# 目录

[1 目录 1](#_Toc513585121)

[2 代码生成测试 1](#_Toc513585122)

[2.1 测试环境与计划 1](#_Toc513585123)

[2.1.1 测试环境与依赖 1](#_Toc513585124)

[2.1.2 测试计划 1](#_Toc513585125)

[2.2 单语法成分测试 2](#_Toc513585126)

[2.2.1 write/writeln语句 2](#_Toc513585127)

[2.2.2 read语句 4](#_Toc513585128)

[2.2.3 头文件 7](#_Toc513585129)

[2.2.3.1 不带输入输出语句 7](#_Toc513585130)

[2.2.3.2 带输入输出语句 9](#_Toc513585131)

[2.2.3.3 对比分析 10](#_Toc513585132)

[2.2.4 运算符和表达式 11](#_Toc513585133)

[2.2.5 引用参数与指针 14](#_Toc513585134)

[2.2.6 变量和常量定义 18](#_Toc513585135)

[2.2.7 程序声明 21](#_Toc513585136)

[2.2.8 数组定义与数组元素引用 23](#_Toc513585137)

[2.2.9 返回值语句 25](#_Toc513585138)

[2.2.10 程序的无参调用 28](#_Toc513585139)

[2.2.11 各种语句混合与缩进 30](#_Toc513585140)

[2.3 综合测试 33](#_Toc513585141)

[2.3.1 快速排序 33](#_Toc513585142)

[2.3.2 最大公因数 36](#_Toc513585143)

# 代码生成测试

## 测试环境与计划

### 测试环境与依赖

* 测试在visual studio 搭建的工程中进行
* 需要词法分析程序能够处理无词法错误的程序并输出记号序列
* 需要语法分析程序能够处理无语法错误的记号序列，并输出抽象语法树
* 需要语义分析程序能够处理无语义错误的情况，并提供符号表接口
* 主程序的命令行接口全部打开，即自动编译生成的C程序，并自动执行生成的可执行文件

### 测试计划

* 有针对性地测试每一种语法成分的代码生成结果
* 编写冒泡排序、快速排序、最大公因数等程序进行综合测试

## 单语法成分测试

### write/writeln语句

* PASCAL-S源程序

program test(input,output);

const a=1; //整型常量

b=3.0; //浮点型常量

c='c'; //字符型常量

blank=' '; //空格

var d:integer; //整型变量

e:real; //浮点型变量

f:char; //字符型变量

g:boolean; //布尔型变量

begin

d:=6;

e:=6.0;

f:='f';

g:=2>1;//不支持true常量，所以这么写

writeln(a,blank,b,blank,c,blank,1,blank,1.0,blank,'y');

write(a,blank,b,blank,c,blank,1,blank,1.0,blank,'y');

writeln;

writeln(g);

writeln(a+b,blank,a\*b,blank,g,blank,a>b,blank,c,blank,d,blank,e,blank,d<>e);

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

const int a = 1;

const float b = 3.0;

const char c = 'c';

const char blank = ' ';

//Overall variable definition

int d;

float e;

char f;

bool g;

//Subprogram declaration

void test();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    d = 6;

    e = 6.0;

    f = 'f';

    g = 2 > 1;

    printf("%d%c%f%c%c%c%d%c%f%c%c\n", a, blank, b, blank, c, blank, 1, blank, 1.0, blank, 'y');

    printf("%d%c%f%c%c%c%d%c%f%c%c", a, blank, b, blank, c, blank, 1, blank, 1.0, blank, 'y');

    printf("\n");

    {

        if(g)

            printf("true");

        else

            printf("false");

    }

    printf("\n");

    printf("%d%c%f%c", a, blank, b, blank);

    {

        if(g)

            printf("true");

        else

            printf("false");

    }

    printf("%c", blank);

    {

        if(a > b)

            printf("true");

        else

            printf("false");

    }

    printf("%c%c%c%d%c%f%c", blank, c, blank, d, blank, e, blank);

    {

        if(d != e)

            printf("true");

        else

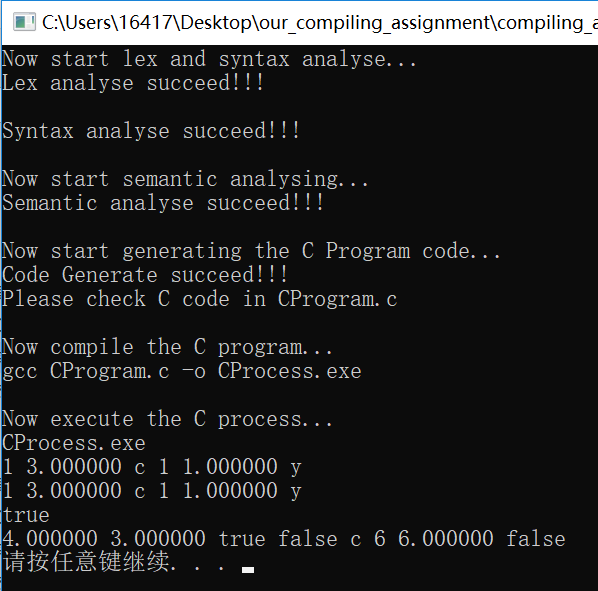
            printf("false");

