阿拉伯數字轉中文數字

丁培毅

實習目標：

1. 進一步認識輸入串流的模型, 同樣的輸入, 不同的解讀方式可以導致不同的處理方式
2. 瞭解程式如何轉換資料的表達形式 – 對照函數 / 對照表格
3. 兩個浮點數相等的比對需要考量資料表示法的誤差
4. 中文字元陣列的長度和編碼方式有關, 字元指標陣列的運用
5. 練習運用迴圈設計進位制轉換的演算法 (2 進位 -> 10 進位)
6. 字元變數與 ASCII 編碼, scanf(" %c", &x); 的用法
7. char buf[100]; scanf("%[0-9]", buf) 的用法
8. 迴圈的設計與偵錯 (每一步執行過程的掌握)

阿拉伯數字轉中文數字

請撰寫一個程式，輸入一個非負整數，印出對應之中文數字

程式輸出範例：

```
請輸入一個非負整數：24622192
二四六二二一九二
```


這個題目的描述很簡單，但是在撰寫程式時會有一些地方需要進一步釐清，不同的寫法能夠處理的情況不太一樣, 例如：

1. 非負整數允許多少位數? (8位數? 9位數? 17? 20? 30?)

```
請輸入一個非負整數：24622192
二四六二二一九二
請輸入一個非負整數：1234567890
一二三四五六七八九〇
```

2. 如果輸入整數前面出現 0 的話，要不要印出對應的中文?

```
請輸入一個非負整數：00123
〇〇一二三
請輸入一個非負整數：00123
一二三
```


分析

1. 這個程式最主要需要完成的工作是**資料轉換**：0 ⇒ 〇,1 ⇒ 一, 2 ⇒ 二, 3 ⇒ 三, 4 ⇒ 四, 5 ⇒ 五, 6 ⇒ 六, 7 ⇒ 七, 8 ⇒ 八, 9 ⇒ 九，資料的轉換也是我們日常生活中常常需要做的工作，例如：老師看到學號要換成同學的姓名，助教看到同學名單要換成手機號碼，電信公司的收費與費率對照，單位郵件收發看到員工姓名找到員工的單位，三個五十元銅板加上兩個十元、三個五元等效於多少一元銅板，97.1 英哩是多少公里...

   有一些資料轉換 (可以看成是數學函式) 可以推導出計算公式，例如英哩轉公里，華氏溫度轉攝氏，有些沒有辦法用公式的，就建立對照表格，這個題目由阿拉伯數字轉換爲中文數字並沒有特殊的計算公式，所以就需要直接建立對照表格

0	1	2	3	4	5	6	7	8	9
〇	一	二	三	四	五	六	七	八	九

   這樣子的對照表格一般來說在程式裡需要用陣列變數來存放

例如男鞋的尺吋對照表:

TW	81	82	83	84	85	86	87	88	89	90
USA	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11

可以用兩個陣列來表示，兩個陣列中位置相同的資料是一組的

```
int TW[10] = {81, 82, 83, 84, 85, 86, 87, 88, 89, 90};
double USA[10] = {6.5, 7, 7.5, 8. 8.5, 9, 9.5, 10, 10.5, 11};
```

如果使用者輸入一個 TW 單位的鞋子尺寸 twSize，希望用程式得到對應的 USA 大小，程式中需要先比對 twSize 和 TW 陣列的內容，看是表格中第幾個，然後 USA 陣列中對應的元素就是轉換的結果

```
int i, twSize, TW[10] = {81, 82, 83, 84, 85, 86, 87, 88, 89, 90};
double usaSize, USA[10] = {6.5, 7, 7.5, 8. 8.5, 9, 9.5, 10, 10.5, 11};
scanf("%d", &twSize);
for (i=0; i<10; i++)
   if (twSize == TW[i]) {
      printf("USA size = %.1f\n", USA[i]);
      break;
   }
```

請注意：如果輸入一個 USA 單位的鞋子尺寸 usaSize，比對的地方需要特別小心，因爲浮點數運算與表達時有相對的誤差，所以需要寫成

```
if ((usaSize > USA[i]-1e-6)&& (usaSize < USA[i]+1e-6)) {
…
```


另外因爲這個表格有很好的規律性，所以其實可以用公式來取代，例如：

```
usaSize = 6.5 + (twSize – 81) * 0.5;
twSize = 81 + (usaSize – 6.5) / 0.5;
```

