

《程式設計概要》

試題評析

今年程式設計的題目難易適中，沒有特殊困難的題目。一如往常一樣，三題程式題及一題流程圖設計。程式題並無指定使用何種語言，所以撰寫程式時比較彈性，建議使用C語言。根據最近幾年的考題趨勢，關於迴圈與遞迴的問題是仍是考題的重點，在準備時應測底了解迴圈及遞迴的設計法則，只要小心回答即可獲取高分。

一、撰寫一程式。（二十五分）

輸入：一個文字檔xxx.txt。

輸出：此文字檔中字元'A'出現之次數。

答：

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

void main(void)
{

FILE *fp;
int count,c;

fp=fopen("test.txt","r");

if(!fp){
printf("Can't open file!\n");
exit(1);
}

count=0;
while((c=getc(fp))!=EOF){
if(c=='A') count++;
printf("%d\n",c);
getch();
}
printf("The number of 'A' is %d\n",count);
getch();
}
```

二、撰寫一程式。（二十五分）

輸入：1.一正整數n。

2.n個數 x_1, x_2, \dots, x_n 。

輸出： $\min_{1 \leq i < j \leq n} |x_i - x_j|$ 之值。

答：

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

```

void main(void)
{
    int i,j,diff,n,num[100],min;

    printf("Please enter a number N: ");
    scanf("%d",&n);

    printf("Please enter the numbers\n");
    for(i=0;i<n;i++)
        scanf("%d",&num[i]);

    min=(num[0]>num[1])?num[0]-num[1]:num[1]-num[0];
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++){
            diff=(num[i]>num[j])?num[i]-num[j]:num[j]-num[i];
            if(diff<min) min=diff;
        }

    printf("The minimum value is %d\n",min);

    getch();
}

```

- 三、(a)請用函數形式給一含有尾端遞迴(tail recursion)之函數。
 (b)請定義一與(a)等價但不含尾端遞迴之函數。
 (c)請簡略說明上列兩函數之對應程式在執行上的優劣。(二十五分)

答：

(a)

```

int sum_rec(int n){
    if(n==1) return(1);
    else return(n+sum_rec(n-1));
}

```

(b)

```

int sum_loop(int n){
    int i,sum=0;

    for(i=1;i<=n;i++)
        sum+=i;

    return(sum);
}

```

(c)使用遞迴(recursive)程式指令個數較少，但執行時間較長。使用迴圈(loop)則反之。

- 四、請設計一可將任一輸入之正整數以二元(binary)表示之程式流程圖。(二十五分)

答：

```

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

```

```

void main(void)
{
    int i,n,top,num,s[100];

    printf("Please enter an integer: ");
    scanf("%d",&num);

    top=0;
    while(num!=0){
        n=num%2;
        num=num/2;
        s[top]=n;
        top++;
    }

    for(i=top-1;i>=0;i--)
        printf("%d",s[i]);

    getch();
}

```

