

弘光科技大學資訊工程系程式能力檢定輔導教材

程式能力檢定題目解說與程式觀摩精選

張貿翔 編寫

中華民國一百零二年七月三十一日

## 程式設計檢定基本題庫內容

- Q1 以迴圈計算  $n!$  的值
- Q2 計算級數和： $(1+2)+(2+4)+(3+6)+\cdots+(n+2*n)$
- Q3 輸出一正整數的所有因數 (參閱基礎題庫 Q39 完美數)
- Q4 Armstrong 數
- Q5 迴文 (palindrome) (參閱進階題庫 Q23)
- Q6 二進位表示法
- Q7 傳址呼叫或傳參考呼叫
- Q10 計算級數和  $(1+2+\cdots+n)$  以及  $n$  的階乘
- Q13 八進位數與十六進位數到十進位數的轉換
- Q21 求  $1+2+\cdots+n$  的值的遞迴函數
- Q25 求陣列的最大值以及最小值的函數
- Q27 計算級數和  $(1+2+\cdots+n)$  以及  $n$  的階乘
- Q28 計算級數和： $x/1+x^2/2!+\cdots+x^n/n!$  (參閱進階題庫 Q15, 基礎題庫 Q47)
- Q31 求兩數的最大公因數
- Q32 排序
- Q33 求一元二次方程式的根
- Q34 複利計算
- Q35 求所有小於  $n$  的質數並列出它們 (參閱進階題庫 Q37)
- Q36 求兩數的最大公因數
- Q38 排序
- Q39 完美數 (Perfect Number)
- Q40 迴圈計算級數和： $1^2+2^2+3^2+4^2+\cdots+47^2+48^2+49^2+50^2$  的值
- Q41 求級數和
- Q42 計算費伯納西數列的函數
- Q43 一星期七天的英文單字查詢
- Q44 計算費伯納西數列的函數
- Q45 計算  $x$  的  $n$  次方的遞迴函數
- Q46 求級數和
- Q47 求級數和 (參閱進階題庫 Q15)
- Q49 英文字母大小寫轉換
- Q50 字串反轉
- Q51 指標運用與英文字母大小寫轉換
- Q52 動態記憶體配置
- Q53 統計英文字母
- Q54 ?A?B 遊戲

## 題目類型

### 1. 條件判斷

三角形判斷 Q29

?A?B 遊戲 Q54

### 2. 簡單計算

加班費計算 Q8

運費計算 Q19

打數計算 Q14

折扣計算 Q18

時針與分針的夾角大小計算 Q30

複利計算 Q34

### 3. 迴圈

階乘 Q1

級數和 Q2、Q28、Q41、Q46, Q47

費伯納西數列 Q42、Q44

### 4. 數的表示法

十進位數表示法 Q4、Q5

二進位數表示法 Q6

進階題 aQ 23

### 5. 因數與質數問題

輸出小於等於  $n$  的  $n$  的因數 Q3

最大公因數 Q31、Q36

輸出小於等於  $n$  的質數 Q35

進階題 aQ21, aQ37

### 6. 文字處理

Q49、Q50、Q53

### 7. 查詢

Q43, aQ26

# 算術運算

算術運算是程式設計的一個很重要的基石，先複習些許國中數學：

## 因數

若乙數可以整除甲數(甲數能被乙數整除)時，乙數就是甲數的因數。

例：

28 的因數有：1，2，4，7，14，28。

## 倍數

一個數的倍數有無限個。例：

4 的倍數有：4，8，12，16，20，.....。

## 因數與倍數之判斷方法

假設 A 與 B 為正整數。

若  $A \% B == 0$ ，則 B 整除 A (A 被 B 整除)。換言之，B 為 A 的因數；A 為 B 的倍數。

## 公因數

一個可以同時整除一個整數集合裡的所有整數的整數稱為該整數集的**公因數**。例：  
整數集合 {12, 20} 的公因數有 2 和 4。

## 最大公因數

一個整數集合(大於 0)的**最大公因數**，為該集合的公因數中最大的那一個數。例：  
{24, 36} 的公因數有：1，2，3，4，6，8，12。其中最大的為 12。因此 {24, 36} 的**最大公因數**為 12。求最大公因數可用短除法，求兩數的最大公因數可利用輾轉相除法。例如下面範例。

2 | 84, 36, 420

-----  
2 | 42, 18, 210

-----  
3 | 21, 9, 105

-----  
7, 3, 35

{84, 36, 420} 的最大公因數 =  $2 \times 2 \times 3 = 12$

## 公倍數

一個整數若同時為一個整數集合 S 的所有整數的倍數則稱其為該整數集的**公倍數**。

例：

{6, 8} 的公倍數有：24，48，.....。

## 最小公倍數

一個整數集合  $S$  的公倍數中最小者為  $S$  的最小公倍數。例：

$\{6, 9\}$  的公倍數有：18, 36, 54, .....。

$\{6, 9\}$  的最小公倍數為 18。

求最小公倍數可用短除法。例：

$$\begin{array}{r|l} 2 & 75, 120, 90 \\ \hline 3 & 75, 60, 45 \\ \hline 5 & 25, 20, 15 \\ \hline & 5, 4, 3 \end{array}$$

$\{75, 120, 90\}$  的最小公倍數  $= 2 \times 3 \times 5 \times 5 \times 4 \times 3 = 1800$

### 質數

除了 1 和本身以外，沒有其他因數的大於 1 的整數稱為質數。例：

19 的因數只有 1 和 19 本身，因此 19 為質數。

1 不是質數也不是合數。

### 合成數

除了 1 和本身以外，還有其他因數的整數稱為合成數。例：

8 的因數除了 1 和 8 本身之外還有因數 2 和 4，因此 8 為合成數。

### 互質

若一個整數集合  $S$  除了 1 以外沒有其他的公因數，稱  $S$  互質。例：

$\{4, 17, 25\}$  除了 1 以外沒有其他的公因數，因此互質。

如何判斷  $x$  是  $n$  的正因數？若  $x$  為正數且  $n$  除以  $x$  的餘數等於 0，則  $x$  是  $n$  的因數。換言之， $n \% x == 0$ ，則  $x$  是  $n$  的因數。假如  $x$  是  $n$  的正因數，則  $0 < x \leq n$ 。例如 17 是 51 的正因數，因為  $17 > 0$  且 17 整除 51。所以  $51 \% 17 == 0$ 。假設  $n$  除以  $x$  的商為  $y$ ，則  $n = x * y$ 。因此， $y$  也可以整除  $n$ 。所以  $x$  與  $y$  同時為  $n$  的因數。假設  $x \leq y$ ，則  $x \leq \sqrt{n}$ 。故  $n$  的正因數個數不大於  $2\lfloor\sqrt{n}\rfloor$ 。下列函式使用暴力法逐一檢驗從 1 到  $n$  (含)之間的整數是否為  $n$  的因數來計算非負整數  $n$  之正因數個數：

```
int factor_num(int n)    /* 假設  $n > 0$  */
{
    int i, r, num = 0;
    for (i = 1; i <= n; i++)    /* 判斷從 1 到  $n$  的每一個數是否為  $n$  的因數 */
    {
        if ((n % i) == 0) num = num + 1; /* 每發現  $n$  的因數，將 num 加 1 */
    }
    return num;
}
```

下列函式根據  $n$  的因數之性質，逐一檢驗從 1 到  $\sqrt{n}$  (含)之間的整數是否為  $n$  的因數來計算非負整數  $n$  正因數的個數：

```
int factor_num(int n)    /* 假設  $n > 0$  */
{
    int i, r, num = 0;
    r = floor(sqrt((float) n)); /* 求  $n$  的平方根之整數部分 */
    for (i = 1; i <= r; i++)    /* 從 1 到  $r$  判斷是否為  $n$  的因數 */
    {
        if ((n % i) == 0) num = num + 2; /*  $i$  與  $n/i$  皆為  $n$  的因數 */
    }
    /* 假如  $n$  為完全平方數，則  $r * r == n$ 
       這時計算  $n$  的因數個數  $r$  只能算 1 次，
       前面 for 迴圈之內計算兩次，故現在必須扣掉 1 */
    if (r * r == n) num = num - 1;
    return num;
}
```

修改上列函式以判斷整數  $n$  是否為完美數：

```
/* 假如回傳值為正數(大於 0)，表示  $n$  為完美數且回傳值為  $n$  的所有正因數之和  
   假如回傳值為負數(小於 0)，表示  $n$  不是完美數且回傳值之絕對值為  $n$  的所有正因數之和  
   */
```

```
int perfect(int n)
{
    int i, r, sum = 0;
    r = floor(sqrt((float) n)); /* 求  $n$  的平方根之整數部分 */
    for (i = 1; i <= r; i++) /* 從 1 到  $r$  判斷是否為  $n$  的因數 */
        if ((n % i) == 0) /* 如果  $i$  整除  $n$ ，則  $i$  是  $n$  的因數 */
            sum = sum + i + (n / i); /*  $i$  與  $n/i$  皆為  $n$  之因數 */
    /* 假如  $n$  為完全平方數，則  $r * r == n$ ，  
       這時計算  $n$  的因數之和只能加 1 次 */
    if (r * r == n) sum = sum - r;
    if ((sum - n) == n) return sum; /* 真因數之和不包括  $n$  */
    else return - sum;
}
```

下面改寫上列 perfect 函式，避免用 floor 與 sqrt 函式。

```
int perfect_No_Math (int n)
{
    int i = 1, r, sum = 0;
    r = n;
    do
    {
        if ((n % i) == 0) /* 如果  $i$  整除  $n$ ，則  $i$  是  $n$  的因數 */
            sum = sum + i + r; /*  $i$  與  $n/i$  皆為  $n$  之因數 */
        i++;
        r = n / i;
    } while (i < r);
    if (i * i == n) sum = sum + i; /* do 迴圈並未計算當  $n$  是完全平方數之情況 */
    if ((sum - n) == n) return sum; /* 真因數之和不包括  $n$  */
    else return - sum;
}
```

Q1. 試利用迴圈撰寫出一程式可計算出  $n!$  的值。(若判斷輸入的  $n$  為負值，則須重新讓使用者輸入至正值再進行計算)

例如輸入：

8

輸出結果

$8!=40320$

例如輸入：

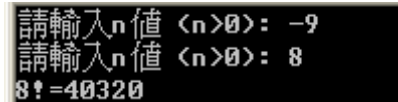
6

輸出結果

$6!=720$

A1、

```
#include <stdio.h>
#include <stdlib.h>
```



```
請輸入n值 <n>0>: -9
請輸入n值 <n>0>: 8
8!=40320
```

```
int main(void)
{
    int n, i, factorial; /* 宣告變數 */
    do
    {
        printf("請輸入 n 值 (n>0): "); /* 提示使用者輸入 n 值 */
        scanf("%d", &n); /* 輸入 n 值 */
    } while (n<=0); /* 若 n <= 0 時重新輸入 n 的值 */
    factorial = 1; /* 設定 factorial 初值為 1 */
    for (i = 1; i <= n; i++) /* 從 i == 1 到 i == n, */
        factorial = factorial * i; /* 逐一累乘 factorial */
    printf("%d!=%d\n", n, factorial);
    return 0;
}
```

Q2. 試撰寫一程式可求出多項式  $(1+2)+(2+4)+(3+6)+\dots+(n+2*n)$  的和。

例如輸入：

8

輸出結果

$(1+2)+(2+4)+(3+6)+(4+8)+(5+10)+(6+12)+(7+14)+(8+16) = 108$

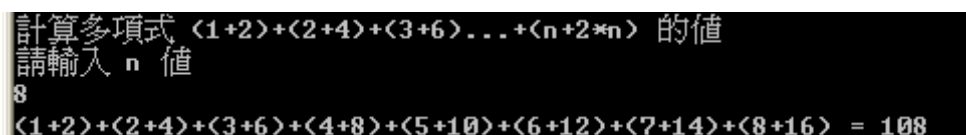
例如輸入：

7

輸出結果

$(1+2)+(2+4)+(3+6)+(4+8)+(5+10)+(6+12)+(7+14) = 84$

A2、



```
計算多項式 <1+2>+<2+4>+<3+6>...+<n+2*n> 的值
請輸入 n 值
8
<1+2>+<2+4>+<3+6>+<4+8>+<5+10>+<6+12>+<7+14>+<8+16> = 108
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(void)
```



```

{
    /* i 為迴圈變數, n 是級數的項數, sum 是用來求級數的和 */
    int i, n, sum;
    printf("計算多項式 (1+2)+(2+4)+(3+6)...+(n+2*n) 的值\n");
    printf("請輸入 n 值\n"); /* 提示使用者輸入 n 值 */
    scanf("%d", &n);          /* 輸入 n 值 */
    sum = 0; /* 設 sum 之初值為 0 */
    for(i = 1; i < n; i++) /* 從 i == 1 到 i == n - 1, */
    {
        sum = sum + (i + 2 * i); /* 累加第 i 項求總和 */
        printf("(%d+%d)+", i, 2 * i); /* 輸出級數第 i 項 */
    }
    sum = sum + (n + 2 * n); /* 累加第 n 項求總和 */
    printf("(%d+%d)=%d\n", n, 2 * n, sum);
    return 0;
}

