

Python 程式設計作業

範圍： 字串的處理

銘傳大學電腦與通訊工程系

班 級	電通四乙
姓 名	陳昱叡
學 號	04052474
作業成果	應繳作業共 <u>13</u> 題，第 1 題 16 分，後 12 題每題 7 分，滿分為 100 分 我共完成 <u>13</u> 題，應得 <u>100</u> 分
授課教師	陳慶逸

■ 請確實填寫自己寫完成題數，並且計算得分。填寫不實者(如上傳與作業明顯無關的答案，或是計算題數有誤者)，本次作業先扣 50 分。

EX1: 鳶尾花資料集是非常著名的生物資訊資料集之一，取自美國加州大學歐文分校的機器學習資料庫 <http://archive.ics.uci.edu/ml/datasets/Iris>，資料的筆數為 150 筆，共有五個欄位：

1. 花萼長度(Sepal Length)：計算單位是公分。
2. 花萼寬度(Sepal Width)：計算單位是公分。
3. 花瓣長度(Petal Length)：計算單位是公分。
4. 花瓣寬度(Petal Width)：計算單位是公分。
5. 類別(Class)：可分為 Setosa，Versicolor 和 Virginica 三個品種。

試計算 [4.21 3.02 1.09 0.1] 與 Iris data 中 150 筆資料的歐式距離值，並輸出與該筆輸入資料最相似的資料及其對應的類別。如：

The closest feature vector is : x x x x

The class of the closest feature vector is: x

程式碼：

```
from sklearn import datasets
iris = datasets.load_iris()

X = iris.data[:, :4]
Y = iris.target

ans=[]
com=[4.21,3.02,1.09,0.1]
for i in X:
    a=(com[0]-i[0])**2
    b=(com[1]-i[1])**2
    c=(com[2]-i[2])**2
    d=(com[3]-i[3])**2
    ans.append((a+b+c+d)**0.5)
small=ans[0]
index=0
for i in range(0,len(ans)):
    if small>ans[i]:
```

```
        small=ans[i]
        index=i
    print(small)
    print('The closest feature vector is :',iris.data[i, :4])
    print('The class of the closest feature vector is:',iris.target[i])
```

執行結果擷圖：

```
In [22]: from sklearn import datasets
iris = datasets.load_iris()

X = iris.data[:, :4]
Y = iris.target

ans=[]
com=[4.21,3.02,1.09,0.1]
for i in X:
    a=(com[0]-i[0])**2
    b=(com[1]-i[1])**2
    c=(com[2]-i[2])**2
    d=(com[3]-i[3])**2
    ans.append((a+b+c+d)**0.5)
small=ans[0]
index=0
for i in range(0,len(ans)):
    if small>ans[i]:
        small=ans[i]
        index=i
print(small)
print('The closest feature vector is :',iris.data[i, :4])
print('The class of the closest feature vector is:',iris.target[i])

0.0927361849549569
The closest feature vector is : [5.9 3. 5.1 1.8]
The class of the closest feature vector is: 2
```

EX 3: 試實現一個 python 程式，它能之接收使用者所輸入的文字字串，將索引值為偶數的字元改為小寫，索引值為奇數的字元改為大寫；再把字串反轉才輸

出。

例如：

輸入：apple

輸出：eLpPa

輸入：circuit analysis: theory and parctice

輸出：eCiTcRaP DnA YrOeHt :SiSyLaNa tIuCrIc

程式碼：

```
string=input("輸入文字字串")
string_output=''
for i in range(0,len(string)):
    if i%2==0:
        lower_word=string[i].lower()
        string_output=string_output+lower_word
    else:
        upper_word=string[i].upper()
        string_output=string_output+upper_word
string_output_2 = string_output[::-1]
print('輸出:',string_output_2)
```

執行結果擷圖：

Code

```
In [17]: string=input("輸入文字字串")
string_output=''
for i in range(0,len(string)):
    if i%2==0:
        lower_word=string[i].lower()
        string_output=string_output+lower_word
    else:
        upper_word=string[i].upper()
        string_output=string_output+upper_word
string_output_2 = string_output[::-1]
print('輸出:',string_output_2)
```