    }

    printf("\n");

}

* 控制台输出及运行结果



* 结果分析
* 编译成功，运行成功
* 生成的C代码中，writeln不带参数地调用，转化为printf("\n");
* 生成的C代码中，输出bool类型时，会转化为用大括号括起来的if判断语句
* 生成的C代码中，不同格式的表达式正确对应到不同的格式控制符
* 运行后，成功输出所有预设的值

### read语句

* PASCAL-S源程序

program test(input,output);

var a,i:integer; //整型变量

b:real; //浮点型变量

c:char; //字符型变量

d:array[1..6] of integer; //数组

procedure pro(var a:integer);

begin

read(a);

end;

begin

read(c,a); //读取字符和整数

for i:=1 to 6 do

pro(d[i]); //循环加调用程序读取6个整数

read(b); //读取浮点数

writeln(c,' ',a); //输出字符和整数

for i:=1 to 6 do //输出6个整数，空格分隔

write(d[i],' ');

writeln; //输出换行

writeln(b); //输出浮点数

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int a;

int i;

float b;

char c;

int d[6];

//Subprogram declaration

void test();

void pro(int \*a);

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    scanf("%c%d", &c, &a);

    for(i = 1; i <= 6; i++)

        pro(&d[i - 1]);

    scanf("%f", &b);

    printf("%c%c%d\n", c, ' ', a);

    for(i = 1; i <= 6; i++)

        printf("%d%c", d[i - 1], ' ');

    printf("\n");

    printf("%f\n", b);

}

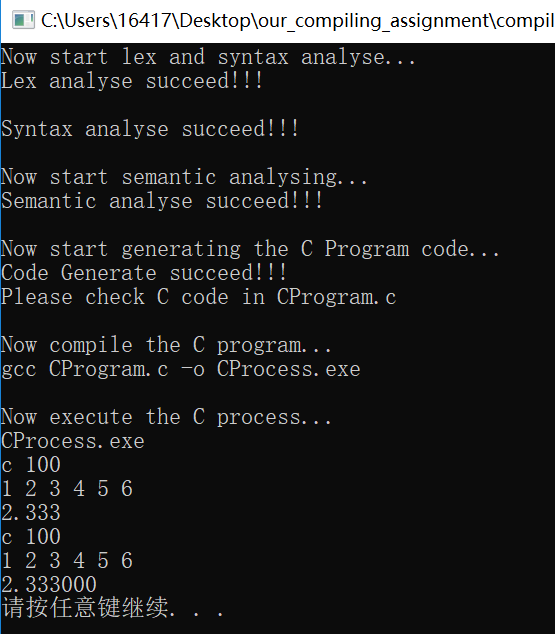
void pro(int \*a)

{

    scanf("%d", a);

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + 生成的C代码中，不同格式的表达式正确对应到不同的格式控制符
  + 生成的C代码中，读取到普通变量时，正确添加了取地址符
  + 生成的C代码中，读取到引用(指针)参数时，不需要添加取地址符，因为指针本身就已经代表地址了
  + 运行后，正确读取所有值，并原样输出所有值

### 头文件

#### 不带输入输出语句

* PASCAL-S源程序

program test(input,output);

var a,b:integer;

begin

a:=3\*b;

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int a;

int b;

//Subprogram declaration

void test();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

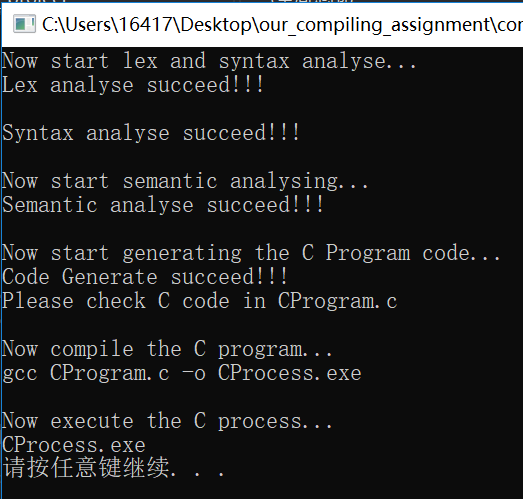
void test()

{

    a = 3 \* b;

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + 输出的C代码中，头文件只包含stdbool.h

#### 带输入输出语句

* PASCAL-S源程序

program test(input,output);

var a,b:integer;

begin

read(b);

a:=3\*b;

writeln(a);

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int a;

int b;

//Subprogram declaration

void test();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

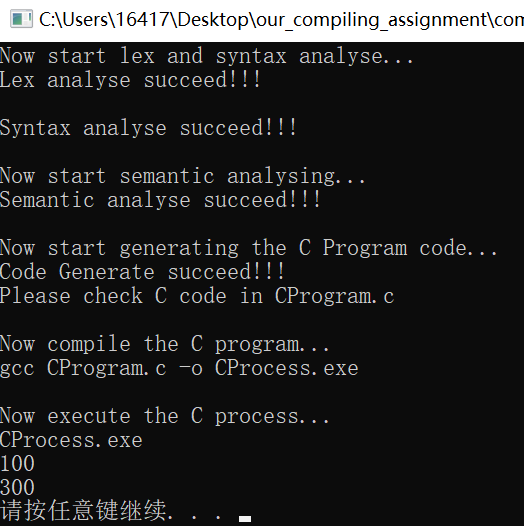
    scanf("%d", &b);

    a = 3 \* b;

    printf("%d\n", a);

