6迴圈

6-1　while迴圈
6-2　for迴圈
6-3　break與continue
6-4　迴圈設計的方法
6-5　多重迴圈

迴圈（loop）

- 迴圈用來控制特定區塊內敘述的重複執行。
- Python 提供二種迴圈敘述型態：while 迴圈與for 迴圈。

6-1 while 迴圈

- while迴圈在條件為真時，重複執行敘述內容。
- while迴圈的語法如下：
  - while condition:
  - statement(s)

- 迴圈內要被重複執行的敘述（statements）被稱作迴圈主體（loop body）。
- 迴圈主體的一次性執行被稱作迴圈的迭代（iteration）或重複（repetition）。
- 迴圈含有condition，為一個控制主體敘述執行的布林運算式。

題目

- 若要印出100次的
  - Learning Python is fun!

範例程式：p6-4.py

```
count = 1
while count <= 100:
    print('Learning Python is fun!')
    count = count +1
```

- while迴圈敘述要注意的地方有二：
  - 1. while count <= 100 後面有一冒號（:），它表示迴圈從以下的敘述組成迴圈主體的敘述。若忘記加冒號將會產生語法錯誤的訊息。
  - 2. 迴圈主體的敘述必須內縮，並且加以對齊。凡是要一起處理的複合敘述（compound statements）必須要這樣撰寫，切記切記。