輸入文字字串apple
輸出: eLpPa

```
In [18]: string=input("輸入文字字串")
string_output=''
for i in range(0,len(string)):
    if i%2==0:
        lower_word=string[i].lower()
        string_output=string_output+lower_word
    else:
        upper_word=string[i].upper()
        string_output=string_output+upper_word
string_output_2 = string_output[::-1]
print('輸出:',string_output_2)
```

輸入文字字串circuit analysis: theory and practice
輸出: eCiTrAp DnA YrOeHt :SiSyLaNa tIuCrIc

EX 4: 試實現一個 python 程式，它能列印出 100 到 400 之間(包含 100 與 400)，構成數值的三個字元均為偶數的整數所構成的串列(list)。

[說明]：我們可在程式中先建立一個空的串列 lst，並利用 lst.append()函式來逐漸加入符合條件的項目。

以本題而言，輸出結果應為：

[200, 202, 204, 206, 208, 220, 222, 224, 226, 228, 240, 242, 244, 246, 248, 260, 262, 264, 266, 268, 280, 282, 284, 286, 288, 400]

程式碼：

```
lst=[]
for i in range(100,401):
    flag=0
    num=str(i)
    for x in range(0,3):
        if int(num[x])%2==0:
            flag=flag+1
    if flag==3:
        lst.append(int(num))
print(lst)
```

執行結果擷圖：

```
In [5]: lst=[]
        for i in range(100,401):
            flag=0
            num=str(i)
            for x in range(0,3):
                if int(num[x])%2==0:
                    flag=flag+1
            if flag==3:
                lst.append(int(num))
        print(lst)

[200, 202, 204, 206, 208, 220, 222, 224, 226, 228, 240, 242, 244, 246, 248, 260, 262, 264, 266, 268, 280, 282, 284, 286, 288, 400]
```

EX 7: 試實現一個 python 程式，當使用者輸入一個字串，該程式能把這個字串中與第一個字元相同的所有字元全部改為'\$'，但第一個字元除外。

例如：

輸入: circuit，輸出: cir\$uit

輸入: digital system，輸出: digital system

輸入: introduction to machine learning，輸出: introduct\$on to mach\$ne learn\$
ng

程式碼：

```
string=input('輸入: ')

string_output=string[0]
string_output+=string[1:].replace(string_output,'$')
print('輸出:',string_output)
```

執行結果擷圖：

```
In [6]: string=input('輸入: ')

string_output=string[0]
string_output+=string[1:].replace(string_output,'$')
print('輸出:',string_output)
```

輸入: circuit
輸出: cir\$uit

```
In [7]: string=input('輸入: ')

string_output=string[0]
string_output+=string[1:].replace(string_output,'$')
print('輸出:',string_output)
```

輸入: digital system
輸出: digital system


```
In [8]: string=input('輸入: ')
        string_output=string[0]
        string_output+=string[1:].replace(string_output,'$')
        print('輸出:',string_output)
```

```
輸入: introduction to machine learning
輸出: introduct$on to mach$ne learn$ng
```

EX 8: 試實現一個 python 程式，當使用者輸入一個字串，該程式能在這個字串最末端加上'ing';但是當字串的最末三個字元已經是'ing'時，則在'ing'後面加上'ly'。

例如:

