```
==> <function><otherFunction> <mainFunction>
==> FUNCTION <type list> LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
         <mainFunction>
==> function <type list> LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
         <mainFunction>
==> function <type list> LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function <type> <type more> LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function <primitive type> <type more> LPARA <parameter list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function INT <type more> LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int <type more> LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int COMMA <type> <type more> LPARA <parameter list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, <type> <type more> LPARA <parameter list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, <primitive type> <type more> LPARA <parameter list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, INT <type more> LPARA <parameter list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int <type more> LPARA <parameter list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int ID LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int, int func1 LPARA <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int, int func1 ( <parameter list> LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int, int func1 ( <type> ID <parameter list more> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( <primitive type> ID <parameter list more> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( INT ID <parameter list more> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int ID <parameter list more> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a <parameter list more> LBRACE
<fn stmts><otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a ASSIGN OP <const value> <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = <const value> <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = NUM <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1 <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1 COMMA <type> ID ASSIGN OP <const value>
<default arg list> LBRACE <fn stmts><otherFunction> <mainFunction>
==> function int, int func1 ( int a = 1, <type> ID ASSIGN OP <const value>
<default arg list> LBRACE <fn stmts><otherFunction> <mainFunction>
==> function int, int func1 ( int a = 1, <primitive type> ID ASSIGN OP <const value>
<default arg list> LBRACE <fn stmts><otherFunction> <mainFunction>
==> function int, int func1 ( int a = 1, INT ID ASSIGN OP <const value>
<default arg list> LBRACE <fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int ID ASSIGN OP <const value>
<default arg list> LBRACE <fn stmts><otherFunction> <mainFunction>
==> function int, int func1 ( int a = 1, int b ASSIGN OP <const value> <default arg list>
LBRACE <fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = <const value> <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = NUM <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 <default arg list> LBRACE
<fn stmts><otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 RPARA LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) LBRACE <fn stmts><otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { <fn stmts><otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { <function><fn stmts>
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { FUNCTION <type list> LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function <type list> LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 (int a = 1, int b = 3) { function <type> <type more> LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function <primitive type>
<type more> LPARA <parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function INT <type more> LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int <type more> LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int ID LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 LPARA
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 (
<parameter list> LBRACE <fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 ( RPARA LBRACE
<fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 ( ) LBRACE
<fn stmts> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { <fn stmts>
<fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { RETURN
<expressions> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<expressions> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<expression><more expression> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<and expr><expression'><more expression> SEMI CL RBRACE <fn_stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<reln expr><and expr'><expression'><more expression> SEMI CL RBRACE <fn stmts>
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<add sub expr><reln expr'><and expr'><expression'><more expression> SEMI CL RBRACE
<fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<mul div expr><add sub expr'><reln expr'><and expr'><expression'><more expression>
SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<un expr><mul div expr'><add sub expr'><reln expr'><and expr'><expression'><more expressio</pre>
n> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<typecast><mul div expr'><add sub expr'><reln expr'><and expr'><expression'><more expressi
on> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<value><mul div expr'><add sub expr'><reln expr'><and expr'><expression'><more expression>
SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
<const value><mul div expr'><add sub expr'><reln expr'><and expr'><expression'><more expre</pre>
ssion> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
NUM<mul div expr'><add sub expr'><reln expr'><and expr'><expression'><more expression>
SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3<add sub expr'><reln expr'><and expr'><expression'><more expression> SEMI CL RBRACE
<fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3<reln expr'><and expr'><expression'><more expression> SEMI CL RBRACE <fn stmts>
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3<and expr'><expression'><more expression> SEMI CL RBRACE <fn stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3<expression'><more expression> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3<more expression> SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3
SEMI CL RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
