

# 测试结果展示

## 测试样例1

▼ Plain Text |

```
1 void main(){
2     return 2;
3 }
```

## 词法分析结果

▼ lex1.txt Shell |

```
1 void <KW,4>
2 main <IDN,main>
3 ( <SE,23>
4 ) <SE,24>
5 { <SE,25>
6     return <KW,5>
7     2 <INT,2>
8     ; <SE,27>
9     } <SE,26>
10
```

## 语法分析结果

```
1  0  program#void  reduction
2  1  compUnit#void reduction
3  2  funcDef#void  reduction
4  3  funcType#void reduction
5  4  void#void  move
6  5  Ident#Ident  move
7  6  (#(      move
8  7  funcFParams#) reduction
9  8  )#)      move
10 9  block#{      reduction
11 10 {#{      move
12 11 blockItem#return reduction
13 12 stmt#return  reduction
14 13 return#return move
15 14 argExp#INT  reduction
16 15 exp#INT      reduction
17 16 assignExp#INT reduction
18 17 INT#INT      move
19 18 mulExpAtom#;  reduction
20 19 addExpAtom#;  reduction
21 20 relExpAtom#;  reduction
22 21 eqExpAtom#;   reduction
23 22 assignExpAtom#; reduction
24 23 ;#;          move
25 24 blockItem#}   reduction
26 25 }#}          move
27 26 compUnit##  reduction
28 27 ###          accept
```

## 测试样例2

```
1  int b = 3;
2
3  void main(){
4      int a = 4;
5      return a + b;
6  }
```

## 词法分析结果

```
lex2.txt Shell |
1  int <KW,1>
2  b <IDN,b>
3  = <OP,12>
4  3 <INT,3>
5  ; <SE,27>
6  void <KW,4>
7  main <IDN,main>
8  ( <SE,23>
9  ) <SE,24>
10 { <SE,25>
11  int <KW,1>
12  a <IDN,a>
13  = <OP,12>
14  4 <INT,4>
15  ; <SE,27>
16  return <KW,5>
17  a <IDN,a>
18  + <OP,9>
19  b <IDN,b>
20  ; <SE,27>
21  } <SE,26>
22
```

## 语法分析结果

```
1  0  program#int      reduction
2  1  compUnit#int     reduction
3  2  decl#int         reduction
4  3  varDecl#int      reduction
5  4  bType#int        reduction
6  5  int#int          move
7  6  varDef#Ident     reduction
8  7  Ident#Ident      move
9  8  argVarDef#       reduction
10 9  =#               move
11 10 initVal#INT       reduction
12 11 exp#INT           reduction
13 12 assignExp#INT     reduction
14 13 INT#INT           move
15 14 mulExpAtom#;      reduction
16 15 addExpAtom#;      reduction
17 16 relExpAtom#;      reduction
18 17 eqExpAtom#;       reduction
19 18 assignExpAtom#;    reduction
20 19 argVarDecl#;      reduction
21 20 ;#;               move
22 21 compUnit#void      reduction
23 22 funcDef#void       reduction
24 23 funcType#void      reduction
25 24 void#void          move
26 25 Ident#Ident        move
27 26 (#(               move
28 27 funcFParams#)      reduction
29 28 )#)               move
30 29 block#{             reduction
31 30 {#{               move
32 31 blockItem#int       reduction
33 32 int#int             move
34 33 varDef#Ident        reduction
35 34 Ident#Ident         move
36 35 argVarDef#         reduction
37 36 =#                 move
38 37 initVal#INT         reduction
39 38 exp#INT             reduction
40 39 assignExp#INT        reduction
41 40 INT#INT             move
42 41 mulExpAtom#;        reduction
43 42 addExpAtom#;        reduction
44 43 relExpAtom#;        reduction
45 44 eqExpAtom#;         reduction
```

```

46 45 assignExpAtom#;    reduction
47 46 argVarDecl#;    reduction
48 47 ;#;    move
49 48 blockItem#return  reduction
50 49 stmt#return    reduction
51 50 return#return  move
52 51 argExp#Ident    reduction
53 52 exp#Ident    reduction
54 53 assignExp#Ident    reduction
55 54 Ident#Ident    move
56 55 callFunc#+    reduction
57 56 mulExpAtom#+    reduction
58 57 addExpAtom#+    reduction
59 58 ++#    move
60 59 mulExp#Ident    reduction
61 60 Ident#Ident    move
62 61 callFunc#;    reduction
63 62 mulExpAtom#;    reduction
64 63 addExpAtom#;    reduction
65 64 relExpAtom#;    reduction
66 65 eqExpAtom#;    reduction
67 66 assignExpAtom#;    reduction
68 67 ;#;    move
69 68 blockItem#}    reduction
70 69 }#}    move
71 70 compUnit##    reduction
72 71 ###    accept

