~~~~~~

一、程式作業繳交注意事項
1. 作業遲交不計分。
2. 程式碼有抄襲者，任何一次抄襲，學期成績不及格。

二、程式作業撰寫規則，違反規則該次作業不予計分
1. 程式碼不得使用全域變數，亦即，宣告在 main 外面的變數！
2. 程式碼不得使用 goto 指令。
3. 程式碼不得使用 system("pause") 功能。
4. 程式碼不得使用 非 C 語言的功能，如 C++ for (int i=0; i<x, i++) {}。
5. 作業上傳請繳交.c 檔，不得繳交 .cdp .zip .rar .7z .exe 等不合法檔案！

三、課程注意事項
1. 跟助教、卓越小老師預約課後輔導時間，請準時，學校會派人來簽到。
2. 作業要跟助教討論除錯問題，請跟助教約時間，當面討論比較有效果。
3. 作業等相關問題可寄至助教信箱詢問。
4. 下列網站提供很好的查詢資源請同學使用。
   http://www.cplusplus.com/reference/clibrary/

~~~~~~
~~~~

week 01

001

同學要選老師做專題 (project)，
N 位學生 (N student)，M 位老師 (M teacher) ，
配對方式，優先根據學生填寫志願，
如有相沖突，則考慮老師志願序。

測試案例一：
學生有 4 位分別為 W、X、Y、Z
老師有 4 位分別為 A、B、C、D
學生志願序：
W：A、B、C、D
X：A、C、B、D
~~~~

Y：D、B、C、A
Z：B、C、A、D
老師志願序：
A：W、X、Y、Z
B：X、W、Y、Z
C：Y、Z、W、X
D：X、W、Z、Y
結果：
W->A
X->C
Y->D
Z->B

測試案例二：
學生志願序：
W：A、B、C、D
X：A、C、B、D
Y：D、B、C、A
Z：B、C、A、D

老師志願序：
A：X、W、Y、Z
B：X、W、Y、Z
C：Y、Z、W、X
D：X、W、Z、Y
結果：
W->C
X->A
Y->D
Z->B

範例輸入
A,B,C,D
A,C,B,D
D,B,C,A
B,C,A,D
W,X,Y,Z

X,W,Y,Z
Y,Z,W,X
X,W,Z,Y
範例輸出
W->A
X->C
Y->D
Z->B

---

week 01

002

解 N 元一次聯立方程式

寫一個程式求解 N 元一次聯立方程式
c11X1 + c12X2 +..+ C1nXn = A1
c21X1 + c22X2 +..+ C2nXn = A2
.....
cn1X1 + cn2X2 +..+ CnnXn = An

hint: 可運用高斯消去法

輸入格式：

3
1 -1 1 4
3 2 1 2
4 2 2 8

輸出格式：

1.000 X1 - 1.000 X2 + 1.000 X3 = 4.000
3.000 X1 + 2.000 X2 + 1.000 X3 = 2.000
4.000 X1 + 2.000 X2 + 2.000 X3 = 8.000
X[1] = -4.000
X[2] = 2.000

X[3] = 10.000

為了方便理解格式，Sample Ouput 中符號'○'代表空格，程式實作時，必須以空格印出，並非以符號'○'印出

Sample Input
4
1 1.5 2.5 3.5 1
0.375 1 0.25 1 1
0.5 2 1 1 1
1 2 2 1 1

Sample Outpt
1.000○X1○+○1.500○X2○+○2.500○X3○+○3.500○X4○=○1.000
0.375○X1○+○1.000○X2○+○0.250○X3○+○1.000○X4○=○1.000
0.500○X1○+○2.000○X2○+○1.000○X3○+○1.000○X4○=○1.000
1.000○X1○+○2.000○X2○+○2.000○X3○+○1.000○X4○=○1.000
X[1]○=○1.538
X[2]○=○0.385
X[3]○=○-0.769
X[4]○=○0.231

---

week 01

003.

臺北科大宿舍商店賣 20 項商品，分別編號 {1, 2, 3, 4, ..., 20}。
同學來買商品最多五項，最少一項。
一天最多有一千筆資料。
請分析哪兩筆資料是最多人一起購買，以作為老闆擺放商品的依據。

輸入說明
第一行為購買資料筆數
第二行之後為每筆購買商品編號，編號未依大小順序，中間以逗號隔開

輸出說明
兩個編號以及出現次數，中間以逗號隔開，編號從小到大出現

測試資料(Test case: input)

7
1,3,5,7,9
2,3
1,3,5
4,7
7,9
3,5,7
9,1

測試資料(Test case: output)
3,5,3

~~~~~~
~~~~
Week 02

004.
走迷宮

如下圖 8 x 8 迷宮地圖中，1 代表牆壁，0 代表可走路徑。

1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 1
1 1 1 0 1 1 1 1
1 0 0 0 0 0 0 1
1 1 0 1 0 1 1 1
1 1 0 1 0 0 0 1
1 1 0 1 1 1 0 1
1 1 1 1 1 1 1 1

請設計一個遞迴程式，找出從左上角出發，到右下角的路徑，
將 0 以 # 標示輸出，如下圖。

1 1 1 1 1 1 1 1
1 # # # 0 0 0 1

1 1 1 # 1 1 1 1
1 0 0 # # 0 0 1
1 1 0 1 # 1 1 1
1 1 0 1 # # # 1
1 1 0 1 1 1 # 1
1 1 1 1 1 1 1 1

輸入說明:
第 1 行輸入 n，n<40
第 2 行到第 n+1 行，為 n x n 地圖資料輸入，1 代表牆壁，0 代表可走路徑。
左上角[1][1]的 0 為起點，右下角 [n-2][n-2]的 0 為終點。

輸出說明:
第 1 行到第 n 行，為地圖與可走路徑，可走路徑以 # 標示。

Smaple Input
8
1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 1
1 1 1 0 1 1 1 1
1 0 0 0 0 0 0 1
1 1 0 1 0 1 1 1
1 1 0 1 0 0 0 1
1 1 0 1 1 1 0 1
1 1 1 1 1 1 1 1

Sample Output
1 1 1 1 1 1 1 1
1 # # # 0 0 0 1
1 1 1 # 1 1 1 1
1 0 0 # # 0 0 1
1 1 0 1 # 1 1 1
1 1 0 1 # # # 1
1 1 0 1 1 1 # 1
1 1 1 1 1 1 1 1

---

week 02

005.

