

1091 NTOUCSE 程式設計 1C 期末考參考答案

110/01/12 (二)

1. [4] 請定義一個包含兩個欄位的使用者自訂結構型態 Book 來存放一本書的資料，兩個欄位依序是字元陣列的書名(name)以及整數型態的價格(price)? [4] 請運用 typedef 語法定義兩個名稱 BOOK 以及 BOOKPTR 作為這種結構型態以及這種結構的指標型態的別名? [4]請定義一個 BOOK 型態的結構變數 book，並且定義一個 BOOKPTR 型態的指標變數 ptr? [4] 請呼叫 fgets() 以及 scanf()函式由鍵盤輸入一系列書名（例如“The Glass Hotel”）以及一個金額（例如 420），存放到結構變數 book 中?

Sol:

```
struct Book {
    char name[100];
    int price;
};
typedef struct Book BOOK, *BOOKPTR;
BOOK book;
BOOKPTR ptr;
fgets(book.name, 100, stdin);
len = 0; while (book.name[len]!='\n') len++; book.name[len]=0;
scanf("%d", &book.price);
```

2. [5] 請撰寫一個 calculateTotal()函式，函式有三個參數包括一個 BOOK 型態的購物車 shoppingCart 陣列，一個整數 numBooks，一個折扣數 discount (例如 0.8 代表八折)，計算並且回傳購物車中所有書籍折扣後四捨五入到整數的總價格?

Sol:

```
int calculateTotal(BOOK shoppingCart[], int numBooks, double discount) {
    int i, sum=0;
    for (i=0; i<numBooks; i++)
        sum += shoppingCart[i].price;
    return sum*discount+0.5;
}
```

3. [8]由於前面兩題中書名的陣列固定為 100 個字元，如果書名較短的話會浪費記憶體空間，如果書名超過這個長度就無法存放，請重新定義包含兩個欄位的結構型態 NewBook，欄位依序是字元指標的書名(name)以及整數型態的價格(price)? 請運用這種資料型態定義結構變數 newbook，並且由題 1 中的結構變數 book 裡拷貝資料到 newbook 中? 注意需要配置正確數量的記憶體才能存放資料，同時請使用 string.h 裡面的函式例如 strlen, strcpy 來處理字串資料。

Sol:

```
#include <malloc.h>
#include <string.h>
struct NewBook {
    char *name;
    int price;
};
struct NewBook newbook;
newbook.name = (char *) malloc(strlen(book.name)+1);
```

```
strcpy(newbook.name, book.name);
newbook.price = book.price;
...
free(newbook.name); // 不再使用時需要釋放記憶體
```

4. [15] 請問下列 C 程式有什麼語法上的錯誤？有什麼執行時的錯誤？該如何修改？

Sol:

C 語法上有兩個錯誤，第一是結構的初始化不能寫在類別定義裡面，唯一的例外是靜態常數欄位，第二是結構定義的語法裡，大括號後面需要有分號，執行的時候也會發生錯誤，主要是 `struct site *ptr;` 敘述定義了一個指標變數之後，並沒有配置結構所需要的記憶體，會導致執行的時候錯誤結束。應該改成

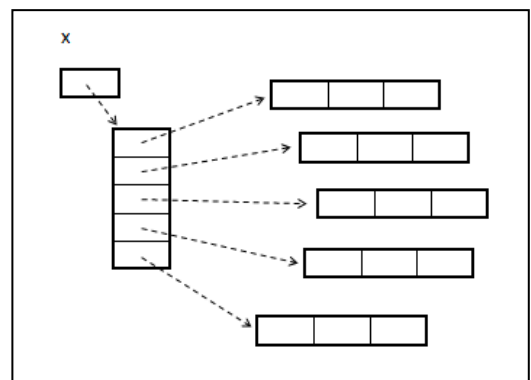
```
#include <stdio.h>
struct site {
    int zipcode = 202;
    char address[] = "No. 2, Bei-Ning Rd.";
}
int main() {
    struct site *ptr;
    printf("%d ", ptr->zipcode);
    printf("%s\n", ptr->address);
    return 0;
}
```

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
struct site {
    int zipcode;
    char address[100];
};
int main() {
    struct site *ptr = (struct site *) malloc(sizeof(struct site));
    ptr->zipcode = 202;
    strcpy(ptr->address, "No. 2, Bei-Ning Rd.");
    printf("%d ", ptr->zipcode);
    printf("%s\n", ptr->address);
    free(ptr);
    return 0;
}
```

5. [10] 請根據右圖動態配置/釋放一個概念上 5×3 的二維整數陣列 `x`? [10] 請以陣列語法將所有元素 `x[i][j]` 設定為 $3*i+j$ ，請撰寫一個函式 `multiply7()` 將這個陣列傳遞進函式，讓函式裡可以讀取/寫入這個陣列，將每一個元素乘上 7?

Sol:

