

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	4
1.6	A-6	4
1.7	A-7	5
1.8	A-8	5
1.9	A-9	6
1.10	A-10	6
2	B 组测试用例	7
2.1	B-1	7
2.2	B-2	8
3	C 组测试用例	9
3.1	C-1	9
3.2	C-2	18
4	D 组测试用例	29
4.1	D-1	29
4.2	D-2	31
4.3	D-3	33
5	E 组测试用例	36
5.1	E1-1	36
5.2	E1-2	40
5.3	E2-1	41
5.4	E2-2	44
5.5	E3-1	44
5.6	E3-2	51
6	结束语	51

1 A 组测试用例

本组测试用例共 10 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

1.1.1 输入

```
1 int main() {  
2     int base = 100;  
3     int step = base $ 3;  
4     return step;  
5 }
```

1.1.2 输出

```
1 Error type A at Line 3: Undefined character '$'.
```

1.1.3 说明

未定义字符 \$（注：也可以识别为 B 类错误。）

1.2 A-2

1.2.1 输入

```
1 int main() {  
2     int count = 10;  
3     int limit = 20;  
4     int 2x = count + limit;  
5     return 0;  
6 }
```

1.2.2 输出

```
1 Error type A at Line 4: Invalid identifier starting with digit '2x'  
  '.
```

1.2.3 说明

以数字开头的非法标识符 2x（注：也可以识别为 B 类错误。）

1.3 A-3

1.3.1 输入

```
1 int clamp(int val) {  
2     if (val < 0) {  
3         return;  
4     }  
5     return val;  
6 }  
7  
8 int main() {  
9     return clamp(5);  
10 }
```

1.3.2 输出

```
1 Error type B at Line 3: Missing return value
```

1.3.3 说明

return 语句缺少返回值

1.4 A-4

1.4.1 输入

```
1 int total;  
2 float ratio = 1.5;  
3  
4 int main() {  
5     return 0;  
6 }
```

1.4.2 输出

```
1 Error type B at Line 2: Global variable initialization not allowed.
```

1.4.3 说明

全局变量带初始化赋值

1.5 A-5

1.5.1 输入

```
1 int main() {  
2     float bad = 3.14.15;  
3     return 0;  
4 }
```

1.5.2 输出

```
1 Error type A at Line 2: Invalid floating point literal '3.14.15'.
```

1.5.3 说明

非法浮点数字面量（多余的小数点）（注：也可以识别为 B 类错误。）

1.6 A-6

1.6.1 输入

```
1 int main() {  
2     int a = 1;  
3     int b = 2;  
4     int arr[1.5];  
5     return 0;  
6 }
```

1.6.2 输出

```
1 Error type B at Line 4: Array dimension must be integer.
```

1.6.3 说明

数组声明中使用浮点数作为维度

1.7 A-7

1.7.1 输入

```
1 int add(int x, int y) {  
2     return x + y;  
3 }  
4  
5 int main() {  
6     int r = add(3,);  
7     return r;  
8 }
```

1.7.2 输出

```
1 Error type B at Line 6: Trailing comma in function call arguments.
```

1.7.3 说明

函数调用实参列表末尾多余逗号

1.8 A-8

1.8.1 输入

```
1 int run_count;  
2  
3 int update(int inc) {  
4     int base = 0;  
5     if (inc > 0) {  
6         base = inc;  
7     }  
8     run_count += base;  
9     return run_count;  
10 }
```

1.8.2 输出

```
1 Error type B at Line 8: Unsupported compound assignment operator
   '+='.

```

1.8.3 说明

不支持的复合赋值运算符 +=

1.9 A-9

1.9.1 输入

```
1 int main() {
2     int x = 3;
3     int y = 4;
4     x = x + y;
5     int z;
6     return z;
7 }

```

1.9.2 输出

```
1 Error type B at Line 5: Variable declaration after statements in
   block.

```

1.9.3 说明

语句块中语句之后出现变量声明

1.10 A-10

1.10.1 输入

```
1 int main() {
2     int a;
3     a = 1;
4     a.;
5     return 0;
6 }

```

1.10.2 输出

```
1 Error type B at Line 4: Missing member name after '.'.
```

1.10.3 说明

点号访问表达式中 DOT 后缺少成员名

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、多报错误或者打印语法树都会导致扣分。

2.1 B-1

2.1.1 输入

```
1 struct Sensor {
2     int id;
3     float value;
4     int active;
5 };
6
7 int readings[2.0];
8
9 int offset;
10
11 int process(int mode,) {
12     struct { int tag; };
13     int total = 0;
14     float weights[4] = {0.25, 0.5, 1.0};
15     struct Sensor s;
16     s.id = mode;
17     s.value = 0.0;
18     s.active = 1;
19
20     while (s.value > 0.0);
21
22     while (s.value > 0.0) {
```

```

23     s.value = s.value - 1.0;
24     total = total + 1;
25 }
26
27 return total;
28 }

```

2.1.2 输出

```

1 Error type B at Line 7: Array dimension must be integer.
2 Error type B at Line 11: Trailing comma in function parameters.
3 Error type B at Line 12: Anonymous struct declaration.
4 Error type B at Line 14: Array initialization not supported.
5 Error type B at Line 20: Empty while statement body.

