

1081 NTOUCSE 程式設計 1C 期末考參考答案

109/01/07 (二)

1. [5] 請定義一個包含四個整數欄位的使用者自訂結構 Time 來存放時間值，四個欄位分別是日(day)、時(hour)、分(minute)、秒(second)?

Sol:

```
struct Time {  
    int day, hour, minute, second;  
};
```

2. [10] 接上題，請定義一個函式 add，傳入兩個 Time 結構的資料，計算並且回傳兩個時間值的和？(請注意和的時間值需要是合法的)

Sol:

```
struct Time add(struct Time x, struct Time y) {  
    struct Time sum = x;  
    sum.second += y.second;  
    sum.minute += y.minute + sum.second/60;  
    sum.second %= 60;  
    sum.hour += y.hour + sum.minute/60;  
    sum.minute %= 60;  
    sum.day += y.day + sum.hour/24;  
    sum.hour %= 24;  
    return sum;  
}
```

3. [10] 請動態配置 100 個 Time 結構的陣列，請檢查配置是否成功，請寫一個迴圈由鍵盤讀取 100 列的日、時、分、秒，另外請呼叫上述 add 函式計算並且列印所有時間值的和，完畢請釋放記憶體

Sol:

```
int i;  
struct Time sum, *x = (struct Time *) malloc(100*sizeof(struct Time));  
if (x==NULL) {  
    printf("Error in allocating memory!\n"); return 1;  
}  
for (i=0; i<100; i++)  
    scanf("%d%d%d%d", &x[i].day, &x[i].hour, &x[i].minute, &x[i].second);  
for (sum=x[0], i=1; i<100; i++) sum = add(sum, x[i]);  
printf("%d 日 %d 時 %d 分 %d 秒\n", sum.day, sum.hour, sum.minute, sum.second);  
free(x);
```

4. [10] 請完成右側程式，函式 allocate 配置指標陣列，並配置一個 10 列的上三角形的二維陣列(第一列有 10 個整數、第二列有 9 個整數、...、第十列有 1 個整數)，內容為呼叫 rand() 得到的亂數值，函式 average 計算所有元素的平均值，函式 deallocate 釋放掉所有的配置的記憶體。

Sol:

```
#include <stdio.h>  
#include <stdlib.h> // malloc and rand  
int **allocate() {  
    int i, j;  
    int **x = (int **) malloc(10*sizeof(int*));  
    for (i=0; i<10; i++) {  
        x[i] = (int *) malloc((10-i)*sizeof(int));  
        for (j=0; j<10-i; j++)  
            x[i][j] = rand();  
    }  
    return x;  
}  
double average(int **x) {  
    int i, j; double sum=0;  
    for (i=0; i<10; i++)  
        for (j=0; j<10-i; j++)  
            sum += x[i][j];  
    return sum/55.0;  
}  
void deallocate(int **x) {  
    int i;  
    for (i=0; i<10; i++)  
        free(x[i]);  
    free(x);  
}  
int main() {  
    int **x = allocate();  
    printf("%f\n", average(x));  
    deallocate(x);  
    return 0;  
}
```

5. [5] 請問右列空格中填入何者為錯誤的?

- A) int y[10][5]
- B) int (*y)[5]
- C) int **y;
- D) int (*const y)[5]
- E) int y[][5]

Sol:
C

```
void process(_____)
{
    ...
}
int main()
{
    int x[10][5];
    process(x);
    return 0;
}
```

6. [10] 下圖為一正確運作的快速排序函式 quicksort，快速排序法的效率和資料原本的順序密切相關，運氣好的話，每一次分割時資料以 pivot 為中心差不多分成兩半，這個時候排序的效