寫一個程式，猜數字。

(1) 程式的使用者設定一個答案 X，四位數，0~9 不重複。
(2) 程式的使用者輸入幾個 4 位數字，Yi, 3<i<10，以及這個 4 位數的暗示結果。
(3) 程式必須輸出使用者設定的答案。

暗示結果的規則
(1) Yi 中有 1 位數字跟答案 X 一樣，且所在位置相同，像千位對千位，或百位對百位，就是 1A。若 2 位都有這情況，就是 2A。
　　例如 X=1234，Y1=1856，1 數字一樣，位置都是在千位，所以是 1A。
(2) Yi 中有 1 位數字跟答案 X 一樣，但所在位置不同，就是 1B。若 2 位都有這情況，就是 2B。
　　例如 X=1234，Y1=8561，1 數字一樣，位置不同，所以是 1B。
(3) 以上兩條規則可以合併使用，但以 (1)優先，之後再考慮 (2)。

---
輸入範例說明:

每一行輸入 4 位數字 Yi，以及暗示的結果 xAxB。
輸入的行數 N<10。
(假設使用者設定的答案是 4237)

1968,0A0B　　數字都沒有對，所以暗示答案為 0A0B。
7052,0A2B　　有 2 個數字對 (7, 2) ，但位置不對，所以暗示答案為 0A2B。
2347,1A3B　　有 1 個數字對且位置對 (7)， 3 個數字對 (2, 3, 4)，所以暗示答案為 1A3B。
3427,1A3B　　有 1 個數字對且位置對 (7)， 3 個數字對 (3, 4, 2)，所以暗示答案為 1A3B。
2473,0A4B　　4 個數字對，位置不對。

輸出範例說明:
4237 輸出猜到使用者設定的答案

Sample Input

1968,0A0B

7052,0A2B
2347,1A3B
3427,1A3B
2473,0A4B

Sample Outpt
4237

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~
week 03

006

賓果遊戲

假設一個簡單的兩人賓果遊戲，每位玩家各自輸入一個九宮格，九個數字 N1,N2,…N9，且 Ni<>Nj 當 i<>j，而 1<=Ni, Nj<=9，1<=i, j<=9。

N1 N2 N3
N4 N5 N6
N7 N8 N9

電腦從 1~9 的整數中選三個數字 C1、C2、C3，且 Ci<>Cj 當 i<>j，而 1<=Ci, Cj<=9, 且 1<=i, j<=3。

獲勝情況有三種，第一位玩家贏，第二位玩家贏、和平手。

當這三個數字，只在一位玩家的九宮格中，成一條水平線，如上圖的 N1,N2,N3、垂直線，
如圖中的 N1,N4,N7，
或對角線，如圖中的 N1,N5,N9，則該玩家獲勝。否則平手。
請寫出一個程式，展現電腦判斷此兩人遊戲獲勝情況。

輸入說明:
輸入多組測試案例，每個測試案例為一次賓果遊戲，格式：
第一行為九個數值介於 1 到 9 且相異的整數，每個數字間隔一個空格，表示第一位玩家填寫的九宮格，對應圖中 N1,N2,…N9。
第二行為九個數值介於 1 到 9 且相異的整數，每個數字間隔一個空格，表示第二位玩家填寫

的九宮格，對應圖中 N1,N2,…N9。
第三行為三個數值介於 1 到 9 且相異的整數，每個數字間隔一個空格，表示電腦選擇的三個數字。
每一組測試案例以 0 間隔。
最後以 -1 結束。

輸出說明:
每一行為每一個測試案例的執行結果，即每一場賓果遊戲的獲勝情形。
若為玩家一獲勝，顯示 Player1 wins；
若為玩家二獲勝，顯示 Player2 wins；
若平手，則顯示 Draw。

Sample Input:
1 2 3 4 5 6 7 8 9
9 8 7 6 5 4 3 2 1
1 2 3
0
1 4 2 5 6 7 3 8 9
9 8 4 7 6 3 5 2 1
2 7 9
0
9 4 2 5 6 1 3 8 7
9 8 4 3 6 7 5 2 1
9 4 8
-1

Sample Output:
Draw
Player1 wins
Player2 wins

---

week 03

007

賴先生騎腳踏車挑戰一日 N 塔，N<10。
每一個塔位在編號 1, 2, 3, ...,N 城市中。
兩兩個城市都有一段距離的公路相連。

請計算從第 1 個城市出發，騎過每一個城市的最短距離。

例如 N = 5，以下是兩兩城市之間公路的距離。

```
-- 1   2   3   4   5
1  0   4   2   3   6
2  4   0   3   1   4
3  2   3   0   2   5
4  3   1   2   0   3
5  6   4   5   3   0
```

例如
城市 1 和城市 2 的距離是 4，
城市 1 和城市 3 的距離是 2，
城市 3 和城市 4 的距離是 2，
城市 5 和城市 4 的距離是 3，

則從城市 1 出發，騎完所有城市的距離最短是，
經由 13245 的距離 = 2+3+1+3= 9。

輸入說明

第 1 筆資料是 N，
第 2 筆資料是第 1 個城市和其他城市的公路距離。
第 3 筆資料是第 2 個城市和其他城市的公路距離。
....
第 N+1 筆資料是第 N 個城市和其他城市的公路距離。

