

编译原理第一次实验测试用例：目录

1	A 组测试用例	2
1.1	A-1	2
1.2	A-2	2
1.3	A-3	3
1.4	A-4	3
1.5	A-5	3
1.6	A-6	4
1.7	A-7	4
1.8	A-8	5
1.9	A-9	6
2	B 组测试用例	6
2.1	B-1	6
2.2	B-2	7
3	C 组测试用例	9
3.1	C-1	9
3.2	C-2	12
4	D 组测试用例	18
4.1	D-1	18
4.2	D-2	20
4.3	D-3	23
5	E 组测试用例	25
5.1	E1.1	25
5.2	E1.2	26
5.3	E1.3	27
6	结束语	31

1 A 组测试用例

本组测试用例共 9 个，每个仅包含单个的词法或者语法错误。除特殊说明外，不可多报。多报、漏报错误，或者打印语法树都会导致扣分。错误编号和行号之后的说明文字不要求与给出的输出完全一致，仅供助教理解使用，不作为评分依据。

1.1 A-1

输入

```
1 int func()  
2 {  
3     int _a_aaaaaaaaaaaaaaaaaaaaaa, 4b;  
4 }
```

输出

```
1 Error type B at line 3 : syntax error, near '4b'
```

说明：错误类型也可以是 A 类，或者一个 A 一个 B，但是只能在第 3 行。这里有一个非法的标识符 4b。

1.2 A-2

输入

```
1 int sub()  
2 {  
3     int a, b;  
4     a = a + 1;  
5     b = a & 2;  
6     return a - b;  
7 }
```

输出

```
1 Error type A at line 5 : mysteriously character '&'
```

说明：必须有 type A 错误；可以多报一个 type B 错误。这里有一个非法的 & 符号。

1.3 A-3

输入

```
1 int add()  
2 {  
3     int a, b, c;  
4     c = ((a + (b + 1)));  
5 }
```

输出

```
1 Error type B at line 4 : syntax error, near ')'
```

说明：第 4 行有未匹配的括号。

1.4 A-4

输入

```
1 int func()  
2 {  
3     int _a, b  
4     b = 0;  
5     _a = b + 1;  
6 }
```

输出

```
1 Error type B at line 3 : syntax error, near 'b'
```

说明：第 3 行结尾少了分号，因此错误位置报在第 4 行也算对。

1.5 A-5

输入

```
1 int arrayDef()  
2 {  
3     int i, a[2][100];  
4     i = 0;
```

```

5     while (i > 100) {
6         a[0][i]= 1;
7         a[1,i] = i;
8         i = i + 1;
9     }
10 }

```

输出

```

1 Error type B at line 7 : syntax error, near ', '

```

说明：第 7 行错误的数组下标格式

1.6 A-6

输入

```

1 int func()
2 {
3     int i = 0, a[2][5.8][10];
4     while (i < 10) {
5         a[0][0][i] = i * i;
6         a[1][0][i] = i / i;
7     }
8 }

```

输出

```

1 Error type B at line 3 : syntax error, near '5.8'

```

说明：第 3 行数组声明的时候使用了浮点数 5.8

1.7 A-7

输入

```

1 struct Vector {
2     float x, y;
3     int a, b;

```

```

4     float c, d;
5 }
6
7 int sdef()
8 {
9     struct Vector v;
10    v.x = v.y = 1.0;
11    v.a = v.b = 1;
12    v.c = v.d = 3.5;
13 }

```

输出

```

1 Error type B at line 7 : syntax error, near 'int'

```

说明：结构体定义时少了分号，因此错误位置在 5、6 行也算正确。

1.8 A-8

输入

```

1 struct Vector {
2     float x, y;
3 };
4
5 int sdef()
6 {
7     Vector v;
8     v.x = v.y = 1.0;
9 }

```

输出

```

1 Error type B at line 7 : syntax error, near 'v'

```

说明：第 7 行定义结构体变量时少了 struct 关键字。

1.9 A-9

输入

```
1 int m1(int arg)
2 {
3     int a;
4     a = arg + 5;
5     return a;
6 }
7
8 int m2()
9 {
10    int b = 2, c;
11    c = m1 (b);
12 }
```

