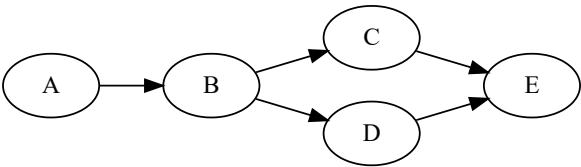


Boss Attack 01 四則運算計算機

Problem Description

小明的老闆希望小明可以交出一個計算機，無奈小明不會寫程式，且只剩下三小時的時間老闆就要看成果了。請好心的你幫助小明做出一個計算機，讓他不要被老闆炒魷魚，小明會很樂意給你學分作為回報。

這個題目總共分成 5 個小題，其相依順序如下圖：



請你依上圖順序作答。

Grammar

```
%token NUM      : ([1-9][0-9]*)|0
%token PLUS     : `+`
%token MINUS    : `-`
%token MUL      : `*`
%token DIV      : `/`
%token LPR      : `(`
%token RPR      : `)`

<expr>          := <term> <expr_tail>

<expr_tail>     := PLUS <term> <expr_tail>
                  | MINUS <term> <expr_tail>
                  | λ

<term>          := <factor> <term_tail>

<term_tail>     := MUL <factor> <term_tail>
                  | DIV <factor> <term_tail>
                  | λ

<factor>        := LPR <expr> RPR
                  | PLUS NUM
                  | MINUS NUM
                  | NUM
```

A. Scanner (Problem ID : 107BA1_A)

在編譯器中，token是組成程式的最小單位，需要由 Scanner 先把 input text 按照規則轉為 Token，才能讓 Parser 進行處理。請依照題目規則，試著寫一個 Scanner 來取得 Tokens 並印出。

Input Format

輸入一條運算式，每筆測試資料只會有一個運算式，但其中可能會夾雜空格或換行。

此題的測試資料不會有錯誤的測試資料

此題的測試資料會在 1000 字元內(不包含換行號)

Output Format

- 請在切割後輸出其 Token 種類，例如 + 則輸出 PLUS。
- 若為數字，需附上其數值，並以一個空白做為區隔。例如 0 則需輸出 NUM 0，以此類推。
- 每個token輸出後請以 \n 分隔。例如

```
NUM 0
PLUS
// there is a new line here
```

Sample Input 1

1+2

Sample Output 1

```
NUM 1
PLUS
NUM 2
```

Sample Input 2

```
( 1+
  - 2
)
```

Sample Output 2

```
LPR
NUM 1
PLUS
MINUS
NUM 2
RPR
```

Sample Input 3

```
(1  +
    2 *    3
  / 4      )
)
```

Sample Output 3

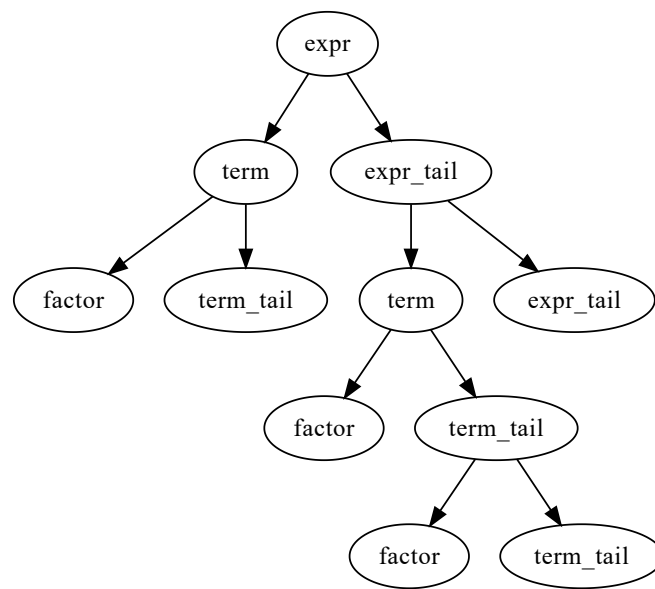
```
LPR
LPR
NUM 1
PLUS
NUM 2
MUL
NUM 3
DIV
NUM 4
RPR
RPR
```

B. Recursive Descent Parser (Problem ID : 107BA1_B)

Parse Tree 是用來表示分析的步驟，他會將分析時走過的 Non-Terminal 與 Terminal 都表示出來。而 Parse Tree 最常被用來檢查 Grammar 是否為 Ambiguous Grammar，若一個 Grammar 能畫出兩棵不同的 Parse Tree，則為 Ambiguous Grammar。

請以 **Recursive-Descent-Parsing** 的方式實作題目的 Grammar。在分析輸入後，以前序^[1]的方式印出其走過的 Non-Terminal 之方式來模擬 Parse Tree 的生成。

- 表達式 1+2*3
- Parse Tree



```

expr
term
factor
term_tail
expr_tail
term
factor
term_tail
factor
term_tail
expr_tail

