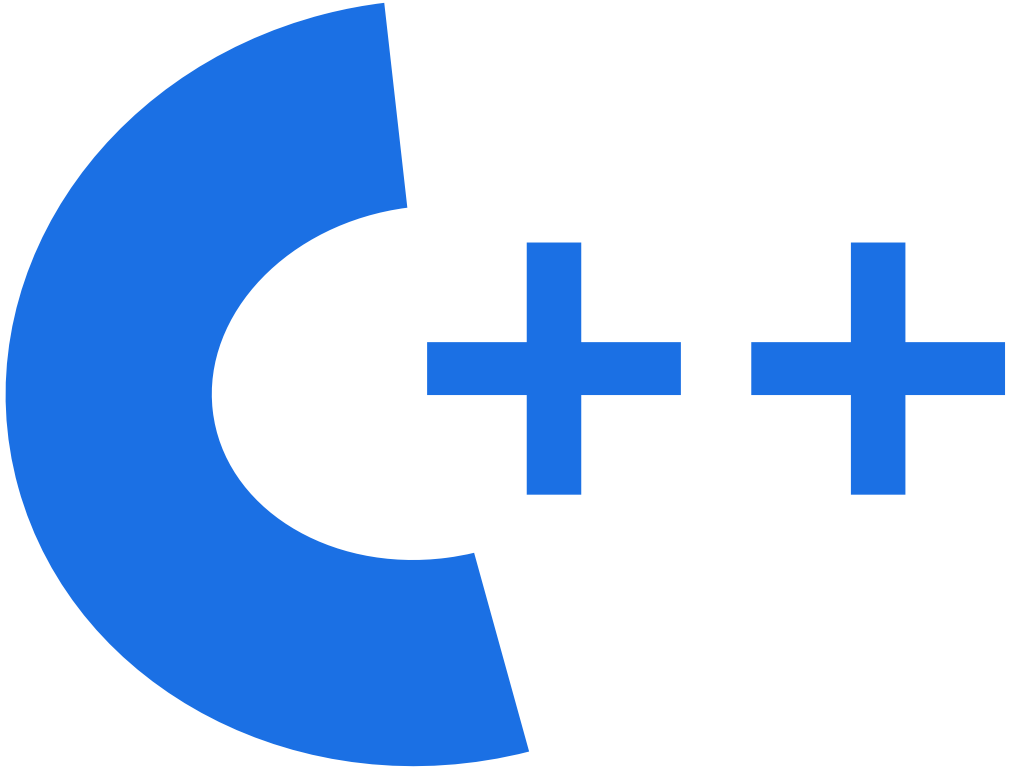
