Python 程式設計

範圍： list、tuple 的應用

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
成　　績	應繳作業共 10 題，每題 10 分，滿分為 100 分 共完成 10 題，應得 100 分
授課教師	陳慶逸

※直接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(每一題都有 5 個不同的輸入參數要執行)，再將執行結果擷圖貼在指定的位置。

※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。

EX 1: 試以串列解析式語法來改寫下面的兩個以 for-loop 所寫的程式：

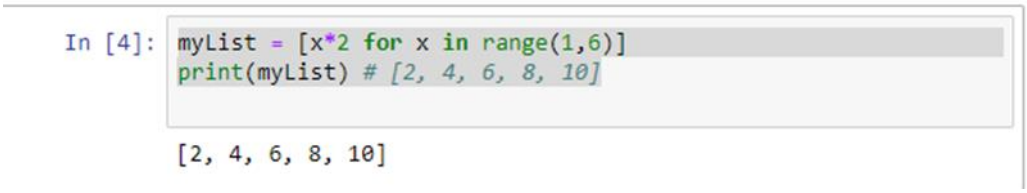
(1) 新串列的內容為原串列的所有元素都乘上 2 的結果：

For-loop 迭代作法：

```
num = [1, 2, 3, 4, 5]

doubledOddNum = []
for n in num:
    doubledOddNum.append(n * 2)

print(doubledOddNum)  #[2, 4, 6, 8, 10]
```

串列解析式語法：

```
myList = [x*2 for x in range(1,6)]
print(myList) # [2, 4, 6, 8, 10]
```



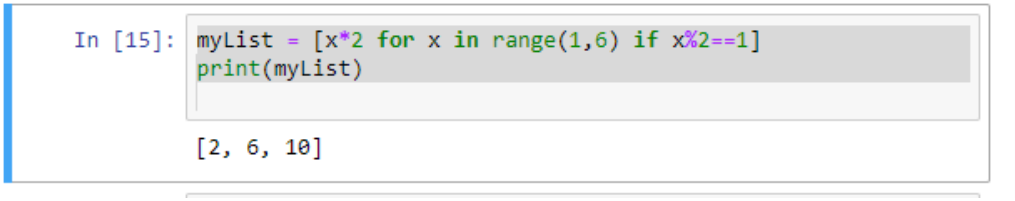
(2) 新串列的內容為原串列的所有奇數值元素都乘上 2 的結果：

For-loop 迭代作法：

```
num = [1, 2, 3, 4, 5]

doubledOddNum = []
for n in num:
    if n % 2 == 1:
        doubledOddNum.append(n * 2)

print(doubledOddNum)  #[2, 6, 10]
```

串列解析式語法：

```
myList = [x*2 for x in range(1,6) if x%2==1]
print(myList)
```



EX 2: 試寫一個 Python 程式來產生一個 3x4x6 的三維陣列。

輸出如下:

[[['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', '0', '0'], ['0', '0', '0', '0', '0', '0']]]

```
mylist = [['0' for col in range(6)] for row in range(4)] for i in
range(3)]

print(mylist)
```

```
In [20]: mylist = [['0' for col in range(6)] for row in range(4)] for i in range(3)]

         print(mylist)

[[['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', '0', '0'], ['0', '0', '0',
'0', '0', '0']]]
```

EX 3: 若有一串列 a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]，試計算此串列裡所有元素的總和。

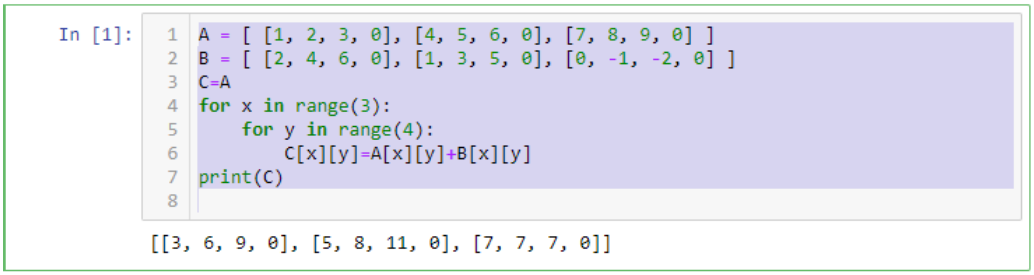
```
a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]
sun=0


for i in range(len(a)):
    for j in range(len(a[i])):
        sun += a[i][j]
 print(sun)
```

```
In [47]: a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]
         sun=0

         for i in range(len(a)):
             for j in range(len(a[i])):
                 sun += a[i][j]
                 #print(a[i][j],end='')
         print(sun)

         45
```

