象限(quadrant)，一個象限有8 x 8個小區，今企業號行進到某一象限時，如上圖所示，左上角部分為8 x 8象限圖，企業號在此象限的座標位置是C3R8（以”-E-“標示），雷達掃描顯示象限內有4艘klingon（以”\*”標示，座標位置由左而右是C1R6, C5R8, C7R5, C8R6），請設計程式摧毀象限內klingon，依序執行下列功能：**

1. **輸入企業號在此象限的座標位置（此例為 3, 8）**
2. **輸入此象限內klingon的個數（此例為 4）**
3. **輸入此象限內klingon的座標位置（此例為 1, 6; 5, 8; 7, 5; 8, 6）**
4. **輸入度度量整數值（0~360，代表發射魚雷(torpedo) 之順時鐘角度，此例若輸入90可摧毀C5R8的klingon）**
5. **顯示魚雷發射後之8 x 8象限圖，需以”@”標示魚雷路徑（路徑規則可自訂，但需合理），若魚雷命中klingon（標示”@”之座標會被蓋寫）則原標示”\*”變成”.”；若無命中，則標示路徑直到邊線處截止**
6. **若klingon已全部摧毀，則顯示訊息結束程式；否則回到步驟IV。**

**Problem 5:**

# 橋牌－牌色整理：

**橋牌所使用的52張撲克牌，共分紅心♥(Heart)、黑桃♠(Spade)、方塊♦(Diamond)、梅花♣(Club)四個花色，為了便利記錄，我們以英文各自開頭一個字母為代表，即紅心(Heart)為H，黑桃(Spade)為S，方塊(Diamond)為D，梅花(Club)為C；點數的大小順序是A-K-Q-J-T-9-8-7-6-5-4-3-2。一張牌的花色、點數即可以兩個英數字代表，如：**

**“SK” 代表 “黑桃 King”**

**“D3” 代表 “方塊3”**

**“HT” 代表 “紅心 10”**

**試撰寫一程式，如下範例，輸入13張牌的花色、點數，將之整理後輸出，花色依紅心、黑桃、方塊、梅花，順序逐行顯示，點數由大而小順序由左而右列出。（輸入、輸出都以一個空格作為分隔符號）**

**範例：**

## Input

**C1 H7 S5 S3 HA CQ CJ S9 HT C3 S2 H5 HK**

## Output

**<Heart> A K T 7 5**

**<Spade> 9 5 3 2**

**<Diamond>**

**<Club> Q J 3 1**

**Problem A**

**男女的安全排列問題**

**有m位男同學與n位女同學要排成一列，並依序每分鐘請一位同學進入原本為空的教室，若過程中任何時刻教室內的男同學人數均不超過女同學人數，則稱該男女同學的排列方式為『安全排列』(secure permutation)。**

**試寫出一個程式，由鍵盤分兩列讀入m與n，然後輸出所有的安全排列，並統計安全排列的總數。安全排列只區分『男』與『女』，不必仔細區分不同的男生為『男1』『男2』，也不必區分不同的女生。**

**注意：程式執行時間不得超過1分鐘。**

**Input格式：**

**由鍵盤輸入，共兩列，第1列為正整數m，第2列為正整數n。其中m與n均不超過6，且m小或等於n。**

**Output格式：**

**在螢幕分兩部分顯示。第一部份為所有的安全排列，每一種安全排列以一列顯示，以『男』『女』兩字分別代表男同學與女同學，排列順序由左而右，若共有x種安全排列，則此部分共佔x列。第二部分為安全排列總數的統計，若共有x種安全排列，則顯示『安全排列總數＝x』。**

**自我測試範例**

**Input:**

**2**

**2**

**Output:**

**女女男男**

**女男女男**

**安全排列總數=2**

**Problem B**

**均分飲料問題**

**有三個無刻度的杯子A, B, C容量分別為x, y, z毫升。起初A杯中有m毫升的飲料而B,C杯都是空的。我們希望透過一些操作步驟將飲料平分置入A,B杯中。操作時每一步驟均涉及兩個杯子，『來源杯』及『目的杯』，來源杯中必須有飲料，目的杯中不一定有飲料，『來源杯』之飲料倒入『目的杯』時，只允許兩種情形，一是『來源杯』被倒空了，二是『目的杯』被裝滿了。**

