语义分析测试报告

# 测试列表

* 重定义
* 未定义
* 标识符类型错误
* 个数不匹配
* 标识符种类错误
* 表达式类型错误
* 赋值语句左右表达式类型不匹配
* 数组下标越界错误
* 运算符两边的操作数类型不一致错误
* 某个操作数类型错误
* read的实参类型错误
* 除数为0的错误
* 其他类型的错误
* integer到real隐式类型转换
* 变量非法引用
* 嵌套层数过多

# 测试详情

## 重定义

### 2.1.1. 标识符与库函数同名

* 预期结果

报与库函数同名错误

* 实际测试结果

[Duplicate defined error!] <Line 3> "write" has already been defined as a lib program.

[Duplicate defined error!] <Line 4> "read" has already been defined as a lib program.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.1.2. 标识符与主程序同名

* 预期结果

报和主程序同名的错误

* 实际测试结果

[Duplicate defined error!] <Line 3> "test" has the same name as the main program.

[Duplicate defined error!] <Line 5> "test" has the same name as the main program.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.1.3. 函数和过程参数使用已定义的名称

* 预期结果

报标识符已经被定义的错误。

* 实际测试结果

[Duplicate defined error!] <Line 4> "swap" has already been defined as a normal variant at line 3.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 未定义

### 2.2.1. for语句循环变量未定义

* 预期结果

报变量未定义的错误

* 实际测试结果

[Undefined identifier!] <Line 4> "c" has not been defined.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.2.2. 函数调用时，函数未定义

* 预期结果

报函数未定义的错误

* 实际测试结果

[Undefined identifier!] <Line 7> "v" has not been defined.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.2.3. 变量引用时，变量未定义

* 预期结果

报变量未定义的错误

* 实际测试结果

[Undefined identifier!] <Line 6> "y" has not been defined.

[Assign statement type mismatch!] <Left at line 6, right at line 6> Left "x" type is integer while right "y" type is error.

* 测试结果分析

不仅报了变量未定义错误，而且由于未定义的变量作为赋值语句的左值，因此还报了赋值符号左右两侧类型不一致的错误。程序正常运行。

## 标识符类型错误

### 2.3.1. for语句中循环变量不是整型

* 预期结果

报循环变量不是整型的错误

* 实际测试结果

[Usage type error!] <Line 5> "c" used for cyclic variable of for statement should be integer but not char.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.3.2. 函数调用时，参数类型错误

* 预期结果

报参数类型的错误并指出是第几个参数

* 实际测试结果

[Usage type error!] <Line 9> "arg no.1" used for gtcomdiv should be integer but not char.

[Usage type error!] <Line 9> "arg no.2" used for gtcomdiv should be integer but not char.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.3.3. variant变量类型错误

* 预期结果

报变量类型非标准类型错误

* 实际测试结果

[Referenced actual parameter error!] <Line 9> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter or array element.

[Referenced actual parameter error!] <Line 9> The 2th actual parameter expression should be a normal variable、value parameter、referenced parameter or array element.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.3.4. case语句判断的变量类型错误

* 预期结果

报case语句判断的变量表达式应该为基本类型的错

* 实际测试结果

[Expression type error!] <Line 15> Expression "m" used for case statement should be integer,char,real or boolean but not m\_.

[Expression type error!] <Line 20> Expression "n" used for case statement should be integer,char,real or boolean but not n\_.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 个数不匹配

### 2.4.1. 数组引用时维数与定义不一致

* 预期结果

报数组维度与数组定义不一致错误

* 实际测试结果

[Array index number mismatch!] <Line 10> Array "arr" should have 2 but not 1 indices.

[Assign statement type mismatch!] <Left at line 10, right at line 10> Left "s[1]" type is error while right "1" type is integer.

