

## 電腦科（1993 年）試卷二

Q2. 某紙牌遊戲中有 20 張牌，分別以數字 0、0、1、1、2、2、3、3、4、4、5、5、6、6、7、7、8、8、9、9 表示，而你和電腦為對手。遊戲開始時，隨意分給每方 3 張牌，其中第 3 張牌面向上，讓彼此得知對手的第 3 張牌。雙方都可選擇不換牌，或以一張新牌取代第 1 或第 2 張牌一次，然後必須從自己的 3 張牌中選出兩張來，根據下列的規則決定勝負：

規則 1：「成對」(pair) 與「不成對」(即：兩牌數值不同) 比較，  
「成對」者勝。

規則 2：兩「成對」比較，點數(數值)高者勝。

規則 3：雙方都不成對，兩牌數值總和高者勝，同值則作和論。

1993

i	0	1	2	3	4	5	6	7	8	9
freq[i]	0	0	0	0	0	0	0	0	0	0

已發牌數目

i	0	1	2
comp[i]	-1	-1	-1

電腦的 3 張牌

i	0	1	2
user[i]	-1	-1	-1

玩家的 3 張牌

<pre>void initArrays(){ // 初始化     int i;     for(i=0; i&lt;10; i++) freq[i]=0;     for(i=0; i&lt;3; i++) comp[i]=-1;     for(i=0; i&lt;3; i++) user[i]=-1; }</pre>	<pre>void draw3cards(){     int i;     for(i=0; i&lt;3; i++){         comp[i]=card();         user[i]=card();     }     displayCards(); }</pre>
---	---

以下子程式 card 會利用 rand() 傳回一個隨機數 (0-9)，  
但每張牌(0-9)只有兩張。

freq[ ] 是用來記錄被抽中次數，確保不會超出上限(2)。

i	0	1	2	3	4	5	6	7	8	9
freq[i]	0	0	0	0	0	+1	0	0	0	0

以下子程式 draw3cards 會利用 card() 分別抽取 3 張牌給 comp 及 user。並呼叫子程式 displayCards 顯示所得紙牌。

i	0	1	2
comp[i]	7	2	5

i	0	1	2
user[i]	0	2	0

以下子程式 `displayCards` 顯示電腦及玩者所得紙牌點數如下。

玩者： 0 2 0	<code>user[i]</code>
電腦： X X 5	<code>comp[i]</code>

1993

```
void displayCards(){
    printf("玩者: \t%d\t%d\t%d\n", user[0],user[1],user[2]);
    printf("電腦: \tX\tX\t%d\n", comp[2]);
}
```

玩者： 0 2 0	玩者： 2 2 0
電腦： X X 5	電腦： X X 5
要換牌嗎 <Y/N> ?      Y	
要換掉哪一張牌 0/1 ?    0	

指標 **pointers** `x` 及 `y`  
數值若有改變  
會傳回外面

```
void changeCard(){
    char ans;  int choice;
    printf("要換牌嗎 <Y/N> ?  ");
    scanf("%c", &ans);
}
}
```

```
void user_select2cards (int *x, int
*y){
    do{
        printf("請選兩張牌(0,1,2)? ");

    }while(
);
}
```

```
void compareCards(){
    int i,x,y;
    user_select2cards (&x,&y);
    YPair =
    YPoint =
    printf("玩者選了: \t%d\t%d\n", user[x], user[y]);

    CPoint =
    printf("電腦選了: \t%d\t%d\n", comp[x], comp[y]);
}
```

```
main(){
    srand(time(NULL));
    do{
        initArrays();      // 初始化
        draw3cards();      // 各抽 3 張
        changeCard();      // 換牌
    }
```

```

compareCards(); // 比較
if(CPoint==YPoint) printf("打和\n");
else if(CPoint>YPoint) printf("電腦勝\n");
else printf("玩者勝\n");
}while(tryAgain());
}

```

此子程序實現下列電腦的替換策略：

如果「成對」，則無需替換。  
 如果「不成對」，而對方的第 3 張牌小於 5，則替換第 1、2 張牌中較小的一張。

Q2. 已知函 KR 及 KL 如下：1996

11	<pre> void KR (char s[]){     s[<u>strlen(s)-1</u>]='\0'; } </pre>	1996
16	<pre> void KL (char s[]){     int i,n=strlen(s);     for (i=0; i&lt;n; i++) s[i]=s[i+1];     s[<u>n-1</u>]='\0'; } </pre>	

(a) 寫出列程序語句的輸出：

1. strcpy(source, "123456"); KR(source); puts(source);
2. strcpy(source, "123456"); KL(source); puts(source);
3. strcpy(source, "123456"); KL(source); KR(source); puts(source);

(b) 描述下列各過程的目的： (i) KR (ii) KL

已知過程 KRN 及 KLN 如下：

<pre> void KRN (char s[], int n){     while (n&gt;0){         KR(s);         n--;     } } </pre>	<pre> void KLN (char s[], int n){     while (n&gt;0){         KL(s);         n--;     } } </pre>
--	--

(c) 寫出下列程序語句的輸出：

1. strcpy(source, "123456"); KRN(source, 2); puts(source);
2. strcpy(source, "123456"); KLN(source, 4); puts(source);

(d) 就下列各情況，寫出過程 KRN(s,n)的結果。

- (i)  $n \leq 0$
- (ii)  $0 < n < \text{strlen}(s)$
- (iii)  $n \geq \text{strlen}(s)$

(e) 若輸入字符"123456"，寫出程序的輸出。

<pre> 91 printf ("Enter a string : ");    gets (source);    n = strlen(source);    for (x=0; x&lt;n; x++){        KRN (source,x);        puts (source);    } </pre>	1996
---	------

(f) 現將主程序修改如下：

<pre> 91 do{    printf ("Enter a string : "); gets (source);    n = strlen(source)/2;    for (x=0; x&lt;n; x++){        KRN (source,x);        KLN (source,x);        strcat(star,"*");        printf ("%s%s%s\n", star,source,star);    } }while (strlen(source)!=0); </pre>	
---	--

就下列各輸入字符串，寫出經修改的程序的輸出。

(i) "123456"

(ii) "123456789"

## 電腦科（1997年）試卷二

1. (a) 文華欲編寫一程序，該程序：

- 接受一已排序且以「\*」作結尾的名字序列；
- 輸出序列內的名字，重複的名字只會輸出一次。

文華使用以下方法來編寫此程序：

逐一接受名字，並將名字置於一陣列內。

輸出陣列的第一個元素。

在餘下的各元素中，若元素與前一個元素不相同，則輸出該元素。

以下是這程序的一個運行樣本：

<pre> Please enter names: Bill Bill Cindy David David David Kelvin * </pre>	<pre> The output is: Bill Cindy David Kelvin </pre>
---	---

1997

文華編寫的程序如下指示：

```

char name[20], A[100][20];
main(){
    int cnt=0, k, done=0;
    printf("Please enter names: \n");
    do{
        gets(name);
        done =
        if(!done){
            strcpy(A[cnt],name);
            cnt++;
        }
    }while (!done);           // while(done==0);

    printf("The output is: \n%s\n", A[0]);
    for(k=1; k<=cnt; k++)
        if( ) printf("%s\n", A[k]);
}

```

Q2.

```

int F1(char ipstr[], char xstr){
    int p=0,flag=0;
    while(flag==0 && p<strlen(ipstr)){
        if(ipstr[p]==xstr) flag=1;
        p++;
    }
    return flag;
}