输出

```
1 Error type B at line 11 : syntax error, near ';;'
```

说明：第 11 行在函数调用时多了一个左括号。

2 B 组测试用例

本组测试用例共 2 个，每个用例包含多处不同的错误。除特殊说明外，漏报、多报错误或者出打印语法树都会导致扣分。

2.1 B-1

输入

```
1 struct Vector {
2     float x y;
3 };
4
5 float squaring(struct Vector v0, struct Vector v1:)
6 {
```

```

7     struct Vector c0 = v0;
8     struct Vector c1 = v1;
9     return (c0.x - c1.x) * (c0.y - c1.y);
10 }
11
12 float main()
13 {
14     strut Vector v0, v1;
15     v0.x = 1.2;
16     v0.y = 3.6;
17     v1.x = 4.5;
18     v1.y = 6.7;
19     return squaring(v0_0, v0_1));
20 }

```

输出

```

1 Error type B at line 2 : syntax error, near 'y'
2 Error type A at line 5 : mysterious character ':'
3 Error type B at line 14 : syntax error, near 'strut'
4 Error type B at line 19 : syntax error, near ';'

```

说明：第 2 行缺少一个逗号；第 5 行多了一个分号；第 14 行 struct 关键字错写为 strut；第 19 行多了一个右括号。

2.2 B-2

输入

```

1 int func()
2 {
3     int _arg1[5], _arg2[5];
4     int i = 0, j = 0, tmp;
5     while (i+1) < 6) {
6         _arg1[i] = i + 2;
7         i = ++;

```

```

8     }
9
10    j = 5;
11    while (j > 0) {
12        _arg2[j] = _arg1[j] - j;
13        if (_arg1[j] > _arg2[j] +i) {
14            tmp = _arg1[j];
15            _arg1[j] = _arg2[j];
16            _arg2[j] = tmp;
17        }
18        j--;
19    }
20
21    i = 0;
22    while (i < 5) {
23        printInt(_arg1[i]);
24        printInt(_arg2[i]);
25        i = i + 1;
26    }
27    return _arg1[0] + _arg2[0]
28 }

```

输出

```

1 Error type B at line 5 : syntax error, near '('
2 Error type B at line 7 : syntax error, near '+'
3 Error type B at line 18 : syntax error, near '-'
4 Error type B at line 28 : syntax error, near '}'

```

说明：第 5 行少了一个左括号符号；第 7 行使用了错误的 ++ 操作符；第 18 行使用了错误的 - 操作符；第 28 行上一句 return 语句少了分号。

3 C 组测试用例

本组测试用例共 2 个，不包含任何错误，需要输出正确的语法树。除特殊说明外，应与给出的语法树完全相同。语法树打印错误酌情扣分。

3.1 C-1

输入

```
1 struct Vector {  
2     int a;  
3     int b;  
4 };  
5  
6 int test()  
7 {  
8     struct Vector v1;  
9     v1.a = -1;  
10    v1.b = v1.a + 2;  
11    return v1.a - v1.b;  
12 }
```

输出

```
1 Program (1)  
2   ExtDefList (1)  
3     ExtDef (1)  
4       Specifier (1)  
5         StructSpecifier (1)  
6           STRUCT  
7           OptTag (1)  
8             ID: Vector  
9           LC  
10          DefList (2)  
11            Def (2)  
12              Specifier (2)
```

```

13         TYPE: int
14     DecList (2)
15         Dec (2)
16         VarDec (2)
17             ID: a
18     SEMI
19     DefList (3)
20         Def (3)
21             Specifier (3)
22                 TYPE: int
23             DecList (3)
24                 Dec (3)
25                 VarDec (3)
26                     ID: b
27     SEMI
28     RC
29     SEMI
30     ExtDefList (6)
31         ExtDef (6)
32             Specifier (6)
33                 TYPE: int
34             FunDec (6)
35                 ID: test
36             LP
37             RP
38         CompSt (7)
39             LC
40             DefList (8)
41                 Def (8)
42                     Specifier (8)
43                         StructSpecifier (8)
44                             STRUCT