輸出說明

輸出從第 1 個城市出發，騎完所有城市最短距離。

Sample Input

```
5
0   4   2   3   6
```

4 0 3 1 4
2 3 0 2 5
3 1 2 0 3
6 4 5 3 0

Sample Output
9

~~~~~~

week 04

008

英文字分析、取代、插入、刪除

輸入一篇英文文章 A，文章中英文字以一個空白間隔。另外輸入 2 個英文字(word) P、Q。
(1) 將文章 A 中 P 字串以 Q 字串取代，並輸出。
(2) 在文章 A 中 P 字串前插入 Q 字串，並輸出。
(3) 將文章 A 中 P 字串刪除，並輸出。
(4) 分析文章 A 所有英文字 (word) 的頻率，依頻率由大自小排序， 頻率相同則以 word 由小自大排序(That > This....)輸出。

輸入範例說明:
第一行，文章 A
第二行，英文字 P
第三行，英文字 Q

輸出範例說明:
第一行，文章 A 將 P 替換成 Q。
第二行，文章 A 將 Q 插入 P 前面。
~~~~~~

第三行，文章 A 將 P 刪除。
第四行之後，每一行依序為英文字、出現頻率次數，中間以逗號間隔。

Sample Input

```
This is a book That is a cook
is
was
```

Sample Outpt

```
This was a book That was a cook
This was is a book That was is a cook
This a book That a cook
a 2
is 2
That 1
This 1
book 1
cook 1
```

Week 4

009.

解密碼

老張是個有錢的大地主，他將所有金銀財寶都收藏在一個藏寶箱裡，藏寶箱要輸入正確密碼才能開啟。
老張將寶箱密碼寫在一卷捲軸，將遺產交給唯一獨生子。
捲軸內有兩行，由數字 0~9 和大小寫英文字母、標點符號、空格組成。
解碼方式是將出現在第一行文章的連續數字取出反轉，將所有反轉後的數字相加，超過四位數，取四位數(小於 10000)，不足四位數，前面取 0，即可取得第一組四位數密碼。
第二組四位數密碼是將第二行文章分析字元出現頻率，找出小於 10 的前四個數字，數字排列依該字元出現次序。

例如 :

Today is my 45 birthday, There are 65 guest coming for the 1000 party, I got 789 gifts to open.
Object-oriented programming is a programming paradigm based on the concept of objects, a app, xxxxxxxxxxx.

第一行取出其中數字 45, 65, 1000, 789， 分別作反轉後相加。54+56+1+987+0=1098 ，第一組密碼即為 1098。
第二行分析字元頻率，(e, 7), (o,7), (r, 6), (a, 8)，第二組密碼即為 7768，其中(x, 11)超過 10 不予計算。

輸入說明 :

輸入二行文章，含有大小寫英文字母，數字 0~9、空格、標點符號。

輸出說明 :

八位數字輸出於一行。

Sample Input

Today is my 45 birthday, There are 65 guest coming for the 1000 party, I got 789 gifts to open.
Object-oriented programming is a programming paradigm based on the concept of objects, a app, xxxxxxxxxxx.

Sample Outout
10987768
~~~~~~

Week 5

010

Recursion

數位電路模擬 I
~~~~~~

寫一個程式模擬一個數位電路。
輸入 n 是二進位 8 位元，輸出是二進位 4 位元。
輸入範圍從 00000000 到 11111111 (十進位 0~255).

此數位電路內具有回饋迴路，其功能如下:
C(m) = m if m = 0 or m = 1
C(m) = C(m/2) if m is even
C(m) = C((m+1)/2) if m is odd
此電路有一個紀錄器，會記錄跑過幾次回饋迴路，最後輸出為回饋電路跑過的次數。

例如 m=00001010(十進位 10)，則電路內部運算回饋電路輸入依序為十進位 5, 3, 2, 1。
C(10)= C(5)=C(3)=C(2)=C(1)=1
共跑過 4 次。則此電路輸出為 0100 (十進位 4)。

輸入說明:
二進位 8 bit 位元
第一行是第一個測試案例資料
接著是一行 0 分隔測試資料
第三行是第二個測試案例資料
....
最後 -1 結束

輸出說明:
二進位 4 bit 位元
每一行是一個測試案例資料的結果

Sample Input:
00000000
0
11111111
0
00000001
0
10000000
0
00111111
-1

Sample Output:

0000
1000
0000
0111
0110

Week 5

011

數位電路模擬 II

寫一個程式模擬一個數位電路。
輸入 n 是二進位 8 位元，輸出是二進位 11 位元。
輸入範圍從 00000000 到 11111111 (十進位 0~255).

此數位電路內具有回饋迴路，其功能如下:
C(m) = m if m = 0 or m = 1
C(m) = C(m/2) if m is even
C(m) = C((m+1)/2) if m is odd
此電路有一個紀錄器，會記錄跑過幾次回饋迴路。

例如 m=00001010(十進位 10)，則電路內部運算回饋電路輸入依序為十進位 5, 3, 2, 1。
C(10)= C(5)=C(3)=C(2)=C(1)=1
共跑過 4 次。則 R(10)=4。

例如 m=00000011(十進位 3)，則電路內部運算回饋電路輸入依序為十進位 2, 1。
C(3)= C(2)=C(1)=1
共跑過 2 次。則 R(3)=2。

此電路另有一個紀錄器，會記錄 m=0, m=1, m=2, ..m=n，所跑過幾次回饋迴路的加總。
即 R(0)+R(1)+...+R(n)。此即最後的輸出。

假設輸入 n = 3，此電路輸出為 R(0)+R(1)+R(2)+R(3)=0+0+1+2=3 (二進位 0011)。

輸入說明:

二進位 8 bit 位元
第一行是第一個測試案例資料
接著是一行 0 分隔測試資料
第三行是第二個測試案例資料
....
最後 -1 結束

輸出說明:
二進位 11 bit 位元
每一行是一個測試案例資料的結果

Sample Input:
00000000
0
11111111
0
10101010
-1

Sample Output:
00000000000
11011111001
10001010001

~~~~~~
week 6

012

葛雷碼 (Gray code)

反射二進位編碼-葛雷碼 (Gray code)，是編碼成兩個連續的位元不同。
輸入 n ，編碼範圍 0<=i<=2^n-1。
n = 3，編碼 0~7 為 000, 001, 011, 010, 110, 111, 101, 100。

其編碼方式如下

G_1 = {0, 1}
~~~~~~

