目錄

[我的步驟 2](#_Toc71231906)

[第一步解析題目要求與主管要求 2](#_Toc71231907)

[第二步google 3](#_Toc71231908)

[第三步先給出一個目前的解法 4](#_Toc71231909)

[寫程式時遇到的問題 7](#_Toc71231910)

[我的筆記 8](#_Toc71231911)

[正規表示法 8](#_Toc71231912)

[TCL 11](#_Toc71231913)

[TCL讀寫 11](#_Toc71231914)

[List 用法 12](#_Toc71231915)

[額外思考的問題 13](#_Toc71231916)

[參考網站: 14](#_Toc71231917)

# 我的步驟

## 第一步解析題目要求與主管要求

**題目:**

**第一題**

是比較簡單，只是文字處理，到網路上找一個超過一百行的文字檔，從裡面找出想要的資訊（tcl 允許 ”exec” or “sh” 來呼叫unix 系統指令，如果剛好熟awl/sed/Perl 都可以用外部指令處理，這個不犯規）

思考:

由於主管在信中已提到用regexp對未來應該比較好，因此我打算一開始就直接朝TCL+regexp的方向去google及思考。

**第二題**

比較像是control flow, 可以透過開檔案或是字串操作把想要的資料丟到另一個檔案

思考:

第一題應該是為了第二題出的，所以應該可以用第一題的程式碼來做抓資料的動作。

**解析主管要求**

主管應該是比較希望看到整個思考過程而不是只有答案，故再作的同時會不斷將自己遇過的問題以及思考的過程記錄下來。

## 第二步google

再進行題目解析之後，再去針對第一步所想做到的去google

Ex:

Regexp 正規表示法

TCL 開讀檔

TCL 字串處理

….等等

Google時碰到的問題:

1. 主要對於Regexp較不熟悉，故對於熟悉Regexp的語法花了很長一段時間學習
2. 網路上大部分Regexp的資料都是用JS寫的。
3. TCL與Regexp結合怎麼搭配使用，在網路上的的資料較少

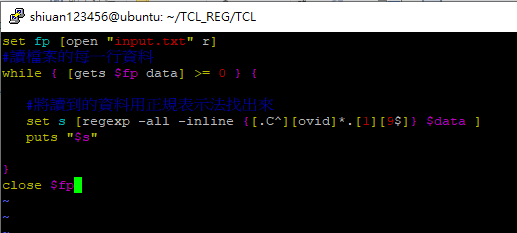
## 第三步先給出一個目前的解法

**第一題**

程式演算法:

TCL開檔案 > match想要的資料(我想到用TCL字串比對，但我研究使用regexp)

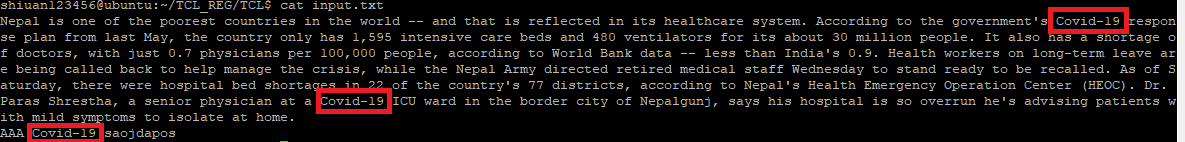
**程式碼**: (參考TCL讀寫那邊)



程式執行結果



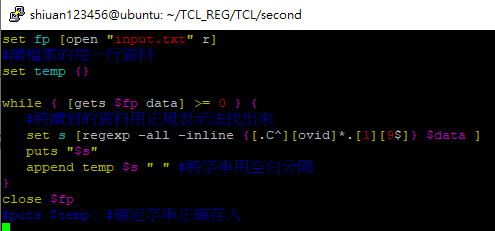
所要的文字檔



**第二題**

1. 可以透過TCL開A檔案
2. 用regexp抓想要的資料 (用第一題的)
3. 將regexp抓到的資料存起來，我的作法:先將字串串接，用append以空白分隔存到list (參考list用法)
4. 透過TCL開B檔案 > 將資料丟到B檔案

