

計算機技術作業#1 -- 請使用 C 語言撰寫下列程式:

1. 請寫出下列輸入輸出格式，輸入名字(10 個字元以內的字串)、系所代碼(單一字元)、和兩次小考成績(整數)後，輸出如下面例子中的格式，名字和系所代碼需靠左，平均分數需表示到小數點以下兩位，其餘欄線需對齊。

```
Please input your name... Kevin
Please select your department... <a:資工 b:電機 c:電子> a
Please input your score on test 1... 90
Please input your score on test 2... 85
```

Name	Department	Test 1	Test 2	Average
Kevin	a	90	85	87.50

2. 請輸入一個整數後，執行下列各種不同情況的判斷與輸出。

- 當所輸入整數非正整數時
Please input a positive integer: -10 ↵
-10 is not a positive integer!!
- 當所輸入整數介於 1~9 時
Please input a positive integer: 7 ↵
7 is a small integer.
- 當所輸入整數介於 10~99 時
Please input a positive integer: 89 ↵
89 的十位數為 8
89 的個位數為 9
- 當所輸入整數大於或等於 100 而且此整數為 5 的倍數時
Please input a positive integer: 105 ↵
105 is a large integer.
105 is divisible by 5.
- 當所輸入整數大於或等於 100 而且此整數除以 5 餘 1 時
Please input a positive integer: 106 ↵
106 is a large integer.
106+5=111
- 當所輸入整數大於或等於 100 而且此整數除以 5 餘 2 時
Please input a positive integer: 107 ↵
107 is a large integer.
107-5=102
- 當所輸入整數大於或等於 100 而且此整數除以 5 餘 3 時
Please input a positive integer: 108 ↵
108 is a large integer.
108*5=540
- 當所輸入整數大於或等於 100 而且此整數除以 5 餘 4 時(請顯示到小數點以下兩位)
Please input a positive integer: 109 ↵
109 is a large integer.
109/5=21.80

計算機技術作業#2 -- 請使用 C 語言撰寫下列程式:

1. 請輸出如下圖的九九乘法表。

1*1=1	2*1=2	3*1=3	4*1=4	5*1=5	6*1=6	7*1=7	8*1=8	9*1=9
1*2=2	2*2=4	3*2=6	4*2=8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
1*3=3	2*3=6	3*3=9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
1*4=4	2*4=8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
1*5=5	2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
1*6=6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
1*7=7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
1*8=8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
1*9=9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

2. 請輸入 n 個整數，計算它們的平均值並找出最大值與最小值。

Please input the number of integers: 4

Integer #1 30

Integer #2 14

Integer #3 42

Integer #4 19

The average is 26.25

The maximum is 42

The minimum is 14

3. 請輸出如下圖中任意高度的數字金字塔。

Please input a number: 13

```

1
222
33333
4444444
555555555
66666666666
7777777777777
888888888888888
99999999999999999
0000000000000000000
11111111111111111111
222222222222222222222
33333333333333333333333
    
```

<<提示: for()

```

{
    for()
        {印空格}
    for()
        {印數字}
}
    
```

計算機技術作業#3 -- 請使用 C 語言撰寫下列程式:

1. 請輸入兩個整數 n1 和 n2 ($n1 \leq n2$)，計算從 n1 累加到 n2 的總和。需判斷 n1 的值是否大於 n2，並且可以重覆輸入執行。其輸入輸出方式如下：

```
Number 1: 20 ↵
Number 2: 10 ↵
Error!! Number 1 > Number 2!!
Number 1: 1 ↵
Number 2: 100 ↵
Sum = 5050
Again(Y/N)?? y ↵
Number 1: 30 ↵
Number 2: 40 ↵
Sum = 385
Again(Y/N)?? n ↵
```

2. 費氏數列(費波那契數列)為 0,1,1,2,3,5,8,13,21,34,55,...，第一個數為 0，第二個數為 1，其它的數為前面兩個數的和，請輸入一個正整數 N，列出前面 N 個費氏數列的值。

Please input a number: 10 ↵

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

```
P2 P1
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
P2 = P1 + item
```

3. 請輸入一個正整數 N，從 1,2,3,...,N 的連續 N 個整數中，找出任意的 4 個不相同整數 n1、n2、n3、n4，使得 $N = n1 + n2 + n3 + n4$ ，此種 (n1,n2,n3,n4) 組合可能會有許多組，舉例來說，當 N=13 時，可以找到 (1,2,3,7)、(1,2,4,6)、(1,3,4,5) 三種組合。請輸出當整數值為 N 時，會有幾種不同組合，其輸出方式如下：

N = 13 ↵
Total is 3.

N = 231 ↵
Total is 80142.

4. 請輸出如下圖中任意高度的數字金字塔。

Please input a number: 13

```

1
121
12321
1234321
123454321
12345654321
1234567654321
123456787654321
12345678987654321
1234567890987654321
123456789010987654321
12345678901210987654321
1234567890123210987654321
```

計算機技術作業#4 -- 請使用 C 語言撰寫下列程式:

- 質數(prime number)的定義為:除了1與本身以外,沒有其他因數的自然數,如2,3,5,7,...等皆是質數,但1不是質數。請輸入任意一個自然數N,列出此範圍(1~N)中所有的質數,並算出從1至N中會出現質數的總次數。

Please input a number: 40

2 3 5 7 11 13 17 19 23 29 31 37

Total is 12

```
<<提示: for (j = 2; j <= i; j++)
{
    // 檢查從2到i-1是否可以被i整除,來判斷i是否為質數
    // 只要可以整除,則 break; 離開for迴圈。
    if (j > i-1)
    {
        // 如果for迴圈沒有被 break; 中斷, j++累加到 j > i-1 才離開
        // ,代表從2到i-1都無法被i整除,則i為質數。
        // 其實只需要檢查從2到√i是否可以被i整除即可。j * j <= i。
    }
}
```