輸入: circuit, 輸出: circuiting

輸入: string, 輸出: stringly

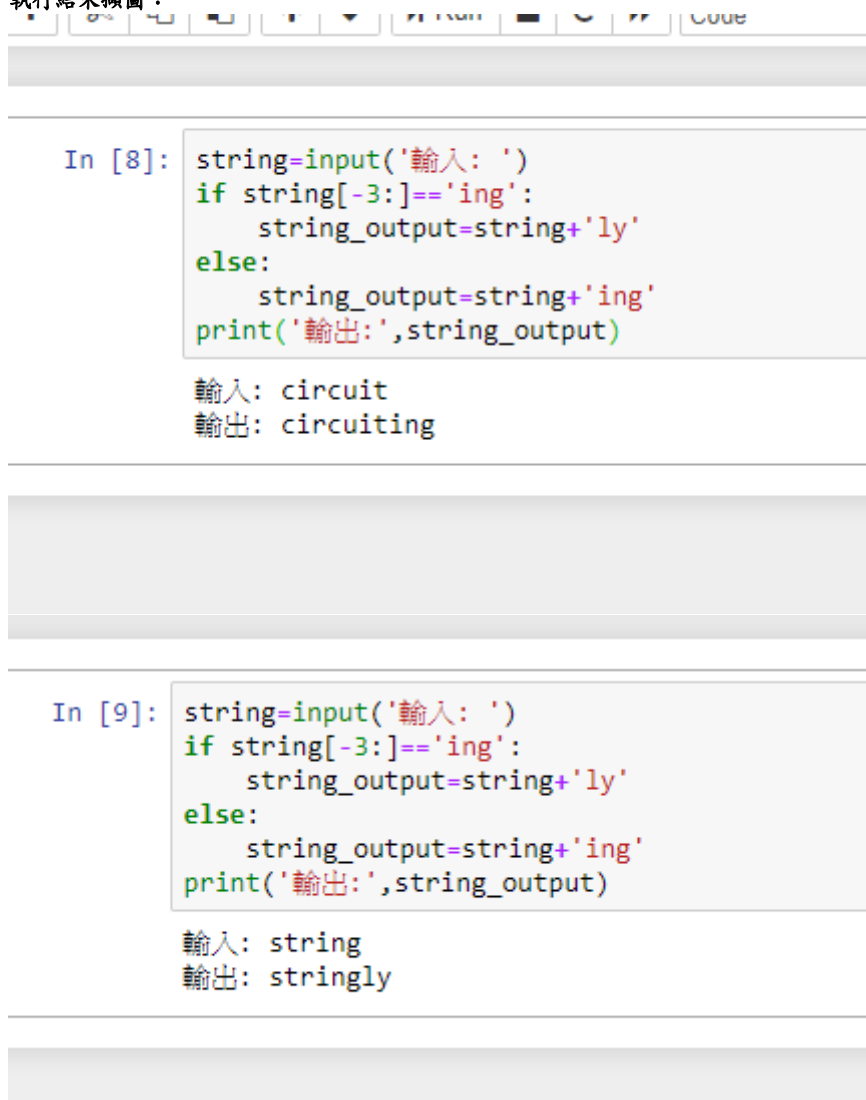
輸入: introduction to machine learning, 輸出: introduct\$on to mach\$ne learn\$ng

程式碼:

```
string=input('輸入: ')
if string[-3:]=='ing':
    string_output=string+'ly'
else:
    string_output=string+'ing'
```

```
print('輸出:',string_output)
```

執行結果擷圖：



The image shows a Jupyter Notebook interface with two code cells. Each cell has a toolbar at the top with icons for undo, redo, insert, delete, and other functions. The first cell, labeled 'In [8]:', contains a Python script that takes user input and appends 'ly' or 'ing' to it based on whether the last three characters are 'ing'. The output shows the input 'circuit' resulting in 'circuiting'. The second cell, labeled 'In [9]:', contains the same script. The output shows the input 'string' resulting in 'stringly'.

```
In [8]: string=input('輸入: ')
        if string[-3:]=='ing':
            string_output=string+'ly'
        else:
            string_output=string+'ing'
        print('輸出:',string_output)

輸入: circuit
輸出: circuiting
```

```
In [9]: string=input('輸入: ')
        if string[-3:]=='ing':
            string_output=string+'ly'
        else:
            string_output=string+'ing'
        print('輸出:',string_output)

輸入: string
輸出: stringly
```

```
In [10]: string=input('輸入: ')
         if string[-3:]=='ing':
             string_output=string+'ly'
         else:
             string_output=string+'ing'
         print('輸出:',string_output)

輸入: introduction to machine learning
輸出: introduction to machine learningly
```

EX 9: 試實現一個 python 程式，當使用者所輸入的字串中，出現'not'和'poor'這兩個子字串，而且'poor'是出現在'not'這個字串的後面(例如'..is not so poor'這樣的情況)，則將整段'not'...'poor'的子字串改以'good'來取代後再輸出改過的字串，否則原句輸出。

例如:

輸入: His english is not so poor.

輸出: His english is good.

輸入: The lyrics of this song are not so poor.

輸出: The lyrics of this song are good.

