

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

==> 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 = 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 = <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 = <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>

```

[illegible]

```

==> 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);}

```