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + 输出的C代码中，头文件不仅包含stdbool.h，还包含stdio.h

#### 对比分析

由于boolean类型是pascal所支持的基本类型，所以无论PASCAL-S源程序是否用到了boolean类型变量，输出的C代码中头文件都要包含stdbool.h，而stdio.h需要根据PASCAL-S源程序是否用到read、write、writeln调用。

### 运算符和表达式

* PASCAL-S源程序

program test(input,output);

const a=3;

b=1.0;

c='c';

var d,e,f:integer;

g,h,i:real;

j,k,l:char;

m,n,o:boolean;

function fun(a:integer):integer;

begin

exit(a+1);

end;

begin

d:=5; e:=10;

g:=0.25; h:=0.5;

j:='a'; k:='b';

f:= d mod (e div fun(d)); //涉及mod、div和函数调用的复杂表达式

writeln(f); //类型为integer

g:= --d + e / d \* ( - b); //涉及加减乘除、取相反数、加括号的复杂表达式

writeln(g); //类型为real

o:= ( ( f mod e ) <> d ) or ( a > b); //涉及<>、or的复杂表达式

writeln(o); //类型为boolean

m:= not not ( ( d + e ) = f ); //涉及取非、=的复杂表达式

writeln(m); //类型为boolean

n:= ( g \* e < f ) and ( (f mod e) >= d ); //涉及and、<、>=、mode的复杂表达式

writeln(n); //类型为boolean

writeln( j <= c); //比较char的大小关系

writeln( m > n); //比较boolean的大小关系

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

const int a = 3;

const float b = 1.0;

const char c = 'c';

//Overall variable definition

int d;

int e;

int f;

float g;

float h;

float i;

char j;

char k;

char l;

bool m;

bool n;

bool o;

//Subprogram declaration

void test();

int fun(int a);

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    d = 5;

    e = 10;

    g = 0.25;

    h = 0.5;

    j = 'a';

    k = 'b';

    f = d % (e / fun(d));

    printf("%d\n", f);

    g = - ( - d) + e / d \* ( - b);

    printf("%f\n", g);

    o = ((f % e) != d) || (a > b);

    {

        if(o)

            printf("true");

        else

            printf("false");

    }

    printf("\n");

    m = ! ( ! ((d + e) == f));

    {

        if(m)

            printf("true");

        else

            printf("false");

    }

    printf("\n");

    n = (g \* e < f) && ((f % e) >= d);

    {

        if(n)

            printf("true");

        else

            printf("false");

    }

    printf("\n");

    {

        if(j <= c)

            printf("true");

        else

            printf("false");

    }

    printf("\n");

    {

        if(m > n)

            printf("true");

        else

            printf("false");

    }

    printf("\n");

}

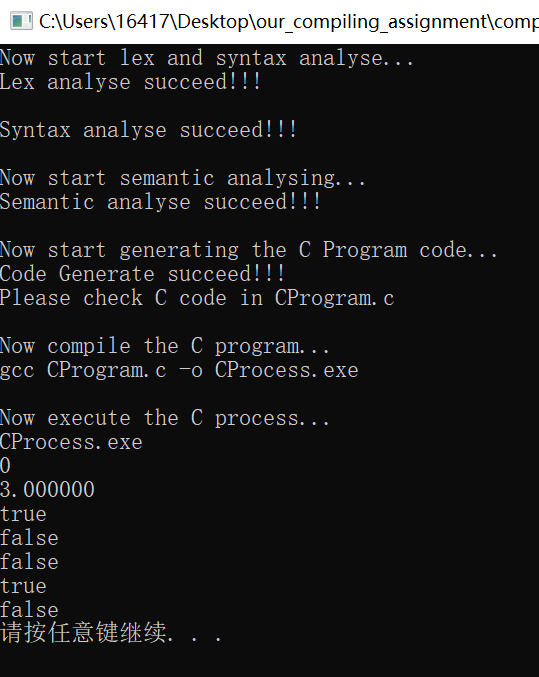
int fun(int a)

{

    return (a + 1);

}

* 控制台输出及运行结果



* 结果分析
  + 生成的C代码中，各种运算符均转化正确
  + 生成的C代码中，对于a:=--b这种连着的单目运算符，添加了括号，更加易读
  + 生成的C代码中，对于不同类型的表达式，输出格式控制符正确
  + bool类型表达式的输出转化为了if-else语句
  + 输出结果符合预设值

### 引用参数与指针

* PASCAL-S源程序

program test(input, output);

var ans:integer;

{

求斐波那契数列值的过程，利用引用参数保存结果;

n表示求第n项斐波那契数列值;

fib(0)=1, fib(1)=1, ……, fib(n)=fib(n-1)+fib(n-2);