不過沒有特別規律性的表格就比較難用公式來簡化了，例如：

Eure	39	40	41	41	42	43	44	44	45	45
USA	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11

這個表格可以部份用公式取代，例如：

```
eureSize = Eure[((usaSize - 6.5)/0.5) + 0.5];
```

四捨五入，避免除 0.5 之後得到像是 0.9999 的數字

如此就可以省去 USA 這個表格

接下來要看我們需要的表格如何定義：

0	1	2	3	4	5	6	7	8	9
〇	一	二	三	四	五	六	七	八	九

第一列有很好的規律性，所以並不需要定義，第二列的中文字比前面的整數或是浮點數陣列複雜一點，每一個中文字是一個字元陣列(或稱字串，一個中文字的編碼在 Big5 表格裡需要兩個位元組，在 UTF-8 最多四個位元組/常用字是三個位元組)

定義方法有兩種，變數型態不同，使用記憶體的方式並不相同，請挑選一種依照範例使用，詳細會在介紹陣列與字串處理時說明：

```
char *chineseNumeral[] = {"〇", "一", "二", "三", "四", "五", "六", "七", "八", "九"};
```

或是

```
char chineseNumeral[][3] = {"〇", "一", "二", "三", "四", "五", "六", "七", "八", "九"};
```

實際由使用者輸入個位數字，列印中文數字的範例如下：

```
int number;
scanf("%d", &number); // 0-9
printf("%s\n", chineseNumeral[number]);
```

如果你不喜歡用不太熟悉的語法也沒有關係，這個程式的轉換功能也可以用條件判斷式來完成：

```
int number;
scanf("%d", &number); // 0-9
if (number == 0)
   printf("〇");
else if (number == 1)
 printf("一");
…
else // if (number == 9)
 printf("九");
```

或是

```
int number;
scanf("%d", &number); // 0-9
switch (number) {
case 0:
   printf("〇"); break;
case 1:
   printf("一"); break;
…
case 9:
   printf("九");
}
```


2. 接下來需要仔細思考由鍵盤讀取連續數字的問題：

   a. 如果把輸入的數字當成十進位數字讀到整數變數裡面，

   ```
   scanf("%d", &x);
   ```

   那麼輸入數字就會有位數的限制，因爲 32 位元的有號整數變數最大可以記錄 2^{31}-1=2147483647 是一個九位數字，如果輸入的數字超過這個最大值，變數 x 裡面記錄的資料就不正確，當然轉換列印出來的結果也會有錯誤，如果使用 64 位元的有號整數變數，最大可以記錄2^{63}-1, 十進位19 位數 (9223372036854775807)

   b. 如果用整數變數的方式由鍵盤讀進來，一開始的 0 因爲不影響數值，scanf 會自動跳過，萬一使用者輸入 000123，程式裡會不曉得使用者在第一個非 0 數字之前到底輸入了幾個 0

   c. 一個整數變數 x, 在計算機內部的表示方法是二進位的，如果要以十進位制檢視其個位數、 十位數… 需要寫一小段程式轉換，這個轉換非常規律，所以我們不需要列表，而將轉換的規律用程式碼撰寫出來，教會 CPU 去自動轉換

接下來寫一段程式，把整數變數 x 裡面的資料以十進位來看，由最低爲開始一位數字一位數字寫到九個元素的整數陣列變數 digits 中，也就是說如果 x 的內容是整數 1234，我們希望 digits 陣列裡最後的資料是

4	3	2	1	?	?	?	?	?

通常要**設計**這樣的**演算法**，我們會先試著用紙筆，找一組簡單的資料，手動計算看看能不能找到一個規律的作法，以上面的資料爲例:

1234 除以 10, 商是 123, 餘數是 **4** (整數除法)
123 除以 10, 商是 12, 餘數是 **3**
12 除以 10, 商是 1, 餘數是 **2**
1 除以 10, 商是 **0**, 餘數是 **1**