- 請輸入任意兩個正整數,再利用輾轉相除法求出此兩個正整數的最大公因數。其輸入的過程如下:

Please input two integers: 210 396

GCD is 6

1	210	396	1
	186	210	
1	24	186	7
	18	168	
	6	18	3
		18	
		0	

- 雞兔同籠問題:一個大籠子中只有養雞(2隻腳)和兔子(4隻腳),已知雞和兔子的總數量和總腳數,請分別找出雞的數量和兔子的數量。本題可以重複輸入執行,直到輸入0+0結束執行。輸出如下:

總數量+總腳數: 252+872

雞 68 隻 兔子 184 隻

總數量+總腳數: 252+504

雞 252 隻 兔子 0 隻

總數量+總腳數: 252+685

無解

總數量+總腳數: 0+0

- X 數列由 n 個正整數所組成 $(a_1, a_2, a_3, \dots, a_n)$, 此數列滿足下列條件:

(1) 如果前一個數 a_i 為偶數, 則下一個數 $a_{i+1} = a_i / 2$,

(2) 如果前一個數 a_i 為奇數, 則下一個數 $a_{i+1} = 3 * a_i + 1$,

(3) 此數列的最後一個數一定會收斂至 $a_n = 1$ 。

例如, 若 $a_1 = 22$, 則此 X 數列為 $(22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1)$, 且數列總合為 $22+11+34+17+52+26+13+40+20+10+5+16+8+4+2+1=281$ 。本題輸入兩個正整數 LowerBound 和 UpperBound, 請找出介於此兩個正整數中(包含 LowerBound 和 UpperBound), 哪個正整數 N 作為 a_1 (X 數列的第一個數) 可以得到最大的數列總合 Sum, (若同時有兩個正整數 N_1 和 N_2 都是得到最大的數列總合 Sum, 則只要輸出整數值較小的 N_i 即可)。其輸入的過程如下:

Please input the LowerBound and the UpperBound: 22 23

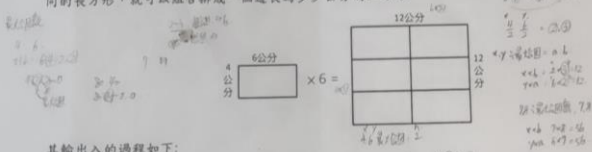
N=23 Sum=633

Please input the LowerBound and the UpperBound: 30 100

N=63 Sum=105098

計算機技術作業#5 -- 請使用 C 語言撰寫下列程式:

1. 數個相同長方形可以組合排成一個正方形，例如下圖中，六個長 4 公分、寬 6 公分的長方形可以組合排成一個邊長為 12 公分的正方形。本題輸入長方形的長和寬(單位公分)，請輸出最少需要幾個相同的長方形，就可以組合排成一個邊長為多少公分的正方形。



其輸入的過程如下:

請輸入長方形的長(公分)為...1386

請輸入長方形的寬(公分)為...3432

最少需要 1092 個相同的長方形，排成邊長為 72072 公分的正方形

2. 質數的定義為：除了 1 與本身以外，沒有其他因數的自然數，如 2, 3, 5, 7, ... 等皆是。任何一對相差 2 的質數在數學上稱為質數雙胞胎(Twin-Prime)，常常被用來做一些特殊的數值計算研究。例如，(3, 5)、(5, 7)、(11, 13)等皆為質數雙胞胎，請注意 1 不是質數。請輸入任意一個自然數 N，求出從 1 至 N 中會出現質數雙胞胎的次數，並列出此範圍中所有的質數雙胞胎。

Please input a number: 100

(3,5) (5,7) (11,13) (17,19) (29,31) (41,43) (59,61) (71,73)

Total is 8

3. 醜數(ugly number)的定義為：1 是最小的醜數，其餘任何一個正整數若其質因數只有 2、3、或 5 所組成，則也為醜數。請輸入任意一個自然數 N 後，由小到大列出 N 個醜數。

Please input a number: 30

1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32, 36, 40, 45, 48, 50, 54, 60, 64, 72, 75, 80

<提示：任何一個 temp 整數可以利用 while(temp % 2 == 0)



把 2 的因數全部除光

4. 請寫出一個猜數字遊戲，先使用亂數產生四個 0-9 之間互不相等的數字答案(例如 5924)，再由玩家任意輸入四個數字來猜測，如果相同位置的數字相同可得 1A，不同位置的數字相同可得 1B，例如若玩家輸入 2947 可獲得 1A2B，請使用迴圈讓玩家一直猜到正確(即 4A)為止，並計算玩家共猜了幾次才猜中。其輸入的過程如下:

Please input four digits (0-9): 9415

The result is 1A 2B

Please input four digits (0-9): 9543

The result is 2A 2B

Please input four digits (0-9): 9534

You got it!! 3 times!!