```

程序語句	語句執行後 flag 的值
flag = F1("AABBCCDDEE", 'A');	
flag = F1("ABCJKLRST", 'W');	

1997

```

void F2(char ipstr[], char xstr, int *p, int *flag){
    *flag=0; *p=0;
    while(*flag==0 && *p<strlen(ipstr)){
        if(ipstr[*p]==xstr) *flag=1;
        (*p)++;
    }
}

```

程序段	執行後 P 的值	執行後 flag 的值
P=2; F2 ("ZXYWLMA", 'W', &P, &flag);		
P=3; F2 ("ABCJKLRST", 'A', &P, &flag);		
P=9; F2 ("ABCABC", 'A', &P, &flag);		
P=1; F2 ("ABCABCD", 'A', &P, &flag);		

1. 若語句 F2 ("ABCABCD", 'A', &P, &flag); 執行後 flag 的值是 1，求此語句剛執行前 P 所有可能的值。
2. 若語句 F2 ("ABCABCD", 'A', &P, &flag); 執行後 flag 的值是 0，求此語句剛執行前 P 所有可能的值。
3. 設語句 F2(ipstring, 'A', &P, &flag); 剛執行前 P=5 及 strlen(ipstring)=10。若語句 F2(ipstring, 'A', &P, &flag); 執行後 flag 的值是 1，寫出在 ipstring 內可能出現字符「A」的所有位置。
4. 若語句執行後 &flag 及 &P 的值分別是 1 及 7，寫出在內出現字符「A」的位置。

下列過程乃按過程 F2 編寫而成：

```
void F3(char str1[], char str2[]){
    int k,p=0,flag=0;
    for(k=0;k<strlen(str2);k++){
        F2(str1, str2[k], &p, &flag);
        printf("f3 p=%d flag=%d\n", p, flag);
    }
}
```

- a 寫出執行語句 F3("ABCIJK","ABK")後所得的輸出。
- b 寫出執行語句 F3("ABK","ABCIJK")後所得的輸出。

### 電腦科（1998年）試卷二

2. 志強編寫了一個 C 程序來模擬一個遊戲。遊戲者如成功地把五隻皇后棋子(Queen) 放置在一個 5x5 的棋盤 (如下圖) 便算勝出：

		列 (Column)				
		1	2	3	4	5
行 (Row)	1	Q				
	2			Q		
	3					
	4		Q			
	5					

1998

放置皇后棋子要依循三條規則 (分別稱為規則一、規則二、規則三)。志強編寫的程序利用 `check1`, `check2` and `check3` 三個過程(子程式)來檢驗有否違反規則。

志強的程序如下：

<pre>#define MAX 8 char GB[MAX][MAX];</pre>	
<pre>void initGameBoard(){     int row, col;     for(row=0;row&lt;MAX;row++)         for(col=0;col&lt;MAX;col++)             GB[row][col]=' '; }</pre>	
<pre>void inputRowCol (int *row, int *col){     do{         printf("Input row and column of the Queen: ");         scanf("%d%d", ____);     }while(____); }</pre>	
<pre>int <b>check1</b> (int R, int C){     int i, ok=1;     while( R&gt;0 &amp;&amp; C&gt;0){         R--; C--;     }     while( ok &amp;&amp; R&lt;8 &amp;&amp; C&lt;8 ){         if(GB[R][C]=='Q') ok=0;         else{             R++; C++;         }     }     return ok; }</pre>	<pre>int <b>check2</b> (int R, int C){     int i, ok=1;     while( R&gt;0 &amp;&amp; C&lt;8){         R--; C++;     }     while( ok &amp;&amp; R&lt;8 &amp;&amp; C&gt;0 ){         if(GB[R][C]=='Q') ok=0;         else{             R++; C--;         }     }     return ok; }</pre>
<pre>int <b>check3</b> (int R, int C){     int i, ok=1;     for      return ok; }</pre>	

1998

```

main(){
    int QC=0, row, col;    // QC = queens' count
    int errs=0, OK1, OK2, OK3;
    initGameBoard();
    displayGameBoard();
    while( !errs && QC<MAX ){
        inputRowCol (&row,&col);    // scanf...
        OK1=check1(row,col);
        OK2=check2(row,col);
        OK3=check3(row,col);

        if(OK1 && OK2 && OK3) QC++;
        else errs=1;
        if(!errs) GB[row][col]='Q';
        displayGameBoard();
    }
    if(QC==MAX) printf("You win!\n");
    else printf("You lose!\n");
}

```

(e) 在下面的棋盤中已放置了一隻皇后棋子：

		列				
		1	2	3	4	5
行	1					
	2		Q			
	3					
	4					
	5					

在棋盤的各空格上，填上

- '1'，若將一皇后棋子置於該格將違反規則一 check1；
- '2'，若將一皇后棋子置於該格將違反規則二 check2；
- '3'，若將一皇后棋子置於該格將違反規則三 check3。

3. 文本檔案 TFILE 中每一行恰好有 20 個字符。每個字符均為英文字母 ('a','b',...,'z', 'A','B',...,'Z') 或冒號 (':')。在 TFILE 各行中，首、末字符均為英文字母。

TFILE 中的一個「字」是指一連串的英文字母，以一個或多個冒號為分隔符。

舉例說，以下 TFILE 中的一行有四個「字」：

AbcDE:dcBA:::AB::::A
----------------------

偉明打算編寫一個 C 程序，將 TFILE 中的各「字」分行打印出來。

舉例說，如果 **TFILE** 有以下各行：

```
AbcDE:dcBA:::AB::::A
SHE:IS:::YESTERDAY:B
NOW:::ABCDE::SCHOOLS
```

則程序輸出以下各行：

```
AbcDE
dcBA
AB
A
SHE
IS
YESTERDAY
B
NOW
ABCDE
SCHOOLS
```

- (a) 試編寫一個過程，接受一個字符並檢驗它是否為英文字母。若字符為英文字母，則過程回送布爾常數 **1(TRUE)**，否則回送 **0(FALSE)**。編寫過程時須使用以各行：

```
int validChar(char ch){
    return ____
}
```

- (b) 試編寫一過程，接受一行文本(字符串)，並將在其中找到的「字」分行打印出來。編寫過程時須使用以下各行 (考生必須使用(a)中的 **ValidChar** 過程。)

```
31 void pWords(char line[]){
    int k,fg=0;
}
```

(註：K 作為循環控制變量使用；fg 為一用作標記 flag 的變量。)

偉明利用上面 (a) 和 (b) 中的 **ValidChar** 和 **pWords**，很容易就編寫好整個程序。偉明的程序須使用以下變量：

變量	用途
<b>inFile</b>	貯存輸入文本檔案
<b>TLine</b>	貯存文本檔案中一行文本
<b>FName</b>	貯存輸入文本檔案的檔案名
<b>Count</b>	循環控制

- (c) 試編寫主程序，以完成偉明的程序。程序須將文本檔案中的字符逐一讀進 TLine，並在 VDU 上顯示以下輸出。(在此輸出中，各問號後面的信息均由用戶以鍵盤輸入，其餘則由程序輸出。)

Name of text file?	<u>TFILE</u>
Words in the file are:	
AbcDE	
dcBA	
...	