```

Q3. 試撰寫一程式可由鍵盤輸入一個正整數  $n$ ，然後由小到大輸出其所有的因數。

例如輸入：

68

輸出結果

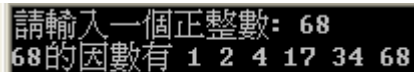
1 2 4 17 34 68

A3、

```

#include <stdio.h>
#include <stdlib.h>

```



```

請輸入一個正整數: 68
68的因數有 1 2 4 17 34 68

```

```

int main(void)
{
    int i, n;
    printf("請輸入一個正整數: "); /* 提示使用者輸入一個正整數 */
    scanf("%d", &n); /* 輸入一個正整數 */
    printf("%d 的因數有 ", n); /* 提示以下將輸出輸入的正整數之因數 */
    for(i = 1; i <= n; i++) /* i 從 1 到 n 逐一檢查是否為 n 之因數 */
        if(n % i == 0) /* 假若 n % i == 0, 則 i 為 n 的因數, */
            printf("%d ", i); /* 故輸出 i */
    return 0;
}

```

Q4.所謂 "Armstrong 數" 是指一個三位數的整數，其各位數字之立方和恰等於該數的本身。

例如：153 是一個 Armstrong 數，因為  $1^3 + 5^3 + 3^3$ 。試撰寫一程式，找出所有的 Armstrong 數。

輸出結果

Armstrong 數有 153 370 371 407

A4、

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i, num, sum = 0;
    int x,y,z;
    printf("Armstrong 數有 ");
    /* 根據定義，阿姆斯壯數為十進位的三位數，介於100與999之間 */
    for(i = 100; i <= 999; i++) /* i值從100到999之間逐一檢驗是否為阿姆斯壯數 */
    {
        x = i/100;    /* 取i之百位數 */
        y = (i/10)%10; /* 取i之十位數 */
        z = i % 10;    /* 取i之個位數 */
        /* 若 i == x * x * x + y * y * y + z * z * z，則i為阿姆斯壯數 */
        if(i == x * x * x + y * y * y + z * z * z) printf("%d ", i);
    }
    return 0;
}

/* 輸出範例：
Armstrong 數有 153 370 371 407
*/
```

Q5.請設計一個程式,能判斷一數是否為迴文數。所謂迴文數(palindrome)就是一個數從左到右與從右到左的讀法相同。例如:12321 就是迴文數，1234 則不是。

例如輸入：

1234

輸出結果

1234 is not a palindrome.

例如輸入：

25752

輸出結果

25752 is a palindrome.

進階題庫 Q23 定義一個正整數的鏡反數(mirror-image number)為原來數字的十進位表示法由右至左順序重新表示的新數字，以函數  $m(n)$  來表示；例如十進位數 123 的鏡反數  $m(123)$  等於十進位數 321。根據題亦可知，一數  $n$  若等於其鏡反數即為迴文數

下列函式回傳  $n$  的鏡反數：

```
int m(int n)
{
    int k;
    k = n % 10;    /* n 除以 10 取餘數得 n 的十進位表示法的個位數 */
    n = n / 10;    /* n 除以 10 取商 */
    while (n != 0)
    {
        /* 只要 n 不等於 0 就取其個位數加上 k * 10 */
        k = k * 10 + (n % 10);
        n = n / 10;
    }
    return k;
}
```

A5、

```
#include <stdio.h>
#include <stdlib.h>

int m(int n);

int main()
{
    int n;
    printf("Please enter a positive integer:");
    scanf("%d", &n);
    while(n > 0)    /* 輸入非正整數，程式終止執行 */
    {
        if (n == m(n))    /* 假如 n 等於其鏡反數 m(n)，則 n 為迴文數 */
            printf("%d is a palindrome.\n", n);
        else
            printf("%d is not a palindrome.\n", n);
        printf("Please enter a positive integer:");
        scanf("%d", &n);
    }
    return 0;
}
```

Q6.將一十進位的數轉換成二進位，秀出轉換後第幾個 bit 為 1，有幾個 bits 為 1。

例如輸入：

12

輸出結果

12

Bit '1' at {2,3}

Bit '1' has 2

例如輸入：

61

輸出結果

61

Bit '1' at {0,2,3,4,5}

Bit '1' has 5

A6、

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int n, num, i; // num 用來計數有幾個位元為 '1'
```

```
    printf("Please enter an integer, or enter a non-positive number to exit: "); // 提示輸入
```

```
    while(scanf("%d", &n) && (n > 0)) // 讀入 n 且當 n > 0 時執行迴圈本體
```

```
    {
```

```
        printf("bit '1' at {"); // 列印輸出第一列的前導文字
```

```
        i = 0; // 初始化 i，代表位元的位置編號
```

```
        num = 0; // 初始化 num，num 用來計數有幾個位元為 '1'
```

```
        while(n > 0) // n 等於 0 時，剩餘位元全部為 0
```

```
        {
```

```
            if(n % 2 == 1) // 將 n 除 2 取餘數得其 LSB，判斷其值是否為 '1'
```

```
            {
```

```
                printf("%d,", i); // 列印出其位置編號
```

```
                num++; // 為 1 的位元數目加 1
```

```
            }
```

```
            n = n / 2; // 去掉已經檢查過的位元
```

```
            i++; // 準備下一次要檢查的位元之位置編號
```

```
        }
```

```
        printf("\b}\n"); // 列印輸出第一列的結束文字
```

```
        printf("bit '1' has %d\n", num); // 列印輸出第二列，位元為 1 的數目
```

```
        printf("Please enter an integer, or enter a non-positive number to exit: "); // 提示輸入
```

```
    }
```

```
    return 0;
```

```
}
```

Q8.假設某便利商店的工讀生的月薪資，可以依照下列方式計算：

60 個小時之內，每小時 75 元

61 ~ 75 個小時，以 1.25 倍計算

76 個小時以後以 1.75 倍計算

例如，如果工作時數為 80 小時，則薪資為  $60*75+15*75*1.25+5*75*1.75=6562.5$  元。請撰寫一個程式，首先在螢幕上輸入某工讀生該月的工作時數(為一整數)，然後計算並列印出實領薪資(取至小數點後第一位)。

結果如下：

1.工作時數=80

實領薪資=6562.5 元

2.工作時數=70

實領薪資=5437.5 元

3.工作時數=50

實領薪資=3750.0 元

A8、

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int hours;
    float salary;
    printf("工作時數： "); // 輸入提示
    scanf("%d", &hours);

    if (hours <= 60)
        salary = hours * 75; // 時數不高於 60，只算基本薪
    else if (hours <= 75)
        // 時數高於 60 但不高於 75，超過 60 的時數以 1.25 倍計
        salary = 60 * 75 + (hours - 60) * 75 * 1.25;
    else // 時數高於 75，60 小時為基本薪，15 小時 1.25 倍計薪，其餘時數以 1.75 倍計
        salary = 60 * 75 + 15 * 75 * 1.25 + (hours - 75) * 75 * 1.75;

    printf("實領薪資 = %f 元\n", salary);
    return 0;
}
```

解決一個問題的程式可以有不同的寫法：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int hours;
    float salary;
    printf("工作時數： "); // 輸入提示
    scanf("%d", &hours);
```

```

salary = hours * 75;      // 先算基本薪
if (hours > 60) // 超過 60 小時部份，加上第二階段的加給
    salary = salary + (hours - 60) * 75 * (1.25 - 1.0);
if (hours > 75) // 超過 75 小時部份，再加上第三階段的加給
    salary = salary + (hours - 75) * 75 * (1.75 - 1.25);

printf("實領薪資 = %f 元\n", salary);
return 0;
}

```

Q9.請撰寫一個程式，利用 do while 迴圈完成九九乘法表。在螢幕上列印結果如下：

```

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

```

A9、

```

#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i, j;
    for (i = 1; i <= 9; i++) // 每一圈列印一列(第 i 列)，共九列
    {
        for (j = 1; j <= 9; j++) // 每一圈印一欄(第 j 欄)，共 9 欄
            printf("%d*%d=%2d ", j, i, i * j); // 列印第 i 列第 j 欄
        printf("\n"); // 列印第 i 列結束時跳下一列的 '\n'
    }
    return 0;
}

```

Q10.請撰寫一程式，包含二個函數：fac(n)=1\*2\*3\*…\*n 及 sum(n)=1+2+3+…+n，首先請 在螢幕上輸入一個整數 n，然後以呼叫函數的方式，計算並列印出 fac(n)=1\*2\*3\*…\*n=? 及 sum(n)=1+2+3+…+n=?

結果範例如下：

請輸入整數 n=7

fac(7)=1\*2\*3\*…\*7=5040

sum(7)=1+2+3+…+7=28

A10、

```

#include <stdio.h>
#include <stdlib.h>

```

```

int sum(int)          /* 定義函數的原型 */
int fac(int);
int main(void)
{
    int n;
    printf("請輸入整數 n= ");
    scanf("%d",&n);
    printf("fac(%d) = 1*2*...*%d = %d\n", n, n, fac(n)); /* 印出 n!的結果 */
    printf("sum(%d) = 1+2+...+%d = %d\n", n, n, sum(n)); /* 印出加總的結果 */
    return 0;
}

int fac(int m)        /* 自訂函數 fac()，計算 m! */
{
    int i, total=1;
    for(i = 1; i <= m; i++) total = total * i;
    return total;
}

int sum(int m)        /* 自訂函數 sum()，計算 1+2+...+m 的結果*/
{
    int i,total=0;
    for(i=1;i<=m;i++)
        total+=i;
    return total;
}

```

Q14. 現在有 200 個蛋，一打是 12 個，請設計 C 程式計算 200 個蛋是幾打，還剩下幾個蛋。

**A14、**

```

#include <stdio.h>
int main( void ) {
    int a = 200;
    printf("幾打= %d\n", a/12);
    printf("剩幾個= %d\n", a%12);
    return 0;
}

```

```

幾打= 16
剩幾個= 8
請按任意鍵繼續 ...

```

Q15.請設計 C 程式輸入整數值，然後顯示數值的十進位、十六進位和八進位值。

**A15、**

```

#include <stdio.h>
int main(void) {
    int a;

```

```

printf("請輸入 a 的值: ");
scanf("%d", &a);
printf("十進位: %d\n", a);
printf("八進位: %o\n", a);
printf("十六進位: %x\n", a);
system("PAUSE");
return 0;
}

```

Q16. 請撰寫 C 程式執行從 1 到 100 的迴圈，但只顯示 45~67 之間的奇數，並且計算其總和。

**A16、**

```

#include <stdio.h>
int main(void) {
    int i, total = 0;
    for (i = 1; i <= 100; i++) {
        if ((i % 2) != 0 && i >= 45 && i <= 67) {
            total += i;
            printf("%d\n", i);
        }
    }
    printf("總和: %d\n", total);
    return 0;
}

```

```

45
47
49
51
53
55
57
59
61
63
65
67
總和: 672

```

Q17. 請寫一個 C 程式依序顯示 1~20 的數值和其平方，每一數值成一行，如下所示：

```

1      1
2      4
3      9
.....

