# 第二部分 上机实践要求

“编译原理与技术”的上机实验要求你对PL/0语言及其编译器进行扩充和修改。每个扩充或修改方式可得到不同的分数，满分为100分。

完成上机作业后，必须提交设计文档，包括：

1. 修改后的PL/0语言文本。包含词法分析（正规式），语法分析（BNF）。
2. 有关修改后的PL/0编译/解释器的说明。详细说明你的编译器是如何编译新的PL/0语言程序的。指出你的程序中最精彩的部分，以及你为什么这样做，你是如何控制和恢复语义错误的。
3. 给出你所改动后的编译器源程序清单，并标记出你所修改的部分。比较你的编译器和原来的编译器之间的差别。
4. 说明你的编译器中可能存在的错误。如果重做一遍，你会有哪些新的改进？

对现存的PL/0编译程序可做如下修改或扩充，其中（1）、（2）、（5）必须完成，剩余的均可任意选择，但总分必须超过60分。

1. 注释（10分）

块注释由/\*和\*/包含，不允许嵌套。

行注释由//开始直到行结束符（回车）。

1. 扩展PL/0中“条件”：（25分）

这种修改包括：

* 1. 增加逻辑运算符**and**、**or**和**not**。
  2. 把PL/0语言中的“条件”概念一般化为C语言那样(表达式值非零即为“真”)。
  3. “条件”的短路计算。

1. 数组（20分）

在PL/0中允许有数组变量声明/对数组元素赋值/在表达式中引用数组元素等。可以有多维数组，数组的维度范围设为常量：

dimDeclaration → **[**const**]** dimDeclaration

dimDeclaration →ε

const → **ident** | **number**

为简单起见，PL/0中基本数据类型就是整型。

这样，在变量声明部分可以出现:

**var i,j,k;** //三个变量均为整型变量

也可以出现：

**var i,j,k[10][10];**

**//i,j为普通整型变量，而k为二维整型数组**

1. 过程参数实现。采用传值方式，并实现简单的语义检查（如实参和形参个数/类型对应等）。（10分）
2. else子句实现。（10分）
3. for语句的实现。语法可参照C语言（10分）
4. exit语句（退出当前执行过程）和break语句（跳出包含它的最内层循环）。（5分）
5. PL/0输入/输出语句。可参考C的相关函数，实现任意个数参数的输入或输出功能。（5分）
6. 分离解释和编译器与加强的语法错误恢复机制（5分）

注意：***上面的这些要求有时会互相影响*：例如数组作为参数传递的情况。**

**为了实现以上功能，可任意增加PL/0处理机的指令。但要注意指令的简单与合理。**