可以看到這個過程裡每一個步驟的餘數就是我們要的答案，接下來要這樣的作法轉換成程式，可以看到

a. 需要有**迴圈**, 迴圈內執行
   "x 除以 10, 得到商和餘數,
   把商設成迴圈下一次執行時的 x, 把餘數記錄下來"
   迴圈的結束條件是 "商爲 0"


```
int x = 1234;
while (x>0) {
   printf("%d ", x % 10);
   x = x / 10;
}
```

b. 在迴圈執行過程中，記錄餘數：
   i. 迴圈執行幾次就有幾個餘數，因爲有多個，所以需要用陣列變數來記錄，前面提到陣列裡元素的個數是有上限的
   ii. 每一次該記錄在陣列的哪一個地方呢? 這就要看迴圈是執行到第幾次了，第一次執行就記錄在陣列的第 1 個元素，第二次記錄在第 2 個元素，以下類推如果是 for 迴圈就用迴圈控制變數，如果是 while 迴圈，就另外設計一個遞增的變數來記錄迴圈是執行到第幾次了

```
int x = 1234, ndigits = 0, digits[9];
while (x>0) {
   digits[ndigits++] = x % 10;
   x = x / 10;
}
```

```
int x = 1234, ndigits, digits[9];
for (ndigits=0; x>0; ndigits++) {
   digits[ndigits] = x % 10;
   x = x / 10;
}
```


3. 在設計程式的時候，隨著你思考方式的轉變、隨著你想像中資料的差異、或是爲了配合程式其它部份的需求，設計出來的演算法是可以有相當差異的，以上面這個 『找到整數變數 x 中存放資料的每一位數 (10 進位) 』的問題來說，如果要求結果要依序由高位排到低位，演算法的設計就有相當的差異。

問題: x = 1234，希望 digits 陣列內成爲

1	2	3	4	?	?	?	?	?

同樣地你需要用紙筆，手動計算看看能不能找到一個規律的作法

1234 除以 1000, 商是 **1**, 餘數是 234
234 除以 100, 商是 **2**, 餘數是 34
34 除以 10, 商是 **3**, 餘數是 4
4 除以 1, 商是 **4**, 餘數是 **0**

可以看到這個過程裡每一個步驟的商就是答案，換成程式

a. 需要有**迴圈**, 迴圈內執行
   "x 除以 10^k, 得到商和餘數,
   把餘數設成迴圈下一次執行時的 x, 把商記錄下來"
   迴圈的結束條件看起來是 "餘數爲 0", 但是再仔細想一下，如果 x = 1200 則第二次迴圈就停了，會少記錄兩個位數

還好如果提早結束了，剩下的位數其實都是 0，只是需要知道還有幾位數而已。另外這個計算的方法需要知道 k 的數值，就是 x 究竟是 10 進位幾位數，以這個例子來說 x 是 4 位數，k = 3，除數由 $10^k = 1000$ 開始，迴圈每執行一次後，除數除以 10，所以也可以用 『除數爲 0』來結束迴圈

b. 在迴圈執行過程中，記錄商：
   i. 迴圈執行幾次就有幾個商，因爲有多個，所以需要用陣列變數來記錄，前面提到陣列裡元素的個數是有上限的
   ii. 每一次該記錄在陣列的哪一個地方呢? 這就要看迴圈是執行到第幾次了，第一次執行就記錄在陣列的第 1 個元素，第二次記錄在第 2 個元素，以下類推如果是 for 迴圈就用迴圈控制變數，如果是 while 迴圈，就另外設計一個遞增的變數來記錄迴圈是執行到第幾次了

c. 計算初始的除數 (上例中就是 1000)，如右圖可以寫一個迴圈，把 x 值每次除 10，直到 0 爲止，除數則由 1 開始每次乘 10

```
int x=1234/10, divisor=1;
while (x>0) {
   divisor = divisor * 10;
   x = x / 10;
}
```

```
int x = 1234, ndigits = 0, digits[9];
int x1 = x / 10, divisor=1;
while (x1>0) {
    divisor = divisor * 10;
    x1 = x1 / 10;
}
while (divisor>0) {
    digits[ndigits++] = x / divisor;
    x = x % divisor;
    divisor = divisor / 10;
}
```

```
int x = 1234, ndigits, digits[9], x1, divisor;
for (x1=x/10,divisor=1; x1>0; x1/=10)
    divisor *= 10;
for (ndigits=0; divisor>0;
      ndigits++, x%=divisor, divisor/=10)
    digits[ndigits] = x / divisor;
```

4. 不論你是用步驟 2 的演算法還是步驟 3 的演算法，在算出每一位數之後，都需要結合步驟 1 列印中文的程式碼，如果是步驟 3 的話，可以不需要把每一位數放在陣列 digits 中，在上面迴圈中 x/divisor 算出每一位數時，直接列印該數字對應的中文即可