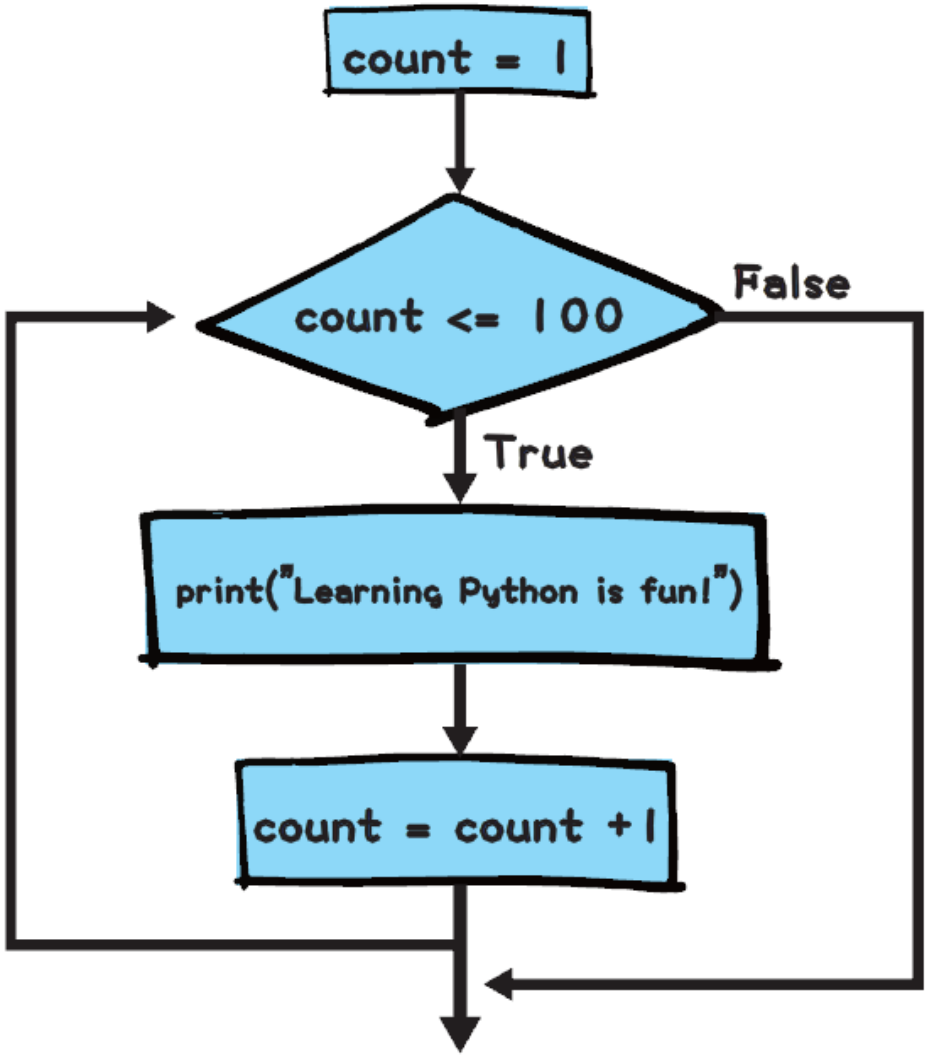


圖6-1　當count<=100為True時，while迴圈會重複執行迴圈主體敘述

題目

- 計算1 加到100的總和

錯誤範例

- 錯誤範例

```
total = 0
k = 1
while k <= 100
    total = total + k
k = k +1
print('1 + 2 + 3 + ... + 100 = ', total)
```

找找錯誤在哪兒

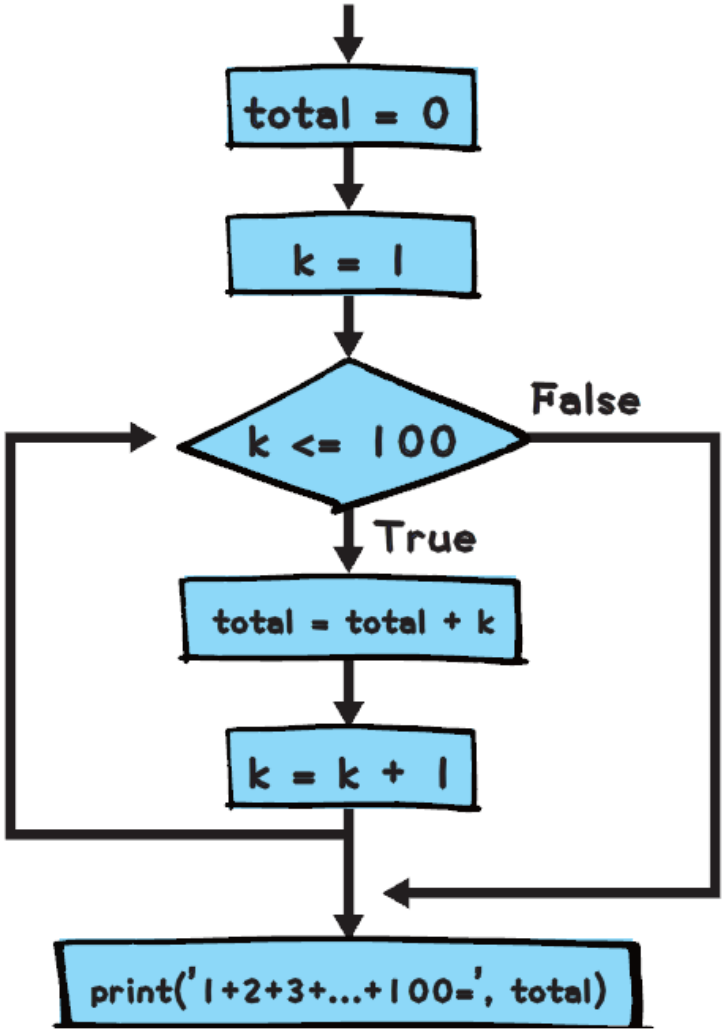
正確範例：p6-6.py

- total = 0
- k = 1
- while k <= 100:
- 　　total = total + k
- 　　k = k +1
- print('1 + 2 + 3 + ... + 100 = ', total)