* 测试结果分析

除了报数组维度错误之外，由于数组引用作为了赋值语句的左值，因此还报了赋值符号两侧类型不一致错误。程序正常运行。

### 2.4.2. procedure调用实参与形参个数不等

* 预期结果

报过程调用参数个数不匹配的错误

* 实际测试结果

[Procedure parameter number mismatch!] <Line 13> Procedure "swap" should have 2 but not 3 parameters.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.4.3. 函数调用实参与形参个数不等

* 预期结果

报过程调用参数个数不匹配的错误

* 实际测试结果

[Function parameter number mismatch!] <Line 9> Function "gtcomdiv" should have 2 but not 1 parameters.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 标识符种类错误

### 2.5.1. for语句中的循环变量不是传值参数、引用参数或普通变量

* 预期结果

报循环变量种类错误的错误

* 实际测试结果

[Symbol kinds mismatch!] <Line 4> "input" defined at line 1 is a parameter of program but not a value parameter, var parameter or normal variant.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.5.2. 将函数当成过程调用

* 预期结果

报将函数错当成过程调用的错误

* 实际测试结果

[Symbol kinds mismatch!] <Line 12> "fun" defined at line 4 is a function but not a procedure.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 表达式类型错误

### 2.6.1. 数组下标表达式类型不是integer

* 预期结果

报数组下标表达式类型错误，正确类型应当为整型

* 实际测试结果

[Expression type error!] <Line 8> Expression "x" used for 2th index of array "arr" should be integer but not real.

[Assign statement type mismatch!] <Left at line 8, right at line 8> Left "s[1][x]" type is error while right "3" type is integer.

* 测试结果分析

不仅报了数组下标表达式类型错误，而且由于该错误的数组元素引用作为了赋值语句的左值，因此还报了赋值符号左右两侧类型不一致错误，程序正常运行。

### 2.6.2. repeat语句条件表达式类型不是boolean

* 预期结果

报repeat语句中条件表达式类型错误

* 实际测试结果

[Expression type error!] <Line 13> Expression "a+b" used for condition of repeat-until statement should be boolean but not integer.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.6.3. while语句条件表达式类型不是boolean

* 预期结果

报while语句中条件表达式类型的错误

* 实际测试结果

[Expression type error!] <Line 6> Expression "b" used for condition of while statement should be boolean but not char.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.6.4. if语句条件表达式类型不是boolean

* 预期结果

报if语句条件表达式类型的错误

* 实际测试结果

[Operands expression type mismatch!] <Left at line 6, right at line 6> Left "b" type is integer while right "c" type is char.

[Expression type error!] <Line 6> Expression "b=c" used for condition of if statement should be boolean but not error.

* 测试结果分析

不仅报了预期的错误，还输出了赋值符号两边类型不相同的错误；

测试结果正常，语义分析程序继续运行

### 2.6.5. for语句循环初始值或终值类型不是integer

* 预期结果

分别报两个for语句中循环的初始值和终值不是整型的错误

* 实际测试结果

[Expression type error!] <Line 5> Expression "c" used for start value of for statement should be integer but not char.

[Expression type error!] <Line 7> Expression "a>b" used for end value of for statement should be integer but not boolean.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.6.6. procedure或function调用实参与形参类型不匹配

* 预期结果

报procedure调用实参与形参类型不匹配的错误

* 实际测试结果

[Expression type error!] <Line 12> Expression "x" used for 1th actual parameter of procedure call of "swap" should be integer but not char.

[Expression type error!] <Line 12> Expression "y" used for 2th actual parameter of procedure call of "swap" should be integer but not char.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 赋值语句左右表达式类型不匹配

* 预期结果

报赋值语句左右表达式类型不匹配的错误

* 实际测试结果

[Assign statement type mismatch!] <Left at line 12, right at line 12> Left "c" type is integer while right "fun(a,b)" type is boolean.

[Assign statement type mismatch!] <Left at line 13, right at line 13> Left "d" type is char while right "c" type is integer.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 数组下标越界错误

* 预期结果

报数组下标越界错误

* 实际测试结果

[Array range out of bound!] <Line 7> The value of expression "8" is 8, but the range of array "arr" 1th index is 1 to 7.