```

```

45         Tag (8)
46         ID: Vector
47     DeclList (8)
48     Dec (8)
49     VarDec (8)
50     ID: v1
51     SEMI
52 StmtList (9)
53     Stmt (9)
54     Exp (9)
55     Exp (9)
56     Exp (9)
57     ID: v1
58     DOT
59     ID: a
60     ASSIGNOP
61     Exp (9)
62     MINUS
63     Exp (9)
64     INT: 1
65     SEMI
66 StmtList (10)
67     Stmt (10)
68     Exp (10)
69     Exp (10)
70     Exp (10)
71     ID: v1
72     DOT
73     ID: b
74     ASSIGNOP
75     Exp (10)
76     Exp (10)

```

```

77         Exp (10)
78         ID: v1
79         DOT
80         ID: a
81         PLUS
82         Exp (10)
83         INT: 2
84     SEMI
85 StmtList (11)
86     Stmt (11)
87     RETURN
88     Exp (11)
89     Exp (11)
90     Exp (11)
91     ID: v1
92     DOT
93     ID: a
94     MINUS
95     Exp (11)
96     Exp (11)
97     ID: v1
98     DOT
99     ID: b
100 SEMI
101 RC

```

说明：使用的空格可以换位 Tab。

3.2 C-2

输入

```

1 int test()
2 {
3     int a[4][5];

```

```

4      int i = 0;
5      a[i][i] = 4;
6      while (i < 4) {
7          a[i][0] = i;
8          a[1][i] = i + 1;
9          i = i + 1;
10     }
11     return a[0][0];
12 }

```

输出

```

1 Program (1)
2   ExtDefList (1)
3     ExtDef (1)
4       Specifier (1)
5         TYPE: int
6       FunDec (1)
7         ID: test
8         LP
9         RP
10      CompSt (2)
11        LC
12        DefList (3)
13          Def (3)
14            Specifier (3)
15              TYPE: int
16            DecList (3)
17              Dec (3)
18                VarDec (3)
19                  VarDec (3)
20                    VarDec (3)
21                      ID: a
22                      LB

```

```

23             INT: 4
24             RB
25             LB
26             INT: 5
27             RB
28         SEMI
29     DefList (4)
30         Def (4)
31             Specifier (4)
32                 TYPE: int
33             DecList (4)
34                 Dec (4)
35                     VarDec (4)
36                         ID: i
37                     ASSIGNOP
38                     Exp (4)
39                         INT: 0
40             SEMI
41     StmtList (5)
42         Stmt (5)
43             Exp (5)
44                 Exp (5)
45                     Exp (5)
46                         Exp (5)
47                             ID: a
48                         LB
49                         Exp (5)
50                             ID: i
51                         RB
52                     LB
53                     Exp (5)
54                         ID: i

```

55	RB
56	ASSIGNOP
57	Exp (5)
58	INT: 4
59	SEMI
60	StmtList (6)
61	Stmt (6)
62	WHILE
63	LP
64	Exp (6)
65	Exp (6)
66	ID: i
67	RELOP
68	Exp (6)
69	INT: 4
70	RP
71	Stmt (6)
72	CompSt (6)
73	LC
74	StmtList (7)
75	Stmt (7)
76	Exp (7)
77	Exp (7)
78	Exp (7)
79	Exp (7)
80	ID: a
81	LB
82	Exp (7)
83	ID: i
84	RB
85	LB
86	Exp (7)

87	INT: 0
88	RB
89	ASSIGNOP
90	Exp (7)
91	ID: i
92	SEMI
93	StmtList (8)
94	Stmt (8)
95	Exp (8)
96	Exp (8)
97	Exp (8)
98	Exp (8)
99	ID: a
100	LB
101	Exp (8)
102	INT: 1
103	RB
104	LB
105	Exp (8)
106	ID: i
107	RB
108	ASSIGNOP
109	Exp (8)
110	Exp (8)
111	ID: i
112	PLUS
113	Exp (8)
114	INT: 1
115	SEMI
116	StmtList (9)
117	Stmt (9)
118	Exp (9)

