

# 1101 NTOUCSE 程式設計 1C 期中考參考答案

110/11/16 (二)

考試時間：**13:20 – 16:00**

- 考試規則：
1. **請闔上課本**，**不可**參考任何文件包括小考、作業、實習、或是其它參考資料
  2. 你可以在題目卷上直接回答，可以使用鉛筆，但是**請在答案卷上註明題號**
  3. 你覺得有需要的話，可以使用沒有教過的語法，但是**僅限於 C/C++ 語言**
  4. 程式撰寫時請寫完整的程式碼，寫 。 。 。 的分數很低（一個程式裡有重複很多遍的敘述是**扣分的**）
  5. **不可**使用電腦、平板、智慧手機、手錶、及工程型計算機
  6. 請**不要**左顧右盼！請勿討論！請勿交換任何資料！對於題目有任何疑問請舉手發問
  7. 如果你提早交卷，請**迅速安靜地離開教室**，請勿在走廊喧嘩
  8. 違反上述考試規則視為不誠實的行為，由學校依學務規章處理
  9. 請在**題目卷及答案卷**上都寫下姓名及學號，交卷時請**繳交題目卷及答案卷**

1. 請指出並且更正下列程式片段中的語法錯誤或是語法陷阱造成的邏輯/功能錯誤

(a) [3] 下列程式希望由鍵盤輸入一 0~9 的字元，如何修改才能在 num 變數裡得到對應的數值

```
01 int num=-1, ch=getchar();
02 if ('9' >= ch >= '0') num = ch - '0';
```

**Sol:**

目前的寫法編譯器會說沒有任何語法錯誤，但是執行結果會發現判斷永遠失敗，因為 '9' >= ch >= '0' 邏輯運算式執行時，運算子優先順序以及結合律使得它 '9' >= ch >= '0' = ('9' >= ch) >= '0' = 1 >= 48 = 0，不論 ch 是 '0' ~ '9' 哪一個數字，運算結果永遠是 0，所以不會執行 num = ch - '0'，需要改成

```
if ('9' >= ch && ch >= '0') num = ch - '0';
```

(b) [3] 請問下列程式片段如何修改印出的數值才會是 0 和 55？

```
01 int x = 10, total;
02 while (x = 0)
03 {
04     total += x;
05     x-=1;
06 }
07 printf("x=%d, total=%d\n", x, total);
```

**Sol:**

第一變數 total 需要初始化為 0

第二迴圈的執行條件運算式 x=0 執行的數值是 0，因此迴圈主體不會是無窮迴圈，事實上根本不會執行到，當然也不會執行希望的 10 次，第 02 列需要改成 x!=0 或是 x>0

(c) [3] 請問下列程式片段為什麼不會列印結果，該怎麼修改？

```
01 sum=0, i=0;
02 while (i<=10)
03     sum += i;
04     i++;
05 printf("sum=%d\n", sum);
```

**Sol:**

這一段程式執行時會進入無窮迴圈，所以會執行不到第 05 列的敘述，應該要在第 03,04 列外面加上大括號

```
while (i<=10) {
    sum += i;
    i++;
}
```

或是把第 03,04 列改成 sum+=i++; 或是 sum+=i, i++;

(d) [3] 請問下列程式片段列印的結果會有什麼異常的現象？該怎麼修改？

```
01 int i=1, x=1;
02 while (i++<=16)
```

```

03     x*=i;
04 printf("%d\n", x);

```

**Sol:**

執行第 04 列的時候應該會看到負數或是不像 16 階層(16!)的數字，主要是因為 16 階層已經超過整數的表示範圍了，如果要能夠正確計算到 16 階層並且顯示的話，需要使用 long long 的資料型態，所以程式需要改成

```

long long int x=1;
...
printf("%lld\n", x);

```

2. 請完成下面判斷一個最長為 100 位數的 10 進位整數是否為 11 的倍數的函式，如果是 11 的倍數請回傳 1 否則請回傳 0

```

01 #include <stdio.h>
02 int isMultipleOf11(_____)
03 {
04     int sum=0, i;
05     for (i=0; _____; i++)
06         sum += (i%2*2-1) * (_____);
07     return _____==0;
08 }
09 int main()
10 {
11     char num[101];
12     while (1==scanf("_____", num))
13         printf("%d\n", isMultipleOf11(num));
14     return 0;
15 }

