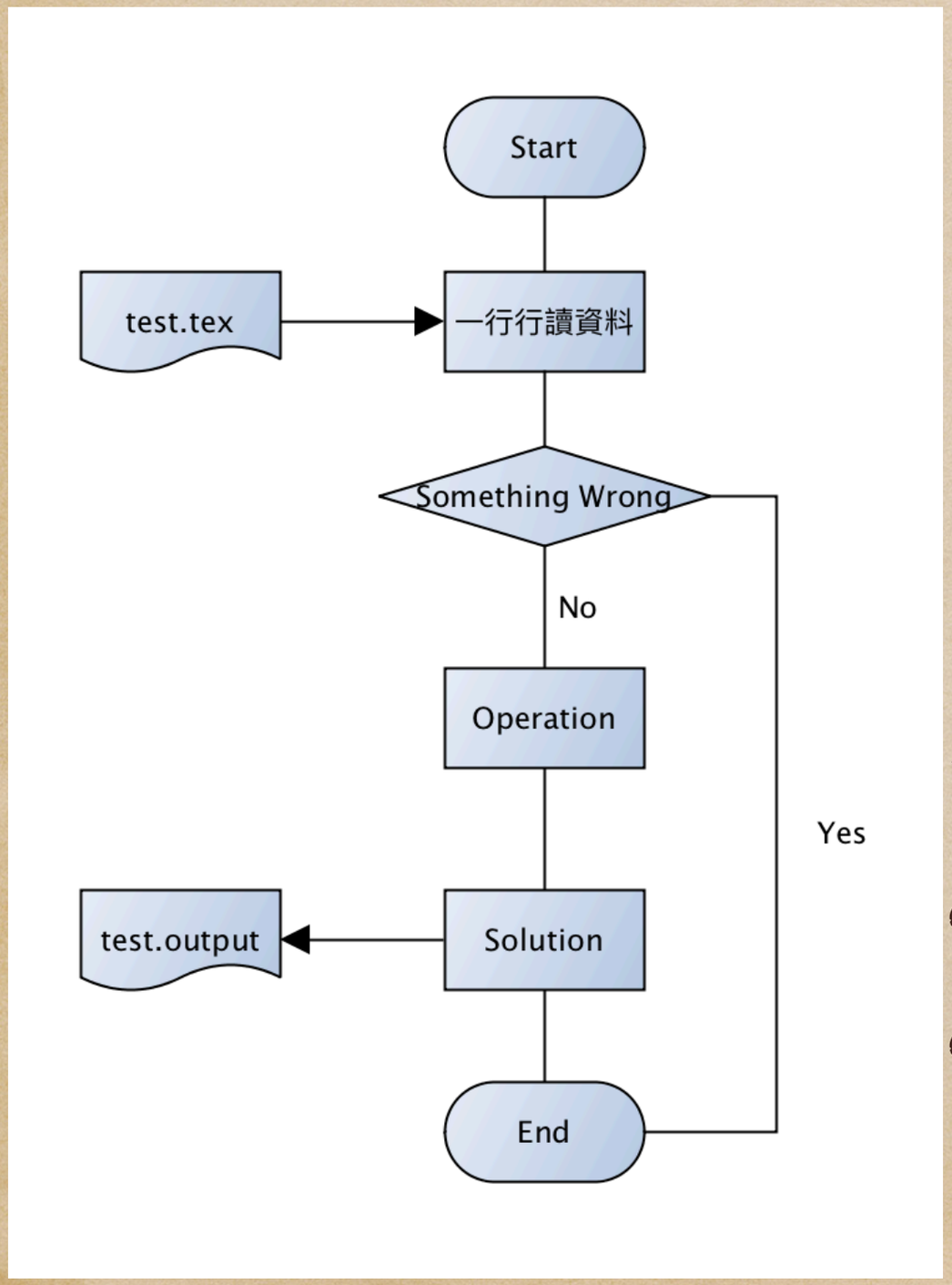
整個程式的主要架構大概如圖

Operation的運作方式在下面幾頁會再詳細解說



Something wrong

主要為檔案無法成功開啟時

包括讀取或寫入檔

發生此種錯誤時程式直接終止

Function call順序

main->operation->recognize->

operation->Insert...等->

operation->main

副函式說明

1. char* operation(char []); 主要運算函式(見下頁)
2. int recognize(char []); 判斷tag的正確性、種類
3. char* Insert(char []); 進行insert
4. char* Proper(char []); 進行Proper
5. char* Upper(char []); 進行toupper
6. char* Lower(char []); 進行tolower
7. char* Reverse(char []); 進行reverse

Operation運作方式

1. 主要是仰賴stack進行判斷標籤是否有匹配
2. 在程式剛開始時即會尋找”<“，找到後就會送入recognize判斷是何種標籤
3. 另外，同時間會將文字放到一個stackA中，以變之後進行修改
4. recognize會辨識出其屬於何種tag
5. 回傳後若是開頭的(<xxx>)那就將其放入另一個stackB
6. 若是結尾(</xxx>)，那就看看stackB的top是否匹配，不配表Interleaved paired tag
7. 若匹配則將stackA中任何一個大於等於現在top的字串送入副程式進行轉換
8. 副程式包括Insert、 Proper、 Upper、Lower、 Reverse（詳見上頁）
9. 其中比較特別的是Insert，他會將其直接插在stackA top字串的最後面
10. 其餘在處理完畢之後都會++top再存
11. 如果判斷到<Newline>跟Insert的處理一樣，會加在top字串的最後面
12. stackA和stackB之所以共用top是因為這樣比較容易處理字串轉換（下頁有說明原因）
13. 每處理完一個tag便會尋找”>”，以便繼續搜尋下一個”<“，使其不重複
14. 利用stackB即可判斷其是否有出現error
15. 若回傳有error，則會告知使用者哪一行有問題。

Main issues and proposed solutions

剛開始看到題目時其實毫無頭緒
但是想到以前練習過的題目(括號匹配問題)
就想到或許可以利用stack來進行原本將兩個stack的top分開寫
但這樣會發生，同樣的文字不知道該如何同時被兩個tag影響的問題
(ex:<xxx><yyy>kkk</xxx></yyy>這時就不知道kkk如何轉換了)
再稍微想了一下發現如果xxx和yyy之間沒有文字也沒關係
直接把stack留空就好
這樣只要把大於現在tag的文字都進行轉換就可以排除這個問題了

其他的主要就是在進行文字處理了
而這方面我覺得跟以前打下的基礎很有關係
如果以前基礎好的，寫起來就很輕鬆

Your thoughts after this assignment

自己在寫這份作業時有用git進行檔案版本管理

真心覺得這太方便了

尤其在這種很多副程式的時候

如果不小心有個副程式寫壞

可以利用git回復舊版本就方便許多

有聽說有同學在寫這份作業時電腦突然當機

那個時候心情應該會超差的...

所以隨時善用git push是很重要的XD

另外寫完這份作業

說真的感覺比較像是大一程設要寫的東西

感覺跟系統程式扯不上關連:(

寫完雖然會覺得很開心，但其實內心還是有點空虛...

Possible extensions

感覺這程式其實可以延伸成html的tag

比較不確定的是C可不可以控制字型之類的

不過顏色是確定可行的

我覺得如果做這樣的延伸應該會蠻有趣的

另外我覺得也可以在tag上面加權重

主要用處是讓tag有優先順序

就不會出現像現在這種外面的tag會蓋掉裡面tag的情況

權重小的就不能蓋掉權重大的tag