```
#include <malloc.h>
void multiply7(int **x) { // or void multiply7(int *x[])
    int i, j;
    for (i=0; i<5; i++)
        for (j=0; j<3; j++)
            x[i][j] *= 7;
}
int main() {
    int i, j;
    int **x;
```



```

x = (int **) malloc(sizeof(int *)*5);
for (i=0; i<5; i++)
    x[i] = (int *) malloc(sizeof(int)*3);
for (i=0; i<5; i++)
    for (j=0; j<3; j++)
        x[i][j] = 3*i+j;
multiply7(x);
for (i=0; i<5; i++)
    free(x[i]);
free(x);
return 0;
}

```

6. 有一個存放 N 個整數的陣列 A ，每一個數字代表一支棍子的長度，請寫一個程式計算這 N 支棍子可以做出多少個不同的三角形？

(a) [5] 請以三層迴圈撰寫一個使用 $O(N^3)$ 時間的函式 `int countTriangles(int A[], int N)`

Sol:

```

int countTriangles(int A[], int N)
{
    int count=0, i, j, k;
    for (i=0; i<N-2; i++)
        for (j=i+1; j<N-1; j++)
            for (k=j+1; k<N; k++)
                if ((A[i]+A[j]>A[k])&&(A[i]+A[k]>A[j])&&(A[j]+A[k]>A[i]))
                    count++;
    return count;
}

```

(b) [5] 請先以 `qsort` 排序後以三層迴圈撰寫一個使用 $O(N^3)$ 時間，但是略有效率的函式 `int countTriangles(int A[], int N)`

Sol:

```

int comp(const void *a, const void *b) { return *(int*)a-*(int*)b; }
int countTriangles(int A[], int n) {
    int count=0, i, j, k;
    qsort(A, N, sizeof(int), comp);
    for (i=0; i<N-2; i++)
        for (j=i+1; j<N-1; j++)
            for (k=j+1; k<N&&A[i]+A[j]>A[k]; k++)
                count++;
    return count;
}

```

(c) [5] 請修改題(b)的函式以兩層迴圈撰寫一個使用 $O(N^2)$ 或是 $\Theta(N^2)$ 時間的函式 `int countTriangles(int A[], int N)`

Sol:

```

O(N2)
int comp(const void *a, const void *b) { return *(int*)a-*(int*)b; }
int countTriangles(int A[], int N) {
    int count=0, i, j, k;
    qsort(A, N, sizeof(int), comp);
    for (count=i=0; i<N-2; i++)
        for (j=i+1, k=i+2; j<N-1; j++)
            while (k<=N-1&&A[i]+A[j]>A[k])
                count += k-j, k++;
}

```

```

    return count;
}
}
Θ(N2)
int comp(const void *a, const void *b) { return *(int*)a-*(int*)b; }
int countTriangles(int A[], int N) {
    int count=0, i, j, k;
    qsort(A, N, sizeof(int), comp);
    for (count=0, k=n-1; k>=2; k--)
        for (i=0, j=k-1; i<j;)
            if (a[i]+a[j]>a[k])
                count += j-i, j--;
            else
                i++;
    return count;
}

```

7. 請寫一個程式，讀入一個 n 個元素 ($1 < n < 1000$) 的整數陣列 x ，在所有 k 個數字 ($1 < k < 100$) 的子集合中，找出和為偶數且和為最大的子集合。例如 x 的內容依序為 4, 9, 8, 2, 6，指定的 k 為 3，則 8, 6, 4 三個數字的總和 18 為最大，這個問題雖然可以暴力地把 n 取 k 個組合做出來，然後在所有和為偶數的組合中找最大值，但是這樣子的運算量太高，因為 k 個元素的和要是偶數，所以如果 k 個元素中有奇數的話，一定是偶數個，要求和為最大，各個元素也要相當大，除非像是 11, 8, 4, 1 這四個數字，11 很大，但是只能配上最小的 1，所以因為 $11+1 < 8+4$ ，如果只挑兩個數字的話，只能選擇 8, 4，放棄掉 11，下面希望得到一個運算量在 $n \log n$ 以下的程式，請根據下列提示完成程式：

(a) [2] 首先程式如右圖讀入指定的 n 、 k 、以及陣列 x ，然後由大到小排列陣列裡的所有資料，請在空格內填入適當敘述？

Sol:

```
*(int*)b-*(int*)a;
```

(b) [4] 由於要求 k 個數字的和是偶數，所以並不是直接挑 x 裡面最大的 k 個，需要把 k 個奇數和 k 個偶數分離出來，如右圖由排序好的 x 陣列中，挑出最大的 k 個偶數放在陣列 $y[0]$ 中，最大的 k 個奇數依序放在陣列 $y[1]$ 中，因為在 x 中偶數或是奇數有可能不足 k 個，所以用另外陣列 ndx 來紀錄偶數的個數 $ndx[0]$ 和奇數的個數 $ndx[1]$ ，請完成右圖程式片段

Sol:

```
x[i]%2
ndx[odd]
```

```

#include <stdio.h>
#include <stdlib.h>
int comp(const void *a, const void *b)
{ return _____; }
int main() {
    int i, n, k, x[1000];
    while (2==scanf("%d%d", &n, &k)) {
        for (i=0; i<n; i++)
            scanf("%d", &x[i]);
        qsort(x, n, sizeof(int), comp);

        ...

        printf("%d\n", sum);
    }
    return 0;
}

```

```

int y[2][100], ndx[2]={0}, odd;
for (i=0; i<n && (ndx[0]<k||ndx[1]<k); i++)
{
    odd = _____;
    y[odd][_____] = x[i];
    ndx[odd]++;
}

```

```

if (k%2==1) {
    if (ndx[0]==0) {
        printf("-1\n");
        continue;
    }
    sum = y[0][0];
}
else
    sum = 0;

y[0][ndx[0]] = y[1][ndx[1]] = 0;
for (i=k%2, j=0; i<ndx[0]; i+=2, j++)
    y[0][j] = y[0][i] + y[0][i+1];
for (i=j=0; i<ndx[1]; i+=2, j++)
    y[1][j] = y[1][i] + y[1][i+1];

```

(c) [6] 由於當 k 是奇數時，如果 x 裡面沒有偶數，不可能得到 k 個整數的和是偶數，請檢查並且列印 -1 ，如果 x 裡面有偶數，則最大的偶數一定在指定的集合裡，所以將 x 裡面最大的偶數設為 sum 接下來只要找到 $k-1$ 個數字就可以；如果 k 是偶數，則設為 0 ；接下來將相鄰的兩個偶數相加，相鄰的兩個奇數相加還是放在陣列 y 中，如下圖程式，請撰寫一個迴圈，由 $y[0]$ 和 $y[1]$ 兩個陣列中把最大的 $k/2$ 個數字加總到 sum 變數中然後印出答案

Sol:

```
int ptr[2]={0};
for (ptr[0]=ptr[1]=0,i=k%2; i<k; i+=2)
{
    if (ptr[0]>=(ndx[0]-k%2)/2)
        odd = 1;
    else if (ptr[1]>=ndx[1]/2)
        odd = 0;
    else
        odd = y[1][ptr[1]] > y[0][ptr[0]];
    sum += y[odd][ptr[odd]++];
}

printf("%d\n", sum);
```

8. [10] 請撰寫一個遞迴函式 `void stringPrint(int len, const char *str, char data[], int pivot)`，當執行 `stringPrint(3, "art", data, 0)` 時依序列印出右圖中的所有 27 個字串，請問不包含前面這個最上層的呼叫，`stringPrint()` 函式被呼叫的次數是多少？如果你的遞迴函式被呼叫到的次數超過 27 次，請修改一下讓它小於等於 27 次

Sol:

```
void stringPrint(int len, const char *str, char data[], int pivot) {
    if (pivot>=len)
        printf("%s\n", data);
    else
        for (int i=len-1; i>=0; i--) {
            data[pivot]=str[i];
            stringPrint(len, str, data, pivot+1);
        }
}
```

39 次

```
void stringPrint(int len, const char *str, char data[], int pivot) {
    for (int i=len-1; i>=0; i--) {
        data[pivot]=str[i];
        if (pivot<len-1)
            stringPrint(len, str, data, pivot+1);
        else
            printf("%s\n", data);
    }
}
```

12 次

```
ttt
ttr
tta
trt
trr
tra
tat
tar
taa
rtt
rtr
rta
rrt
rrr
rra
rat
rar
raa
att
atr
ata
art
arr
ara
aat
aar
aaa
```

