

验收样例使用说明

测试代码

```
1  program right;
2  const a:=5;
3  var x,y,b,c,k,fibreturn,return2;
4      procedure fib(x);
5          var sum;
6          begin
7              if x<=2 then
8                  fibreturn:=1
9              else
10                 begin
11                     sum:=0;
12                     call fib(x-1);
13                     sum:=sum+fibreturn;
14                     call fib(x-2);
15                     sum:=sum+fibreturn;
16                     fibreturn:=sum
17                 end
18             end;
19
20         procedure test1();
21             procedure in();
22                 begin
23                     y:=y+a
24                 end
25             begin
26                 y:=y+a;
27                 call in()
28             end;
29
30         procedure judge(x);
31             begin
32                 if odd x+1 then
33                     return2:=1
34                 else
35                     return2:=0
36             end;
37
```

```

38     procedure test2(m,k);
39     var i,sum;
40     begin
41         i:=0;
42         sum:=0;
43         while i<k do
44             begin
45                 sum:=sum+m;
46                 i:=i+1
47             end;
48         write(sum);
49         call judge(sum)
50     end
51
52 begin
53     read(x);
54     call fib(x);
55     write(fibreturn);
56
57     read(y);
58     call test1();
59     write(y);
60
61     read(b,c,k);
62     call test2(b+c,k);
63     write(return2)
64 end

```

说明

该测试样例共由3个测试函数组成 各函数含义及测试目的如下：

fib

测试目的

测试递归调用的实现

功能

输入数 x 计算第 x 个斐波那契数

测试样例

- 输入
10
- 输出
55

test1

测试目的

测试程序如下功能的实现：

- 非局部变量的访问
- 嵌套调用的实现
- 对过程本身子过程的调用

功能

输入数 y 计算表达式 $y+2a$ 的值 并将其赋值给 y

测试样例

- 输入
27
- 输出
37

test2

测试目的

- （多）参数传递
- 参数传递是否多余
- 代码回填正确性（控制流语句的翻译）

- 嵌套调用的实现
- 过程对并列子过程的调用

功能

输入数 b, c, k 计算表达式 $(b+c)*k$ 的值 并判断该值是否为偶数 若为偶数则返回值 `return2` 为 1 否则为 0

测试样例

- 输入
3 9 5
- 输出
60
1



测试结果

The screenshot shows a code editor with a Pascal program. The program defines a function `judge` to check if a number is even, and a procedure `test2` to calculate $(b+c)*k$ and return 1 if the result is even, otherwise 0. The main procedure `test1` reads input x, y, b, c, k and calls `test2` with b, c, k . The output window shows the execution results for the test case $b=3, c=9, k=5$.

```
1 while i<k do
2   begin
3     sum:=sum+m;
4     i:=i+1
5   end;
6 write(sum);
7 call judge(sum)
8 end
9
10 begin
11   read(x);
12   call fib(x);
13   write(fibreturn);
14
15   read(y);
16   call test1();
17   write(y);
18
19   read(b,c,k);
20   call test2(b+c,k);
21   write(return2)
```

运行 Compiler x

/Users/plotnickslope/Desktop/学习资料/编译原理/课设/代码/cmake-build-debug/Compiler
current file:right
Lexical Analysis done!
Please input a number:

 10
55
 Please input a number:
27
37
Please input a number:
3
Please input a number:
9
Please input a number:
5
60
1
Grammatical Analysis done!

进程已结束，退出代码为 0