}

procedure fib(n:integer;var res:integer);

var tmp1,tmp2:integer;

begin

if n=0 then

res:=1

else

if n=1 then

res:=1

else

begin

fib(n-1,tmp1);

fib(n-2,tmp2);

res:=tmp1+tmp2;

end;

end;

{

自己编写的输出函数，可以实现输出n次out值;

递归调用自身，n每次减1;

out值采用的是引用参数;

}

procedure mywrite(n:integer; var out:integer);

begin

if n=0 then

exit

else

begin

writeln(out);

mywrite(n-1, out);

end;

end;

begin

fib(10, ans); //ans=fib(10)

mywrite(3, ans); //输出ans 3次

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int ans;

//Subprogram declaration

void test();

void fib(int n, int \*res);

void mywrite(int n, int \*out);

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    fib(10, &ans);

    mywrite(3, &ans);

}

void fib(int n, int \*res)

{

    int tmp1;

    int tmp2;

    if(n == 0)

        \*res = 1;

    else

        if(n == 1)

            \*res = 1;

        else

        {

            fib(n - 1, &tmp1);

            fib(n - 2, &tmp2);

            \*res = tmp1 + tmp2;

        }

}

void mywrite(int n, int \*out)

{

    if(n == 0)

        return;

    else

    {

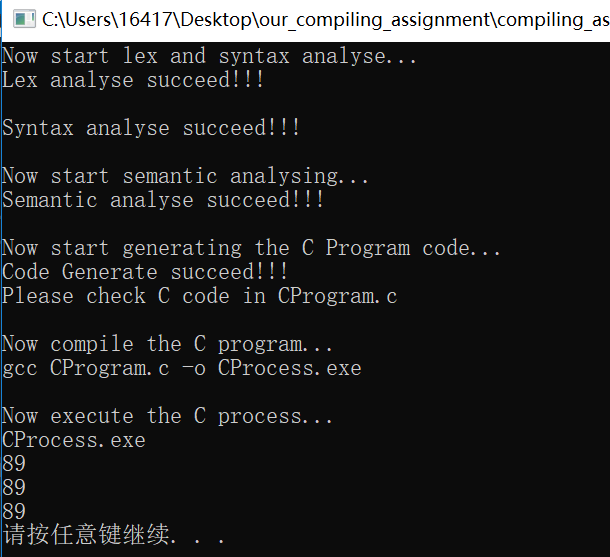
        printf("%d\n", \*out);

        mywrite(n - 1, out);

    }

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + PASCAL-S源程序中普通变量作为引用形参对应的实参调用时，在生成的C代码中加上&取地址符
  + PASCAL-S源程序中引用参数作为引用形参对应的实参调用时，在生成的C代码中无需前置任何符号，因为引用参数对应到C代码中已经是指针了，所以调用时，传本身就已经是地址了。
  + PASCAL-S源程序中，用到引用参数的地方（除了2中的情况），在生成的C代码中，都需加上前置的"\*"，表示解引用
  + 成功输出fib(10)=89 三次

### 变量和常量定义

* PASCAL-S源程序

program test(input, output);

const a=1; //整型常量定义

b=2.0; //浮点型常量定义

c='c'; //字符型常量定义

var d,dd,ddd:integer; //整型变量定义

e,ee,eee:real; //浮点型变量定义

f,ff,fff:char; //字符型变量定义

g,gg,ggg:boolean; //boolean型变量定义

h,hh,hhh:array[1..3] of integer; //整型数组定义

i,ii,iii:array[1..3, 4..9] of real; //浮点型数组定义

j,jj,jjj:array[1..3, 4..9, 10..18] of char; //字符型数组定义

k,kk,kkk:array[1..3, 4..9, 10..18, 19..30] of boolean; //布尔型数组定义

procedure pro;

const a=1; //整型常量定义

b=2.0; //浮点型常量定义

c='c'; //字符型常量定义

var d,dd,ddd:integer; //整型变量定义

e,ee,eee:real; //浮点型变量定义

f,ff,fff:char; //字符型变量定义

g,gg,ggg:boolean; //boolean型变量定义

h,hh,hhh:array[1..3] of integer; //整型数组定义

i,ii,iii:array[1..3, 4..9] of real; //浮点型数组定义

j,jj,jjj:array[1..3, 4..9, 10..18] of char; //字符型数组定义

k,kk,kkk:array[1..3, 4..9, 10..18, 19..30] of boolean; //布尔型数组定义

begin

end;

begin

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdbool.h>

//Overall constant definiton

const int a = 1;

const float b = 2.0;

const char c = 'c';

//Overall variable definition

int d;

int dd;

int ddd;

float e;

float ee;

float eee;

char f;

char ff;

char fff;

bool g;

bool gg;

bool ggg;

int h[3];

int hh[3];

int hhh[3];

float i[3][6];

float ii[3][6];

float iii[3][6];

char j[3][6][9];

char jj[3][6][9];

char jjj[3][6][9];

bool k[3][6][9][12];

bool kk[3][6][9][12];

bool kkk[3][6][9][12];

//Subprogram declaration

void test();

void pro();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

}

void pro()

