（1）以下是C语言数组变量声明初始化的示例：

```
int f[][8][5] = { {{1},{2}},{3,},4,{5,},{6,},{7},8, };
```

填写合适的数字，补全数组变量f声明[]处的空白！

并给出初始化描述中数值1~8在数组f中的下标。

f[2][8][5]

1: f[0][0][0]

2: f[0][1][0]

3: f[1][0][0]

4: f[1][1][0]

5: f[1][2][0]

6: f[1][3][0]

7: f[1][4][0]

8: f[1][5][0]

(2) 针对习题3.1文法，给出(a,((a,a)))的最左、最右推导和分析树。

3.1 考虑文法

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

最左推导：

$S \rightarrow (L) \rightarrow (L,S) \rightarrow (S,S) \rightarrow (a,S)$

$\rightarrow (a,(L)) \rightarrow (a,(S)) \rightarrow (a,((L))) \rightarrow (a,((L,S)))$

$\rightarrow (a,((S,S))) \rightarrow (a,((a,S))) \rightarrow (a,((a,a)))$

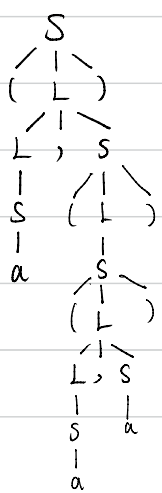
最右推导：

$S \rightarrow (L) \rightarrow (L,S) \rightarrow (L,S) \rightarrow (L,(L))$

$\rightarrow (L,(S)) \rightarrow (L,((L))) \rightarrow (L,((L,S))) \rightarrow (L,((L,a)))$

$\rightarrow (L,((S,a))) \rightarrow (L,((a,a))) \rightarrow (S,((a,a))) \rightarrow (a,((a,a)))$

分析树：



(3) 习题3.2(a)。

3.2 考虑文法

$S \rightarrow aSbS \mid bSaS \mid \varepsilon$

(a) 为句子 $abab$ 构造两个不同的最左推导，以此说明该文法是二义的。

第一种：

$S \rightarrow aSbS \rightarrow abSaSbS \rightarrow abaSbS \rightarrow ababS \rightarrow abab$

第二种

$S \rightarrow aSbS \rightarrow abS \rightarrow abaSbS \rightarrow ababS \rightarrow abab$

（4）有如下C程序：

```
int main()
{
   int i = 0;
   int *p = &i;//语句1
   ++ *p ++ ; //语句2
   return i;
}
```

经x86-64 gcc 13.2编译生成如下intel汇编代码：

```
      push   rbp
      mov    rbp, rsp
      mov    DWORD PTR [rbp-12], 0
      lea    rax, [rbp-12]
      mov    QWORD PTR [rbp-8], rax
      mov    rax, QWORD PTR [rbp-8]
      lea    rdx, [rax+4]
      mov    QWORD PTR [rbp-8], rdx
      mov    edx, DWORD PTR [rax]
      add    edx, 1
      mov    DWORD PTR [rax], edx
      mov    eax, DWORD PTR [rbp-12]
      pop    rbp
      ret
```

• 给出语句1所对应的汇编代码；

• 给语句2中表达式添加嵌套小圆括号()来表示计算次序，越内层越优先计算;
再给出每层()所对应的主要汇编代码。例如，(p)将是最内层括号,对应汇编代码
mov rax, QWORD PTR [rbp-8]。

• 语句1对应的汇编代码：

```
lea     rax, [rbp-12]
mov     QWORD PTR [rbp-8], rdx
```

- (++(*((p)++)))

(p):

```
mov rax, QWORD PTR [rbp-8]
```

((p)++):

```
lea rdx, [rax+4]
mov QWORD PTR [rbp-8], rdx
```

(*((p)++))

```
mov edx, DWORD PTR [rax]
```

(++(*((p)++)))

```
add edx, 1
mov DWORD PTR [rax], edx
```