

# Python 程式設計

## 期末考試題

### 銘傳大學電腦與通訊工程系

班 級	電通四乙
姓 名	李易軒
學 號	05050084
成 績	試卷共 <u>10</u> 題，每題 10 分 共完成 <u>10</u> 題，應得 <u>100</u> 分
授課教師	陳慶逸

- ※ 將你的程式碼貼在指定的欄位裡，並且執行題目要求的輸入參數(為避免打錯，建議使用複製和貼上的功能來執行程式的參數)，再複製執行結果並貼在指定的欄位裡。
- ※ 需將執行結果擷圖貼上，以確保是程式執行結果，請確實依題目要求來完成。
- ※ 答完試題後，直接上傳 word 檔即可，不必轉成 pdf。
- ※ 不得使用任何通訊軟體和手機，切記：千萬不要作弊! 千萬不要作弊! 千萬不要作弊! 否則不論首從均以零分計算。

1: 請以 Python 實現一個函式 `prod_in_list(list)`，該函式能求得所傳入之串列(list)中，相鄰元素兩兩相加再除二(即相鄰兩點求平均值)所構成的串列，並且回傳。

例如：

輸入	執行結果
<code>print(prod_in_list([27, 3, 4, 5, 7, 1]))</code>	<code>[15.0, 3.5, 4.5, 6.0, 4.0]</code>
<code>print(prod_in_list([2, 3, 14, 5, 7, 41]))</code>	<code>[2.5, 8.5, 9.5, 6.0, 24.0]</code>

我的作答(包含程式碼、貼上結果的數值、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

<pre>def prod_in_list(a):     a=list(a)     s=[]     for i in range(5):         s.append((a[i]+a[i+1])/2)      return s</pre>
---

請依下面要求輸入參數，並將執行結果的數值貼在指定欄位：

<code>print(prod_in_list([27, 13, 8, 5, 7, 1]))</code>	<code>[20.0, 10.5, 6.5, 6.0, 4.0]</code>
<code>print(prod_in_list([-20, -10, -40, 6, 27, 17]))</code>	<code>[-15.0, -25.0, -17.0, 16.5, 22.0]</code>
<code>print(prod_in_list([2, 13, 14, 55, 7, 41]))</code>	<code>[7.5, 13.5, 34.5, 31.0, 24.0]</code>
<code>print(prod_in_list([60, 2, 400, 3, 8, 12]))</code>	<code>[31.0, 201.0, 201.5, 5.5, 10.0]</code>
<code>print(prod_in_list([21, 1, 44, 55, 17, 280]))</code>	<code>[11.0, 22.5, 49.5, 36.0, 148.5]</code>

執行結果擷圖：

```
In [23]: def prod_in_list(a):
          a=list(a)
          s=[]
          for i in range(5):
              s.append((a[i]+a[i+1])/2)

          return s

          print(prod_in_list([27, 13, 8, 5, 7, 1]))
          print(prod_in_list([-20, -10, -40, 6, 27, 17]))
          print(prod_in_list([2, 13, 14, 55, 7, 41]))
          print(prod_in_list([60, 2, 400, 3, 8, 12]))
          print(prod_in_list([21, 1, 44, 55, 17, 280]))

          [20.0, 10.5, 6.5, 6.0, 4.0]
          [-15.0, -25.0, -17.0, 16.5, 22.0]
          [7.5, 13.5, 34.5, 31.0, 24.0]
          [31.0, 201.0, 201.5, 5.5, 10.0]
          [11.0, 22.5, 49.5, 36.0, 148.5]
```

2: 請以 Python 實現一個函式 `missing_char(str, n)`，函式輸入的參數 `str` 為非空的字串(non-empty string)，參數 `n` 為正整數。函式執行後會傳回一個由原字串 `str` 所構成的新字串，但新字串全部為大寫，而且其中索引 `n` 對應的字元會變成 '\$'。

注意：`n` 的值得是原始字串 `str` 中字元的有效索引值（即 `n` 必須在 `0..len (str)` -1 的範圍之內）。

例如：

輸入	執行結果
<code>print(missing_char('kitten', 1))</code>	K\$TTEN
<code>print(missing_char('Kitten', 0))</code>	\$ITTEN
<code>print(missing_char('kitTen', 4))</code>	KITT\$N

我的作答：(包含程式碼、貼上結果的數值、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

<pre>def missing_char(a,n):     la=list(a)     for i in range(len(la)):         if i==n:             la.pop(i)             la.insert(i,'\$')      a= "".join(la)     return a</pre>
---