```

**A17、**

```

#include <stdio.h>

```



```

int main(void) {
    int i;
    for (i = 1; i <= 20; i++) {
        printf("%d\t%d\n", i, i*i);
    }
    return 0;
}

```

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100
11	121
12	144
13	169
14	196
15	225
16	256
17	289
18	324
19	361
20	400

Q18.百貨公司正在周年慶，消費者消費滿千元(含)以上，打八折優待，未滿千元沒有折扣。請建立 C 程式讓消費者輸入消費金額，輸出折扣後應付的金額。消費金額以元為單位，折扣後應付的金額計算至小數點後一位。

**A18、**

```

#include <stdio.h>
int main(void) {
    float amount;
    printf("輸入金額=> ");
    scanf("%f", &amount);
    if ( amount >= 1000 ) amount = amount * 0.8;
    printf("付款金額: %.1f\n", amount);
    return 0;
}

```

輸入金額=>2500 付款金額: 2000.0
----------------------------

Q19.請設計 C 程式計算網路購物的運費，基本物流處理費 199，1~5 公斤，每公斤 50 元，超過 5 公斤，每一公斤為 30 元，在輸入購物重量為 3.5、10、25 公斤，請計算和顯示購物所需的運費+物流處理費？

## A19、

```
#include <stdio.h>
int main(void) {
    float weight, fee;
    printf("輸入重量=>");
    scanf("%f", &weight);
    if ( weight <= 5 )
        fee = weight * 50 + 199;
    else
        fee = 5*50+(weight-5)*30+199;
    printf("運費: %f\n", fee);
    return 0;
}
```

```
輸入重量=>10
運費: 599.000000
請按任意鍵繼續 ...
```

Q21.請寫遞迴函數 sum(int)，計算 1 到參數值的和，例如：sum(5)，就計算 5+4+3+2+1。

## A21、

```
#include <stdio.h>
int sum(int n) {
    if (n == 1) return n;
    else
        return n + sum(n-1);
}
int main(void) {
    printf("sum(5) = %d\n", sum(5));
    return 0;
}
```

```
sum(5) = 15
```

Q24.請設計 C 程式建立 2 個各 5 個元素的一維陣列 arr1 和 arr2，其初值如下所示，然後建立 5 個元素的一維陣列 result，使用迴圈計算 arr1 和 arr2 相同索引元素的和，將它存入陣列 result，最後顯示出的陣列內容，其格式如下所示：

索引	arr1	arr2	result
0	2	3	5
1	34	56	90
2	33	10	43
3	23	20	43
4	67	73	140

## A24、

```
#include <stdio.h>
int main(void) {
    int arr1[5] = {2, 34, 33, 23, 67};
    int arr2[5] = {3, 56, 10, 20, 73};
    int i, result[5];
    for ( i = 0; i < 5; i++ )
        result[i] = arr1[i] + arr2[i];
    printf("索引\tarr1\tarr2\tresult\n");
    for ( i = 0; i < 5; i++ )
        printf("%d\t%d +\t%d =\t%d\n",i,arr1[i],arr2[i],result[i]);
    return 0;
}
```

Q28.請撰寫一個程式，使用一個 for 迴圈來計算下面的數學式：

$$\text{my\_fun}(x,n)=\sum_{k=1}^n \frac{x^k}{k!} = \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

首先請在螢幕上輸入 x 及 n，並在螢幕上列印出 my\_fun(x,n)之計算結果。

結果範例如下：

1. 請輸入 x=0.1

請輸入 n=5

my\_fun(0.100000,5)=0.105171

2.請輸入 x=0.2

請輸入 n=5

my\_fun(0.200000, 5)=0.221403

**這題與進階題 Q15 高度相似。**

## A28、

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    double x, PF = 1.0, Sum = 0.0;
    int n, k;
    printf("請輸入 x=");
    scanf("%lf", &x);
    printf("請輸入 n= ");
    scanf("%d", &n);
    for(k = 1; k <= n; k++) /* 以 for 迴圈逐項累加求級數和 */
    {
        PF = PF * x / k; /* PF = x^k / k! */
        Sum = Sum + PF;
    }
    printf("my_fun(%lf,%d)=%lf\n",x,n,sum);
    return 0;
}
```

Q29.請設計一 C 程式，由使用者任意輸入三角形的三邊長（邊長為整數，且輸入值並未經過大小排列）後，判斷此三邊所組成之三角形是何種三角形（正三角形、等腰三角形、一般三角形或非三角形），並印出答案。（972 新增）

【輸入輸出格式】：

```
請輸入第一個整數:3
請輸入第二個整數:2
請輸入第三個整數:1
無法形成三角形
請按任意鍵繼續 . . .
```

```
請輸入第一個整數:2
請輸入第二個整數:2
請輸入第三個整數:2
正三角形
請按任意鍵繼續 . . .
```

【提示】：假設輸入的三邊長分別為： $a$ ， $b$ ， $c$ ；且其中  $a \leq b \leq c$

若  $a=b$  或  $b=c$  則為等腰三角形(isosceles triangle)

$a=b=c$  則為正三角形(regular triangle)

$c \geq a + b$  則為非三角形(non triangle)

【註】：任二邊長之總和皆大於第三邊則可形成三角形；若為正三角形，請勿再列印出等腰三角形。

A29、

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a,b,c;
    printf("請輸入三角形三邊長: ");
    scanf("%d %d %d",&a,&b,&c);
    if (a <= 0 || b <= 0 || c <= 0 || c >= a + b || b >= a + c || a >= b + c)
        printf("無法形成三角形");
    else if (a==b && b==c)
        printf("正三角形");
    else if (a == b || b==c || c == a)
        printf("等腰三角形");
    else
        printf("一般三角形");
    return 0;
}
```

Q30.請撰寫一個程式可由鍵盤輸入時間(小時、分鐘)，然後在螢幕上顯示出對應的時鐘上分針與時針之間的夾角為幾度。（972 考題）

例如：

請輸入現在時間 (小時:分鐘)

3:15

3 點 15 分時，時針與分針之夾角為 7.5 度

### A30、

#### 參考程式碼

```
-----
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int h, m;
    float degree_h, degree_m, angle;

    printf("請輸入現在時間 (小時:分鐘)\n");
    scanf("%d:%d", &h, &m);

    degree_h = 30.0 * (h + m/60.0);
    degree_m = 6.0 * m;

    angle = degree_m - degree_h;
    if(angle < 0)    //角度為正值
        angle = -angle;
    if(angle > 180)    //角度應在 0-180 度間
        angle = 360-angle;

    printf(" %d 點 %d 分時，時針與分針之夾角為 %5.1f 度\n", h, m, angle);
    system("pause");
    return 0;
}
```

Q34.請撰寫一個程式，可計算定期定額投資收益結果，利率採複利計算。程式可輸入年存額為 X 元，年複利率為 R%，第 N 年後領回。(請將結果四捨五入至整數位)

例如：每年存 100000，存 3 年，若是複利率為 2.25%，3 年後的本利和如下：

第一年初存入 100000，本金=100000

第二年初，前一年本利=100000\*(1+2.25%)=102250，加上存入的 100000，合計 202250

第三年初，前一年本利=202250\*(1+2.25%)=206801，加上存入的 100000，合計 306801

第三年尾合計領回的本利和=306801\*(1+2.25%)=313704

例如輸入 (X R N)：

**10000 12 10**

輸出結果

**196546**

例如輸入 (X R N)：

**10000 15 20**

輸出結果

**1178101**

### A34、

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>

int main(void)
{
    int    N,year=1;
    double X,R,amount;
    scanf("%lf %lf %d",&X,&R,&N);
    amount=0;
    do{
        amount=amount+X;
        amount=amount*(1+R/100);
        year++;
    }while(year<=N);
    printf("%d",(int)(amount+0.5));
    return 0;
}

```

Q35.大於 1 的整數，除了 1 與本身以外沒有其他因數，稱之為『質數』。換言之，質數僅被 1 及本身整除。因此，2,3,5,7,11....等均稱為質數，而 33 可被 1,3,11,33 整除，非質數。請撰寫一個程式，可輸入一整數 N，計算出所有小於等於 N 的所有質數，並以每 10 個一列輸出，最後亦輸出質數總個數。

例如輸入：

**100**

輸出結果

**2 3 5 7 11 13 17 19 23 29**

**31 37 41 43 47 53 59 61 67 71**

**73 79 83 89 97**

**There are 25 primes**

A35、

範例程式一：

下列程式，利用 C 語言進階語法 switch 關鍵字。其實稍微用心思考，可以用比較簡單的語法撰寫程式，一樣可以達成相同的功能。

```

#include <stdio.h>
#include <stdlib.h>

```

```

void main(void)
{
    int n, count = 0;
    int i, j;
    bool prime;

    scanf("%d",&n);

    for(i = 1; i <= n; i++){
        prime = true;

```

```

switch(i){
    case 1 :      // 1 非質數
        prime = false;
        break;
    case 2 :      // 2 是最小的質數
        break;
    default :
        for(j = 2; j < i; j++){
            if((i % j) == 0){// i 可被 j 整除，非質數
                prime = false;
                break;
            }
            else;
        }
    }
    if(prime == true){    // 印出質數
        count ++;
        printf("%d ", i);

        if(count % 10 == 0){    // 每印滿 10 個質數就跳下一行
            printf("\n");
        }
        else;
    }
    else;
}

printf("\nThere are %d primes", count);
}

```

**A35、**

**範例程式二：**

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(void)
```

```

{
    int n, counter;
    int i, j;
    scanf("%d",&n);
    if(n < 2)
    {
        printf("沒有小於 2 的質數");
        return 0;
    }
    printf("2");    /* 輸出第一個質數 2 */
    counter = 1;    /* 這個變數用來數輸出的質數個數 */
    if(n == 2)
    {
        printf("\nThere are %d primes", counter);
        return 0;
    }
}

```

```

}
for(i = 3; i <= n; i++) /* i 從 3 到 n 逐一檢查其是否為質數 */
{
    j = 2;
    while(j < i && i % j != 0) j++; /* j 不整除 i，繼續試下一個 j 直至 j == i */
    if(j == i) /* 如果 j == i，則 i 為質數 */
    {
        counter++; /* 又找到一個質數，質數個數加 1 */
        if (counter % 10 == 1)
            printf("%d", i); /* 每十個質數的第一個 */
        else
            printf(" %d", i);
        if (counter % 10 == 0) /* 每十個質數的第十個 */
            printf("\n");
    }
}

printf("\nThere are %d primes", counter);
return 0;
}

```

Q36. 試撰寫一程式可由鍵盤輸入兩個數字，並求得此兩數的最大公因數。

例如輸入：

**54 48**

輸出結果


**54 與 48 的最大公因數為 6**

A36、

```

#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int num1,num2; /* 要求最大公因數的兩個數 */
    int a,b,c; /* 依次為除數、被除數、餘數 */
    printf("請輸入兩個數字\n");
    scanf("%d %d",&num1,&num2);
    c = num1%num2; /* 先計算一次餘數 */
    b = num2; /* 以其中一個數為被除數 */
    while(c!=0) /* 計算輾轉相除的迴圈 */
    {
        a=b; /* 指定前一次除數為新的被除數 */
        b=c; /* 指定前一次餘數為新的除數 */
        c=a%b; /* 求出新的餘數 */
    }
    printf("%d 與 %d 的最大公因數是 %d\n",num1,num2,b);
    return 0;
}

```



```

請輸入兩個數字
54 48
54 與 48 的最大公因數是 6

```



Q39.如果有一數  $n$ ，其真因數（Proper factor）的總和等於  $n$ ，則稱之為完美數（Perfect Number），例如以下幾個數都是完美數：

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

$$496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$$

現在請寫一個程式，在輸入一個數字  $n$  後，能印出正因數的和並回答其是否為完美數。

例如輸入：

496

輸出結果

The sum of these divisors is 992

496 is a perfect number!

例如輸入：

49

輸出結果

The sum of these divisors is 57

49 is not a perfect number!

### A39、

這個範例程式使用暴力法，可參考前面解說，撰寫更有效率的程式。

```
#include <stdio.h>
```

```
main(){
    int n,x,sum;
    printf("Input an integer:");
    scanf("%d",&n);

    sum=0; /* sum 用來累計 n 的因數的和 */
    x=1;
    while(n >= x) /* 從 x 等於 1 到 n 逐一判斷 x 是否為 n 的因數 */
    {
        if(n % x == 0) /* 判斷 x 是否為 n 的因數 */
            sum += x;
        x++;
    }
    printf("The sum of these divisors is %d\n", sum);
    if(n * 2 == sum) /* 若所有因數的和等於 2 倍 n，則 n 為完美數 */
        printf("%d is a perfect number!\n", n);
    else
        printf("%d is not a perfect number!\n", n);
    return 0;
}
```

Q41. 撰寫一函數double my\_fun (int n)，可用來計算下面的數學式，並可由使用者任意輸入n的值後，列印出my\_fun(n) 的計算結果。

$$\text{my\_fun}(n) = \sum_{k=1}^n \frac{1}{k!} = \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

執行畫面說明範例

**範例一：**

請輸入 n 值：

10

my\_fun(10)=1.718282

**範例二：**

請輸入 n 值：

8

my\_fun(10)=1.718279

A41、

```
#include <stdio.h>
#include <stdlib.h>

double my_fun(int n);

int main(void)
{
    int n;
    printf("請輸入 n = ");
    scanf("%d", &n);
    printf("my_fun(%d) = %10.9f\n", n, my_fun(n));
    return 0;
}

double my_fun(int n)
{
    double k_term = 1.0, sum = 0.0;
    int k;
    for(k = 1; k <= n; k++) /* 從第 1 項開始逐項累計級數的和 */
    {
        k_term = k_term/k; /* 累計級數的第 k 項 */
        sum = sum + k_term; /* 累計級數從第 1 項到第 k 項的和 */
    }
    return sum;
}
```