[Assign statement type mismatch!] <Left at line 7, right at line 7> Left "s[1][8]" type is error while right "3" type is integer.

* 测试结果分析

不仅报了数组下标越界错误，而且由于该错误的数组元素引用作为了赋值语句的左值，因此还报了赋值符号两侧类型不一致错误，程序正常运行。

## 运算符两侧的操作数类型不一致错误

### 2.9.1. 关系运算符两侧的操作数类型不一致错误

* 预期结果

报关系运算符两侧表达式类型不一致错误；

表达式整体类型错误；

* 实际测试结果

[Operands expression type mismatch!] <Left at line 10, right at line 10> Left "x" type is boolean while right "y" type is integer.

[Expression type error!] <Line 10> Expression "x=y" used for condition of if statement should be boolean but not error.

* 测试结果分析

结果符合预期

## 某个操作数类型错误

### 2.10.1. 单目运算符not后表达式类型错误

* 预期结果

报not云算符后表达式类型错误以及整体表达式类型错误

* 实际测试结果

[Operand expression type error!] <Line 8> Expression "y" type should be boolean but not integer.

[Expression type error!] <Line 8> Expression "not y" used for condition of if statement should be boolean but not error.

* 测试结果分析

测试结果符合预期

### 2.10.2. 负号或正号后表达式类型错误

* 预期结果

报负号或正号后表达式类型错误，正确类型应当为integer或real

* 实际测试结果

[Operand expression type error!] <Line 7> Expression "'a'" type should be integer or real but not char.

[Assign statement type mismatch!] <Left at line 7, right at line 0> Left "y" type is integer while right "-'a'" type is error.

* 测试结果分析

不仅报了预期的错误，而且由于此时表达式在赋值语句中作为右值，因此还报了赋值符左右表达式类型不匹配错误，程序正常运行

### 2.10.3. 算数运算符操作数类型错误

* 预期结果

报算术运算符某侧操作数类型错误，正确类型应当为integer或real

* 实际测试结果

[Operand expression type error!] <Line 9> Expression "y" type should be integer or real but not char.

[Assign statement type mismatch!] <Left at line 9, right at line 9> Left "x" type is integer while right "z+y" type is error.

* 测试结果分析

不仅报了预期的算术运算符操作数类型错误，而且由于算术表达式位于赋值语句的右侧，因此还报了赋值符号两侧类型不一致错误。

### 2.10.4. 逻辑运算符操作数类型错误

* 预期结果

报逻辑运算符某侧操作数类型错误，正确类型应当为boolean

* 实际测试结果

[Operand expression type error!] <Line 9> Expression "x" type should be boolean but not integer.

[Assign statement type mismatch!] <Left at line 9, right at line 9> Left "y" type is boolean while right "x and z" type is error.

* 测试结果分析

不仅报了预期的逻辑运算符操作数类型错误，而且由于逻辑表达式位于赋值语句的右侧，因此还报了赋值符号两侧类型不一致错误。

## read的实参类型错误

* 预期结果

对程序中出现的将函数名，数组名，常量作为read参数的情况，分别报read读取的参数类型不是变量或者数组元素的错误

* 实际测试结果

[Actual parameter of read procedure type error!] <Line 13> "read" 1th expression parameter "fun(a,b)" is not a variant or an array element.

[Invalid reference!] <Line 14> "c" is a array, it can't be referenced.

[Actual parameter of read procedure type error!] <Line 14> "read" 1th expression parameter "c" is not a variant or an array element.

[Actual parameter of read procedure type error!] <Line 15> "read" 1th expression parameter "m" is not a variant or an array element.

* 测试结果分析

除了预期结果之外，还报了数组名不能被引用的错误；

测试结果正常，语义分析程序继续运行

## 除数为0的错误

### 2.12.1. 运算符“/”除数为0

* 预期结果

报运算符 “/” 后除数为0错误

* 实际测试结果

[Divide zero error!] <Line 7> The value of expression is 0, which is the second operand of operation "/".