{

    const int a = 1;

    const float b = 2.0;

    const char c = 'c';

    int d;

    int dd;

    int ddd;

    float e;

    float ee;

    float eee;

    char f;

    char ff;

    char fff;

    bool g;

    bool gg;

    bool ggg;

    int h[3];

    int hh[3];

    int hhh[3];

    float i[3][6];

    float ii[3][6];

    float iii[3][6];

    char j[3][6][9];

    char jj[3][6][9];

    char jjj[3][6][9];

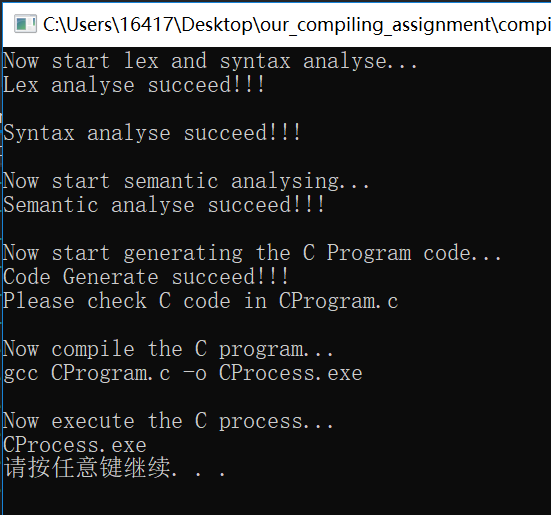
    bool k[3][6][9][12];

    bool kk[3][6][9][12];

    bool kkk[3][6][9][12];

}

* 控制台输出及运行结果



* 结果分析
  + 生成的C代码中，各种类型的常量定义均正确的加上了类型关键字，类型关键字的转化正确;
  + 生成的C代码中，各种类型的变量定义正确;
  + 生成的C代码中，各种类型的数组定义正确，各维大小正确;

### 程序声明

* PASCAL-S源程序

program test(input,output);

procedure pro(var a,b,c:integer; d:real; e,f:char; var g,h,i:boolean);

begin

end;

function fun(a:integer; var b,c,d:real; var e,f,g:boolean; h,i:char):real;

begin

fun:=1;

end;

begin

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

//Subprogram declaration

void test();

void pro(int \*a, int \*b, int \*c, float d, char e, char f, bool \*g, bool \*h, bool \*i);

float fun(int a, float \*b, float \*c, float \*d, bool \*e, bool \*f, bool \*g, char h, char i);

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

}

void pro(int \*a, int \*b, int \*c, float d, char e, char f, bool \*g, bool \*h, bool \*i)

{

}

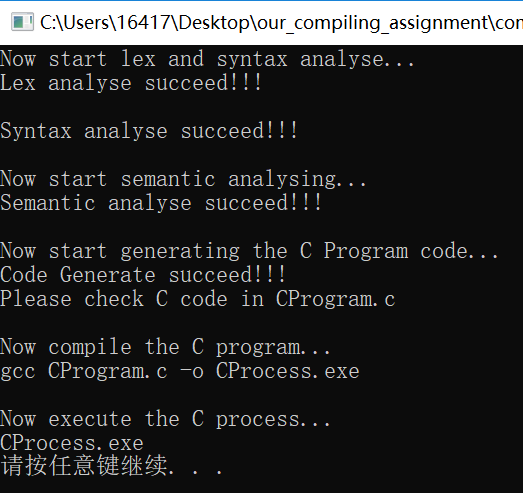
float fun(int a, float \*b, float \*c, float \*d, bool \*e, bool \*f, bool \*g, char h, char i)

{

    return (1);

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + 生成的C代码中，子程序声明正确;
  + 生成的C代码中，子程序定义的子程序头正确;
  + 生成的C代码中，子程序声明和子程序头的参数的声明顺序与PASCAL-S源程序一致
  + 生成的C代码中，子程序声明和子程序头的参数中，PASCAL-S源程序的引用参数成功的转化为了指针

### 数组定义与数组元素引用

* PASCAL-S源程序

program test(input,output);

var i,j,k:integer;

a:array[0..1, 2..3, 4..5] of integer; //数组定义

begin

for i:=0 to 1 do

for j:=2 to 3 do

for k:=4 to 5 do

read(a[i,j,k]); //三重循环，读入每一个数组元素，共8个

for i:=0 to 1 do

for j:=2 to 3 do

for k:=4 to 5 do

writeln(a[i,j,k]) //三重循环，输出每一个数组元素，换行符分隔

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int i;

int j;

int k;

int a[2][2][2];

//Subprogram declaration

void test();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    for(i = 0; i <= 1; i++)

        for(j = 2; j <= 3; j++)

            for(k = 4; k <= 5; k++)

                scanf("%d", &a[i][j - 2][k - 4]);

    for(i = 0; i <= 1; i++)

