Compiler Laboratory #2

本實驗目的是計算 first-set, follow-set 以及 predict-set。

輸入為由 standard input 餵入的一個 grammar，規範如下：

- 當 terminal 是 token class 時，表示方式為 identifier；identifier 的規範即實驗一的。比如：`name`, `int_number`。
- 當 terminal 是 lexeme 時，表示方式為 string；string 的規範即實驗一的。比如：`"if"`, `"+"`。
- Nonterminal 的表示方式為 identifier；identifier 的規範即實驗一的。
- 對一個 nonterminal，其 production rules 必須寫在一起。格式如下：

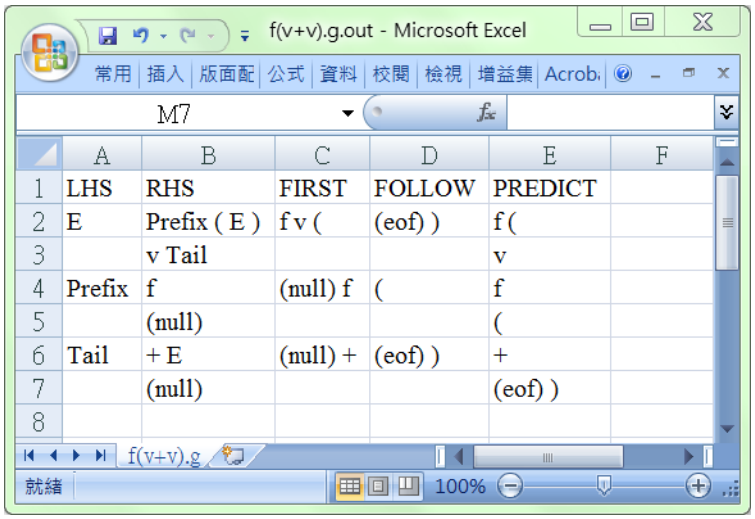
  nonterminal `:` *rule_1* `|` *rule_2* `|` ... `|` *rule_n* `;`

  其中 `:` `|` `;` 是 meta-characters。
- 整個 grammar 第一條 production rule 的 nonterminal 即為 start symbol。
- 依照上述規則，一個 identifier 可能為 nonterminal 或是 token class。當一個 identifier 曾經出現在 production rule 的 left-hand-side，則視其為 nonterminal；否則視其為 token class。
- 建議你可以修改實驗一的 scanner 程式，來讀入輸入的 grammar。
- 本實驗的輸入不需要考慮不合理、不合法的 grammar 寫法。

底下為輸入範例：

```
E       : Prefix "(" E ")"
        | v Tail
        ;
Prefix  : f
        |                 <-- 代表 Prefix → λ
        ;
Tail    : "+" E
        |
        ;
```

輸出結果假如可以用表格表示，使用方便性會較佳；如下列 excel：



	A	B	C	D	E	F
1	LHS	RHS	FIRST	FOLLOW	PREDICT	
2	E	Prefix (E)	f v (	(eof))	f (	
3		v Tail			v	
4	Prefix	f	(null) f	(	f	
5		(null)			(	
6	Tail	+ E	(null) +	(eof))	+	
7		(null)			(eof))	
8						

為達成此一目的，本實驗輸出為送到 standard output 用 tab 區隔欄位的若干行，範例如下：

```
LHS ➢ RHS ➢ FIRST ➢ FOLLOW ➢ PREDICT
E ➢ Prefix ( E ) ➢ f v ( ➢ (eof) ) ➢ f (
 ➢ v Tail ➢ ➢ ➢ v
Prefix ➢ f ➢ (null) f ➢ ( ➢ f
 ➢ (null) ➢ ➢ ➢ (
Tail ➢ + E ➢ (null) + ➢ (eof) ) ➢ +
 ➢ (null) ➢ ➢ ➢ (eof) )
```

詳細規範如下：

- 第一行必須是標題。
- 第二行以下是各個 production rule，次序同輸入。
- 一個 nonterminal 的第二個以上的 production rules，不用寫出 LHS。
- 一個 nonterminal 僅須在其第一個 production rule 那行，寫出其 first-set 以及 follow-set。
- 每個 production rule，都須寫出其 predict-set。
- Lexeme 不用印出左右雙引號。
- λ 與 $ 分別印成 `(null)` 與 `(eof)`。
- 同一格有兩個以上的 symbols，中間用一個空白隔開。此外不要加空白。
- FIRST, FOLLOW, PREDICT 那三種 sets，元素必須排序。印出的次序為：特殊符號 (λ 與 $)、token classes (依一般 ASCII 字串次序排列)、lexemes (依一般 ASCII 字串次序排列)。比如，{ "if", int_number, float_number, "+", "*", $ } 印出：

  ```
  (eof) float_number int_number * + if
  ```

本實驗可用任一語言開發，如 C, C++, C#或 Java。本實驗鼓勵利用任一 open source 或學校有版權的 IDE 協助開發，如 Eclipse, NetBeans 或 Visual Studio；但仍須可在 command mode 執行。比如附檔的 `f(v+v).g.out` 是如此產生的：

```
java -jar compilerTools.jar < f(v+v).g > f(v+v).g.out
```

請交出一個 zip 或 rar 檔，結構必須為 (對學號 b96902888 的同學而言)：

```
lab2\
└── b96902888\
    ├── 程式原始檔與執行檔或整個 IDE project
    └── IWantToSay.txt（請說明編譯以及執行方式）
```

解壓縮後，第一層必須是 lab2。第二層必須是學號，英文字母小寫。