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
```

```
{
    int n;
    srand(time(NULL));
    n = rand() % 10;
```

<提示：亂數產生的函數名稱為 rand()，n=rand()%10; 可產生一個 0-9 的亂數，使用亂數產生函數前需先執行 srand(time(NULL)); 指令，利用目前時間來產生一個新的亂數序列，否則產生的亂數永遠都一樣，另外使用 rand()、srand()、time() 等函數需 include stdlib.h 和 time.h。連續輸入四個單一數字可以使用 scanf("%d%d%d%d",&a,&b,&c,&d); 取得四個變數值

計算機技術作業#6 -- 請使用 C 語言撰寫下列程式:

1. 請寫出兩正方形矩陣(N*N)相乘的程式, 首先輸入正方形矩陣的大小 N (N≤10), 輸入兩個整數正方形矩陣後, 再輸出兩正方形矩陣相乘的結果(N*N)。其輸出的過程如下:

```

Please input the size of matrix: 5
Please input matrix1:
1 2 3 4 5
6 7 8 9 0
5 4 3 2 1
0 9 8 7 6
2 4 6 8 0
Please input matrix2:
0 0 6 4 2
1 3 5 7 9
9 7 5 3 1
1 2 3 4 5
5 4 3 2 1
matrix1 * matrix2 =
58 63 58 53 40
88 143 128 133 128
58 81 74 67 68
118 121 124 127 130
66 86 86 86 86
    
```

$$\begin{aligned}
 & \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 0 \\ 5 & 4 & 3 & 2 & 1 \\ 0 & 9 & 8 & 7 & 6 \\ 2 & 4 & 6 & 8 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 6 & 4 & 2 \\ 1 & 3 & 5 & 7 & 9 \\ 9 & 7 & 5 & 3 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix} \\
 & \begin{bmatrix} 58 & 63 & 58 & 53 & 40 \\ 88 & 143 & 128 & 133 & 128 \\ 58 & 81 & 74 & 67 & 68 \\ 118 & 121 & 124 & 127 & 130 \\ 66 & 86 & 86 & 86 & 86 \end{bmatrix}
 \end{aligned}$$

<<提示: 可以先輸入正方形矩陣的大小 n, 再宣告二維陣列 int m1[n][n], 以 5*5 矩陣為例, 相乘結果的 m1[2][1] = m1[1][0]*m2[0][1] + m1[1][1]*m2[1][1] + m1[1][2]*m2[2][1] + m1[1][3]*m2[3][1] + m1[1][4]*m2[4][1] >>

2. 請寫出一個程式來驗證信用卡卡號的檢查碼是否正確, 根據信用卡卡號的編碼規則, 一般信用卡的卡號為 15 或 16 碼, 其中最左邊的開始幾個數碼代表不同卡別 (例如第一碼為 4 代表 Visa Card, 前兩碼介於 51 至 55 則代表 Master Card), 而最右邊一碼即為檢查碼, 檢查碼的驗證方法如下:
- (1) 將信用卡的每個數碼設定權重: 從右向左, 檢查碼除外, 每個數碼的權重分別是 2、1、2、1、2、1... 交錯 (若信用卡有 16 碼, 那麼最左邊數碼的權重是 2; 若卡號只有 15 碼, 那麼最左邊數碼的權重就是 1)。
 - (2) 將每個數碼乘上權重, 所得的加權數字若小於或等於 9, 則此數碼的輸出數即為此加權數字; 反之若所得加權數字大於 9, 則此數碼的輸出數為將加權數字的個位數和十位數相加。
 - (3) 將所有數碼的輸出數全部加總起來, 並取加總後的個位數為餘數。
 - (4) 卡號的檢查碼即為 10 減掉此餘數後所得之值 (但餘數為 0 時除外, 餘數為 0 時, 檢查碼仍為 0)。

例如, 若某一張 Visa 信用卡的卡號是 4311-4656-0640-6131, 則其計算過程如下:

卡號	4	3	1	1	4	6	5	6	0	6	4	0	6	1	3	1
(1) 權重	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	x
(2) 加權數字	8	3	2	1	8	6	10	6	0	6	8	0	12	1	6	x
							=>1						=>3			
(3) 計算總和	59															
(4) 計算檢查碼	59 的個位數是 9, 所以檢查碼是 10 - 9 = 1, 正確!															

本題可以重複多次輸入卡號進行檢查, 若輸入 15 碼或 16 碼的卡號數字, 會判斷輸出 "卡號正確!!" 或 "卡號錯誤!!"; 若輸入非 15 碼或 16 碼的卡號, 或是輸入非數字的字元, 會出現 Error 錯誤訊息, 並結束程式。

請輸入卡號: 713257498001340

卡號正確!!

請輸入卡號: 12345678901234a

Error! 非 15 或 16 個卡號數字!

<<提示: 由於卡號數量不固定, 可以利用字串輸入, 宣告字元陣列 char input[17], 再以 scanf("%s", input), 輸入字串。然後再將字元轉整數, 可以利用 (int) input[i] - (int) '0' 或 (int) input[i] - 48 來將字元轉整數。n = strlen(input); 可以傳回字串長度給 n 變數, 但需先 #include <string.h> >>

計算機技術作業#7 -- 請使用 C 語言撰寫下列程式:

1. 請寫出一個二進位與十進位轉換的程式，首先輸入十進位的正整數 D，輸出 D 的 unsigned 二進位格式於螢幕上，然後輸入任意位數的 unsigned 二進位數 B，再輸出 B 的十進位格式於螢幕上。其輸入的過程如下：

Please input a decimal number: 723

Please input a binary number: 11001011

203

2. 兩個正整數 A 和 B 之間的所有連續正整數中 (包含 A 和 B, 且 $1 \leq A \leq B \leq 10^7$), 請計算每個 0-9 數字各出現了幾次。例如 A=7 且 B=13 時 (也就是 7, 8, 9, 10, 11, 12, 13), 每個 0-9 數字各出現的次數為 1, 1, 1, 0, 0, 0, 1, 1, 1, 1。其輸出格式如下:
- $$\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

请输入 A:B..... 7:13 ↵

0-9 數字的出現次數為..... 1, 5, 1, 1, 0, 0, 0, 1, 1, 1

请输入 A:B..... 4000:9876543 ↵

请输入 A:B..... 4000:9876543 J

6935248, 6905248, 6705248

3. 猜出一個程式，可以檢查一個字串 string2 是否為另一個字串 string1 的一部份(子字串)，以及共包含幾次子字串。本題字串的長度不超過 100 個字元，其輸出入格式如下：

Please input the string: abcdabcdabdbbbb

Please input the string 2: bcd ↵

String 2 is a substring of string 1

Repeat 3 times

Please input the string 1: abcdefg ↵

Please input the string 2: dd ↵

String 2 is not a substring of string 1

Please input the string 1: abbababababbabba .

Please input the string 2: aba

String 2 is a substring of string 1

Repeat 4 times

4. 丹布朗的「達文西密碼」一書中提到以下的變位字謎：

O, Draconian devil! Oh, lame saint! (啊！嚴峻的魔鬼！啊！跛足的聖人！)

其隱含的雙位字為 Leonardo da Vinci! The Mona Lisa! (李奧納多達文西！蒙娜麗莎！)

變位字是將一段文字裡面的相同字母重新排列成另一段意義不同的文字，標點符號如逗號、句號、連字號、空格、驚嘆號、問號等可以不相等，而且變位字只要字母相同即可，大小寫也可以不同。舉例來說，“Evil eye.”和“Eye-Live!”兩者都是變位字，但“Abba”和“Aba”兩者則不是變位字，因為第二段文字少了一個b。本題請輸入兩個字串，每個字串的长度不超過 50，檢查此兩個字串是否為變位字，若是變位字則輸出 Yes，若不是變位字則輸出 No。