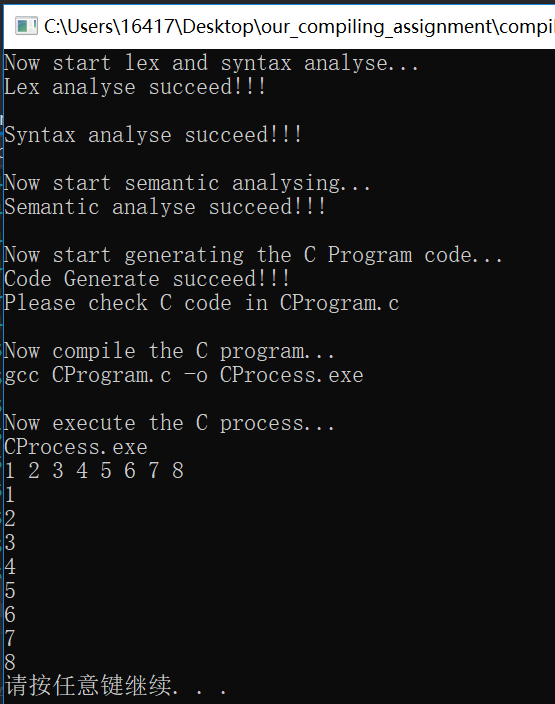
        for(j = 2; j <= 3; j++)

            for(k = 4; k <= 5; k++)

                printf("%d\n", a[i][j - 2][k - 4]);

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + 生成的C代码中，数组定义的各维大小正确
  + 生成的C代码中，两个三重循环中，对数组元素的引用，下标表达式添加了正确的偏移量
  + 输入的8个元素，按照原样输出，以换行符分隔

### 返回值语句

* PASCAL-S源程序

program test(input,output);

var b,c,d:integer;

procedure pro;

begin

exit;

end;

function fun(a:integer):char;

begin

if a<60 then

exit('l')

else

if a=60 then

fun:='m'

else

exit('h');

end;

function fun2(a:integer; var b,c,d:integer):integer;

begin

if a=0 then

fun2:=b+c+d

else

if a=1 then

fun2:=b\*c\*d

else

exit(-b-c-d);

end;

begin

read(b,c,d);

writeln(fun(b), ' ', fun(c), ' ', fun(d));

writeln(fun2(0, b, c, d), ' ', fun2(1, b, c, d), ' ', fun2(2, b, c, d));

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int b;

int c;

int d;

//Subprogram declaration

void test();

void pro();

char fun(int a);

int fun2(int a, int \*b, int \*c, int \*d);

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    scanf("%d%d%d", &b, &c, &d);

    printf("%c%c%c%c%c\n", fun(b), ' ', fun(c), ' ', fun(d));

    printf("%d%c%d%c%d\n", fun2(0, &b, &c, &d), ' ', fun2(1, &b, &c, &d), ' ', fun2(2, &b, &c, &d));

}

void pro()

{

    return;

}

char fun(int a)

{

    if(a < 60)

        return ('l');

    else

        if(a == 60)

            return ('m');

        else

            return ('h');

}

int fun2(int a, int \*b, int \*c, int \*d)

{

    if(a == 0)

        return (\*b + \*c + \*d);

    else

        if(a == 1)

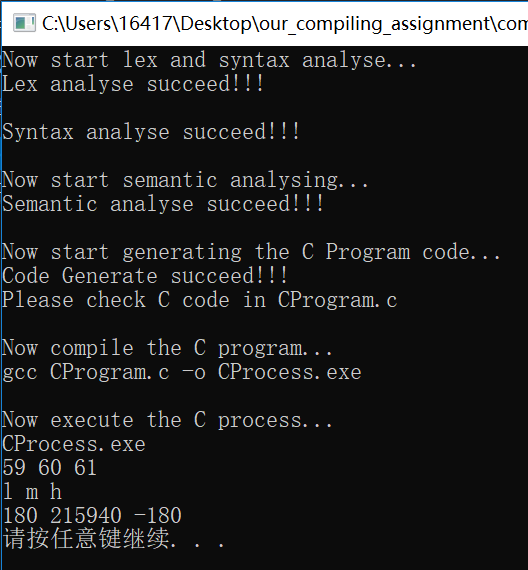
            return (\*b \* \*c \* \*d);

        else

            return ( - \*b - \*c - \*d);

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功;
  + 赋值形式的返回值语句正确的转化为了C程序中的return语句;
  + exit带参数的返回值语句正确的转化为了C程序中的return语句;
  + exit不带参数的返回语句正确的转化为了C程序中的return语句;
  + 输出结果符合预设值

### 程序的无参调用

* PASCAL-S源程序

program test(input,output);

var a:integer;

procedure pro;

begin

writeln(a);

end;

function fun:integer;

begin

fun:=100;

end;

begin

writeln(fun);

a:=fun mod 3;

pro;

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int a;

//Subprogram declaration

void test();

void pro();

int fun();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    printf("%d\n", fun());

    a = fun() % 3;

    pro();

}

void pro()

{

    printf("%d\n", a);

