

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	電通四甲
姓 名	趙啟翔
學 號	05050130
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
dclass={0:'Setosa',1:'Versicolor',2:'Virginica'}
MyList=[]
place=0
for i in X:
    MyList.append((((i[0]-4.21)**2+(i[1]-3.02)**2+(i[2]-1.09)**2+(i[3]-0.1)**2)**0.5)
small=MyList[0]
for i in range(len(MyList)):
    if small>MyList[i]:
        small=MyList[i]
        place=i
print('The closest feature vector is : {} {} {} {}
{}'.format(X[place][0],X[place][1],X[place][2],X[place][3]))
print('The class of the closest feature vector is : {}'.format(dclass[Y[place]]))
```

執行結果擷圖：

```
1 from sklearn import datasets
2 iris = datasets.load_iris()
3 X = iris.data[:, :4]
4 Y = iris.target
5 dclass={0:'Setosa',1:'Versicolor',2:'Virginica'}
6 MyList=[]
7 place=0
8 for i in X:
9     MyList.append(((i[0]-4.21)**2+(i[1]-3.02)**2+(i[2]-1.09)**2+(i[3]-0.1)**2)**0.5)
10 small=MyList[0]
11 for i in range(len(MyList)):
12     if small>MyList[i]:
13         small=MyList[i]
14         place=i
15 print('The closest feature vector is : {} {} {} {}'.format(X[place][0],X[place][1],X[place][2],X[place][3]))
16 print('The class of the closest feature vector is : {}'.format(dclass[Y[place]]))

The closest feature vector is : 4.3 3.0 1.1 0.1
The class of the closest feature vector is : Setosa
```

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：

```
str = input('')
for i in range(len(str)-1,-1,-1):
    if(i%2==1):
        print(str[i].upper(),end='')
    else:
```

```
print(str[i].lower(),end='')
```

執行結果擷圖：

```
In [4]: 1 str = input('')
        2 for i in range(len(str)-1,-1,-1):
        3     if(i%2==1):
        4         print(str[i].upper(),end='')
        5     else:
        6         print(str[i].lower(),end='')

apple
eLpPa
```

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(100,401,1):
    a=i%10
    b=int(i/10%10)
    c=int(i/100%10)

    if(a%2==0 and b%2==0 and c%2==0):
```

```
        lst.append(i)
    else:
        continue
print(lst)
```

執行結果擷圖：

```
In [14]: 1 lst=[]
2 for i in range(100,401,1):
3     a=i%10
4     b=int(i/10%10)
5     c=int(i/100%10)
6
7     if(a%2==0 and b%2==0 and c%2==0):
8         lst.append(i)
9     else:
10         continue
11 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]
```

```
In [ ]: 1
```

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：

```
a=input()
b=''
for i in range(len(a)):
    if a[0]==a[i] :
        if i==0:
```

```
        b+=a[0]
    else:
        b+=' '$'
    else:
        b+=a[i]
print(b)
```

執行結果擷圖：

```
: 1 a=input()
  2 b=''
  3 for i in range(len(a)):
  4     if a[0]==a[i]:
  5         if i==0:
  6             b+=a[0]
  7         else:
  8             b+=' '$'
  9     else:
10         b+=a[i]
11 print(b)
```

```
circuit
cir$uit
```

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：

```
str = input('')
if(str[-3:]=='ing'):
    print(str + 'ly')
else:
    print(str + 'ing')
```

執行結果擷圖：

```
In [19]: 1 str = input('')
          2 if(str[-3:]=='ing'):
          3     print(str + 'ly')
          4 else:
          5     print(str + 'ing')

circuit
circuiting
```

```
In [20]: 1 str = input('')
          2 if(str[-3:]=='ing'):
          3     print(str + 'ly')
          4 else:
          5     print(str + 'ing')

string
stringly
```

```
1 str = input('')
2 if(str[-3:]=='ing'):
3     print(str + 'ly')
4 else:
5     print(str + 'ing')

introduction to machine learning
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：

```
str = input('')
a = int(str.find('not'))
b = int(str.find('poor'))
c = b+4
if(str.find('p')>str.find('n')):
    str1 = str.replace((str[a:c]),'good')
    print(str1)
else:
    print(str)
```

執行結果擷圖：

In [1]:

```
1 str = input('')
2 a = int(str.find('not'))
3 b = int(str.find('poor'))
4 c = b+4
5 if(str.find('p')>str.find('n')):
6     str1 = str.replace((str[a:c]),'good')
7     print(str1)
8 else:
9     print(str)
```

His english is not so poor.
His english is good.

In [2]:

```
1 str = input('')
2 a = int(str.find('not'))
3 b = int(str.find('poor'))
4 c = b+4
5 if(str.find('p')>str.find('n')):
6     str1 = str.replace((str[a:c]),'good')
7     print(str1)
8 else:
9     print(str)
```

The lyrics of this song are not so poor.
The lyrics of this song are good.

In [3]:

```
1 str = input('')
2 a = int(str.find('not'))
3 b = int(str.find('poor'))
4 c = b+4
5 if(str.find('p')>str.find('n')):
6     str1 = str.replace((str[a:c]),'good')
7     print(str1)
8 else:
9     print(str)
```

circuit analysis
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：

```
a=input()
print(a.rsplit(' '))
```

執行結果擷圖：

```
In [4]: 1 a=input()
        2 print(a.rsplit(' '))

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

```
a=input()
print(a.rsplit(','))
```

執行結果擷圖：

```
In [5]: 1 a=input()
        2 print(a.rsplit(','))

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

```
a=input('Please input a string:')
n=int(input('Please word length n:'))
lst=[]
lst1=[]
lst1=a.rsplit(' ')
for i in range (len(lst1)):
    if(len(lst1[i])>n):
        lst.append(lst1[i])
    else:
        continue
print('輸出: ',end='')
print(lst)
```

執行結果擷圖：

```
In [6]: 1 a=input('Please input a string:')
2 n=int(input('Please word length n:'))
3 lst=[]
4 lst1=[]
5 lst1=a.rsplit(' ')
6 for i in range (len(lst1)):
7     if(len(lst1[i])>n):
8         lst.append(lst1[i])
9     else:
10         continue
11 print('輸出: ',end='')
12 print(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：

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

執行結果擷圖：

```
In [7]: 1 str = 'https://www.ite.mcu.edu.tw/python-exercises/string'
2 print(str.rsplit('/',2)[0])
```

`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：

```
x=float(input('輸入: x =' ))
print('輸出: Formatted Number: %.2f' %(x))
```

執行結果擷圖：

```
In [9]: 1 x=float(input('輸入: x =' ))
        2 print('輸出: Formatted Number: %.2f' %(x))

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：

```
str = input('')
print(str.replace(' ',''))
```

執行結果擷圖：

In [10]:

```
1 str = input('')
2 print(str.replace(' ', ''))
3
```

```
embedded system
embeddedsystem
```

In [11]:

```
1 str = input('')
2 print(str.replace(' ', ''))
3
4
```

```
Python String Exercise
PythonStringExercise
```

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内容)。

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)

```

```

In [12]: 1 lst=[]
2 string=input('輸入: ')
3 for i in range(len(string)):
4     if (string[i]=='a' or string[i]=='e' or string[i]=='i' or string[i]=='o' or string[i]=='u' or
5         string[i]=='A' or string[i]=='E' or string[i]=='I' or string[i]=='O' or string[i]=='U'):
6         lst.append(string[i])
7     print("輸出:",len(lst),' ',lst)
8

```

輸入: apple
輸出: 2 , ['a', 'e']

```

In [13]: 1 lst=[]
2 string=input('輸入: ')
3 for i in range(len(string)):
4     if (string[i]=='a' or string[i]=='e' or string[i]=='i' or string[i]=='o' or string[i]=='u' or
5         string[i]=='A' or string[i]=='E' or string[i]=='I' or string[i]=='O' or string[i]=='U'):
6         lst.append(string[i])
7     print("輸出:",len(lst),' ',lst)
8
9

```

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

```

In [14]: 1
2 lst=[]
3 string=input('輸入: ')
4 for i in range(len(string)):
5     if (string[i]=='a' or string[i]=='e' or string[i]=='i' or string[i]=='o' or string[i]=='u' or
6         string[i]=='A' or string[i]=='E' or string[i]=='I' or string[i]=='O' or string[i]=='U'):
7         lst.append(string[i])
8     print("輸出:",len(lst),' ',lst)
9

```

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

執行結果擷圖：