```

- (a) [4] 請完成第 02 列的字串陣列參數定義？

**Sol:**

char digits[101] 或是 char digits[] 或是 char \*const digits; 或是  
const char digits[101] 或是 const char digits[] 或是 const char \*const digits;  
陣列名稱可以自訂，只要和第 05,06,07 列一致即可

- (b) [4] 請問第 05 列迴圈結束條件該如何設定？

**Sol:**

digits[i]!=0 或是 digits[i]!='\0'，這是運用 scanf 讀入字串時所放入的結束字元來測試字串是否結束

- (c) [4] 請問第 06 列 (i%2\*2-1) 的目的為何？

**Sol:**

i%2 是 i 除以 2 的餘數，餘數是 1 代表 i 是奇數，餘數是 0 代表 i 是偶數

i%2\*2-1 = (i%2)\*2-1，所以當 i 是奇數時數值是 1，當 i 是偶數時數值是-1

所以這個計算可以根據 i 的數值得到增加或是減少的效果

- (d) [4] 請完成並說明第 06 列空格處字元轉為數字的轉換？

**Sol:**

digits[i]-'0'

- (e) [4] 請完成並說明第 07 列空格處檢查是否為 11 的倍數的敘述

**Sol:**

sum%11

- (f) [4] 請解釋這個演算法判斷所輸入數字為 11 的倍數的理由

**Sol:**

一個 n 位數的十進位數字  $a_{n-1}a_{n-2}...a_1a_0$  所代表的數值可以表示成  $a_{n-1}10^{n-1} + a_{n-2}10^{n-2} + ... + a_110^1 + a_0$ ，這個數值也可以寫成  $a_{n-1}(11-1)^{n-1} + a_{n-2}(11-1)^{n-2} + ... + a_1(11-1)^1 + a_0$ ，用二項式定理把  $(11-1)^x$  展開，可以得

到  $11^x - C_1 11^{x-1} 1^1 + C_2 11^{x-2} 1^2 - \dots + (-1)^x C_x 11^{x-x} 1^x$ ，這個數字除以 11 的餘數是  $(-1)^x$ ，所以我們得到

$$(a_{n-1}(11-1)^{n-1} + a_{n-2}(11-1)^{n-2} + \dots + a_1(11-1)^1 + a_0) \% 11 = (a_{n-1}(-1)^{n-1} + a_{n-2}(-1)^{n-2} + \dots + a_1(-1)^1 + a_0) \% 11$$

，只要  $a_0 - a_1 + a_2 - \dots + (-1)^{n-1} a_{n-1}$  是 11 的倍數，原本的數字就可以被 11 整除

(g) [4] 請填入第 12 列空格處 scanf 的格式轉換命令 (請填入保證它不會寫超過陣列大小的限制數值)

**Sol:**

%100s

3. 下面是一個計算整數 n 的二進位表示方法中最長的封閉 0 序列長度的程式片段，例如 n=529 的二進位表示為 1000010001 裡面有兩段封閉 0 序列，長度分別為 4 及 3，因此程式輸出 4。注意 n=20 的二進位表示為 10100，其中只有一段封閉 0 序列，長度為 1，n=15 的二進位表示為 1111，其中沒有封閉 0 序列，n=32 的二進位表示為 100000，也沒有封閉 0 序列

```
01 while (n%2==0 && n>0) n/=2;
02 maxLen = 0;
03 while (n>0) {
04     for (n/=2, len=0; n%2==0 && n>0; n/=2) len++;
05     if (_____) _____;
06 }
07 printf("%d\n", maxLen);
```

(a) [4] 請以 n=20 為例解釋第 01 列的 while 迴圈的條件 n%2==0 目的為何?