G_1_r = {1, 0}

G_n　　= {0G_(n-1), 1G_(n-1)_r }
if G_n　= {g1, g2, g3, ..., gn}
　G_n_r = {g_n, g_(n-1), g_(n-2), ..., g_1}

[G_n_r 是 G_n 的逆向順序]

G_(n+1)　= {0G_n, 1G_n_r}

例如

G_2　= {0G_1, 1G_1_r} = {00, 01, 11, 10}
G_2_r = {10, 11, 01, 00}

G_3　= {0G_2, 1G_2_r} = {000, 001, 011, 010, 110, 111, 101, 100}
G_3_r =　　　　{100, 101, 111, 110, 010, 011, 001, 000}

G_4　= {0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101, 1111, 1110, 1010, 1011, 1001, 1000}
G_4_r = {1000, 1001, 1011, 1010, 1110, 1111, 1101, 1100, 0100, 0101, 0111, 0110, 0010, 0011, 0001, 0000}

其遞迴公式為，
G(n, k) = k　　　　if n=1
G(n, k) = 0G(n-1, k)　　if k<2^(n-1)
G(n, k) = 1G(n-1, 2^n-1-k)　if k>=2^(n-1)

當 G(4, 7) = 0G(4-1, 7) = 0G(3, 7) = 01G(3-1, 2^3-1-7) = 01G(2, 0) =　010G(2-1, 0) = 010G(1, 0) = 0100

依此撰寫遞迴程式，輸入 n, k，輸出 Gray code。

輸入說明:

第一行是一個測試案例資料，整數 n k
接著是一行 0 分隔測試資料
第三行是第二個測試案例資料
最後 -1 結束

輸出說明:

二進位 Gray code
每一行是一個測試案例資料的結果

Sample Input:
1 1
0
2 3
0
3 6
0
4 12
-1

Sample Output:
1
10
101
1010

Week 6

013

小英要為登玉山準備，練習爬 101 大樓的樓梯。
可以一步踏一階、二階、最長跨三階。因此有很多種爬階方法。
例如要爬到第三階，有四種爬法:
(1)一次一階。
(2)先爬二階，再爬一階
(3)先爬一階，再爬二階
(4)一次三階。

要爬到第四階，有七種爬法。

最大要爬 70 層。

此程式執行時間不可超過 8 秒。

Hint:
1. 一次可跨 1~n 階時(n>=1)，從第 n+1 階開始，其全部方法為前 n 階方法數總和。
   若 n=3, f(n) = f(n-1) + f(n-2) + f(n-3)

2. 請考慮使用遞迴與非遞迴方法，或配合指標使用。

3. 請使用 unsifned long long int x; printf("%lld", x);

輸入說明

整數 K，K<70，代表共爬 K 階。

輸出說明

整數 M，代表共有幾種爬法。

Sample Input

4

Sample Output

7

Sample Input

12

Sample Output

927

Sample Input

67

Sample Output

333269972246340068

Week 6

014

高中職畢業生希望選擇心目中理想的大學。
假設每一大學可以用下列七種屬性表示：

BC(Big Campus)：代表有大校園。
NC(Next to City)：代表鄰近有大城市。
CT(Convenient Transportation)：代表交通方便。
NS(Next to Sea)：代表靠海。
NM(Next to Mountain)：代表依山。
HL(Has Lake)：代表校園有湖。
NL(Near Landscape)：代表附近有風景區。

使用者可輸入理想中的大學條件，用 + 號區格的條件代表 "或" 的關係。

例如： BC NS + CT HL 代表需找出有大校園且靠海，或交通方便且校園有湖的所有大學名稱。

輸入說明：

第一行有一個正整數，代表大學個數 n ，請注意 n<=10 。

其後 n 行，每一行第一項為大學名稱，接著為大學具備的屬性。
大學名稱及各屬性間以一個空白分隔。
一個大學名稱最多有 10 個字母，各項屬性為 2 個字母。大學與屬性資料均為英文字母，

接下來的一行有一個正整數，為查詢的個數 m ， m<=10 。

其後 m 行，每一行有一個查詢。
查詢條件為校園屬性組成，每個校園屬性為兩個字元，
用 + 號區格的條件代表 "或" 的關係，沒有 + 區隔的條件代表 "且" 的關係。
屬性之間以及和 + 之間可能有空白，也可能沒有空白。
其格式為以下任一種:

XXYY+AABB
XX YY + AA BB

意思為屬性條件為: XX 且 YY，或是 AA 且 BB。

輸出說明：
印出 m 行，第 i 列印出第 i 個查詢中，所有符合之大學名稱。
若有多個大學符合一個查詢，各大學間以一個空白分隔。

Sample Input:

5
NSYSU NC CT NS NM
NTU BC NC CT NS
NCCU BC NL HL
Providence BC NC
NTHU BC NS
2
BC NS + CT HL
NM + BC NL + BC NC

Sample Output:
NTU NTHU
NSYSU NTU NCCU Providence

Week 8

015.

銀行存款

假設某人有兩個帳戶 A、B，
帳戶 A 的金額預設 0、帳戶 B 的金額預設 0。
設計一個程式，功能有存款、提款、查詢餘額及查詢兩個帳戶餘額的百分比。

其指令為:
a: 選擇帳戶 A。
b: 選擇帳戶 B。
v: 查詢所選帳戶餘額，可能為負值。
w: 所選帳戶提款。
s: 所選帳戶存款。
p: 查詢兩個帳戶餘額的百分比，計算方式:
若一餘額為負數或 0，另一餘額為正數，則比例為 0%:100%。
兩餘額均為 0，比例為 50%:50%。
其餘情況 (帳戶 A) : (帳戶 B) 其公式為 A/(A + B) %，(1-A/(A + B))%，
先計算 A 帳戶百分比，百分比捨去小數，B 百分比為 100-A 百分比。
e : exit