RBRACE <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } <stmt><fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } <gen stmt><fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } ID <func ass decl><fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a <func ass decl><fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a <array> <leftHandSide1><fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a <leftHandSide1><fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a<assign operators> <rightHandSide> <fn stmts> <otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a ASSIGN OP <rightHandSide> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <rightHandSide> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <expression> <rightHandSideMultiple> <fn stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <and expr><expression'><rightHandSideMultiple> <fn stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <reln expr><and expr'><expression'><rightHandSideMultiple> <fn stmts>
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <add sub expr><reln expr'><and expr'><expression'><rightHandSideMultiple>
<fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <mul div expr> <add sub expr'> <reln expr'> <and expr'> <expression'>
<rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <un expr> <mul div expr'> <add sub expr'> <reln expr'> <and expr'>
<expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <typecast> <mul div expr'> <add sub expr'> <reln expr'> <and expr'>
<expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = <value> <mul div expr'> <add sub expr'> <reln expr'> <and expr'> <expression'>
<rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = ID <value rest> <mul div expr'> <add sub expr'> <reln expr'> <and expr'>
<expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 <value rest> <mul div expr'> <add sub expr'> <reln expr'> <and expr'>
<expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 <funcCall inside Value> <mul div expr'> <add sub expr'> <reln expr'>
<and expr'> <expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 LPARA <funcCall inside Value rest> <mul div expr'> <add sub expr'>
<reln expr'> <and expr'> <expression'> <rightHandSideMultiple> <fn stmts>
<otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (<funcCall inside Value rest> <mul div expr'> <add sub expr'>
<reln expr'> <and expr'> <expression'> <rightHandSideMultiple> <fn stmts>
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (RPARA <mul div expr'> <add sub expr'> <reln expr'> <and expr'>
<expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () <mul div expr'> <add sub expr'> <reln expr'> <and expr'>
<expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () <add sub expr'> <reln expr'> <and expr'> <expression'>
<rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () <reln expr'> <and expr'> <expression'> <rightHandSideMultiple>
<fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () <and expr'> <expression'> <rightHandSideMultiple> <fn stmts>
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () <expression'> <rightHandSideMultiple> <fn stmts> <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () <rightHandSideMultiple> <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 () SEMI CL<fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); <fn stmts> <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); RETURN <expressions> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return <expressions> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return <expression><more expression>SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return <and expr><expression'><more expression>SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return <reln expr><and expr'><expression'><more expression>SEMI CL
RBRACE <otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return <add sub expr> <reln expr'> <and expr'> <expression'>
<more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return <mul div expr><add sub expr'> <reln expr'> <and expr'>
<expression' > <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return <un expr> <mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return <typecast> <mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return <typecast> <mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return <value><mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return ID <value rest>mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return b<value rest>mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return b<value rest>mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b<array>mul div expr'><add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b <mul div expr'><add sub expr'> <reln expr'> <and expr'>
<expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b <add sub expr'> <reln expr'> <and expr'> <expression'>
<more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b <reln expr'> <and expr'> <expression'> <more expression>
SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b <and expr'> <expression'> <more expression> SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b <expression'> <more expression> SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b COMMA <expression><more expression> SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b, <expression > < more expression > SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b, <and expr><expression'><more expression> SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b, <reln expr><and expr'><expression'><more expression>
SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b, <add sub expr> <reln expr'> <and expr'> <expression'>
<more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b,<mul div expr><add sub expr'><reln expr'> <and expr'>
<expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 ( ); return b, <un expr> <mul div expr'> <add sub expr'> <reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return
3; } a = func2 (); return b,<typecast> <mul div expr'><add sub expr'><reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b, <value><mul div expr'><add sub expr'><reln expr'> <and expr'>
<expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b, ID <value rest><mul div expr'><add sub expr'><reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a <value rest><mul div expr'><add sub expr'><reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 ( ); return b,a <array> <mul div expr'> <add sub expr'><reln expr'>
<and expr'> <expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a<mul div expr'> <add sub expr'><reln expr'> <and expr'>
<expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a<mul div expr'> <add sub expr'><reln expr'> <and expr'>
<expression'> <more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a<add sub expr'><reln expr'> <and expr'> <expression'>
<more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a<reln expr'> <and expr'> <expression'> <more expression> SEMI CL
RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a<and expr'> <expression'> <more expression> SEMI CL RBRACE
<otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a<expression'> <more expression> SEMI CL RBRACE <otherFunctions>
<mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a<more expression> SEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,aSEMI CL RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; RBRACE <otherFunctions> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a; }<otherFunction> <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }MAIN LPARA RPARA LBRACE <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a; }mainLPARA RPARA LBRACE <mainFunction>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main(RPARA LBRACE <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main()LBRACE <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { <mainFunction>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { <stmt><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { <gen stmt>stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { <other stmt><smts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { <primitive declaration stmt><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( );  return b,a;  }main(){  <primitive type> ID <assign more><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int ID<assign more><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x<assign more><<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x<assign comma><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int xCOMMA ID <assign more> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x,ID<assign more> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main(){ int x,y<assign more> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 (); return b,a; }main() { int x,ySEMI CL<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x,y; <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x,y;<stmt><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x,y;<gen stmt><stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main() { int x,y;ID <func ass decl> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; }
a = func2 ( ); return b,a; }main(){ int x,y;x <func ass decl> <stmts>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y;x <array> <lefthandSide1>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y;x <lefthandSide1> <stmts>
==> function int, int func1 (int a = 1, int b = 3) { function int func2 () { return 3;
} a = func2 ( ); return b,a; }main() { int x,y;x COMMA <var> <lefthandSideMultiple>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, <var> <lefthandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, ID <array> <lefthandSideMultiple>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y <array> <lefthandSideMultiple>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 ( ); return b,a; }main() { int x,y; x, y <lefthandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; } main() { int x,y; x, y ASSIGN_OP < rightHandSide> < stmts>}
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; main() int x,y; x, y = <rightHandSide> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
a = \text{func2}(); \text{ return b,a; } \text{main}() \{ \text{ int } x,y; x, y = < \text{expression} > \text{main}() \}
<rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 ( ); return b,a; main() \{ int x, y; x, y = \langle and expr \rangle \langle expression' \rangle \}
<rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y = <reln expr><and expr'><expression'>
<rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; main() { int x,y; x, y = <add sub expr><reln expr'>
<and expr'><expression'><rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; } main() { int x,y; x, y = }
<mul div expr><add sub expr'><reln expr'> <and expr'><expression'><rightHandSideMultiple>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; } main() { int x,y; x, y = }
<mul div expr><add sub expr'><reln expr'> <and expr'><expression'><rightHandSideMultiple>
<stmts>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y =
<mul div expr><add sub expr'><reln expr'> <and expr'><expression'><rightHandSideMultiple>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; } main() { int x,y; x, y = }
<mul div expr'><add sub expr'><reln expr'>
<and expr'><expression'><rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y = <add sub expr'><reln expr'>
<and expr'><expression'><rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y = < reln expr'>}
<and expr'><expression'><rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 ( ); return b,a; }main() { int x,y; x, y =
<and expr'><expression'><rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
a = func2 (); return b,a; }main() { int x,y; x, y = }
<expression'><rightHandSideMultiple> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; } main() { int x,y; x, y = < rightHandSideMultiple> < stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; main() { int x,y; x, y = ID < rightHandSideMultiple> < stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y = func1 <rightHandSideMultiple>
<stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main(){ int x,y; x, y = func1 LPAREN<arg list> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; main() { int x,y; x, y = func1 (<arg list> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > < stmts > } main() { int x,y; x, y = func1 (ID < rightSide > stmts > ) { int x,y; x, y = func1 (ID < rightSide > stmts > ) { int x,y; x
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main(){ int x,y; x, y = func1 (b < rightSide > < stmts > }
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y = func1 (b EQ OP <value> <stmts>
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
a = \text{func2}(); \text{ return b,a; } \text{main}() \{ \text{ int } x,y; x, y = \text{func1 } (b = \text{value}) < \text{stmts} \}
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3;
} a = func2 (); return b,a; }main() { int x,y; x, y = func1 (b = <num> RPAREN SEMI CL
<stmts>
```

```
==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5 RPAREN SEMI_CL <stmts> ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5) SEMI_CL <stmts> ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5); <stmts> ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5); <stmts> ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5); <stmt>RCURLY ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5); RCURLY ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5); RCURLY ==> function int, int func1 ( int a = 1, int b = 3 ) { function int func2 () { return 3; } a = func2 ( ); return b,a; }main() { int x,y; x, y = func1 (b = 5); }
```