

# 1101 NTOUCSE 程式設計 1C 期末考參考答案

111/01/18 (二)

1. 請依照下列要求完成程式，讀入 a, b 兩個正整數，計算 a 到 b 之間(含端點)所有奇數的和

(a) [5] 請完成下列 for 迴圈及條件判斷敘述

```
int sum, i, a, b;  
scanf("%d%d", &a, &b);  
for (sum=0, i=a; i<=b; i++)  
    if (i%2) sum += i;      或是 i%2==1
```

(b) [3] 題 (a) 的 if 敘述可以改為 sum += i%2 ? i : 0 ;

(c) [3] 請以 for 迴圈的起始條件控制 (不要使用條件判斷敘述)

```
for (sum=0, i=a+1-a%2 ; i<=b ; i+=2 )  
    sum += i;
```

(d) [3] 請以整數的算術運算來挑選奇數進行加總的運算

```
for (sum=0, i=1; i<=b; i++)  
    sum += i * (i%2) ;
```

(e) [5] 請推導梯形公式將 [a,b] 範圍內所有的奇數加總，並且填入下列空格

sum = (a-a%2+b+b%2) \* (b+b%2-a+a%2) / 4 ;

(f) [6] 請在空格內完成下列遞迴函式的設計將 [a,b] 範圍內所有的奇數加總

```
int sum(int a, int b) {  
    return a>b ? 0 : a+ sum(a+2, b) ;  
}  
int main() {  
    ...定義變數並且讀入任意兩個正整數 a, b ...  
    printf("%d\n", sum(a+1-a%2, b)); // hint:大於等於 a 的第一個奇數  
    ...  
}
```

2. 請撰寫遞迴程式找到具有下列特性的九位數整數

最高位被 1 整除

前兩位數被 2 整除

前三位數被 3 整除

...

全部九位數被 9 整除

(a) [10] 如果每一位數是{1, 2,3,4,5,6,7,8,9}集合中**任意一個**數字，請完成下列函式

下列函式參數 num 是一個滿足前 ith-1 個條件的 ith-1 位數字，此函式嘗試找到第 ith 位數字一直到第 9 位數，滿足第 ith, ith+1, ..., 一直到第 9 個條件

```

#include <stdio.h>
int find(int num, int ith) {
    int i, c;
    if ( ith==10 ) { // 遞迴的結束條件
        printf("%d\n", num);
        return 1;
    }
    else
        for (i= 1; i< 10; i++) // 嘗試 1~9 所有的數字作為第 ith 位數
            if ((c=num*10+i)% ith ==0)
                if (find(c, ith+1)==1)
                    return 1;

    return 0;
}
int main() {
    find(0,1);
    return 0;
}

```

- (b) [10] 題(a)的程式會找到第一個滿足條件的 9 位數數字，請修改程式讓它把所有滿足條件的數字印出來，計算並且印出這樣的數字總共有幾個?(請不要使用全域變數)

```

#include <stdio.h>
void find(int num, int ith, int *count) {
    int i, c;
    if (ith==10) { // 遞迴的結束條件
        printf("%d\n", num);
        (*count)++;
    }
    else
        for (i=1; i<10; i++)
            if ((c=num*10+i)%ith==0)
                find(c, ith+1, count);
    }
int main() {
    int count;
    find(0,1,&count);
    printf("count=%d\n", count);
    return 0;
}

```

- (c) [10] 如果每一位數是{1, 2,3,4,5,6,7,8,9}集合中**不同**的數字，我們可以修改上面的程式來完成，下面的程式用一個陣列 digits 來存放可以使用的數字，一開始時存放 9 個數字，用掉一個以後，就把那個數字和陣列第 9-ith 個數字交換，如此確保第 ith 位數字有 10-ith 個數字可以挑選

```
#include <stdio.h>

void swap(int digits[], int i, int j) {
    int tmp = digits[i];
    digits[i] = digits[j];
    digits[j] = tmp;
}

int find(int num, int ith, int digits[]) {
    int i, c, r;
    if (ith==10) {
        printf("%d\n", num);
        return num;
    }
    for (i=0; i< 10-ith; i++) //嘗試 digits 陣列中剩下的數字作為第 ith 位數
        if ((c=num*10+ digits[i] )% ith ==0) {
            swap(digits, i, 9-ith);
            if (r=find(c, ith+1, digits))
                return r;
            swap(digits, i, 9-ith);
        }
    return 0;
}

int main() {
    int digits[]={1,2,3,4,5,6,7,8,9};
    find(0,1,digits);
    return 0;
}
```

3. 在一個遊戲程式中需要紀錄一個角色在空間中的位置(position)、速度(velocity)、以及質量(mass)

- (a) [5] 請定義一個包含三個欄位的使用者自定結構型態 body，其中位置、速度都是三個倍精準浮點數的陣列代表 x, y, z 方向的分量，質量是一個整數

**Sol:**

```
struct body {
    double position[3];
    double velocity[3];
    int mass;
```

```
};
```

- (b) [5] 假設整數變數 n 裡有一個介於 500~1000 的資料，請定義一個 n 列 10 行每個元素是 struct body 的二維陣列的指標變數 army

**Sol:**

```
struct body (*army)[10];
```

- (c) [10] 請以 malloc 函式配置 n 列 10 行的二維 struct body 陣列，紀錄在 army 指標變數中，配置記憶體後請檢查 malloc 函式是否有配置成功，失敗時列印錯誤訊息並結束程式

**Sol:**

```
army = (struct body (*)[10]) malloc(sizeof(struct body)*n*10);
if (army==NULL) {
    printf("Error in allocating memory for army\n");
    return 1;
}
```

- (d) [10] 請定義一個 readData 函式，可以用 readData(n, army) 來呼叫，函式內開啟一個檔案 "sprites.dat"，以迴圈讀取檔案內的 n\*10 筆資料，每筆資料一列有 7 個數值，並且關閉檔案

**Sol:**

```
void readData(int n, struct body (*army)[10]) {
    int i, j, k;
    double *d;
    FILE *fp = fopen("sprites.dat", "r");
    if (fp==NULL) printf("error in opening sprites.dat\n");
    for (i=0; i<n; i++)
        for (j=0; j<10; j++) {
            for (k=0; k<3; k++)
                fscanf(fp, "%lf", &(army[i][j].position[k]));
            for (k=0; k<3; k++)
                fscanf(fp, "%lf", &(army[i][j].velocity[k]));
            fscanf(fp, "%d", &(army[i][j].mass));
        }
    fclose(fp);
}
```

- (e) [10] 請撰寫 comp 函式並且完成下列迴圈運用 qsort 將 army 陣列裡面每一列的 10 筆資料依照質量以及與原點的歐幾里得距離排序，主要依照質量由大到小排序，質量相等時再依照與原點距離由小到大排序

```
int comp(const void *a, const void *b) {

}

for (i=0; i<n; i++)
    qsort(_____, _____, _____, _____);
```

**Sol:**

```

int comp(const void *a, const void *b) {
    struct body *p[2]={(struct body*)a, (struct body*)b};
    double d[2];
    int i, j;
    if (p[1]->mass==p[0]->mass) {
        for (i=0; i<2; i++)
            for (d[i]=j=0; j<3; j++)
                d[i] += p[i]->position[j] * p[i]->position[j];
        return d[0]<d[1] ? -1 : 1;
    }
    else
        return p[1]->mass - p[0]->mass;
}

```

```

for (i=0; i<n; i++)
    qsort(array[i], 10, sizeof(struct body), comp);

```

(f) [5] 請作適當的檢查後釋放動態配置的記憶體

**Sol:**

```

if (array) free(array);

```