在while 的條件運算式後加冒號，還有要將
k = k + 1
和
total = total + k
對齊。

輸出結果

```
1 + 2 + 3 +...+ 100 = 5050
```



● 圖6-2　1加到100的流程圖

撰寫迴圈注意事項

- 1.初始值
- 2.迴圈何時結束
- 3.每次迴圈的增量

練習題p6-8

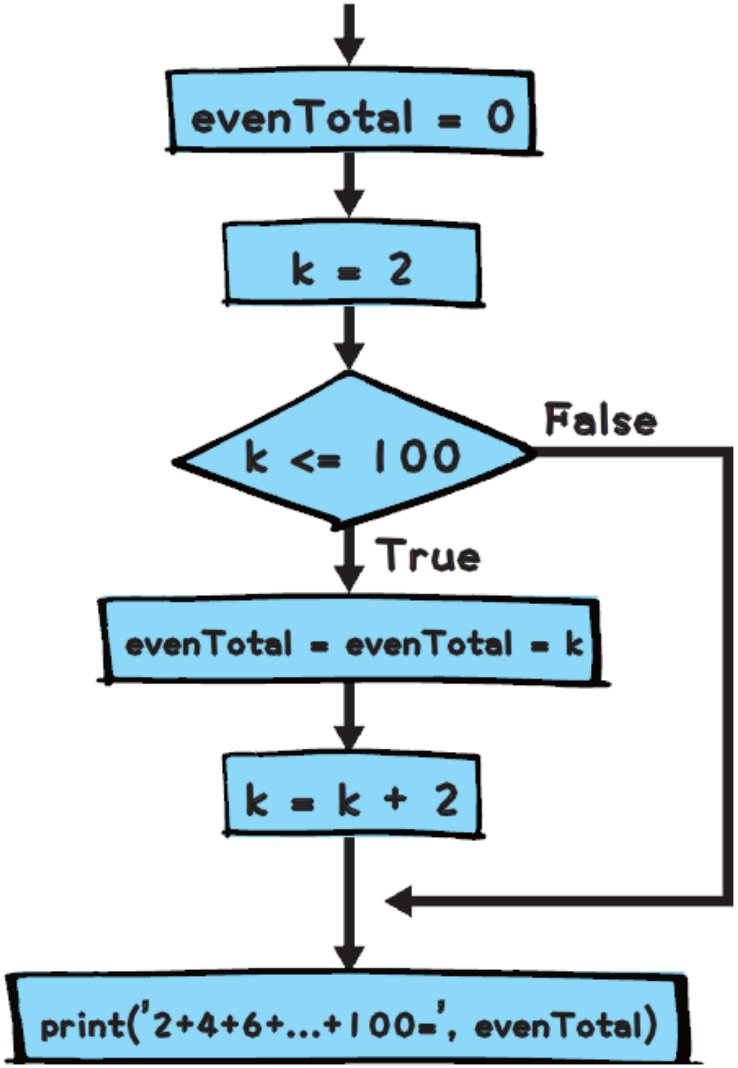
- 撰寫從1 加到100 的偶數和

範例程式：p6-8.py

```
evenTotal = 0
k = 2
while k <= 100:
    evenTotal = evenTotal + k
    k = k + 2
print('2 + 4 + 6 + ...  +100 = ', evenTotal)
```

輸出結果

2 + 4 + 6 +... + 100 = 2550



● 圖6-3　1加100的偶數和之流程圖

6-2 for 迴圈

- 基本上while迴圈和for迴圈是可以互用的。例如，while迴圈如下：

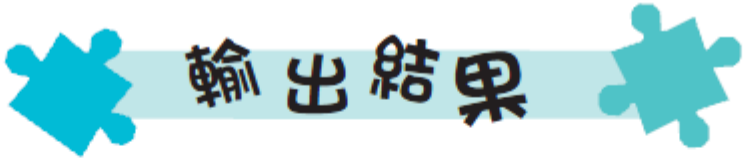
```
k = initialValue:
while k < endValue:
        # 迴圈主體的敘述
    …
        k += 1
```

將它轉換爲for迴圈，如下所示：

```
for k in range(initialValue, endValue):
        # 迴圈主體的敘述
    …
```

- for迴圈敘述和while迴圈敘述類似，需注意的地方有二：
  - 1. for k in range(initialValue, endValue)後面有一冒號（:），它表示迴圈從以下的敘述組成迴圈主體的敘述。若忘記加上冒號，將會產生語法錯誤的訊息。
  - 2. 迴圈主體的敘述必須內縮，並且要加以對齊。這和while迴圈敘述是一樣的，沒有對齊的敘述不會一起執行。

```
>>> for k in range(1, 6):
        print(k)
```




```
1
2
3
4
5
```