119	Exp (9)
120	ID: i
121	ASSIGNOP
122	Exp (9)
123	Exp (9)
124	ID: i
125	PLUS
126	Exp (9)
127	INT: 1
128	SEMI
129	RC
130	StmtList (11)
131	Stmt (11)
132	RETURN
133	Exp (11)
134	Exp (11)
135	Exp (11)
136	ID: a
137	LB
138	Exp (11)
139	INT: 0
140	RB
141	LB
142	Exp (11)
143	INT: 0
144	RB
145	SEMI
146	RC

说明：考察对数组的翻译。

4 D 组测试用例

本组测试用例共 3 个，针对不同分组进行测试。对应分组的同学需要输出语法树，提示错误则不得分；其他分组的同学只要提示错误即可，如果打印了语法树，则将视为违规，将会倒扣分。

4.1 D-1

输入

```
1 int func()  
2 {  
3     int oct = 0125;  
4     int dec = 239;  
5     int hex1 = 0x0B0c;  
6     int dhex2 = 0XD01;  
7 }
```

输出

```
1 Program (1)  
2   ExtDefList (1)  
3     ExtDef (1)  
4       Specifier (1)  
5         TYPE: int  
6       FunDec (1)  
7         ID: func  
8         LP  
9         RP  
10      CompSt (2)  
11        LC  
12      DefList (3)  
13        Def (3)  
14          Specifier (3)  
15            TYPE: int  
16          DecList (3)
```

```

17         Dec (3)
18         VarDec (3)
19             ID: oct
20         ASSIGNOP
21         Exp (3)
22             INT: 85
23     SEMI
24 DefList (4)
25     Def (4)
26         Specifier (4)
27             TYPE: int
28         DecList (4)
29             Dec (4)
30                 VarDec (4)
31                     ID: dec
32                 ASSIGNOP
33                 Exp (4)
34                     INT: 239
35     SEMI
36 DefList (5)
37     Def (5)
38         Specifier (5)
39             TYPE: int
40         DecList (5)
41             Dec (5)
42                 VarDec (5)
43                     ID: hex1
44                 ASSIGNOP
45                 Exp (5)
46                     INT: 2828
47     SEMI
48 DefList (6)

```

```

49         Def (6)
50             Specifier (6)
51                 TYPE: int
52             DecList (6)
53                 Dec (6)
54                     VarDec (6)
55                         ID: dhex2
56                         ASSIGNOP
57                         Exp (6)
58                             INT: 3329
59                 SEMI
60     RC

```

说明：1.1 分组（任务 ID 1,4,7,10,13,16）的同学需要输出该语法树，8 进制和 16 进制数必须正确转换（85、239、2828 和 3329）；其他分组的同学只要提示有错误，而且不输出语法树即可。

4.2 D-2

输入

```

1 int main()
2 {
3     float X_1 = 2.42e-3;
4     float X_2 = 1.12E2;
5     float X_3 = 66.2e+2;
6     float X_4 = 1.62E+2;
7     float X_5 = .221E-2;
8     float X_6 = 3.24e1;
9 }

```

输出

```

1 Program (1)
2     ExtDefList (1)
3         ExtDef (1)
4             Specifier (1)

```

```

5      TYPE: int
6  FunDec (1)
7      ID: main
8      LP
9      RP
10  CompSt (2)
11      LC
12      DefList (3)
13          Def (3)
14              Specifier (3)
15                  TYPE: float
16              DecList (3)
17                  Dec (3)
18                      VarDec (3)
19                          ID: X_1
20                          ASSIGNOP
21                          Exp (3)
22                              FLOAT: 0.002420
23                      SEMI
24              DefList (4)
25                  Def (4)
26                      Specifier (4)
27                          TYPE: float
28                      DecList (4)
29                          Dec (4)
30                              VarDec (4)
31                                  ID: X_2
32                                  ASSIGNOP
33                                  Exp (4)
34                                      FLOAT: 112.000000
35                              SEMI
36                      DefList (5)