請依下面要求輸入參數，並將執行結果的數值貼在指定欄位：

<code>print(missing_char('Hi', 0))</code>	\$i
<code>print(missing_char('Hi', 1))</code>	H\$
<code>print(missing_char('code', 0))</code>	\$ode
<code>print(missing_char('code', 3))</code>	cod\$
<code>print(missing_char('chocolate', 8))</code>	chocolat\$

執行結果擷圖：

```
In [29]: def missing_char(a,n):  
        la=list(a)  
        for i in range(len(la)):  
            if i==n:  
                la.pop(i)  
                la.insert(i,'$')  
  
        a= "".join(la)  
        return a  
  
print(missing_char('Hi', 0))  
print(missing_char('Hi', 1))  
print(missing_char('code', 0))  
print(missing_char('code', 3))  
print(missing_char('chocolate', 8))
```

```
$i  
H$  
$ode  
cod$  
chocolat$
```

3: 試實現一個 python 函式 `add_string(str1)`，當使用者輸入一個英文句子的字串，該程式能將句子的每一個單字以空格切開，轉成一個串列(list)。並依序在字串中的每個元素最末端加上'ed'；但是當元素的最末兩個字元已經是'ed'時，則在'ed'後面加上'ly'。

例如：

輸入	執行結果
<code>add_string('try aborted')</code>	<code>['tryed', 'abortedly']</code>
<code>add_string('listed around')</code>	<code>['listedly', 'arounded']</code>

我的作答：(包含程式碼、貼上結果的數值、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
def add_string(a):
    a=a.split()
    a1=list(a[0])
    a2=list(a[1])

    for i in range(len(a1)):
        if(a[0][-2:]=='ed'):
            a1.append('ly ')
            break
        else:
            a1.append('ed ')
            break
    for i in range(len(a2)):
        if(a[1][-2:]=='ed '):
            a2.append('ly ')
            break
        else:
            a2.append('ed ')
            break
```

```
a3=a1+a2
    la= "".join(a3)

    return la
```

請依下面要求輸入參數，並將執行結果的數值貼在指定欄位：

<code>print(add_string('string read'))</code>	stringed readed
<code>print(add_string('abced goodly'))</code>	abcedly goodlyed
<code>print(add_string('elected circuit'))</code>	electedly circuited
<code>print(add_string('bring book'))</code>	bringed booked
<code>print(add_string('trying work'))</code>	tryinged worked

執行結果擷圖：

```
In [65]: def add_string(a):
          a=a.split()
          a1=list(a[0])
          a2=list(a[1])

          for i in range(len(a1)):
              if(a[0][-2:]=='ed'):
                  a1.append('ly ')
                  break
              else:
                  a1.append('ed ')
                  break
          for i in range(len(a2)):
              if(a[1][-2:]=='ed '):
                  a2.append('ly ')
                  break
              else:
                  a2.append('ed ')
                  break
          a3=a1+a2
          la= "".join(a3)

          return la

print(add_string('string read'))
print(add_string('abced goodly'))
print(add_string('elected circuit'))
print(add_string('bring book'))
print(add_string('trying work'))
```

```
stringed readed
abcedly goodlyed
electedly circuited
bringed booked
tryinged worked
```



4: 試寫一個名稱為 `generateDict(n)` 的函式，它會根據我們所給予的  $n$  值，回傳一個內容為  $\{1:\cos(1/2), 2:\cos(2/2), \dots, n:\cos(n/2)\}$  的字典(dictionary)。

例如：

輸入: `generateDict(3)`

輸出：

`{1: 0.8775825618903728, 2: 0.5403023058681398, 3: 0.0707372016677029}`

我的作答：(包含程式碼、貼上結果的數值、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
import math
a={}
def generateDict(n):
    for i in range(1,n+1):
        a[i] = math.cos(i/2)
    return a
```

請執行 `generateDict(5)`，並將執行結果的數值貼在指定欄位：

輸入：	輸出：
<code>generateDict(5)</code>	<code>{1: 0.8775825618903728, 2: 0.5403023058681398, 3: 0.0707372016677029, 4: -0.4161468365471424, 5: -0.8011436155469337}</code>