```
Please input string #1: O, Draconian devil! ٧
```

Please input string #2: Leonardo da Vinci ↵

Yes

<提示: 如果輸入字符串中有包含空格, 可利用 `scanf("%[^\n]",str)`; 取代 `scanf("%s",str)`, 進行輸入。
`%[^\n]` 代表接收除了 `\n` (跳行字元) 以外的所有字元。`n=strlen(str)`, 可以傳回 `str` 字符串的長度給 `n` 變數, 但需將 `#include <string.h>>>`

計算機技術作業#8 -- 請使用 C 語言撰寫下列程式:

1. 一個簡單的整數加密與解密方法，是將 32 bits (4 bytes) 整數中每個 byte 的前 4 個 bits 與後 4 個 bits 對調，則使用同一個程式就可以進行加密和解密。舉例來說，
(1) 十進制整數 1988 的十六進制為 00 00 07 C4，則 1988 加密後的十六進制為 00 00 70 4C，因此，加密後得到的十進制整數為 28748，再執行一次又可以將 28748 解密為 1988。
(2) 十進制負數 -1988 的十六進制為 FF FF F8 3C，加密後為 FF FF 8F C3，轉換十進制整數為 -28733，同一個程式又可以將 -28733 解密為 -1988。
(3) 十進制整數 135285699 的十六進制為 08 10 4B C3，加密後為 80 D1 B4 3C，轉換十進制整數為 -2147371972，同一個程式又可以將 -2147371972 解密為 135285699。

本題請輸入一個十進制整數，然後輸出其加密後的十進制整數，輸出的過程如下:

```
Please input a decimal number: 1988 ↵
28748
Please input a decimal number: -1988 ↵
-28733
Please input a decimal number: 135285699 ↵
-2147371972
```

<<提示: C 語言 32 bits 整數的表示範圍為 $-2^{31} \sim 2^{31}-1$ ($-2147483648 \sim 2147483647$)，負數的二進制是以二補數(2's complement)來表示，亦即，將對應正數的每個位元取補數(0 和 1 互變)後再加 1。例如，1988 的二進制為 0000 0000 0000 0000 0111 1100 0100 (即 00 00 07 C4)，則 -1988 的二進制為 1111 1111 1111 1111 1000 0011 1011 + 1 (即 FF FF F8 3C)。反之，正數也是將相對應負數的每個位元取補數後再加 1，例如 FF FF F8 3C 取補數 00 00 07 C3 + 1 >>

2. 迴文是指在一串字元中，由前面讀到後面的字元出現順序與從後面讀到前面的字元出現順序一樣。例如，a3BBcBB3a 是一串迴文，而 aa3baab3acc 不是一串迴文，不過，aa3baab3acc 字串中仍然可以找出最長的迴文字子串為 a3baab3a，其最長迴文字子串的长度為 8。迴文長度可以為偶數或奇數，例如 AbbA 為偶數长度的迴文，aBa 則為奇數长度的迴文。迴文长度最少為 1，例如 a 也可以視為长度 1 的迴文。一個字串中可能會包含多個长度相同的最長迴文字子串，例如 7BCBCdWWWaTNT 字串中包含四個长度為 3 的最長迴文字子串，包括 BCB、CBC、WWW、和 TNT。本題請輸入一個长度小於 100 個字元的字串後，列出此字串中所有的最長迴文，字串由大小寫的英文字母和數字所組成，且大寫與小寫的英文字母視為不同，例如 aBbA 不是迴文。其輸出的過程如下:

```
請輸入一個字串: 7BCBCdWWWaTNT ↵
最長迴文為: BCB, CBC, WWW, TNT
```

<<提示: 可以宣告一個二維陣列 `int list[100][2]`; 來儲存最多 100 個相同长度的最長迴文之開始位置 `list[i][0]` 與結束位置 `list[i][1]`。 `n=strlen(input)`; 可傳回字串长度給 `n` 變數，需先 `#include <string.h>` >>

3. 質數的定義為: 除了 1 與本身以外，沒有其他因數的自然數，如 2, 3, 5, 7, ... 等皆是。非質數的自然數稱為合成數。請輸入任意一個自然數 `N`，求從 1 至 `N` 中會出現連續四個數均為合成數的次數，同一數字可以出現在不同的組合中，例如 (24, 25, 26, 27) 及 (25, 26, 27, 28) 皆有 25, 26, 27 等三個數字。