```

Input Format

輸入一條運算式，每筆測試資料只會有一個運算式，但其中可能會夾雜空格或換行。
 此題的測試資料不會有錯誤的測試資料
 此題的測試資料會在 **1000** 字元內(不包含換行號)

Output Format

依照 Grammar 分析其走過的 Non-Terminal 並印出。例如若走過 <expr> 則印出 `expr`。
 每個 Non-Terminal 印出後須以 `\n` 做為區隔。

Sample Input 1

```
1+2
```

Sample Output 1

```

expr
term
factor
term_tail
expr_tail
term
factor
term_tail
factor
term_tail
expr_tail

```

Sample Input 2

```
( 1+
  - 2
)
```

Sample Output 2

```
expr
term
factor
expr
term
factor
term_tail
expr_tail
term
factor
term_tail
expr_tail
term_tail
expr_tail
```

C. 計算數值(Evaluation) (Problem ID : 107BA1_C)

利用先前所實作的 Scanner 與 Parser，試著運算出算式的結果並印出。

Input Format

輸入一條運算式，每筆測試資料只會有一個運算式，但其中可能會夾雜空格或換行。
此題的測試資料不會有錯誤的測試資料
此題的測試資料會在 **1000** 字元內(不包含換行號)

Output Format

以**整數**方式印出算式結果，並以 `\n` 做結尾。

Sample Input 1

1+2

Sample Output 1

3

Sample Input 2

```
1+
  -2*  (
    3-  + 15)
  /-1    ++1
```

Sample Output 2

-22

Sample Input 3

```
(1  +
    2 *    3
  / 4    )
)
```

Sample Output 3

2

D. 錯誤處理 (Error Handling) (Problem ID : 107BA1_D)

在各種不同階段會有不同的錯誤產生。舉例來說，在 Scanner 階段，若是有無法辨認的 Token，則會造成 unrecognized token 的錯誤發生。在 Parser 階段，若有不合乎 Grammar 的輸入，會造成 unexpected token 的錯誤。而在運算階段，則會有 divide by zero 的錯誤發生。試著改寫前面

所完成的程式，來避免以上的錯誤發生。

Input Format

輸入一條運算式，每筆測試資料只會有一個運算式，但其中可能會夾雜空格或換行。可能會有 Token 錯誤、Grammar 錯誤或是除以 0 的錯誤發生。
此題的測試資料會在 **1000** 字元內(不包含換行號)

Output Format

輸出分為下列五種情況，且每行印出後須以 \n 作為結尾。

- 1. 沒有錯誤：Correct.
 - 2. Token錯誤：Error: Unrecognized token 'XXX'.
 - 3. Grammar錯誤：Error: Unexpected token 'XXX'.
 - 4. 非預期情況遇到 EOF：Error: Unexpected token 'EOF'.
 - 5. 除以 0：Error: Divided by Zero.
- 一個 **expression** 有多個錯誤的狀況下，只輸出一種錯誤，輸出優先順序如下
 - (1) Error: Unrecognized token 'XXX'.
 - (2) Error: Divided by Zero.
 - (3) Error: Unexpected token 'XXX'.

Sample Input 1

1+2

Sample Output 1

Correct.

Sample Input 2

1++-2

Sample Output 2

Error: Unexpected token '-'.

Sample Input 3

((1+2)

Sample Output 3

Error: Unexpected token 'EOF'.

Sample Input 4

1+1/0

Sample Output 4

Error: Divided by Zero.

Sample Input 5

1+
2 *-3 + 4
-5

Sample Output 5

Error: Unrecognized token '#'.

E. 計算與錯誤處理 (Combine All) (Problem ID : 107BA1_E)

試著以前面所做的程式，做一個能進行錯誤處理並進行運算的計算機。

Input Format

輸入一條運算式，每筆測試資料只會有一個運算式，但其中可能會夾雜空格或換行。可能會有 Token 錯誤、Grammar 錯誤或是除以 0 的錯誤發生。

此題的測試資料會在 **1000** 字元內(不包含換行號)

Output Format

輸出分為下列五種情況，且每行印出後須以 `\n` 作為結尾。

1. 沒有錯誤：以**整數**方式印出算式結果。
 2. Token 錯誤：Error: Unrecognized token 'XXX'.
 3. Grammar 錯誤：Error: Unexpected token 'XXX'.
 4. 非預期情況遇到 EOF：Error: Unexpected token 'EOF'.
 5. 除以 0：Error: Divided by Zero.
- 一個 **expression** 有多個錯誤的狀況下，只輸出一種錯誤，輸出優先順序如下
 - (1) Error: Unrecognized token 'XXX'.
 - (2) Error: Divided by Zero.
 - (3) Error: Unexpected token 'XXX'.

Sample Input 1

1+2

Sample Output 1

3

Sample Input 2

1+/2

Sample Output 2

Error: Unexpected token '/'.

1. 如果一個 Non-Terminal 用一個函數實作的話，在進入函數時印出的順序便是前序。 