執行結果擷圖：

```
In [6]: import math
a={}
def generateDict(n):
    for i in range(1,n+1):
        a[i] = math.cos(i/2)
    return a

generateDict(5)
```

```
Out[6]: {1: 0.8775825618903728,
2: 0.5403023058681398,
3: 0.0707372016677029,
4: -0.4161468365471424,
5: -0.8011436155469337}
```

5: 假設 A 向量的值為 $[A_1, A_2, A_3]$ ，B 向量是 $[B_1, B_2, B_3]$ ，則向量 A 與向量 B 的夾角餘弦值(餘弦相似度)為

$$\cos\theta = \frac{\sum_{i=1}^n (A_i * B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} * \sqrt{\sum_{i=1}^n (B_i)^2}} = \frac{A_1 B_1 + A_2 B_2 + A_3 B_3}{\sqrt{A_1^2 + A_2^2 + A_3^2} * \sqrt{B_1^2 + B_2^2 + B_3^2}}$$

舉例來說，A： $[1, 2, 2, 1, 1, 1, 0]$ ，B： $[1, 2, 2, 1, 1, 2, 1]$ ，則

$$\begin{aligned}\cos\theta &= \frac{1 \times 1 + 2 \times 2 + 2 \times 2 + 1 \times 1 + 1 \times 1 + 1 \times 2 + 0 \times 1}{\sqrt{1^2 + 2^2 + 2^2 + 1^2 + 1^2 + 1^2 + 0^2} \times \sqrt{1^2 + 2^2 + 2^2 + 1^2 + 1^2 + 2^2 + 1^2}} \\ &= \frac{13}{\sqrt{12} \times \sqrt{16}} \\ &= 0.938\end{aligned}$$

試寫一個函式 CosSimilarity(a,b)，當我們輸入 a,b 兩筆相同長度的向量時，函式可輸出這兩筆向量的餘弦相似度。

例如：

輸入	執行結果
<code>CosSimilarity([1, 2, 2, 1, 1, 1, 0], [1, 2, 2, 1, 1, 2, 1])</code>	0.938194187
<code>CosSimilarity([4, 3, 2], [1, 6, 2])</code>	0.754192

我的作答：(包含程式碼、貼上結果的數值、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
##5
import math
import numpy as np
```

```
def CosSimilarity(a,b):
    s=[]
    d=[]
    e=[]
    cos=0
    q=0
    w=0
    t=0
    for i in range(len(a)):
        s.append(a[i]*b[i])
        d.append(a[i]*a[i])
        e.append(b[i]*b[i])
    q=sum(s)
    w=sum(d)
    t=sum(e)

    cos=q/((w**0.5)*(t**0.5))

    print(cos)
```

請依下面要求輸入參數，並將執行結果的數值貼在指定欄位

print(CosSimilarity([4.91, 3.2, 1.58, 0.19], [5.85, 3.12, 4.78, 1.6]))	0.9206174965347947
print(CosSimilarity([7.6, 2.56, 6.89, 2.2 ], [5.85, 3.12, 4.78, 1.6]))	0.9884172906405863
print(CosSimilarity([7.6, 2.2 ], [4.78, 1.6]))	0.9991500241453282
print(CosSimilarity([2.56, 6.89], [3.12, 1.6]))	0.7376590245281031
print(CosSimilarity([4.91, 3.2, 1.58, 0.19, 3.2, 2], [8.2, 5.85, 3.12, 4.78, 1.6, 1]))	0.8577360709120262

執行結果擷圖：

```

In [9]: ##5
import math
import numpy as np

def CosSimilarity(a,b):
    s=[]
    d=[]
    e=[]
    cos=0
    q=0
    w=0
    t=0
    for i in range(len(a)):
        s.append(a[i]*b[i])
        d.append(a[i]*a[i])
        e.append(b[i]*b[i])
    q=sum(s)
    w=sum(d)
    t=sum(e)

    cos=q/((w**0.5)*(t**0.5))

    return cos

print(CosSimilarity([4.91, 3.2, 1.58, 0.19], [5.85, 3.12, 4.78, 1.6]))
print(CosSimilarity([7.6, 2.56, 6.89, 2.2 ], [5.85, 3.12, 4.78, 1.6]))
print(CosSimilarity([7.6, 2.2 ], [4.78, 1.6]))
print(CosSimilarity([2.56, 6.89], [3.12, 1.6]))
print(CosSimilarity([4.91, 3.2, 1.58, 0.19, 3.2, 2], [8.2, 5.85, 3.12, 4.78, 1.6, 1]))

