

1121 NTOUCSE 程式設計 1C 期中考

姓名：_____ 系級：_____ 學號：_____

112/10/31 (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**請在答案卷上寫下題號**，並且註明在題目卷上
 3. 有需要的話，可以使用沒有教過的語法，但是**僅限於 C 語言**
 4. 程式撰寫時請寫完整的程式碼，寫... 的分數很低 (程式裡有重複很多遍的敘述本來就是**扣分**的)
 5. **不可**使用電腦、平板、電子紙、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. 請寫出下面程式片段執行的結果：

- (a) [5] 01 int x=3, y=6;
02 x += y-->5 && x%2==1;
03 printf("x=%d y=%d", x, y);
- (b) [5] 01 int x=3, y=5;
02 x += x/2>1 && (y*=2);
03 printf("x=%d y=%d", x, y);
- (c) [5] 01 int i, a[] = {1, 3, 5, 7, 2, 4, 6, 0};
02 for (i=2; i<5; i++)
03 printf("%d ", a[a[i]]);
- (d) [5] 01 int i, a[] = {1, 3, 5, 7, 2, 4, 6, 0};
02 int j, t, n = sizeof(a)/sizeof(int);
03 for (i=0; i<n-1; i++) {
04 for (j=n-1; j>i; j--)
05 if (a[j-1]<a[j])
06 t=a[j], a[j]=a[j-1], a[j-1]=t;
07 printf("%d ", a[i]);
08 }
- (e) [5] 01 int f(int a[], int i, int j) {
02 if (i>j)
03 return 0;
04 return f(a, i, j-2)*8 + a[j];
05 }
06
07 int main() {
08 int a[] = {1, 3, 5, 7, 2, 4, 6, 0};
09 printf("%d", f(a, 2, 7));
10 return 0;
11 }

2. 請閱讀下列程式並回答相關問題

```
01 #include <stdio.h>
02
03 int balance = 0;
04 void save(int amount) {
05     balance += amount;
06     printf("%d ", balance);
07 }
08
09 int main() {
```

```

10     int i, balance=0;
11     for (i=30; i<100; i+=30)
12         save(i);
13     printf("balance=%d\n", balance);
14     return 0;
15 }

```

- (a) 請問程式中哪個變數是全域變數? [2] 全域變數有什麼特性?[3]
- (b) [5] 請問這個程式會印出什麼數值?
- (c) [5] 第 13 列為什麼印不出第 6 列最後一次印出的數值? 該如何修改 main() 函式裡面的變數定義使得第 13 列印出第 6 列最後一次印出的數值?
- (d) [5] 請修改 (c) 的答案, 以函式參數及區域變數修改此程式以避免使用全域變數的語法?
- (e) [5] 請問此程式使用全域變數最大的好處為何?
- (f) [5] 請問為什麼職場中大家都不建議使用全域變數?
3. [15] 下列程式中二維整數陣列 `int points[10][2]`; 存放的是平面上多個點的 x 座標和 y 座標, 請完成下列程式運用 `stdlib` 中的 `qsort` 函式將 `points` 陣列中的資料先依照 x 座標由大到小排序, x 座標相等時再依照 y 座標由小至大排序 (在答案卷上請標示程式列號作答, 如果寫在題目卷上請在答案卷上註明)

```

01 #include <stdio.h>
02 #include _____
03
04 int compare(_____ a, _____ b) {
05     int *a1 = (int *)a, *b1=(int *)b;
06     return _____ ? _____ : _____;
07 }
08
09 int main() {
10     int points[][2] = {{40,17}, {10,19}, {25,35}, {40,21}, {40,15}, {20,11}, {25,16}};
11     int i, n=sizeof(points)/sizeof(int[2]);
12     qsort(_____, _____, _____, compare);
13     for (i=0; i<n; i++)
14         printf("%d,%d\n", points[i][0], points[i][1]);
15     return 0;
16 }

```

程式執行所列印的結果如下：

```

40, 15
40, 17
40, 21
25, 16
25, 35
20, 11
10, 19

```

4. [20] 一個以 b 進位表示的 d 位數整數 $N = (n_1 n_2 n_3 \dots n_d)_b, n_i \in \{0, 1, 2, \dots, b-1\}$, 如果滿足 $N = n_1^d + n_2^d + \dots + n_d^d$, 我們稱它為自戀數, 例如 10 進位整數 $(153)_{10} = 1^3 + 5^3 + 3^3$ 、3 進位整數

$(122)_3 = 1^3 + 2^3 + 2^3$ 、或是 5 進位整數 $(3134)_5 = 3^4 + 1^4 + 3^4 + 4^4 = 419$ 都是自戀數，10 進位整數 $(1321)_{10}$ 並不是自戀數因為 $1^4 + 3^4 + 2^4 + 1^4 = 99 \neq 1321$ 。請完成下列程式判斷一個數字是否為自戀數，程式的輸入有多列，每一列是一筆測資，包含一個整數 b 代表進位制 ($b \leq 10$)，以及一個 d 位元的數字 ($d < 100$)，是的話請輸出字串 YES，否則輸出字串 NO (在答案卷上請標示程式列號作答，如果寫在題目卷上請在答案卷上註明)

```

01 #include <stdio.h>
02
03 int main() {
04     int b, i, j, d, z, zd, sum, x;
05     char num[100];
06     while (2==scanf("%d%s", &b, num)) {
07         d=0; while (num[d]_____ 0) _____; // 計算有幾位數
08         for (x=sum=i=0; i<d; i++) {
09             z = num[i]-_____ ;
10             x = x*_____ + _____; // 計算 b 進位數字之數值 x
11             for (zd=z,j=1; _____; j++) zd *= z; // 計算出 zd 為每一位數的 d 次方
12             sum += _____; // 計算每一位數 d 次方的總和
13         }
14         printf(_____ ? "YES\n" : "NO\n");
15     }
16     return 0;
17 }

```

上面第 11 列是一個計算 z 的 d 次方的迴圈，需要執行 $d-1$ 次，效率不好，可以用下面執行 $\log d$ 次的迴圈取代

```

11         for (zd=1, dp=d; dp>0; dp/=2, z*=_____)
12             if (_____) zd *= z;

```

其中 z 的 d 次方仍然紀錄在變數 zd 中，迴圈執行的次數運用 dp 來控制，一開始數值設為想要計算的次方數 d ，迴圈每執行一次就會除以 2，以 $d=13, z=3$ 為例，目標是計算 $z^d = 3^{13} = 3^{2^0 \cdot 1 + 2^1 \cdot 0 + 2^2 \cdot 1 + 2^3 \cdot 1} = (3)^1 \cdot (3^2)^0 \cdot ((3^2)^2)^1 \cdot (((3^2)^2)^2)^1$ ，所以此迴圈就像是把次方數 13 轉為 2 進位，一位數一位數由最低位數開始處理，配合逐步計算出來的權重 $3, 3^2, (3^2)^2, ((3^2)^2)^2, \dots$ 來計算 3^{13}

5. (a) [5] 請解釋為甚麼下列程式列印的結果是 17000000.000000 而不是 17000001.000000？

```

01 float x=17000001;
02 printf("x=%f\n", x);

```

- (b) [5] 接上題，請問下列程式片段列印的結果是什麼？迴圈主體共執行了幾次？為什麼？

```

01 float x;
02 int y;
03 for (y=x=17000000; x<17000010; x+=1)
04     if (!y--) break;
05 printf("x=%.0f y=%d\n", x, y);

```