|  |  |  |  |
| --- | --- | --- | --- |
| 测试输入 | 期望输出 | 实际输出 | 备注 |
| 正常用例 | | | |
| const a=10;  var b,c;  begin  read(b);  c:=a+b;  write(c);  end. | (constsym,const)  (ident,a)  (eql,=)  (number,10)  (semicolon,;)  (varsym,var)  (ident,b)  (comma,,)  (ident,c)  (semicolon,;)  (beginsym,begin)  (readsym,read)  (lparen,()  (ident,b)  (rparen,))  (semicolon,;)  (ident,c)  (becomes,:=)  (ident,a)  (plus,+)  (ident,b)  (semicolon,;)  (writesym,write)  (lparen,()  (ident,c)  (rparen,))  (semicolon,;)  (endsym,end)  (period,.) | (constsym,const)  (ident,a)  (eql,=)  (number,10)  (semicolon,;)  (varsym,var)  (ident,b)  (comma,,)  (ident,c)  (semicolon,;)  (beginsym,begin)  (readsym,read)  (lparen,()  (ident,b)  (rparen,))  (semicolon,;)  (ident,c)  (becomes,:=)  (ident,a)  (plus,+)  (ident,b)  (semicolon,;)  (writesym,write)  (lparen,()  (ident,c)  (rparen,))  (semicolon,;)  (endsym,end)  (period,.) | **覆盖单词：**  基本字：begin end const read write  运算符：+ :=  界符：( ) , ; . |
| **标志符出错** | | | |
| const xy34,10a=10;  var b=0,c23;  begin  READ(b);  c:=a+b;  write(C);  end. | (constsym,const)  (ident,xy34)  (comma,,)  **(error,10a)**  (eql,=)  (number,10)  (semicolon,;)  (varsym,var)  (ident,b)  (eql,=)  (number,0)  (comma,,)  (ident,c23)  (semicolon,;)  (beginsym,begin)  (readsym,READ)  (lparen,()  (ident,b)  (rparen,))  (semicolon,;)  (ident,c)  (becomes,:=)  (ident,a)  (plus,+)  (ident,b)  (semicolon,;)  (writesym,write)  (lparen,()  (ident,C)  (rparen,))  (semicolon,;)  (endsym,end)  (period,.) | (constsym,const)  (ident,xy34)  (comma,,)  **(error,10a)**  (eql,=)  (number,10)  (semicolon,;)  (varsym,var)  (ident,b)  (eql,=)  (number,0)  (comma,,)  (ident,c23)  (semicolon,;)  (beginsym,begin)  (readsym,READ)  (lparen,()  (ident,b)  (rparen,))  (semicolon,;)  (ident,c)  (becomes,:=)  (ident,a)  (plus,+)  (ident,b)  (semicolon,;)  (writesym,write)  (lparen,()  (ident,C)  (rparen,))  (semicolon,;)  (endsym,end)  (period,.) | **覆盖单词：**  同上  **说明：**  10a为非法标志符；  关键字不区分大小写 |
| **number出错** | | | |
| var n, f;  begin  n := 0;  f := 1.0;  while n # 1e2 do  begin  n := n + 1;  f := f \* n;  end;  call print;  end. | (varsym,var)  (ident,n)  (comma,,)  (ident,f)  (semicolon,;)  (beginsym,begin)  (ident,n)  (becomes,:=)  (number,0)  (semicolon,;)  (ident,f)  (becomes,:=)  **(error,1.0)**  (semicolon,;)  (whilesym,while)  (ident,n)  (neq,#)  **(error,1e2)**  (dosym,do)  (beginsym,begin)  (ident,n)  (becomes,:=)  (ident,n)  (plus,+)  (number,1)  (semicolon,;)  (ident,f)  (becomes,:=)  (ident,f)  (times,\*)  (ident,n)  (semicolon,;)  (endsym,end)  (semicolon,;)  (callsym,call)  (ident,print)  (semicolon,;)  (endsym,end)  (period,.) | (varsym,var)  (ident,n)  (comma,,)  (ident,f)  (semicolon,;)  (beginsym,begin)  (ident,n)  (becomes,:=)  (number,0)  (semicolon,;)  (ident,f)  (becomes,:=)  **(error,1.0)**  (semicolon,;)  (whilesym,while)  (ident,n)  (neq,#)  **(error,1e2)**  (dosym,do)  (beginsym,begin)  (ident,n)  (becomes,:=)  (ident,n)  (plus,+)  (number,1)  (semicolon,;)  (ident,f)  (becomes,:=)  (ident,f)  (times,\*)  (ident,n)  (semicolon,;)  (endsym,end)  (semicolon,;)  (callsym,call)  (ident,print)  (semicolon,;)  (endsym,end)  (period,.) | **覆盖单词：**  基本字 begin end var do while call  运算符：# + \* :=  界符： , ; .  **错误说明：**  “1.0” “1e2”出错PL/0语言中无此类数字类型 |