```

```

37      Def (5)
38          Specifier (5)
39              TYPE: float
40      DecList (5)
41          Dec (5)
42              VarDec (5)
43                  ID: X_3
44                  ASSIGNOP
45              Exp (5)
46                  FLOAT: 6620.000000
47      SEMI
48  DefList (6)
49      Def (6)
50          Specifier (6)
51              TYPE: float
52      DecList (6)
53          Dec (6)
54              VarDec (6)
55                  ID: X_4
56                  ASSIGNOP
57              Exp (6)
58                  FLOAT: 162.000000
59      SEMI
60  DefList (7)
61      Def (7)
62          Specifier (7)
63              TYPE: float
64      DecList (7)
65          Dec (7)
66              VarDec (7)
67                  ID: X_5
68                  ASSIGNOP

```

```

69         Exp (7)
70         FLOAT: 0.002210
71     SEMI
72     DefList (8)
73     Def (8)
74     Specifier (8)
75     TYPE: float
76     DecList (8)
77     Dec (8)
78     VarDec (8)
79     ID: X_6
80     ASSIGNOP
81     Exp (8)
82     FLOAT: 32.400000
83     SEMI
84 RC

```

说明：1.2 分组（任务 ID 2,5,8,11,14,17）的同学需要输出语法树注意科学计数法浮点数的正确转换。其他分组同学只要提示出错，而且不输出语法树即可。

4.3 D-3

输入

```

1 int test()
2 { // a line of comments
3     float a = 4.5;
4     /*
5     float a = 0.1;
6     /* */
7 //} // float b = 4;
8     float b = 3;
9     /* float b = 0.4; /*****
10    int b = 2
11    */

```

```
12     return a + b;
13 }
```

输出

```
1 Program (1)
2   ExtDefList (1)
3     ExtDef (1)
4       Specifier (1)
5         TYPE: int
6       FunDec (1)
7         ID: test
8         LP
9         RP
10      CompSt (2)
11        LC
12        DefList (3)
13          Def (3)
14            Specifier (3)
15              TYPE: float
16            DecList (3)
17              Dec (3)
18                VarDec (3)
19                  ID: a
20                  ASSIGNOP
21                  Exp (3)
22                    FLOAT: 4.500000
23              SEMI
24            DefList (5)
25              Def (5)
26                Specifier (5)
27                  TYPE: float
28                DecList (5)
29                  Dec (5)
```



```

30          VarDec (5)
31          ID: b
32          ASSIGNOP
33          Exp (5)
34          INT: 3
35          SEMI
36      StmtList (7)
37      Stmt (7)
38      RETURN
39      Exp (7)
40      Exp (7)
41      ID: a
42      PLUS
43      Exp (7)
44      ID: b
45      SEMI
46  RC

```

说明：1.3 分组（任务 ID 3,6,9,12,15,18）的同学需要输出语法树，不能提示有语法错误；其他分组同学只需提示有错误，且不输出语法树即可。

5 E 组测试用例

本组测试用例共 6 个，针对不同分组进行测试

5.1 E1.1

这组测试用例针对 1.1 分组的同学

输入（E1-1）

```

1  int func()
2  {
3      int a = 0X1ha;
4  }

```

输出

```
1 Error type B at line 3 : syntax error, near 'ha'
```

说明：仅 1.1 分组同学需要测试这个用例，针对错误的 16 进制数 0X1ha。

输入（E1-2）

```
1 int test()  
2 {  
3     int _C_ = 0109;  
4 }
```

输出

```
1 Error type B at line 3 : syntax error, near '09'
```

说明：仅 1.1 分组同学需要测试这个用例，针对错误的 8 进制数 0109。

5.2 E1.2

输入（E2-1）

```
1 float main()  
2 {  
3     float C_C = 4.23e.-1;  
4     float A_A = 1.69e-1;  
5     return C_C + A_A;  
6 }
```