輸入說明

a: 選擇帳戶 A。輸出 Select A。
b: 選擇帳戶 B。輸出 Select B。
v: 查詢所選帳戶餘額，可能為負值。
w: 所選帳戶提款。輸出 [所選帳戶 Withdraw]。輸出所選帳戶餘額。
s: 所選帳戶存款。輸出 [所選帳戶 Deposit]。輸出所選帳戶餘額。
p: 查詢兩個帳戶餘額的百分比。
e : exit

輸出說明
a: 選擇帳戶 A。輸出 [Select A]，中間一個空白。
b: 選擇帳戶 B。輸出 [Select B]，中間一個空白。
v: 查詢所選帳戶餘額，可能為負值。輸出[帳號名稱:帳戶餘額]，中間沒有空白。
w: 所選帳戶提款。輸出 [所選帳戶:Withdraw,帳戶餘額]，中間沒有空白。
s: 所選帳戶存款。輸出 [所選帳戶:Deposit,帳戶餘額]，中間沒有空白。
p: 查詢兩個帳戶餘額的百分比 A/(A + B)。輸出[A:所佔百分比,B:所佔百分比]，中間沒有空白。

==========

Sample input:

```
a
v
p
s
2000
b
v
s
2000
p
a
w
1000
b
w
1000
b
w
1500
a
w
500
p
b
s
2000
p
a
s
500
b
s
500
p
e
```

==========

Sample output:

```
Select A
A:0
A:50%,B:50%
A:Deposit,2000
A:2000
Select B
B:0
B:Deposit,2000
B:2000
A:50%,B:50%
Select A
A:Withdraw,1000
A:1000
Select B
B:Withdraw,1000
B:1000
Select B
B:Withdraw,1500
B:-500
Select A
A:Withdraw,500
A:500
A:100%,B:0%
Select B
B:Deposit,2000
B:1500
A:25%,B:75%
Select A
A:Deposit,500
A:1000
Select B
B:Deposit,500
B:2000
A:33%,B:67%
```

Week 8

016.

電腦在網際網路的 ip 位址是由四個正整數 (0~255)組成，中間以據點隔開，例如:

140.124.1.255

為了方便記憶，可以取一個暱稱。
同一個網段的網址，第一個和第二個正整數相同。

請撰寫程式處理一串網址，辨識是否在同一個網段。

輸入說明:

每一筆是一個網址，加上其暱稱，中間以逗號間隔。
0.0.0.,none 結束輸入符號。

輸出說明:

以暱稱顯示所有同網段的電腦，
印出的順序為輸入順序，其格式為

machines [暱稱 1] and [暱稱 2] are on the same local network.

若有三個相同(依此類推)，其格式為
machines [暱稱 1], [暱稱 2] and [暱稱 3] are on the same local network.

若有不合法的網址輸入 (任一數字>255)，輸出錯誤訊息置於最後，
每一個錯誤 ip 輸出一行，輸出的順序同輸入順序，其格式為

machines [暱稱] is error ip

Sample Input:

111.22.3.44,blue
55.66.7.88,red
111.22.55.55,black
111.2.5.66,green
555.66.1.2,John
111.22.1.1,Tom

55.66.11.22,Mary
1111.1.1.1,Ken
0.0.0.0,none

Sample Output:
---
machines blue, black and Tom are on the same local network.
machines red and Mary are on the same local network.
machine John is error ip
machine Ken is error ip

Week 9

加分題

017

小英要為登玉山準備，練習爬 101 大樓的樓梯。
可以一步踏一階、二階、三階、四階...。因此有很多種爬階方法。
例如可以跨 1~3 階，要爬到第三階，有四種爬法:
(1)一次一階。
(2)先爬二階，再爬一階
(3)先爬一階，再爬二階
(4)一次三階。

最大要爬 70 層。

此程式執行時間不可超過 8 秒。

Hint:
1. 一次可跨 1~n 階時(n>=1)，從第 n+1 階開始，每一階的全部方法為前 n 階方法數總和。
   若 n=3, f(n) = f(n-1) + f(n-2) + f(n-3)
   若 n=4, f(n) = f(n-1) + f(n-2) + f(n-3) + f(n-4)

2. 請考慮使用遞迴與非遞迴方法，或配合指標使用。

3. 請使用 unsifned long long int x; printf("%lld", x);

輸入說明

第一筆，整數 K，K<70，代表共爬 K 階。
第二筆，整數 n，n<K，代表一次可爬 1~n 階。

輸出說明

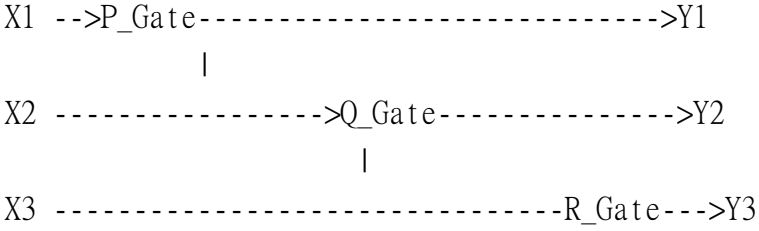
整數 M，代表共有幾種爬法。

Week 10

018

本題必須使用後所附程式碼架構~

以下邏輯電路圖，輸入為 X1, X2, X3，輸出為 Y1, Y2, Y3。



P_Gate 邏輯閘可設定為 NOT 或空，輸入為 X1，輸出為 Y1 和 Q_Gate 邏輯閘的輸入。
Q_Gate 邏輯閘可設定為 AND 或 OR，輸入為 X2 和 P_Gate 邏輯閘的輸出，輸出為 Y2 和 R_Gate 邏輯閘的輸入。
R_Gate 邏輯閘可設定為 AND 或 OR，輸入為 X3 和 Q_Gate 邏輯閘的輸出，輸出為 Y3。
輸入 X1、X2、X3，以及設定 P、Q、R、三個邏輯閘的種類。