```
Please input a number: 100 ↵
```

```
(24, 25, 26, 27) (25, 26, 27, 28) (32, 33, 34, 35) (33, 34, 35, 36) (48, 49, 50, 51) (49, 50, 51, 52)
(54, 55, 56, 57) (55, 56, 57, 58) (62, 63, 64, 65) (63, 64, 65, 66) (74, 75, 76, 77) (75, 76, 77, 78)
(84, 85, 86, 87) (85, 86, 87, 88) (90, 91, 92, 93) (91, 92, 93, 94) (92, 93, 94, 95) (93, 94, 95, 96)
Total is 18
```


計算機技術作業#9 -- 請使用 C 語言撰寫下列程式:

- 請寫出遞迴程式來解決河內塔問題(Towers of Hanoi)，共有 A,B,C 三個柱子，一開始 A 柱子可以放任意個數的環，依序由上到下的編號分別為 1~N，每個環的大小都不一樣，編號越大的環越大，假設每次只能從柱子的頂端移動一個環，搬到其他柱子放，而且較大的環永遠都不能放在較小環的上方，請顯示最少共需幾個步驟以及如何把所有的環從 A 柱子搬到 C 柱子。其輸入的過程如下：(共 4~C)

請輸入 A 柱中的環個數: 3 ↵

ring 1: A => C ↵

ring 2: A => B ↵

ring 1: C => B ↵

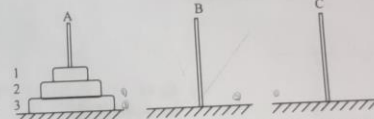
ring 3: A => C ↵

ring 1: B => A ↵

ring 2: B => C ↵

ring 1: A => C ↵

共需 7 個步驟



<<提示：寫一個 void towers(int n, char from, char aux, char to) 遞迴函數，可將 n 個環從 from 柱搬到 to 柱，

aux 為中間暫時存放的柱，主函數 main() 中可用 towers(n, 'A', 'B', 'C'); 來開始執行遞迴呼叫。>>

- 將字串 abc 的字元進行任意排列，可以有 abc、acb、bac、bca、cab、cba 等 6 種排法，而字串 abbc 則可以有 abbc、abcb、acbb、babc、bacb、bbac、bbca、bcab、bcba、cabb、cbab、cbba 等 12 種排法，本題請輸入任一字串(字串長度不超過 12)，計算此字串任意排列的排法數目。

Please input a string: abcba ↵

The number of permutations is 30

Please input a string: 1112222334 ↵

The number of permutations is 138600

<<提示：字串任意排列的排法數目為 $\frac{n!}{m_1!m_2! \dots m_k!}$ ，n 為字串總長度， m_i 為字串中同一個相同字元的數量，可

以寫一個 int fact(int n) 階層函數以重複使用，階層函數內的階層計算使用迴迴會比遞迴有效率。>>

- 請寫出一個數字地雷遊戲，遊戲規則如下：首先由電腦亂數隨機產生一個正整數 N， $10 \leq N \leq 40$ 於螢幕上，然後由電腦與玩家輪流累加數字，最少需加 1，最多可加 3，累加到 N 者為輸家，每次都由電腦從 0 開始往上累加。例如：當 N=12，電腦與玩家比賽過程可能如下：

N=12

Sum=0, Computer add 3

Sum=3, You add: 2 ↵

Sum=5, Computer add 2

Sum=7, You add: 3 ↵

Sum=10, Computer add 1

Sum=11, You add: 1 ↵

Sum=12, You lose!!

請撰寫一個智慧型程式，讓電腦可以擁有絕大部分的勝算來贏玩家。

<提示：不管玩家每次累加多少數目，電腦都可以累加成為 4，(3+1=4, 2+2=4, 1+3=4)，所以電腦只要留 4 的倍數加 1 給玩家就保證一定贏。亂數產生的函數名稱為 rand()，使用亂數產生函數前需先執行 srand(unsigned(time(NULL))); 指令，且需 include stdlib.h 和 time.h >

計算機技術作業#10 請使用 C 語言撰寫下列程式 (本次作業規定都必須使用指標)

1. 請利用指標(pointer)寫出一個排序函式 void sort(int *arr, int n)，利用泡沫排序法(bubble sort)的 swap 方式將參數 arr 陣列中的 n 個整數由小排到大。本題先在 main() 中輸入 n 個整數，存到 num 陣列，然後呼叫排序函式 sort(num, n)，將此 n 個整數的 num 陣列排序，最後在 main() 中依照由小至大的順序顯示出來。其輸入輸出如下：