1998

```
main(){
    printf("Name of text file (9803.txt)? ");

    printf("Words in the file are:\n");

}
```

### 電腦科（1999 年）試卷二

- Q1 (a) 小偉打算編寫一個程序。該程序把兩個  $4 \times 4$  的表內的數字相加。表 3 內的數字是由表 1 和表 2 應格子內的數字相加而成。例如：

表 1					表 2					表 3			
1	2	3	4	+	0	1	0	1	=	1	3	3	5
2	3	4	5		1	0	1	0		3	3	5	5
0	0	0	0		3	4	5	6		3	4	5	6
4	5	6	7		1	1	1	1		5	6	7	8

該程序須在 VDU 上顯示如下的輸出。(在此輸出中，各冒號後面的數據均由用戶以鍵盤輸入，其餘則由程序輸出。)

Table 1 row 1 :	1 2 3 4	The sum of tables 1&2 >
Table 1 row 2 :	2 3 4 5	1 3 3 5
Table 1 row 3 :	0 0 0 0	3 3 5 5
Table 1 row 4 :	4 5 6 7	3 4 5 6
Table 2 row 1 :	0 1 0 1	5 6 7 8
Table 2 row 2 :	1 0 1 0	
Table 2 row 3 :	3 4 5 6	
Table 2 row 4 :	1 1 1 1	

小偉的程序如下：

```
#define MAX 4  
int table1[MAX][MAX], table2[MAX][MAX], i, j;
```

1999

```
void output(){  
    printf ("\nThe sum of tables 1&2 >\n");
```

```
    for
```

```
}
```

```
void inputTable1(){
```

```
    for
```

```
}
```

```
void inputTable2(){
```

```
    for
```

```
}
```

```
void process(){
```

```
    for
```

```
}
```

```
main(){
```

```
    printf("Enter two %dx%d matrix: \n\n", MAX,MAX);
```

```
    inputTable1();
```

```
    inputTable2();
```

```
    process(); // table1+table2
```

```
    output();
```

```
}
```

## 2. 細閱以下『程式』並回答以下各問題。

1999

```

#define NUM 5
const int digit=2;
int bin[10][NUM], data[NUM] = { 37,23,43,20,15};
int key, pass, i,j,k, empty;

main(){
200 printf("No.of positive integers to be sorted: %d\n", NUM);
210 for(i=0; i<NUM; i++){
230     printf("Enter a %d-digit integer: ",digit);
240     scanf("%d",&data[i]);    // 37,23,43,20,15
    }

260 for(pass=1; pass<=2; pass++){
280     for(i=0; i<10; i++){
300         for(j=0; j<4; j++) bin[i][j]=0;

310         for(i=0; i<5; i++){
330             switch (pass){
340                 case 1: key=data[i]%10; break;
350                 case 2: key=data[i]/10; break;
360             }
370             k=empty=0;
380             do{
390                 if(bin[key][k]>0) k++;
400                 else empty=1;
410             }while (empty==0);
420             bin[key][k] = data[i];
            }

440         k=0;
450         for(i=0; i<10; i++){
460             for(j=0; j<4; j++){
470                 if (bin[i][j]>0){
490                     data[k]=bin[i][j];
500                     k++;
                }
            }
        }

520 for(i=0; i<NUM; i++) printf("%i\n", data[i]);

```

bin[i][j]

	0	1	2	3
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

bin[i][j]

	0	1	2	3
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

- (a) 當 for 語句 pass=1 完成執行後，寫出在陣列 bin[i][j] 中貯存的數值。  
 (b) 當 for 語句 pass=2 完成執行後，寫出在陣列 bin[i][j] 中貯存的數值。

Q2 某學生編寫了以下的 C 程序，把用戶輸入的兩位正整數排序。

請小心細閱程序並回答以下各問題。附加在程序中之行號只作參考用。

(a) 指出哪一行決定須排序的正整數數目。

(b) 在 280 行至 300 行之間的嵌套 for 循環有什麼作用？

(c) 考慮 260 行、280 行及 290 行。300 行會執行多少次？

假設用戶把整數 37、23、43、20、15 順序輸入至程序中。

(d) (i) 當 Pass = 1，每次剛執行完 340 行時 Key 之值為何？試填寫下表。

I	Key
1	7
2	3
3	
4	
5	

1999

(ii) 當 Pass = 1，下表顯示輸入的各整數在 Bin 中貯存的位置。

試寫出程序在 250 行至 550 行之輸出。

輸入的整數	整數在 Bin 中貯存的位置
37	
23	
43	
20	
15	

(e) (i) 當 Pass = 2，每次剛執行完 350 行時 Key 之值為何？試填寫下表。

I	Key
1	2
2	2
3	
4	
5	

(ii) 當 Pass = 2，填寫下表以顯示輸入的各整數剛在 520 行之前在 Bin 中貯存的位置。

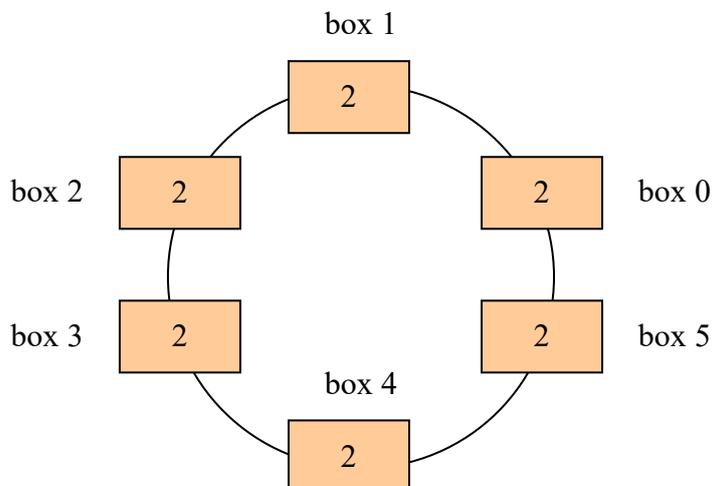
輸入的整數	整數在 Bin 中貯存的位置
37	
23	
43	
20	
15	

(iii) 當 Pass = 2，寫出程序在 520 行至 550 行之輸出。

(f) 建議如何修改此程序，使此算法可將三位正整數排序。

Q3 彼得以 C 語言編寫一電腦遊戲程序。此遊戲的規格說明如下：

- I 此遊戲由玩者和電腦對奕。  
 II 此遊戲使用六個圍成一個圈的格子，分別稱為 `box[0]`、`box[1]`、`box[2]`、`box[3]`、`box[4]`、`box[5]`，如下圖：



1999

- III 開始遊戲時，每個格都有一數值 2。  
 IV 遊戲由玩者開始。玩者和電腦輪流擲骰。  
 每次擲出的隨機數  $n$  會根據下列規則用來更新某些格子之值：

規則甲：`box n` 之值會增加 2。

規則乙：旁邊兩格之值會各減少 1。

例如：

如果  $n=3$ ，則 `box 3` 會加 2，而 `box 2` 和 `box 4` 會各自減 1。

如果  $n=1$ ，則 `box 1` 會加 2，而 `box 2` 和 `box 0` 會各自減 1，餘此類推。

- III 若擲骰後有任何格子之值變為負數，則對方勝。

下圖顯示兩局遊戲輸出例子：

輸出例一：	輸出例二：
Box # : 0 1 2 3 4 5	Box # : 0 1 2 3 4 5
Value : 2 2 2 2 2 2	Value : 2 2 2 2 2 2
按任意鍵...擲骰	按任意鍵...擲骰
玩者擲了 4.	玩者擲了 4.
Box # : 0 1 2 3 4 5	Box # : 0 1 2 3 4 5
Value : 2 2 1 4 1 2	Value : 2 2 1 4 1 2
電腦擲了 6.	電腦擲了 1.
Box # : 0 1 2 3 4 5	Box # : 0 1 2 3 4 5
Value : 1 2 1 4 0 4	Value : 4 1 1 4 1 1
按任意鍵...擲骰	按任意鍵...擲骰
玩者擲了 2.	玩者擲了 3.

