



C 語言

重複結構與判斷式



本單元主題

- 重複結構與判斷式應用的例子
- if...break
- if...continue
- 程式的結束條件EOF
- switch範例說明

練習

請撰寫一程式，輸入一個10人的班級學生的電腦成績，然後輸出及格與不及格人數。



while-if字句

```
#include <stdio.h>
int main(){
    int counter=1,grade,pass=0,nopass=0;
    while(counter<=10){
        scanf("%d",&grade);
        if (grade>=60)
            pass=pass+1;
        else
            nopass=nopass+1;
        counter=counter+1;
    }
    printf("及格%d人\n",pass);
    printf("不及格%d人\n",nopass);
    return 0;
}
```

在重複結構中，挑出符合判斷式的條件

改變程式的控制流程

break

- 與while, for, do while 或switch敘述內執行
- 會使得程式馬上開那個敘述式

continue

- 與while, for 或 do while 一起使用
- 迴圈條件會在continue之後馬上檢驗

break

- 有時為了程式執行效能，希望迴圈提早結束
- 當程式執行到break時，跳出迴圈，直接執行迴圈之後的敘述
- 與判斷式if搭配使用

可以與 while/do...while/for/switch 一起使用

```
for( 初始值; 條件式; 變異量 ){  
    ...  
    if( 中斷條件 )  
        break;  
    ...  
}
```

練習

計算整數1到n的偶數和，但是遇到10的倍數就終止，印出符合條件的數。

輸入範例：

50

輸出範例：

2
4
6
8



參考程式碼-使用break

```
#include <stdio.h>

int main(){
    int i,n,m;
    scanf("%d",&n);
    for(i=2,sum=0;i<=n;i=i+2){
        if(i%10==0)
            break;
        printf("%d\n",i);
    }
    return 0;
}
```

輸入：

50

輸出：

2

4

6

8

遇到10的倍數，程式終止

練習

寫一個進程，輸入某數 N ，判斷此數是否為質數。

輸入範例：
1000

輸出範例：
1000不是質數



參考程式碼-使用break

```
#include <stdio.h>

int main(){
    int i,num,flag=0;
    scanf("%d",&num);
    for(i=2;i<num;i++){
        if(num%i==0){
            flag=1;
            break;
        }
    }
    if(flag==1)
        printf("%d不是質數\n",num);
    else
        printf("%d是質數\n",num);

    return 0;
}
```

只要發現第一個可以num整除的 i，就可以證明此數不是質數，再往下找已經沒有意義，因此，使用break終止。

continue

- 使用continue，可跳過迴圈剩下的部分，繼續下一個迴圈的條件判斷
- 為了清楚表明程式的流程，表明在某種狀態下，會回到迴圈的起點

練習

計算整數**1到n的偶數和**，但是遇到10的倍數跳過再繼續，印出符合條件的偶數與總和。



參考程式碼-使用continue

```
#include <stdio.h>

int main(){
    int i,n,sum;
    scanf("%d",&n);
    for(i=2,sum=0;i<=n;i=i+2){
        if(i%10==0)
            continue;
        printf("%d\n",i);
    }
    return 0;
}
```

輸入：

50

輸出：

2

4

6

8

12

...

44

46

48

只要發現10的倍數，跳過此條件，然後從迴圈的起點開始執行

break and continue

- 某些工程師覺得break與continue違反結構化程式設計的規範。因為這兩個敘述式的效果可以由結構化程式設計取代，因此有些工程師不使用這兩個指令
- 當正確使用break與continue，可以有效提供程式執行的效率

有一種結束重複的方式

讀到檔案末端(END OF FILE)停止

檔案末端即指結束的條件是以EOF(End of File)當成警示值，使用者根據系統的指定鍵鍵入某種按鍵的組合，來表示end of file，表示我沒有任何資料要輸入了。

代表EOF的按鍵組合是隨系統而變的，

- 在Windows中，需按下ctrl+z
- 在Unix中，需按下ctrl+d，DICE平台作業系統是Ubuntu，屬於Unix系統，因此結束條件是ctrl+d

練習

請撰寫一程式，輸入一個整數，印出其相反數與絕對值，格式如輸出範例所示，**以上的程式可以重複執行，直到輸入結束或說檔案末端(end of file)為止。**

輸入範例：

6
7
8
-6
-7
-8

輸出範例：

6的相反數-6
6的絕對值6
7的相反數-7
7的絕對值7
8的相反數-8
8的絕對值8
-6的相反數6
-6的絕對值6
-7的相反數7
-7的絕對值7
-8的相反數8
-8的絕對值8

執行到檔案末端，然後結束重複

```
int n;  
while(scanf( "%d", &n ) != EOF){  
    敘述式;  
}
```

→ 變數n可以重複輸入，直到輸入端以CTRL+D(在DICE中的end of file按鍵組合)結束輸入為止。

練習

程式讓使用者輸入分數等級，而學生們的成績等級屬於A、B、C、D與F各個等級，請計算各個等級的個數，程式可以重複執行直到檔案末端。

輸入範例:

A
A
B
b
C
c
D
d
F
f

輸出範例:

A 2人
B 2人
C 2人
D 2人
F 2人



參考程式碼說明，以A等級為例

```
#include <stdio.h>
int main()
{
    int grade,aCount=0,bCount=0,cCount=0,dCount=0,fCount=0;
    while((grade=getchar())!=EOF){
        switch(grade){
            case 'A':
            case 'a':
                ++aCount;
                break;
            .
            .
            .
            default:
                break;
        }
    }
    printf("A %d人\n",aCount);
    return 0;
}
```

■Getchar函式

來自標準輸入與輸出函式，從鍵盤中讀取一個字元，並將此字元存放到整數變數grade中

字元通常是存到char型別的變數中，C有一項重要功能，可以將字元存放成任何型態的資料型別