| **关键字出错** | | | |
| var x,n,y,z;  begin  n := 45;  x:= 78;  y :=34;  PROCEDURE m;  var sum,first,count;  BEGINN  sum:=first+count\*10  end;  call m;  if x >= y  then x:=x-y;  if x = y  then x:=x\*10;  if x < y  than y:=y/5;  if ODD n THEN  z := x + y;  end. | (varsym,var)  (ident,x)  (comma,,)  (ident,n)  (comma,,)  (ident,y)  (comma,,)  (ident,z)  (semicolon,;)  (beginsym,begin)  (ident,n)  (becomes,:=)  (number,45)  (semicolon,;)  (ident,x)  (becomes,:=)  (number,78)  (semicolon,;)  (ident,y)  (becomes,:=)  (number,34)  (semicolon,;)  (proceduresym,PROCEDURE)  (ident,m)  (semicolon,;)  (varsym,var)  (ident,sum)  (comma,,)  (ident,first)  (comma,,)  (ident,count)  (semicolon,;)  **(errorBEGINN)**  (ident,sum)  (becomes,:=)  (ident,first)  (plus,+)  (ident,count)  (times,\*)  (number,10)  (endsym,end)  (semicolon,;)  (callsym,call)  (ident,m)  (semicolon,;)  (ifsym,if)  (ident,x)  (geq,>=)  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (ident,x)  (minus,-)  (ident,y)  (semicolon,;)  (ifsym,if)  (ident,x)  (eql,=)  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (ident,x)  (times,\*)  (number,10)  (semicolon,;)  (ifsym,if)  (ident,x)  (lss,<)  (ident,y)  **(error,than)**  (ident,y)  (becomes,:=)  (ident,y)  (slash,/)  (number,5)  (semicolon,;)  (ifsym,if)  (oddsym,ODD)  (ident,n)  (thensym,THEN)  (ident,z)  (becomes,:=)  (ident,x)  (plus,+)  (ident,y)  (semicolon,;)  (endsym,end)  (period,.) | (varsym,var)  (ident,x)  (comma,,)  (ident,n)  (comma,,)  (ident,y)  (comma,,)  (ident,z)  (semicolon,;)  (beginsym,begin)  (ident,n)  (becomes,:=)  (number,45)  (semicolon,;)  (ident,x)  (becomes,:=)  (number,78)  (semicolon,;)  (ident,y)  (becomes,:=)  (number,34)  (semicolon,;)  (proceduresym,PROCEDURE)  (ident,m)  (semicolon,;)  (varsym,var)  (ident,sum)  (comma,,)  (ident,first)  (comma,,)  (ident,count)  (semicolon,;)  **(ident,BEGINN)**  (ident,sum)  (becomes,:=)  (ident,first)  (plus,+)  (ident,count)  (times,\*)  (number,10)  (endsym,end)  (semicolon,;)  (callsym,call)  (ident,m)  (semicolon,;)  (ifsym,if)  (ident,x)  (geq,>=)  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (ident,x)  (minus,-)  (ident,y)  (semicolon,;)  (ifsym,if)  (ident,x)  (eql,=)  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (ident,x)  (times,\*)  (number,10)  (semicolon,;)  (ifsym,if)  (ident,x)  (lss,<)  (ident,y)  **(ident,than)**  (ident,y)  (becomes,:=)  (ident,y)  (slash,/)  (number,5)  (semicolon,;)  (ifsym,if)  (oddsym,ODD)  (ident,n)  (thensym,THEN)  (ident,z)  (becomes,:=)  (ident,x)  (plus,+)  (ident,y)  (semicolon,;)  (endsym,end)  (period,.) | **覆盖单词：**  基本字 begin end var if then procedure odd call  运算符：- + \* / := < >= =  界符： , ; .  **错误说明：**  “BEGINN”“than”为关键字输入错误，但作为标志符是合法的，因此词法分析会将其识别为标志符 |
| **单目和双目运算符识别** | | | |
| var as, fa,x,n,y,z;  begin  as := 0;  n := 45;  fs := 30;  x:= 78;  y :=34;  if as > fa  then as:= as\*12.5  if as <= fa  then as:=as/fa;  if x >= y  then x:=x-y;  if x = y  then x:=10;  if x <= y  then y:5;  end. | (varsym,var)  (ident,as)  (comma,,)  (ident,fa)  (comma,,)  (ident,x)  (comma,,)  (ident,n)  (comma,,)  (ident,y)  (comma,,)  (ident,z)  (semicolon,;)  (beginsym,begin)  (ident,as)  (becomes,:=)  (number,0)  (semicolon,;)  (ident,n)  (becomes,:=)  (number,45)  (semicolon,;)  (ident,fs)  (becomes,:=)  (number,30)  (semicolon,;)  (ident,x)  (becomes,:=)  (number,78)  (semicolon,;)  (ident,y)  (becomes,:=)  (number,34)  (semicolon,;)  (ifsym,if)  (ident,as)  (gtr,>)  (ident,fa)  (thensym,then)  (ident,as)  (becomes,:=)  (ident,as)  (times,\*)  **(error,12.5)**  (ifsym,if)  (ident,as)  **(leq,<=)**  (ident,fa)  (thensym,then)  (ident,as)  (becomes,:=)  (ident,as)  (slash,/)  (ident,fa)  (semicolon,;)  (ifsym,if)  (ident,x)  **(geq,>=)**  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (ident,x)  (minus,-)  (ident,y)  (semicolon,;)  (ifsym,if)  (ident,x)  (eql,=)  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (number,10)  (semicolon,;)  (ifsym,if)  (ident,x)  **(leq,<=)**  (ident,y)  (thensym,then)  (ident,y)  **error!(semicolon,;)**  (endsym,end)  (period,.) | (varsym,var)  (ident,as)  (comma,,)  (ident,fa)  (comma,,)  (ident,x)  (comma,,)  (ident,n)  (comma,,)  (ident,y)  (comma,,)  (ident,z)  (semicolon,;)  (beginsym,begin)  (ident,as)  (becomes,:=)  (number,0)  (semicolon,;)  (ident,n)  (becomes,:=)  (number,45)  (semicolon,;)  (ident,fs)  (becomes,:=)  (number,30)  (semicolon,;)  (ident,x)  (becomes,:=)  (number,78)  (semicolon,;)  (ident,y)  (becomes,:=)  (number,34)  (semicolon,;)  (ifsym,if)  (ident,as)  (gtr,>)  (ident,fa)  (thensym,then)  (ident,as)  (becomes,:=)  (ident,as)  (times,\*)  **(error,12.5)**  (ifsym,if)  (ident,as)  **(leq,<=)**  (ident,fa)  (thensym,then)  (ident,as)  (becomes,:=)  (ident,as)  (slash,/)  (ident,fa)  (semicolon,;)  (ifsym,if)  (ident,x)  **(geq,>=)**  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (ident,x)  (minus,-)  (ident,y)  (semicolon,;)  (ifsym,if)  (ident,x)  (eql,=)  (ident,y)  (thensym,then)  (ident,x)  (becomes,:=)  (number,10)  (semicolon,;)  (ifsym,if)  (ident,x)  **(leq,<=)**  (ident,y)  (thensym,then)  (ident,y)  **error!(semicolon,;)**  (endsym,end)  (period,.) | **覆盖单词：**  基本字 begin end var if then  运算符：- \* / := > < >= <= =  界符： , ; .  **说明：**  “:”非合法运算符输出错误；  “>=”“<=”为双目运算符，被整体识别 |
| **未知字符报错** | | | |
| const asd@112=0;  var b,c&!;  begin  b:=23;  while a < b do$  begin  c := a \* b;  a := a+1;  end;  end. | (constsym,const)  (ident,asd)  **(error,@112)**  (eql,=)  (number,0)  (semicolon,;)  (varsym,var)  (ident,b)  (comma,,)  (ident,c)  **(error,&!)**  (semicolon,;)  (beginsym,begin)  (ident,b)  (becomes,:=)  (number,23)  (semicolon,;)  (whilesym,while)  (ident,a)  (lss,<)  (ident,b)  (dosym,do)  **(error,$)**  (beginsym,begin)  (ident,c)  (becomes,:=)  (ident,a)  (times,\*)  (ident,b)  (semicolon,;)  (ident,a)  (becomes,:=)  (ident,a)  (plus,+)  (number,1)  (semicolon,;)  (endsym,end)  (semicolon,;)  (endsym,end)  (period,.) | (constsym,const)  (ident,asd)  **(error,@112)**  (eql,=)  (number,0)  (semicolon,;)  (varsym,var)  (ident,b)  (comma,,)  (ident,c)  **(error,&!)**  (semicolon,;)  (beginsym,begin)  (ident,b)  (becomes,:=)  (number,23)  (semicolon,;)  (whilesym,while)  (ident,a)  (lss,<)  (ident,b)  (dosym,do)  **(error,$)**  (beginsym,begin)  (ident,c)  (becomes,:=)  (ident,a)  (times,\*)  (ident,b)  (semicolon,;)  (ident,a)  (becomes,:=)  (ident,a)  (plus,+)  (number,1)  (semicolon,;)  (endsym,end)  (semicolon,;)  (endsym,end)  (period,.) | **覆盖单词：**  基本字 begin end var const while  运算符：+ \*:=  界符：, ; .  **错误说明：**  @ ￥ &！等特殊字符报错 |