

《資料結構》

- 一、給予如下二元樹節點的宣告，分別寫出 C 的遞迴程式計算二元樹節點個數及計算二元樹葉節點 (leaves) 個數 (Count the number of nodes in a binary tree and count the number of leaf nodes in a binary tree, respectively)。(25 分)

```
struct node {
    int info;
    struct node *left;
    struct node *right;
}
typedef struct node *NODEPTR;

void countTree(NODEPTR tree) {

}

void countLeaves(NODEPTR tree) {

}
```

試題評析	本題屬於二元樹基本處理程式的設計問題，題目本身非常簡單，只是在多年未考到此類程式以來，或許有部份考生都把焦點集中在演算法和其他較理論的部份，而不再準備樹的程式設計問題，山雨欲來，也反應出即將到來的考科變化。
考點命中	1.《資料結構》高點文化出版，王致強編著，頁6-25，精選例題25。 2.《資料結構》高點文化出版，王致強編著，頁6-25~26，精選例題26。

答：

(一)

```
int countTree(NODEPTR tree) {
    if (tree == NULL) return 0;
    else return countTree(tree->left) + countTree(tree->right) + 1;
}
```

}

(二)

```
int countLeaves(NODEPTR tree) {
    if (tree == NULL) return 0;
    else {
        int leaves=countLeaves(tree->left) + countLeaves(tree->right);
        if (leaves>0) return leaves;
        else return 1;
    }
}
```

}

- 二、給予如下二元樹節點的宣告，寫一 C 的遞迴程式 swapTree (NODEPTR tree) 將每一節點的左、右節點互換 (Swap the left and right children of every of a binary tree)。(25 分)

```

struct node {
    int info;
    struct node *left;
    struct node *right;
}
typedef struct node *NODEPTR;

void swapTree(NODEPTR tree) {

```

試題評析	本題亦屬於二元樹基本處理程式的設計問題，題目本身也十分簡單，此題也多次出現於許多考試。
考點命中	《資料結構》高點文化出版，王致強編著，頁6-28，精選例題31。

答：

```

int swapTree(NODEPTR tree) {
    if (tree != NULL) {
        NODEPTR temp;
        temp=tree->left;
        tree->left = tree->right;
        tree->right = temp;
        swapTree(tree->left);
        swapTree(tree->right);
    }
}

```

三、給予如下程式，假設 $x[] = [30, 75, 53, 47, 21, 94, 88, 39]$ ， $lb = 0$ ， $ub = 7$ ，請問執行完下列程式後， $x[]$ 的內容為何？（25分）

```

void divide&conquer (int x[], int lb, int ub, int *pj)
{
    int a, down, temp, up;
    a = x[lb];
    up = ub;
    down = lb;
    while (down < up) {
        while (x[down] <= a && down < ub)
            down++;
        while (x[up] > a)
            up--;
        if (down < up) {
            temp = x[down];
            x[down] = x[up];
            x[up] = temp;
        }
    }
    x[lb] = x[up];
}

```

```

x[up] = a;
*pj = up;
}

```

試題評析	本題是Quick Sort的partition處理程式，即使未看過此一程式，細心追蹤程式，仍可以寫出正確的執行結果。
考點命中	《資料結構》高點文化出版，王致強編著，頁9-24，第9-6節；partition程序。

答：

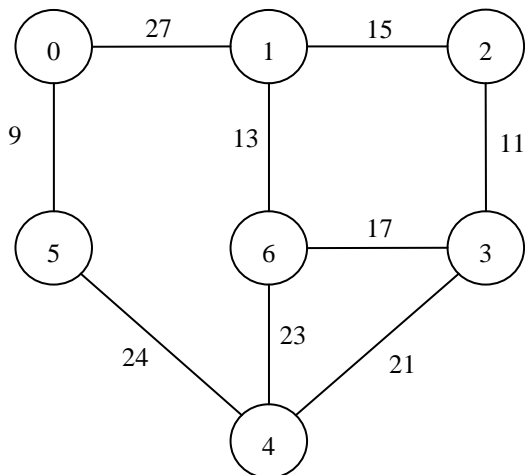
本題是Quick Sort所使用的partition程序，以30做為pivot，將資料分成<30與>30的兩群，最後並且將30移到兩群的中間。最後x[]的內容如下：
[21, 30, 53, 47, 75, 94, 88, 39]

四、用 $G = (V, E)$ 表示一個無方向性圖形，其中 V 是點的集合， E 是一組節點 (Vertices) 形成一個邊及對應權重 (Weights) 所組成的集合，例如： $(0, 1, 28)$ 表示節點 0 至節點 1 有一個邊，而且權重為 28。今有一圖形 $G = (V, E)$ ， $V = \{0, 1, 2, 3, 4, 5, 6\}$ ， $E = \{(0, 1, 27), (1, 2, 15), (2, 3, 11), (0, 5, 9), (1, 6, 13), (4, 5, 24), (4, 6, 23), (3, 4, 21), (3, 6, 17)\}$ 。請利用 Kruskal 演算法計算最小擴張樹 (Minimum spanning tree) 之最低權重或成本值。(25 分)

試題評析	本題是常考的最低成本伸展樹問題，只要熟悉Kruskal演算法，就可以做出正確的結果。
考點命中	《資料結構》高點文化出版，王致強編著，頁8-54~56，精選例題38。

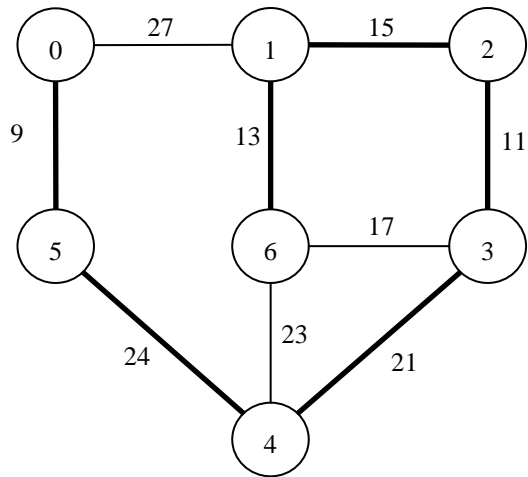
答：

題目原圖如下：



使用Kruskal演算法選用的邊以組線條表示如下：

【版權所有，重製必究！】



伸展樹的最低權重 $=9+11+13+15+21+24=93$ 。

高
點
·
高
上

【版權所有，重製必究！】