此迴圈的initialValue是1，而endValue是6，所以表示從1到5加以印出。
切記，只印到endValue－1。

- 此迴圈的initialValue是1，而endValue是6，所以表示從1到5加以印出。
- 切記，只印到endValue – 1。

range 函式

- range 函式的語法有三種，分別是 1 個、2 個或 3 個參數。1 個參數的語法為：

```
串列變數 = range( 整數值 )
```

- range 函式包含2 個參數的語法為：

```
串列變數 = range( 起始值 , 終止值 )
```

- 起始值及終止值皆可為負整數，例如：

```
list3 = range(-6, -2)   #list3=[-6,-5,-4,-3]
```

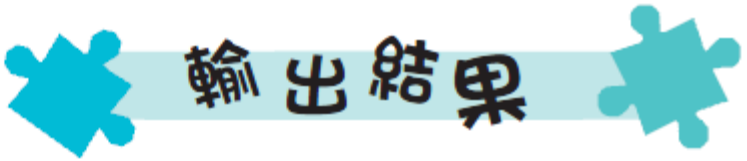
- 產生的串列是由起始值開始，每次會遞增間隔值，直到「終止值 – 1」為止的串列，例如：

```
list4 = range(3, 8, 1)   #list4=[3,4,5,6,7]
list5 = range(3, 8, 2)   #list5=[3,5,7] , 元素值每次增加 2
```

■若省略initialValue的起始值，其預設為0，如下所示：

```
>>> for k in range(6):
        print(k)
```

找找錯誤在哪兒




```
0
1
2
3
4
5
```

```
for k in range(initialValue, endValue, stepValue)
```

其中stepValue是每次的增量，例如：

```
>>> for k in range(1, 10, 2):
        print(k)
```

找找錯誤在哪兒


```
1
3
5
7
9
```

- 此迴圈範圍從1到9，每次的增量為2，而不是1（預設值），並加以印出。
- 所以印出的答案為1、3、5、7、9。

- 注意，stepValue也可以是負值。若是負值，則其範圍是從initialValue到endValue+1，例如：

```
>>> for k in range(10, 1, -1):
        print(k)
```

找找錯誤在哪兒

輸出結果

```
10
9
8
7
6
5
4
3
2
```

此時迴圈的增量為 -1，所以從10到2，每次皆加(-1)。

範例程式

- **1. 印出100次的Learning Python is fun!**
  - #p6-13.py
  - for k in range(1, 101):
  - print('Learning Python is fun!')

2. 計算1加到100的總和

```
#p6-14-1.py
total = 0
for i in range(1, 101):
    total += i
print('1 + 2 + 3 + ... + 100 = ', total)
```

3. 計算1加到100的偶數和

```
#p6-14-2.py
total = 0
for i in range(2, 101, 2):
    total += i
print('2 + 4 + 6 + ... + 100 = ', total)
```

6-3 break 與continue

- 迴圈執行時，如果要中途結束迴執行，可使用 break 命令強制離開迴圈，例如：

```
for i in range(1,11):
    if(i==6):
        break
    print(i, end=",")     # 執行結果：1,2,3,4,5,
```

- continue 命令則是在迴圈執行中途暫時停住不往下執行，而跳到迴圈起始處繼續執行，例如：

```
for i in range(1,11):
    if(i==6):
        continue
    print(i, end=",")     # 執行結果：1,2,3,4,5,7,8,9,10,
```

6-3 break 與continue

- 假設我們從1開始累加，要找出加到哪一位數字時，總和是大於或等於100。

#p6-15.py

```
total = 0
number = 0
while True:
    number += 1
    total += number
    if total >= 100:
        break
print('The number is', number)
print('Total is', total)
```

輸出結果

```
The number is 14
Total is 105
```

練習題6-16

- 若要找出一直累積偶數和，直到大於或等於1000，此數字是多少？

#p6-16.py

```
total = 0
number = 0
while True:
    number += 2
    total += number
    if total >= 1000:
        break
print('The number is', number)
print('Total is', total)
```