```
請輸入整數的個數: 10 ↵
請輸入第 1 個數: 5 ↵
請輸入第 2 個數: -2 ↵
請輸入第 3 個數: 30 ↵
請輸入第 4 個數: 1 ↵
請輸入第 5 個數: 28 ↵
請輸入第 6 個數: -6 ↵
請輸入第 7 個數: 7 ↵
請輸入第 8 個數: 5 ↵
請輸入第 9 個數: 1 ↵
請輸入第 10 個數: 22 ↵
排序結果: -6 -2 1 1 5 5 7 22 28 30
```

2. ISBN 國際標準書碼分為 10 碼和 13 碼兩種，其檢查書碼正確性的方式不同，分別如下所示：

ISBN 10 碼：

每個碼皆為 0-9 的值，但最後一個碼可以是 X，若為 X 則代表 $X=10$ 。將 1-10 位數的碼分別乘以權重 10-1，加總得到總和 sum，檢查 sum 是否為 11 的倍數，若為 11 的倍數即為正確書碼，以 013162959X 為例。

ISBN 10 碼	0	1	3	1	6	2	9	5	9	X
(1) 權重	10	9	8	7	6	5	4	3	2	1
(2) 加權數字	0	9	24	7	36	10	36	15	18	10 (X=10)
(3) 計算總和	sum=165									
(4) 檢查方式	165 為 11 的倍數，正確!									

ISBN 13 碼：

13 碼的 ISBN 不會出現 X 的編碼，取前面 12 個碼乘以權重，奇位數的碼權重為 1，偶位數的碼權重為 3，加總得到總和 sum，第 13 碼為檢查碼，必須等於 10 減掉 sum 的個位數所得之值，即 $10-(sum\%10)$ ，但若 sum 的個位數為 0 時除外，sum 的個位數為 0 時，檢查碼仍為 0，以 9789861817286 為例。

ISBN 13 碼	9	7	8	9	8	6	1	8	1	7	2	8	6
(1) 權重	1	3	1	3	1	3	1	3	1	3	1	3	檢查碼
(2) 加權數字	9	21	8	27	8	18	1	24	1	21	2	24	檢查碼
(3) 計算總和	sum=164												
(4) 檢查方式	164 的個位數是 4，所以檢查碼是 $10-4=6$ ，正確!												

本題請寫出 void check_ISBN10(char *arr) 和 void check_ISBN13(char *arr) 兩個函數，參數傳入 arr 陣列來檢查是否為正確的 ISBN 書碼，並直接在螢幕上印出正確或錯誤(沒有回傳值)。主函數 main() 中可宣告字元陣列 char input[14]，並以 scanf("%s", input); 輸入書碼，再以 strlen(input) 判斷書碼長度來呼叫適當函數。本題可以重覆多次輸入書碼進行檢查，若輸入非 10 碼或 13 碼，會出現 Error 訊息並結束程式。其輸入輸出的過程如下：