**試寫出一個程式，讀入最初的x, y, z, m的值，然後輸出所有的步驟(以三個杯子內的飲料量逐次變化來表示)，最後達成A,B杯各有m/2毫升而C杯為空的最後狀態。解答必須使用最少步驟才算正確；若10個步驟內無法完成，程式需輸出Error。**

**注意：程式執行時間不得超過1分鐘。**

**Input格式：**

**由鍵盤輸入四個正整數x, y, z, m，其間以空白字元隔開。這四個正整數均不超過1000。**

**Output格式：**

**若10個步驟內無法完成，程式只需輸出Error。若10步之內可完成，則在螢幕上第一列先顯示最初三個杯子中的飲料量，然後每經過一個步驟，三個杯子中的飲料量就顯示在下一行。每一行均為三個整數，依序代表A, B, C杯中的飲料量，三個整數必須以空白字元隔開。注意，必須使用最少步驟才算正確。**

**自我測試範例1:**

**Input:**

**800 500 300 300**

**Output:**

**Error!**

**自我測試範例2 :**

**Input:**

**800 500 300 600**

**Output:**

**600 0 0**

**300 0 300**

**300 300 0**

**Problem C**

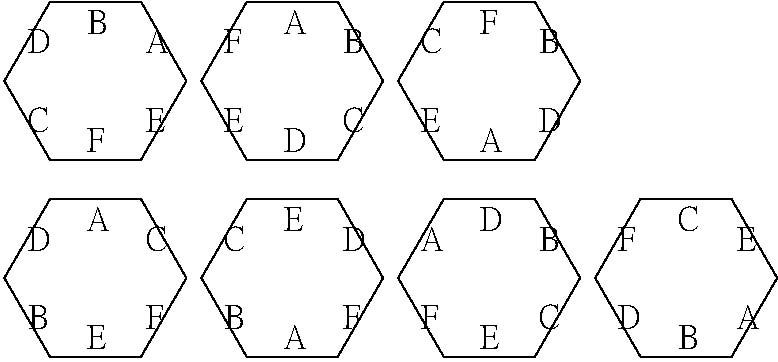
**六角拼圖問題**

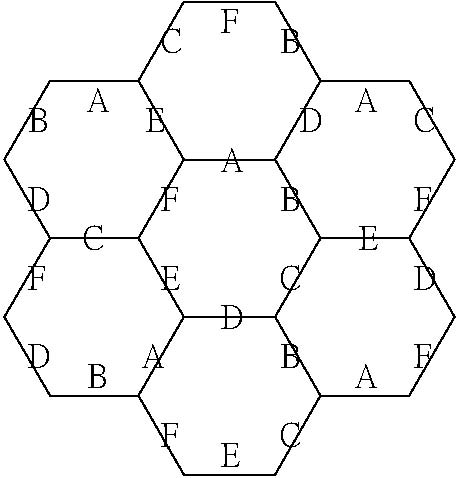
**一套『六角拼圖』是由7片六角形的厚紙片組成，每一片的每一邊都有標示一個英文符號。要解開六角拼圖問題，就是要想辦法如下圖所示將一塊紙片放在中間，另外六塊紙片圍繞在旁邊，而且讓每一對靠在一起的邊(來自不同紙片)都標示著同一種英文符號。拼圖時，每一個紙片都可以自由旋轉(順時針或逆時針)，但不能翻面(Flip)。**