下圖為第1~3步的程式碼 ( ps:用puts去看字串是否正確存入)



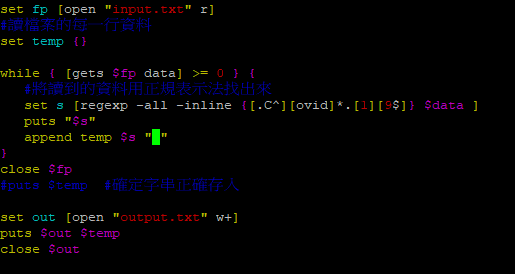
串接前



串接後



下圖為多加第4步對output.txt做讀寫





### 寫程式時遇到的問題

* 網路上大多數都是 ^B表示B開頭的字串，但在TCL要寫成 B^
* 寫Regexp的時候一定要用 [] 把 regex括起來
* 遇到兩種寫法

regexp {[C^][ovid]\*.[1][9$]} " CCCCCovid-19999" matched

puts “$matched”

set s [regexp -all -inline {[C^][ovid]\*.[1][9$]} "CCCCCovid-19999"]

puts "$s"

# 我的筆記

## 正規表示法

**Regexp 用法**

Regexp reg的規則 要被替換的字串 存到哪個變數(變1 變2 變3)

regexp把匹配整個正規表達式的子字符串賦給第一個變數

匹配正規表達式的最左邊的子表達式的子字符串給第二個變量

% regexp { ([0-9]+) \*([a-z]+)} " there is 100 apples" total num word

% puts " $total ,$num,$word"

100 apples ,100,apples

**Regsub用法**

Regsub 要替換的pattern 目標字串 要替換字串中哪一個pattern 存到哪個變數

**-all**

一直查找替换，否則只替換1次。返回替換次數。

regsub zz abbab a x # x=zzbbab

regsub -all zz abbab a x # x=zzbbzzb 返回值为2

8進制數字的數量

在JS裡面

**/ /**

包起來代表正規表式法

regexp -all {[0-7]} $string

-nocase，表示忽略字母大小寫。

**{}**

想重複的話(類似迴圈)加上這個

EX:想搜尋電話號碼 {8}代表 \d\d\d\d\d\d\d\d

/^09\d{8}$/

{2,} 最少2次

{2} 2次

{3,5} 3~5次

{3,5} 3~5 選到匹配最少次的

那如果是不定長度呢?

/^09\d+$/

**()**

表示我想知道這裡面配對到的東西它的內容是甚麼

Ex:

/^A\d+Z$/ 會印出 A12345Z

/^A(\d)+Z$/ 會印出 A12345Z 12345

**+**

意義:要找的pattern出現1次以上，並將配對到的pattern全部印出來

**注意要印出**一整個字串要先把要印的字串[]起來，+放外面

Ex:

**regexp {[xy]+} "axy" d #會將xy印出來**

^

開頭字元符合即匹配

**注意**

**如果是要比對一個字元就要把^放到後面 (如例1)**

**如果是要比對多個位元就要把^放到前面 (如例2)**

Ex:

例1 regexp {[x^y]+} “axy” d #印出x開頭的，即xy

例2 regexp {^[ax]+} “axy” d #印出ax開頭的，即axy

**\***

出現0個以上，並將配對到的字全部印出來

regexp {x\*} "baxxxy" d # 這裡會出錯，請不要將\*用在只有一個pattern時

regexp {xx\*} “baxxxy” d # 會印出xxx

.

比對任意一個字元除了換行符號

**/s**

可以配對到任何空白

**/k**

匹配非字母和數字

**\.**

跳脫字元可以想成把.變成一般的.