Q43.試撰寫一程式可提供使用者查詢一星期中每一日的英文單字，若輸入的內容非星期一至星期日的範圍，則須顯示”查無此日 out of range”的訊息。

程式執行範例：

**查詢星期一~星期日的英文單字**

**請輸入中文名稱**

**星期五**

**星期五 的英文單字為 Friday**

(寫這個程式，應該先背一週七天的英文單字)

```

#include <stdio.h>
#include <string.h>

int main(void)
{
    /* 儲存一周七天的中文名稱的字串陣列 */
    char C_weekdays[7][7]= {"星期一",
                              "星期二",
                              "星期三",
                              "星期四",
                              "星期五",
                              "星期六",
                              "星期日"
                              };
    /* 儲存一周七天的英文文名稱的字串陣列*/
    char E_weekdays[7][10]= {"Monday",
                              "Tuesday",
                              "Wednesday",
                              "Thursday",
                              "Friday",
                              "Saturday",
                              "Sunday"
                              };

    char line[60];    /* 儲存由鍵盤輸入的搜尋字串 */
    int k;

    printf("查詢星期一~星期日的英文單字\n");
    printf("請輸入中文名稱\n");
    scanf("%s", line);
    k = 0; /* 從 k = 0 開始到 k < 7 逐一比較輸入，直到 k = 7 或匹配成功 */
    while(k < 7 && strcmp(line, C_weekdays[k]) != 0) k = k + 1;
    if (k == 7)
        printf("查無此日 out of range \n");
    else
        printf("%s 的英文名稱為 %s\n", C_weekdays[k], E_weekdays[k]);
    return 0;
}

```

Q44.費式數列 (Fibonacci) 是第一個和第二個數字為 1， $F_0=F_1=1$ ，其它是前兩個數字的和  $F_n = F_{n-1} + F_{n-2}$ ,  $n \geq 2$ ，請設計 fibonacci()函數顯示費式數列，參數是顯示數字的個數。

程式執行時輸出範例：

$F_0 = 1, F_1 = 1, F_2 = 2, F_3 = 3, F_4 = 5, F_5 = 8$

A44、

```
#include <stdio.h>
#include <stdlib.h>
int fibonacci(int n)
{
    int t1 = 1, t2 = 1, t, k;
    if (n < 0) return 0;
    printf("F0 = 1");
    if (n == 0) return 1;
    printf(", F1 = 1");
    if (n == 1) return 1;
    for(k = 2; k <= n; k++)
    {
        t = t1 + t2;
        t1 = t2;
        t2 = t;
        printf(", F%d = %d", k, t);
    }
    return t;
}

int main(void)
{
    fibonacci(5);
    printf("\n");
    return 0;
}
```

Q45. 請設計遞迴函數計算  $X^n$  的值，例如： $5^7$ 、 $8^5$  等。

執行程式輸出範例：

$5^7 = 78125.000000$   
 $8^5 = 32768.000000$

A45、

```
#include <stdio.h>
float Xpower(int x, int n)
{
    if (n == 0) return 1.0;
    else return x * Xpower(x, n-1);
}

int main(void)
{
    printf("5^7 = %f\n", Xpower(5, 7));
    printf("8^5 = %f\n", Xpower(8, 5));
    return 0;
}
```

Q46.請設計一個程式由使用者輸入一個正整數 n，計算下列級數計算至第 n 項的值，並將此計算值輸出至螢幕。(961 考題)

$$S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots$$

A46、

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i, n;
    double sum;
    printf("please enter a number: ");
    scanf("%d",&n);
    sum = 1.0;
    printf("S = 1 ");
    for(i = 2; i <= n; i++) /* 從第一項開始逐項累計級數的和 */
    {
        if(i % 2==1)
        {
            sum = sum + (1.0 / (double) i); /* i 若為奇數，加上 1/i */
            printf("+ 1/%d ", i); /* 輸出級數的第 i 項(奇數項) */
        }
        else
        {
            sum = sum - (1.0 / (double) i); /* i 若為偶數，減去 1/i */
            printf("- 1/%d ", i); /* 輸出級數的第 i 項(偶數項) */
        }
    }
    /* 以下輸出級數的和 */
    printf("\n", sum);
    return 0;
}
```

Q49.試寫一個程式，可將使用者輸入的字串中小寫的英文字轉成大寫的英文字，字串中大寫的英文字轉成小寫的英文字。(992 新增)

例如：

輸入資料

Good morning

輸出結果

gOOD MORNING

A49、

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    char str [256];
    char ASCHIDist = 'a' - 'A';
    int k = 0;
    printf("Input: ");
    gets(str);
    while (str[k] != '\0')
    {
        if((str[k] <= 'z' && str[k] >= 'a') || (str[k] <= 'Z' && str[k] >= 'A'))
            str[k] = str[k] ^ ASCHIDist;
        k++;
    }
    printf("Output: %s\n", str);
    return 0;
}
```

Q50.試寫一個程式，可將使用者輸入的字串做反轉。(992 新增)

例如：

輸入資料

Good morning

輸出結果

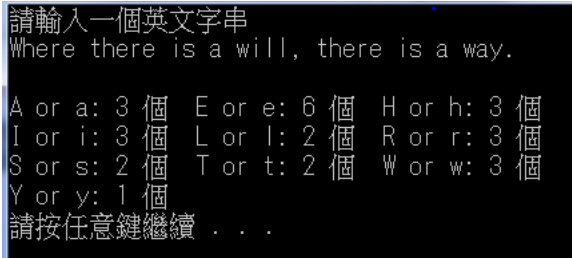
gninrom dooG

A50、

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void)
{
    char str [256];
    int length, k;
    printf("Input: ");
    gets(str);
    length = strlen(str);
    printf("Output: ");
    for(k = length - 1; k >= 0; k--)
        printf("%c", str[k]);
    printf("\n");
    return 0;
}
```

Q53.試撰寫一程式可讓使用者輸入一個字串，程式可計算此字串中所出現的各字母個數。利用字母的 ASCII 碼加以計數，並將統計的結果輸出於畫面當中(僅計算有出現的字母即可，未出現的字母不需顯示)。(100.2 新增)

以下為執行時的參考畫面：



```
請輸入一個英文字串
Where there is a will, there is a way.

A or a: 3 個 E or e: 6 個 H or h: 3 個
I or i: 3 個 L or l: 2 個 R or r: 3 個
S or s: 2 個 T or t: 2 個 W or w: 3 個
Y or y: 1 個
請按任意鍵繼續 . . .
```

A53、

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    char string[60];    /* 宣告儲存從鍵盤輸入字串的陣列 */
    int c, i, j;
    int number[26] = {0}; /* 宣告儲存 26 個字母數目的陣列，且初始值定為 0 */

    printf("請輸入一個英文字串： \n");
    gets(string);

    i = 0;
    while (string[i] != '\0')
    { /* 假設輸入的字母都是英文字母 */
        c = (string[i] | 32) - 97; /* 請注意位元運算子 '|' 以及 32 與 97 的意義 */
        number[c]++; /* 累計該字母出現之個數 */
        i++;
    }
    j = 0;
    for(i = 0; i < 26; i++) /* 輸出計算結果 */
    {
        If (number[i] != 0)
        {
            if (j % 3 == 0)
                printf("\n");
            printf("%c or %c: %d 個 ", i + 65, i + 97, number[i]);
            j++;
        }
    }
    printf("\n");
    return 0;
}
```

Q54.請撰寫一個”1A2B”的猜數字遊戲，程式預設”1234”為答案，讓使用者每次輸入四個不同的數字(一次僅可輸入一個數字，且須判斷相同的數字是否曾重覆輸入)。若數值相同且又在同樣的位置上，則表示可得一個 A。若數值相同但位置不同，則可得到一個 B。當使用者輸入完四個數字，系統可回覆該組數字可得?A?B，並持續讓使用者輸入至完全答對(4A0B)為止。(100.2 新增)

例如：

請輸入四個數字：

第 1 個數字：4

第 2 個數字：3

第 3 個數字：2

第 3 個數字：2

2 已經出現過了，請重新輸入：

第 4 個數字：1

您輸入的字串 4321：0 A 4 B

請輸入四個數字：

第 1 個數字：1

第 2 個數字：2

第 3 個數字：3

第 4 個數字：4

您輸入的字串 1234：4 A 0 B

以下為執行時的參考畫面：

```
請輸入四個數字：
第 0 個數字：4
第 1 個數字：3
第 2 個數字：2
第 3 個數字：2
2 已經出現過了，請重新輸入：
第 3 個數字：1
您輸入的字串 4321：0 A 4 B
請輸入四個數字：
第 0 個數字：1
第 1 個數字：2
第 2 個數字：3
第 3 個數字：4
您輸入的字串 1234：4 A 0 B 請按任意鍵繼續 . . .
```



#### A54、

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char target[10] = {0,1,1,1,1,0,0,0,0,0}; /* target[i] == 1 表示數字 i 在被猜的數字當中 */
    char target_pos[10] = {0, 1, 2, 3, 4, 0, 0, 0, 0, 0}; /* target[i] == k 表示數字 i 在第 k 個位置 */
    char input[10]; /* input[i] == 1 表示數字 i 在玩家輸入的數字當中 */
    char input_pos[10]; /* input[i] == k 表示數字 i 在玩家輸入的第 k 個位置 */
    char input_char[5]; /* 以文字形式儲存玩家輸入的數字 */
    int i, A = 0, B=0, ch;
    while(A != 4) /* A == 4 表示 */
    {
        A = 0, B=0;
        for(i=0; i < 10; i++) input[i] = 0; /* 清除輸入 */
        for(i=0; i < 10; i++) input_pos[i] = 0; /* 清除輸入 */
        printf("請輸入四個 1 ~ 9 之間的數字：\n\n");
        for (i=0; i < 4; i++)
        {
            printf ("請輸入第 %d 個數字：", i+1);
            scanf ("%d", &ch);
            while (ch < 1 || ch > 9 || input[(ch)] == 1)
            {
                if(ch > 0 && ch < 10 && input[(ch)] == 1)
                    printf("%d 已經輸入過了，請輸入尚未輸入過的數字!\n", ch);
                else printf("請輸入 1 ~ 9 之間之數字！\n");
                printf("請輸入第 %d 個數字：", i+1);
                scanf("%d", &ch);
            }
            input_char[i] = ch + 48; /* 以文字形式儲存玩家輸入的數字 */
            input[ch] = 1; /* 記錄玩家輸入的數字 */
            input_pos[ch] = i+1; /* 記錄玩家輸入的數字之位置 */
        }
        for (i = 1; i < 9; i++) /* 檢查輸入的數字與目標書字之差異 */
        {
            if (input[i] == 1 && target[i] == 1 && input_pos[i] == target_pos[i])
                A++; /* 數字與位置都猜對之個數 */
            else if(input[i] == 1 && target[i] == 1)
                B++; /* 數字猜對與位置沒猜對之個數 */
        }
    }
}
```

```
    }  
    printf("\n 您輸入的字串  %s : %d A %d B\n\n", input_char, A, B);  
}  
return 0;  
}
```

## 程式設計檢定進階題庫內容

- Q1 兌換零錢問題 (排列組合)
- Q2 多項式相乘
- Q7 統一發票兌獎 (檢索類題)
- Q9 兌換零錢問題 (排列組合)
- Q11 字串處理
- Q12 矩陣相加
- Q15 計算級數和
- Q16 最大公因數的疊代演算法
- Q17 二維陣列處理
- Q18 讀取文字檔計算列數
- Q19 文字檔處理
- Q20 亂數產生與陣列處理
- Q21 質因數分解 (醜數)
- Q22 亂數產生與二維陣列
- Q23 平方鏡反數
- Q26 日期寫法格式轉換
- Q28 大樂透兌獎
- Q29 讀取文字檔，統計總字元數
- Q30 讀取文字檔
- Q31 矩陣轉換為轉置矩陣
- Q37 列印 1 到  $n$ (含)之間得質數

Q1. 常用的中華民國硬幣面額有 50 元、10 元、5 元以及 1 元四種。輸入一個金額  $n$  元，印出將  $n$  元全部兌換成 50 元、10 元、5 元以及 1 元四種硬幣的所有可能的兌換方法以及可能的兌換方法數。

例如輸入：

18

輸出如下：

\$50	\$10	\$5	\$1
0	1	1	3
0	1	0	8
0	0	3	3
0	0	2	8
0	0	1	13
0	0	0	18

Total: 6

註 1：上述輸出表示金額 18 元，共有 6 種兌換方法，第 1 列表示可兌換 50 元硬幣 0 枚、10 元硬幣 1 枚、5 元硬幣 1 枚、1 元硬幣 3 枚。第 2 列表示可兌換 50 元硬幣 0 枚、10 元硬幣 1 枚、5 元硬幣 0 枚、1 元硬幣 8 枚。第 3 列至第 6 列以此類推。

註 2：輸出順序以兌換硬幣幣值大的枚數多的先輸出。

## Q1 解說與程式觀摩

