# 语法分析结果

**题目要求:**

1. 语法分析程序需按文法规则，用递归子程序法对每种语法成分进行分析，并在结束每种语法程序分析之前，在控制台输出当前所识别的语法成分的信息，例如“This is a read statement！”  
2. 需提交可在机房机器上运行的完整工程文件、之前设计并改进完善的测试程序（命名为学号\_test.txt), 和测试结果文件（说明上述测试程序产生的输出结果，并说明该结果是否与预期的相符，如果不相符，存在什么问题，测试结果文件命名为“学号\_姓名\_语法分析结果.docx”），将上述文件放在一个命名为“学号\_姓名\_语法分析”的文件夹中，压缩为“学号.zip”，提交该zip文件。

我的测试程序主要是实现了计算组合数这样的一个功能。主要写了三个函数：

int factorial(int n) 递归求解n的阶乘  
int C(int m, int n) 用公式计算组合数C(n,m)  
int mymax(int x, int y) 计算x和y中的较大值

在之前的基础上增加了三个特殊判断的函数，来专门测试特定语句的输出：

void test\_switch() 测试switch语句

void test\_for(int Int) 测试for循环语句

test\_call() 测试函数调用语句

生成的规则如下，对于大多数语法成分，我都会输出This is a \*\*\*. 如果是带上参数或者是有个数的，我还会输出::: x variables诸如此类，最后会:::输出对应的内容。例如“**This is a const\_char statement::: 9elements::: const char ch1 = '+', ch2 = '-', ch3 = '\*', ch4 = '/', ch5 = '\_', ch6 = 'a', ch7 = '0', ch8 = '"', ch9 = '9'”** ,说明这是一句const语句，定义的是char类型的常量，有9个元素，后面是对应语句的内容。

具体代码见文件。只要源代码是正确的，我的程序就不会输出“!!!ERROR!!!”。在我自己的测试程序测试后，生成的文件如下，可以看到并没有出现!!!，代表没有出错，经逐个分析也可以看到是符合预期的正确结果。

在这个版本之后，我也写了一个带了一点错误处理的版本，主要是处理了跳过到分号和右花括号的情形，错误处理之后还会继续补充。

This is a const\_int statement::: 2elements::: const int N = 100, M = -10000

This is a const\_int statement::: 1elements::: const int K = 0

This is a const\_char statement::: 9elements::: const char ch1 = '+', ch2 = '-', ch3 = '\*', ch4 = '/', ch5 = '\_', ch6 = 'a', ch7 = '0', ch8 = '"', ch9 = '9'

This is a variable\_int statement::: 10 variables::: int a[100], i, j, n, m, k, x, y, z, mx

This is a variable\_char statement::: 2 variables:::

char ch, s[10000]

This is a function\_int statement mymax ::: 2 parameters::: int mymax(int x, int y)

This is a variable\_int statement::: 1 variables::: int ans

This is a factor statemnt::: x

This is a term statemnt::: x

This is a expression statemnt::: x

This is an assign statement::: ans = x

This is a factor statemnt::: y

This is a term statemnt::: y

This is a expression statemnt::: y

This is a factor statemnt::: x

This is a term statemnt::: x

This is a expression statemnt::: x

This is a condition ::: y > x

This is a factor statemnt::: y

This is a term statemnt::: y

This is a expression statemnt::: y

This is an assign statement::: ans = y

This is a if\_elsestatement::: if (y > x){

ans = y;

}

else

;

This is a factor statemnt::: ans

This is a term statemnt::: ans

This is a expression statemnt::: ans

This is a return statement::: return (ans)

siz = 2519 now = 346

This is a function\_int statement factorial ::: 1 parameters::: int factorial(int n)

This is a variable\_int statement::: 1 variables::: int res

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a expression statemnt::: 0

This is a condition ::: n < 0

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: -1

This is a return statement::: return (-1)

This is a ifstatement::: if (n < 0)

return (-1);

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 1

This is an assign statement::: res = 1

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 1

This is an assign statement::: res = 1

This is cases statement with::: 2 cases::: case 0: res = 1;

case 1: res = 1;

This is a factor statemnt::: n

This is a factor statemnt::: n

This is a term statemnt::: n

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: n-1

This is a parameter with ::: 1 parameters::: n-1

This is function\_call statement::: factorial(n-1)