EX 4: 若串列 A = [[1, 2, 3, 0], [4, 5, 6, 0], [7, 8, 9, 0]]，串列 B = [[2, 4, 6, 0], [1,

3, 5, 0], [0, -1, -2, 0]]；試產生一個相同尺寸的串列 C，且其內容為 A + B (矩陣相加)的結果。

```
A = [ [1, 2, 3, 0], [4, 5, 6, 0], [7, 8, 9, 0] ]
B = [ [2, 4, 6, 0], [1, 3, 5, 0], [0, -1, -2, 0] ]
C=A
for x in range(3):
    for y in range(4):
        C[x][y]=A[x][y]+B[x][y]
print(C)
```



EX 5: 試寫一個 Python 程式，它能移除輸入之串列中所有的偶數數值。

例如:

輸入: [7, 8, 120, 25, 44, 20, 27]

輸出: [7, 25, 27]

```
x=[7, 8, 120, 25, 44, 20, 27]
out=[]
for j in x:
    if j%2!=0:
        out.append(j)
print(out)
```

```
In [3]:
x=[7, 8, 120, 25, 44, 20, 27]
out=[]
for j in x:
    if j%2!=0:
        out.append(j)
print(out)

[7, 25, 27]
```

EX 6: 試撰寫一個 Python 函式 tuplexEdit(tuplex)，並利用資料型別轉換的作法，刪除所傳遞進來之 tuple 從前面算過來第 4 個項目，以及從末端算過來第 4 個項目的內容(假設 tuple 的長度均超過 10)，並回傳更改後的 tuple 內容。

例如:

輸入	執行結果
print(tuplexEdit(("E", "m", "b", "e", "d", "e", "d", "S", "y", "s", "t", "e", "m")))	('E', 'm', 'b', 'd', 'e', 'd', 'S', 'y', 't', 'e', 'm')
print(tuplexEdit(("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13")))	('1', '2', '3', '5', '6', '7', '8', '9', '11', '12', '13')

我的作答：

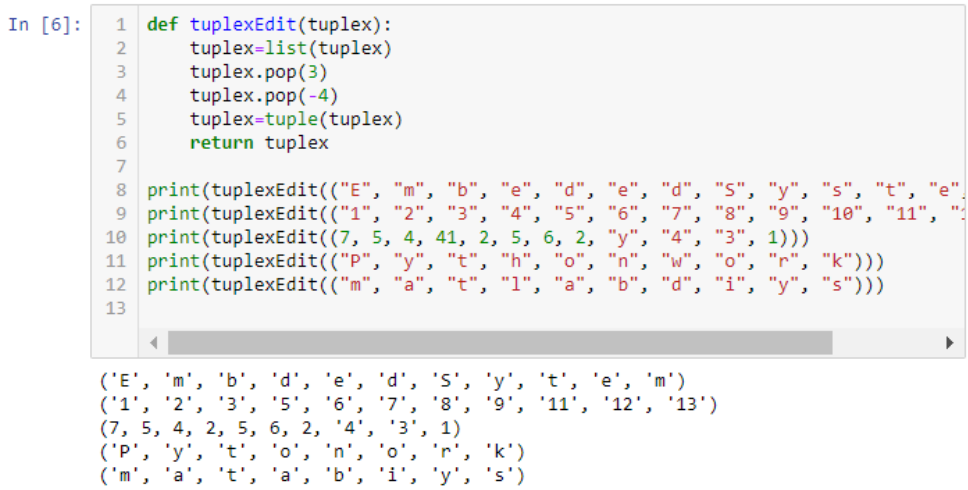
請在下面欄位貼上程式碼：

```
def tuplexEdit(tuplex):
    tuplex=list(tuplex)
    tuplex.pop(3)
    tuplex.pop(-4)
    tuplex=tuple(tuplex)
    return tuplex

print(tuplexEdit(("E", "m", "b", "e", "d", "e", "d", "S", "y", "s",
"t", "e", "m")))
print(tuplexEdit(("1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12", "13")))
print(tuplexEdit((7, 5, 4, 41, 2, 5, 6, 2, "y", "4", "3", 1)))
print(tuplexEdit(("P", "y", "t", "h", "o", "n", "w", "o", "r", "k")))
print(tuplexEdit(("m", "a", "t", "l", "a", "b", "d", "i", "y", "s")))
```