输出

```
1 Error type B at line 3 : syntax error, near '.-'
```

说明：仅 1.2 分组同学需要测试这组用例，错误在于指数为 e.-1

输入（E2-2）

```
1 float func()  
2 {  
3     float a = .E-1;  
4 }
```

输出

```
1 Error type B at line 3 : syntax error, near '.'
```

说明：仅 1.2 分组同学需要测试这组用例，错误在于底数只有小数点

5.3 E1.3

输入（E3-1）

```
1 // \ \ // \ \ // \ \ \ \
2 // \ \ // \ \ // \ \ // \ \ //
3 /*
4 * I am a line of comments, but plz do not ignore me ~_~
5 */
6
7 float /* ^ ^ * > < * _/\_ */test()
8 {
9     /*
10    //~/~::~::~::~::~::~::~::~::~::~::~ */
11    float a, b, c;
12    // */
13    a = 1.5;
14    // a = b + 1;
15    /* *****\ \ \
16    // a line of comments
17    float a = 5;
18    */
19    b = 1.5;
20    c = (a + b) + (a + b);
21 } //a line of comments
22 /*
23 I am a line of comments
24 */
```

输出

```

1 Program (1)
2   ExtDefList (1)
3     ExtDef (1)
4       Specifier (1)
5         TYPE: float
6       FunDec (1)
7         ID: test
8         LP
9         RP
10      CompSt (2)
11        LC
12        DefList (3)
13          Def (3)
14            Specifier (3)
15              TYPE: float
16            DecList (3)
17              Dec (3)
18                VarDec (3)
19                  ID: a
20                COMMA
21              DecList (3)
22                Dec (3)
23                  VarDec (3)
24                    ID: b
25                  COMMA
26                DecList (3)
27                  Dec (3)
28                    VarDec (3)
29                      ID: c
30                SEMI
31          StmtList (5)
32            Stmt (5)

```

33	Exp (5)
34	Exp (5)
35	ID: a
36	ASSIGNOP
37	Exp (5)
38	FLOAT: 1.500000
39	SEMI
40	StmtList (7)
41	Stmt (7)
42	Exp (7)
43	Exp (7)
44	ID: b
45	ASSIGNOP
46	Exp (7)
47	FLOAT: 1.500000
48	SEMI
49	StmtList (8)
50	Stmt (8)
51	Exp (8)
52	Exp (8)
53	ID: c
54	ASSIGNOP
55	Exp (8)
56	Exp (8)
57	LP
58	Exp (8)
59	Exp (8)
60	ID: a
61	PLUS
62	Exp (8)
63	ID: b
64	RP

65	PLUS
66	Exp (8)
67	LP
68	Exp (8)
69	Exp (8)
70	ID: a
71	PLUS
72	Exp (8)
73	ID: b
74	RP
75	SEMI
76	RC

说明：必须输出正确的语法树，否则该用例不得分

输入 (E3-2)

```

1  /*
2  * @a line of comments
3  */
4
5  // a line of comments //
6
7  float /*~~~*/ test()//test(int a)
8  {
9      /*
10     float a, b, c, d; /* /float a, b;
11     c =  2 * a  2 * a - 3 * b; /3 * b;
12     /\ deleting ... 1, 2, 3 /\
13     d = a + b  a - b;
14  } */
15
16  float a, b, c, d; /* /float a, b; */
17  c = /* 2 * a */ 2 * a - 3 * b; //3 * b;
18  // deleting ... 1, 2, 3 \
19  d = /* a + b */ a - b; /* ; // a / b /

```

```
19 }//a line of comments
```

输出

```
1 提示语法错误即可
```

说明：针对未终止的注释进行测试，如果打印了语法树，或者程序异常终止、死循环无法退出等，则该用例不得分。不限定错误类型以及提示方式，但是出错位置必须限定在 19 行或者以后的位置；直接提示“未终止的注释”也可以。

6 结束语

如果对本测试用例有任何疑议，可以写邮件与杨文华助教联系，注意同时抄送给许老师。