**寫一個程式來讀入7片厚紙片邊上標示的英文符號，並輸出解答。若解答有好幾組，則必須依『字典順序』(alphabetical order)僅輸出最前面的一組；若無解，請輸出『No solution』。**

**注意：程式執行時間不可超過1分鐘。**

**範**

****

****

**Input格式:**

**由鍵盤(標準輸入)讀入七列。每一列包含6個英文字母，代表一個厚紙片上依順時針方向各邊的標示符號。**

**Output格式:**

**在螢幕(標準輸出)上顯示七列。每一列包含6個英文字母。第一列代表解答的中間紙片由正上方開始依順時針方向各邊的標示符號。第2列代表解答的最上方紙片由正上方開始依順時針方向各邊的標示符號。其他的紙片各邊的標示符號依此類推，分別輸出在第3至第7列 (請注意上面範例拼圖與下面自我測試範例1的對應)。注意：無解時，請輸出『No solution』；若解答有好幾組，則必須依『字典順序』(alphabetical order)僅輸出最前面的一組。(其實任何一組解答都可以經由旋轉變成另外一些『字典順序』不同的解答)**

**自我測試範例1:**

**Input:**

**BAEFCD**

**ABCDEF**

**FBDAEC**

**ACFEBD**

**EDFABC**

**DBCEFA**

**CEABDF**

**Output:**

**ABCDEF**

**FBDAEC**

**ACFEBD**

**EDFABC**

**DBCEFA**

**CEABDF**

**AEFCDB**

**自我測試範例2:**

**Input:**

**AAAAAA**

**BBBBBB**

**CCCCCC**

**DDDDDD**

**EEEEEE**

**FFFFFF**

**GGGGGG**

**Output:**

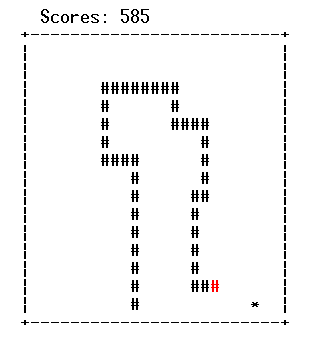
**No solution**

**簡單遊戲程式--貪食蛇(選自網路)**

rainbow

**SNAKE**

如下圖所示，這個作業是希望同學能夠練習製作有互動性的簡單遊戲程式：



這個文字界面的 snake 程式寫作起來複雜度不高，不要被 "遊戲" 嚇到了，其實沒什麼複雜的演算法，不過這個程式可以練習下列的程式技巧：

1. 各種流程控制： for, while, if, switch, 函式等等
2. 鍵盤互動界面
3. 文字模式繪圖控制
4. 檔案輸入輸出
5. typedef 敘述
6. 結構、指標
7. 動態記憶體配置
8. 串列
9. enum 敘述

[範例執行程式](http://squall.cs.ntou.edu.tw/cpp/92Spring/snake1.exe)

greenline

**請先自行構思你的程式，然後再參考下面的說明**

greenline

**如何以 C 語言製作這樣子的程式呢?**

首先將程式的要求簡化一些，讓我們先撰寫鍵盤互動、文字模式輸出界面、以及程式主要的控制邏輯，請參考：

[範例執行程式](http://squall.cs.ntou.edu.tw/cpp/92Spring/snake0.exe)

1. 程式的主要邏輯是一個迴圈，判斷使用者是否有按鍵，如果有按鍵的話，執行該按鍵 (上、下、左、右、) 所代表的轉向功能，如果沒有的話則按照原方向往前行走一步，例如：

while (!gameStopped)  
{  
　　讀取按鍵並決定方向  
　　向前走一步  
}

1. 上面這個迴圈有兩個問題存在：
   * 迴圈不斷迅速地重複執行，那 snake 不是走得飛快，一下子就走到邊邊了嗎? 而且電腦愈快，走得愈快，這不是我們想要看見的表現，請運用 windows.h 中的 Sleep(millisecond) 函式來加上適當的延遲， TURBO C 中請用 dos.h 中的 delay() 函式。
   * 我們所練習過的 stdio.h 中，或是 conio.h 中的輸入函式 scanf() gets() getchar() getch() 等等函式都有一個共同的表現：