```
#include <stdio.h>
#include <stdlib.h>
int changes(int n);
int main(){
    int n, counter;
    printf("Please enter your money:");
    while(scanf("%d", &n) == 1){
        printf("    $50    $10    $5    $1\n");
        printf("-----\n");
        // 呼叫 change(n) 函式，列印兌換組合並且計算兌換方法數回傳
        counter = changes(n);
        printf("Total: %d\n", counter); // 列印兌換方法數
        printf("Please enter your money:");
    }
    return 0;
}

int changes(int n) // 列印所有兌換組合並回傳總共有幾種兌換方法。
{
    int i, j, k, n_10, n_5, n_1;
    int counter = 0; // 用來計數總共有幾種兌換方法
    for(i = n/50; i >= 0; i--) // 每一種 50 元硬幣的可能兌換枚數。
    {
        n_10 = n - 50 * i; // 決定了 50 元硬幣的兌換枚數後，剩餘之金額
        for(j = n_10/10; j >= 0; j--) // 每一種 10 元硬幣的可能兌換枚數。
        {
            // 先計算決定了 50 元以及 10 元硬幣的兌換枚數後剩餘之金額
            n_5 = n_10 - 10 * j;
            for(k = n_5/5; k >= 0; k--) // 每一種 5 元硬幣的可能兌換枚數。
            {
                // 先計算決定了 50、10 以及 5 元硬幣的兌換枚數後剩餘之金額
                n_1 = n_5 - 5 * k; // 剩餘金額全數兌換成 1 元硬幣。
                counter++; // 得到一種兌換組合之後，計數器加 1
                printf("%7d%7d%7d%7d\n", i, j, k, n_1); // 列印兌換組合
            }
        }
    }
    return counter; // 回傳兌換方法總數
}
```

Q9.準備發放消費券時，每一張面額可以被設計為 1000，500，200，100，50 元等不同的大小。

請設計一個程式，分析給定一個金額後所有的發放方式。

(註：消費券金額至少為一百元的整數倍)(972 考題)

輸出結果範例：

請輸入消費券總金額:300

\$1000   \$500   \$200   \$100   \$50

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

Total: 6

註 1：上述輸出表示金額 300 元，共有 6 種發放方法，第 1 列表示可發放 1000 元券 0 張、500 元券 0 張、200 元券 1 張、100 元券 1 張、50 元券 0 張。第 2 列表示可發放 1000 元券 0 張、500 元券 0 張、200 元券 1 張、100 元券 0 張、50 元券 2 張。第 3 列至第 6 列以此類推。

註 2：輸出順序以發放面額大的張數多的先輸出。

註 3：這是進階 Q1 的類似題。可將 50 元面額券視為 1 元硬幣，將 100 元面額券視為 2 元硬幣 ... 等等以此類推，1000 元面額券視為 20 元硬幣。程式寫法即類似進階 Q1。

## Q9 程式觀摩

```
#include<stdio.h>
#include <stdlib.h>

int changes(int n);
int main()
{
    int n, k;
    printf("請輸入消費券總金額: ");
    scanf("%d", &n);
    n = n/50; //由於消費券面額皆為 50 之倍數，故可轉換為基數 20, 10, 4, 2, 1
    printf(" $1000  $500  $200  $100  $50 \n");
    printf("-----\n");
    printf("Total: %d\n", changes(n));
    return 0;
}

int changes(int n) // 列印所有兌換組合並回傳總共有幾種兌換方法。
{
    int i, j, k, h, n_10, n_4, n_2, n_1;
    int counter = 0; // 用來計數總共有幾種兌換方法
    for(i = n/20; i >= 0; i--) // 1000 元消費券的可能發放的張數。
    {
        n_10 = n - 20 * i; // 決定了 1000 元消費券的發放張數後，剩餘之金額
        for(j = n_10/10; j >= 0; j--) // 500 元消費券可能發放的張數。
        {
            // 先計算決定了 1000 元以及 500 元消費券的發放張數後剩餘之金額
            n_4 = n_10 - 10 * j;
            for(k = n_4 / 4; k >= 0; k--) // 200 元消費券的可能發放的張數。
            {
                //先計算決定了 1000、500 以及 200 元消費券的發放張數後剩餘之金額
                n_2 = n_4 - 4 * k;
                for(h = n_2/2; h >= 0; h--)
                {
                    /*先計算決定了 1000、500 、200 元以及 100 元消費券的
                    發放張數後剩餘之金額 */
                    n_1 = n_2 - 2 * h; // 剩餘金額全數發放 50 元券。
                    counter++; // 得到一種發放組合之後，計數器加 1
                    printf("%7d %7d%7d%7d%7d\n",i,j,k,n_1); //列印發放組合
                }
            }
        }
    }
    return counter;    // 回傳發放方法總數
}
```

Q2. 輸入兩個一元多項式  $P$  與  $Q$ ，它們的最高次方不大於 100。多項式與多項式相乘結果仍為多項式。輸入多項式請先輸入其最高冪次方，再依冪次方由低至高的順序依序輸入各項的係數。

輸入共四列。第一列是多項式  $P$  的最高冪次方  $n$ ，第二列依冪次方由低至高的順序依序為多項式  $P$  各項的係數，各係數之間以空白字元隔開。第三列是多項式  $Q$  的最高冪次方  $m$ ，第四列依冪次方由低至高的順序依序為多項式  $Q$  各項的係數，各係數之間以空白字元隔開。

輸入多項式  $P$  與  $Q$  之後，請依冪次方由高至低的順序輸出  $P * Q$  的各項係數，係數與係數之間以兩個空白字元隔開，輸出常數項之後輸出一個 newline ('\\n') 字元。

例如輸入

$$P = x + 1$$

$$Q = x - 1$$

$$\text{則 } P * Q = x^2 - 1$$

輸出

1 0 -1

輸入範例：

1

1 1

1

1 -1

輸出範例：

1 0 -1



## Q2 解說與程式觀摩

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i, j;
    int n, m;
    /*
    n 與 m 分別為多項式 P 與 Q 的最高冪次方；
    將多項式 P 的各項係數依冪次方由低至高的順序依序儲存於陣列 P[]
    將多項式 Q 的各項係數依冪次方由低至高的順序依序儲存於陣列 Q[]
    依題意 P 與 Q 的最高冪次方不大於 100
    因此 P * Q 的最高冪次方不大於 200
    將多項式 P * Q 的各項係數依冪次方由低至高的順序依序儲存於
    陣列 PQ[]
    */
    int P[101], Q[101], PQ[201] = {0};
    /*上述宣告 PQ[201] = {0} 已經將 PQ[] 陣列所有元素初值設為 0 */

    scanf("%d", &n); /*輸入多項式 P 之最高冪次方*/
    for(i = n; 0 <= i; i--) scanf("%d", &P[i]);
    /* 以下輸入多項式 Q */
    scanf("%d", &m); /*輸入多項式 Q 之最高冪次方*/
    for(i = 0; 0 <= i; i++) scanf("%d", &Q[i]);
    /* 下列雙層迴圈進行 P 和 Q 的相乘運算 */
    for(i = 0; i <= n; i++)
    {
        for(j = 0; j <= m; j++)
        {
            PQ[i + j] += P[i] * Q[j];
        }
    }
    /* 下列依高次方項的係數先之順序輸出 P 和 Q 的乘積 */
    for(i = n + m; i >= 0; i--) printf("%d ", PQ[i]);
    return 0;
}
```

Q7.請依照下列資料，設計一個統一發票對獎程式，並至少使用 10 筆統一發票號碼，進行測試，且將中獎情況及金額輸出至螢幕。(961 考題)

96 年 9 月、10 月統一發票中獎號碼單	
月份	9~10 月
特獎	41292387
	同期統一發票收執聯 8 位數號碼與上列號碼相同者獎金 200 萬元
頭獎	32971009
	39376966
	50336841
	同期統一發票收執聯 8 位數號碼與上列號碼相同者獎金 20 萬元
二獎	同期統一發票收執聯末 7 位數號碼與頭獎中獎號碼末 7 位相同者各得獎金 4 萬元
三獎	同期統一發票收執聯末 6 位數號碼與頭獎中獎號碼末 6 位相同者各得獎金 1 萬元
四獎	同期統一發票收執聯末 5 位數號碼與頭獎中獎號碼末 5 位相同者各得獎金 4 千元
五獎	同期統一發票收執聯末 4 位數號碼與頭獎中獎號碼末 4 位相同者各得獎金 1 千元
六獎	同期統一發票收執聯末 3 位數號碼與頭獎中獎號碼末 3 位相同者各得獎金 2 百元

## Q7 解說與程式觀摩之一

A7\_1.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main (void)
```

```
{
    int i, j, k, win;
    int lottery[4] = {41292387, 32971009, 39376966, 50336841}; // 特獎與頭獎中獎號碼
    char bonus[10][40] = {
        "抱歉你沒有中獎！",
        "抱歉你沒有中獎！",
        "抱歉你沒有中獎！",
        "恭喜你中了六獎 2 百元！",
        "恭喜你中了五獎 1 千元！",
        "恭喜你中了四獎 4 千元！",
        "恭喜你中了三獎 1 萬元！",
        "恭喜你中了二獎 4 萬元！",
        "恭喜你中了頭獎 20 萬元！",
        "恭喜你中了特獎 200 萬元！"
    };
    /* 中獎代碼：
        9: 特獎 200 萬元
        8: 頭獎 20 萬元
        7: 二獎 4 萬元
        6: 三獎 1 萬元
        5: 四獎 4 千元
        4: 五獎 1 千元
        3: 六獎 2 百元
        2~0: 沒中獎 */

    int mylottery; //用來儲存發票號碼。
    printf("請輸入你的統一發票號碼： ");
    scanf("%d", &mylottery);
    while(mylottery <= 999999999 && mylottery >= 0) /*
    {
        win = 0;
        // 先檢查是否中特獎？
        if(lottery[0] == mylottery) win=9;
        else
        {
```

```

k = 1000; /* k 用來取出頭獎獎號的末 j 位數字，剛開始先對六獎時 j = 3 */
for(j = 3; j < 9; j++) /* 從六獎 (j == 3) 到頭獎 (j == 8) 逐一對獎 */
{
    for (i = 1; i <= 3; i++) // 針對頭獎三個獎號逐一對獎
    {
        /* 取出第 i 個頭獎號碼的末 j 位數字比對
           如果中獎，設定中獎代碼為 j */
        if (lottery[i] % k == mylottery % k) win = j;
    }
    k = k * 10; /* 準備對下一個更好的獎項 */
}
}
printf("%s\n", bonus [win] );
printf("請輸入你的統一發票號碼： ");
scanf("%d", &mylottery);
}
return 0;
}

```

## Q7 解說與程式觀摩之二

A7\_2.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main (void)
{
    int i, j, k, win;
    char lottery[4][9]= {"41292387","32971009","39376966","50336841"};
    char bonus[10][40] = { "抱歉你沒有中獎!",
                           "抱歉你沒有中獎!",
                           "抱歉你沒有中獎!",
                           "恭喜你中了六獎 2 百元!",
                           "恭喜你中了五獎 1 千元!",
                           "恭喜你中了四獎 4 千元!",
                           "恭喜你中了三獎 1 萬元!",
                           "恭喜你中了二獎 4 萬元!",
                           "恭喜你中了頭獎 20 萬元!",
                           "恭喜你中了特獎 200 萬元!" };
    /* 中獎代碼：
       9: 特獎 200 萬元
       8: 頭獎 20 萬元
       7: 二獎 4 萬元
       6: 三獎 1 萬元
       5: 四獎 4 千元
       4: 五獎 1 千元
       3: 六獎 2 百元
       2~0: 沒中獎 */
    char mylottery[9]; //用來儲存發票號碼。
    printf("請輸入你的統一發票號碼： ");
    gets(mylottery);
    if(strlen(mylottery) != 8)
    {
        printf("發票號碼應為 8 碼。\\n");
        return 0;
    }
    win = 0;
    // 先檢查是否中特獎？
    if(strcmp(lottery[0], mylottery)==0)
```

```

        win=9;
else
{
for(j = 3; j < 9; j++)
    { /*
        j 等於 3 時對六獎，後三碼與某一個頭獎號碼的後三碼相同
        j 等於 4 時對五獎，後四碼與某一個頭獎號碼的後四碼相同
        j 等於 5 時對四獎，後五碼與某一個頭獎號碼的後五碼相同
        j 等於 6 時對三獎，後六碼與某一個頭獎號碼的後六碼相同
        j 等於 7 時對二獎，後七碼與某一個頭獎號碼的後七碼相同
        j 等於 8 時對頭獎，後八碼與某一個頭獎號碼的後八碼相同
        從六獎開始對獎
        */
        for(i = 1; i <= 3; i++)
            // 特獎號碼有三個。
            // i 等於 1 時對第一個頭獎號碼 "32971009"
            // i 等於 2 時對第二個頭獎號碼 "39376966"
            // i 等於 3 時對第三個頭獎號碼 "50336841"
            { // 判斷是否中了第 i 個號碼的第 j 獎。
                k = 8 - j;
                /* 從索引為 k = 8 - j 處開始比對，正好比對末 j 碼。例如對六獎
                時 j 值為 3，從索引為 8 - j = 5 之處開始比對，正好比對末 j 碼。 */
                while(lottery[i][k] == mylottery[k] && mylottery[k] != '\0') k++;
                if(k == 9) win = j;
            }
        }
    }
}
printf("%s\n", bonus [win] );
return 0;
}

