**1. 判斷使用者是否只有輸入大小寫。**

介紹一個Python的標準模組--- unicodedata

裡面有一組函式: unicodedata.name，

這組函式可以依據輸入的字元去判斷它是屬於哪一種種類的文字，例如

>>> unicodedata.name('t')

'LATIN SMALL LETTER T'

>>> unicodedata.name('T')

'LATIN CAPITAL LETTER T'

>>> unicodedata.name(' ')

'SPACE'

>>> unicodedata.name(',')

'COMMA'

輸入小寫’t’，會顯示出'LATIN SMALL LETTER T'；

輸入大寫’T’，會顯示出' LATIN CAPITAL LETTER T’；

輸入空白間隔，會顯示出'SPACE'；

輸入','，會顯示出'COMMA'

以下設計判斷機制，首先，設計一擷取器，並設定用於擷取輸入字元後所顯示的判別字串，例如:

>>> b = unicodedata.name('T')

>>> b

' LATIN CAPITAL LETTER T’

>>> c = unicodedata.name('t')

>>> c

'LATIN SMALL LETTER T'

>>>

把上述b,c兩個變數的字串做切片，分別取位址0~18以及位址0~20，結果將如下:

>>> c[:18]

'LATIN SMALL LETTER'

>>> b[:20]

'LATIN CAPITAL LETTER'

相同的技巧運用於之後的字元輸入，設定擷取的字串切片長度為[:18]與[:20]，可用於和'LATIN SMALL LETTER' 以及 'LATIN CAPITAL LETTER'這兩個字串做邏輯比對，如果比對後回傳是 True，就表示該字元符合大寫或小寫字母，否則就必須請使用者重新輸入。

**2. 判斷字串長度**

#使用len()函數回傳字串長度，例如:

>>> a = 'ab cd'

>>> len(a)

5

**3. 移除list中的元素(以下還未測試)**

"""參考

2. 使用list方法pop删除元素

li = [1, 2, 3, 4]

li.pop(2)

print(li)

# Output [1, 2, 4]

**4. 使用set()**

set()是python的基本物件，用該物件，可以回傳list裡面所包含的元素，再搭配len()運用，可以作為後面程式解題的一個技巧。例如:

>>> b = [1,2,2]

>>> set(b)

{1, 2}

>>> len(set(b))

2

>>> c = [1,1,1]

>>> len(set(c))

1

**5. 使用max()**

可用於找尋list中最大的值，可以作為後面程式解題的一個技巧。例如:

>>> b = [1,2,2]

>>> max(b)

2

**問題描述:**

透過問題描述，我們可以重新的檢視問題的內涵，並增進與確保我們對問題的理解。重新用一段話，描述成程式開發者自身所理解的內容，例如:

**問題需求界定**

**解題策略**

**(一) 第一階段: 條件限制與前處理**

**(二) 第二階段: 字串交錯截片，連續判斷**

interlaced\_strings = [] #存放被截取的交錯字串

a\_str\_length = len(a\_str) #使用len()函數回傳字串長度，len()也可用於計算list的元素數量，所以也可透過檢測result[]內的元素數量可得知輸入的字串長度

s = a\_str\_length #用於要控制交錯字串判斷的迴圈次數

while s>k的情況下，迴圈會一直跑

result[s-k:s]

s = s-1

先判斷前k個位址內的字元是否都是大寫或小寫，

如果不是，砍掉第一個元素，累計計算變數s重新變回k

strat\_add = 0 #交錯字串初始位址，只用在判斷迴圈中

end\_add = k #交錯字串最後位址，只用在判斷迴圈中

切片策略採用分段切與組合的策略，避免因迴圈計算而耗掉效能

為避免字串長度與k交錯字串除不盡的錯誤，必須採用餘數判斷的機制

if result[k-s] == result[k-s+1] && r

else:

remove result[k-s]

s = k

以判斷前k個字元都是大寫或小寫後，判斷k+1個字元是否有反轉成小寫或大寫，如果不是，砍掉第一個元素，累計計算變數s重新變回k

result[k-1] != result[k-1]

###################

進入交錯字串判定後，就開始定位，從初始切片到交錯字串中斷的最後位址要紀錄，並把這一段字串紀錄到一個另一個list裡。

然後再砍掉第一個元素，重複繼續檢測。

直到檢測到最後一個字元為止

把記錄到的所有有交錯字串的切片麻出來測量字串長度，回報字串長度。

#使用len()函數回傳字串長度

a\_str\_length = len(a\_str)

for x in a[:4]:

算出interlaced\_strings = []裡面的每一個交錯字串長度，並存入另一個list

找出list裡面最大的數字，就是本次題目中要找的最長的交錯字串。

**範例程式碼:**