

1. 请写出下面的变量a的类型表达式

```
int a[][3];
int *a[3];
int (*a)[3];
int *(*a)[3];
int **a[3];
int *(*a[3])[2];
```

```
int a[][3];           pointer(array(3,int))
int *a[3];            array(3,pointer(int))
int (*a)[3];          pointer(array(3,int))
int *(*a)[3];          pointer(array(3,pointer(int)))
int **a[3];            array(3,pointer(pointer(int)))
int *(*a[3])[2];       array(3,pointer(array(2,pointer(int))))
```

教材 5.6 下列文法定义字面常量表的表。符号的解释和图5.2文法的那些相同，增加了类型list，它表示类型T的元素表。

```
P --> D; E
D --> D; D | id : T
T --> list of T | char | integer
E --> (L) | literal | num | id | nil
L --> E, L | E
```

注意：

1. 上述文法中分号`;`、冒号`:`、逗号`,`、标识符`id`、空表`nil`、数值`num`、字面文本`literal`、左括号`(`、右括号`)`、`list`、`of`、`char`、`integer`都是终结符
2. `L`表示表达式`E`或者由逗号分隔的表达式列表，`(L)`表示list类型的元组

写一个类似5.3节中的翻译方案，以确定表达式(E)和表(L)的类型。

```
P -> D ; E { P.type = if D.type == E.type then D.type
              else type_error }
D -> D1 ; D2 { D.type = if D1.type == D2.type then D1.type
                    else type_error }
D -> id : T    { addtype(id.entry , T.type) }
T -> list of T1 { T.type = list(T1.type)}
T -> char      { T.type = char }
T -> integer    { T.type = integer }
E -> (L)        { E.type = L.type }
E -> literal    { E.type = char }
E -> num        { E.type = integer }
E -> id         { E.type = lookup(id.entry) }
E -> nil        { E.type = nil }
L -> E, L1      { L.type = if E.type == L1.type then E.type
                    else type_error}
L -> E          { L.type = E.type}
```