```

## 程式能力檢定進階題 Q29, Q11, Q18, Q19

進階題 Q11, Q18 ~ Q19 以及 Q29 這三題主題為文字檔案的開檔、讀檔、改寫檔案內容、以及關檔。首先必須了解程式在處理一個檔案之前必須執行開檔程序。開檔成功之後才可進行讀取檔案內容或寫入之動作。檔案處理完畢之後還必須執行觀檔程序。

開檔函式： `fopen`

讀取檔案函式： `fgets`, `fscanf`, `fgetc` 等

寫入檔案函式： `fputs`, `fprintf`, `fputc` 等

關檔函式： `fclose`

請先閱讀 C 語言手冊，了解這些函式的參數之型態與意義。

英文的文字檔是由字符 (characters) 組成。常見的英文字符編碼標準為 ASCII 編碼。除了英文的大、小寫字母、阿拉伯數字 0~9，以及標點符號之外，有一部份字符是由用來做為呈現文字的控制碼。例如 C 語言的 `newline` (`'\n'`) 字符是用來表示要另起一行輸出文字，有時也用來做為文字的分隔符號。例如 `fgets` 函式會從目前位置開始讀入，直至讀入的字符數目到達指定的上限或是讀到 `newline` (`'\n'`) 字符時停止。通稱文字檔內或由檔案開頭或由上一個 `newline` (`'\n'`) 字符之後開始到下一個 `newline` (`'\n'`) 字符(含)之一段字符為一**列 (line)**。

`char * fgets ( char * str, int num, FILE * stream );`

Get string from stream：由檔案讀入一字串，明確講是由檔案 `stream` 讀入最多 `num` 的字符或於讀入 `newline` (`'\n'`) 或 `end-of-file` (EOF)時停止。並在讀入的字符之後補上 `NULL` 字符 (`'\0'`)。**請注意 `newline` (`'\n'`)會被放入輸入陣列中。**

Q29. 請建立 C 程式輸入檔案名稱後，讀取文字檔案內容計算總共有幾個字元，在讀完後顯示檔案的總字元數。(992 新增)

### A29 程式觀摩與解說

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    char str[256]; /*用來存放檔案路徑與檔名，
                  不可超出 128 字符 */
    FILE *fp; /* 宣告檔案指標 (file pointer) */
    int count = 0; /* 用來計數字元數 */
    int c; /* 儲存讀入的字元 */
    printf("Input filename: ");
    gets(str); /* 讀取用來輸入的唯讀檔案之檔名 */
    fp = fopen(str,"r"); /* 開啟用來輸入的唯讀檔案 */
    if(fp == NULL)
    {
        printf("Error opening file"); /* 開檔失敗列印錯誤訊息 */
        return 1;
    }
    c = fgetc(fp); /* fgetc 函式讀入一個字元 */
    while(c != EOF) /* c != EOF 表示成功讀入一個字元 */
    {
        count++; /* 成功讀入一個字元，計數器加 1 */
        c = fgetc(fp); /* 繼續讀入下一個字元 */
    }
    fclose(fp); /* 關閉讀入的檔案 */
    printf("字元數為： %d\n", count); /* 依題意輸出檔案的字元數 */
    return 0;
}
```



Q18. 請建立 C 程式輸入檔案名稱後，讀取文字檔案內容計算總共有幾行(列)，在讀完後顯示檔案的總行數。

### Q18 程式觀摩與解說

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    FILE *fp;          /* 宣告檔案指標 (file pointer) */
    char filename[128]; /* 用來存放檔案路徑與檔名，
                        不可超出 128 字符 */
    char line[512];     /* 用來存放讀入的一列資料，
                        假設一行不超出 512 字符 */
    int count=0;        /* 用來計數列數 */
    printf("輸入檔案名稱： ");
    scanf("%s", filename); /* 讀入檔案路徑與檔名 */

    fp = fopen(filename, "r"); /* 開啟唯讀檔案 */
    if ( fp != NULL )
    { /* 開檔成功 */
        while ( fgets(line, 512, fp) != NULL ) /* 只要成功讀入一行，
                                                繼續讀入 */
        {
            count++; /* 每成功讀入一行，計數器加 1 */
        }
        printf("檔案內容共%d行\n", count); /* 依提議要求輸出列數 */
        fclose(fp); /* 成功讀完檔案，關閉檔案 */
    }
    else /* 開檔失敗 */
        printf("錯誤：檔案開啟錯誤...\n");
    return 0;
}
```

Q19.請建立 C 程式在輸入程式碼檔案名稱後（可以使用 scanf()函數或命令列參數），讀取整列程式碼後，在每一列程式碼前加上列號，輸出成 output.txt 檔案。

### A19 程式觀摩與解說

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    FILE *fp,*fp1;
    char filename[128], line[512];
    char output[20] = "output.txt";
    int count=0;
    fp1 = fopen(output, "w");    /* 開啟用來輸出的檔案 */
    printf("輸入檔案名稱： ");
    scanf("%s", filename);      /* 讀取用來輸入的唯讀檔案之檔名 */
    fp = fopen(filename, "r");   /* 開啟用來輸入的唯讀檔案 */
    if ( fp != NULL )
    {
        while( fgets(line, 512, fp) != NULL ) /* 只要成功讀入一行，
                                                繼續讀入 */
        {
            count++;    /* 每成功讀入一行，計數器加 1 */
            fprintf(fp1, "%2d: %s", count, line); /* 讀入成功，加上列號寫入
                                                    output.txt 檔案 */
        }
        printf("檔案內容共%d行\n", count); /* 輸出檔案之列數 */
        fclose(fp); /* 關閉讀入的檔案*/
    }
    else
        printf("錯誤：檔案開啟錯誤...\n");
    fclose(fp1); /* 關閉輸出的檔案*/
    return 0;
}
```

Q11. 請撰寫一個程式，可以讀出英文文字檔 A.TXT 內容，統計其中所含字母數目（不含空白）及單字的數目。所謂英文字母是指大、小寫英文字符(以 ASCII 編碼)。所謂單字為檔案開頭至第一個非英文字母之前之一段英文字母；或是檔案內某個非英文字母之後開始至下一個非英文字母之前之一段不為空的英文字母；或是檔案內最後一個非英文字母之後開始至檔案結束之前之一段不為空之英文字母。例如 A.TXT 內容如下：

**John is a good student. He will pass this exam.**

則輸出結果如下：

**36 characters**

**10 words**

## Q11 程式觀摩與解說

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    FILE *fpl;
    int word=0, character=0, inword = 0, alphaFlag, c;
    fpl = fopen("A.TXT", "r"); /* 開檔 */
    if(fpl == NULL) /* 開檔不成功，列印錯誤訊息之後程式停止執行 */
    {
        printf("Error opening file!");
        return 0;
    }
    while((c = fgetc(fpl)) != EOF) /* 逐一讀取字元直至讀入 EOF 為止 */
    {
        /* 檢查是否為英文字母，是則 alphaFlag 為 1；否則 alphaFlag = 0。
        if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z')) alphaFlag = 1;
        else alphaFlag = 0;
        /* c 為英文字母，且正在讀取單字狀態，英文字母數目加 1 */
        if(inword == 1 && alphaflag == 1) character++;
        else if (inword == 1 && alphaflag == 0)
        {
            /* 讀取單字狀態中，讀到非英文字母 */
            word++; /* 單字數目加 1 */
            inword = 0; /* 離開讀取單字狀態 */
        }
        else if (inword == 0 && alphaflag == 1)
            /* 不是在讀取單字狀態中，讀到英文字母 */
        {
            character++; /* 英文字母數目加 1 */
            inword = 1; /* 進入讀取單字狀態 */
        }
    }
    printf("%d characters \n", character); /* 列印英文字母數目 */
    printf("%d words \n", word); /* 列印英文單字數目 */
    fclose(fpl); /* 關閉檔案 */
    return 0;
}
```

## 程式能力檢定進階題 Q12, Q17

Q12. 試撰寫一程式可從鍵盤輸入二個  $3 \times 3$  的矩陣，並由程式計算後顯示出兩矩陣相加之後的結果。

**例如：**

矩陣 a:

請輸入(1,1) 的值(列,行): 10

請輸入(1,2) 的值(列,行): 29

請輸入(1,3) 的值(列,行): 30

請輸入(2,1) 的值(列,行): 48

請輸入(2,2) 的值(列,行): 51

請輸入(2,3) 的值(列,行): 28

請輸入(3,1) 的值(列,行): 38

請輸入(3,2) 的值(列,行): 11

請輸入(3,3) 的值(列,行): 9

矩陣 b:

請輸入(1,1) 的值(列,行): 156

請輸入(1,2) 的值(列,行): 38

請輸入(1,3) 的值(列,行): 11

請輸入(2,1) 的值(列,行): 28

請輸入(2,2) 的值(列,行): 68

請輸入(2,3) 的值(列,行): 12

請輸入(3,1) 的值(列,行): 67

請輸入(3,2) 的值(列,行): 90

請輸入(3,3) 的值(列,行): 18

**輸出結果**

矩陣 a:

10 29 30

48 51 28

38 11 9

矩陣 b:

156 38 11

28 68 12

67 90 18

開始計算...

答案:

166 67 41

76 119 40

105 101 27

## A12 程式觀摩與解說

```
#include <stdio.h>
#define COL 3          /* 定義行數 */
#define ROW 3          /* 定義列數 */
void input(int[][COL]); /* 輸入數值的函式 */
void output(int[][COL]); /* 輸出結果的函式 */
/* 計算矩陣相加的函式 */
void add (int mat_1[ROW][COL], int mat_2[ROW][COL], int sum[ROW][COL]);

int main (void)
{
    /* 主程式讀入矩陣 a 與矩陣 b 之後，
       呼叫矩陣相乘函式然後輸出乘積。 */
    int a[ROW][COL] = {{0}}, b[ROW][COL] = {{0}}; /* 宣告矩陣 a 與矩陣 b */
    int c[ROW][COL] = {{0}}; /* 宣告矩陣儲存兩矩陣之和 */
    printf("矩陣 a:\n"); /* 提示輸入矩陣 a */
    input(a);             /* 呼叫 input 函式，輸入矩陣 a */
    printf("矩陣 b:\n"); /* 提示輸入矩陣 b */
    input(b);             /* 呼叫 input 函式，輸入矩陣 b */
    printf("矩陣 a:\n"); /* 提示輸出矩陣 a */
    output(a);            /* 呼叫 output 函式，輸出矩陣 a */
    printf("矩陣 b:\n"); /* 提示輸出矩陣 b */
    output(b);            /* 呼叫 output 函式，輸出矩陣 b */
    printf("開始計算...\n"); /* 提示將開始執行矩陣相乘 */
    printf("答案:\n");
    add (a,b, c);          /* 呼叫 cal 函式，執行矩陣相乘，並輸出結果 */
    output(c); /* 呼叫 output 函式，輸出矩陣相乘結果矩陣 */
    return 0;
}
```

/\* 定義從鍵盤輸入資料的函式 \*/

```
void input(int matrix[ROW][COL])
{
    int i, j;
    for(i = 0; i < ROW; i++)
        for(j = 0; j < COL; j++)
        {
            printf("請輸入(%d, %d) 的值(列,行): ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
}
```

/\* 定義輸出陣列內容的函式 \*/

```
void output(int matrix[ROW][COL])
{
    int i, j;
    for (i = 0; i < ROW; i++)
    {
        for(j = 0; j < COL; j++) printf("%3d ", matrix[i][j]);
        printf("\n");
    }
}
```

/\* 定義計算兩矩陣相加的函式 \*/

```
void add (int mat_1[ROW][COL], int mat_2[ROW][COL], int sum[ROW][COL])
{
    int i, j;
    for (i = 0; i < ROW; i++)
    {
        for (j = 0; j < COL; j++)
        {
            sum[i][j] = mat_1[i][j] + mat_2[i][j];
        }
    }
    return;
}
```