請依下面要求輸入參數，並將執行結果擷圖：

`print(tuplexEdit(("E", "m", "b", "e", "d", "e", "d", "S", "y", "s", "t", "e", "m")))`	`('E', 'm', 'b', 'd', 'e', 'd', 'S', 'y', 't', 'e', 'm')`
`print(tuplexEdit(("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13")))`	`('1', '2', '3', '5', '6', '7', '8', '9', '11', '12', '13')`
`print(tuplexEdit((7, 5, 4, 41, 2, 5, 6, 2, "y", "4", "3", 1)))`	`(7, 5, 4, 2, 5, 6, 2, '4', '3', 1)`
`print(tuplexEdit(("P", "y", "t", "h", "o", "n", "w", "o", "r", "k")))`	`('P', 'y', 't', 'o', 'n', 'o', 'r', 'k')`
`print(tuplexEdit(("m", "a", "t", "l", "a", "b", "d", "i", "y", "s")))`	`('m', 'a', 't', 'a', 'b', 'i', 'y', 's')`

執行結果擷圖：



```
In [6]:
def tuplexEdit(tuplex):
    tuplex=list(tuplex)
    tuplex.pop(3)
    tuplex.pop(-4)
    tuplex=tuple(tuplex)
    return tuplex

print(tuplexEdit(("E", "m", "b", "e", "d", "e", "d", "S", "y", "s", "t", "e",
print(tuplexEdit(("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "
print(tuplexEdit((7, 5, 4, 41, 2, 5, 6, 2, "y", "4", "3", 1)))
print(tuplexEdit(("P", "y", "t", "h", "o", "n", "w", "o", "r", "k")))
print(tuplexEdit(("m", "a", "t", "l", "a", "b", "d", "i", "y", "s")))
```

```
('E', 'm', 'b', 'd', 'e', 'd', 'S', 'y', 't', 'e', 'm')
('1', '2', '3', '5', '6', '7', '8', '9', '11', '12', '13')
(7, 5, 4, 2, 5, 6, 2, '4', '3', 1)
('P', 'y', 't', 'o', 'n', 'o', 'r', 'k')
('m', 'a', 't', 'a', 'b', 'i', 'y', 's')
```

EX 7: 試撰寫一個 Python 函式 tuplexcount(tuplex)，它能計算傳遞進來的 tuple 之每一個項目出現的次數 ，並儲存成一個 list 回傳內容。

例如:

輸入	**執行結果**
print(tuplexcount(("E", "m", "b", "b", "E")))	[2, 1, 2, 2, 2]
print(tuplexcount(("1", "3", "3", "4", "4", "6", "7", "3")))	[1, 3, 3, 2, 2, 1, 1, 3]

我的作答：

請在下面欄位貼上程式碼：

```
def tuplexcount(tuplex):
    want=[]
    for x in tuplex:
        want.append(tuplex.count(x))
    return want

print(tuplexcount(("E", "m", "b", "b", "E")))
print(tuplexcount(("1", "3", "3", "4", "4", "6", "7", "3")))
print(tuplexcount((7, 5, 4, 4, 2, 7, 7, 2, "y", "4", "3", 5)))
print(tuplexcount(("P", "y", "t", "h", "o", "n", "w", "o", "r",
"k")))
print(tuplexcount(("m", "a", "t", "l", "a", "b", "d", "i", "y",
"s")))
```

In [8]:

```
1  def tuplexcount(tuplex):
2      want=[]
3      for x in tuplex:
4          want.append(tuplex.count(x))
5      return want
6
7  print(tuplexcount(("E", "m", "b", "b", "E")))
8  print(tuplexcount(("1", "3", "3", "4", "4", "6", "7", "3")))
9  print(tuplexcount((7, 5, 4, 4, 2, 7, 7, 2, "y", "4", "3", 5)))
10 print(tuplexcount(("P", "y", "t", "h", "o", "n", "w", "o", "r", "k")))
11 print(tuplexcount(("m", "a", "t", "l", "a", "b", "d", "i", "y", "s")))
12
```

```
[2, 1, 2, 2, 2]
[1, 3, 3, 2, 2, 1, 1, 3]
[3, 2, 2, 2, 2, 3, 3, 2, 1, 1, 1, 2]
[1, 1, 1, 1, 2, 1, 1, 2, 1, 1]
[1, 2, 1, 1, 2, 1, 1, 1, 1, 1]
```