0.9206174965347947
0.9884172906405863
0.9991500241453282
0.7376590245281031
0.8577360709120262

```

6: 試撰寫一個 Python 程式來列印'F'這個字型的形狀如下：

```
*****
*
*****
*
*
*
*
*
```

我的作答：(包含程式碼、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
k=7
for i in range(7):
    if i ==0 or i == 2:
        for j in range(k):
            print("*",end="")
        k=k-1
        print("")
    else:
        print('*')
```

執行結果擷圖 ([本題請連同程式碼與執行結果一併擷圖](#))：

```
In [71]: ##6
k=7
for i in range(7):
    if i ==0 or i == 2:
        for j in range(k):
            print("*",end="")
        k=k-1
        print("")
    else:
        print('*')
```

```
*****
*
*****
*
*
*
*
```

7: 試寫一個 Python 函式 `remove_5t(num)`，它能移除輸入之串列(list)中所有為 4 的倍數或 5 的倍數的元素，並將結果放在一個 tuple 中回傳。

例如：

輸入	執行結果
<code>print(remove_5t([2,7, 11, 16, 20, 22, 25, 34, 55]))</code>	<code>(2, 7, 11, 22, 34)</code>
<code>print(remove_5t([12,3,4,5,16,8,10,24,34,35,25,55]))</code>	<code>(3, 34)</code>

我的作答：(包含程式碼、貼上結果的數值、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
def remove_5t(num):  
  
    a = [i for i in num if not(i % 4 == 0 or i % 5 == 0)]  
  
    return a
```

請依下面要求輸入參數，並將執行結果的數值貼在指定欄位：

<code>print(remove_5t([13,32, 116, 24, 67]))</code>	<code>[13, 67]</code>
<code>print(remove_5t([7,12, 37, 4, 237]))</code>	<code>[7,37,237]</code>
<code>print(remove_5t([11, 203, 77, 55, 40]))</code>	<code>[11, 203, 77]</code>
<code>print(remove_5t([2,6, 26, 255, 765]))</code>	<code>[2, 6, 26]</code>
<code>print(remove_5t([24, 16, 17, 56, 123]))</code>	<code>[17, 123]</code>

執行結果擷圖：



```
In [73]: ##7
def remove_5t(num):

    a = [i for i in num if not(i % 4 == 0 or i % 5 == 0)]

    return a

print(remove_5t([13,32, 116, 24, 67]))
print(remove_5t([7,12, 37, 4, 237]))
print(remove_5t([11, 203, 77, 55, 40]))
print(remove_5t([2,6, 26, 255, 765]))
print(remove_5t([24, 16, 17, 56, 123]))
```

```
[13, 67]
[7, 37, 237]
[11, 203, 77]
[2, 6, 26]
[17, 123]
```

8: 試以 NumPy 程式產生一個由 0 所構成的 8\*8 矩陣(Fig.1(a)) ,

- (1) 將周邊一圈以外的 0 全部變為 2，輸出型式如 Fig.1(b)：
- (2) 將周邊第一圈和周圍第二圈以外的值全部變為 4，輸出型式如 Fig.1(c)：
- (3) 將周邊第一圈、第二圈和第三圈以外的值全部變為 6，輸出型式如 Fig.1(d)：

<pre>[[0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]]</pre> <p>(a) "Original array:"</p>	<pre>[[0. 0. 0. 0. 0. 0. 0. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]]</pre> <p>(b)</p>
<pre>[[0. 0. 0. 0. 0. 0. 0. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 4. 4. 4. 4. 2. 0.]  [0. 2. 4. 4. 4. 4. 2. 0.]  [0. 2. 4. 4. 4. 4. 2. 0.]  [0. 2. 4. 4. 4. 4. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]]</pre> <p>(c)</p>	<pre>[[0. 0. 0. 0. 0. 0. 0. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 2. 4. 4. 4. 4. 2. 0.]  [0. 2. 4. 6. 6. 4. 2. 0.]  [0. 2. 4. 6. 6. 4. 2. 0.]  [0. 2. 4. 4. 4. 4. 2. 0.]  [0. 2. 2. 2. 2. 2. 2. 0.]  [0. 0. 0. 0. 0. 0. 0. 0.]]</pre> <p>(d)</p>

Fig.1

我的作答：(包含程式碼、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
import numpy as np
#1
x = np.zeros((8, 8))
x[1:7, 1:7] = 2
print(x)

print("\n")
#2
import numpy as np
x = np.ones((8, 8))
x[1:7, 1:7] = 2
x[2:6, 2:6] = 4
print(x)
print("\n")
#3
import numpy as np
x = np.ones((8, 8))
x[1:7, 1:7] = 2
x[2:6, 2:6] = 4
x[3:5, 3:5] = 6
print(x)
```

