

題目一： 多重字串搜尋

說明：

讀入關鍵字與文本後，在文本內尋找出現最多次的關鍵字與次數，若是出現最多次數相同，請依照字典排序由小至大印出，關鍵字與文本以-----為分隔。關鍵字大小寫需一致，例如 DS 與 ds 視為不同字。注意標點符號與字中字，如 cloud 與 cloud. 為同一個字，關鍵字 water 找到 watermelon 不能算。

Input:

[關鍵字]

[關鍵字]

[關鍵字]

[文本]

Output:

[出現次數最多的關鍵字] [次數]

Sample Input:

a

cloud

The bird wishes it were a cloud.

The cloud wishes it were a bird.

Sample Output:

a 2

cloud 2

題目二： 二元搜尋樹

說明：

讀入資料後，建立一個二元搜尋樹(Binary Search Tree)，每個節點存放 keyword 與其次數。資料每列包含一個運算子與一個單字，單字為純英文，大小以 ASCII 大小為順序排列，如 $b > a$ ， $a > A$ ，運算子+代表該單字於 BST 中 +1，算子-代表該單字於 BST 中-1。

- +apple 表示新增一個 apple，若 apple 已存在則增加其個數，無則為 apple 新增一個節點，並將個數設為 1。
- -apple 表示刪減一個 apple，若 apple 已存在則減少其個數，減完為 0 時則刪除該點。

最後以 preorder traversal 印出所有單字與其個數。

Input:

[運算子] [單字]

[運算子] [單字]

[運算子] [單字]

Output:

[單字] [個數]

[單字] [個數]

[單字] [個數]

Sample Input:

+apple

+banana

+lemon

+apple

+apple

-lemon

+apple

-apple

+lemon

+lemon

Sample Output:

apple 3

banana 1

lemon 2

題目三：最短路徑

說明：

讀入路徑總數後，依照總數讀入點編號、長度、點編號之資料，如資料 1 2 3 即是點 1 與點 3 之距離為 2，點編號與長度皆為正整數。利用演算法，算出 X 點到 Y 點的最短路徑長。

路徑皆為雙向道，兩點之間不存在兩條以上的路徑。

若是兩者間不存在路徑，則印出 404。

Input:

[路徑數]

[點編號] [長度] [點編號]

[點編號] [長度] [點編號]

[點編號] [長度] [點編號]

[X 點] [Y 點]

Output:

[X 點到 Y 點的最短路徑長]

Sample Input:

3

1 1 2

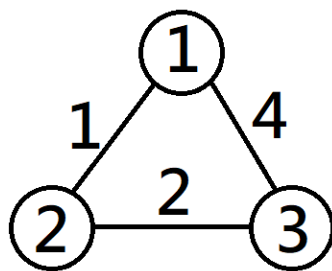
2 2 3

1 4 3

1 3

Sample Output:

3



Hint：建議使用 Dijkstra 演算法。