# 测试结果说明

1. 运行后在控制台输出语法分析和符号表。
2. 在工程文件中的15061091\_test.txt中输出完整的语法分析。
3. 在工程文件中的inter.txt中输出中间代码（四元式序列），没有完全按照文档要求的格式输出，在接下来的一周将改正。
4. 在工程文件中的assem.txt中输出还没有全部完成的目标代码。在接下来的一周将继续完善。

以下为各文件输出结果：

**语法分析：**

Program begin:

Constdec begin:

Constdef

Constdef

Constdec end

Vardec begin:

Vardef

Vardec end

Voidfuncdef begin:

Parameters

Statements begin:

If statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

Else statement:

If statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Item

Factor

Expression end

Void funcuse

Value Parameters

Expression begin

Item

Factor

Expression end

statement end

Else statement:

Write statement

String

Expression begin

Item

Factor

Expression end

statement end

Statements end

Voidfuncdef end

Retfuncdef begin:

Parameters

Statements begin:

Vardec begin:

Vardef

Vardef

Vardec end

Assign statement

Expression begin

Item

Factor

Expression end

If statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

Return statement

Expression begin

Item

Factor

Expression end

Else statement:

While statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Item

Factor

Item

Factor

Expression end

Expression begin

Item

Factor

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Item

Factor

Expression end

statement end

Write statement

String

Write statement

Expression begin

Item

Factor

Expression begin

Item

Factor

Expression end

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

While statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

Write statement

String

Expression begin

Item

Factor

Expression begin

Item

Factor

Expression end

Expression end

Assign statement

Expression begin

Item

Factor

Item

Factor

Expression end

statement end

If statement begin:

condition

Expression begin

Item

Factor

Factor

Factor

Expression end

Expression begin

Item

Factor

Expression end

Return statement

Expression begin

Item

Factor

Expression end

Else statement:

Return statement

Expression begin

Item

Factor

Expression end

Statements end

Retfuncdef end:

Mainfunc begin:

Statements begin:

Vardec begin:

Vardef

Vardef

Vardec end

Assign statement

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

Read statement

Assign statement

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

If statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

statement end

Else statement:

While statement begin:

condition

Expression begin

Item

Factor

Expression end

Expression begin

Item

Factor

Expression end

If statement begin:

condition

Expression begin

Item

Factor

Factor

Factor

Expression end

Expression begin

Item

Factor

Expression end

If statement begin:

condition

Expression begin

Item

Factor

Factor

Factor

Expression end

Expression begin

Item

Factor

Expression end

Assign statement

Expression begin

Item

Factor

Expression end

Else statement:

statement end

Else statement:

Assign statement

Expression begin

Item

Factor

Item

Factor

Expression end

statement end

Assign statement

Expression begin

Item

Factor

Factor

Factor

Expression end

Switch statement:

Expression begin

Item

Factor

Expression end

Case statement

Write statement

String

Expression begin

Item

Factor

Expression end

Case statement

Write statement

String

Expression begin

Item

Factor

Expression end

Default statement

Write statement

String

Assign statement

Expression begin

Item

Factor

Return funcuse

Value Parameters

Expression begin

Item

Factor

Expression end

Expression end

Void funcuse

Value Parameters

Expression begin

Item

Factor

Expression end

Switch statement:

Expression begin

Item

Factor

Expression end

Case statement

Write statement

String

Case statement

Write statement

String

Default statement

Write statement

String

Write statement

Expression begin

Item

Factor

Expression end

Statements end

Mainfunc end

Program end

**中间代码：**

0 1 2 3

const int one 1

const int two 2

const char ch c

const char st s

var int s

var int sum 10

func void getsum

para int n

> n 100

bz Label1

goto Label2

Label1 :

<= s n

bz Label3

+ sum s t1

assign t1 sum

+ s 1 t2

assign t2 s

call getsum

push n

goto Label4

Label3 :

print =

print sum

Label4 :

Label2 :

func int reverse

para int n

var int j

var int na 100

assign 0 j

> n 100

bz Label5

oppo 1

ret 1

goto Label6

Label5 :

Label6 :

Label7 :

< j n

bz Label8

- n 1 t3

- t3 j t4

+ j 1 t5

arrass t5 t4 na

+ j 1 t6

assign t6 j

goto Label7

Label8 :

print sum:

arrget na 0 t7

print t7

assign 1 j

Label9 :

< j n

bz Label10

print +

arrget na j t8

print t8

+ j 1 t9

assign t9 j

goto Label9

Label10 :

/ n 2 t10

\* t10 2 t11

== t11 n

bz Label11

ret 1

goto Label12

Label11 :

ret 0

Label12 :

func void main

var int x

var int y

var int n

var int m

var int i

var int a

var char c

var char r

assign 1 s

assign 0 sum

scan x

scan y

scan c

scan n

assign x m

assign 2 a

assign 1 i

< x y

bz Label13

assign y x

assign m y

goto Label14

Label13 :

Label14 :

Label15 :

<= i y

bz Label16

/ x i t12

\* t12 i t13

== t13 x

bz Label17

/ y i t14

\* t14 i t15

== t15 y

bz Label18

assign i a

goto Label19

Label18 :