5. 前面幾個步驟已經可以完成所有功能，但是可以練習的東西還有很多，接下來請考慮另外一種選項 - 由鍵盤讀取連續數字，但是不要放在單一的整數變數裡，就是以字元的方式讀取鍵盤的輸入，只要輸入是連續的 0~9 的數字，就把對應的中文字印出去

   這個時候就沒有步驟 2 裡面所說的位數限制了，只要輸入的是連續的數字，就可以列印

   程式的主體就是一個單一迴圈，讀入數字，列印中文數字，直到讀到非數字爲止

   這裡需要注意的是

```
char c;
scanf(" %c", &c);
while ((c>='0')&&(c<='9')) {
    列印字元 c 對應的中文數字
    scanf("%c", &c);
}
```

   a. scanf 用 %c 讀到的是 '0' 到 '9' 的 ASCII 編碼
   b. ASCII 編碼中 '0', '1', …, '9' 是連續的，分別是 48, 49, …, 57
   c. 在 C 程式中寫 '1' 和寫 49 是完全一樣的整數值
   d. scanf(" %c", &c); 敘述中 % 前的空格是要求 scanf 跳過所有的空格, \t, 和 \n字元，讀取一個字元到變數 c 中

6. 步驟 5 你也可以嘗試
   a. 用 char digits[100]; scanf("%s", digits);
      一次把一串字元讀到陣列裡，再用迴圈檢查輸入字元是否爲數字，把一個一個 '0' 到 '9' 字元轉換爲中文數字
   b. 也可以用 c = getchar() 取代 scanf("%c", &c);
   c. 用 scanf("%[0-9]", digits); 取代　scanf("%s", digits);
      如此 scanf 在讀資料的時候就直接檢查是否是 '0' 到 '9' 的字元，如果不是的話, 不會讀任何字元進入 digits[], 串流指標停在那個不是 '0' 到 '9' 的字元上
   d. 用 char str[2]; scanf("%1[0-9] ", str); 取代
      char c; scanf("%c", &c);
      讀到的字元已經確認是 '0' 到 '9'，不需要另外檢查
7. 列印 Big5 編碼的中文字還有一種方式，例如印出中文數字三：
   char *cdigits = "〇一二三四五六七八九", c = '3';
   printf("%c%c", cdigits[2*(c-'0')],cdigits[2*(c-'0')+1]);

8. 請注意，設計迴圈的時候應該要嘗試把一個簡單的範例的資料變化狀況表列出來，例如右側程式中第一個迴圈設計之前，應該要嘗試列出

```
int x = 1234, ndigits = 0, digits[9];
int x1 = x / 10, divisor=1;
while (x1>0) {
    divisor = divisor * 10;
    x1 = x1 / 10;
}
while (divisor>0) {
    digits[ndigits++] = x / divisor;
    x = x % divisor;
    divisor = divisor / 10;
}
```

迴圈次數	初始	一	二	三
x1	123	12	1	**0**
divisor	1	10	100	1000

   如果程式測試的時候，發現迴圈執行完畢後 divisor 數值不是 1000，就可以在迴圈內加上一列 printf("divisor=%d x1=%d\n", divisor, x1); 來觀察迴圈執行時變數的內容，與上表比對就知道是不是如預期般運作，如果和上表的結果不太相同，就從開始不一樣的地方思考是哪一個敘述出的問題

   同樣地第二個迴圈設計之前應該要列出下列變數的變化情形: divisor, x, ndigits, digits[]，發生不如預期的狀況才可以迅速找出問題所在

延伸練習

阿拉伯	羅馬
1	I
5	V
10	X
50	L
100	C
500	D
1000	M

1. 如果題目改成輸入一個 1～3999 的整數，輸出對應的羅馬數字? (規則請查維基百科)
2. 這個題目的轉換就像是台語的白讀一樣，一個數字一個數字直接轉爲中文數字，可是我們其實知道有意義的中文數字讀法中間會夾雜 “億” “萬” “千” “百” “十” 這些數詞，能不能寫一個程式完成右列的轉換?

```
請輸入一個非負整數：2453
二千四百五十三
請輸入一個非負整數：2403
二千四百〇三
請輸入一個非負整數：2003
二千〇三
```

3. 題 2 中的〇一二三四五六七八九十百千萬應該可以很容易換成零壹貳參肆伍陸柒捌玖拾佰仟萬