

Sudo code:

- 1.建立一個 `node class` 其中包含了左連結 右連結 編號 擁有的字母等資料。
- 2.讀入第一個空白前的所有字元 結合成一個數字，代表開始的編號。
- 3.接著持續讀入字元結合成數字，遇到空白後，若為第一個 `node`，建立一個新的 `node`，若非第一個 `node`，接在最後一個 `node` 的後面，建立雙向的連結，形成雙向的 `link list`。
- 4.遇到負號後，把最後一個 `node` 跟第一個 `node` 做連結，形成 `circular link list`。
- 5.建立一個指標指向開始的編號所在的 `node`
- 6.跳過負數
- 7.持續讀入字元，若為字母，根據方向把字母加入當前所在 `node` 的 `word` 前面或後面，接著觀察下一個字元，若為換行，代表一輪已完成，印出當前 `node` 所擁有的所有字母，`delete` 掉這個 `node`，最後根據方向把當前 `node` 改成左邊或右邊的 `node`。
- 8.最後 `link list` 裡面會剩下兩個 `node`，而目前 `node` 指向的是贏家，所以先把另一個 `node` 印出，`delete`，最後印出當前 `node`，`delete`。

時間複雜度分析:

我的 `code` 會從頭到尾對每個字元進行處理，因此若總字元數為 n ，大致上的時間是 $O(n)$ ，而若英文字母總數為 m ，代表額外花了 m 次時間在印出所有字母，總共的時間大約是 $O(m+n)$ 。