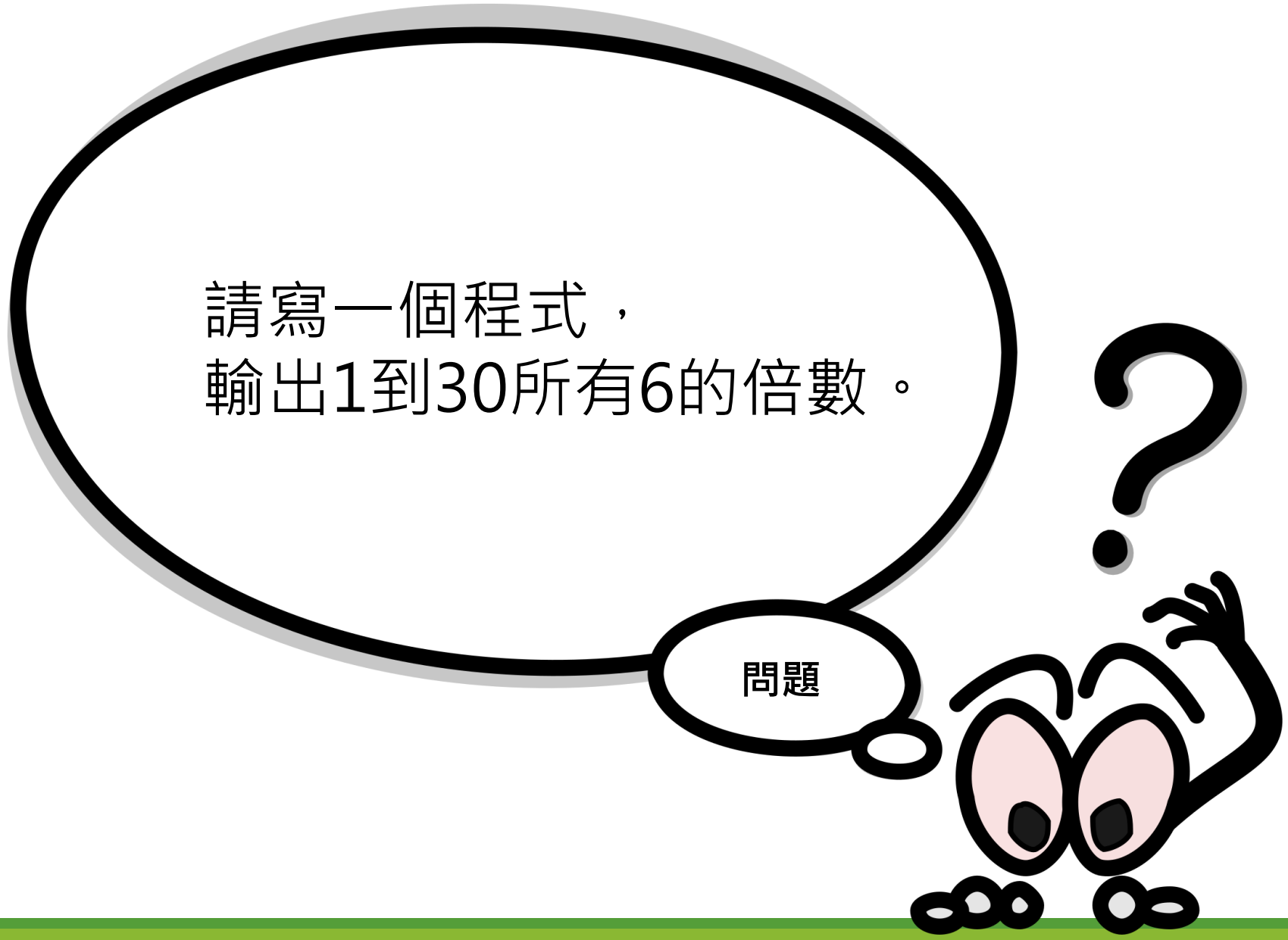