<pre>01 #include <stdio.h> 02 #include <stdlib.h> 03 04 void swap(int a[], int i, int j) { 05 int tmp=a[i]; 06 a[i]=a[j]; 07 a[j]=tmp; 08 } 09 10 int partition(int a[], int start, int end) { 11 int tmp, front = start+1, rear = end; 12 while (rear >= front) 13 { 14 while (front<=end && a[front]<a[start]) front++; 15 while (rear>=start+1 && a[rear]>=a[start]) rear--; 16 if ((rear>=start+1)&&(front<=end)&&(rear>front)) 17 swap(a, front++, rear--); 18 } 19 return rear; 20 } 21</pre>	<pre>22 void quicksort(int a[], int start, int end) { 23 if (end-start <= 0) return; 24 swap(a, start, start+rand()%(end-start+1)); 25 int pivot = partition(a, start, end); 26 swap(a, start, pivot); 27 quicksort(a, start, pivot-1); 28 quicksort(a, pivot+1, end); 29 } 30</pre>
	<pre>31 int main() { 32 int i, n, seed, a[30001]; 33 scanf("%d%d", &n, &seed); 34 srand(seed); 35 for (i=0; i<n; ++i) 36 scanf("%d", &a[i]); 37 quicksort(a, 0, n-1); 38 for (i=0; i<n; ++i) 39 printf("%d ", a[i]); 40 return 0; 41 }</pre>

率最好，如果原本是 n 筆資料 ($n \leq 30000$)，差不多需要分割 $\log n$ 次，每一次分割資料 partition 函式需要看過所有的資料一次，所以這個演算法平均需要 $n \log n$ 的時間。為了能夠主動把資料打散，得到平均的效能，所以在上面第 24 列加上一個隨機更換 pivot 資料的敘述，當然，運氣很不好的時候，資料的原始順序還是有可能讓這個演算法的效率變得很差，請完成下列程式，產生讓上面這個程式效率表現最不好的資料，輸入是一個整數 n 以及 seed。(請注意表現最不好的資料在每次分割時，pivot 不是剛好最大就是剛好最小，請讓下面程式每次 pivot 剛好都是最小的資料，資料包括 0,1,2,...,n-1 這 n 個數字，這個程式可以用一個陣列 seq 來完成，也可以用兩個陣列來完成)

Sol:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i, n, seed, seq[30001], p;
    scanf("%d%d", &n, &seed);
    srand(seed);

    for (i=0; i<n; i++) seq[i] = -1;
    for (i=n; i>=1; i--)
    {
        p = rand()%i + (n-i);
        while (seq[p]>=0) p = seq[p];
        seq[p] = n-i;
    }

    for (i=0; i<n-1; i++)
        printf("%d ", seq[i]);
    printf("%d\n", seq[n-1]);
    return 0;
}
```

```

#include <stdio.h>
#include <stdlib.h>

int main() {
    int i, n, seed, seq[30001], rev[30001], j, tmp;
    scanf("%d%d", &n, &seed);
    srand(seed);

    for (i=0; i<n; i++) rev[i] = i;
    for (i=0; i<n; i++)
        tmp=rev[i], rev[i]=rev[j=i+rand()%(n-i)], rev[j]=tmp;
    for (i=0; i<n; i++) seq[rev[i]] = i;

    for (i=0; i<n-1; i++)
        printf("%d ", seq[i]);
    printf("%d\n", seq[n-1]);
    return 0;
}

```

7. [5] 請問下圖中程式的輸出是什麼？

Sol:

```

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

```

```

01 #include <stdio.h>
02 void mystery(int data[], int istart, int iend) {
03     if (istart>iend) {
04         for (int i=0; i<=iend; i++)
05             printf("%d ", data[i]);
06         printf("\n");
07     }
08     else {
09         data[istart] = 1-data[istart];
10         mystery(data, istart+1, iend);
11         data[istart] = 1-data[istart];
12         mystery(data, istart+1, iend);
13     }
14 }

```

```

15 int main() {
16     int data[3]={1, 1, 1};
17     mystery(data, 0, 2);
18     return 0;
19 }

```

(b)

```

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0

```

(c)

```

0 0 0
0 0 1
0 0 2
0 1 2
0 1 1
0 1 0
0 2 0
0 2 1
0 2 2
1 2 2
1 2 1
1 2 0
1 1 0
1 1 1
1 1 2
1 0 2
1 0 1
1 0 0
2 0 0
2 0 1
2 0 2
2 1 2
2 1 1
2 1 0
2 2 0
2 2 1
2 2 2

```

[5] 如果希望輸出如圖(b)，請問需要修改哪些地方？

Sol:

1. 第 16 列改成 `int data[3]={0};`;
2. 刪除第 9 列

[10] 請問如果希望輸出如圖(c)，程式 9~12 列需要怎樣修改？

Sol: (這一題因為考卷上的輸出印錯了，最後多了一些東西，所以大家都有加 10 分)

將第 9,10,11,12 列換成

或是