```

## 测试样例3

▼ Plain Text

```

1 void main(){
2     int a, b0, _c;
3     a = 1;
4     b0 = 2;
5     _c = 3;
6     return b0 + _c;
7 }

```

## 词法分析结果

```
1 void <KW,4>
2 main <IDN,main>
3 ( <SE,23>
4 ) <SE,24>
5 { <SE,25>
6   int <KW,1>
7   a <IDN,a>
8   , <SE,28>
9   b0 <IDN,b0>
10  , <SE,28>
11  _c <IDN,_c>
12  ; <SE,27>
13  a <IDN,a>
14  = <OP,12>
15  1 <INT,1>
16  ; <SE,27>
17  b0 <IDN,b0>
18  = <OP,12>
19  2 <INT,2>
20  ; <SE,27>
21  _c <IDN,_c>
22  = <OP,12>
23  3 <INT,3>
24  ; <SE,27>
25  return <KW,5>
26  b0 <IDN,b0>
27  + <OP,9>
28  _c <IDN,_c>
29  ; <SE,27>
30  } <SE,26>
31
```

## 语法分析结果

```

1  0  program#void    reduction
2  1  compUnit#void  reduction
3  2  funcDef#void   reduction
4  3  funcType#void  reduction
5  4  void#void      move
6  5  Ident#Ident    move
7  6  (#(          move
8  7  funcFParams#) reduction
9  8  )#)          move
10 9  block#{        reduction
11 10 {#{          move
12 11 blockItem#int  reduction
13 12 int#int        move
14 13 varDef#Ident   reduction
15 14 Ident#Ident    move
16 15 argVarDef#,     reduction
17 16 argVarDecl#,    reduction
18 17 ,#,           move
19 18 varDef#Ident    reduction
20 19 Ident#Ident     move
21 20 argVarDef#,     reduction
22 21 argVarDecl#,    reduction
23 22 ,#,           move
24 23 varDef#Ident    reduction
25 24 Ident#Ident     move
26 25 argVarDef#;     reduction
27 26 argVarDecl#;    reduction
28 27 ;#;           move
29 28 blockItem#Ident reduction
30 29 stmt#Ident       reduction
31 30 exp#Ident        reduction
32 31 assignExp#Ident  reduction
33 32 Ident#Ident      move
34 33 callFunc#=       reduction
35 34 mulExpAtom#=     reduction
36 35 addExpAtom#=     reduction
37 36 relExpAtom#=     reduction
38 37 eqExpAtom#=      reduction
39 38 assignExpAtom#=  reduction
40 39 =#=             move
41 40 eqExp#INT        reduction
42 41 INT#INT          move
43 42 mulExpAtom#;     reduction
44 43 addExpAtom#;     reduction
45 44 relExpAtom#;     reduction

```

```

46 45 eqExpAtom#;    reduction
47 46 assignExpAtom#;    reduction
48 47 ;#;    move
49 48 blockItem#Ident    reduction
50 49 stmt#Ident reduction
51 50 exp#Ident  reduction
52 51 assignExp#Ident    reduction
53 52 Ident#Ident    move
54 53 callFunc#= reduction
55 54 mulExpAtom#=    reduction
56 55 addExpAtom#=    reduction
57 56 relExpAtom#=    reduction
58 57 eqExpAtom#=    reduction
59 58 assignExpAtom#=    reduction
60 59 =#=    move
61 60 eqExp#INT  reduction
62 61 INT#INT    move
63 62 mulExpAtom#;    reduction
64 63 addExpAtom#;    reduction
65 64 relExpAtom#;    reduction
66 65 eqExpAtom#;    reduction
67 66 assignExpAtom#;    reduction
68 67 ;#;    move
69 68 blockItem#Ident    reduction
70 69 stmt#Ident reduction
71 70 exp#Ident  reduction
72 71 assignExp#Ident    reduction
73 72 Ident#Ident    move
74 73 callFunc#= reduction
75 74 mulExpAtom#=    reduction
76 75 addExpAtom#=    reduction
77 76 relExpAtom#=    reduction
78 77 eqExpAtom#=    reduction
79 78 assignExpAtom#=    reduction
80 79 =#=    move
81 80 eqExp#INT  reduction
82 81 INT#INT    move
83 82 mulExpAtom#;    reduction
84 83 addExpAtom#;    reduction
85 84 relExpAtom#;    reduction
86 85 eqExpAtom#;    reduction
87 86 assignExpAtom#;    reduction
88 87 ;#;    move
89 88 blockItem#return    reduction
90 89 stmt#return    reduction
91 90 return#return    move
92 91 argExp#Ident    reduction
93 92 exp#Ident  reduction

