本程序运行需要输入对应**文法的 txt 文件路径**以及需要进行语法分析的对应的**字符串**
运行后的待输入状态如图：

```
请输入文法文件目录：(如果不存在默认为SLR.py文件同目录下的grammar.txt，如果grammar.txt不存在则会自动创建)：
grammar.txt默认文法为：["E -> E + T","E -> T","T -> T * F","T -> F","F -> ( E )","F -> id"]

请输入需要语法分析的字符串(字符间以空格隔开)：
id + id * id * ( id + id )
```

Tip：如果文法目录为空，系统则会自动检索本程序 py 文件同目录下的 grammar.txt 文件，如果依然不存在则会自动按照["E -> E + T","E -> T","T -> T * F","T -> F","F -> (E)","F -> id"]文法进行创建 grammar.txt 文件。
Tip：字符串的输入则要以空格隔开每一个字符，文法文件中的文法也是同理。
Tip：截图以默认文法：["E -> E + T","E -> T","T -> T * F","T -> F","F -> (E)","F -> id"]
字符串：id + id * (id + id)生成。
Tip：本程序由于纯手写可能部分算法执行效率略低，请谅解！

运行后程序会在控制台上显示：可行前缀 DFA 集合，DFA 构建预测分析表，对应归约集合，移进-归约过程以及结果字符串（结果字符串也会保存在 SLR.py 同目录下 result.txt 文件中）。对应结果如下图：

```
可行前缀DFA集合如下：
0  [['E1', '->', '·', 'E'], ['E', '->', '·', 'E', '+', 'T'], ['E', '->', '·', 'T'], ['T', '->', '·', 'T', '*', 'F'], ['T', '->', '·', 'F'], ['F', '->', '·', '(', 'E', ')'], ['F', '->', '·', 'id']]

1  [['E1', '->', 'E', '·'], ['E', '->', 'E', '·', '+', 'T']]

2  [['E', '->', 'T', '·'], ['T', '->', 'T', '·', '*', 'F']]

3  [['T', '->', 'F', '·']]

4  [['F', '->', '(', '·', 'E', ')'], ['E', '->', '·', 'E', '+', 'T'], ['E', '->', '·', 'T'], ['T', '->', '·', 'T', '*', 'F'], ['T', '->', '·', 'F'], ['F', '->', '·', '(', 'E', ')'], ['F', '->', '·', 'id']]

5  [['F', '->', 'id', '·']]

6  [['E', '->', 'E', '+', '·', 'T'], ['T', '->', '·', 'T', '*', 'F'], ['T', '->', '·', 'F'], ['F', '->', '·', '(', 'E', ')'], ['F', '->', '·', 'id']]

7  [['T', '->', 'T', '*', '·', 'F'], ['F', '->', '·', '(', 'E', ')'], ['F', '->', '·', 'id']]

8  [['F', '->', '(', 'E', '·', ')'], ['E', '->', 'E', '·', '+', 'T']]

9  [['E', '->', 'E', '+', 'T', '·'], ['T', '->', 'T', '·', '*', 'F']]

10 [['T', '->', 'T', '*', 'F', '·']]

11 [['F', '->', '(', 'E', ')', '·']]
```

DFA构建预测分析表如下：

	+	*	(	)	id	$	E	T	F
0	err	err	s4	err	s5	err	1	2	3
1	s6	err	err	err	err	acc	err	err	err
2	r2	s7	err	r2	err	r2	err	err	err
3	r4	r4	err	r4	err	r4	err	err	err
4	err	err	s4	err	s5	err	8	2	3
5	r6	r6	err	r6	err	r6	err	err	err
6	err	err	s4	err	s5	err	err	9	3
7	err	err	s4	err	s5	err	err	err	10
8	s6	err	err	s11	err	err	err	err	err
9	r1	s7	err	r1	err	r1	err	err	err
10	r3	r3	err	r3	err	r3	err	err	err
11	r5	r5	err	r5	err	r5	err	err	err

对应归约集合如下：

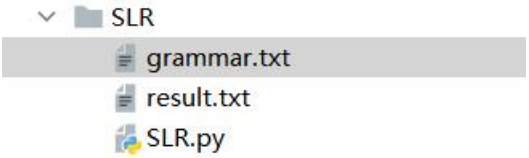
```
[['E1', '->', 'E'], ['E', '->', 'E', '+', 'T'], ['E', '->', 'T'], ['T', '->', 'T', '*', 'F'], ['T', '->', 'F'], ['F', '->', '(', 'E', ')'], ['F', '->', 'id']]
```