```

if (data[istart]==2) {
    for (;data[istart]>=0; data[istart]--)
        mystery(data, istart+1, iend);
    data[istart] = 0;
}
else { // if (data[istart]==0)
    for (;data[istart]<=2; data[istart]++)
        mystery(data, istart+1, iend);
    data[istart] = 2;
}

```

```

int diff, end;
diff = data[istart]==2 ? -1 : 1;
end = data[istart]==2 ? -1 : 3;
while (data[istart]!=end) {
    mystery(data, istart+1, iend);
    data[istart] += diff;
}
data[istart] -= diff;

```

8. [15] 下面程式中有兩個整數陣列 arr1 和 arr2，由鍵盤輸入如下性質的資料，arr2 裡面所有的資料都不相等，每一個數字都出現在 arr1 陣列中，請完成下面程式，運用 qsort 把 arr1 裡面的資料依照 arr2 裡面資料的順序排好，如果有沒有出現在 arr2 裡面的資料，請由小到大排在最後，兩個陣列的大小都不大於 1000，陣列中所有資料都大於等於 0、小於等於 1000?

範例輸入

11
2,3,1,3,2,4,6,19,9,2,7
6
2,1,4,3,9,6

範例輸出

2 2 2 1 4 3 3 9 6 7 19

```
#include <stdio.h>
#include <stdlib.h>

int comp(const void *a, const void *b) {
    return *(int*)a-*(int*)b;
}

int main() {
    int i, len1, arr1[1000], len2, arr2[1000];
    int ri[1001];

    scanf("%d", &len1);
    for (i=0; i<len1; i++)
        scanf("%d,", &arr1[i]);

    scanf("%d", &len2);
    for (i=0; i<len2; i++)
        scanf("%d,", &arr2[i]);

    for (i=0; i<=1000; i++) ri[i] = 2000+i;
    for (i=0; i<len2; i++)
        ri[arr2[i]] = i;

    for (i=0; i<len1; i++)
        arr1[i] = ri[arr1[i]];

    qsort(arr1, len1, sizeof(int), comp);

    for (i=0; i<len1; i++)
        arr1[i] = arr1[i]<=1000 ? arr2[arr1[i]] : arr1[i]-2000;

    for (i=0; i<len1-1; i++)
        printf("%d ", arr1[i]);
    printf("%d\n", arr1[len1-1]);

    return 0;
}
```

9. [15] 實習時用迴圈與函式寫過一個「旋轉法」產生所有排列的程式，課堂裡我們也有用建構的方式解釋為什麼這個方法能夠產生所有的排列，產生的排列的順序如下圖，由這個建構的方式

0123	2013	1203	1023	2103	0213
3012	3201	3120	3102	3210	3021
2301	1320	0312	2310	0321	1302
1230	0132	2031	0231	1032	2130

也可以很快地完成一個遞迴的程式來產生上面的排列，請完成下列程式?

```
void permuteFromRotation(int n, char items[], char pdata[], int len)
{
    int i, j, tmp;
    if (len>=n) {
        pdata[n] = 0;
        printf("%s\n", pdata);
        return;
    }
    pdata[len] = items[len];
    for (i=0; i<=len; i++) {
        permuteFromRotation(n, items, pdata, len+1);
        for (tmp=pdata[len],j=len; j>0; j--)
            pdata[j] = pdata[j-1];
        pdata[0] = tmp;
    }
}
```

輸入資料：

0123
abcde

```
int main() {
    char result[20], items[20];
    int n;
    while (1==scanf("%s", items)) {
        n=0; while (items[n]!=0) n++;
        result[0] = items[0];
        permuteFromRotation(n, items, result, 1);
        printf("\n");
    }
    return 0;
}
```