呼叫過該函式後，必須等到操作的人鍵入一些資料以後，函式才會回返到呼叫點繼續執行下去

這樣子的表現在本程式中並不適合，如果在程式執行到了讀取按鍵的時候，操作的人沒有按下任何按鍵的話，程式就停在那兒等待，也就不會往下執行到 "向前走一步" 的動作了，螢幕上的 "\*" 符號就不會依照原方向往前走了。

這個問題在 TURBO C/Borland C 環境中有兩種解決的方法，在 Visual C++ 中可以用下面的第一個方法：

使用 conio.h 中的 kbhit() 及 getch() 兩個函式，呼叫 kbhit() 函式之時，不管操作者有沒有按鍵，該函式都會立刻執行完畢，傳回一個數值，傳回 0 代表沒有任何按鍵被按下，傳回 1 代表有按鍵被按下，通常使用如下的程式來讀取按鍵值，

if (kbhit()) c = getch();

上面的程式還有一些問題，當操作者按下功能鍵 F1, F2, ... F10 或是上下左右鍵時，得到的 c 值是 0，(在 VC 環境下是 0xe0) 但是如果再呼叫一次 getch() 則可以得到按鍵的值，請參考：

* + - * TURBO C / MSDN Library kbhit() 的線上說明
      * [kbhit.c 測試程式](http://squall.cs.ntou.edu.tw/cpp/92Spring/kbhit.c) (請注意對於函式庫內任何一個別人提供給你使用的函式都應該要撰寫一個小程式來測試它的功能，才能夠正確、放心地去使用它)
    - 在 TURBO C 中除了 kbhit() 函式之外，還有一個更一般化的函式也可以達到相同的功能，就是 bios.h 裡的 bioskey() 函式，請參考：
      * TURBO C bioskey() 的線上說明?
      * [bioskey.c 測試程式](http://squall.cs.ntou.edu.tw/cprog/Assignments/99Fall/bioskey.c)

1. 在 TURBO C/Borland C 環境中可以用 conio.h 中的 gotoxy() 和 clrscr() 函式來輸出畫面，在 VC 環境下請下載 [utilwin32.c](http://squall.cs.ntou.edu.tw/cpp/92Spring/utilwin32.c), [utilwin32.h](http://squall.cs.ntou.edu.tw/cpp/92Spring/utilwin32.h), 這兩個檔案請和你的程式放在一起, 在你的函式使用 gotoxy() 之前先 #include "utilwin32.h", 你還是一樣用 gotoxy() 和 clrscr(), 請使用選單 Project/Add to project.../Files 把 utilwin32.c 和 utilwin32.h 加入你的 project 中一起編譯/連結就可以了。

請注意: 上面的作法是假設你的主程式是 xxxx.c, 如果你的程式檔案名稱的副檔名用 xxxx.cpp 的話, 請將 utilwin32.c 檔案改名為 utilwin32.cpp 如此在連結的時候才可以順利地用到 gotoxy(), clrscr() 等函式, 否則 linker 在工作的時候會找不到 gotoxy() 及 clrscr() 這些 C 的函式, 另外一種方法是更改 utilwin32.h 內容如下:

extern "C"

{

void gotoxy(int x, int y);

void clrscr();

void delay(int milliSecond);

}

造成這個問題的原因是 C++ 函式名稱的 name mangling, 請注意上課時的解說。

(VC6 / Win32 環境中對於 console mode 的支援請參考 MSDN Library "Console sample (console functions)" 這個完整的[範例程式](http://squall.cs.ntou.edu.tw/cpp/92Spring/console.zip))

1. 在製作這個程式的時候，你可以嘗試運用 enum 敘述來定義一個有限的命令集合，例如： left, right, up, down, esc 等等命令，並且將鍵盤讀到的按鍵轉換為程式內定義的這組數值，如此若是需要更改按鍵的時候，程式內所須要改變的範圍就會限制下來，不會導致全面性的修改程式。

