1071 NTOUCSE 程式設計 1C 期中考

姓名：_____ 系級：_____ 學號：_____ 107/11/06 (二)

考試時間：**13:20 – 16:00**

考試規則：1. **請闔上課本**，除了印給你的之外**不可**參考任何文件包括小考、作業、實習、或是其他參考資料
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 … 不予計分 (通常答案不會是需要重複寫很多遍的東西)
5. **不可**使用電腦、平板、智慧手機、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**寫下姓名及學號**，交卷時請繳交**題目卷**及**答案卷**

1. [5] 請問 C 程式中變數或是函式的命名原則是什麼?

Sol:

編譯器要求的是

a. 只能用大小寫英文字母、0~9 數字、以及底線(underscore)
b. 不可以有空格
c. 第一個字母不可以是數字
d. 不可以和保留字 (int, float, double, if, for, …) 衝突
e. 某些編譯器有長度的限制 (例如 32 個字元或是 256 個字元)

你自己的要求是

a. 要清楚地描述變數所存放資料的意義或是描述函式的功能
b. 可以用多個英文字來表示，例如 number_of_days 或是 numberOfDays

2. [5] 請問下列程式中 while 迴圈會執行多少次?

```
char i = 1;
while (i>0)
    printf("i=%d\n", i++);
```

Sol:

印出 i=1, i=2, …, i=127，所以迴圈總共執行 127 次

3. [5] 請問下列 C 函式，請問以 func(20)呼叫時回傳值為何?

```
int func(int n)
{
    if (n<=3)
        return n;
    else
        return func(n-1)/2 + func(n-2);
}
```

Sol:

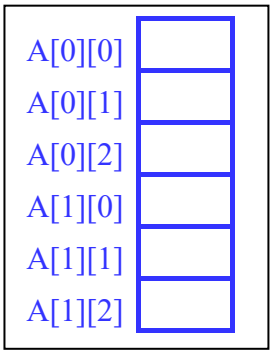
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
f(n)	1	2	3	3	4	5	6	8	10	13	16	21	26	34	43	55	70	90	115	**147**
f(n)/2	-	-	1	1	2	2	3	4	5	6	8	10	13	17	21	27	35	45	57	

4. [5] 數值方法常用二分逼近法來計算封閉區間 $[x_1, x_2]$ 中連續函數 $f(x)=0$ 的解，假設區間中有一個解且 $f(x_1)f(x_2)<0$，先計算區間的中心點 $x_3=(x_1+x_2)/2$，然後運用函數連續的特性，根據 $f(x_1)f(x_3)>0$ 與否判斷解在 $[x_1, x_3]$ 區間中還是 $[x_3, x_2]$ 區間中，不斷重複二等分的動作，就可以逐步逼近 $f(x)=0$ 的解，請問如果希望解的誤差小於等於 ε，需要重複切割的動作幾次 (請以 x_1, x_2, ε 的 log 或是 exp 或是多項式函數表示)？ _____

Sol:

$$\left\lceil \log_2 \frac{x_2 - x_1}{\epsilon} \right\rceil$$

5. [5] C 程式中概念上的二維整數陣列 int A[2][3]; 的 6 個整數存放在連續的記憶體中，請問 A[1][1] 這個整數變數是 6 個裡面的第幾個? 由於 C 語法中其實只有一維的陣列，所以在這個整數陣列的定義裡，它是以 3 個整數為單位(每個單位的型態是 int[3])，共有 2 個單位的陣列，請用這樣子的概念畫出這六個整數變數 A[0][0]到 A[1][2]在記憶體中的順序圖



Sol:

如右圖所示 A[1][1] 是 6 個整數變數裡的第 5 個，每一個整數變數佔 4 個位元組的記憶體

6. [5] 請完成下面程式由鍵盤讀入一個長度沒有限制的英文段落直到串流結束，將所有小寫字元轉換為大寫字元並且輸出在螢幕上: (注意請不要用神奇的數字)

```
#include <stdio.h>
int main()
{
    char ch;
    while ((ch=getchar()) != EOF)
    {
        if (ch >= 'a' && ch <= 'z' ) ch += 'A'-'a' ;
        putchar(ch);
    }
    return 0;
}
```

```
範例輸入
3
7
3 6 -1 4 6 5 3
4
0 0 0 0
14
-4 45 2 0 3 5 11 -7 854 25 3 -7 4 -3
```

Sol: 如上

7. [10] 請撰寫一個程式讀取右上圖資料，其中第一列是有多少筆資料，每一筆測資包含兩列輸入，第一列是資料個數 n，1<=n<=40，第二列依序有 n 個整數資料，每一筆測資有兩列輸出，第一列是 n 筆整數資料中最小值的位置，第二列是 n 筆整數資料中最大值的位置，請注意每一列輸出中的每一個位置之間有一個空格，以換列結束。

```
範例輸出
3
2 5
1 2 3 4
1 2 3 4
8 12
9
```