```



```

94 93 assignExp#Ident    reduction
95 94 Ident#Ident      move
96 95 callFunc#+      reduction
97 96 mulExpAtom#+     reduction
98 97 addExpAtom#+     reduction
99 98 ++              move
100 99 mulExp#Ident     reduction
101 100 Ident#Ident     move
102 101 callFunc#;      reduction
103 102 mulExpAtom#;    reduction
104 103 addExpAtom#;    reduction
105 104 relExpAtom#;    reduction
106 105 eqExpAtom#;     reduction
107 106 assignExpAtom#;  reduction
108 107 ;#;             move
109 108 blockItem#}      reduction
110 109 }#}             move
111 110 compUnit##        reduction
112 111 ###              accept

```

## 测试样例4

▼ Plain Text

```

1 void main(){
2     const int a = 10, b = 5;
3     return b;
4 }

```

## 词法分析结果

```
1 void <KW,4>
2 main <IDN,main>
3 ( <SE,23>
4 ) <SE,24>
5 { <SE,25>
6   const <KW,6>
7   int <KW,1>
8   a <IDN,a>
9   = <OP,12>
10  10 <INT,10>
11  , <SE,28>
12  b <IDN,b>
13  = <OP,12>
14  5 <INT,5>
15  ; <SE,27>
16  return <KW,5>
17  b <IDN,b>
18  ; <SE,27>
19  } <SE,26>
20
```

## 语法分析结果

```
1  0  program#void  reduction
2  1  compUnit#void reduction
3  2  funcDef#void  reduction
4  3  funcType#void reduction
5  4  void#void  move
6  5  Ident#Ident  move
7  6  (#(      move
8  7  funcFParams#) reduction
9  8  )#)      move
10 9  block#{      reduction
11 10 {#{      move
12 11 blockItem#const  reduction
13 12 const#const  move
14 13 bType#int  reduction
15 14 int#int  move
16 15 constDef#Ident reduction
17 16 Ident#Ident  move
18 17 =#=      move
19 18 constInitVal#INT reduction
20 19 constExp#INT  reduction
21 20 INT#INT  move
22 21 mulExpAtom#,  reduction
23 22 addExpAtom#,  reduction
24 23 relExpAtom#,  reduction
25 24 eqExpAtom#,  reduction
26 25 assignExpAtom#, reduction
27 26 argConst#, reduction
28 27 ,#,      move
29 28 constDef#Ident reduction
30 29 Ident#Ident  move
31 30 =#=      move
32 31 constInitVal#INT reduction
33 32 constExp#INT  reduction
34 33 INT#INT  move
35 34 mulExpAtom#;  reduction
36 35 addExpAtom#;  reduction
37 36 relExpAtom#;  reduction
38 37 eqExpAtom#;  reduction
39 38 assignExpAtom#; reduction
40 39 argConst#; reduction
41 40 ;#,      move
42 41 blockItem#return reduction
43 42 stmt#return  reduction
44 43 return#return move
45 44 argExp#Ident  reduction
```

```
46 45 exp#Ident reduction
47 46 assignExp#Ident reduction
48 47 Ident#Ident move
49 48 callFunc#; reduction
50 49 mulExpAtom#; reduction
51 50 addExpAtom#; reduction
52 51 relExpAtom#; reduction
53 52 eqExpAtom#; reduction
54 53 assignExpAtom#; reduction
55 54 ;#; move
56 55 blockItem#} reduction
57 56 }#} move
58 57 compUnit## reduction
59 58 ### accept
```

