1. PL/0 Discription

<prog> → program <id>；<block>

<block> → [<condecl>][<vardecl>][<proc>]<body>

<condecl> → const <const>{,<const>}

<const> → <id>:=<integer>

<vardecl> → var <id>{,<id>}

<proc> → procedure <id>（[<id>{,<id>}]）;<block>{;<proc>}

<body> → begin <statement>{;<statement>}end

<statement> → <id> := <exp>

|if <lexp> then <statement>[else <statement>]

|while <lexp> do <statement>

|call <id>（[<exp>{,<exp>}]）

|<body>

|read (<id>{，<id>})

|write (<exp>{,<exp>})

<lexp> → <exp> <lop> <exp>|odd <exp>

<exp> → [+|-]<term>{<aop><term>}

<term> → <factor>{<mop><factor>}

<factor>→<id>|<integer>|(<exp>)

<lop> → =|<>|<|<=|>|>=

<aop> → +|-

<mop> → \*|/

<id> → l{l|d} （l is an alphabet）

<integer> → d{d}