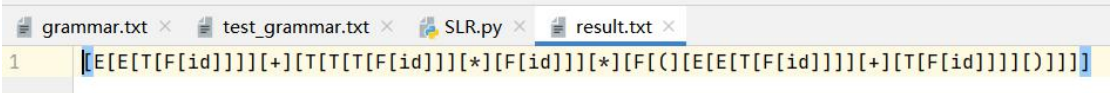
移进-归约过程如下：

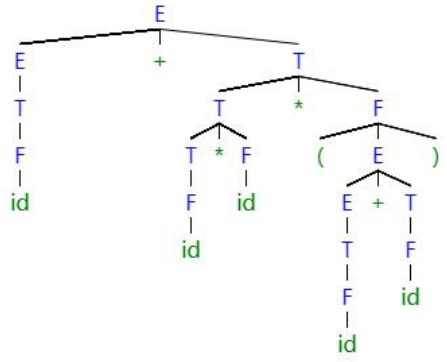
```
[0, 'id', 5] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*', 'id', '+'] []
[0, 'F', '3'] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*', 'id', '+'] [['F', '->', 'id']]
[0, 'T', '2'] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*', 'id', '+'] [['F', '->', 'id'], ['T', '->', 'F']]
[0, 'E', '1'] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*', 'id', '+'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T']]
[0, 'E', '1', '+', 6] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*', 'id'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T']]
[0, 'E', '1', '+', 6, 'id', 5] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T']]
[0, 'E', '1', '+', 6, 'F', '3'] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id']]
[0, 'E', '1', '+', 6, 'T', '9'] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7] ['$', ')', 'id', '+', 'id', '(', '*', 'id'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, 'id', 5] ['$', ')', 'id', '+', 'id', '(', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->',
 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, 'F', '10'] ['$', ')', 'id', '+', 'id', '(', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->',
 'F'], ['F', '->', 'id']]
[0, 'E', '1', '+', 6, 'T', '9'] ['$', ')', 'id', '+', 'id', '(', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F', '->',
 'id'], ['T', '->', 'T', '*', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7] ['$', ')', 'id', '+', 'id', '('] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F', '->',
  'id'], ['T', '->', 'T', '*', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4] ['$', ')', 'id', '+', 'id'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F',
 '->', 'id'], ['T', '->', 'T', '*', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'id', 5] ['$', ')', 'id', '+'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F',
  '->', 'id'], ['T', '->', 'T', '*', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'F', '3'] ['$', ')', 'id', '+'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'],
 ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'T', '2'] ['$', ')', 'id', '+'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'],
 ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'E', '8'] ['$', ')', 'id', '+'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'],

 ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'E', '8', '+', 6] ['$', ')', 'id'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'],
 ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'E', '8', '+', 6, 'id', 5] ['$', ')'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->',
 'F'], ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'E', '8', '+', 6, 'F', '3'] ['$', ')'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->',
 'F'], ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'E', '8', '+', 6, 'T', '9'] ['$', ')'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->',
 'F'], ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, '(', 4, 'E', '8'] ['$', ')'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F', '->',
 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'E', '+', 'T']]
[0, 'E', '1', '+', 6, 'T', '9', '*', 7, 'F', '10'] ['$', ')'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F', '->', 'id'], ['T',
 '->', 'T', '*', 'F'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'E', '+', 'T'], ['F', '->', '(', 'E', ')']]
[0, 'E', '1', '+', 6, 'T', '9'] ['$', ')', 'id', '+', 'id', '(', '*', 'id', '*'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'],
 ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'E', '+', 'T'], ['F', '->', '(', 'E', ')'], ['T', '->', 'T', '*', 'F']]
[0, 'E', '1'] ['$'] [['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['F', '->', 'id'], ['T', '->', 'T', '*', 'F'], ['F', '->',
 'id'], ['T', '->', 'F'], ['E', '->', 'T'], ['F', '->', 'id'], ['T', '->', 'F'], ['E', '->', 'E', '+', 'T'], ['F', '->', '(', 'E', ')'], ['T', '->', 'T', '*', 'F'], ['E', '->',
 'E', '+', 'T']]
```

结果集如下：

```
[E[E[T[F[id]]]][+][T[T[T[F[id]]][*][F[id]]][*][F[(][E[E[T[F[id]]]][+][T[F[id]]]][)]]]]
```

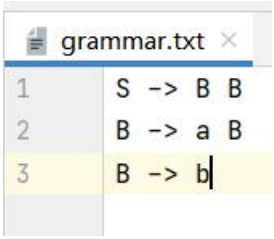






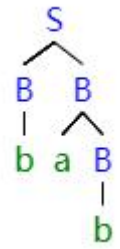
其他运行结果：

文法：　　字符串：b a b

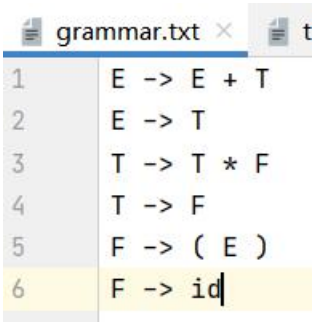


结果集：[S[B[b]][B[a][B[b]]]]

分析树：



文法：　　字符串：id + (id + id) * id



结果集：[E[E[T[F[id]]]][+][T[T[F[(][E[E[T[F[id]]]][+][T[F[id]]]][)]]][*][F[id]]]]

分析树：