```
Please input an ISBN code: 3880531137 ↵
The ISBN code is correct!!
Please input an ISBN code: 9788175257610 ↵
The ISBN code is correct!!
Please input an ISBN code: 572314786X ↵
The ISBN code is wrong!!
Please input an ISBN code: 123 ↵
Error!
```

計算機技術作業#11 -- 請使用 C 語言撰寫下列程式:

1. Farey 數列: 將所有介於 0 與 1 之間的分數, 其分母必須小於(或等於)某一自然數 N 的所有已約分數, 由小到大順序排列成一排, 再加上 0 和 1, 稱之為第 N 階的 Farey 數列。請利用指標(pointer)寫出一個分數排序函式 void sort_fraction(int (*ptr)[2], int n), 利用泡沫排序法(bubble sort)的 swap 方式將參數二維陣列中的 n 個分數由小排到大。並請寫出一個求最大公因數的函式 int gcd(int n1, int n2), 利用輾轉相除法求出自傳 $n1$ 與 $n2$ 的最大公因數。此程式可由鍵盤輸入任意自然數 N ($N \leq 20$), 由螢幕輸出其第 N 階的 Farey 數列, 以及共幾個數(含 0 和 1)。其輸入的過程如下:

Please input a number: 5
 0, 1/5, 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 1 (total numbers: 11)
 Please input a number: 10
 0, 1/10, 1/9, 1/8, 1/7, 1/6, 1/5, 2/9, 1/4, 2/7, 3/10, 1/3, 3/8, 2/5, 3/7, 4/9, 1/2, 5/9, 4/7, 3/5, 5/8, 2/3, 7/10, 5/7, 3/4, 7/9, 4/5, 5/6, 6/7, 7/8, 8/9, 9/10, 1 (total numbers: 33)

2. 請算出任一個世紀有多少個黑色星期五(每月 13 日且為星期五)。世紀的算法為元年至百年, 例如 19 世紀為 1801 年至 1900 年, 20 世紀為 1901 年至 2000 年, 而 21 世紀為 2001 年至 2100 年。

Please input the century: 21
 172 times!!
 <<提示: 每年第一天星期幾的計算公式 $week = (y + (int)((y-1)/4) - (int)((y-1)/100) + (int)((y-1)/400)) \% 7$; y 為年份, 例如 $y=1901$ 時, $week=2$, 代表 1901/1/1 為星期二, $y=2006$ 時, $week=0$, 代表 2006/1/1 為星期日。閏年的計算方式, 可以被 4 整除的那一年為閏年, 百年不算是閏年, 但可以被 400 整除的那一年又算是閏年, 例如: 1896 是閏年, 1900 不是閏年, 2000 是閏年。閏年的 2 月有 29 天。某個月的 1 日如果是星期日, 則當月會有黑色星期五, 因為 13 日算法為 $(0+12)\%7=5$ >>

3. 某個汽車接送服務公司提供四種車型的機場接送服務, 分別是(1)四門房車: 可接送 4 位乘客, 定價 2500 元; (2)商務休旅車: 可接送 7 位乘客, 定價 3800 元; (3)小型巴士: 可接送 13 位乘客, 定價 6400 元; 以及(4)中型巴士: 可接送 16 位乘客, 定價 7500 元。假設四種車型皆可不限數量提供服務, 本題請計算滿足各種乘客人數需求的最便宜訂車方案(總價格最低), 若有兩種以上訂車方案的價格相同且最低, 只需列出其中一種即可。例如當乘客總人數為 21 人時, 1 台中型巴士+1 台商務休旅車(可搭乘 23 人, 共 11300 元)為最便宜的訂車方案, 會比 1 台小型巴士+2 台四門房車(可搭乘 21 人, 共 11400 元)或 3 台商務休旅車(可搭乘 21 人, 共 11400 元)的價格低。其輸入格式如下:

請輸入乘客總人數: 105
 最少需要 50300 元
 四門房車 0 台
 商務休旅車 0 台
 小型巴士 2 台
 中型巴士 5 台

請輸入乘客總人數: 107
 最少需要 51300 元
 四門房車 1 台
 商務休旅車 1 台
 小型巴士 0 台
 中型巴士 6 台

105 = 21 * 5
 107 = 21 * 5 + 2

105 = 21 * 5
 107 = 21 * 5 + 2

計算機技術作業#12-- 請使用 C 語言撰寫下列程式:

- 請依次讀入 t1.txt 檔案中的字元，統計此檔案中各種聲音字元(A,E,I,O,U)各出現的次數，並將結果輸出到 t2.txt 檔案中。(請到學校網頁雲端學院的 1111 計算機技術課程公告下載 t1.txt 的測試資料)

t1.txt: The AEIOU is the upper case
of aeioa.

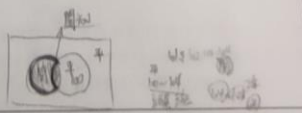
讀取 t1.txt
有 aeioa, upper

t2.txt: A E I O U
3 6 3 3 3

- 請依次讀入 t3.txt 檔案中的字串，統計此檔案中每個單字(字串)的出現次數，並將結果按照每個單字出現之先後順序輸出到 t4.txt 檔案中。檔案中的所有英文單字都為大寫，英文單字由空格、逗號、和句號分開，同一個單字的不同時態或單複數視為不同之單字。(請到雲端學院下載 t3.txt 的測試資料)

t3.txt: MACHINE LANGUAGES AND ASSEMBLY LANGUAGES ARE LOW LEVEL
LANGUAGES. C LANGUAGES, C++ LANGUAGES, AND JAVA LANGUAGES
ARE HIGH LEVEL LANGUAGES. LOW LEVEL LANGUAGES ARE CLOSER
TO THE HARDWARE THAN ARE HIGH LEVEL LANGUAGES.

t4.txt: MACHINE: 1
LANGUAGES: 9
AND: 2
ASSEMBLY: 1
ARE: 4
LOW: 2
LEVEL: 4
C: 1
C++: 1
JAVA: 1
HIGH: 2
CLOSER: 1
TO: 1
THE: 1
HARDWARE: 1
THAN: 1



<<提示: 宣告 char word[100][20]; 可以儲存 100 個(word[0]~word[99])長度最多為 19 個字元的單字，字串的結束符號為'\0'，strlen(字串變數) 可以回傳字串的長度，strcpy(字串 1, 字串 2) 可以將字串 2 的內容複製儲存到字串 1，strcmp(字串 1, 字串 2) 可以比較字串 1 和字串 2 的內容，當字串 1==字串 2 時會回傳 0 的值，但需先 #include <string.h>。>>

- 閏年是為了彌補人為訂定的紀年與地球公轉產生的差異，而規定某些年份的二月要多一天為 2 月 29 日，此西元年份即為閏年。閏年的判定方式如下：

- (1) 西元年份除以 400 可整除，為閏年。
- (2) 西元年份除以 4 可整除並且除以 100 不可整除，為閏年。
- (3) 西元年份除以 4 不可整除，為平年。
- (4) 西元年份除以 100 可整除並且除以 400 不可整除，為平年。

本題請輸入一個西元開始日期(yyyy/mm/dd)和一個西元結束日期(yyyy/mm/dd)，西元年份介於 1~2147483647 之間，請輸出這個開始到結束的期間內共有多少個 2 月 29 日，本題程式的執行有時限制，限定必須在 1 秒內完成執行結果。其輸入的過程如下：

請輸入西元開始日期(yyyy/mm/dd)為...2000/02/04
請輸入西元結束日期(yyyy/mm/dd)為...1987654312/08/12
此期間的 2 月 29 日共有 482005686 次

計算機技術作業#13-- 請使用 C 語言撰寫下列程式:

1. 請比對一篇文章 A 是否抄襲另一篇文章 B，並計算出相似度百分比。A 和 B 兩篇文章皆為英文，為簡化起見，兩篇文章的英文單字數量都小於 2000 個單字，且相似度百分比的計算採用下列規則：
 - (1) 英文單字可由空格、逗號、和句號分開，標點符號和大小寫都不影響相同單字的判斷，例如 the amount of time he jogs. 和 He jogs every morning. 視為重複了 he jogs，不會因為 H 大寫和句號而視為不同單字。