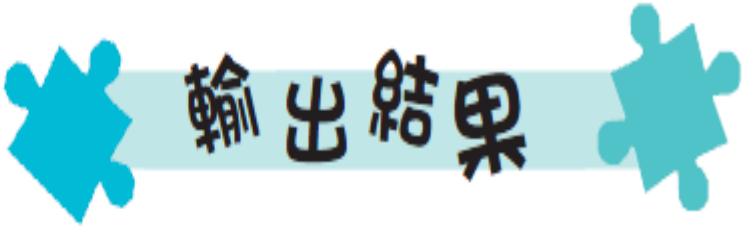

```
The number is 64
Total is 1056
```

練習題目p6-17-1

- 假設從1 加到100，但不加5 的倍數之數值

#p6-17-1.py

```
total = 0
number = 1
while number <= 100:
    if number % 5 == 0:
        number += 1
        continue
    total += number
    number += 1
print('the total is', total)
```




```
the total is 4000
```

6-5　多重迴圈

- 多重迴圈表示迴圈中的敘述又有迴圈敘述稱之。
- 在多重迴圈中有內迴圈和外迴圈之稱。
- 在迴圈外圍的稱為外迴圈，在迴圈內層的稱為內迴圈。

#p6-30.py

```
for i in range(1, 10):
    for j in range(1, 10):
        print('%d*%d=%2d '%(j, i, i*j), end = ' ')
    print()
```

輸出結果

1*1= 1	2*1= 2	3*1= 3	4*1= 4	5*1= 5	6*1= 6	7*1= 7	8*1= 8	9*1= 9
1*2= 2	2*2= 4	3*2= 6	4*2= 8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
1*3= 3	2*3= 6	3*3= 9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
1*4= 4	2*4= 8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
1*5= 5	2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
1*6= 6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
1*7= 7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
1*8= 8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
1*9= 9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

撰寫<九九乘法表>，程式常犯的錯誤，
請參考p6-25.py、p6-28.py、p6-29.py

6-6　範例集錦

6-6-1　產生100個亂數

- Python的亂數是在random模組下，經由呼叫random()或randint()所產生的數字。
- 呼叫random()函式將會產生浮點數的亂數，而randint(a, b)將會產生介於a~b之間的亂數。

random的函數還有很多，參考資料：

https://docs.python.org/3.1/library/random.html

#p6-32-1.py

```
import random
for i in range(1, 101):
    randomNumber = random.randint(1, 100)
    print(randomNumber, end = ' ')
```

輸出結果

23 42 56 83 63 98 56 10 15 41 86 89 22 47 1 84 34 95 61 49 90 21 22 2 87 88 64 5
80 7 28 58 22 31 33 69 58 36 74 45 48 37 24 20 83 71 97 45 100 88 58 38 60 23 89
51 100 19 2 78 71 35 81 89 99 41 82 63 57 59 18 40 34 59 7 12 88 16 45 43 65 97
73 72 9 22 82 1 99 96 72 18 75 5 86 81 10 54 74 10

- 由於要呼叫random模組下的randint函式，所以需要將random模組載入進來，因此在程式中需要import random這一敘述。
- 注意，random.randint(1,100)表示其產生的亂數是介於1到100之間。再次提醒您，for i in range(1,101)其區間為1到100。

練習題目p6-32-2

- 計算100個亂數中有幾個偶數，有幾個奇數

#p6-32-2.py

```
import random
evenCount = 0
oddCount = 0
for i in range(1, 101):
    randomNumber = random.randint(1, 100)
    print(randomNumber, end = ' ')
    if randomNumber % 2 == 0:
        evenCount += 1
oddCount = 100 - evenCount
print('\nEven number: ', evenCount, '\nOdd number: ',
oddCount)
```