This is a factor statemnt::: factorial(n-1)

This is a term statemnt::: n\*factorial(n-1)

This is a expression statemnt::: n\*factorial(n-1)

This is a factor statemnt::: (n\*factorial(n-1))

This is a term statemnt::: (n\*factorial(n-1))

This is a expression statemnt::: (n\*factorial(n-1))

This is an assign statement::: res = (n\*factorial(n-1))

This is a default statement::: default: res = (n\*factorial(n-1));

This is a switch statement::: switch (n){

case 0: res = 1;

case 1: res = 1;

default: res = (n\*factorial(n-1));

}

This is a factor statemnt::: res

This is a term statemnt::: res

This is a expression statemnt::: res

This is a return statement::: return (res)

siz = 2519 now = 556

This is a function\_int statement C ::: 2 parameters::: int C(int m, int n)

This is a factor statemnt::: m

This is a term statemnt::: m

This is a expression statemnt::: m

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a condition ::: m <= n

This is a ifstatement::: if (m <= n) ;

This is a factor statemnt::: m

This is a term statemnt::: m

This is a expression statemnt::: m

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a condition ::: m > n

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: -1

This is a return statement::: return (-1)

This is a ifstatement::: if (m > n)

return (-1);

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a parameter with ::: 1 parameters::: n

This is function\_call statement::: factorial(n)

This is a factor statemnt::: factorial(n)

This is a factor statemnt::: m

This is a term statemnt::: m

This is a expression statemnt::: m

This is a parameter with ::: 1 parameters::: m

This is function\_call statement::: factorial(m)

This is a factor statemnt::: factorial(m)

This is a factor statemnt::: n

This is a term statemnt::: n

This is a factor statemnt::: m

This is a term statemnt::: m

This is a expression statemnt::: n-m

This is a parameter with ::: 1 parameters::: n-m

This is function\_call statement::: factorial(n-m)

This is a factor statemnt::: factorial(n-m)

This is a term statemnt::: factorial(n)/factorial(m)/factorial(n-m)

This is a expression statemnt::: factorial(n)/factorial(m)/factorial(n-m)

This is a return statement::: return (factorial(n)/factorial(m)/factorial(n-m))

siz = 2519 now = 695

This is a void statement init ::: 0 parameters::: void init()

This is a factor statemnt::: 5

This is a term statemnt::: 5

This is a expression statemnt::: 5

This is an assign statement::: m = 5

This is a scanf statement::: 4 variables::: scanf(n,m,k, ch)

This is a return statement::: return

siz = 2519 now = 756

This is a function\_char statement work ::: 2 parameters::: char work(int m, int n)

This is a variable\_int statement::: 1 variables::: int ans

This is a factor statemnt::: m

This is a term statemnt::: m

This is a expression statemnt::: m

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a parameter with ::: 2 parameters::: m,n

This is function\_call statement::: C(m,n)

This is a factor statemnt::: C(m,n)

This is a term statemnt::: C(m,n)

This is a expression statemnt::: C(m,n)

This is an assign statement::: ans = C(m,n)

This is a factor statemnt::: ans

This is a term statemnt::: ans

This is a factor statemnt::: 2

This is a factor statemnt::: ans

This is a factor statemnt::: 2

This is a term statemnt::: ans / 2

This is a expression statemnt::: ans / 2

This is a factor statemnt::: (ans / 2)

This is a term statemnt::: 2\*(ans / 2)

This is a expression statemnt::: ans - 2\*(ans / 2)

This is a factor statemnt::: 'O'

This is a term statemnt::: 'O'

This is a expression statemnt::: 'O'

This is a return statement::: return ('O')

This is cases statement with::: 1 cases::: case 1: return ('O');

This is a factor statemnt::: 'E'

This is a term statemnt::: 'E'

This is a expression statemnt::: 'E'

This is a return statement::: return ('E')

This is a default statement::: default: return ('E');

This is a switch statement::: switch (ans - 2\*(ans / 2)){

case 1: return ('O');

default: return ('E');

}

This is a factor statemnt::: '\*'

This is a term statemnt::: '\*'

This is a expression statemnt::: '\*'