請依下面要求輸入參數，並將執行結果擷圖：

`print(tuplexcount(("E", "m", "b", "b", "E")))`	`[2, 1, 2, 2, 2]`
`print(tuplexcount(("1", "3", "3", "4", "4", "6", "7", "3")))`	`[1, 3, 3, 2, 2, 1, 1, 3]`
`print(tuplexcount((7, 5, 4, 4, 2, 7, 7, 2, "y", "4", "3", 5)))`	`[3, 2, 2, 2, 2, 3, 3, 2, 1, 1, 1, 2]`
`print(tuplexcount(("P", "y", "t", "h", "o", "n", "w", "o", "r", "k")))`	`[1, 1, 1, 1, 2, 1, 1, 2, 1, 1]`

print(tuplexcount(("m", "a", "t", "l", "a", "b", "d", "i", "y", "s")))	[1, 2, 1, 1, 2, 1, 1, 1, 1, 1]

執行結果擷圖：

```
print(tuplexcount(("E", "m", "b", "b", "E")))
print(tuplexcount(("1", "3", "3", "4", "4", "6", "7", "3")))
print(tuplexcount((7, 5, 4, 4, 2, 7, 7, 2, "y", "4", "3", 5)))
print(tuplexcount(("P", "y", "t", "h", "o", "n", "w", "o", "r", "k")))
print(tuplexcount(("m", "a", "t", "l", "a", "b", "d", "i", "y", "s")))
```

```
[2, 1, 2, 2, 2]
[1, 3, 3, 2, 2, 1, 1, 3]
[3, 2, 2, 2, 2, 3, 3, 2, 1, 1, 1, 2]
[1, 1, 1, 1, 2, 1, 1, 2, 1, 1]
[1, 2, 1, 1, 2, 1, 1, 1, 1, 1]
```

EX 8: 試撰寫一個 Python 函式 lsttuple(lst)，其傳遞進來的 list 裡是由 tuple 所構成的內容。函式的功能是將所傳遞進來的 list 裡每一筆 tuple 的末端值更改為 100，並且回傳。

例如:

輸入	執行結果
print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))	[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))	[(20, 100), (400, 100), (70, 100), (2, 100)]

我的作答：

請在下面欄位貼上程式碼：

```
def lsttuple(lst):
    for i in range(len(lst)):
        lst[i]=list(lst[i])

    for i in range(len(lst)):
        lst[i][-1]=100

    for i in range(len(lst)):
        lst[i]=tuple(lst[i])

    return lst
print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))
print(lsttuple([(100, 20), (140, 350, 60)]))
print(lsttuple([(210, 220, 33, 333), (400, 150, 77), (1170, 180)]))
print(lsttuple([(200,), (400, 150, 77, 1170, 180)]))
```

請依下面要求輸入參數，並將執行結果擷圖：

print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))	[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))	[(20, 100), (400, 100), (70, 100), (2, 100)]
print(lsttuple([(100, 20), (140, 350, 60)]))	[(100, 100), (140, 350, 100)]
print(lsttuple([(210, 220, 33, 333), (400, 150, 77), (1170, 180)]))	[(210, 220, 33, 100), (400, 150, 100), (1170, 100)]
print(lsttuple([(200,), (400, 150, 77, 1170, 180)]))	[(100,), (400, 150, 77, 1170, 100)]

執行結果擷圖：

In [9]:

```python
def lsttuple(lst):
    for i in range(len(lst)):
        lst[i]=list(lst[i])

    for i in range(len(lst)):
        lst[i][-1]=100

    for i in range(len(lst)):
        lst[i]=tuple(lst[i])

    return lst
print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))
print(lsttuple([(100, 20), (140, 350, 60)]))
print(lsttuple([(210, 220, 33, 333), (400, 150, 77), (1170, 180)]))
print(lsttuple([(200,), (400, 150, 77, 1170, 180)]))
```

```
[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
[(20, 100), (400, 100), (70, 100), (2, 100)]
[(100, 100), (140, 350, 100)]
[(210, 220, 33, 100), (400, 150, 100), (1170, 100)]
[(100,), (400, 150, 77, 1170, 100)]
```

EX 9: 同 EX3 的題目，試以串列解析式語法撰寫一個 Python 函式 lsttuple(lst)，其傳遞進來的 list 裡是由 tuple 所構成的內容。函式的功能是將所傳遞進來的 list 裡每一筆 tuple 的末端值更改為 100，並且回傳。

例如:

輸入	執行結果
print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))	[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))	[(20, 100), (400, 100), (70, 100), (2, 100)]