輸入說明:
第一行依次輸入 X1、X2、X3 為 0 或 1，中間以逗號間隔。
第二行輸入 P、Q、R 邏輯閘的設定，A 代表 AND 邏輯閘，O 代表 OR 邏輯閘，N 代表 NOT 邏輯閘，E 代表空的邏輯閘，中間以逗號間隔。

輸出說明:
輸出 Y1、Y2、Y3 為 0 或 1，中間以逗號間隔。
範例:

Sample Input:
0,1,0
N,A,O

Sample Output:
1,1,1

------程式碼架構規範之範例----------

```
#include <stdio.h>
#include <stdlib.h>
#define GATEVALUE(TYPE) int(*GateValue)(void)
typedef struct _Gate{
    GATEVALUE();
}Gate;
int GateGetValue(){return 0;}
typedef struct _GateAnd{
    GATEVALUE();
}GateAnd;
int GateAndValue(){return 1;}
void CreateGate(Gate *obj){
    obj->GateValue = GateGetValue;
}
void CreateGateAND(GateAnd *obj){
    obj->GateValue = GateAndValue;
}
int main(int argc, char *argv[]){
    Gate gate;
    CreateGate(&gate);
    GateAnd and;
    CreateGateAND(&and);
    printf("Gate = %d, GateAND = %d\n", gate.GateValue(), and.GateValue());
    return 0;
}
```

Week 10

019
http://www.cc.ntut.edu.tw/~jykuo/course/ploym01.c

依據以上超連結程式碼為基礎，設計程式計算各個圖形的周長，以及所有圖形的周長加總。

利用結構 struct 定義 Shape（圖形）, Circle（圓）, Rectangle（矩形）, Square（正方形）, Triangle（三角形）。

圓有半徑，矩形有寬和高，正方形有邊長，三角形有三個邊。

輸入說明 ：

第一行輸入圖形個數 N。

第二行到第 N+1 行輸入圖形種類、以及該圖形所需整數資料，以空白間隔。

圖形種類以字元表示，C 代表圓、R 代表矩形、S 代表正方形、T 代表三角形。

如果輸入 C 要跟隨一個數值為半徑，
　　輸入 R 要跟隨兩個數值寬和高，
　　輸入 S 要跟隨一個數值代表邊長，
　　輸入 T 要跟隨三個數值代表三個邊。

輸出說明：

輸出 N+1 行，

前 N 行整數，代表 N 個圖形的周長。

第 N+1 行整數，代表 N 個圖形的周長總和。

PI 設為 4。

Sample Input:

```
5
T 3 4 5
S 1
R 2 3
C 1
T 3 4 6
```

Sample Output:

```
12
4
10
8
13
47
```

Week 10

020

高精確度小數運算 High Precision

http://youtu.be/vr9S0xH5t6A

定義一個結構 high_precision_t 表達一個 40 位十進位數字的精確度。
此結構包含
digit 一個 40 元素的整數陣列，
decpt 一個整數表達小數點的位置。
sign 一個整數表達正負號。

例如 -8.127 和 0.0094328 可以被存成

```
digit=8127
decpt=1
sign=-1

digit=94328
decpt=-2
```

sign=1

製作一 function 可以從使用者取得數字，
此 function 必須偵測格式的錯誤，然後回傳 -1。

```
int input(high_precision_t * n, int size);
```

製作一個 function 印出此結構的值。

寫出程式，計算兩個數的相加、相減、相乘。

若超過精確度，無條件捨去。

輸入說明：
```
--------------------
-8.127
0.0094328
a
```

第 1 行為 被加數
第 2 行為 加數
第 3 行為 運算符號 a (+) 、 s (-) 、 m (*)

輸出說明：
```
--------------------
digit=81175672 (一個 20 元素的整數陣列，最後為 0 不顯示)
decpt=1 (小數點的位置)
sign=-1 (正負號)
```

若輸入數值格式錯誤，則輸出
```
input error
```

Sample Input:
```
-------------------
-8.127
0.0094328
a
```

Sample Output:

```
------------------
digit=81175672
decpt=1
sign=-1

Sample Input:
------------------
-05434510.7
-0.1
m


Sample Output:
------------------
input error


Sample Input:
------------------
-11.15415618
-1.148484
s

Sample Output:
------------------
digit=1000567218
decpt=2
sign=-1


==========
==========
week 11
021

輸入 1
新增輸入，原有程式記憶體資料內容不會被刪除，
依序輸入 n 個 byte (0~255)，直到 -1 停止。
結束要輸出 bytes 數量及資料。
```

e.g.
23
124
56
48
-1

輸入 2

假設從二進制檔案(binary)讀入資料內容，

(本題不需要實作檔案操作 FILE * fp; fp = fopen() ...)

一個 byte 一個 byte 讀入，原有程式記憶體內容將被清除。
例如有 n bytes，則利用動態記憶體配置 n byte (char)儲存。
檔案為測試系統平台內建。

輸入 3
以 binary 方式輸出目前程式記憶體內容。

輸入 4
輸入要變更第 n 個 bit (從0計算)，
輸入要變更的值。
每次修改完畢，要輸出資料內容。

第n個算法：由上而下，由右至左
例如變更第 12 bit 為 1 ，則依序輸入:
4
12
1

輸入 5
程式結束

Sample input

2
a.bin
1

23
124
56
48
-1
4
1
0
5

Sample output

size=4bytes
0 0 0 1 0 1 1 1
0 1 1 1 1 1 0 0
0 0 1 1 1 0 0 0
0 0 1 1 0 0 0 0

0 0 0 1 0 1 0 1
0 1 1 1 1 1 0 0
0 0 1 1 1 0 0 0
0 0 1 1 0 0 0 0

==========

week 11
022

簡易電子投票

1. 管理者註冊使用者
(1) 以字母 "A" 開頭，輸入註冊使用者帳號，密碼，電子郵件地址。
(2) 第一筆資料輸入為註冊管理者帳號密碼等資料。
輸入範例為 "A,user9130,i2q9_yt8,oper@java.com.tw"。
新增成功則輸出 "Add user successful"，資料驗證錯誤則輸出 "Add user error"。
驗證錯誤，先輸出 "username error"，沒有錯誤再輸出"password error"，再輸出 "Add user error"。

