测试报告：

我们使用了如下的完整测试代码:

//测试基本的词法元素读入

25;

int askdjiw=10;

"allahu" "akbar";

int p=0;

//测试运算符，基本的表达式求值

p++;p--;

if(p==0)

printf("ln 7");

p=+p;

if(p==0)

printf("ln 10");

p=-p;

if(p==0)

printf("ln 13");

p=10;

p=p\*p;

if(p==100){

printf("ln 17");

}

p=p/p;

//测试if else结构

if(p==0)

printf("ln 21 should not appear");

else

printf("ln 23 should.");

//测试printf嵌套语句

if(p==1)

printf("ln 21%d",p=127+p);

if(p==128)

printf("ln 23%d",p=128-p);

if(p==0){

printf("ln 25");

}

//测试循环结构及行号打印

while (p<25)

p++;

while(p<=25) p++;

do {

p--;

}while(p>0);

for(;;) {printf("program should not die here.");

break;

}

//测试注释结构

//testing.

/\* still testing. \*/

;;;;;

;

;

;

//进一步测试for语句

for(k = 0;k<=51;k++){

if(k==50){

printf("appear only once.");

//测试break是否有效

break;

}

}

//测试作用域

{

//new scope

int k=25;

}

//测试作用域是否污染

if(k==25)

printf("program boomed.");

if(k==50)

printf("yeah.");

while(k!=100) k++;

if(k==100);

printf("yeah.");

//奇怪的表达式测试。

//Compound expressions.

int c=p+++ +k;

int d=p+++-k;

if(c==100){

if(d==-99)

if(p==2)

printf("sure.");

}

int a=1,b=1,c=1;

d=a+++b+++c++;

if(d==3)

printf("yeah.");

//Extreme tests.

//相当极端的复合测试

do {

;

;

;

for(int a = 1; a < 2; a++) {

for(int b = 1; b < 3; b++);

}

for(int a = 1; a < 2; a++) {

for(int b = 1; b < 3; b++)for(int c = 1; c < 8; c++);

break;

}

if(a == 2)

break;

}while(c++ < 65);;

/\*

c = 66;

\*/

do for(;;)break;

while(c++ < 66);

/\*

c = 67;

\*/

测试结果符合预期。

其中测试内容按照顺序涉及

词法元素：整型常量，标识符，字符串面值常量，操作符：自增，自减，正，负，加，减，乘，除，大于，小于，大于等于，小于等于，等于，不等于，赋值操作。

语句：

if语句，[else ]

while 语句

do while语句

for循环语句

printf语句

声明语句

表达式（包含空表达式）