Sol:

```
#include <stdio.h>
int main()
{
    int nTests, iTest, n, i, j, a[40], max, min;
    scanf("%d", &nTests);
    for (iTest=0; iTest<nTests; iTest++)
    {
        scanf("%d", &n);
        for (i=0; i<n; i++)
```

```
            scanf("%d", &a[i]);
        max = min = 0;
        for (i=1; i<n; i++)
        {
            if (a[i]>a[max]) max = i;
            if (a[i]<a[min]) min = i;
        }
        for (j=i=0; i<n; i++)
            if (a[i]==a[min]) printf("%s%d", j++==0?"":" ", i+1);
        printf("\n");
        for (j=i=0; i<n; i++)
            if (a[i]==a[max]) printf("%s%d", j++==0?"":" ", i+1);
        printf("\n");
    }
    return 0;
}
```

8. n 階乘定義為 n!=1×2×…×n，這樣的數字非常大，如果你用整數 int 型態的變數來計算 n! 的話，只能計算到 12!= 479,001,600 (2^{31}-1=2,147,483,647)，如果你用 long long 型態的變數來計算的話，只能計算到 20!= 2,432,902,008,176,640,000 (2^{63}-1=9,223,372,036,854,775,801)，現在希望你寫一個程式計算 n! 這個數字用十進位表示時結尾有幾個 0，例如上面的 12! 的結尾有 2 個 0，20! 的結尾有 4 個 0，當然 n 不會只到 20 而已，n 的範圍是 1≤n≤2147483647，請根據下面的要求撰寫這個程式

   (a) [10] 請寫兩層迴圈計算 n!=1×2×…×n 等號右邊的數字裡可以分解出幾個 5 的因數，其中任何一個數字可以分解出幾個 5 請用迴圈執行除法(或是乘法)來計算 (例如 625，可以用迴圈除以 5 一直到商為 0 的時候結束)，例如 16!= 1×2×…×5×…×10×…×15×16=2^k×5^3×a，其中 k 不需要算出來，k 一定大於或等於 3，a 則不是 5 的倍數和表示法最後有幾個 0 無關，所以 16! 的十進位表示法最後有 3 個 0，另外外層的迴圈似乎只需要算 5, 10, 15, … 就夠了

Sol:

```
#include <stdio.h>
int main()
{
    int n, i, five, x;
    while (1==scanf("%d", &n))
    {
        for (five=0,i=5; i<=n; i+=5)
            for (x=5; i%x==0; x*=5) five++;
        printf("%d\n", five);
    }
    return 0;
}
```

   (b) [15] 題 (a) 是個暴力的作法，如果 n 值很大，或是要計算的資料筆數很多的時候就很花 CPU 時間了，請你仔細算一下 5^m! 這個數字有幾個 5 的因數：例如 5!裡面有 1 個 5，25! 這個數字分成 1~5, 6~10, 11~15, 16~20, 21~25 這五段每一段都有 1 個 5，再加上 25 是 5 的平方會多出 1 個 5 的因數，所以是 1*5+1=6 個 5 的因數，125! 這個數字分成 1~25, 25~50, 51~75,

76~100, 101~125 這五段每一段分別有 6 個 5，再加上 125 是 5 的三次方會多出一個 5 的因數，所以共有 6*5+1=31 個 5 的因數，接下來 625! 這個數字應該有 31*5+1=156 個 5 的因數，3125! 這個數字應該有 156*5+1 個 5 的因數...，如果數字 n 不能寫成 5^m，還是可以寫成 5 的次方組合，例如 n=334=2*5^3+3*5^2+1*5^1+4，所以 334! 這個數字裡會有 2*31 + 3*6 + 1*1 個 5，請你整理一下，用這個方法來計算 n! 裡有幾個 5 的因數，這個數字也就是 n! 用十進位表示時最後會有幾個 0 了，請先撰寫一個迴圈計算 5^m! 這樣的數字裡有多少個 5 的因數，把結果紀錄在一個陣列中，再撰寫一個迴圈完成上面的轉換。

Sol:

```
#include <stdio.h>
int main()
{
    int n, i, five, x, count[15]={0};
    for (i=1; i<15; i++) count[i]=count[i-1]*5+1;
    while (1==scanf("%d", &n))
    {
        for (five=i=0; n>0; i++,n/=5) five+=(n%5)*count[i];
        printf("%d\n", five);
    }
    return 0;
}
```