This is a return statement::: return ('\*')

siz = 2519 now = 935

This is a void statement test\_switch ::: 0 parameters::: void test\_switch()

This is a variable\_int statement::: 2 variables::: int Int, a[1000]

This is a factor statemnt::: 2

This is a term statemnt::: 2

This is a expression statemnt::: 2

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: -1

This is an assign statement::: a[2] = -1

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 1

This is a factor statemnt::: 2

This is a term statemnt::: 2

This is a expression statemnt::: -2

This is an assign statement::: a[1] = -2

This is a factor statemnt::: 2

This is a term statemnt::: 2

This is a factor statemnt::: 2

This is a factor statemnt::: 5

This is a factor statemnt::: 4

This is a term statemnt::: 2\*5/4

This is a expression statemnt::: 2\*5/4

This is a factor statemnt::: a[2\*5/4]

This is a term statemnt::: a[2\*5/4]

This is a expression statemnt::: 2+a[2\*5/4]

This is a factor statemnt::: a[2+a[2\*5/4]]

This is a term statemnt::: a[2+a[2\*5/4]]

This is a expression statemnt::: a[2+a[2\*5/4]]

This is a factor statemnt::: a[a[2+a[2\*5/4]]]

This is a term statemnt::: a[a[2+a[2\*5/4]]]

This is a expression statemnt::: a[a[2+a[2\*5/4]]]

This is an assign statement::: Int = a[a[2+a[2\*5/4]]]

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a expression statemnt::: Int

This is a printf statement::: printf("666666")

This is a printf statement::: printf("HHHHH")

This is cases statement with::: 2 cases::: case -1: printf("666666");

case -2: {printf("HHHHH"); }

This is a default statement::: default: ;

This is a switch statement::: switch (Int){

case -1: printf("666666");

case -2: {printf("HHHHH"); }

default: ;

}

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a factor statemnt::: 'b'

This is a term statemnt::: 'b'

This is a expression statemnt::: Int+'b'

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: Int+1

This is an assign statement::: Int = Int+1

This is a printf statement::: printf("23333")

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a expression statemnt::: Int

This is a printf statement::: printf("\*\*\*\*\*",Int)

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a expression statemnt::: Int

This is a printf statement::: printf(Int)

This is cases statement with::: 3 cases::: case 1: Int = Int+1;

case 'a': {printf("23333"); }

case -123: {printf("\*\*\*\*\*",Int); printf(Int); }

This is a scanf statement::: 3 variables::: scanf(Int,Int,Int)

This is a default statement::: default: {scanf(Int,Int,Int);}

This is a switch statement::: switch (Int+'b'){

case 1: Int = Int+1;

case 'a': {printf("23333"); }

case -123: {printf("\*\*\*\*\*",Int); printf(Int); }

default: {scanf(Int,Int,Int);}

}

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 1

This is a factor statemnt::: a[1]

This is a term statemnt::: a[1]

This is a expression statemnt::: 1-a[1]

This is a factor statemnt::: a[1-a[1]]

This is a term statemnt::: a[1-a[1]]

This is a expression statemnt::: a[1-a[1]]

This is a factor statemnt::: a[a[1-a[1]]]

This is a term statemnt::: a[a[1-a[1]]]

This is a expression statemnt::: a[a[1-a[1]]]

This is an assign statement::: Int = a[a[1-a[1]]]

siz = 2519 now = 1370

This is a void statement test\_for ::: 1 parameters::: void test\_for(int Int)

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a factor statemnt::: -2

This is a term statemnt::: -2

This is a expression statemnt::: 0+-2

This is an assign statement::: Int = 0+-2

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: Int +1

This is an assign statement::: Int = Int +1

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: Int +1

This is an assign statement::: Int = Int +1

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a expression statemnt::: Int

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: 1000

This is a term statemnt::: 1000

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a expression statemnt::: 1000-Int

This is a condition ::: i <= 1000-Int

This is a factor statemnt::: Int

This is a term statemnt::: Int

This is a expression statemnt::: Int

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a condition ::: i <= i

This is a loop statement::: for (i = Int; i <= i; i = i+1){

;

}

This is a loop statement::: for (i = Int; i <= 1000-Int; i = i+1){

for (i = Int; i <= i; i = i+1){

;