輸入: circuit analysis

輸出: circuit analysis

程式碼:

```
string=input('輸入: ')
not_index=string.find('not')
poor_index=string.find('poor')

if 'not' in string and 'poor' in string and not_index<poor_index:
    string_output=string[:not_index]+'good'+string[poor_index+4:]
    print('輸出:',string_output)
else:
    print('輸出:',string)
```

執行結果擷圖：

```
In [11]: string=input('輸入: ')
not_index=string.find('not')
poor_index=string.find('poor')

if 'not' in string and 'poor' in string and not_index<poor_index:
    string_output=string[:not_index]+'good'+string[poor_index+4:]
    print('輸出:',string_output)
else:
    print('輸出:',string)
```

輸入: His english is not so poor.

輸出: His english is good.

```
In [12]: string=input('輸入: ')
not_index=string.find('not')
poor_index=string.find('poor')

if 'not' in string and 'poor' in string and not_index<poor_index:
    string_output=string[:not_index]+'good'+string[poor_index+4:]
    print('輸出:',string_output)
else:
    print('輸出:',string)
```

輸入: The lyrics of this song are not so poor.
輸出: The lyrics of this song are good.

```
In [13]: string=input('輸入: ')
not_index=string.find('not')
poor_index=string.find('poor')

if 'not' in string and 'poor' in string and not_index<poor_index:
    string_output=string[:not_index]+'good'+string[poor_index+4:]
    print('輸出:',string_output)
else:
    print('輸出:',string)
```

輸入: circuit analysis
輸出: circuit analysis

EX 10: 試利用 `str.rsplit()` 函式的功能，將輸入的字串以空白為分割點，轉換成 `list`。

例如:

輸入: A Symbolic Analysis of Relay and Switching Circuits

輸出: ['A', 'Symbolic', 'Analysis', 'of', 'Relay', 'and', 'Switching', 'Circuits']

程式碼:

```
string=input('輸入: ')
string_output=string.rsplit(' ')
print('輸出:',string_output)
```

執行結果擷圖：

```
In [1]: string=input('輸入: ')
        string_output=string.rsplit(' ')
        print('輸出:',string_output)

輸入: A Symbolic Analysis of Relay and Switching Circuits
輸出: ['A', 'Symbolic', 'Analysis', 'of', 'Relay', 'and', 'Switchin
g', 'Circuits']
```

EX 11: 試利用 `str.rsplit()` 函式的功能，將輸入的字串以逗號為分割點，轉換成 `list`。

例如：

輸入: The World of Peter, Jane, and Thomas Jefferson

輸出: ['The World of Peter', ' Jane', ' and Thomas Jefferson']

程式碼：

```
string=input('輸入: ')
string_output=string.rsplit(',')
print('輸出:',string_output)
```

執行結果擷圖：

```
In [1]: string=input('輸入: ')
        string_output=string.rsplit(',')
        print('輸出:',string_output)
```

```
輸入: The World of Peter, Jane, and Thomas Jefferson
輸出: ['The World of Peter', ' Jane', ' and Thomas Jefferson']
```

EX 12: 試利用 `str.rsplit()` 函式的功能，將輸入的字串以空白為分割點，轉換成 `list`，而且只保留長度大於 `n` 的 `word` (建議使用 `lst.append()` 功能)。

例如:

輸入:

Please input a string: A Symbolic Analysis of Relay and Switching Circuits

Please word length n: 6

輸出: ['Symbolic', 'Analysis', 'Switching', 'Circuits']

程式碼:

```
string=input('Please input a string: ')
n=input('Please word length n: ')
lst=[]
string_output=string.rsplit(' ')
for x in string_output:
    if len(x)>6:
        lst.append(x)
print('輸出:',lst)
```

執行結果擷圖：

```
In [1]: string=input('Please input a string: ')
n=input('Please word length n: ')
lst=[]
string_output=string.rsplit(' ')
for x in string_output:
    if len(x)>6:
        lst.append(x)
print('輸出:',lst)

Please input a string: A Symbolic Analysis of Relay and Switching Circuits
Please word length n: 6
輸出: ['Symbolic', 'Analysis', 'Switching', 'Circuits']
```

EX 13: 試利用 `str.rsplit()` 函式的功能，達到下述要求的效果。

```
str = 'https://www.ite.mcu.edu.tw/python-exercises/string'
```

列印輸出: <https://www.ite.mcu.edu.tw>

程式碼：

```
string = 'https://www.ite.mcu.edu.tw/python-exercises/string'
string_output=string.rsplit('/',2)
print('列印輸出:',string_output[0])
```