* 测试结果分析

测试结果符合预期

### 2.12.2. 运算符“div”除数为0

* 预期结果

报运算符 “div” 后除数为0错误

* 实际测试结果

[Divide zero error!] <Line 8> The value of expression is 0, which is the second operand of operation "div".

* 测试结果分析

测试结果符合预期

### 2.12.3. 运算符“mod”除数为0

* 预期结果

报运算符 “mod” 后除数为0错误

* 实际测试结果

[Divide zero error!] <Line 9> The value of expression is 0, which is the second operand of operation "mod".

* 测试结果分析

测试结果符合预期

## 其他类型的错误

### 2.13.1. 赋值语句中左值为常量

* 预期结果

报常量不能作为赋值语句左值的错误

* 实际测试结果

[Constant as l-value error!] <Line7> Costant "c" can't be referenced as l-value.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.13.2. 为函数返回值赋值时类型错误

* 预期结果

报返回值赋值语句类型错误

* 实际测试结果

[Return type of funciton mismatch!] <Line 6> The type of return expression is boolean ,but not integer as function "fun" defined.

[Return type of funciton mismatch!] <Line 7> The type of return expression is boolean ,but not integer as function "fun" defined.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.13.3. read或write的参数为0个

* 预期结果

报错提示read和write过程至少要有一个参数

* 实际测试结果

[read actual parameter missing!] <Line 11> procedure "read" must have at least one actual parameter.

[write actual parameter missing!] <Line 13> procedure "write" must have at least one actual parameter.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.13.4. 引用参数对应的实参只能是普通变量、参数或者数组元素

* 预期结果

针对测试用例中出现的将常量、复杂表达式作为引用的实参的情况，分别报引用参数对应的实参种类错误

* 实际测试结果

[Referenced actual parameter error!] <Line 13> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter or array element.

[Referenced actual parameter error!] <Line 15> The 1th actual parameter expression should be a normal variable、value parameter、referenced parameter or array element.

[Referenced actual parameter error!] <Line 15> The 2th actual parameter expression should be a normal variable、value parameter、referenced parameter or array element.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.13.5. 形式参数与函数、过程或主程序同名

* 预期结果

报形参与函数或过程或主程序同名的错误

* 实际测试结果

[Invalid formal parameter name!] <Line 3>The formal parameter "test" has the same name as the program "test" at line 1.

[Invalid formal parameter name!] <Line 10>The formal parameter "swap1" has the same name as the procedure "swap1" at line 3.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.13.6. 为自定义类型名赋值

* 预期结果

报为类型名赋值的错误

* 实际测试结果

[Type name as l-value error!] <Line7> Type name "b" can't be referenced as l-value.

[Type name as l-value error!] <Line8> Type name "a" can't be referenced as l-value.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 2.14. integer到real隐式类型转换

### 2.14.1. 赋值语句中的类型转换

* 预期结果

输出用integer变量为real变量赋值的警告信息

* 实际测试结果

[Implicit type conversion warning!] <Line12> Assign a integer varible to a real variable.

[Implicit type conversion warning!] <Line13> Assign a integer varible to a real variable.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.14.2. 传参时的类型转换

* 预期结果

输出实参类型为integer而形参类型为real的警告信息

* 实际测试结果

[Implicit type conversion warning!] <Line12> The 1th actual parameter of procedure call is integer while the corresponding formal parameter is real.

[Implicit type conversion warning!] <Line12> The 2th actual parameter of procedure call is integer while the corresponding formal parameter is real.

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

### 2.14.3. 整型常量超出范围

* 预期结果

报常量超出范围并自动转为real类型的警告

* 实际测试结果

[Implicit type conversion warning!] <Line2> The integer constant "c" is out of range.Automatically converted to real!

* 测试结果分析

测试结果符合预期，语义分析程序继续运行

## 2.15. 变量非法引用

### 2.15.1. 非法函数名作为赋值语句左值

* 预期结果

报函数名不能作为左值错误并报赋值语句左右两侧类型不符错误。

* 实际测试结果

[Invalid reference!] <Line 10> function name "fun1" can't be referenced as l-value.