}

}

siz = 2519 now = 1641

This is a void statement test\_call ::: 0 parameters::: void test\_call()

This is a factor statemnt::: 2

This is a term statemnt::: 2

This is a expression statemnt::: 2

This is a factor statemnt::: 2

This is a term statemnt::: 2

This is a expression statemnt::: 2

This is a factor statemnt::: 3

This is a factor statemnt::: 5

This is a term statemnt::: 3\*5

This is a expression statemnt::: 3\*5

This is a parameter with ::: 2 parameters::: 2,3\*5

This is function\_call statement::: mymax(2,3\*5)

This is a factor statemnt::: mymax(2,3\*5)

This is a factor statemnt::: 2

This is a term statemnt::: 2

This is a expression statemnt::: 2

This is a factor statemnt::: 4

This is a term statemnt::: 4

This is a expression statemnt::: 4

This is a parameter with ::: 2 parameters::: 2, 4

This is function\_call statement::: mymax(2, 4)

This is a factor statemnt::: mymax(2, 4)

This is a factor statemnt::: 100

This is a term statemnt::: 100

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 100-1

This is a factor statemnt::: (100-1)

This is a term statemnt::: mymax(2,3\*5)/mymax(2, 4)\*(100-1)

This is a expression statemnt::: mymax(2,3\*5)/mymax(2, 4)\*(100-1)

This is a parameter with ::: 2 parameters::: 2, mymax(2,3\*5)/mymax(2, 4)\*(100-1)

This is function\_call statement::: mymax(2, mymax(2,3\*5)/mymax(2, 4)\*(100-1))

siz = 2519 now = 1709

This is a factor statemnt::: mx

This is a term statemnt::: mx

This is a expression statemnt::: mx

This is a printf statement::: printf("mx = ", mx)

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a expression statemnt::: 0

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: m

This is a term statemnt::: m

This is a expression statemnt::: m

This is a condition ::: i <= m

This is a factor statemnt::: m

This is a term statemnt::: m

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: m - i

This is an assign statement::: j = m - i

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a expression statemnt::: 0

This is a condition ::: i == 0

This is a factor statemnt::: j

This is a term statemnt::: j

This is a expression statemnt::: j

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a parameter with ::: 2 parameters::: j,n

This is function\_call statement::: work(j,n)

This is a factor statemnt::: work(j,n)

This is a term statemnt::: work(j,n)

This is a expression statemnt::: work(j,n)

This is a printf statement::: printf(work(j,n))

This is a ifstatement::: if (i == 0){

printf(work(j,n));

}

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a printf statement::: printf("N ",n)

This is a factor statemnt::: j

This is a term statemnt::: j

This is a expression statemnt::: j

This is a printf statement::: printf("M ",j)

This is a factor statemnt::: j

This is a term statemnt::: j

This is a expression statemnt::: j

This is a factor statemnt::: n

This is a term statemnt::: n

This is a expression statemnt::: n

This is a parameter with ::: 2 parameters::: j, n

This is function\_call statement::: C(j, n)

This is a factor statemnt::: C(j, n)

This is a term statemnt::: C(j, n)

This is a expression statemnt::: C(j, n)

This is a printf statement::: printf("C ",C(j, n))

This is a printf statement::: printf("")

This is a loop statement::: for (i = 0; i <= m; i = i +1) {

j = m - i;

if (i == 0){

printf(work(j,n));

}

printf("N ",n);

printf("M ",j);

printf("C ",C(j, n));

printf("");

}

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 1

This is a factor statemnt::: -2

This is a term statemnt::: -2

This is a expression statemnt::: --2

This is a parameter with ::: 2 parameters::: 1, --2

This is function\_call statement::: mymax(1, --2)

This is a factor statemnt::: mymax(1, --2)

This is a term statemnt::: mymax(1, --2)

This is a expression statemnt::: mymax(1, --2)

This is an assign statement::: x = mymax(1, --2)

This is a factor statemnt::: x

This is a term statemnt::: x

This is a factor statemnt::: +8

This is a term statemnt::: +8

This is a expression statemnt::: x++8

This is a factor statemnt::: 9

This is a term statemnt::: 9

This is a expression statemnt::: 9

This is a parameter with ::: 2 parameters::: x++8, 9

This is function\_call statement::: mymax(x++8, 9)