執行結果擷圖：

```
In [9]: string = 'https://www.ite.mcu.edu.tw/python-exercises/string'
string_output=string.rsplit('/',2)
print('列印輸出:',string_output[0])

列印輸出: https://www.ite.mcu.edu.tw
```


EX 14: 試利用字串格式功能，取浮點數至小數點以下兩位。

例如：

輸入: x = 3.1415926, 輸出: Formatted Number: 3.14

程式碼：

```
string=input('輸入: x = ')
start=string.find('.')
string_output=string[:start]+string[start:start+3]
print("輸出: Formatted Number:",string_output)
```

執行結果擷圖：

```
In [2]: string=input('輸入: x = ')
        start=string.find('.')
        string_output=string[:start]+string[start:start+3]
        print("輸出: Formatted Number:",string_output)|

輸入: x = 3.1415926
輸出: Formatted Number: 3.14
```

EX 15: 試利用 str.replace()這個內建函式的功能，移除輸入字串裡的空格。

例如：

輸入: embedded system, 輸出: embeddedsystem

輸入: Python String Exercise, 輸出: PythonStringExercise

程式碼：

```
string=input('輸入: ')
string_output=string.replace(' ','')
print("輸出:",string_output)
```

執行結果擷圖：

```
In [3]: string=input('輸入: ')
string_output=string.replace(' ','')
print("輸出:",string_output)
```

```
輸入: embedded system
輸出: embeddedsystem
```

```
In [ ]:
```

```
In [4]: string=input('輸入: ')
string_output=string.replace(' ','')
print("輸出:",string_output)
```

```
輸入: Python String Exercise
輸出: PythonStringExercise
```

```
In [ ]:
```

EX 16: 英文字母中的母音共有"a","e","i","o","u"等五個字母，若將同樣字母的大小寫視為不同字元的話，則共有十個不同的字元。試實現一個能夠計算和顯示輸入字串中之母音的 Python 程式(建議搭配使用 lst.append()功能來儲存串列裡的內容)。

例如：

輸入: Verilog & VHDL code, 輸出: 5 , ['e', 'i', 'o', 'o', 'e']

輸入: Apple, 輸出: 2 , ['A', 'e']

輸入: Emma Watson, 輸出: 4 , ['E', 'a', 'a', 'o']

程式碼：

```
lst=[]
string=input('輸入: ')
for i in range(len(string)):
    if (string[i]=='a' or string[i]=='e' or string[i]=='i' or
        string[i]=='o' or string[i]=='u' or
        string[i]=='A' or string[i]=='E' or string[i]=='I' or
        string[i]=='O' or string[i]=='U'):
        lst.append(string[i])
print("輸出:",len(lst),' ',lst)
```

執行結果擷圖：

```
In [5]: lst=[]
string=input('輸入: ')
for i in range(len(string)):
    if (string[i]=='a' or string[i]=='e' or string[i]=='i' or string[i]=='o' or string[i]=='u' or
        string[i]=='A' or string[i]=='E' or string[i]=='I' or string[i]=='O' or string[i]=='U'):
        lst.append(string[i])
print("輸出:",len(lst),' ',lst)

輸入: Verilog & VHDL code
輸出: 5 , ['e', 'i', 'o', 'o', 'e']
```

In []:

```
In [6]: lst=[]
string=input('輸入: ')
for i in range(len(string)):
    if (string[i]=='a' or string[i]=='e' or string[i]=='i' or string[i]=='o' or string[i]=='u' or
        string[i]=='A' or string[i]=='E' or string[i]=='I' or string[i]=='O' or string[i]=='U'):
        lst.append(string[i])
print("輸出:",len(lst),',',lst)

輸入: Apple
輸出: 2 , ['A', 'e']
```

In []:

```
In [7]: lst=[]
string=input('輸入: ')
for i in range(len(string)):
    if (string[i]=='a' or string[i]=='e' or string[i]=='i' or string[i]=='o' or string[i]=='u' or
        string[i]=='A' or string[i]=='E' or string[i]=='I' or string[i]=='O' or string[i]=='U'):
        lst.append(string[i])
print("輸出:",len(lst),',',lst)

輸入: Emma Watson
輸出: 4 , ['E', 'a', 'a', 'o']
```

In []:

- -