}

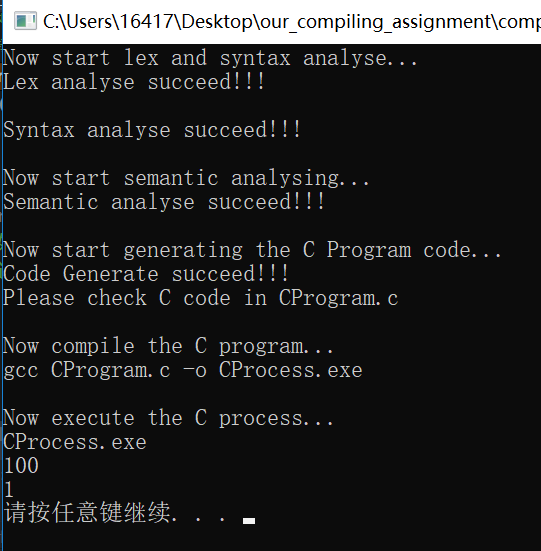
int fun()

{

    return (100);

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + 过程的无参调用正确的转化为了C代码，添加了一对括号
  + 函数的无参调用正确的转化为了C代码，添加了一对括号

### 各种语句混合与缩进

* PASCAL-S源程序

program test(input, output);

var i:integer;

a:array[1..10] of integer;

procedure swap(var a,b:integer); //交换两个数的值

var tmp:integer;

begin

tmp:=a;

a:=b;

b:=tmp;

end;

procedure bubbleSort; //冒泡排序

var i,j:integer;

begin

for i:= 1 to 10 do //for语句

begin //compound语句

j:=i+1;

while j <= 10 do //while语句

begin //compound语句

if a[i]>a[j] then //if语句

swap(a[i],a[j]);

j:=j+1;

end

end;

end;

begin

i:=1;

repeat //repeat语句

begin //compound语句

read(a[i]);

i:=i+1

end

until i=11;

bubbleSort;

for i:= 1 to 10 do //for语句

write(a[i],' ');

writeln;

end.

* 预期结果

没有词法、语法、语义错误，正确生成的C程序，可以用C编译器正确编译，并正确运行，输出结果正确。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int i;

int a[10];

//Subprogram declaration

void test();

void swap(int \*a, int \*b);

void bubblesort();

//Main function

int main()

{

    test();

    return 0;

}

//Subprogram definition

void test()

{

    i = 1;

    do

    {

        scanf("%d", &a[i - 1]);

        i = i + 1;

    }

    while(!(i == 11));

    bubblesort();

    for(i = 1; i <= 10; i++)

        printf("%d%c", a[i - 1], ' ');

    printf("\n");

}

void swap(int \*a, int \*b)

{

    int tmp;

    tmp = \*a;

    \*a = \*b;

    \*b = tmp;

}

void bubblesort()

{

    int i;

    int j;

    for(i = 1; i <= 10; i++)

    {

        j = i + 1;

        while(j <= 10)

        {

            if(a[i - 1] > a[j - 1])

                swap(&a[i - 1], &a[j - 1]);

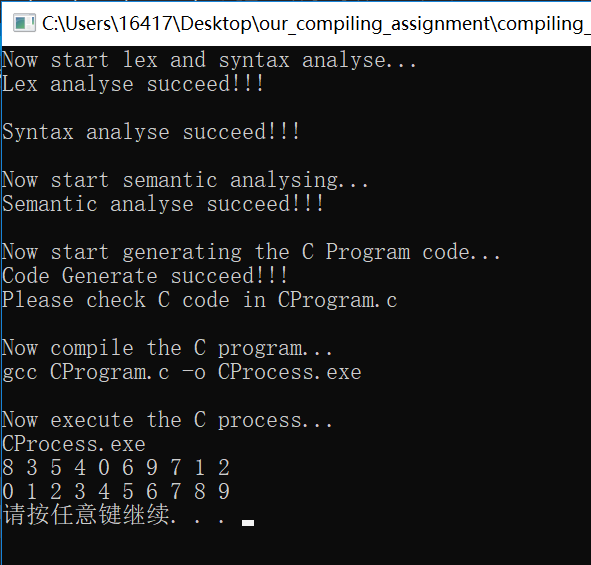
            j = j + 1;

        }

    }

}

* 控制台输出及运行结果



* 结果分析
  + 编译成功，运行成功
  + if语句转化正确
  + while语句转化正确
  + for语句转化正确
  + repeat语句转化正确，特别是条件表达式，需要取非
  + compound语句转化正确
  + 语句的缩进美观，且符合设计
  + 输入的10个数正确的被排序，按照从小到大的顺序输出

## 综合测试

### 快速排序

* PASCAL-S源程序

program quicksort(input,output);

var

n,i:integer;

a:array[0..100000] of integer;

procedure kp(l,r:integer);

var

i,j,mid:integer;

begin

if l>=r then exit;

i:=l;j:=r;mid:=a[(l+r) div 2];

repeat

    begin

while a[i]<mid do i:=i+1;

while a[j]>mid do j:=j-1;

if i<=j then

begin

a[0]:=a[i];a[i]:=a[j];a[j]:=a[0];

i:=i+1;j:=j-1;

end

    end

until i>j;

kp(l,j);

kp(i,r)

end;

begin

read(n);

for i:=1 to n do

read(a[i]);

kp(1,n);

for i:=1 to n do

write(a[i],' ');

end.

