

## 《程式設計概要》

### 試題評析

今年程式設計概要已不強調使用何種語言撰寫程式，因此只要熟悉一種語言即可，建議使用C語言。而在考題上注意迴圈及遞迴用法，如第1、2、4題使用迴圈，第3題使用遞迴。迴圈問題仔細作答應沒問題，遞迴部分需程式設計經驗豐富者始可拿高分，因此今年一般同學應可得六、七十分左右分數。

本試題可使用C, PASCAL, COBOL, BASIC, FORTRAN或其他程式語言撰寫程式，惟請標示清楚你所用的語言。程式撰寫的評分依據包括：正確性、效率與可讀性。

一、請設計一個程式來計算 $2! + 4! + 6! + 8! + 10!$ ，其中!的意義為：

$n! = n * (n - 1) * \dots * 2 * 1$  (25分)

答：#include <stdio.h>

```
int factor(int n){
    int i,ans=1;
```

```
    if(n>0)
        for(i=1;i<=n;i++)
            ans*=i;
    else ans=0;
    return(ans);
}
```

```
void main(void){
    int i,product=0;
```

```
    for(i=2;i<=10;i=i+2)
        product+=factor(n);
    printf("2!+4!+6!+8!+10!=",product);
}
```

二、請設計一個程式檢查營利事業統一號碼是否正確。其處理方式如下：營利事業統一號碼共有八位，前七位為流水號，每一位的權重分別為8, 7, 6, 5, 4, 3, 2。營利事業統一號碼的前七碼與對應的權重相乘後，累加起來，累加的總和除以10以後的餘數，應該要與最後一位的檢查碼相同，才是正確的營利事業統一號碼；否則就是錯誤的營利事業統一號碼。(25分)

答：#include <stdio.h>

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

```
int parse(char str[]){ /* 0 is unsuccessful and 1 is successful */
    char token;
    int sum=0,length,i;
```

```
    length=strlen(str);
    for(i=0;i<length-1;i++)
        if(str[i]<'0' || str[i]>'9') return(0);
    else sum+=atoi(str[i])*(8-i);
```

```
if((sum%10)==itoa(str[length-1])) return(1);
else return(0);
}

void main(void){
char input[10];

printf("Please input a number(8 digits)\n");
scanf("%s",input);
if(parse(input)==1) printf("The number is right!!\n");
else printf("The number is wrong!!\n")

}
```

三、所謂二元搜尋樹 ( Binary Search Tree ) 是一種特殊的二元樹，對於樹上的每一個內部節點而言，它的左子樹所有的節點的值都小於它的值，而右子樹所有的節點的值都大於它的值。試寫一個在二元搜尋樹中，增加一個新節點的副程式。( 25分 )

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

struct node{
int value;
struct node *left;
struct node *right;
};

void insert(struct node *ptr,struct node *new1){

if(ptr==NULL){
ptr = new1;
return;
}

if(new1->value < ptr->value) insert(ptr->left,new1);
else insert(ptr->right,new1);
}

void main(void){
int in;
struct node *head,*newnode;

printf("Please enter a integer: ");
scanf("%d",in);
newnode = (struct node *)malloc(sizeof(struct node));
insert(head,newnode);

}
```

四、如果有一個電腦語言只能認識下列的敘述 ( statement ) :

a = b                      將變數b的值存到變數a中

<code>a = a + 1</code>	將變數a的值加1
<code>a = a - 1</code>	將變數a的值減1
<code>if a &gt; 0 then goto L</code>	如果變數a的值大於0，則跳到地址標示 ( label ) L的那一個敘述
<code>halt</code>	停止執行

每一個敘述的前面，如果有需要，可以加上地址標示，例如：

```
L : a = a + 1
```

試寫一個程式，計算 $x = a + b$ （假設a為正數，b則正、負數均可）（25分）

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

void main(void){
    int a,b,x;

    printf("Please enter two number, the first is positive: ");
    scanf("%d %d",&a,&b);

    if(a<0){
        printf("Wrong number!\n");
        exit(0);
    }

    x = b;
    x=x-1;
    a=a+1;
loop:
    a=a-1;
    x=x+1;
    if(a>0) goto loop;
    printf("The sum is %d\n",x);
}
```