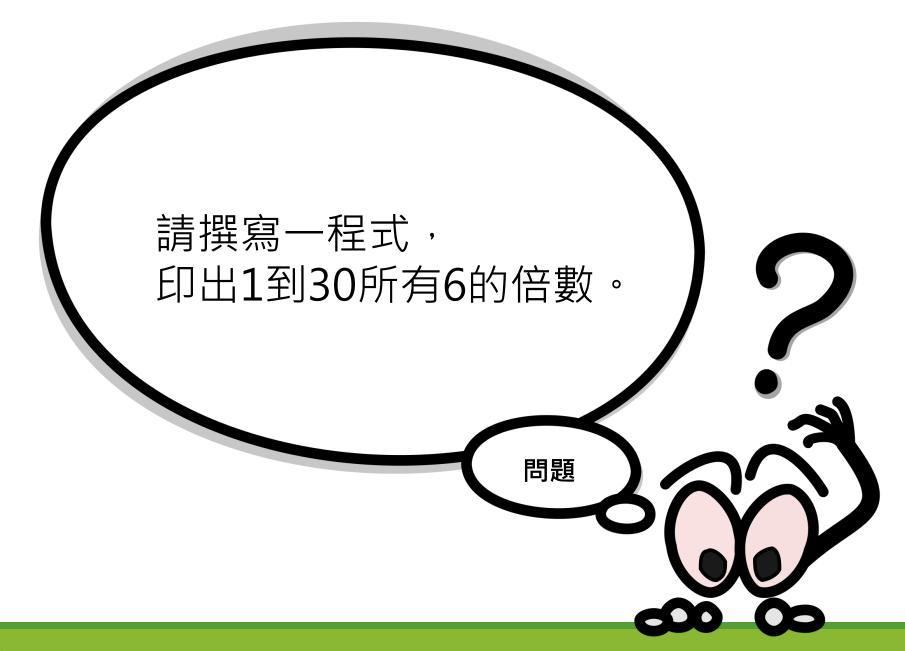


# 重複結構與判斷式



# 判斷式結構 常常會包在重複結構中



# while-if字句

```
#include<stdio.h>
int main()
 int i=6;
                            在重複結構中,
  while(i <= 30){
                            挑出符合判斷式的條件
   if(i%6==0)
      printf("%d\n",i);
    i=i+1;
  return 0;
```

022/3/16

# 另一種結束重複的方式

讀到檔案末端(END OF FILE)停止

# 問題

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。

此種重複類型的結束條件是直接手動輸入結束按鍵Ctrl+c。 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