## 测试样例5

```
1  struct my_struct1 {
2
3      int value;
4
5      void my_func(struct a b, int my_int, struct my_struct ccc) {}
6
7      struct my_struct me;
8  } s1, s2;
9
10 struct my_struct2 {
11
12     int value;
13
14     void my_func(struct a b, int my_int, struct my_struct ccc) {}
15
16     struct my_struct me;
17 } s;
18
19 struct my_struct3 {
20
21     int value;
22
23     void my_func(struct a b, int my_int, struct my_struct ccc) {}
24
25     struct my_struct me;
26
27     union my_union mu;
28 };
```

## 词法分析结果

```
1 void <KW,4>
2 main <IDN,main>
3 ( <SE,23>
4 ) <SE,24>
5 { <SE,25>
6   int <KW,1>
7   x <IDN,x>
8   = <OP,12>
9   2 <INT,2>
10  ; <SE,27>
11  int <KW,1>
12  y <IDN,y>
13  = <OP,12>
14  3 <INT,3>
15  ; <SE,27>
16  int <KW,1>
17  = <OP,12>
18  2 <INT,2>
19  ; <SE,27>
20  switch <KW,31>
21  ( <SE,23>
22  x <IDN,x>
23  ) <SE,24>
24  { <SE,25>
25  case <KW,33>
26  1 <INT,1>
27  : <SE,35>
28  { <SE,25>
29  y <IDN,y>
30  = <OP,12>
31  10 <INT,10>
32  ; <SE,27>
33  break <KW,34>
34  ; <SE,27>
35  } <SE,26>
36  case <KW,33>
37  2 <INT,2>
38  : <SE,35>
39  { <SE,25>
40  y <IDN,y>
41  = <OP,12>
42  20 <INT,20>
43  ; <SE,27>
44  func <IDN,func>
45  ( <SE,23>
```

```
46  y <IDN,y>
47  ) <SE,24>
48  ; <SE,27>
49  break <KW,34>
50  ; <SE,27>
51  } <SE,26>
52  default <KW,32>
53  : <SE,35>
54  { <SE,25>
55  return <KW,5>
56  0 <INT,0>
57  ; <SE,27>
58  } <SE,26>
59  } <SE,26>
60  } <SE,26>
61
```

## 语法分析结果

```

1  0  program#struct reduction
2  1  compUnit#struct  reduction
3  2  structDef#struct reduction
4  3  struct#struct  move
5  4  structType#Ident reduction
6  5  Ident#Ident    move
7  6  #{#  move
8  7  structBlockElem#int  reduction
9  8  int#int      move
10 9  varDef#Ident  reduction
11 10 Ident#Ident  move
12 11 argVarDef#;  reduction
13 12 argVarDecl#; reduction
14 13 ;#;  move
15 14 structBlockElem#void  reduction
16 15 void#void move
17 16 Ident#Ident  move
18 17 (#(  move
19 18 funcFParams#struct reduction
20 19 funcFParam#struct reduction
21 20 struct#struct  move
22 21 structType#Ident reduction
23 22 Ident#Ident  move
24 23 Ident#Ident  move
25 24 argFunctionF#, reduction
26 25 ,#,  move
27 26 funcFParam#int reduction
28 27 int#int      move
29 28 Ident#Ident  move
30 29 argFunctionF#, reduction
31 30 ,#,  move
32 31 funcFParam#struct reduction
33 32 struct#struct  move
34 33 structType#Ident reduction
35 34 Ident#Ident  move
36 35 Ident#Ident  move
37 36 argFunctionF#) reduction
38 37 )#)  move
39 38 block#{  reduction
40 39 #{#  move
41 40 blockItem#}  reduction
42 41 }#}  move
43 42 structBlockElem#struct reduction
44 43 structDecl#struct reduction
45 44 struct#struct  move