<pre>Box # : 0 1 2 3 4 5 Value : 0 4 0 4 0 4  電腦擲了 6. Box # : 0 1 2 3 4 5 Value :-1 4 0 4 -1 6  玩者勝.</pre>	<pre>Box # : 0 1 2 3 4 5 Value : 4 0 3 3 1 1  電腦擲了 6. Box # : 0 1 2 3 4 5 Value : 3 0 3 3 0 3  按任意鍵...擲骰 玩者擲了 3. Box # : 0 1 2 3 4 5 Value : 3 -1 5 2 0 3  電腦勝.</pre>
--	---

彼得已編寫好首四行如下：

1	#include <stdio.h>
2	int box[6], n, turn;
3	char turn[10];
4	main() {

1999

變量	描述
box	貯存六個格子之值的陣列
turn	貯存輪到哪方的變量 (0=電腦; 1=玩者) 或 (computer=電腦; player=玩者)
n	貯存擲骰所得之值的變量

(a) 試編寫一子程序 `init`，對六個格子賦初始值。

<pre>void init(){     int i;     srand(time(NULL));     for }</pre>	<pre>void applyRules(int n){ } }</pre>
---	--

(b) 試編寫一子程序 `applyRules`，落實遊戲規格說明中的規則甲和規則乙。

(c) 試編寫一子程序 `displaybox`，以下面的格式顯示各個格子的內容：

```
Box #      : 0 1 2 3 4 5
Value     : X X X X X X
```

其中 X 為各格子中的值。

<pre>void displayBox(){     int i;     printf("Box#: 0 1 2 3 4 5\n"); }</pre>	<pre>void changePlayer(){ }</pre>
---	-----------------------------------

(d) 試編寫一子程序 `endgame`，若任何一格子負值時回輸 1，否則回輸 0。

```
int endGame(){
    int i, theEnd=0;
}
```

(e) 使用(a)至(d)中的子程序編寫主程序。

```
main(){
    int n;
    init();
    displayBox();
    strcpy(turn,"Player");
    while(!endGame()){
        if(strcmp(turn,"Player")==0){
            printf("Press any key to roll the dice.\n");
            getch();
        }
        n = rand()%6;
        printf("\n%s rolls a %d.\n", turn,n);
        applyRules(n);
        displayBox();
        changePlayer();
    }
    printf("%s wins.\n",turn);
}
```

## 電腦科（2000 年）試卷二

Q1 (a) 試編寫一個C程序，接受兩字符串並輸出訊息，說明字符串2是否字符串1的一部分。該程序須在VDU上顯示如下的輸出。（在此輸出中，各冒號後面的數據均由用戶以鍵盤輸入，其餘則由程序輸出。）

```
Enter string1 : sixteen
Enter string2 : teen
string2 is a substring of string1.
```

2000

子程序 `substr()`：傳回0或1

```
int substr(char s1[], char s2[]){
    int n,i;
    char t[100]; ...
}

main(){
    char s1[50], s2[50];
```

```

gets(s1); gets(s2);
printf("string2 is %s a substring of string1\n",
      (substr(s1,s2)==0)?"NOT ":"");
}

```

- (b) 一小型雜貨店只售賣A,B,C,D,E 五種貨品。這五種貨品可分為三個類別。A,B,C屬於第1類，D屬於第2類，而E則屬於第3類。當顧客到櫃位付款時，收銀員的電腦會執行一程序來計算顧客購物的總數和每類別貨物的小計。該電腦程序的算法如下所示：

步驟1	說明total, i, qty為整數變量， pitem為字符變量，complete為布爾boolean變量(true/false)， 而subtotal為一元整數陣列，其大小為3。
步驟2	把陣列subtotal初始化為0， total初始化為0； 而complete初始化為false。
步驟3	當complete為false，重複步驟4至9。
步驟4	讀入貨品種類至pitem。
步驟5	若pitem等於 'Q'，則賦值true予complete；否則
步驟6	讀入顧客購買該貨品的數量至qty。
步驟7	當qty的輸入值不在1至10間（包括1和10），重複步驟6。
步驟8	依pitem的類別，將subtotal的對應元素增加qty。
步驟9	將total的值增加qty。
步驟10	顯示total和subtotal陣列。

試將算法改寫成C程序。

貨品種類：A 貨品數量：10 貨品種類：C 貨品數量：10 貨品種類：E 貨品數量：10 貨品種類：Q subtotal[1]=20 subtotal[2]=0 subtotal[3]=10 total = 30	貨品種類：D 貨品數量：20 貨品種類：B 貨品數量：10 貨品種類：Q subtotal[1]=10 subtotal[2]=10 subtotal[3]=0 total = 20	
--	---	--

- Q2 下列C程序讀入一字符串集合，把該集合貯存在一陣列，稍後利用另一整數陣列把該字符串排序和參照。程序中加入的行號只作參考用。

```

100 #define MAX 40
    int slist[MAX]; // link to next node
    char varray[MAX][10]; // unordered list
    int minv;

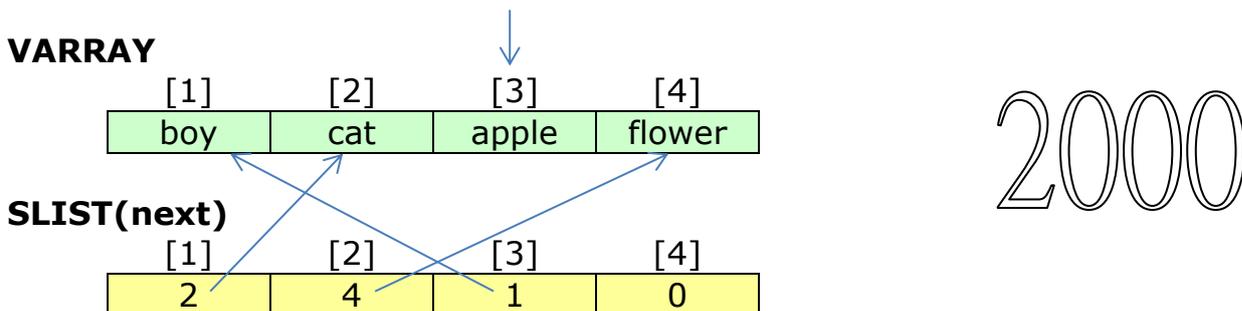
```

```

170 void PrtOrdSeq (int pos){
      while (pos>0){
          printf ("%s\n", varray[pos]);
          pos = slist[pos];
      }
}

260 int P1 (char sitem[], int nextstr) . . . . .
390 int P2 (char sitem[], int i, int nextstr) . . . . .
610 main(){
      int size1=0, i;
      char sitem[10];
      for(i=0; i<MAX; i++) slist[i]=0;
      do{
          printf ("Enter string %i: ", size1+1);   gets(sitem);
          if(strlen(sitem)>0)
              size1 = P2(sitem, size1, minv);
      }while(strlen(sitem)>0);
      PrtOrdSeq (minv);
}
    
```

- (a) 用來計算輸入字符串數目的變量是什麼？
- (b) 本程序中，陣列SLIST是用來參照和排序貯存在陣列VARRAY中的字符串。SLIST的指標對應貯存在VARRAY的字符串。貯存在SLIST的元素值是用來尋找VARRAY的下一個元素的字符串。若該值為0則表示再沒有下一個字符串了。下列為在輸入四個字符串後，VARRAY的前四個元素和SLIST的前四個元素的例子：



