

程式語言作業

Description

學會 Parser 之後，是時候動手設計一個簡單的編譯器了。請同學使用 Lex 和 Yacc 編譯一段簡單的程式碼，這段程式碼只會包含 3 種語法，規則如下：

類型	說明
int 變數名稱 = 整數;	assignment statement 指定敘述 宣告一個變數其值為整數
int A = 10;	※變數名稱只能包含英文字母、數字和底線且第一個字不能是數字
if(變數 1 > 變數 2) { ... }	conditional expressions 條件判斷式 檢查如果變數 1 大於變數 2 則執行大括號的內容
if(A > B) { printf("A is Win"); }	※只實作大於 (>) 的比較，如果輸入為其他比較運算子 (< 、 == ...) 請當作錯誤處裡 ※如果變數在此之前沒有被宣告過，請當作錯誤處裡
printf("字串");	standard output 標準輸出 印出字串
printf("A is Win");	※printf 函式的參數只能是字串，如果輸入為變數名稱、整數等其他非字串的參數，請當作錯誤處裡

請依據上表配合 Lex 和 Yacc 編譯程式碼，輸入將遵循以下規則：

1. 正確輸入只會有 3 行。
2. 正確輸入必定由 2 個指定敘述、1 個條件判斷式和 1 個條件為真時執行的標準輸出所組成。
3. 任何不符合上述規則的輸入都應當作錯誤處裡。

輸出須遵照以下幾點：

1. 正確輸入的程式碼應在條件判斷式成立時輸出一行字串，並在最後輸出一行“
Compiled successfully”。
2. 對於所有錯誤的輸入統一輸出一行“ Invalid input”。

Sample Input

```
int A = 10;  
int B = 5;  
if(A > B) { printf("A is Win"); }
```

Sample Output

A is Win

Compiled successfully

Sample Input

```
int LEX123 = 123;  
int yacc_is_not_easy = -50;  
if(yacc_is_not_easy > LEX123) { printf("Hello World!"); }
```

Sample Output

Compiled successfully

Sample Input

```
int A = 10;  
int B = 5;  
if(C > D) { printf("C is Win"); }
```

Sample Output

Invalid input