

《程式設計概要》

試題評析

93年的考題相當容易，包含第一題使用遞迴方式計算費氏級數、第三題資料結構的循序搜尋法與第四題的佇列中加入元素、刪除元素的問題在高上課堂上均已詳細講解過。至於第二題印出星號的問題，也曾是課堂在講解雙重迴圈時，特別強調的必考重點。因此，本年度的考題高上學員要得到七、八十分應該不成問題。

一、已知費氏數列 (Fibonacci sequence) 如下：

0, 1, 1, 2, 3, 5, 8, 13, 21, ……

撰寫一遞迴函數fibonacci，其輸入參數為大於或等於0之整數n，輸出為費氏費列第n個值。假設0, 1, 1, …分別稱為第0個值、第1個值、第2個值，依此類推。(25分)

答：

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

```
int fib(int n){
    if(n<0) return(-1); /* input error */
    if(n==0||n==1) return(n);
    else return(fib(n-1)+fib(n-2));
}
```

```
void main(void){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    printf("The Fibonacci number is: %d",fib(n));
}
```

二、撰寫一程式：(25分)

輸入：一個正整數n。

輸出：印出如下之n列資料，第一列是一個*號、第二列是二個*號、直到第n列是n個*號。

```
*
**
***
****
.....
```

答：

```
#include <stdio.h>

void star(int n){
    int i,j;
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
            printf("*");
        printf("\n");
    }
}

void main(void){
```



```

int n;
do{
    printf("Enter a number: ");
    scanf("%d",&n);
    if(num<1) printf("Input Error\n");
}while(n<1);

star(n);
}

```

三、撰寫一函數Linear Search，能從有n個元素的整數陣列b，用線性搜尋法（linear search），搜尋一個整數值key在陣列b中的位址，若key不在陣列b中，則傳回-1。用Big-O符號表示此函數之複雜度（complexity），並說明之。（25分）

答：

(1)

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

```

int linear(int *b,int n,int key){
    int i;
    for(i=0;i<n;i++){
        if(b[i]==key) return(i);
    }
    return(-1);
}

```

```

void main(void){
    int i,key,b[1000],n;
    do{
        printf("Enter Array Size: ");
        scanf("%d",&n);
        if(n<1) printf("Input Error\n");
    }while(n<1);

```

```

    for(i=0;i<n;i++){
        printf("The number %d is: ",i);
        scanf("%d",&b[i]);
    }

```

```

    printf("Enter the number you want to search: ");
    scanf("%d",&key);
    printf("The location is: %d",linear(b,n,key));
}

```

(2) $O(n)$, linear search有可能第1筆,第2筆...第n筆找到,也可能找不到,因此平均時間複雜度為 $(1+2+3+\dots+n)/n=(n+1)/2$,所以為 $O(n)$

四、利用陣列 (array) 宣告一個可作佇列 (queue) 的結構及相關變數。撰寫兩個佇列 (queue) 常用的運算函數 insert 與 delete。(25分)

答：

```
#include <stdio.h>

int flag=0,q[20],n=20,front=0,rear=0;

int insert(int data){
    if(front==rear && flag==1){
        printf("queue full\n");
        exit(); /* queue full */
    }

    rear=(rear+1)%n;
    q[rear]=data;
    if(front==rear) flag=1;
    return(1); /* Successful */
}

int del(void){
    int data;
    if(front==rear && flag==0){
        printf("queue empty\n");
        exit(); /* queue empty */
    }

    front=(front+1)%n;
    data=q[front];
    if(front==rear) flag=0;
    return(data);
}

void main(void){
    int i;
    insert(1);
    insert(2);
    insert(3);
    insert(4);

    del();
    del();

    for(i=0;i<20;i++)
        printf("%4d",q[i]);
}
```