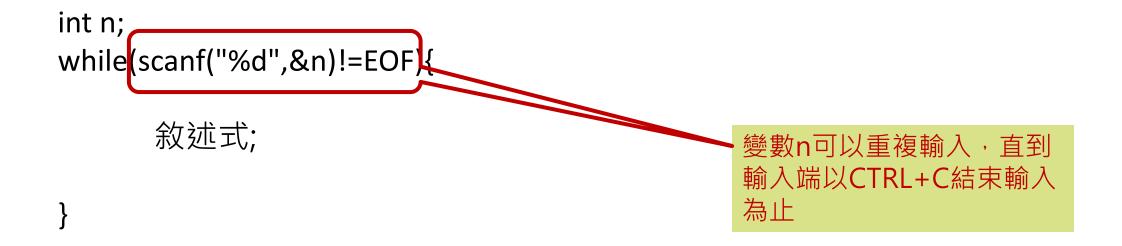
-7的相反數7

-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



2022/3/16

#### 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;
...
}
```

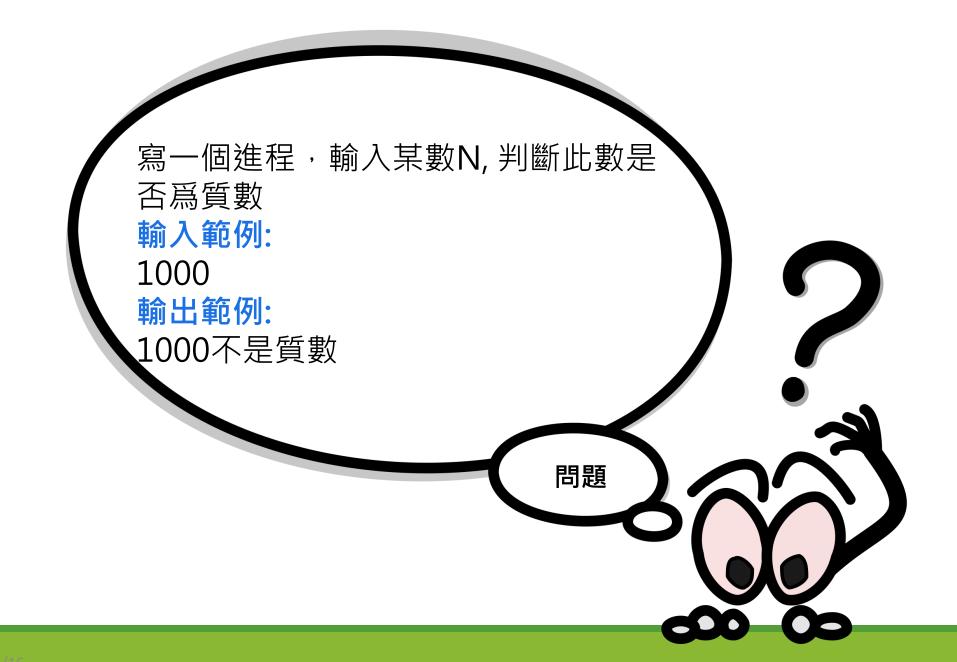


#### 參考程式碼-使用break

```
#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)
            break;
        printf("%d\n", i);
    }
    return 0;
}

遇到10的倍數,程式終止
```

12



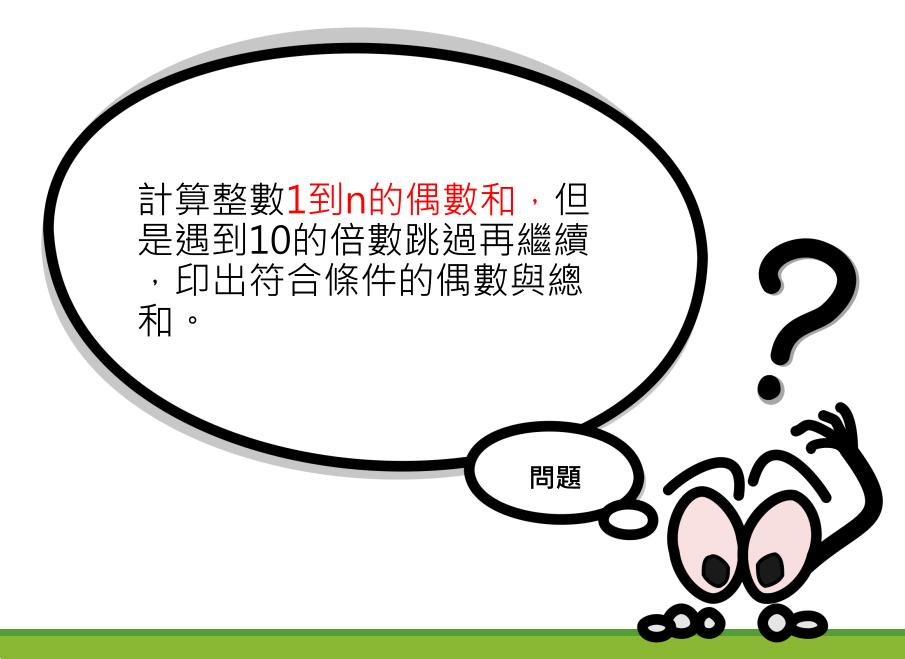
#### Break的應用

```
#include <stdio.h>
int main()
    int i, num, flag=0;
    scanf("%d", &num);
    for (i=2; i \le 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被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



# 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;
}</pre>
```

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

```
輸入:50
輸出:
6
8
12
14
16
18
22
24
26
28
32
34
36
38
42
44
46
48
```