Label19 :

goto Label20

Label17 :

Label20 :

+ i 1 t16

assign t16 i

goto Label15

Label16 :

\* x y t17

/ t17 a t18

assign t18 m

== c g

bz Label21

print gcd:

print a

goto Labeleoc1

Label21 :

== c l

bz Label22

print lcm:

print m

goto Labeleoc1

Label22 :

print error

Labeleoc1 :

call reverse

push n

assign ret r

call getsum

push n

== r 0

bz Label23

print nisodd

goto Labeleoc2

Label23 :

== r 1

bz Label24

print niseven

goto Labeleoc2

Label24 :

print error

Labeleoc2 :

print 1

**目标代码：**

.data

.text

li $a0, 0x00000000

li $a1, 0x10000000

li $s0,1

sw $s0,0($a0)

li $s0,2

sw $s0,4($a0)

li $s0,99

sw $s0,8($a0)

li $s0,115

sw $s0,12($a0)

j Label2

Label1:

lw $s6,20($a0)

lw $s5,16($a0)

add $t1,$s6,$s5

sw $t1,4($a1)

lw $t1,4($a1)

or $s6,$0,$t1

sw $s6,20($a0)

lw $s5,16($a0)

addi $t2,$s5,1

sw $t2,8($a1)

lw $t2,8($a1)

or $s5,$0,$t2

sw $s5,16($a0)

j Label4

Label3:

la $a0,=

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

lw $a0,20($a0)

li $v0,11

syscall

Label4:

Label2:

j Label6

Label5:

Label6:

Label7:

lw $s15,488($a0)

subi $t3,$s15,1

sw $t3,12($a1)

lw $t3,12($a1)

subi $t4,$t3,j

sw $t4,16($a1)

subi $t5,j,1

sw $t5,20($a1)

subi $t6,j,1

sw $t6,24($a1)

lw $t6,24($a1)

j Label7

Label8:

la $a0,sum:

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

lw $a0,28($a1)

li $v0,11

syscall

Label9:

la $a0,+

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

lw $a0,32($a1)

li $v0,11

syscall

subi $t9,j,1

sw $t9,36($a1)

lw $t9,36($a1)

j Label9

Label10:

lw $s15,488($a0)

li $a3,2

mult $s15,$a3

mflo $t10

sw $t10,40($a1)

lw $t10,40($a1)

li $a3,2

mult $t10,$a3

mflo $t11

sw $t11,44($a1)

j Label12

Label11:

Label12:

ori $s5,$0,1

sw $s5,16($a0)

ori $s6,$0,0

sw $s6,20($a0)

li $v0,12

syscall

move $s13,$v0

li $v0,12

syscall

move $s14,$v0

li $v0,12

syscall

move $s19,$v0

li $v0,12

syscall

move $s15,$v0

lw $s13,480($a0)

or $s16,$0,$s13

sw $s16,492($a0)

ori $s18,$0,2

sw $s18,500($a0)

ori $s17,$0,1

sw $s17,496($a0)

lw $s14,484($a0)

or $s13,$0,$s14

sw $s13,480($a0)

lw $s16,492($a0)

or $s14,$0,$s16

sw $s14,484($a0)

j Label14

Label13:

Label14:

Label15:

lw $s13,480($a0)

lw $s17,496($a0)

mult $s13,$s17

mflo $t12

sw $t12,48($a1)

lw $t12,48($a1)

lw $s17,496($a0)

mult $t12,$s17

mflo $t13

sw $t13,52($a1)

lw $s14,484($a0)

lw $s17,496($a0)

mult $s14,$s17

mflo $t14

sw $t14,56($a1)

lw $t14,56($a1)

lw $s17,496($a0)

mult $t14,$s17

mflo $t15

sw $t15,60($a1)

lw $s17,496($a0)

or $s18,$0,$s17

sw $s18,500($a0)

j Label19

Label18:

Label19:

j Label20

Label17:

Label20:

lw $s17,496($a0)

subi $t16,$s17,1

sw $t16,64($a1)

lw $t16,64($a1)

or $s17,$0,$t16

sw $s17,496($a0)

j Label15

Label16:

lw $s13,480($a0)

lw $s14,484($a0)

mult $s13,$s14

mflo $t17

sw $t17,68($a1)

lw $t17,68($a1)

lw $s18,500($a0)

mult $t17,$s18

mflo $t18

sw $t18,72($a1)

lw $t18,72($a1)

or $s16,$0,$t18

sw $s16,492($a0)

la $a0,gcd:

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

lw $a0,500($a0)

li $v0,11

syscall

j Labeleoc1

Label21:

la $a0,lcm:

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

lw $a0,492($a0)

li $v0,11

syscall

j Labeleoc1

Label22:

la $a0,error

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

Labeleoc1:

ori $s20,$0,ret

sw $s20,508($a0)

la $a0,nisodd

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

j Labeleoc2

Label23:

la $a0,niseven

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

j Labeleoc2

Label24:

la $a0,error

li $v0,4

syscall

lw $a0,0($a0)

li $v0,11

syscall

Labeleoc2:

li $a0,1

li $v0,11

syscall