 - (2) 單字的不同時態或單複數則視為不同之單字，例如 jog 和 jogs 會視為不重複之單字。
 - (3) 若文章 A 有連續 7 個英文單字(含)以上與文章 B 重複，即視為相似句子，累計其重複單字的數量，6 個以下的重複單字則不計入數量。
 - (4) 若連續英文單字與文章 B 中有多處重複，則僅取重複最多單字的一次數量計入，並不會多次計入。
 - (5) 相似度百分比的計算公式為 (文章 A 中累計重複單字的總數量)/(文章 A 總單字數量)。

t10.txt 文章 A 中的 high level languages, low level languages are closer to the hardware 與 t11.txt 文章 B 中的 high level languages. Low level languages are closer to the hardware 重複了 11 個單字。Unlike high level languages, low level languages 則重複了 7 個單字，但因為其中的 high level languages, low level languages 6 個單字已經計入過重複單字的數量(在 11 個單字中)，只有 Unlike 1 個單字需要再累計數量，所以重複單字的總數量為 12 個單字。t10.txt 的文章 A 總單字數量為 21，因此相似度百分比為 $12/21 = 57.14\%$ 。(顯示到小數點以下兩位)。(請到雲端學院下載 t10.txt 和 t11.txt 的測試資料)

t10.txt: Machine languages and assembly languages are low level languages. Unlike high level languages, low level languages are closer to the hardware.

t11.txt: C languages, C++ languages, and Java Languages are high level languages. Low level languages are closer to the hardware than are high level languages. High-level languages are designed to simplify computer programming. Unlike high level languages, low level languages can be converted to machine code without using a compiler or interpreter.

輸出: $12/21 = 57.14\%$

<<提示: char word[2000][20] 可以儲存 2000 個(word[0]-word[1999])長度最多為 19 個字元的單字，字串的結束符號為 '\0'。strcpy(字串 1, 字串 2) 可以將字串 2 的內容複製儲存到字串 1，strcmp(字串 1, 字串 2) 當字串 1 == 字串 2 時會回傳 0 的值。(char)(int)ch+32 可將 ch 變數存放的大寫字元轉為小寫。>>

2. 文書編輯軟體通常會提供「word wrap」自動換行的功能，可依據文字編輯視窗的可視寬度，將超過可視寬度的英文句子截斷，顯示於下一行，同時保留句子中單字的完整性而不被切割。假設某一個文書編輯軟體的 word wrap 功能如下：(請到雲端學院下載 t12.txt，單字長度皆不超過 30 個字元)

- (1) 若英文句子的長度超過可視寬度，則該行盡量保留最多的完整單字和空格，其餘顯示於下一行。
 - (2) 若該行的最後一個單字剛好顯示填滿可視寬度，則下一行第一個字元為空格開始。
 - (3) 若單字之長度大於可視寬度，則此單字就必須被切割。當此單字為該行開始的第一個或第二個字元(第一個字元為空格)時，此單字顯示於該行，填滿該行的可視寬度後，單字剩下的部分顯示於下一行。當此單字不是該行的第一個或第二個字元，則此單字是從下一行再開始顯示。
- 請輸入可視寬度的大小後，讀入 t12.txt 檔案中的文章，再將 word wrap 結果輸出到 t13.txt 檔案中。

The width of word wrap: 12

t12.txt: A neighborhood is a geographically localized whole community within a larger city.

t13.txt: neighborhood is a geographically localized whole community within a larger city.

計算機技術作業#14 -- 請使用 C 語言撰寫下列程式:

1. Cool Word 是在一個單字(字串)中至少有兩種(含)以上的不同字元, 並且每個不同字元的出現次數都必須不同, 但同一個英文字母的大寫和小寫視為同一個字元。舉例來說, "Ada"這個字串的 'a' 出現 2 次, 'd' 出現 1 次, 兩種字元的出現次數不同, 因此 "Ada" 為 Cool Word。"BANANA" 這個字串也是 Cool Word, 因為 'a' 出現 3 次, 'n' 出現 2 次, 'b' 出現 1 次。而 "bbaccdd" 就不是 Cool Word, 因為 'a' 和 'd' 都是出現 1 次。本題請讀檔 t15.txt 輸入, 檔案中的列數不固定, 最多為 30 列, 每列有一個字串, 每個字串的長度最多為 30 個字元。請依序輸出 t15.txt 檔案中的每個字串是否為 Cool Word, 是 Cool Word 則輸出 Yes, 不是 Cool Word 則輸出 No。(請到雲端學院下載 t15.txt 檔案)

t15.txt: Ada bbaaccd 輸出: Yes
BANANA No
Mammal Yes
Needed Yes
Illness No
Papaya Yes
Referee Yes
A No
Senselessness Yes

<<提示: (char)(int)ch+32)
可將 ch 變數存放的大寫字
元轉為小寫。>>

2. 某零售業者想要找出所銷售商品之間的關聯性, 希望將關聯性最高的兩項商品放在一起以提高銷售量, 其銷售商品共 30 種, 商品代號分別為 1、2、3、...、30。假設共有 8 筆商品銷售記錄如下:

(1) 1 2 3 4 5 6
(2) 1 3 5 7 9
(3) 2 3 4
(4) 5 6
(5) 6 8 9
(6) 1 3 4
(7) 4 5 7
(8) 3 4 8 9

從這 8 筆銷售記錄可以發現 (3,4) 兩項商品的銷售組合為最多次 (4 次), 而 (1,3) 兩項商品的組合次之 (3 次), 其餘組合更少, 因此找出 (3,4) 為關聯性最高的兩項商品。請讀入 t9.txt 檔案中的資料, 第一列的整數為商品銷售記錄的筆數, 之後每一列為一筆銷售記錄, 每列的第一個整數為此筆銷售記錄的商品數量, 其後列舉的整數為此筆銷售記錄的商品代號(由小到大排列)。請輸出關聯性最高的兩項商品組合及出現次數, 若同時有多種組合的關聯性最高, 則全部組合都列出。(請下載 t9.txt 檔案)

t9.txt: 8
6 1 2 3 4 5 6
5 1 3 5 7 9
3 2 3 4
2 5 6
3 6 8 9
3 1 3 4
3 4 5 7
4 3 4 8 9

輸出: (3,4) 4 次

3. 在一個 $S_n = \{1, 2, \dots, n\}$ 的 n 連續序列中, 如果可以找到一個自然數 i 來分成兩組自然數集合, 並使得前後兩組自然數的和為相等, 也就是在 $S_n = \{1, 2, \dots, i-1, i, i+1, \dots, n-1, n\}$ 中, $1+2+\dots+(i-1) = (i+1)+\dots+(n-1)+n$, 則稱 i 為 n 連續序列 S_n 的數字中心。舉例來說, $S_8 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ 的數字中心為 6, 因為 $\{1, 2, 3, 4, 5\}$ 和 $\{7, 8\}$ 兩組自然數集合的和為相等, 都是 15。而 $S_4 = \{1, 2, 3, 4\}$ 則找不到數字中心, 因為無法分成前後兩組自然數的和為相等。本題請讀檔 t16.txt 輸入, 檔案中的列數不固定, 最多為 30 列, 每列有一個自然數 n (表示一個 n 連續序列 S_n)。請依序輸出 t16.txt 檔案中的每個 n 連續序列 S_n 的數字中心 i 值, 若找不到數字中心則輸出 null。(請下載 t16.txt 檔案)

t16.txt: 157
49
288
2471

輸出: 6
null
35
204
null