**Sol:**

n%2==0 相當於 (n/2)==0，也就是判斷 n 是不是偶數，如果 n=20 也就是它的二進位表示 10100 的最右邊一位數是 0，這個 0 不是任何封閉 0 序列的一部分，所以是不需要的，可以直接拿掉

(b) [4] 接題(a)，請問第 01 列條件 n>0 目的為何?

**Sol:**

如果 n 一開始就是 0，這個迴圈會永遠停不下來，所以加上 n>0 的測試可以確保 n 為 0 時迴圈會停止；如果 n 一開始並不是 0，不斷除以 2 的時候一定會使得 n 成為奇數，也就是 n%2==0 這個條件會不滿足而離開迴圈

(c) [4] 接題(b)，請問第 01 列 n/=2; 的作用為何?

**Sol:**

n/=2 就是要把題(a)中最右邊的 0 刪除掉，n/=2; 在迴圈中就會把連續不屬於任何封閉 0 序列的 0 都刪除掉，直到最右邊的一位數是 1 才會停止

(d) [4] 第 04 列希望用一個 for 迴圈計算一段封閉 0 序列的長度 len，請接續上題以 n=20 由第一列開始執行，說明此列初始化的地方 n/=2 的用意?

**Sol:**

接上一題的說明，執行到第 04 列的時候，n 經過第 03 列的測試，一定不是 0，而且二進位表示法中最右邊的一位數一定是 1，所以初始化的地方 n/=2 是把最右邊的 1 刪除掉，在滿足 n>0 的情況下，接下來的 for 迴圈除了把連續的 0 拿掉，這些 0 是封閉 0 序列的一部分，需要紀錄它的長度在 len 變數裡。這裡有一點可以提昇效率的，就是應該把連續的 1 都一起拿掉，不需要在 03~06 列的 while (n>0) 迴圈裡一次只拿掉一個 1，也就是說可以改寫成

```
while (n%2) n/=2;
for (len=0; n%2==0 && n>0; n/=2) len++;
```

(e) [4] 請完成第 05 列的 if 敘述以計算出最長封閉 0 序列的長度 maxLen

**Sol:**

```
if (len>maxLen) maxLen=len;
```

4. 請撰寫一個程式讀取右圖輸入串流中的測資，每一筆測資有兩列，每一列為一個數列，例如 5 1 2 5 9 17，其中第一個數

|              |
|--------------|
| 範例輸入         |
| 5 1 2 5 9 17 |
| 4 8 13 23 25 |
| 4 7 8 9 10   |
| 6 1 2 3 8 8  |

|                       |
|-----------------------|
| 範例輸出                  |
| 1 2 5 8 9 13 17 23 25 |
| 1 2 3 7 8 8 8 9 10 15 |

字 5 是數列的長度，接下來是由小到大排列的 5 個數字，你的程式需要將兩個數列合併列印，維持由小到大的順序。

(a) [4] 如果測資中每個數列最長有 100 個元素，請定義適當大小的兩個陣列

**Sol:**

```
int x[100], y[100];
```

(b) [4] 請撰寫讀取資料的迴圈

**Sol:**

```
int i, n, m;
scanf("%d", &n);
for (i=0; i<n; i++)
    scanf("%d", &x[i]);
scanf("%d", &m);
for (i=0; i<m; i++)
    scanf("%d", &y[i]);
```

(c) [8] 請撰寫一個迴圈以及條件判斷式來合併列印兩個數列

**Sol:**

```
int ix=0, iy=0;
while (ix<n||iy<m)
    if (iy==m||(ix<n&& x[ix]<=y[iy]))
        printf("%d", x[ix++]);
    else
        printf("%d", y[iy++]);
```

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 5 | 1 | 3 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |

5. 右圖中框框裡的數字代表一條森林路徑上各個位置上香菇的數量，假設採菇的人從 k=4 的位置開始採集，每走一格算是一步而且就把那個地方所有的香菇採集完畢，請寫一個程式計算問他在最多走 m=6 步的情況下可以採集最多的香菇數量？因為每次回頭都會有好幾步是沒有任何收穫的，所以仔細觀察一下會發現採集最多香菇的人最多只回頭一次，請說明並且完成下列程式：

```
01 #include <stdio.h>
02 int main()
03 {
04     int k, m, n, i, x, psum[1001]={0}, max, t, left, right;
05     while (3==scanf("%d%d%d", &k, &m, &n))
06     {
07         for (i=1; i<=n; i++)
08         {
09             scanf("%d", &x);
10             _____;
11         }
12         for (max=0, i=k-m<0?0:k-m; i<k; i++)
13         {
14             right = k;
15             if (m-(k-i)+i > right) right = m-(k-i)+i;
16             if (right>=n) right = n-1;
```

範例輸入

4 6 7  
2 3 7 5 1 3 9

範例輸出

25

```

17         if ((t=_____)>max) max = t;
18     }
19     for (i=_____; i>k; i--)
20     {
21         left = k;
22         if (_____ < left) left = _____;
23         if (left<0) left = 0;
24         if ((t=_____)>max) max = t;
25     }
26     printf("%d\n", max);
27 }
28 return 0;
29 }

```

- (a) [4] 採菇人由起始位置 4 走到 2 然後回頭走到 6，經過了 3,2,3,4,5,6 共 6 步，所採集的數量其實就是由 2 到 6 的所有香菇數量，我們不需要每次把陣列的內容都加一次，如果能夠由香菇數量的陣列做出一個前綴和(prefix sum)的陣列 psum，以這個例子來說，香菇數量是 2 3 7 5 1 3 9，前綴和的陣列  $\text{psum}[i] = \sum_{j=0}^i \text{mushrooms}[j]$  內容就是 2 5 12 17 18 21 30，那麼 2 到 6 所有的香菇數量就是  $\text{psum}[6] - \text{psum}[2-1] = 30 - 5 = 25$ ，請問第 10 列該如何計算出 psum 陣列裡的「前綴和」資料？

**Sol:**

$\text{psum}[i] = \text{psum}[i-1] + x$ ; 這個陣列的第  $i$  個其實是  $\sum_{j=0}^{i-1} \text{mushrooms}[j]$

- (b) [4] 第 12 列到第 18 列計算由  $k$  向左走到第  $i$  格然後回頭到達位置  $\text{right}$  時所採集數量的最大值，請解釋第 14 列、第 15 列、第 16 列的意義

**Sol:**

由第  $k$  格向左走到第  $i$  格，所以至少  $i-k$  格的香菇都採集起來了， $\text{right}$  紀錄的是最右邊的那一格，所以一開始第 14 列設在第  $k$  格，接下來第 15 列  $m-(k-i)+i$  是計算如果可以走  $m$  步，扣掉由  $k$  走到  $i$  的  $k-i$  步，可以由  $i$  開始往右走到第幾格，如果比先前紀錄在  $\text{right}$  裡面的  $k$  要大，就用  $m-(k-i)+i$  作為新的  $\text{right}$ ，第 16 列則是怕往右走已經到達第  $n-1$  格的盡頭了。這三列計算完畢以後，就可以確定在  $i \sim \text{right}$  之間所有的香菇都可以被採收起來。

- (c) [4] 請完成第 17 列採集香菇數量的計算

**Sol:**

此列計算採集的香菇數量( $t = \text{psum}[\text{right}+1] - \text{psum}[i]$ )，如果大於  $\text{max}$  就紀錄在  $\text{max}$  裡面

- (d) [4] 第 19 列到第 25 列計算由  $k$  向右走到第  $i$  格然後回頭到達位置  $\text{left}$  時所採集數量的最大值，請完成第 19 列迴圈的起始  $i$  值的計算

**Sol:**

$i = k + m \geq n ? n-1 : k+m$

- (e) [4] 請完成第 22 列的  $\text{left}$  的計算

**Sol:**

$\text{if } (i - (m - (i - k)) < \text{left}) \text{ left} = i - (m - (i - k));$

- (f) [4] 請完成第 24 列採集香菇數量的計算

**Sol:**

此列計算採集的香菇數量( $t = \text{psum}[i+1] - \text{psum}[\text{left}]$ )，如果大於 max 就紀錄在 max 裡面