重複結構與判斷式

選擇結構
常常會包含在重複結構中

請寫一個程式，
輸出1到30所有6的倍數。



while-if字句

```
int main()
{
    int i=6;
    while(i<=30){
        if(i%6==0)
            cout<<i<<endl;
        i=i+1;
    }
    return 0;
}
```

在迴圈中，
挑出符合條件表達式的條件

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

```
while(!cin.eof()){
        cin>>n;

}
```

變數n可以重複輸入，直到輸入端以CTRL+C結束輸入為止

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;條件表達式;增量表達式){
...
    if(中斷條件)
        break;
...
}
```

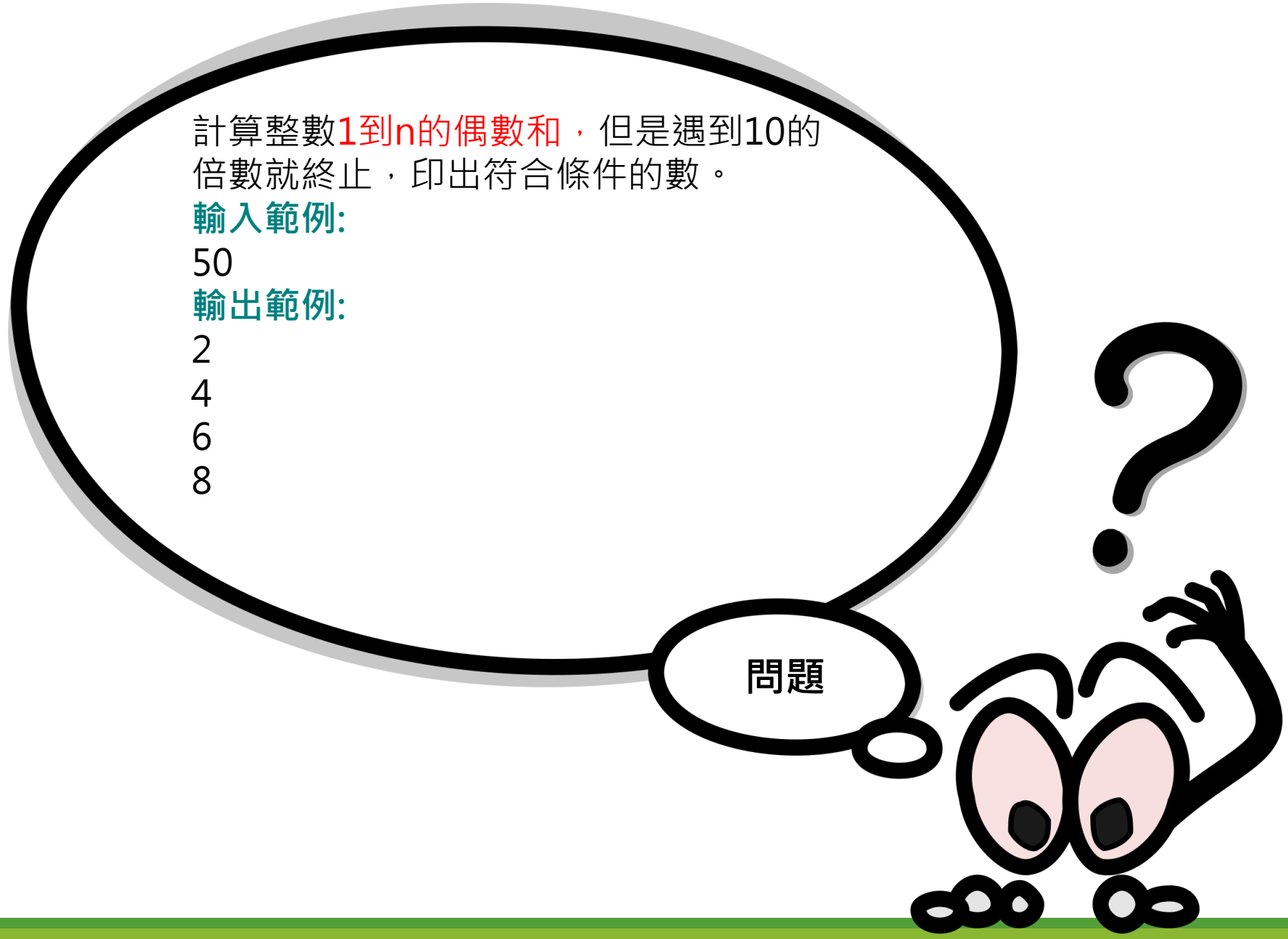
計算整數1到n的偶數和，但是遇到10的倍數就終止，印出符合條件的數。

輸入範例:

50

輸出範例:

2
4
6
8



break的範例程式

```
#include <iostream>
using namespace std;
int main()
{
    int i,n,sum;
    cin>>n;
    for(i=2,sum=0;i<=n;i=i+2){
        if (i % 10==0)
            break;
        cout<<i<<endl;
    }
    return 0;
}
```