[Assign statement type mismatch!] <Left at line 10, right at line 10> Left "fun1" type is error while right "a+b" type is integer.

* 测试结果分析

测试结果符合预期

### 2.15.2. 引用不存在的结构体属性

* 预期结果

报结构体属性非法引用错误

* 实际测试结果

[Invalid reference!]<Line 11> record member reference is illegal

[Assign statement type mismatch!] <Left at line 11, right at line 11> Left "u.s" type is error while right "1" type is integer.

* 测试结果分析

不仅报了预期的结构体属性非法引用错误，由于非法的结构体属性引用作为赋值语句的左值，还报了赋值符号左右类型不一致错误。程序正常运行。

### 2.15.3. 数组元素引用维度超出定义

* 预期结果

报数组元素非法引用错误

* 实际测试结果

[Invalid reference!]<Line 12> illegal array element access or illegal record attribute access

[Assign statement type mismatch!] <Left at line 12, right at line 12> Left "s[1][2][3]" type is error while right "1" type is integer.

* 测试结果分析

不仅报了预期的数组元素非法引用错误，由于非法的数组元素引用作为赋值语句的左值，还报了赋值符号左右类型不一致错误。程序正常运行。

### 2.15.4. 非结构体变量引用结构体属性

* 预期结果

报结构体属性非法引用错误

* 实际测试结果

[Invalid reference!]<Line 12> illegal array element access or illegal record attribute access

[Assign statement type mismatch!] <Left at line 12, right at line 12> Left "u.b.b" type is error while right "1" type is integer.

* 测试结果分析

不仅报了预期的结构体属性非法引用错误，由于非法的结构体属性引用作为赋值语句的左值，还报了赋值符号左右类型不一致错误。程序正常运行。

## 2.16. 嵌套层数过多

### 2.16.1. 函数或过程调用层数过多（超出一层）

* 预期结果

报出超出调用层数的函数以及位置

* 实际测试结果

[too much layers!] in "func" at line 9

* 测试结果分析

输出代码错误原因，以及发生错误的函数名以及错误位置的行号

### 2.16.2. 函数或过程调用层数过多（超出多层）

* 预期结果

仅报出超出调用层数的函数以及位置，而不报出更深层次的函数（直接忽略）

* 实际测试结果

[too much layers!] in "inner4" at line 9

* 测试结果分析

输出代码错误原因，以及发生错误的函数名以及错误位置的行号。更深层次的函数名不会出现因为在超出的那一层就已经跳出递归循环了，后面的递归也不会再进行

### 2.16.3. 并行函数或过程调用层数过多

* 预期结果

报出超出所有并行调用层数的函数以及位置

* 实际测试结果

[too much layers!] in "inner4" at line 9

[too much layers!] in "inner44" at line 13

* 测试结果分析

输出代码错误原因，以及发生错误的函数名以及错误位置的行号；会有多行是因为在同一层中有多个函数超出调用层次了;

# 测试用例附录

### 用例2.1.1

program test(input, output);

var x,y: integer;

write: char;

function read(a,b: integer):integer;

begin

read(x);

write(1);

end;

begin

end.

### 用例2.1.2

program test(input, output);

var x,y: integer;

test:char;

function fun(a,b: integer):integer;

var test:char;

begin

read(x);

write(1);

end;

begin

end.

### 用例2.1.3

program test(input, output);

var x,y: integer;

swap:boolean;

procedure swap;

begin

end;

begin

end.

### 用例2.2.1

program test(input,output);

var a,b:integer;

begin

for c:=a to a+b do

write(c);

end.

### 用例2.2.2

program test(input, output);

var x,y: integer;

function gtcomdiv(a,b: integer):integer;

    begin

    end;

begin

    x:=v(x,y);

end.

### 用例2.2.3

program test(input, output);

var

x:integer;

begin

x:=y

end.