輸出結果

```
7 35 95 63 93 52 32 3 71 26 7 100 6 44 19 56 13 84 12 79 42 60 21 35 39 26 19 28
9 50 49 35 37 57 64 8 44 65 65 58 30 68 35 99 12 95 85 37 80 59 79 77 81 98 93 44
67 51 91 13 52 77 1 92 33 96 80 59 21 3 30 95 28 59 61 39 17 50 73 55 95 53 86 64
81 84 4 26 3 62 75 92 48 78 31 83 64 83 76 40
Even number:  43
Odd number:  57
```

6-6-3　判斷它是否為質數

- 若一數字只能被1和本身整除，則此數字稱為質數（prime number）。
- 今由使用者輸入一數字，然後判斷此數字是否為質數。

程式解析

- 判斷一數字（假設是number）是否為質數，簡單的做法是將數字除以一變數（假設為divisor），此變數從2開始，一直遞增1，直到number < divisor的判斷條件式為假才結束。
- 在while迴圈中，利用if敘述更改flag變數，若number除以divisor的餘數為0時，則將flag設為0，此時已得知此數字不是質數，並利用break結束迴圈。

#p6-35.py

```
number = eval(input('Please input an integer: '))
divisor = 2
flag = 1
while divisor < number:
    if number % divisor == 0:
        # If true, number is not prime number
        flag = 0
        break
    divisor += 1
if flag == 0:
    print(number, 'is not prime number')
else:
    print(number, 'is pr...umber')
```

輸出結果

```
Please input an integer: 223
223 is prime number
```

6-6-4　判斷1~1000數字中的質數

- 假設今要將1~1000的數字中，若是質數，則將其印出。如下範例程式所示：

#p6-38.py

題目：將結果每十個列印於一列

```
count = 0
for number in range(2, 1001):
    divisor = 2
    flag = 1
    while divisor  <= number / 2:
        if number % divisor == 0:
            flag = 0
            break
        divisor += 1
```

```
if flag == 1:
        count += 1
        if count % 10 == 0:
            print(format(number, '5d'))
        else:
            print(format(number, '5d'), end = '')
```

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997		

程式解析

- 程式利用count變數來計數，當count除以10的餘數為0時，則以5d的格式規格印出並跳行，否則，以同樣的格式規格印出，但不跳行。

6-6-5　計算兩整數的最大公因數

- 兩整數的最大公因數（Greatest Common Divisor, GCD）是取其兩個整數的公因數當中最大者，如8與12，這兩數的公因數計有2和4，取其最大的公因數4，即為其最大公因數。

程式解析

- 程式一開始設定gcd為1，並設定一變數k來判斷是否小於或等於number1及number2，若成立，再判斷這兩數是否都可以整除k，若是，則將此數k指定給gcd。
- 接著將k加1，直到k大於number1或number2。

#p6-40.py

```
number1 = eval(input('Please input an integer: '))
number2 = eval(input('Please input an integer: '))
gcd = 1
k = 2
while k <= number1 and k <= number2:
    if number1 % k == 0 and number2 % k == 0:
        gcd = k
    k += 1
print('The GCD for', number1, 'and', number2, 'is', gcd)
```

輸出結果

```
Please input an integer: 125
Please input an integer: 2525
The GCD for 125 and  2525 is 25
```

6-6-7 產生大樂透的六個數字

```
#p6-42.py
import random
for i in range(1, 7):
    k = random.randint(1, 49)
    print(k, end = ' ')
```


```
17 26 19 2 41 39
```

再多執行幾次時可能會碰到產生重複的數字，如以下的輸出結果：


```
20 3 6 9 35 35
```

Lab 6-1-1

- 有1、2、3 3個數字，能組成多少個互不相同且無重複數字的三位數？ 數字都是多少？

digit3.py

Lab 6-1-2

■計算薪資：

假設您到一家行銷公司上班，薪水包括底薪和佣金。底薪是6,000，而其佣金率如下表：

銷售金額	佣金率
0.01 ~ 5,000	10%
5,001 ~ 10,000	12%
10,001 ~ 15,000	14%
15,001以上	16%

試撰寫一程式，要求使用者輸入一數字k代表接下來有k位員工。接著讓使用者輸入k位員工的銷售金額並計算k位員工的薪水。

exercise6-2-ok.py