```
輸入：50
輸出：
2
4
6
8
```

遇到10的倍數，程式終止

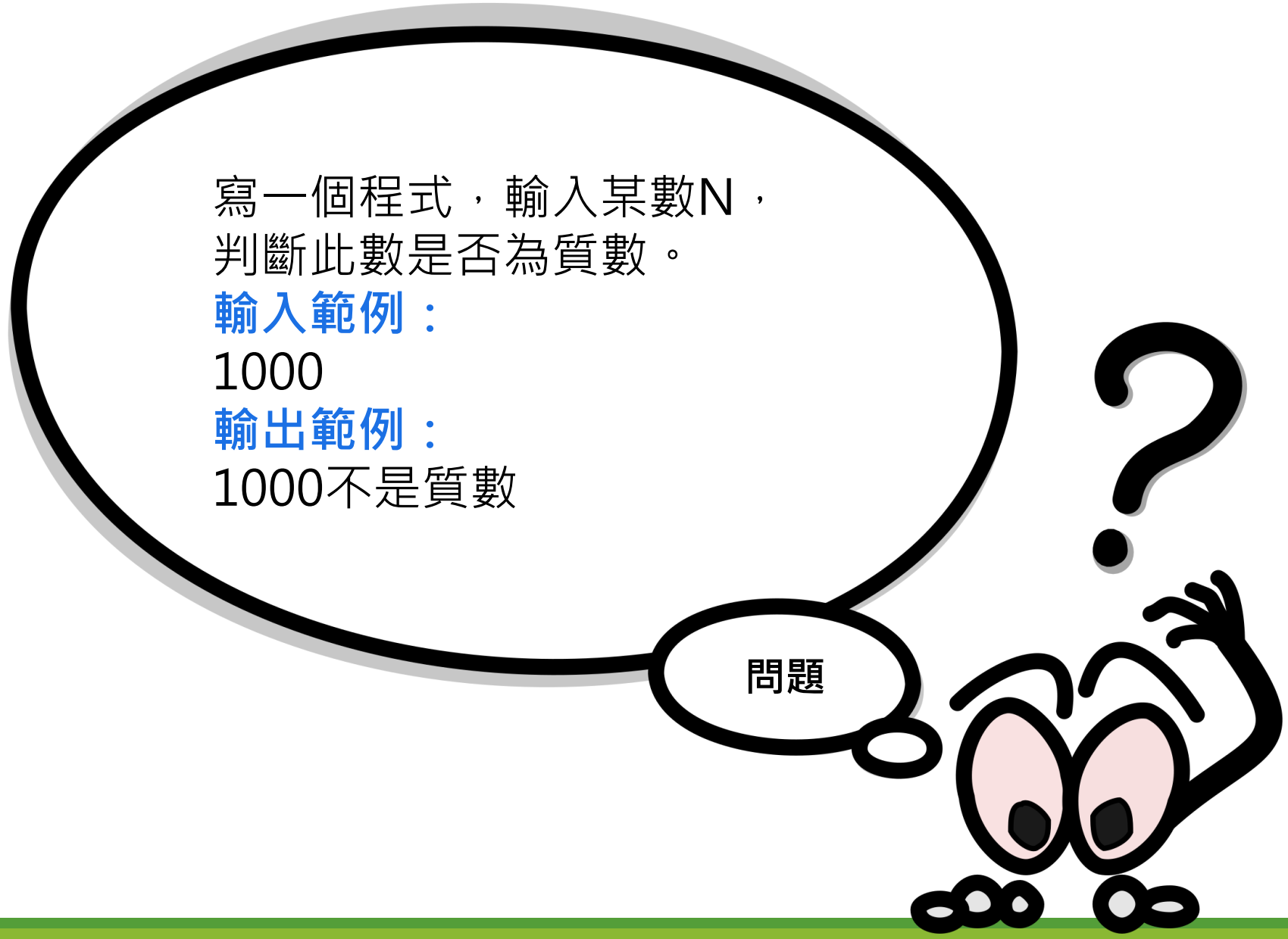
寫一個程式，輸入某數N，
判斷此數是否為質數。

輸入範例：

1000

輸出範例：

1000不是質數



Break的應用

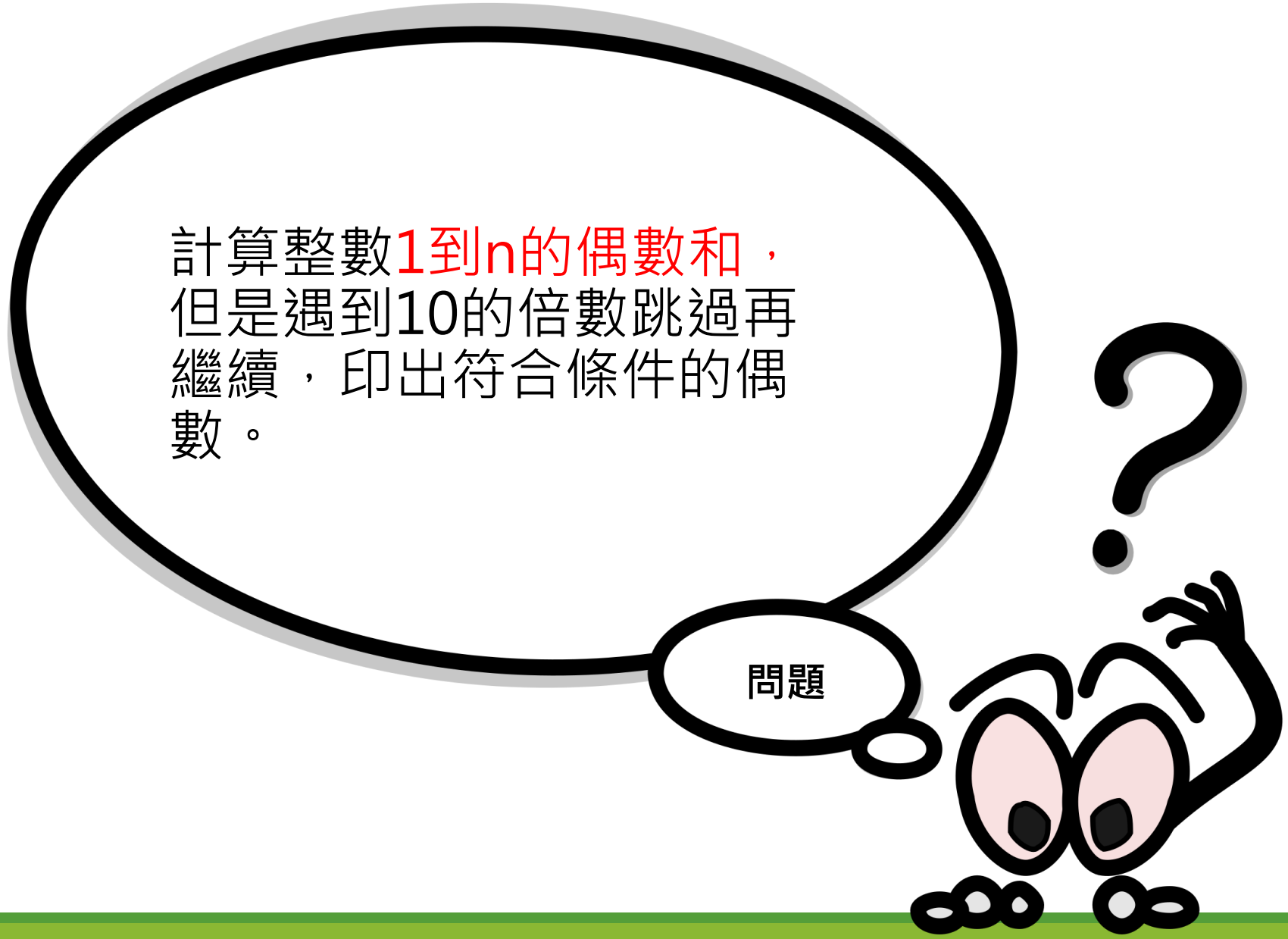
```
int main()
{
    int i,j=0,n;
    cin>>n;
    for(i=1;i<n;i++){
        if(n%i==0){
            j=1;
            break;
        }
    }
    if(j==0)
        cout<<n<<"是質數"<<endl;
    else
        cout<<n<<"不是質數"<<endl;

    return 0;
}
```

只要發現第一個可以整除n的i，
就可以證明此數不是質數，
再往下找已經沒有意義。

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

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。



參考程式碼-使用continue

```cpp
#include <iostream>
using namespace std;
int main()
{
    int i,n,sum;
    cin>>n;
    for(i=2,sum=0;i<=n;i=i+2){
        if (i % 10==0)
            continue;
        cout<<i<<endl;
    }
    return 0;
}
```

遇到10的倍數，跳過，迴圈從頭開始

```
輸入：50
輸出：
2
4
6
8
12
14
16
18
22
24
26
28
32
34
36
38
42
44
46
48
```