(3) 使用者資料驗證方式

帳號須以英文字母和數字混和的 8 位長度，不可與已有帳號重複，若有錯誤須輸出 "username error"；

密碼為{數字,_,-,英文字母,@,#,%}組成之字串，長度為 1~8，不可以為帳號，若有錯誤須輸出 "password error" ；

2. 管理者輸入候選人資料

(1) 以字母 M 開頭，輸入管理者帳號、密碼，候選人號次和姓名，輸入範例為 "M,Admi9,er09_1c8,1,Lin"。

(2) 號次或名字重複則輸出 "Candidate data error"，管理者帳號密碼錯誤則輸出 "Login error"，

若沒有錯誤則輸出 "Add candidate successful"。

3. 使用者登入投票

(1) 輸入以字母 V 開頭，接著使用者帳號、密碼、候選人號數，一人只能投一次票，

輸入範例為 "V,user9130,i2q9_yt8,3"。

(2) 若有重複投票，或管理者投票，須輸出錯誤訊息 "Voting error"；

若選擇之號碼數為不存在的候選人號碼，須輸出錯誤訊息 "Candidate error"，

若沒有錯誤則輸出 "Voting successful"。

帳號密碼錯誤則輸出 "Login error"。

4. 使用者登入查詢各候選人得票數

(1) 輸入以字母 Q 開頭，接著使用者帳號密碼，輸入範例為 "Q,user9130,i2q9_yt8"。

(2) 若 1 號 5 票，2 號 3 票，輸出範例為 (1,name1,5),(2,name2,3)。

帳號密碼錯誤則輸出 "Login error"。

5. 最後以 E 結束，輸出 "Exit"。

輸入說明

第一筆資料，第一筆資料輸入為註冊管理者帳號、密碼、電子郵件資料，中間沒有空白間隔。

輸入範例 "A,Admi9,er09_1c8,oper@java.com.tw"，中間沒有空白間隔。

接著輸入資料，每一筆可為:

管理者輸入候選人資料，輸入範例 "M,Admi9,er09_1c8,1,Lin"，中間沒有空白間隔。

管理者輸入註冊使用者資料，輸入範例 "A,user9130,i2q9_yt8,oper@java.com.tw"，中間沒有空白間隔。

接著輸入資料，每一筆可為:

使用者輸入投票資料，輸入範例 "V,user99uq,ujr0p4,1"，中間沒有空白間

隔。
接著輸入資料，每一筆可為:
使用者查詢各候選人得票數，輸入範例 "Q,user9130,i2q9_yt8"，中間沒有空白間隔。
最後輸入 E 結束。

Sample Input

A,Admi9111,er09_1c8,oper@java.com.tw
M,Admi9111,er09_1c8,1,Lin
M,Admi9111,er09_1c8,2,Lee
M,Admi9111,er09_1c8,1,Lin
A,user99uq,ujr0p4,user@cpp.com.tw
A,user9130,i2q9_yt8,oper@cpp.com.tw
V,user9130,i2q9_yt8,3
V,user99uq,ujr0p4,1
V,user9130,i2q9_yt8,2
Q,user9130,i2q9_yt8
E

Sample output

Add user successful
Add candidate successful
Add candidate successful
Candidate data error
Add user successful
Add user successful
Candidate error
Voting successful
Voting successful
(1,Lin,1),(2,Lee,1)
Exit

==========

week 14

023

Stack 可使用串列 link list 或陣列實作，本題使用串列實作。

https://youtu.be/w3nwIuhZ2K0

允許在一端進行後進先出（LIFO, Last In First Out）的原理運作。

兩種基本操作：push 和 pop

push：將數據放入堆疊的頂端（陣列形式或串列形式），堆疊頂端 top 指標加一。

pop：將頂端數據資料輸出（回傳），堆疊頂端資料減一。

輸入說明:

1 代表 push，再輸入整數

2 印出 pop 回傳的數字， The data %d is pop\n
若 stack 中為空則印出 The Stack is empty\n

3 結束程式。

Sample Input

```
1 2
1 5
2
1 3
2
2
1 3
2
2
3
```

Sample output

The data 5 is pop
The data 3 is pop
The data 2 is pop
The data 3 is pop
The Stack is empty

~~~~~~

Week 14

024.

Link List 多項式

使用 Link List，輸入兩個多項式，輸出相加的結果。例如:

2x^4 + 3x^3 + x -1
x^5 - x^3 + 4x^2 - 3x +2

結果：

x^5 + 2x^4 + 2x^3 +4x^2 - 2x + 1

輸入說明

輸入兩筆資料，分別代表兩個多項數。
每一筆輸入 n 個整數，第一個代表 n-1 次方的係數，第 n 個代表 0 次方係數。

輸出說明

兩個多項式相加後的係數。

Sample Input

2 3 0 1 -1
~~~~~~

1 0 -1 4 -3 2

Sample Output
1 2 2 4 -2 1

~~~~~~

Week 14

025

Link List 分數

```
#include
typedef struct {
int id; //學號
int score; //分數
} student_t;

typedef struct node {
student_t data;
struct node *next;
} node_t;
typedef node_t * nodep_t;
```

(1) void printList(nodep_t p); // 印出 List 內容
(2) void insertInOrder(nodep_t *p, student_t data);// 加入之後的LIST是依照由小到大分數順序的

輸入說明：

1. 輸入 p 表示印出 List 內容 function 1
2. 輸入 i 表示加入一筆資料 function 2
   之後接著輸入學號、分數，例如 1,2， 以逗號隔開，中間沒有空白。
3. 輸入 e 結束程式。
~~~~~~

輸出說明：

p (印出):
1. list 空的，印出 null。
2. list 內有元素，按照分數排列，印出學號、分數，中間以逗號間隔。
3. 分數相同則學號小的在前。

Sample Input:

```
p
i
1,2
p
i
3,0
p
i
2,99
p
i
0,99
p
i
7,99
p
e
```

Sample Output:

```
null
1,2
3,0,1,2
3,0,1,2,2,99
3,0,1,2,0,99,2,99
3,0,1,2,0,99,2,99,7,99
```