**-nocase**

忽略大小寫

**範例:**

(0x)表示0x一起作爲一項

?表示前一項(0x)可以出現0次或多次，

[0-9a-fA-F]表示可以是任意0到9之間的單個數字或a到f或A到F之間的單個字母，(可用\w代替)

＋表示像前面那樣的單個數字或字母可以重複出現一次或多次。

regexp {^((0x)?[0-9a-fA-F]+|[0-9]+)$} ab

0xAA

**\r**

回車即回到第一個字

**\n**

換行即跳下一行

所以我們按Enter等於 \r\n (不能\n\rs)

\s \S

\s: 所有空白代表空白 tab 換頁 換行

\S: 代表非空白字元等同 [^\s]

**要怎麼把比對出來的值取出給某個變數?**

Set s [正規表示法規則]

Puts “$s”

## TCL

**計算字串長度**

set s "hello world"

puts "s length = [string length $s]"

**從字串中取出某幾個字**

set d "Hello World"

puts [string range $d 4 6]

**匹配字符串**

**Match**

<http://tw.gitbook.net/tcl/tcl_strings.html>

### TCL讀寫

寫入檔案:

w+: 寫入檔案，若沒有該檔案名稱則創檔

set fp [open "input.txt" w+]

puts $fp "test"

close $fp

讀取檔案

set fp [open "input.txt" r]

set file\_data [read $fp]

puts $file\_data

close $f

讀取檔案 (一行一行讀)

Gets語法:

Gets 檔案名 檔案每一行資料並儲存之

set fp [open "input.txt" w+]

puts $fp "test\ntest"

close $fp

set fp [open "input.txt" r]

while { [gets $fp data] >= 0 } {

puts $data

}

close $fp

### List 用法

Append

做字串分隔: append 串完的值存到這 $要串的值 “,” (ps:以逗號為分隔)

舉例:

set var 0 (串字串的話 set var {} )

for {set i 1} {$i<=10} {incr i} {

append var “,” $i

}

puts $var

# Prints 0,1,2,3,4,5,6,7,8,9,10

創建list: set s{值} or set s [list 值]

列表長度: Length $變數名稱

puts [llength $var]

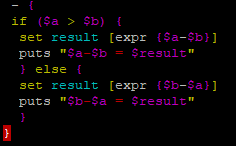
插入索引項目: linsert $變數名稱 第幾個位子 要插入的值

set var [linsert $var 3 black white]

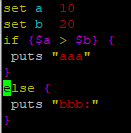
將多個list合成一個list: contact list 列表1 列表2

## TCL遇到的問題

1.if的括號是大括號而不是小括號



2. '{' '}'的前後都要空一行



## 額外思考的問題

1. 有沒有辦法一次將所有規則寫入，使我不用一直改。

然後利用()去括出我要的規則並用puts印出。

( ) : capture grouping

1. 有沒有辦法將底下的JS+Reg改寫成TCL+Reg

var re = /^.+@(.+?)\./

for(let email of emails) {

var result = email.match(re)

console.log(result[1])

}

1. 也可以列出總共有幾個一樣的訊息
2. 所要的資料分別在第幾行

## 參考網站:

Regex最淺顯易懂的

<https://blog.techbridge.cc/2020/05/14/introduction-to-regular-expression/>

Regex互動式

<https://refrf.shreyasminocha.me/en/chapters/alternation>

Regex基本符號的介紹

<https://reurl.cc/8ygxq4>

Regex特殊字元參考網站

<https://www.fooish.com/regex-regular-expression/escaped-characters.html>

主管說的exec

<https://5xruby.tw/posts/15min-regular-expression>

TCL 基本指令

<https://mouse.oit.edu.tw/htdocs/Hope/199803/tcl.htm>

TCL一些檔案處理可以看這裡

<http://chiumog.blogspot.com/2015/02/tcl_46.html>

簡書: 使我了解{}和[]怎麼用

<https://www.jianshu.com/p/d530fe024ffd>