```



```

46 45 structType#Ident    reduction
47 46 Ident#Ident      move
48 47 Ident#Ident      move
49 48 ;#;      move
50 49 structBlockElem#}  reduction
51 50 }#}      move
52 51 structVar#Ident    reduction
53 52 Ident#Ident      move
54 53 argStructVar#,      reduction
55 54 ,#,      move
56 55 Ident#Ident      move
57 56 argStructVar#;      reduction
58 57 ;#;      move
59 58 compUnit#struct    reduction
60 59 structDef#struct    reduction
61 60 struct#struct      move
62 61 structType#Ident    reduction
63 62 Ident#Ident      move
64 63 {#{      move
65 64 structBlockElem#int    reduction
66 65 int#int      move
67 66 varDef#Ident      reduction
68 67 Ident#Ident      move
69 68 argVarDef#;      reduction
70 69 argVarDecl#;      reduction
71 70 ;#;      move
72 71 structBlockElem#void    reduction
73 72 void#void      move
74 73 Ident#Ident      move
75 74 (#(      move
76 75 funcFParams#struct    reduction
77 76 funcFParam#struct    reduction
78 77 struct#struct      move
79 78 structType#Ident    reduction
80 79 Ident#Ident      move
81 80 Ident#Ident      move
82 81 argFunctionF#,      reduction
83 82 ,#,      move
84 83 funcFParam#int      reduction
85 84 int#int      move
86 85 Ident#Ident      move
87 86 argFunctionF#,      reduction
88 87 ,#,      move
89 88 funcFParam#struct    reduction
90 89 struct#struct      move
91 90 structType#Ident    reduction
92 91 Ident#Ident      move
93 92 Ident#Ident      move

```

```

94 93 argFunctionF#) reduction
95 94 )#) move
96 95 block#{ reduction
97 96 #{ move
98 97 blockItem#} reduction
99 98 }#} move
100 99 structBlockElem#struct reduction
101 100 structDecl#struct reduction
102 101 struct#struct move
103 102 structType#Ident reduction
104 103 Ident#Ident move
105 104 Ident#Ident move
106 105 ;#; move
107 106 structBlockElem#} reduction
108 107 }#} move
109 108 structVar#Ident reduction
110 109 Ident#Ident move
111 110 argStructVar#; reduction
112 111 ;#; move
113 112 compUnit#struct reduction
114 113 structDef#struct reduction
115 114 struct#struct move
116 115 structType#Ident reduction
117 116 Ident#Ident move
118 117 #{ move
119 118 structBlockElem#int reduction
120 119 int#int move
121 120 varDef#Ident reduction
122 121 Ident#Ident move
123 122 argVarDef#; reduction
124 123 argVarDecl#; reduction
125 124 ;#; move
126 125 structBlockElem#void reduction
127 126 void#void move
128 127 Ident#Ident move
129 128 (#( move
130 129 funcFParams#struct reduction
131 130 funcFParam#struct reduction
132 131 struct#struct move
133 132 structType#Ident reduction
134 133 Ident#Ident move
135 134 Ident#Ident move
136 135 argFunctionF#, reduction
137 136 ,#, move
138 137 funcFParam#int reduction
139 138 int#int move
140 139 Ident#Ident move
141 140 argFunctionF#, reduction

```

```

142 141 ,#,      move
143 142 funcFParam#struct reduction
144 143 struct#struct move
145 144 structType#Ident reduction
146 145 Ident#Ident move
147 146 Ident#Ident move
148 147 argFunctionF#) reduction
149 148 )#)      move
150 149 block#{    reduction
151 150 #{      move
152 151 blockItem#} reduction
153 152 }#}      move
154 153 structBlockElem#struct reduction
155 154 structDecl#struct reduction
156 155 struct#struct move
157 156 structType#Ident reduction
158 157 Ident#Ident move
159 158 Ident#Ident move
160 159 ;#;      move
161 160 structBlockElem#union reduction
162 161 unionDecl#union reduction
163 162 union#union move
164 163 unionType#Ident reduction
165 164 Ident#Ident move
166 165 Ident#Ident move
167 166 ;#;      move
168 167 structBlockElem#} reduction
169 168 }#}      move
170 169 structVar#; reduction
171 170 ;#;      move
172 171 compUnit## reduction
173 172 ###      accept

```

## 测试样例6

```
1  void main() {  
2      int x = 2;  
3      int y = 3;  
4      int =2;  
5      switch (x) {  
6          case 1: {  
7              y = 10;  
8              break;  
9          }  
10         case 2: {  
11             y = 20;  
12             func(y);  
13             break;  
14         }  
15         default: {  
16             return 0;  
17         }  
18     }  
19 }
```