上例中，由SLIST表示的四個字符串的排序為 'apple', 'boy', 'cat' 和 'flower'。SLIST[3]的指標對應VARRAY的第3個元素 (VARRAY[3])，即 'apple'。下一個字符串的位置可由貯存在SLIST[3] 的值求得 (即1，而在VARRAY中，該位置 (即VARRAY[1]) 貯存 'boy' 類似地，下一個字符串 (即 'boy' 後的字符串) 的位置可由SLIST[1] 找到。

過程P1定義如下：

```

260 int P1 (char sitem[], int nextstr){
      int pos, done=0;
      do{
          pos = nextstr;
    
```

```

        nextstr = slist[pos];
        if(nextstr==0 ||
           (strcmp(sitem,varray[nextstr])<=0)) done=1;
    }while(!done);
    return pos;
}

```

(i) 空運行P1並完成下表：

情況	傳入程序的值	回傳值
1	sitem = "bus" 和 nextstr = 3	
2	sitem = "archie" 和 nextstr = 3	
3	sitem = "dog" 和 nextstr = 1	

(ii) P1的目的是什麼？

2000

(c) 程序P2定義如下：

```

390 int P2 (char sitem[], int i, int nextstr){
    int curstr, newpos;
    newpos = i+1;
    strcpy(varray[newpos],sitem);
    if(i==0) minv=1;
    else{
        if (sitem <= varray[minv]){
            slist[newpos] = minv;
            minv = newpos;
        }else{
            curstr = P1(sitem, nextstr);
            if(slist[curstr]>0)
                slist[newpos] = slist[curstr];
            slist[curstr] = newpos;
        }
    }
    return i+1;
}

```

(i) 用部 (c) 的例子空運行P2，並完成下表：

情況	傳入程序的值	剛離開程序前的slist[i+1]值
1	sitem = 'bus', i=4 和 nextstr=3	
2	sitem = 'come', i=5 和 nextstr=1	

(ii) P2的目的是什麼？

(d) 假定程序由頭開始運行，而用戶將下列字符串依次輸入程序：

Peter Sang  
Sue Black  
Peter Ho  
John June  
quit

(i) 完成下表以顯示執行行680前的MINV值。

執行行680前的sitem值	minv的值
Peter Sang 1	
Sue Black	
Peter Ho	
John June	

(ii) 寫出執执行程序後的輸出。

2000

Q3 小慧用C語言編寫了一電腦遊戲程序。該遊戲的規格如下：

- I 遊戲由二人對奕。
- II 遊戲利用一大小為  $n \times n$  的表格（即表格有  $n$  行和  $n$  列），其中  $n$  為 2 至 8 間（包括 2 和 8）的整數。表格內每一單元格可由所在的行數和列數參照。
- III 遊戲開始，電腦要求遊戲者輸入  $n$  後，便會建立一大小為  $n \times n$  的表格。表格內各單元格的值為介乎 1 至 9 間（包括 1 和 9）的隨機整數。例如，若玩者輸入的  $n$  值為 3，則電腦會建立如下的  $3 \times 3$  表格：

	行1	行2	行3
列1	5	6	3
列2	2	1	7
列3	1	4	9

- IV 對奕二人輪流選擇表格中的單元格。當某人選擇了一單元格後，表格的內容會根據下列規則改變：

規則A：在任何情況下，已為 0 值的單元格，其值不會改變。

規則B：將被選取單元格的值減一。

規則C：將被選取單元格同一列的其他各單元格的值減一。

規則D：將被選取單元格同一行的其他各單元格的值減一。

例子，

若玩者1選擇了列2、行1的單元格，則表格內各單元格的值將更新如下：

	行1	行2	行3
列1	4	(6)	3
列2	1	0	6
列3	0	4	9

若玩者2選擇了列1、行2的單元格，則表格內各單元格的值將更新如下：

	行1	行2	行3
列1	3	5	2
列2	1	0	6
列3	0	3	9

- V 每次輪流選擇後，若符合以下三個條件中的任何一個，則該次玩者勝出，遊戲終結。

條件1：任何一列的總和為0。

條件2：任何一行的總和為0。

條件3：任何一對角線的總和為0。

下列顯示遊戲的一個樣本輸出。

(在此輸出中，各冒號後面的數據均由用戶以鍵盤輸入，其餘則由程序輸出。)

```

Enter the size of table: 3
A 3×3 table is generated and the values of the cells are >
  5 6 3
  2 1 7
  1 4 9

Turn > Player 1
Please enter the row number: 2
Please enter the column number: 1
The values of the cells in the table are >
  4 6 3
  1 0 6
  0 4 9

Turn > Player 2
Please enter the row number: 1
Please enter the column number: 2
The values of the cells in the table are >
  3 5 2
  1 0 6
  0 3 9

Turn > Player 1
Please enter the row number: 1
Please enter the column number: 1
The values of the cells in the table are >
  2 4 1
  0 0 6
  0 3 9

Turn > Player 2
Please enter the row number: 1
Please enter the column number: 3
The values of the cells in the table are >
  1 3 0
  0 0 5
  0 3 8
Player 2 wins the game.

```

2000

小慧已編寫了程序的說明和主體部分，但遺漏了若干過程。部分程序如下所列：

1	int square[8][8], finish; int turn, row_chosen, column_chosen, n;
---	--

81	<pre> main(){     printf("Enter the size of table: "); scanf("%d",&amp;n);     init();     printf("A %d X %d table is generated ",n,n);     printf("and the values of the cell are &gt;\n");     show_square();     turn=1;     finish=0;      while(!finish){         turn++;         printf("\nTurn &gt; Player %d\n", turn%2+1);         getRowCol();         process(row_chosen,column_chosen);         show_square();         finish=endGame();     }     printf("\nPlayer %d wins the game.\n", turn%2+1); } </pre>
----	---

標識符	描述
square	貯存表格內各小格的值的二維陣列
turn	貯存輪到那一位玩者的變量
row_chosen	貯存列號的變量
column_chosen	貯存行號的變量
n	貯存表格大小的變量

在答本題時，考生不得加入新的變量。

(a) 過程init是用來賦值隨機整數予 $n \times n$ 表格。填寫下列程序段的空格以完成init過程。

11	<pre> void init(){     int r,c;     srand(time(NULL));     for(r=0;r&lt;n;r++)         for(c=0;c&lt;n;c++)             square[r][c] = ____ } </pre>
----	---

(b) 編寫過程get\_data。取得由玩者輸入的行和列數目，不需為數據作有效性檢驗。

21	<pre> void getRowCol(){     printf("Please enter the row number: "); </pre>
----	---

	<pre>       _____ printf("Please enter the column number: ");       _____     } </pre>
--	--

(c) 編寫用來顯示表格內容的過程show。

31	<pre> void show_square(){     int r,c;     printf("\nThe values of the cells in the table are &gt;\n");     for     } </pre>
----	--