This is a factor statemnt::: mymax(x++8, 9)

This is a term statemnt::: mymax(x++8, 9)

This is a expression statemnt::: mymax(x++8, 9)

This is an assign statement::: x = mymax(x++8, 9)

This is a factor statemnt::: 4

This is a term statemnt::: 4

This is a expression statemnt::: 4

This is an assign statement::: y = 4

This is a factor statemnt::: 7

This is a term statemnt::: 7

This is a expression statemnt::: 7

This is an assign statement::: z = 7

This is a factor statemnt::: x

This is a term statemnt::: x

This is a expression statemnt::: x

This is a factor statemnt::: y

This is a term statemnt::: y

This is a expression statemnt::: y

This is a condition ::: x > y

This is a factor statemnt::: x

This is a term statemnt::: x

This is a expression statemnt::: x

This is a factor statemnt::: z

This is a term statemnt::: z

This is a expression statemnt::: z

This is a condition ::: x > z

This is a factor statemnt::: x

This is a term statemnt::: x

This is a expression statemnt::: x

This is an assign statement::: mx = x

This is a factor statemnt::: z

This is a term statemnt::: z

This is a expression statemnt::: z

This is an assign statement::: mx = z

This is a if\_elsestatement::: if (x > z)

mx = x;

else

mx = z;

This is a factor statemnt::: y

This is a term statemnt::: y

This is a expression statemnt::: y

This is a factor statemnt::: z

This is a term statemnt::: z

This is a expression statemnt::: z

This is a condition ::: y > z

This is a factor statemnt::: y

This is a term statemnt::: y

This is a expression statemnt::: y

This is an assign statement::: mx = y

This is a factor statemnt::: z

This is a term statemnt::: z

This is a expression statemnt::: z

This is an assign statement::: mx = z

This is a if\_elsestatement::: if (y > z)

mx = y;

else mx = z;

This is a if\_elsestatement::: if (x > y){

if (x > z)

mx = x;

else

mx = z;

}

else if (y > z)

mx = y;

else mx = z;

This is a factor statemnt::: mx

This is a term statemnt::: mx

This is a expression statemnt::: mx

This is a printf statement::: printf("mx = ", mx)

This is function\_call statement::: init()

This is a factor statemnt::: k

This is a term statemnt::: k

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: k+1

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a expression statemnt::: 0

This is a condition ::: i >= 0

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: k

This is a term statemnt::: k

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: k+i

This is an assign statement::: a[i] = k+i

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: ch

This is a term statemnt::: ch

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: ch + i

This is an assign statement::: s[i] = ch + i

This is a loop statement::: for (i = k+1; i >= 0; i = i - 1){

a[i] = k+i;

s[i] = ch + i;

}

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a expression statemnt::: 0

This is a factor statemnt::: i

This is a term statemnt::: i

This is a expression statemnt::: i

This is a factor statemnt::: k

This is a term statemnt::: k

This is a expression statemnt::: k

This is a condition ::: i > k

This is a factor statemnt::: k

This is a term statemnt::: k

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: k + 1

This is an assign statement::: k = k + 1

This is a loop statement::: for (i = 0; i > k; i = i-1) {

k = k + 1;

}

This is a factor statemnt::: k

This is a term statemnt::: k

This is a expression statemnt::: k

This is a printf statement::: printf("k = ", k)

This is a factor statemnt::: k

This is a term statemnt::: k

This is a expression statemnt::: k

This is a factor statemnt::: s[k]

This is a term statemnt::: s[k]

This is a expression statemnt::: s[k]

This is a printf statement::: printf("ch = ",s[k])

This is function\_call statement::: test\_switch()

This is a factor statemnt::: 1

This is a term statemnt::: 1

This is a expression statemnt::: 1

This is a parameter with ::: 1 parameters::: 1

This is function\_call statement::: test\_for(1)

This is function\_call statement::: test\_call()

This is a factor statemnt::: 0

This is a term statemnt::: 0

This is a expression statemnt::: 0

This is a return statement::: return (0)

siz = 2519 now = 2519

Program ended with exit code: 0