## 词法分析结果

```
1  struct <KW,29>
2  my_struct1 <IDN,my_struct1>
3  { <SE,25>
4  int <KW,1>
5  value <IDN,value>
6  ; <SE,27>
7  void <KW,4>
8  my_func <IDN,my_func>
9  ( <SE,23>
10 struct <KW,29>
11 a <IDN,a>
12 b <IDN,b>
13 , <SE,28>
14 int <KW,1>
15 my_int <IDN,my_int>
16 , <SE,28>
17 struct <KW,29>
18 my_struct <IDN,my_struct>
19 ccc <IDN,ccc>
20 ) <SE,24>
21 { <SE,25>
22 } <SE,26>
23 struct <KW,29>
24 my_struct <IDN,my_struct>
25 me <IDN,me>
26 ; <SE,27>
27 } <SE,26>
28 s1 <IDN,s1>
29 , <SE,28>
30 s2 <IDN,s2>
31 ; <SE,27>
32 struct <KW,29>
33 my_struct2 <IDN,my_struct2>
34 { <SE,25>
35 int <KW,1>
36 value <IDN,value>
37 ; <SE,27>
38 void <KW,4>
39 my_func <IDN,my_func>
40 ( <SE,23>
41 struct <KW,29>
42 a <IDN,a>
43 b <IDN,b>
44 , <SE,28>
45 int <KW,1>
```

```

46 my_int <IDN,my_int>
47 , <SE,28>
48 struct <KW,29>
49 my_struct <IDN,my_struct>
50 ccc <IDN,ccc>
51 ) <SE,24>
52 { <SE,25>
53 } <SE,26>
54 struct <KW,29>
55 my_struct <IDN,my_struct>
56 me <IDN,me>
57 ; <SE,27>
58 } <SE,26>
59 s <IDN,s>
60 ; <SE,27>
61 struct <KW,29>
62 my_struct3 <IDN,my_struct3>
63 { <SE,25>
64 int <KW,1>
65 value <IDN,value>
66 ; <SE,27>
67 void <KW,4>
68 my_func <IDN,my_func>
69 ( <SE,23>
70 struct <KW,29>
71 a <IDN,a>
72 b <IDN,b>
73 , <SE,28>
74 int <KW,1>
75 my_int <IDN,my_int>
76 , <SE,28>
77 struct <KW,29>
78 my_struct <IDN,my_struct>
79 ccc <IDN,ccc>
80 ) <SE,24>
81 { <SE,25>
82 } <SE,26>
83 struct <KW,29>
84 my_struct <IDN,my_struct>
85 me <IDN,me>
86 ; <SE,27>
87 union <KW,30>
88 my_union <IDN,my_union>
89 mu <IDN,mu>
90 ; <SE,27>
91 } <SE,26>
92 ; <SE,27>
93

```

# 语法分析结果

```
1  0  program#void    reduction
2  1  compUnit#void   reduction
3  2  funcDef#void    reduction
4  3  funcType#void   reduction
5  4  void#void       move
6  5  Ident#Ident     move
7  6  (#(      move
8  7  funcFParams#)   reduction
9  8  )#)      move
10 9  block#{        reduction
11 10 {#{      move
12 11 blockItem#int   reduction
13 12 int#int        move
14 13 varDef#Ident    reduction
15 14 Ident#Ident     move
16 15 argVarDef#==     reduction
17 16 ==#           move
18 17 initVal#INT      reduction
19 18 exp#INT          reduction
20 19 assignExp#INT     reduction
21 20 INT#INT          move
22 21 mulExpAtom#;      reduction
23 22 addExpAtom#;      reduction
24 23 relExpAtom#;      reduction
25 24 eqExpAtom#;      reduction
26 25 assignExpAtom#;   reduction
27 26 argVarDecl#;      reduction
28 27 ;#;             move
29 28 blockItem#int     reduction
30 29 int#int          move
31 30 varDef#Ident      reduction
32 31 Ident#Ident       move
33 32 argVarDef#==      reduction
34 33 ==#             move
35 34 initVal#INT       reduction
36 35 exp#INT           reduction
37 36 assignExp#INT      reduction
38 37 INT#INT           move
39 38 mulExpAtom#;      reduction
40 39 addExpAtom#;      reduction
41 40 relExpAtom#;      reduction
42 41 eqExpAtom#;      reduction
43 42 assignExpAtom#;   reduction
44 43 argVarDecl#;      reduction
45 44 ;#;             move
```



```
46 45 blockItem#int reduction
47 46 int#int move
48 47 varDef# error
```

## 测试样例7

▼ Plain Text

```
1 union my_union1 {
2
3 int value;
4 float weight;
5 } u1, u2;
6
7
8 union my_union2 {
9
10 int value;
11
12 union my_union1 my_union;
13
14 struct my_struct1 my_struct;
15 } u;
16
17 union my_union3 {
18
19 int value;
20 char id;
21 };
```