(d) 編寫用來實現遊戲規格中各規則的過程process。

41	<pre> void process(int row, int column){     int r,c; ... } </pre>
----	--

(e) (i) 編寫用來傳回某一列總和的過程sumrow。

61	<pre> int sumRow (int r){     int c, sum=0; ... } </pre>
----	--

(ii) 編寫用來傳回某一行總和的過程sumcolumn。

71	<pre> int sumColumn (int c){     int r, sum=0; ... } </pre>
----	---

(iii) 編寫用來傳回左上角至右下角對角線總和的過程sumdiagonal1。

81	<pre> int sumDiagonal1(){     int pt, sum=0; ... } </pre>
----	---

(iv) 編寫用來傳回右上角至左下角對角線總和的過程sumdiagonal2。

91	<pre> int sumDiagonal2(){     int pt, sum=0; ... } </pre>
----	---

- (f) 利用 (e) 部分的四個過程，編寫一布爾過程 `endgame`，當符合規格 V 所述的三個條件中的任何一個時，傳回值為 `true`，否則，傳回值為 `false`。

```

101 int endGame(){
      int r,c,sum,theEnd=0; ...
      for
    }

```

### 電腦科 (2001 年) 試卷一 (乙部)

6. 李先生是一位數學教師。他利用一文字檔來記錄學生在兩次測驗的分數。文字檔中，每行記錄著一名學生的數據，而檔案的行數是不固定的。每行有 3 數據欄而數據欄之間由空白字元分隔。每行的格式如下所示：

4 位數字的學生編號	測驗一分數	測驗二分數
------------	-------	-------

# 2001

下列為 1999 年及 2000 年的數據檔：(考生可假設分數為正整數而數據檔起碼包括一行。)

DATA1999.TXT			DATA2000.TXT		
5760	73	93	4731	47	86
8288	18	25	4667	52	66
7654	45	10	3267	99	72
0157	99	98	1249	17	26
3547	77	41	5738	65	24
7499	60	62			
1974	44	50			
4537	78	66			
6823	92	13			
6982	70	22			
3085	96	40			
1459	53	82			

李先生編寫了一程以計算所有學生的平均分數及找出最高平均分的學生。若有兩個或以上的學生有相同的最高平均分，則程序只會找出第一個學生。

程序須在 VDU 上顯示如下的輸出。

(在此輸出中，各問號後面的數據均由用戶以鍵盤輸入，其餘則由程序輸出。)

樣本輸出 1：

File name?	DATA1999.TXT
Overall mean score:	58.63
Highest average score:	98.5
The best student:	0157

樣本輸出 2 :

File name?	DATA2000.TXT
Overall mean score:	55.40
Highest average score:	85.5
The best student:	3267

李先生的程序如下：

行號	程序語句
10	<pre>int test1, test2, i, count; float score, maxScore, sum; char num[5], maxNum[5], fn[20]; FILE *fp;</pre>
20	<pre>main(){     printf("File name (ce01q6.txt)? "); gets(fn);     if(strlen(fn)==0) strcpy(fn,"ce01q6.txt");     if((fp=fopen(fn,"r"))==NULL){         printf("Error - unable to open source file!\n");         system("pause");         return -1;     } }</pre>
30	<pre>strcpy(maxNum,"0000"); maxScore = -1; count = sum = 0;</pre>
40	<pre>while(!feof(fp)){     fscanf(fp, "%4s %d %d", num, &amp;test1, &amp;test2);     score = (test1+test2)/2;     if(score&gt;maxScore){         maxScore = score;         strcpy(maxNum,num);     }     sum += score;     count++; }</pre>
50	<pre>fclose(fp); printf("Overall mean score: %5.2f\n", sum/count); printf("Highest average score: %.0f\n", maxScore); printf("The best student: %s\n", maxNum); }</pre>

7. 下列演算法將十進數轉換成二進數。

步驟 1: 說明 `dena`、`remdr` 為整數變量，而 `newnum`、`bina` 為字符串變量。

步驟 2: 讀入十進數至變量 `dena`。

步驟 3: 賦值「空字符串」予變量 `bina`。

步驟 4: 只要 `dena` 的數值大於 0，重複步驟 5 至 8：

步驟 5: 賦值 `dena mod 2` 予變量 `remdr`。

步驟 6: 賦值 `dena div 2` 予變量 `dena`。

步驟 7: 將變量 `remdr` 轉換成字符串，並貯存在變量 `newnum`。

步驟 8: 合併變量 `newnum` 和 `bina`，並將結果貯存在變量 `bina`。

步驟 9: 顯示變量 `bina` 所貯存的值。

將以上演算法轉換成一 C 程序。

2001

8. 全程陣列變量 `list` 說明和過程 P 如下所示：

2	<code>int list[16];</code>
11	<code>int P (int t, int f, int e){</code>
12	<code>int m, flag=0;</code>
15	<code>while ( !flag &amp;&amp; f&lt;=e ){</code>
17	<code>    m = (f+e) /2;</code>
18	<code>    printf ("Before the Segment X, ");</code>
19	<code>    printf ("f =%i, e=%i, m = %i\n", f,e,m);</code>
20	
21	<code>        if (list[m] == t)</code>
22	<code>            flag = 1;</code>
23	<code>        else{</code>
24	<code>            if (list[m] &gt; t) e = m-1;</code>
26	<code>            else f = m+1;</code>
	<code>        }</code>
28	
29	<code>    printf ("After the Segment X, ");</code>
30	<code>    printf ("f =%i, e=%i, m = %i\n", f,e,m);</code>
31	<code>    }</code>
32	<code>    if (flag==1) return m; else return -1;</code>
33	<code>}</code>

(a) 假設陣列變量 `list` 貯存下列數值：

m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
list [m]	10	14	17	24	33	36	39	48	56	61	69	77	82	86	87	99

變量 `SearchResult` 為一整數。若執行下列語句：

`P (36, 0, 15, SearchResult);`

試完成下列程序的輸出：

Before the Segment X,	f =	e =	m =
After the Segment X,	f =	e =	m =
Before the Segment X,	f =	e =	m =
After the Segment X,	f =	e =	m =
Before the Segment X,	f =	e =	m =
After the Segment X,	f =	e =	m =

(4分)

(b) 假設陣列 list 的數值反向排序如下：

m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
list[m]	99	87	86	82	77	69	61	56	48	39	36	33	24	17	14	10

(i) 重寫程序段 X，使過程 P 能處理上述陣列。

2001

(ii) 執行下列語句後：

P (41, 0, 15, SearchResult);  
SearchResult 的值為何？

(c) 過程 P 所用的是什麼演算法？

9. 某選舉的候選人數目不超過30,000，用1至30,000編號表示。投票時，每一投票者只能選擇一名候選人，把該候選人的編號寫在選票上。

小玲欲編寫一 C 程序來執行下述作業：

作業 A:	輸入候選人數目並初始化各候選人獲得的選票數目。
作業 B:	透過詢問選票上候選人的編號以數算各候選人所得的選票。若選票上只得一個編號，該編號將被輸入電腦。多於一個編號的選票作廢，資料不會被輸入電腦。當用戶輸入'0'作為候選人編號時，程序將停止數算。
作業 C:	顯示選舉結果，格式如下：  <div style="text-align: center;"> <b>Result</b>  Candidate 1   *****  Candidate 2   *****  .....  在圖表中，一顆星號 (*) 代表一張選票。 </div>

下列顯示程序的一個樣本輸出。

(在此輸出中，各冒號後面的數據均由用肩以鍵盤輸入，其餘則由程序輸出。)

Number of candidates: 3	Candidate # (0 to stop): 1
Candidate # (0 to stop): 3	Candidate # (0 to stop): 1
Candidate # (0 to stop): 1	Candidate # (0 to stop): 0
Candidate # (0 to stop): 3	
Candidate # (0 to stop): 3	Result
Candidate # (0 to stop): 4	Candidate 1   ***
Candidate # (0 to stop): 68	Candidate 2
Candidate # (0 to stop): 3	Candidate 3   ****