* 预期结果

没有词法、语法、语义错误，正确生成快速排序的C程序，可以用C编译器正确编译，并正确运行。

C程序运行时，先读入一个数字n表示待排序的数字个数，再输入n个数字。最后输出从小到大排序的结果。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int n;

int i;

int a[100001];

//Subprogram declaration

void quicksort();

void kp(int l, int r);

//Main function

int main()

{

    quicksort();

    return 0;

}

//Subprogram definition

void quicksort()

{

    scanf("%d", &n);

    for(i = 1; i <= n; i++)

        scanf("%d", &a[i]);

    kp(1, n);

    for(i = 1; i <= n; i++)

        printf("%d%c", a[i], ' ');

}

void kp(int l, int r)

{

    int i;

    int j;

    int mid;

    if(l >= r)

        return;

    i = l;

    j = r;

    mid = a[(l + r) / 2];

    do

    {

        while(a[i] < mid)

            i = i + 1;

        while(a[j] > mid)

            j = j - 1;

        if(i <= j)

        {

            a[0] = a[i];

            a[i] = a[j];

            a[j] = a[0];

            i = i + 1;

            j = j - 1;

        }

    }

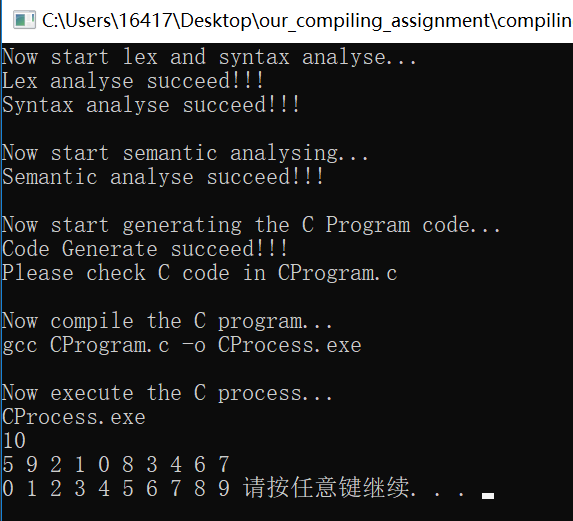
    while(!(i > j));

    kp(l, j);

    kp(i, r);

}

* 控制台输出及运行结果



* 结果分析

编译程序各部分均为发现错误，且成功生成C代码，通过编译生成可执行程序。快速排序运行结果正确，所有数字按照从小到大的顺序进行了输出。

### 最大公因数

* PASCAL-S源程序

program example(input,output);

var n,i,x,y:integer;

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

begin

    if b=0 then gcd:=a

    else gcd:=gcd(b,a mod b);

end;

begin

read(n);

for i:= 1 to n do

begin

read(x,y);

writeln(gcd(x,y))

end;

end.

* 预期结果

没有词法、语法、语义错误，正确生成求最大公因数的C程序，可以用C编译器正确编译，并正确运行。

C程序运行时，先读入一个数字n表示待求最大公因数的数字对数，再输入n对数字，每对数字有一个输出，表示这对数字的最大公因数。

* 输出的C程序

//Head files

#include<stdio.h>

#include<stdbool.h>

//Overall constant definiton

//Overall variable definition

int n;

int i;

int x;

int y;

//Subprogram declaration

void example();

int gcd(int a, int b);

//Main function

int main()

{

    example();

    return 0;

}

//Subprogram definition

void example()

{

    scanf("%d", &n);

    for(i = 1; i <= n; i++)

    {

        scanf("%d%d", &x, &y);

        printf("%d\n", gcd(x, y));

    }

}

int gcd(int a, int b)

{

    if(b == 0)

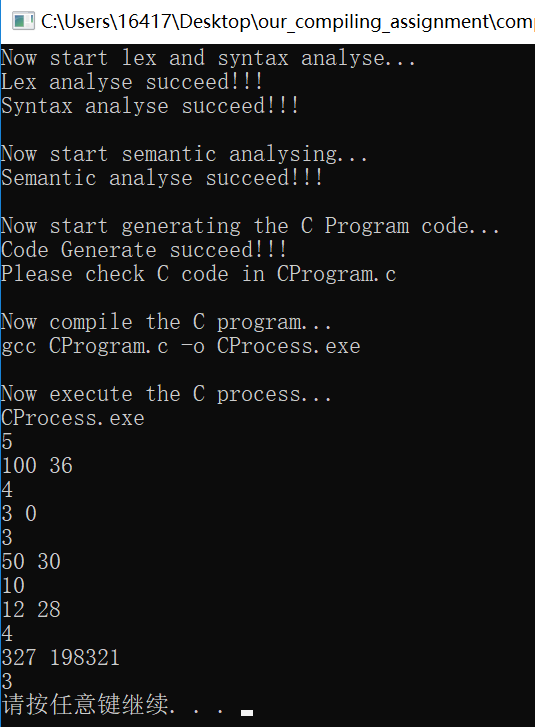
        return (a);

    else

        return (gcd(b, a % b));

}

* 控制台输出及运行结果



* 结果分析

编译程序各部分均为发现错误，且成功生成C代码，通过编译生成可执行程序。求最大公因数运行结果正确。