我的作答：

請在下面欄位貼上程式碼：

```
def lsttuple(lst):
    ans=[]
    [ans.append(i[:-1]+(100,)) for i in lst]

    return ans
print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))
print(lsttuple([(100, 20), (140, 350, 60)]))
print(lsttuple([(210, 220, 33, 333), (400, 150, 77), (1170, 180)]))
print(lsttuple([(200,), (400, 150, 77, 1170, 180)]))
```

請依下面要求輸入參數，並將執行結果擷圖：

print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))	[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))	[(20, 100), (400, 100), (70, 100), (2, 100)]
print(lsttuple([(100, 20), (140, 350, 60)]))	[(100, 100), (140, 350, 100)]
print(lsttuple([(210, 220, 33, 333), (400, 150, 77), (1170, 180)]))	[(210, 220, 33, 100), (400, 150, 100), (1170, 100)]
print(lsttuple([(200,), (400, 150, 77, 1170, 180)]))	[(100,), (400, 150, 77, 1170, 100)]

執行結果擷圖：

In [10]:

```
def lsttuple(lst):
    ans=[]
    [ans.append(i[:-1]+(100,)) for i in lst]

    return ans
print(lsttuple([(10, 20, 40), (40, 50, 60), (70, 80, 90)]))
print(lsttuple([(20, 20), (400, 150), (70, 80), (2, 3)]))
print(lsttuple([(100, 20), (140, 350, 60)]))
print(lsttuple([(210, 220, 33, 333), (400, 150, 77), (1170, 180)]))
print(lsttuple([(200,), (400, 150, 77, 1170, 180)]))

```

```
[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
[(20, 100), (400, 100), (70, 100), (2, 100)]
[(100, 100), (140, 350, 100)]
[(210, 220, 33, 100), (400, 150, 100), (1170, 100)]
[(100,), (400, 150, 77, 1170, 100)]
```

EX 10:試撰寫一個 Python 函式 lsttupleL(lst)，函式的功能是將所傳遞進來的 list 裡的空 tuple (empty tuple)刪除後回傳。

例如:

輸入	執行結果
`print(lsttupleL([(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')])) # ('',) is a tuple`	`[('',), ('a', 'b'), ('a', 'b', 'c'), 'd']`
`print(lsttupleL([(100, 20), (''), (140, 350, 60)])) # ('') not a tuple, it's a string`	`[(100, 20), '', (140, 350, 60)]`

我的作答：

請在下面欄位貼上程式碼：

```
def lsttupleL(lst):
    ans=[]
    for i in range(len(lst)):
        if lst[i]!=():
            ans.append(lst[i])
    return ans

print(lsttupleL([(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]))
# ('',) is a tuple
print(lsttupleL([(20, 20), (), (), (2, 3)]))
print(lsttupleL([(100, 20), (''), (140, 350, 60)]))  # ('') not a
tuple, it's a string
print(lsttupleL([(210, 220, 33, 333), (), (1170, 180)]))
print(lsttupleL([(200,), ('',)]))
```

請依下面要求輸入參數，並將執行結果擷圖：

`print(lsttupleL([(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')])) # ('',) is a tuple`	`[('',), ('a', 'b'), ('a', 'b', 'c'), 'd']`
`print(lsttupleL([(20, 20), (), (), (2, 3)]))`	`[(20, 20), (2, 3)]`
`print(lsttupleL([(100, 20), (''), (140, 350, 60)]))  # ('') not a tuple, it's a string`	`[(100, 20), '', (140, 350, 60)]`
`print(lsttupleL([(210, 220, 33, 333), (), (1170, 180)]))`	`[(210, 220, 33, 333), (1170, 180)]`
`print(lsttupleL([(200,), ('',)]))`	`[(200,), ('',)]`

執行結果擷圖：

In [13]:

```
def lsttupleL(lst):
    ans=[]
    for j in range(len(lst)):
        if lst[j]!=():
            ans.append(lst[j])
    return ans

print(lsttupleL([(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')])) # ('',
print(lsttupleL([(20, 20), (), (), (2, 3)]))
print(lsttupleL([(100, 20), (''), (140, 350, 60)]))  # ('') not a tuple, it'
print(lsttupleL([(210, 220, 33, 333), (), (1170, 180)]))
print(lsttupleL([(200,), ('',)]))

```

```
[('',), ('a', 'b'), ('a', 'b', 'c'), 'd']
[(20, 20), (2, 3)]
[(100, 20), '', (140, 350, 60)]
[(210, 220, 33, 333), (1170, 180)]
[(200,), ('',)]
```