Q17. 請設計 C 程式宣告 3X3 的二維陣列，陣列值使用亂數產生，其範圍是 1~50，然後顯示二維陣列的元素值，並且計算每一列的總和和平均，最後計算出陣列對角線元素的總和。

#### A17 程式觀摩與解說

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h> /* 為了呼叫時間函式 time()， 需要加上這列*/
int main(void)
{
    int arr[3][3];
    int i, j, total;
    srand(time(NULL)); /* 透過執行這列敘述，確保每次執行產生不同亂數 */
    /* 下列迴圈透過呼叫亂數產生函式，產生並且輸出矩陣 */
    for ( i = 0; i < 3; i++)
    {
        for ( j = 0; j < 3; j++)
        { /* 下列利用亂數產生函式產生 1 到 50 之間的值給 arr[i][j] */
            arr[i][j] = (rand() % 50)+1;
            printf("%2d ", arr[i][j]);
        }
        printf("\n");
    }
    for ( i = 0; i < 3; i++)
    { /* 列印每列的和與其平均值 */
        total = 0;
        for ( j = 0; j < 3; j++) total += arr[i][j]; /* 計算第 i 列的和 */
        printf("Rows:%d Total=%d Average=%f\n",i,total,total/3.0);
    }
    total = 0;
    /* 下列迴圈計算對角線的和 */
    /* 對角線的元素其行與列的索引相同 */
    /* 本題 3x3 矩陣之對角線元素為 arr[0][0] arr[1][1], arr[2][2] */
    for ( i = 0; i < 3; i++) total += arr[i][i]; /* 計算對角線元素和 */
    printf("對角線的和= %d\n", total);
    return 0;
}
```



Q15. 請撰寫一個程式，包含一個函數：double my\_fun(double x, int n)用來計算下面的數學式：

$$\text{my\_fun}(x, n) = \sum_{k=1}^n \frac{x^k}{k!} = \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

首先請在螢幕上輸入  $x$  及  $n$ ，並在螢幕上列印出  $\text{my\_fun}(x, n)$  之計算結果。

結果範例如下：

1. 請輸入  $x = 0.1$

請輸入  $n = 5$

$\text{my\_fun}(0.100000, 5) = 0.105171$

2. 請輸入  $x = 0.2$

請輸入  $n = 5$

$\text{my\_fun}(0.200000, 5) = 0.221403$

## 解說與程式觀摩：

我們提供兩個程式讓大家觀摩。第二個程式(A15-2)呼叫函式逐一計算級數的每一項再累加求得級數的和。計算級數的每一項時直接計算  $k!$  與  $x^k$ 。 $k!$ 是指數成長的函數， $k$  值不到 20， $k!$  得值就超過 unsigned int 型態變數所能儲存的值之上限。因此這個演算法所能計算的級數最多不到 20 項。

觀察得知級數的第  $k$  項等於第  $k-1$  項乘以  $x/k$ ，因此計算第  $k$  項並不需要如 A15-2 程式般計算  $k!$  與  $x^k$ 。A15-1 程式將第  $k-1$  項乘以  $x/k$  得到第  $k$  項的值是一個比較簡潔而且有效率的算法。利用此法計算，可以避開計算過程中產生變數無法儲存的中間值。

請務必好好觀察這兩個程式的差異。程式 A15-1 不只是簡短，除了效率，它還可以計算級數項數  $n$  比較大的值。程式 A15-2 根本無法計算級數項數  $n$  超過 20 的級數。

這一題為**基礎題庫 Q28**的類似題。

### A15-1

```
#include <stdio.h>
#include <stdlib.h>
double my_fun (double x, int n);

int main(void)
{
    double x;
    int n;
    printf("請輸入 x = ");
    scanf("%lf", &x); // 讀入 double 變數
    printf("請輸入 n = ");
    scanf("%d", &n);
    printf("my_fun(%lf, %d) = %lf\n", x, n, my_fun(x, n));
    return 0;
}

double my_fun(double x, int n)
{
    int k;
    double lastTerm = 1.0; /* 用來記錄級數的第 k 項的值，初始為第 0 項 */
    double result = 0.0; /* 用來記錄級數累計到第 k 項的和，初始累計至第 0 項 */
    for(k = 1; k <= n; k++)
    {
        // 先計算級數的第 k 項的值，計算前 lastTerm 儲存的是第 k - 1 項的值
        lastTerm = lastTerm * (x / k);
        // 將級數的第 k 項累加入 result
        result = result + lastTerm;
    }
    return result;
}
```

## A15-2

```
#include <stdio.h>
#include <stdlib.h>
double my_fun(double, int); /* 宣告函數 my_fun()的原型 */
double power(double, int); /* 宣告函數 power()的原型 */
int fac(int);             /* 宣告函數 fac()的原型 */
int main(void)
{
    double x;
    int n;
    printf("請輸入 x = ");
    scanf("%lf", &x);
    printf("請輸入 n = ");
    scanf("%d", &n);
    printf("my_fun(%lf, %d) = %lf\n", x, n, my_fun(x, n));
    return 0;
}

double my_fun(double x, int n)
{
    int i;
    double sum=0.0;
    for(i = 1; i <= n; i++)
        sum += power(x, i)/fac(i);
    return sum;
}

double power(double x, int n) /* power()函數的定義：計算  $x^n$  */
{
    //
    int i;
    double pow = 1.0;
    for(i = 1; i <= n; i++) /* for() 迴圈，用來將 x 連乘 n 次 */
        pow = pow * x;
    return pow;
}

int fac(int n) /* 自訂函數 fac()，計算 n! */
{
    int i, total = 1;
    for(i = 1; i <= n; i++)
        total *= i;
    return total;
}
```

## 程式能力檢定進階題 Q21

Q21. 醜數(ugly number) 的定義是一個數只含有 2, 3, 5 的質因數。數列  
1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...  
列出了前 11 個 ugly number。為了方便起見，1 也算是 ugly number。  
請寫一個程式求出第 1500 個 ugly number。

### 輸入要求：

這個程式不需輸入。

在解說這個題目時，我們讓程式輸入整數  $n$ 。

### 輸出要求：

輸出依由小到大的順序第  $n$  個醜數。依題意，1 是第一個醜數，2 是第二個醜數等等。

輸出只須印出一行字，如下所示，並將  $\langle n \rangle$  以  $n$  的值取代， $\langle \text{number} \rangle$  以第  $n$  個醜數取代：

The  $\langle n \rangle$ 'th ugly number is  $\langle \text{number} \rangle$

### 輸入範例：

456

### 輸出範例：

The 456'th ugly number is 589824.

## Q21. 解說與程式觀摩

如何判斷正整數  $n$  是否為醜數?

除了 1 以外，所有醜數皆可以被 2 或 3 或 5 整除，因此

如果  $n == 1$ ，那麼  $n$  為醜數；

否則如果  $n$  可被 2 整除，那麼  $n$  為醜數若且唯若  $n$  除以 2 的商 ( $n/2$ ) 為醜數；

否則如果  $n$  可被 3 整除，那麼  $n$  為醜數若且唯若  $n$  除以 3 的商 ( $n/3$ ) 為醜數；

否則如果  $n$  可被 5 整除，那麼  $n$  為醜數若且唯若  $n$  除以 5 的商 ( $n/5$ ) 為醜數；

否則  $n$  不為醜數。

先寫個判斷整數  $k$  是否為醜數之函式，命名為 `ugly_num`，函式原型如下：

```
int ugly_num(int k);
```

若  $k$  為醜數，函式 `ugly_num` 值為 1；否則函式 `ugly_num` 值為 0。

`ugly_num`，函式定義如下：

```
int ugly_num(int k)
{
    if (k < 1) return 0;          /* 小於 1 的數不為醜數 */
    while (k % 2 == 0) k = k / 2; /* 只要 2 可以整除 k，將 k 除以 2 取商 */
    while (k % 3 == 0) k = k / 3; /* 只要 3 可以整除 k，將 k 除以 3 取商 */
    while (k % 5 == 0) k = k / 5; /* 只要 5 可以整除 k，將 k 除以 5 取商 */
    if (k == 1) return 1;         /* 假如 k 值為 1，表示 k 為醜數。*/
    else return 0;
}
```

有了上列判斷整數  $k$  是否為醜數的函式之後，主程式如下：

```
int main ()
{
    int k, count, n;
    scanf("%d", &n);
    count = 0;
    k = 0;
    while (count < n) /* 從 1 開始逐一檢查 k 是否為醜數，直至發現第 n 個醜數 */
    {
        k = k + 1;
        if (ugly_num(k) == 1) count = count + 1; /* 若 k 為醜數，醜數數目加 1 */
    }
    printf ("The %d'th ugly number is %d.", n, k);
    return 0;
}
```

Q23. 一個正整數的鏡反數(mirror-image number)為原來數字由右至左重新組合後的新數字，可定義函數  $m(n)$  來表示，如  $m(123)=321$ 。如果一個數字的鏡反數之平方等於該數字平方之鏡反數，即  $m(n) * m(n) = m(n * n)$ ，則該數字可稱為平方鏡反數。請找出 1000 以下所有的平方鏡反數，並依序列印出來。(972 新增)

輸出範例：

$$m(11) \times m(11) = m(11 \times 11)$$

$$m(12) \times m(12) = m(12 \times 12)$$

...

$$m(n) \times m(n) = m(n * n)$$

共有\_\_個平方鏡反數。

注意：

任何一位數，0, 1, 2, 3, 4, 5, 6, 7, 8, 或 9 皆為平方鏡反數。於計算平方鏡反數的個數時，皆不計算在內。

如果某數  $x$  為平方鏡反數，乘以 10 之後仍然是平方鏡反數。於計算平方鏡反數的個數時，不列入 10 的倍數。例如 10、110 等均為平方鏡反數，但因為他們也是 10 的倍數，因此於計算平方鏡反數的個數時，皆不被計算在內。

想法：

先寫個函式計算一數  $n$  的鏡反數：

```
int m(int n)
{
    int k;
    k = n % 10;    /* n 除以 10 取餘數得 n 的十進位表示法的個位數 */
    n = n / 10;     /* n 除以 10 取商 */
    while (n != 0)
    {
        /* 只要 n 不等於 0 就取其個位數加上 k * 10 */
        k = k * 10 + (n % 10);
        n = n / 10;
    }
    return k;
}
```

例如求 123 的鏡反數的程序如下：

$$(((3 * 10) + 2) * 10) + 1 = 321$$

有例如求 45369 的鏡反數的程序如下：

$$((((9 * 10) + 6) * 10 + 3) * 10 + 5) * 10 + 4 = 96354$$

然後我們就可以利用上述的鏡反數函式來判斷  $n$  是否為平方鏡反數了：

```
int squareM (int n)
{
    /* 檢查  $n$  是否為平方鏡反數 */
    /* 當  $n$  是平方鏡反數時將 mFlag 直設為 1 */
    int mFlag = 0, MN;
    MN = m(n);
    if (m(n * n) == MN * MN) mFlag = 1;
    return mFlag;
}
```

有了上述兩個函數之後，我們就可以逐一測試從 11 開始到 999 之間的數是否為平方鏡反數：

```
#include <stdio.h>
#include <stdlib.h>

int squareM (int n);
int m(int n);

int main(void)
{
    int n, count=0;
    for(n = 11; n < 1000; n++)
    {
        /* 不可以是 10 的倍數 */
        if((n % 10 != 0) && (squareM (n) == 1))
        {
            printf("m(%d) * m(%d) = m (%d * %d) \n", n, n, n, n);
            count++;
        }
    }
    printf("共有 %d 個平方鏡反數 \n", count);
    return 0;
}
```

## 程式能力檢定進階題 Q26

Q26. 日常生活中的日期寫法有文字與數字的不同，英文慣用文字式的寫法是：

16 September 2000，一般也常用數字表示日期:9/16/2000。請撰寫一 C 程式可以將上述文字式日期的表示法轉換為數字式的表示法 **(991 新增)**

例如：

輸入文字式日期> 15 January 1993

轉換日期結果

15 January 1993 = 1/15/1993

想法：

我們先需要將英文的月份轉換成數字的函式。這個函式的參數與回傳值之間的關係如下：

**(寫這個程式你應先背 12 個月份的英文名稱)**

January	→ 1
February	→ 2
March	→ 3
April	→ 4
May	→ 5
June	→ 6
July	→ 7
August	→ 8
September	→ 9
October	→ 10
November	→ 11
December	→ 12