小玲已寫好了說明和程序的主體部分，但還未完成過程。部分程序如下：

1	<code>int vote[3000];</code>	2001
10	<code>int init(){     int i, n;     printf("Election\nNumber of candidates: "); }</code>	
20	<code>void countBallots(int n){ // 投票及數選票     int ballot;     do {         printf("Candidate# 1-%d (0 to stop) : ", n);     } while ____ }</code>	
30	<code>void showChart(int n){ // 顯示****圖表     int i,j;     printf("Result\n=====\n"); }</code>	
91	<code>main(){     int n; n=init();     countBallots(n);     showChart(n); }</code>	

下面為程序所需應用的標識符及其描述：

標識符	描述
vote	用來貯存各候選人獲得的選票數目的陣列
n	貯存候選人數目的變量

在答本題時，考生不得加入新的變量。

- 完成下列過程的參數，並寫出對應作業 A 的 Init 過程。
- 寫出對應作業 B 的 CountBallots 過程。
- 寫出對應作業 C 的 ShowChart 過程。

## 電腦科 2002 試卷一乙

6 李先生欲編寫一程序來模擬文字處理軟件包的「尋找及取代」功能。[2002]  
有一文字檔「CE.txt」存貯在磁碟內，其內容如下：

Hong Kong Examinations Authority*	2002
Hong Kong Certificate Of Education Examination 2002*	
Computer Studies*	
Paper 1B and 1C*	

每行的結尾都有一星號 (\*)。

該程序將文本逐行讀入，用冒號 (:) 取代所有空白字符，並在 VDU 顯示如下的輸出。

Hong:Kong:Examinations:Authority*
Hong:Kong:Certificate:Of:Education:Examination:2002*
Computer:Studies*
Paper:1B:and:1C*

Number of replacements: <input type="text" value="13"/>
---

```
const char searchChar=' ', replaceChar=': ';
main() {
    FILE *fp;
    char s[100];
    int i, totalReplaced = 0;
}
```

7 下列是一程序的算法，可產生一正整數序列。該等整數必須小於或等於整數變量 limit 的值，而該值是在程序執行時由用戶以鍵盤輸入的。[2002]

步驟 1：說明 former、latter、limit 和 cnt 為整數變量。

步驟 2：讀入一整數至變量 limit。

步驟 3：賦值 1 予變量 former。

步驟 4：賦值 1 予變量 latter。

步驟 5：賦值 0 予變量 cnt。

步驟 6：只要 former 的值小於或等於 limit 的值，重複步驟 7 至 11。

步驟 7：顯示 former 的值，並保持游標在同一行。

步驟 8：將 latter 的值加上 former 的值，並將結果賦值予變量 latter。

步驟 9：將 latter 的值減去 former 的值，並將結果賦值予變量 former。

步驟 10：把 cnt 的值加 1。

步驟 11：若 cnt 的值能為 6 所整除，則將游標下移一行。

試將算法改寫成 C 程序。

--

8 下列程序包括一將英文字編碼的過程 Encode 及其相應解碼的過程 Decode。這解碼過程可還原由 Encode 編碼的英文字，而當中有某些程序語句仍未完成。[2002]

const int M=5,D=21,N=26,A=65; char ENG[50], SEC[50];	(a) (ii)
void Encode(char s[]){ int len=strlen(s), i, P; for(i=0;i<len;i++){ P = (M*(s[i]-A))%N; s[i] = P+A; } }	void Decode( ){\br/>int len=strlen(s), i, C; }

(a) (i) 若把'ABLE'存貯在字串變量ENG，再執行Encode(ENG)；ENG會變成\_\_

(ii) 程序中編碼及解碼公式分別為  $P = (5 \times C) \bmod 26$  和  $C = (21 \times P) \bmod 26$ ，其中 P和C 均為由 0到25 的整數。試利用這兩條公式，完成 Decode 過程，把由 Encode 過程產生及存貯在變量 SEC 的字串解碼。

```
main(){
    printf("Please type an English word: ");
    gets(ENG); Encode(ENG);
    printf("The resulting secret word is: %s\n", ENG);
    printf("Please type a secret word: ");
    gets(SEC); Decode(ENG);
    printf("The original English word is: %s\n", ENG);
}
```

(iii) 若將上述 Decode 過程應用在上述 (a)(i) 變量 SEC 存貯的字串，而解碼公式改寫成  $C = (20 \times P) \bmod 26$ ，寫出變量 ENG 的輸出。

(b) 若此程序現只需要處理小寫字母 (a .. z)，其中哪一語句需要作出修改？

Q9 小克編寫一 C 程序來檢查某個英文字是否另一個英文字的變位字 (anagram)。

所謂「變位字」，是指一個英文字在字母次序改變後而形成另一個英文字，

則這英文字就是另一個英文字的變位字 (anagram)。例如，

(1) ITEM 是 TIME 的變位字 (2) EXAM 不是 TIME 的變位字

2002

假設所有輸入皆為正確英文字，並只由英文字母組成。小克將編寫一C程序來完成下列工作：

工作 I.	初始化兩陣列以便存貯兩字中各英文字母的頻率。	
工作 II.	輸入兩英文字。	InputWords()
工作 III.	檢查兩字字母數目是否相同。	CheckLength()
工作 IV.	將字中所有字母轉成大寫。	ChangeUpper()
工作 V.	計算兩字中各字母的頻率。	CalFreq()
工作 VI.	檢查第二個英文字是否第一個英文字的變位字。	CheckMatching()
工作 VII.	顯示檢查結果。	

下列為程序的樣本輸出。

樣本輸出1 Word1: <u>TIME</u> Word2: <u>ITEM</u> ITEM is an anagram of TIME	樣本輸出2 Word1: <u>time</u> Word2: <u>ITEM</u> ITEM is an anagram of TIME
樣本輸出3 Word1: <u>TIME</u> Word2: <u>EXAM</u> EXAM is <b>not</b> an anagram of TIME	

小克已寫好程序的說明及主體部分，但有些過程仍未完成。部分程序如下所示：

```

#define MAX 126
int freq1[MAX], freq2[MAX];
char word1[50], word2[50];

void initFreq(){
    char c;
    for(c=0;c<MAX;c++)
        freq1[c]=freq2[c]=0;
}

main(){
    int match=1;
    do{
        initFreq(); inputWords();
        match=checkLength();
        if(match){
            changeUpper(word1);
            changeUpper(word2);
            calFreq();
            match=checkMatching();
        }
        if(match) printf("anagram\n");
    }while(1);
}

```

程序中使用到的標識符及其描述如下：

標識符	描述
freq1, freq2	用來存貯第一、二個英文字各字母頻率的陣列
word1, word2	用來存貯第一、二個英文字的變量
c	一字元char變量
i	用作計數器的整數變量
match	一布爾值變量 <b>Boolean(True/False)</b>
st	用作 ChangeUpper 過程中的字符串變量參數

- 編寫 InputWords() 過程，以執行工作 II。
- 編寫 CheckLength() 過程，以執行工作 III。
- 填寫下列過程的參數，並編寫 ChangeUpper(...) 過程，以執行工作 IV。
- 編寫 CalFreq() 過程，以執行工作V。
- 編寫 CheckMatching() 過程，以執行工作 VI。

## 電腦科 2003 試卷一乙

6. 陳先生編寫了一個程序來讀取一個十進制數，[2003]  
並把它轉換成以二進制補碼表示的8位二進制數。該程序須在VDU上顯示如下的輸出。

樣本輸出 1

```
Please enter a number: 31
In 8-bit 2's complement, 31 is 00011111
```