==========

week 15

026. 建構唯一二元樹

給定前序或後序以及中序，產生唯一個 Binary Tree，依序印出 Tree 的內容，印出順序，由上而下，由左而右印出。

前序代碼：P

中序代碼：I

後序代碼：O

輸入說明

第一筆輸入前序、中序或後序代碼。
第二筆輸入上一筆輸入種類尋訪的結果，大寫英文字母。
第三筆輸入前序、中序或後序代碼。
第四筆輸入上一筆輸入種類尋訪的結果，大寫英文字母。

輸出說明

輸出為一二元樹的內容，由上而下，由左而右。

Sample input

P
ABCDEFGHI
I
BCAEDGHFI

Sample output

ABDCEFGIH

---

week 15

027. binary tree

```
#include
typedef struct node_s {
int data;
struct node_s * left;
struct node_s * right;
} node_t;
typedef struct node_s {
node_t * root;
}
typedef node_t * nodep_t;
```

以 data 大小為依據，左子樹小於等於根，右子樹大於根，建立二元樹。

```
//插入資料進二元樹
(1)void Insert(nodep_t *node, student_t key);
//中序巡訪印出：左中右
(2)void Inorder(nodep_t *node);
```

輸入說明：

```
p function(2)
i function(1)
5 欲插入的數字
p
i
6
p
i
7
p
i
```

3
p
i
4
p
e　結束輸入

輸出說明：

p (印出):
(1)沒有 tree 印出 null
(2)第一個插入的數字為 root
(3)小於、等於 root 為左子樹
(4)大於 root 為右子樹
3 4 5 6 7 每個 data 中間有空白

Sample Input:

p
i
5
p
i
6
p
i
7
p
i
3
p
i
4
p
i
2
p
i
6

p
e

Sample Output:

null
5
5 6
5 6 7
3 5 6 7
3 4 5 6 7
2 3 4 5 6 7
2 3 4 5 6 6 7

~~~~~~
week 15

028. Queue 使用 Link list

功能說明：

1. CreateQueue
   建立一個 Queue，並給予 Queue 的名稱。

2. EnQueue
   輸入一個 Queue 的名稱，並插入資料 (int)，
   若 Queue 的名稱不存在，則 printf("Queue %s isn't exist\n", name)。

3. DeQueue
   輸入一個 Queue 的名稱，並進行 Dequeue，
   若 Queue 的名稱不存在，則 printf("Queue %s isn't exist\n", name)，
   若 queue 為空，輸出 Queue is empty。
   若沒有錯誤，只需取出數值，不用輸出。

4. MergeQueue
   輸入兩個 Queue 的名稱 A ,B，進行 QueueA 與 QueueB 相接(B 接到 A)，連接完畢後，刪除 QueueB。
   若 Queue 的名稱不存在，則 printf("Queue %s isn't exist\n", name)
~~~~~~

5. PrintAllQueue
依序印出所有的Queue, Queue size, Queue element
printf("**\n");
printf("Queue Name:%s Queue Size:%d Queue Element:", queue->name, count);
printf("**\n");
若不存在任何Queue，則printf("Isn't have any queue\n");
如果queue有名字 但是queue裡面是空的話 要printf("Queue is empty\n");
6. Exit

Sample Input
5
1
AAA
3
AAA
1
BBB
1
CCC
1
DDD
1
EEE
1
FFF
2
AAA
1
2
AAA
3
2
AAA
4
2
AAA
5
2
AAA

6
2
BBB
7
2
BBB
8
2
BBB
9
2
BBB
10
2
CCC
11
2
CCC
12
2
CCC
13
2
CCC
14
2
CCC
15
2
DDD
16
2
DDD
17
2
DDD
18
2
DDD
19

2
DDD
20
2
EEE
21
2
EEE
22
2
EEE
23
2
EEE
24
2
EEE
25
2
FFF
26
2
FFF
27
2
FFF
28
2
FFF
29
2
FFF
30
5
4
AAA
FFF
5
4
BBB

CCC
5
4
AAA
BBB
5
6

Sample Output

```
**************************************
Isn't have any queue
**************************************
Queue is empty
**************************************
Queue Name:AAA Queue Size:5 Queue Element:1 3 4 5 6
Queue Name:BBB Queue Size:4 Queue Element:7 8 9 10
Queue Name:CCC Queue Size:5 Queue Element:11 12 13 14 15
Queue Name:DDD Queue Size:5 Queue Element:16 17 18 19 20
Queue Name:EEE Queue Size:5 Queue Element:21 22 23 24 25
Queue Name:FFF Queue Size:5 Queue Element:26 27 28 29 30
**************************************
**************************************
Queue Name:AAA Queue Size:10 Queue Element:1 3 4 5 6 26 27 28 29 30
Queue Name:BBB Queue Size:4 Queue Element:7 8 9 10
Queue Name:CCC Queue Size:5 Queue Element:11 12 13 14 15
Queue Name:DDD Queue Size:5 Queue Element:16 17 18 19 20
Queue Name:EEE Queue Size:5 Queue Element:21 22 23 24 25
**************************************
**************************************
Queue Name:AAA Queue Size:10 Queue Element:1 3 4 5 6 26 27 28 29 30
Queue Name:BBB Queue Size:9 Queue Element:7 8 9 10 11 12 13 14 15
Queue Name:DDD Queue Size:5 Queue Element:16 17 18 19 20
Queue Name:EEE Queue Size:5 Queue Element:21 22 23 24 25
**************************************
**************************************
Queue Name:AAA Queue Size:19 Queue Element:1 3 4 5 6 26 27 28 29 30 7 8 9 10 11
12 13 14 15
Queue Name:DDD Queue Size:5 Queue Element:16 17 18 19 20
Queue Name:EEE Queue Size:5 Queue Element:21 22 23 24 25
```

****