```
int month2Num(char* mName) /* 參數為字串，是英文的月份名稱 */
{
    int i;
    char * month_names[12] = {
        "January",
        "February",
        "March",
        "April",
        "May",
        "June",
        "July",
        "August",
        "September",
        "October",
        "November",
        "December"};
    for(i = 0; i < 12; i++)
    {
        if (strcmp(mName, month_names[i]) == 0)
            return i + 1;
    }
    return -1; /* 月份名稱不對，回傳 -1 */
}
```



有了上述程式之後，我們就可以輕鬆的寫主程式了：

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int month2Num(char* mStr);

int main()
{
    char monthName[15];
    int day, month, year;
    printf("輸入文字式日期> ");
    scanf("%d", &day); /* 讀入日 */
    scanf("%s", monthName); /* 讀入月份英文名稱 */
    scanf("%d", &year); /* 讀入西元年 */
    month = month2Num(monthName); /* 呼叫函式將英文月份名稱轉成數字 */
    if (month < 1)
    {
        printf("月份名稱錯誤!\n");
        return -1;
    }
    printf("%d %s %d = %d/%d/%d\n", day, monthName, year, month, day, year);
    return 0;
}
```

Q31. 請寫一個程式讀入二維的陣列 A (3x4 的陣列)，以求得其轉置矩陣  $A^T$  (4x3 的陣列)。  
註： $A^T$  中的元素  $A^T_{ij}$  恰為 A 中的元素  $A_{ji}$ 。(100.1 新增)

說明：

寫這題程式你(妳)應先知道何謂轉置矩陣(transpose matrix)。根據題目註解， $m \times n$  矩陣 (二維陣列) A 的轉置矩陣  $A^T$  為  $n \times m$  的矩陣。矩陣  $A^T$  的第一列為矩陣 A 的第一行；矩陣  $A^T$  的第二列為矩陣 A 的第二行；…依此類推，矩陣  $A^T$  的第  $n$  列為矩陣 A 的第  $n$  行。換言之，矩陣  $A^T$  的第  $i$  列第  $j$  行的元素等於矩陣 A 的第  $i$  行第  $j$  列的元素。假設矩陣 T 為矩陣 A 的轉置矩陣，則  $T[i][j] = A[j][i]$ 。例如下列  $5 \times 3$  的矩陣

$$A = \begin{pmatrix} 12 & 8 & 15 \\ 3 & 9 & 6 \\ 21 & 1 & 35 \\ 18 & 22 & 46 \\ 7 & 11 & 85 \end{pmatrix}$$

之轉置矩陣如下：

$$A^T = \begin{pmatrix} 12 & 3 & 21 & 18 & 7 \\ 8 & 9 & 1 & 22 & 11 \\ 15 & 6 & 35 & 46 & 85 \end{pmatrix}$$

有關轉置矩陣的相關概念在「線性代數」課應該學過。**原題意只要讀入一個  $3 \times 4$  的矩陣並輸出其轉置矩陣。我們將放寬限制，允許使用者先輸入矩陣之列數  $m$ ，再輸入矩陣之行數  $n$  後，再依序讀入矩陣的第一列之元素、第二列之元素 … 等直到讀入陣列之所有元素後再直接輸出該矩陣之轉置矩陣。原題意並未規定矩陣元素的形態，我們將矩陣元素之型態宣告為 **double** 型態。這題雖然有點煩，但是程序直覺，屬於簡單題。**

### Q31 解說與程式觀摩

```
#include <stdio.h>
#include <stdlib.h>
/* 先假設你讀入的矩陣的列數與行數均不超過 100 */
#define MAX_ROWS 100
#define MAX_COLS 100

int main()
{
    int i, j, n, m;
    /* 宣告的空間可大於你讀入的空間沒關係 */
    double A[MAX_ROWS][MAX_COLS]; /* 宣告空間儲存將讀入的矩陣 */
    double AT[MAX_COLS][MAX_ROWS]; /* 宣告儲存轉置矩陣的空間 */
    scanf("%d", &m); /* 讀入矩陣之列數 m */
    scanf("%d", &n); /* 讀入矩陣之行數 n */
    /* 讀入資料 */
    for(i=0; i < m; i++) /* 對矩陣的每一列 */
    {
        for(j=0; j < n; j++) /* 對矩陣第 i 列的每一個元素 */
        { /* 下列的輸入提示可省略 */
            /* printf("Enter value of a[%i][%i]: ", i, j); */
            scanf("%lf", &A[i][j]); /* 讀入矩陣第 i 列第 j 行的元素值 */
        }
    }

    /* 求轉置矩陣，請注意轉置矩陣有 n 列 m 行 */

    for(i=0; i < m; i++) /* 對轉置矩陣的每一行 */
    {
        for(j=0; j < n; j++) /* 對轉置矩陣第 i 行的每一個元素 */
        {
            AT[j][i] = A[i][j]; /* 寫入轉置矩陣第 i 行第 j 列的元素值 */
        }
    }

    /* 列印原矩陣 */
    printf("A =\n");
    for(i=0; i < m; i++)
```

```

    {
        putchar('|');
        for(j=0; j < n; j++)
            {
                printf("%10f", A[i][j]);
            }
        puts("|");
    }

/* 列印轉置矩陣 */
printf("AT =\n");
for(i=0; i < n; i++)
    {
        printf("| ");
        for(j=0; j < m; j++)
            {
                printf("%10f", AT[i][j]);
            }
        puts(" |");
    }
return(0);
}

```

範例輸入：

```

5 3
12 8 15
3 9 6
21 1 35
18 22 46
7 11 85

```

Q33. 數學大師歐勒 (Euler, 1707-1783), 找到了一個計算圓周率的無窮乘積:

$$\frac{\pi}{2} = \frac{3}{2} \times \frac{5}{6} \times \frac{7}{6} \times \frac{11}{10} \times \frac{13}{14} \times \frac{17}{18} \times \frac{19}{18} \times \frac{23}{22} \times \dots$$

這個公式裡, 所有的分子都是大於 2 的質數, 分母則是不能被 4 整除, 且最靠近分子的偶數。試撰寫一函數 `double Euler(int n)`, 用來估算圓周率的值到第  $n$  項, 並計算 `Euler(10)`、`Euler(100)`、`Euler(1000)` 與 `Euler(10000)` 的結果。

輸出結果:

`Euler(10)= 3.10152`

`Euler(100)= 3.13398`

`Euler(1000)= 3.13773`

`Euler(10000)= 3.14074`

以下的示範程式, 讓使用者輸入一個正整數  $n$ , 輸出尤拉圓周率公式計算至第  $n$  項的圓周率  $\pi$  的值。

尤拉圓周率公式要計算至第  $n$  項, 必需從 3 開始, 由小到大依序排列  $n$  個質數, 作為每項的分子, 並據以求每項之分母。最簡便的方法是利用篩法減質數表。令 `prime_table[]` 為質數表。其用法如下:

假如  $k$  為質數, 則 `prime_table[k] == PRIME`;

否則 `prime_table[k] == NONPRIME`。

要知道  $k$  是否為質數, 只要判斷 `prime_table[k]` 是否等於 `PRIME` 即可。

### Q33 解說與程式觀摩

A33、

```
#include <stdio.h>
#include <stdlib.h>
```

```
#define PRIME 1
#define NONPRIME 0
#define MAX_PRIME 1000000
```

```
/*
分子是 numerator
分母是 denominator
```

質數表 prime\_table[] 的用法：  
假如 k 為質數，則 prime\_table[k] == PRIME；  
否則 prime\_table[k] == NONPRIME。

下列 construct\_prime\_table(char prime\_table[]) 函式用來建構質數表：

```
*/
int construct_prime_table(char prime_table[]);

int main()
{
    char prime_table[MAX_PRIME + 1];
    int i, j, k, n, numerator, denominator;
    double PI;
    /* 建構 1 到 10000 之間(含) 的質數表 */
    printf("There are %d primes!\n", construct_prime_table(prime_table));
    scanf("%d", &n);
    PI = 1.0;
    k = 3; /* 尤拉公式的第一項分子 */
    for(i = 1; i <= n; i++)
    {
        /* 搜尋下一個質數 */
        while(prime_table[k] == NONPRIME && k <= MAX_PRIME) k++;
        if(k > MAX_PRIME) break; /* 檢查是否超出質數表之範圍 */
        /* 以下計算第 i 項的分子與分母 */
        if(prime_table[k] == PRIME)
        {
```

```

        numerator = k;
        if ((k+1) % 4 != 0) denominator = k + 1;
        else denominator = k - 1;
    }
    /* 計算圓周率至第 i 項的值 */
    PI = PI * ((double)numerator/(double)denominator);
    j = i; /* 保留 i 值以備萬一無法找到第 n 項之分子時，
           可列印出計算至最後一項的值 */
    k++;
}
printf("計算至第 %d 項的圓周率為 %10.9lf\n", j, 2.0 * PI);
return 0;
}
/* 以下函式建構質數表 */
int construct_prime_table(char prime_table[])
{
    int i, k, num = 0;
    for (i = 0; i <= MAX_PRIME; i++) prime_table[i] = PRIME; /* 初始化質數的符號 */
    /* 假如 i 不是質數，prime_table[i]將被改重設為 NOTPRIME */
    for (i = 2; i <= MAX_PRIME; i++)
    {
        /* 依由小到大的順序，逐一檢視 i 是否為質數 */
        if (prime_table[i] == PRIME) /* 如果 i 是質數 */
        {
            /* 以下 while 迴圈將 i 的倍數 k (不大於 n 的) 設定為非質數 */
            for(k = 2 * i; k <= MAX_PRIME; k = k + i) /* 計算 k = i * j 之後，j 值被加 1 */
            {
                prime_table[k] = NONPRIME; /* 設定 k (為 i 的倍數) 不是質數 */
            }
        }
    }
    /* 經過上列迴圈之後，小於 n 且沒被設定為非質數的皆為質數 */
    for (i = 2; i <= MAX_PRIME; i++)
        if(prime_table[i] == PRIME) num++;
    return num;
}

```

Q37. 一維陣列常見的應用之一，是作為標記(mark)之用。例如要印出 2 到 1000 之間的質數，我們可以宣告一維陣列 `char mark[1000]`，並且先初設元素均為 1。接著從 2 開始，凡是 2 的倍數，就將標記設為 0，凡是 3 的倍數也將標記設為 0，以此類推…。最後掃瞄陣列，元素值仍為 1 的就是質數。請寫一程式，接受輸入一正整數  $n$ ，由小到大依序印出從 2 到  $n$  之間的所有質數， $0 < n < 10000$ 。質數與質數之間以一個空白字元隔開，輸出最後一個質數之後，輸出一個 newline ('`\n`') 字元。(101.1-2 新增)

這是一個製作質數表的方法。通常簡稱為「篩法」。列出所有正整數。從 2 開始，刪掉 2 的所有倍數（註記它們不是質數）。再找下一個未被刪掉的（未被註記為不是質數）數字，找到 3，刪掉 3 的倍數。找下一個未被刪掉的數字，找到 5，刪掉 5 的倍數。如此不斷下去，就能刪掉所有合數，找到所有質數。

這一題與基礎題庫的 Q35 相似。基礎題庫的 Q35 額外要求輸出每十個質數一行，最後再輸出 1 至  $n$  之間質數的個數。



### Q37 解說與程式觀摩

```
#include <stdio.h>
#include <stdlib.h>
#define MAXITEM 10000
#define PRIME 1
#define NOTPRIME 0
int main(void)
{
    int    n, i, j, k;
    char Mark[MAXITEM];
    printf("請輸入大於 2 且小於 %d 之正整數 n: ", MAXITEM);
    scanf("%d", &n); /* 輸入 n */
    if ((n >= MAXITEM) || (n < 2))
    {
        printf(" 輸入錯誤!\n");
        return 1;
    }
    for (i = 0; i <= n; i++) Mark[i] = PRIME; /* 初始化質數的符號 */
    /* 假如 i 不是質數，Mark[i]將被改重設為 NOTPRIME */
    for (i = 2; i <= n; i++)
    { /* 依由小到大的順序，逐一檢視 i 是否為質數 */
        if (Mark[i] == PRIME) /* 如果 i 是質數 */
        { /* 以下 while 迴圈將 i 的倍數 k (不大於 n 的) 設定為非質數 */
            j = 2;
            while ((k = i * j++) <= n) /* 計算 k = i * j 之後，j 值被加 1 */
            {
                Mark[k] = NOTPRIME; /* 設定 k (為 i 的倍數) 不是質數 */
            }
        }
    }
    /* 經過上列迴圈之後，小於 n 且沒被設定為非質數的皆為質數 */
    printf("\n The Prime Numbers between 2 and %d are : \n", n);
    for (i = 2; i <= n; i++) /* 由小到大依序列印出不比 n 大的所有質數 */
        if (Mark[i] == PRIME)
            printf(" %d ", i);
    printf("\n");
    return(0);
}
```