### 用例2.3.1

program test(input,output);

var a,b:integer;

c:char;

begin

for c:=a to a+b do

write(c);

end.

### 用例2.3.2

program test(input, output);

var x,y: integer;g,h:char;

function gtcomdiv(a,b: integer):integer;

    begin

        if b=0 then gtcomdiv:=a

        else gtcomdiv:=gtcomdiv(b,a mod b)

    end;

begin

    x:=gtcomdiv(g,h)

end.

### 用例2.3.3

program test(input, output);

var x,y: integer; d,e: real;

function gtcomdiv(var a,b: real):real;

begin

gtcomdiv:=a;

end;

begin

read(x,y);

write(gtcomdiv(1.0,1.0));

end.

### 用例2.3.4

program test(input);

var a:integer;

b:real;

c:char;

d:boolean;

m : array[1..5] of integer;

n : record x:integer; y:real end;

begin

read(a,b,c,d);

case a+n.x of

0 : a:=0;

1 : a:=1

end;

case m of

0 : a:=0;

1 : a:=1

end;

case n of

0 : a:=0;

1 : a:=1

end;

end.

### 用例2.4.1

program test(input, output);

type

arr=array[1..10,2..7] of integer;

var

s,r:arr;

begin

s[1]:=1

end.

### 用例2.4.2

program test(input,output);

var x,y:integer;

c:boolean;

procedure swap(a,b: integer);

var z:integer;

begin

z:=a;

a:=b;

b:=z;

end;

begin

read(x,y);

swap(x,y,c);

write(x,y);

end.

### 用例2.4.3

program test(input, output);

var x,y: integer;

function gtcomdiv(a,b: integer):integer;

    begin

        if b=0 then gtcomdiv:=a

        else gtcomdiv:=gtcomdiv(b,a mod b)

    end;

begin

    x:=gtcomdiv(x)

end.

### 用例2.5.1

program test(input,output);

var a,b:integer;

begin

for input:=a to a+b do

write(c);

end.pas

### 用例2.5.2

program test(input,output);

var a,b:integer;

c:boolean;

function fun(var x,y:integer):integer;

begin

if x<=y

then fun:=1

else fun:=0;

end;

begin

read(a,b);

fun(a,b);

write(a,b);

end.

### 用例2.6.1

program test(input, output);

type

arr=array[1..10,1..7] of integer;

var

s,r:arr;

x:real;

begin

s[1][x]:=3

end.

### 用例2.6.2

program test(input,output);

var a,b:integer;

c:boolean;

d:real;

e:char;

begin

a:=1;

repeat

begin

write(a);

a:=a+1

end

until a+b;

end.

### 用例2.6.3

program test(input,output);

var a:integer;

b:char;

begin

a:=1;

while b do

begin

write(a);

a := a+1

end;

end.

### 用例2.6.4

program test(input,output);

var a,b:integer;

c:char;

begin

a:=1;

if b=c then

begin

a:=a+1;

write(a)

end

else

begin

a:=a+10;

write(a)

end;

end.

### 用例2.6.5

program test(input,output);

var a,b:integer;

c:char;

begin

for a:=c to a+b do

write(a);

for b:=b to a>b do

write(a+b)

end.

### 用例2.6.6

program test(input,output);

var x,y:char;

procedure swap(a,b: integer);

var z:integer;

begin

z:=a;

a:=b;

b:=z;

end;

begin

read(x,y);

swap(x,y);

write(x,y);

end.

### 用例2.7

program test(input,output);

var a,b,c:integer;

d:char;

function fun(x,y:integer):boolean;

begin

if x<=y

then fun:=true

else fun:=false;

end;

begin

read(a,b);

c:=fun(a,b);

d:=c;

write(c,d);

end.

### 用例2.8

program test(input, output);

type

arr=array[1..10,1..7] of integer;

var

s,r:arr;

begin

s[1][8]:=3

end.

### 用例2.9.1

program test(input, output);

var

x:boolean;

y:integer;

z:real;

begin

x:=true;

y:=1;

if(x=y) then z:=0.1

else z:=0.2;

end.

