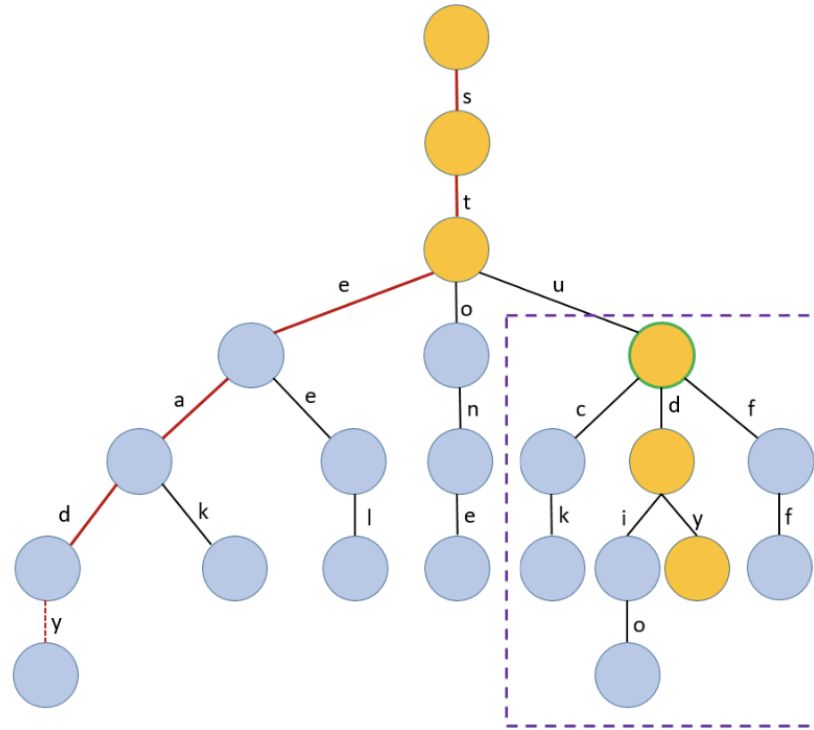


Prefix tree 是一種建立字典時常用的資料結構，屬於一種多元樹（n-ary tree）。如下圖所示，有相同前綴的字會被儲存在同一個子樹之中。也因為如此，如果要在這個樹之中尋找一個詞（word），只需要根據它的字（letter），一一向下尋找即可。例如若需要尋找 study 這個詞，便會經過下圖中黃色的 node。對於一個具有 n 個詞的 prefix tree，尋找一個詞的複雜度平均而言可以落在 $O(\log n)$ 。



同理，如果要在這個樹之中加入新的詞，就從這個詞的第一個字開始，一一尋找在對應的層級中是否已經具有代表這個字的 child，若有則沿著這個 arc 往下走一層到下一個 node，若沒有則在現在所在的 node 新增一個 child，代表這個字。對這個詞的每一個字都進行這個過程後，在代表最後一個字的 node 進行標記，代表到這個 node 為止的字可以組成一個詞。例如若在這個樹之中加入 steady 這個字，就會經過圖中紅色的邊，並且於最左方的 lead node 新增一個 child（圖中虛線處）。

在本題中，我們將會給定多個詞，以及一些前綴，請將這些詞放入一個 prefix tree 之中，並將包含這些前綴的詞尋找出來。例如若要在上圖的樹中尋找包含 stu 這個前綴的所有詞，只需要尋找代表 stu 的這個 node（圖中綠框者），並對其 subtree（圖中紫框處）進行 preorder traversal，即可依照字典順序取得包含這個前綴的所有詞。

輸入輸出格式

系統會提供 20 組的測試資料，每組測試資料裝在一個檔案裡。在每個檔案之中，會有 3 行。第一行會有 2 個整數 m 及 n ，二者以空白字元隔開，分別代表詞的總數，以及要搜尋的前綴總數。第二行包含 m 個字串 s_1 至 s_m ，字串之間以空白字元分隔。第三行包含 n 個字串 w_1 至 w_n ，字串之間以空白字元分隔。所有字串都不會重複，且只會包含英文大小寫字母。其中 $1 \leq m \leq 10^8$ ， $1 \leq n \leq 1000$ ，且任一字串 (word) 的長度都不會超過 10000。

讀入以上資料後，請在第 i 行印出在 s_1 到 s_m 之中，字首包含 w_i 的那些字串。字串之間以一個空白字元分隔，印出時請依照字典順序印出。

舉例來說，如果輸入是

```
8 2
steel steak stuff study stone studio stuck stead
stu ste
```

則輸出應該是

```
stuck studio study stuff
stead steak steel
```