執行結果擷圖 (因結果較長，可連同程式碼和結果分兩次擷圖，再分別貼在欄位左邊和右邊位置)：

```
In [75]: import numpy as np
#1
x = np.zeros((8, 8))
x[1:7, 1:7] = 2
print(x)

print("\n")
#2
import numpy as np
x = np.ones((8, 8))
x[1:7, 1:7] = 2
x[2:6, 2:6] = 4
print(x)
print("\n")
#3
import numpy as np
x = np.ones((8, 8))
x[1:7, 1:7] = 2
x[2:6, 2:6] = 4
x[3:5, 3:5] = 6
print(x)
```

```
[0. 0. 0. 0. 0. 0. 0. 0.]
[0. 2. 2. 2. 2. 2. 2. 0.]
[0. 2. 2. 2. 2. 2. 2. 0.]
[0. 2. 2. 2. 2. 2. 2. 0.]
[0. 2. 2. 2. 2. 2. 2. 0.]
[0. 2. 2. 2. 2. 2. 2. 0.]
[0. 2. 2. 2. 2. 2. 2. 0.]
[0. 0. 0. 0. 0. 0. 0. 0.]
```

```
[[1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 2. 2. 2. 2. 2. 2. 1.]
 [1. 2. 4. 4. 4. 4. 2. 1.]
 [1. 2. 4. 4. 4. 4. 2. 1.]
 [1. 2. 4. 4. 4. 4. 2. 1.]
 [1. 2. 4. 4. 4. 4. 2. 1.]
 [1. 2. 2. 2. 2. 2. 2. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]]
```

```
[[1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 2. 2. 2. 2. 2. 2. 1.]
 [1. 2. 4. 4. 4. 4. 2. 1.]
 [1. 2. 4. 6. 6. 4. 2. 1.]
 [1. 2. 4. 6. 6. 4. 2. 1.]
 [1. 2. 4. 4. 4. 4. 2. 1.]
 [1. 2. 2. 2. 2. 2. 2. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1.]]
```

9: 對於 txt 資料，在 Numpy 裡可以使用 `.loadtxt` 或是 `np.genfromtxt` 來讀取它。下面輸入的程式可以下載 iris data 的 150 資料(`irisdata`)，每筆資料有 4 個維度(花萼的長度、寬度和花瓣的長、寬)。試將 `irisdata` 第 0 行(花萼的長度)的 150 筆資料進行正規化，使其值的分布介於 0 到 1 之間。

輸入：

```
import numpy as np
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
irisdata = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0,1,2,3])
```

我的作答：(包含程式碼、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

sepallengthmin, sepallengthmax = sepallength.min(), sepallength.max()
# 求最大最小值
sepallength = (sepallength-sepallengthmin)/(sepallengthmax-
sepallengthmin) # (矩阵元素-最小值)/(最大值-最小值)
print(sepallength)
```

執行結果擷圖：

```

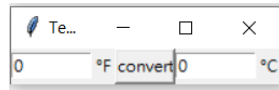
In [10]: url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])

sepallengthmin, sepallengthmax = sepallength.min(), sepallength.max() # 求最大最小值
sepallength = (sepallength-sepallengthmin)/(sepallengthmax-sepallengthmin) # (矩阵元素-最小值)/(最大值-最小值)
print(sepallength)