樣本輸出 2

```
Please enter a number: -31
In 8-bit 2's complement, -31 is 11100001
```

樣本輸出 3

```
Please enter a number: -128
In 8-bit 2's complement, -128 is 10000000
```

2003

陳先生的程序如下所示:

```
main() {
    int num, i, n;
    char bin[9]="00001111";
    do {
        printf("Please enter a number (-128..+127): "); ...
        printf("In 8-bit 2's complement, % 3d is %s\n", num,bin);
    } while(num!=0);
}
```

2's complement

7. 下列程序是用來為圖書證編號進行有效性檢驗。圖書證編號最右的一個數字是檢查數位

行號	程序語句
110	const int card_len=8;
120	
140	void Read_string(char x[]){
150	int flag,i;
180	do{
190	flag=1;
200	printf("Input card number: ");
210	gets(x);
220	if(strlen(x)!=card_len){
240	flag=0;
250	printf("Incorrect no. of digits!\n");
260	}
270	}while(flag==0);
280	}

2003

```

290 void test(){
      int i, sum=0, n;
      char inp[80];
300   Read_string(inp);
310
320   for(i=0;i<card_len;i++)
330       if(i%2==1){
350           n = (inp[i]-'0')*2;
360           sum += n%10 + n/10;
380       }else{
400           n = inp[i]-'0';
410           sum += n;
      }

430   if(sum%10==0)
440       printf("valid card number 合法\n");
450   else printf("invalid card number 不合法\n");
460 }

470 main(){
480     while(1) test();
490 }

```

作答本題時，考生不得加入新的變量，否則不獲給分。

(a) 檢查數位有什麼功用？

(b) 當程序執行時，用戶輸入 '23731654'

(i) 試完成下表

當執行 FOR(320) 循環後 i 的值	n	sum
1		
2		
3		
4		

(ii) 當程序執行後，

(1) sum 的值是

(2) 程序的輸出是

(c) 若想對圖書證編號進行範圍檢查，過程 Read\_string 須作修改。編號只能包括 '0' 至 '9' 的數字。在行號 210 和 220 之間加入程序語句，使能進行範圍檢查。

```

211     for (i=0;

```

程序經修改後，會輸出一部分樣本如下：

```
Input card number: abc18698315gc
Digits from "0" to "9" only!
Incorrect no. of digits!
Input card number: 1234...
```

(d) 現重新編寫程序，以便為每圖書證編號產生檢查數位。

假設行號 430–450 被刪除了，而行號 110 被修改為：

```
const int card_len = 7;
```

請在行號 430 插入一程序語句，以便在輸入一個 7 位數字的圖書證編號後，程序能產生及輸出一檢查數位。

330	
-----	--

下列為程序的樣本輸出：

```
Input card number: 9843512
Check digit is 6
```

8. 下列算法顯示某兩數間的所有質數，首尾兩數也包括在內：[2003] 步驟

2003

- 1 說明 NUM, COUNT, LOWER 和 UPPER 為整數變量。
- 2 讀入兩個整數至變量 LOWER 和變量 UPPER，分別代表下限和上限
- 3 賦變量 LOWER 的值予變量 NUM。
- 4 當變量 NUM 的值少於或等於變量 UPPER 的值，執行步驟 5 至 9
- 5 賦值 2 予變量 COUNT。
- 6 當變量 NUM 的值不能為變量 COUNT 的值整除，執行步驟 7
- 7 將變量 COUNT 的值加 1。
- 8 若變量 NUM 的值等於變量 COUNT 的值，顯示變量 NUM 的值
- 9 時變量 NUM 的值加 1。

試將算法改寫成 C 程序

```
main() {
    int num, divisor, lower, upper;
    printf("Q: Lower(>1) & Upper limits : ");
    scanf ("%d%d", &lower, &upper); ...
}
```

9. 大衛欲編寫一 C 程序來玩猜字遊戲。程序首先會在一文字檔中隨機選出一英文字，該文字檔包含一些由小楷字母組成的英文字，然後提示玩者猜測組成該英文字的字母。玩者若能在一定的次數內正確猜中所有組成該英文字的字母，便可勝出。該程序會處理下列工作：

工作 A	從文字檔中讀取所有英文字， 並從中隨機選出一英文字作為遊戲的「秘密字」(secret)。
工作 B	初始化 (1)變量(solution)為一串「*」符號，其長度與秘密字相同，及 (2)變量(noOfAttempts)為剩餘的嘗試次數，其值是 secret 字長的兩倍
工作 C	每一回合要求玩者輸入他所猜測的字母(letter)。 若該字母為秘密字的字母， 則該字母將取代在字符串變量 solution 中對應位置的'*'符號， 而每一回合的猜測結果及剩餘的嘗試次數，將會在屏幕上顯示。
工作 D	當玩者猜測出「所有」組成秘密字的字母，(secret=solution) 或當剩餘的嘗試次數(noOfAttempts)等於 0，則遊戲結束。
工作 E	顯示結果：玩者勝利或失敗的

執行畫面：該程序須在 VDU 上顯示如下的輸出：

```
The secret word to be guessed is *****
No. of Attempts remaining = 10

Please input the letter you guess: e
****e

No. of Attempts remaining = 9
Please input the letter you guess: A
****e

No. of Attempts remaining = 8
Please input the letter you guess: a
a***e

No. of Attempts remaining = 7
Please input the letter you guess: p
app*e

No. of Attempts remaining = 6
Please input the letter you guess: l
apple

No. of Attempts remaining = 5
You win!!
```

solution

noOfAttempts  
每次減 1

letter

hangman

大衛已編寫好程序的說明及主體部分，但有些過程還未完成。部分程序如下所示  
CE1993-2003-2A-prog.docx

```

main(){
    if(!selectSecret(secret)) return 1;
    init();
    printf("The secret word to be guessed is %s\n", solution);
    printf("No. of Attempts remaining = %d\n", noOfAttempts);
    do{
        letterGuess();
        done=winCheck();
    }while(done==0);
    printf("You %s!!\n", strcmp(solution,secret)?"lose":"win");
    printf("Answer: %s\n",secret);
}

```

2003

下表顯示該程序中每一標識符的描述:

標識符	變量	描述	例子
secret	字符串	存貯遊戲的秘密字	"apple"
solution	字符串	存貯猜測過程中的臨時結果	"-----", "-pp--"
done	布爾	1=遊戲結束, 0=未結束	0 或 1
letter	字元	存貯玩者輸入的字母	'p'
i	整數	計數器	
noOfAttempts	整數	存貯剩餘的嘗試次數	10

該文字檔包含小楷字母組成的英文字，存貯不多於 50 個英文字，其部分內容如下:

```

apple
boy
dog
cat
...

```

作答本題時，考生不得加入新的變量，否則不獲給分。

- (a)完成 selectSecret 過程， 以執行工作 A
- (b)編寫 init 過程， 以執行工作 B
- (c)編寫 letterGuess 過程， 以執行工作 C
- (d)編寫 winCheck 過程， 以執行工作 D

```

char secret[20], solution[20]; // "apple", "-----"
int done, i, noOfAttempts;
char letter, word[50][20]; // word list without hints

```

```

int selectSecret (char secret[]){
    FILE *fp;
    int NoOfword,i=0;
    fp=fopen("words.txt","r");

```

```

void init(){

```

```

void letterGuess(){

```

```

int winCheck(){

```