* 有了上面的說明以後，應該可以比較順利地完成這個簡化的範例程式了。
* 另外我們可以藉由調整 delay()/Sleep() 的時間來得到不同級數的遊戲內容。
* 在完成這個程式之後，相信大家也可以體會到程式不單可以控制 CPU 進行一些資料處理的動作之外，也可以接受操作者的指令來執行相對應的動作。

greenline

**增加 snake 程式的功能**

上面的程式中還欠缺的主要功能包括：

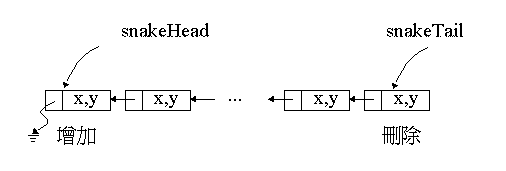
1. 目標物的產生和被 snake 吞食的動作
2. 多節 snake 在記憶體中如何表示
3. 排名表的製作

**提示與要求：**

1. 目標物在任何時候都至少存在一個，它的內部狀態應該包括位置 (x 座標, y 座標), 請定義一個結構來存放此種型態的資料，並設計在不同等級時有不同個數的目標物產生。
2. 在遊戲中我們看到 snake 的長度隨著被吃掉的目標物而逐漸增長，在記憶體中我們需要把 snake 完全記錄下來，如此
   1. 才能在螢幕上繪出有多個轉折的 snake
   2. 才能偵測 snake 在前進時會不會撞到自己
   3. 目標物產生時也才不會和 snake 重疊

記錄的方法至少有兩種：

* 1. 因為有那麼多個點，第一種方法是用陣列來記錄，每一個元素是一組 x, y 座標值，如此每一個 snake 移動的動作發生時，可以用一個迴圈將每一個元素的 x, y 座標值更新即可
  2. 上面的記錄方式對點數很多的 snake 而言，在移動的時候顯然不太有效率，所以在這次作業裡請盡量不要用這種方式來儲存；實際上每一小步的移動只有第一點和最後一點改變座標值，這種功能我們用一個簡單的佇列 (Queue) 就可以完成，如下圖：



**要求：**

* + 1. 請運用 typedef 及 struct 定義節點的結構
    2. 請定義 Queue 的結構
    3. 請為這個 Queue 結構製作 insertAtHead() 及 removeAtTail() 的函式
    4. 在上面兩個函式中請使用動態配置 malloc()/free() 來建立這個 snake

1. 排名表的製作：

請運用 FILE 及 fopen()/fclose() 開啟記錄排名表的文字檔案，排名最主要是顯示成績、姓名、以及時間三個資料，時間資料可以用 time() 函式求得自 1970 年 1 月 1 日 0 時 0 分 0 秒到現在的秒數，以長整數表示，並且存放起來，列印在螢幕上的時候可以藉由 ctime() 函式來顯示文字格式的時間。

使用文字檔案來記錄排名表時，請特別注意有些使用者會輸入 "空白字元" 作為姓名的一部分，這可能導致你的程式在重新讀入時的錯誤，請小心處理。

當然你也可以嘗試使用二進位的方式來存取檔案。

[VC 6 的 conio 和 stdio 函式間有一些合作上的小問題](http://squall.cs.ntou.edu.tw/cpp/92Spring/coniostdio.html)

1. 注意 snake 在撞到自己和撞到牆壁時的表現，請參考範例執行程式。

greenline

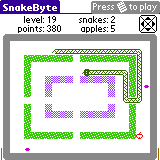
**進一步增加 snake 程式的功能**

如果你很順利地完成上一節的功能，也可以再考慮增加下面程式的功能：

[範例執行程式](http://squall.cs.ntou.edu.tw/cpp/92Spring/snake2.exe)

在上面的範例執行程式中，你可以注意到每隔一段 (不固定長度) 的時間，會出現一種 "@" 的目標物，這種目標物在一段時間內如果沒有被 snake 吃掉的話就會自動消失，如果被吃掉的話可以加一些分數。或是某一種目標物一旦被吃掉以後貪食蛇的速度會突然加快，讓使用者比較難控制...。

如下圖你也可以增加一些障礙:



**Enjoy! Enjoy! Enjoy!**