### 用例2.10.1

program test(input, output);

var

x:boolean;

y:integer;

begin

y:=3;

if(not y) then x:=true

else x:=false;

end.

### 用例2.10.2

program test(input, output);

var

y:integer;

begin

y:=-'a'

end.

### 用例2.10.3

program test(input, output);

var

x,z:integer;

y:char;

begin

y:='a';

z:=2;

x:=z+y

end.

### 用例2.10.4

program test(input, output);

var

x:integer;

y,z:boolean;

begin

x:=1;

z:=false;

y:= x and z

end.

### 用例2.11

program test(input,output);

const m=233;

var a,b:integer;

c:array[1..5] of integer;

function fun(var x,y:integer):integer;

begin

if x<=y

then fun:=1

else fun:=0;

end;

begin

read(a,b,c[1]);

read(fun(a,b));

read(c);

read(m);

end.

### 用例2.12.1、用例2.12.2、用例2.12.3

program test(input, output);

var

x,z:integer;

begin

z:=3;

x:=z / 0;

x:=z div 0;

x:=z mod 0

end.

### 用例2.13.1

program test(input,output);

const c = 1.5;

var a:integer;

begin

read(a);

for a := 1 to 10 do

c := c+1;

end.pa

### 用例2.13.2

program test(input,output);

var a,b,c:integer;

function fun(var x,y:integer):integer;

begin

if x<=y

then fun:=true

else fun:=false;

end;

begin

read(a,b);

c:=fun(a,b);

write(c);

end.

### 用例2.13.3

program test(input,output);

var x,y:integer;

procedure swap(a,b: integer);

var z:integer;

begin

z:=a;

a:=b;

b:=z;

end;

begin

read;

swap(x,y);

write;

end.

### 用例2.13.4

program test(input,output);

const c=20;

var a,b:integer;

procedure pro(var x,y:integer);

var z:boolean;

begin

if x<=y

then z:=true

else z:=false;

end;

begin

read(a,b);

pro(a,b);

pro(c,b);

pro(c,a+b);

end.

### 用例2.13.5

program test(input, output);

var x,y: integer;

procedure swap1(var test,b: integer);

var y:integer;

5 begin

y:=x;

x:=b;

b:=y;

end;

procedure swap2(swap1,y: integer);

var d:integer;

begin

d:=x;

x:=y;

y:=d;

end;

begin

read(x,y);

swap1(x,y);

write(x,y);

swap2(x,y);

write(x,y);

end.

### 用例2.13.6

{type类型定义与应用}

program test(input, output);

type a=integer;b=real;

var s:a; z:b;

begin

s:=1;

b:=0.01;

a:=-1;

end.

### 用例2.14.1

program test(input,output);

var a,b:integer;

r1,r2:real;

function fun(var x,y:integer):integer;

begin

if x<=y

then fun:=1

else fun:=0;

end;

begin

read(a,b);

r1:=fun(a,b);

r2:=a\*b-b;

write(r1,r2);

end.

### 用例2.14.2

program test(input,output);

var a,b:integer;

procedure pro(x,y:real);

var z:boolean;

begin

if x<=y

then z:=true

else z:=false;

end;

begin

read(a,b);

pro(a,b);

end.

### 用例2.14.3

program test(input, output);

const c=32768;

begin

end.

### 用例2.15.1

program test(input,output);

var a,b:integer;

c:boolean;

function fun1(var x,y:integer):integer;

begin

fun1:=x+y;

end;

begin

fun1:=a+b;

end.

### 用例2.15.2

program test(input, output);

type

rec=record a:arr;b:integer;c:real end;

var

u,v,w:rec;

begin

u.s:=1

end.

### 用例2.15.3

program test(input, output);

type

arr=array[1..10,1..7] of integer;

rec=record a:arr;b:integer;c:real end;

rec1=record a:rec;b:arr end;

var

s,r:arr;

u,v,w:rec;

u1,v1:rec1;

begin

s[1][2][3]:=1;

end.

