

程式語言作業 4

Description

Type checking 乍看是個簡單的功能，其實實作起來也不容易。助教設計了一個只有兩種型別的語言，請同學幫忙寫一個 type checker。

語法	範例	說明
宣告	<pre>int A; float B; float C;</pre>	<ul style="list-style-type: none">● 由型別、變數名稱和一個分號組成● 型別只有 int 、 float 兩種，分別代表整數和浮點數型別● 變數名稱只會是單一的大寫英文字母 ([A-Z])
賦值	<pre>A = 10; B = 1.2 + 4.0; C = B - 8.77;</pre>	<ul style="list-style-type: none">● 等於的左半邊是已宣告的變數，右半邊可能是單一數值或變數，也可能是兩個數值或變數中間有一個運算符號，最後加上分號。● 不會有三個以上的數值運算● 右半邊所有的數值或變數型別必須和左半邊一致● 運算符號只會有加減法● 數值中有 . 代表浮點數，沒有則代表是整數
輸出	<pre>print A; print C; print B;</pre>	<ul style="list-style-type: none">● 由 print 、變數和分號組成

以上是這個語言的語法說明，而你的 `type checker` 需要能夠檢查出下列幾種錯誤。

類型	範例	輸出
重複宣告	<pre>int A; A = 5; float A;</pre>	Variable A redeclaration. Error at line 3.
變數未定義	<pre>int A; print C;</pre>	Variable C undefined. Error at line 2.
	<pre>int A; A = 123 + D;</pre>	Variable D undefined. Error at line 2.
型別錯誤	<pre>int B; B = 2.5;</pre>	Type error. Error at line 2.
	<pre>float A; A = 10.5 + 3;</pre>	Type error. Error at line 2.
	<pre>float A; int B; A = 4.0 - B;</pre>	Type error. Error at line 3.

請依據上表配合 `Lex` 和 `Yacc` 撰寫程式碼，並注意以下幾點：

1. 輸入的語法必定正確，這次作業著重在找出上面的三種錯誤
2. 輸出請包含錯誤發生在第幾行
3. 遇到第一個錯誤就停止檢查

Sample Input

```
int A;  
int B;  
A = 5;  
B = 2.5;  
print C;
```

Sample Output

Type error. Error at line 4.

Sample Input

```
float B;  
B = 2.2 + 3;  
B = 2.2 + 3.0;  
int A;  
print B;
```

Sample Output

Type error. Error at line 2.

Sample Input

```
float A;  
int B;  
A = A;  
int A;  
B = A;
```

Sample Output

Variable A redeclaration. Error at line 4.

Sample Input

```
int A;  
A = 123 - D;
```

Sample Output

Variable D undefined. Error at line 2.