2022 年编译原理上机实验

一．以课程链接上的供参考的 PL/0 实验（2017 版）为基础，

扩展 PL0 功能：

1） 数组。参看 2017 版的上机实践要求第（3）项

2） 内置输出函数 print。参看 2017 版的上机实践要求第（6）项（print 输出功能）

如 print（i,j），打印变量 i 和 j 的值。print()，输出换行。

3） for 语句，语法形式：

for_statement -> **for (var ID : (**low**,** up**,** step**))** statement

| **for (var ID : (**low**,** up**))** statement

其中，循环控制变量 ID 的初值为 low，终值为 up，且每次循环后 ID += step。如 step 没出现，则缺省为 1。

例如：

for(var i : (1,10)) …

for(var j : (1,10,2)) …

for(var k : (10,1,-1)) …

4） else 子句。

5） 将原 PL0 的赋值语句，扩展为 赋值表达式。（可参考 C 语言语法）

6） 内置函数 setjmp 和 longjmp 实现，用于模拟 C 语言标准库函数 setjmp 和 longjmp。这两个 C 标准库函数，可参见（网上资料很多）：

C 语言：setjmp 和 longjmp 函数使用详解_houxiaoni01 的博客-CSDN 博客_longjmp

为简化实现，约定 PL0 实现中：

1 ***跳转缓冲区*的定义如下：**

```
#define JMPMAX 4096
int jmp_buf[JMPMAX];//setjmp/longjmp buffer
```

2 两个函数使用方式如下：

setjmp(buf_idx) ：参数 buf_idx 是一个整型表达式，表示“***跳转缓冲区***”的位置下标。

longjmp(buf_idx, val)：参数 buf_idx 意思同 setjmp。参数 val 则是 setjmp 的新返回值。

一个 PL0 例子（语法格式供参考）：

```
var result;

procedure banana;
procedure apple;
```

```
begin
        print(1000);
        longjmp(1,2);
        print(2000);
end;
begin
    for(var i:(0,10)) call apple;
end;

begin
       if((result := setjmp(1)) = 1) then
              print(1111);
        else if(result = 2) then
               print(2222);
        else
        begin
              print(999);
              call banana;
        end;
end.
```

输出结果：

999 1000 2222

二．可以自行组队（至多 5 人）合作，明确分工，提交设计文档。