### 用例2.15.4

program test(input, output);

type

arr=array[1..10,1..7] of integer;

rec=record a:arr;b:integer;c:real end;

rec1=record a:rec;b:arr end;

var

s,r:arr;

u,v,w:rec;

u1,v1:rec1;

begin

u.b.b:=1;

end.

### 用例2.16.1

program main(x); {Pascal\_S语法：program IDENTIFIER(IDENTIFIER,...);}

var a, b, ret: integer; {全局变量}

procedure test; {无参过程：省略括号}

procedure inner;

procedure inner1(a,b:integer;var ret:integer);

procedure inner3(c:integer);

procedure inner4(d:integer);

function func(a,b:integer):integer;

begin

func:=a+b;

end; { end of function func }

begin

d:=1;

end;

begin

c:=1;

end;

{inner1}

begin

a:=1;

b:=2;

a:=func(a,b);

ret:=a+b;

end; { end of procedure inner1 }

(\*不支持跨过程调用

procedure inner2(x,y:integer;var ret:integer);

begin

writeln(func(a,b));

end;

\*)

procedure noPara;

begin

write('n');

end; { end of procedure noPara }

{inner}

begin

inner1(a,b,ret);

(\*writeln(func(a,b));\*) {不支持隔层调用：func属于inner1，inner不能调用}

noPara;

end; { end of procedure inner }

{test}

begin

inner;

end; { end of procedure test }

{main}

begin

a:=10;

b:=20;

test;

end.

### 用例2.16.2

program main(x); {Pascal\_S语法：program IDENTIFIER(IDENTIFIER,...);}

var a, b, ret: integer; {全局变量}

procedure test; {无参过程：省略括号}

procedure inner;

procedure inner1(a,b:integer;var ret:integer);

function func(a,b:integer):integer;

procedure inner3(a:integer);

procedure inner4(b:integer);

procedure inner44(c:integer);

begin

c:=1;

end;

begin

b:=1;

end;

begin

a:=1;

end;

begin

func:=a+b;

end; { end of function func }

{inner1}

begin

a:=1;

b:=2;

a:=func(a,b);

ret:=a+b;

end; { end of procedure inner1 }

(\*不支持跨过程调用

procedure inner2(x,y:integer;var ret:integer);

begin

writeln(func(a,b));

end;

\*)

procedure noPara;

begin

write('n');

end; { end of procedure noPara }

{inner}

begin

inner1(a,b,ret);

(\*writeln(func(a,b));\*) {不支持隔层调用：func属于inner1，inner不能调用}

noPara;

end; { end of procedure inner }

{test}

begin

inner;

end; { end of procedure test }

{main}

begin

a:=10;

b:=20;

test;

end.

### 用例2.16.3

program main(x); {Pascal\_S语法：program IDENTIFIER(IDENTIFIER,...);}

var a, b, ret: integer; {全局变量}

procedure test; {无参过程：省略括号}

procedure inner;

procedure inner1(a,b:integer;var ret:integer);

function func(a,b:integer):integer;

procedure inner3(a:integer);

procedure inner4(b:integer);

begin

b:=1;

end;

procedure inner44(c:integer);

begin

c:=1;

end;

begin

a:=1;

end;

begin

func:=a+b;

end; { end of function func }

{inner1}

begin

a:=1;

b:=2;

a:=func(a,b);

ret:=a+b;

end; { end of procedure inner1 }

(\*不支持跨过程调用

procedure inner2(x,y:integer;var ret:integer);

begin

writeln(func(a,b));

end;

\*)

procedure noPara;

begin

write('n');

end; { end of procedure noPara }

{inner}

begin

inner1(a,b,ret);

(\*writeln(func(a,b));\*) {不支持隔层调用：func属于inner1，inner不能调用}

noPara;

end; { end of procedure inner }

{test}

begin

inner;

end; { end of procedure test }

{main}

begin

a:=10;

b:=20;

test;

end.