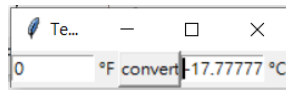
[0.22222222 0.16666667 0.11111111 0.08333333 0.19444444 0.30555556
 0.08333333 0.19444444 0.02777778 0.16666667 0.30555556 0.13888889
 0.13888889 0.         0.41666667 0.38888889 0.30555556 0.22222222
 0.38888889 0.22222222 0.30555556 0.22222222 0.08333333 0.22222222
 0.13888889 0.19444444 0.19444444 0.25         0.25         0.11111111
 0.13888889 0.30555556 0.25         0.33333333 0.16666667 0.19444444
 0.33333333 0.16666667 0.02777778 0.22222222 0.19444444 0.05555556
 0.02777778 0.19444444 0.22222222 0.13888889 0.22222222 0.08333333
 0.27777778 0.19444444 0.75         0.58333333 0.72222222 0.33333333
 0.61111111 0.38888889 0.55555556 0.16666667 0.63888889 0.25
 0.19444444 0.44444444 0.47222222 0.5         0.36111111 0.66666667
 0.36111111 0.41666667 0.52777778 0.36111111 0.44444444 0.5
 0.55555556 0.5         0.58333333 0.63888889 0.69444444 0.66666667
 0.47222222 0.38888889 0.33333333 0.33333333 0.41666667 0.47222222
 0.30555556 0.47222222 0.66666667 0.55555556 0.36111111 0.33333333
 0.33333333 0.5         0.41666667 0.19444444 0.36111111 0.38888889
 0.38888889 0.52777778 0.22222222 0.38888889 0.55555556 0.41666667
 0.77777778 0.55555556 0.61111111 0.91666667 0.16666667 0.83333333
 0.66666667 0.80555556 0.61111111 0.58333333 0.69444444 0.38888889
 0.41666667 0.58333333 0.61111111 0.94444444 0.94444444 0.47222222
 0.72222222 0.36111111 0.94444444 0.55555556 0.66666667 0.80555556
 0.52777778 0.5         0.58333333 0.80555556 0.86111111 1.
 0.58333333 0.55555556 0.5         0.94444444 0.55555556 0.58333333
 0.47222222 0.72222222 0.66666667 0.72222222 0.41666667 0.69444444
 0.66666667 0.66666667 0.55555556 0.61111111 0.52777778 0.44444444]

```

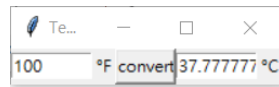
- 10: 試利用 GUI 的文字方塊功能實現一個華氏對攝氏的轉換程式( $\text{攝氏} = (\text{華氏} - 32) * 5/9$ )，其介面外觀如下：



當我們在左側文字方塊輸入 0°F 時，按下 convert 按鈕，右邊的文字方塊會顯示 -17.777°C：



當我們在左側文字方塊輸入 100°F 時，按下 convert 按鈕，右邊的文字方塊會顯示 37.777°C：



我的作答：(包含程式碼、將執行結果擷圖貼上)：

請在下面欄位貼上程式碼：

```
from tkinter import *
def add():
    n3.set((n1.get()-32)*5/9)

window = Tk()
window.title("EX3")                # 視窗標題

n1 = IntVar()
n3 = IntVar()

e1 = Entry(window,width=8,textvariable=n1)           # 文字方塊1
label = Label(window,width=3,text='F')              # °C
label2 = Label(window,width=3,text='°C')            # °F
btn = Button(window,width=5,text='convert',command=add) # =按鈕
e3 = Entry(window,width=8,textvariable=n3)          # 儲存結果文字方塊
```

```

e1.grid(row=0,column=0)          # 定位文字方塊1
label.grid(row=0,column=1)       # 定位°C
btn.grid(row=0,column=3)         # 定位=按鈕
e3.grid(row=0,column=4)          # 定位儲存結果
label2.grid(row=0,column=5)      # 定位°F

window.mainloop()

```

執行結果擷圖：

```

In [*]: ##10
from tkinter import *
def add():
    n3.set((n1.get()-32)*5/9)

window = Tk()
window.title("EX3")

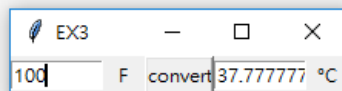
n1 = IntVar()
n3 = IntVar()

e1 = Entry(window,width=8,textvariable=n1)          # 文字方塊1
label = Label(window,width=3,text='°F')             # °C
label2 = Label(window,width=3,text='°C')            # °F
btn = Button(window,width=5,text='convert',command=add) # =按鈕
e3 = Entry(window,width=8,textvariable=n3)          # 儲存結果文字方塊

e1.grid(row=0,column=0)          # 定位文字方塊1
label.grid(row=0,column=1)       # 定位°C
btn.grid(row=0,column=3)         # 定位=按鈕
e3.grid(row=0,column=4)          # 定位儲存結果
label2.grid(row=0,column=5)      # 定位°F

window.mainloop()

```



# 視窗標題

例如：

左側文字方塊輸入值	執行結果擷圖（請擷圖並貼在欄位上）
50°F	



80°F

EX3

80 F convert 26.666666 °C