## 词法分析结果

```
1  union <KW,30>
2  my_union1 <IDN,my_union1>
3  { <SE,25>
4  int <KW,1>
5  value <IDN,value>
6  ; <SE,27>
7  float <KW,2>
8  weight <IDN,weight>
9  ; <SE,27>
10 } <SE,26>
11 u1 <IDN,u1>
12 , <SE,28>
13 u2 <IDN,u2>
14 ; <SE,27>
15 union <KW,30>
16 my_union2 <IDN,my_union2>
17 { <SE,25>
18 int <KW,1>
19 value <IDN,value>
20 ; <SE,27>
21 union <KW,30>
22 my_union1 <IDN,my_union1>
23 my_union <IDN,my_union>
24 ; <SE,27>
25 struct <KW,29>
26 my_struct1 <IDN,my_struct1>
27 my_struct <IDN,my_struct>
28 ; <SE,27>
29 } <SE,26>
30 u <IDN,u>
31 ; <SE,27>
32 union <KW,30>
33 my_union3 <IDN,my_union3>
34 { <SE,25>
35 int <KW,1>
36 value <IDN,value>
37 ; <SE,27>
38 char <KW,3>
39 id <IDN,id>
40 ; <SE,27>
41 } <SE,26>
42 ; <SE,27>
43
```

# 语法分析结果

```
1  0  program#union  reduction
2  1  compUnit#union reduction
3  2  unionDef#union reduction
4  3  union#union   move
5  4  unionType#Ident reduction
6  5  Ident#Ident   move
7  6  {#{         move
8  7  unionBlockElem#int reduction
9  8  int#int       move
10 9  varDef#Ident  reduction
11 10 Ident#Ident   move
12 11 argVarDef#;   reduction
13 12 argVarDecl#;  reduction
14 13 ;#;          move
15 14 unionBlockElem#float reduction
16 15 float#float    move
17 16 varDef#Ident  reduction
18 17 Ident#Ident   move
19 18 argVarDef#;   reduction
20 19 argVarDecl#;  reduction
21 20 ;#;          move
22 21 unionBlockElem#} reduction
23 22 }#}          move
24 23 unionVar#Ident reduction
25 24 Ident#Ident   move
26 25 argUnionVar#,  reduction
27 26 ,#,          move
28 27 Ident#Ident   move
29 28 argUnionVar#;  reduction
30 29 ;#;          move
31 30 compUnit#union reduction
32 31 unionDef#union reduction
33 32 union#union   move
34 33 unionType#Ident reduction
35 34 Ident#Ident   move
36 35 {#{         move
37 36 unionBlockElem#int reduction
38 37 int#int       move
39 38 varDef#Ident  reduction
40 39 Ident#Ident   move
41 40 argVarDef#;   reduction
42 41 argVarDecl#;  reduction
43 42 ;#;          move
44 43 unionBlockElem#union reduction
45 44 unionDecl#union reduction
```

```

46 45 union#union      move
47 46 unionType#Ident  reduction
48 47 Ident#Ident      move
49 48 Ident#Ident      move
50 49 ;;              move
51 50 unionBlockElem#struct reduction
52 51 struct#struct    move
53 52 structType#Ident  reduction
54 53 Ident#Ident      move
55 54 Ident#Ident      move
56 55 ;;              move
57 56 unionBlockElem#}  reduction
58 57 }#}              move
59 58 unionVar#Ident    reduction
60 59 Ident#Ident      move
61 60 argUnionVar#;      reduction
62 61 ;;              move
63 62 compUnit#union     reduction
64 63 unionDef#union     reduction
65 64 union#union        move
66 65 unionType#Ident    reduction
67 66 Ident#Ident        move
68 67 {#{              move
69 68 unionBlockElem#int reduction
70 69 int#int            move
71 70 varDef#Ident       reduction
72 71 Ident#Ident        move
73 72 argVarDef#;        reduction
74 73 argVarDecl#;      reduction
75 74 ;;              move
76 75 unionBlockElem#char reduction
77 76 char#char          move
78 77 varDef#Ident       reduction
79 78 Ident#Ident        move
80 79 argVarDef#;        reduction
81 80 argVarDecl#;      reduction
82 81 ;;              move
83 82 unionBlockElem#}  reduction
84 83 }#}              move
85 84 unionVar#;         reduction
86 85 ;;              move
87 86 compUnit##         reduction
88 87 ###              accept

```