9. [20] Windows 小算盤的小數運算只能算出小數點後三十幾位，小明想知道兩個整數相除時更精準的小數值，請你幫他做出可以運算並且印出 N 位小數的程式： 請撰寫一個程式讀入三個整數 a, b 和 N，1<=N<=10000，1<=a,b<=2147483647，請輸出 a 除以 b 的運算結果，精確到小數點後第 N 位，第 N 位以後請無條件捨去。請注意因為需要印出的小數位數 N 可能很大，所以不能夠只用浮點數的除法和 printf 來處理，需要一位數、一位數來計算，如果不知道怎麼做，可以先試著用筆做做看長除法，範例輸入與輸出如下:

範例輸入	範例輸出
18467 41 20	450.41463414634146341463

Sol:

```
#include <stdio.h>
int main()
{
    int a, b, n, i;
    while (3==scanf("%d%d%d", &a, &b, &n))
    {
        printf("%d.", a/b);
        for (a%=b,i=0; i<n; i++,a=a*10%b)
            printf("%1d", a*10/b);
        printf("\n");
    }
    return 0;
}
```

範例輸入	範例輸出
2 3 4 1234 25000	0:left 1:right 1:right 0:right 1:right 0:right 2:left 3:right 5:left 6:left 7:right 0:right 1:left 3:left 5:right 6:right 7:left 8:right 9:right

10. [15] 有一個等臂天平，兩端各有一個秤盤可以同時放物品以及所有砝碼，現在有 10 個砝碼，第 i 個砝碼的質量是 3^i 單位，0<=i<=9，3^9 = 19683，質量 m 的物品固定放在左側秤盤，請寫

一個程式讀取一個整數 m，代表物品的質量，1<=m<=29524，輸出該把哪一個砝碼放在左右哪一個秤盤可以得到平衡 (提示: 先想看看 3 個砝碼如何秤出質量 1~26 的物品，程式很短只是需要仔細思考而已)

Sol:

```
#include <stdio.h>
int main()
{
    int i, m, offset;
    for (offset=1,i=0; i<10; i++) offset*=3;
    offset = (offset-1)/2;
    while (1==scanf("%d", &m))
    {
        for (m+=offset,i=0; m>0; i++,m/=3)
            if (m%3!=1)
                printf("%d:%s\n", i, m%3==2?"right":"left");
        printf("\n");
    }
    return 0;
}
```

	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
9	L	L	L	L	L	L	L	L	L	*	*	*	*	*	*	*	*	*	R	R	R	R	R	R	R	R	R
3	L	L	L	*	*	*	R	R	R	L	L	L	*	*	*	R	R	R	L	L	L	*	*	*	R	R	R
1	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R

想像物品放在左邊秤盤，如果每一個質量的砝碼都有的話，直接放在右邊秤盤就能夠平衡了

假設現在我們有三個砝碼 1,3,9，不過如上表我們可以自由地把砝碼放在左側或是右側來準備出 -13 到 13 的所有的質量刻度，上表中第一行 -13 的地方 L,L,L 代表 3 個砝碼都放在左邊秤盤，如此可以得到 -9-3-1＝-13 的刻度，*代表不放，R 代表放在右邊秤盤，檢查一下如此可以加總出所有-13 到 13 的質量，不過我們的物品質量一定是一個正數，該如何運用這張表格呢? 其實負數的部份只要在右邊多放一個更大的砝碼(第四個砝碼)例如 27 就可以變成 14~26 了，也就是如下表，不過三個砝碼是可以完整表示下圖粗框中 0~13 的

	14	15	16	17	18	19	20	21	22	23	24	25	26	0	1	2	3	4	5	6	7	8	9	10	11	12	13
27	R	R	R	R	R	R	R	R	R	R	R	R	R	*	*	*	*	*	*	*	*	*	*	*	*	*	*
9	L	L	L	L	L	L	L	L	L	*	*	*	*	*	*	*	*	*	R	R	R	R	R	R	R	R	R
3	L	L	L	*	*	*	R	R	R	L	L	L	*	*	*	R	R	R	L	L	L	*	*	*	R	R	R
1	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R	L	*	R

如果你把 R 換成 2，*換成 1，L 換成 0，這張表變成一個很漂亮的三進位制表格，你需要在讀入 m 的時候加上 13=(27-1)/2，mod 27，例如 m+13=12+13=**25** 得到下圖中咖啡色的數值，轉換為 3 進位數，221，就是代表 RR* 也就是 1:right, 2:right 的輸出

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	13	14	15	16	17	18	19	20	21	22	23	24	25	26
9	1	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	2	2	2	0	0	0	1	1	1	2	2	2
1	1	2	0	1	2	0	1	2	0	1	2	0	1	2