```

2.1.3 说明

同一程序中同时出现 5 类不同语法错误

2.2 B-2

2.2.1 输入

```

1 int data[5][];
2
3 int size;
4
5 int fill(int val) {
6     int i = 0;
7     while (i < 5) {
8         data[i] = val;
9         i = i + 1;
10    }
11    int result = 0;
12    return result;
13 }
14
15 int process(int n) {

```



```

16  float avg = 0.0;
17  avg = avg + 1.0;
18  int count;
19  count = 0;
20  while (avg > 0.0) {
21      count = count + 1;
22      avg = avg - 0.5;
23  }
24  int final;
25  final = count;
26  return final;
27  }

```

2.2.2 输出

```

1 Error type B at Line 1: Extra closing bracket in array declaration.
2 Error type B at Line 11: Variable declaration after statements in
  block.
3 Error type B at Line 18: Variable declaration after statements in
  block.
4 Error type B at Line 24: Variable declaration after statements in
  block.

```

2.2.3 说明

语句后出现变量声明（多处）及多余括号

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

3.1.1 输入

```

1 struct Stack {
2     int data[10];
3     int top;

```

```

4  };
5
6  int main() {
7      struct Stack st;
8      int i;
9      int total;
10
11     st.top = 0;
12     i = 0;
13     total = 0;
14
15     while (i < 5) {
16         st.data[st.top] = i * i;
17         st.top = st.top + 1;
18         i = i + 1;
19     }
20
21     while (st.top > 0) {
22         st.top = st.top - 1;
23         total = total + st.data[st.top];
24     }
25
26     if (total > 30) {
27         return total - 30;
28     }
29
30     return total;
31 }

```

3.1.2 输出

```

1  Program (1)
2      ExtDefList (1)
3          ExtDef (1)
4              Specifier (1)
5                  StructSpecifier (1)
6                      STRUCT

```

```

7      OptTag (1)
8      ID: Stack
9      LC
10     DefList (2)
11     Def (2)
12     Specifier (2)
13     TYPE: int
14     DecList (2)
15     Dec (2)
16     VarDec (2)
17     VarDec (2)
18     ID: data
19     LB
20     INT: 10
21     RB
22     SEMI
23     DefList (3)
24     Def (3)
25     Specifier (3)
26     TYPE: int
27     DecList (3)
28     Dec (3)
29     VarDec (3)
30     ID: top
31     SEMI
32     RC
33     SEMI
34 ExtDefList (6)
35 ExtDef (6)
36 Specifier (6)
37 TYPE: int
38 FunDec (6)
39 ID: main
40 LP
41 RP
42 CompSt (6)
43 LC

```

```

44      DefList (7)
45          Def (7)
46              Specifier (7)
47                  StructSpecifier (7)
48                      STRUCT
49                      Tag (7)
50                      ID: Stack
51      DecList (7)
52          Dec (7)
53          VarDec (7)
54              ID: st
55      SEMI
56      DefList (8)
57          Def (8)
58              Specifier (8)
59                  TYPE: int
60          DecList (8)
61              Dec (8)
62              VarDec (8)
63                  ID: i
64      SEMI
65      DefList (9)
66          Def (9)
67              Specifier (9)
68                  TYPE: int
69          DecList (9)
70              Dec (9)
71              VarDec (9)
72                  ID: total
73      SEMI
74      StmtList (11)
75          Stmt (11)
76              Exp (11)
77                  Exp (11)
78                      Exp (11)
79                          ID: st
80                          DOT

```

81	ID: top
82	ASSIGNOP
83	Exp (11)
84	INT: 0
85	SEMI
86	StmtList (12)
87	Stmt (12)
88	Exp (12)
89	Exp (12)
90	ID: i
91	ASSIGNOP
92	Exp (12)
93	INT: 0
94	SEMI
95	StmtList (13)
96	Stmt (13)
97	Exp (13)
98	Exp (13)
99	ID: total
100	ASSIGNOP
101	Exp (13)
102	INT: 0
103	SEMI
104	StmtList (15)
105	Stmt (15)
106	WHILE
107	LP
108	Exp (15)
109	Exp (15)
110	ID: i
111	RELOP
112	Exp (15)
113	INT: 5
114	RP
115	Stmt (15)
116	CompSt (15)
117	LC

118	StmtList (16)
119	Stmt (16)
120	Exp (16)
121	Exp (16)
122	Exp (16)
123	Exp (16)
124	ID: st
125	DOT
126	ID: data
127	LB
128	Exp (16)
129	Exp (16)
130	ID: st
131	DOT
132	ID: top
133	RB
134	ASSIGNOP
135	Exp (16)
136	Exp (16)
137	ID: i
138	STAR
139	Exp (16)
140	ID: i
141	SEMI
142	StmtList (17)
143	Stmt (17)
144	Exp (17)
145	Exp (17)
146	Exp (17)
147	ID: st
148	DOT
149	ID: top
150	ASSIGNOP
151	Exp (17)
152	Exp (17)
153	Exp (17)
154	ID: st

155	DOT
156	ID: top
157	PLUS
158	Exp (17)
159	INT: 1
160	SEMI
161	StmtList (18)
162	Stmt (18)
163	Exp (18)
164	Exp (18)
165	ID: i
166	ASSIGNOP
167	Exp (18)
168	Exp (18)
169	ID: i
170	PLUS
171	Exp (18)
172	INT: 1
173	SEMI
174	RC
175	StmtList (21)
176	Stmt (21)
177	WHILE
178	LP
179	Exp (21)
180	Exp (21)
181	Exp (21)
182	ID: st
183	DOT
184	ID: top
185	RELOP
186	Exp (21)
187	INT: 0
188	RP
189	Stmt (21)
190	CompSt (21)
191	LC

192	StmtList (22)
193	Stmt (22)
194	Exp (22)
195	Exp (22)
196	Exp (22)
197	ID: st
198	DOT
199	ID: top
200	ASSIGNOP
201	Exp (22)
202	Exp (22)
203	Exp (22)
204	ID: st
205	DOT
206	ID: top
207	MINUS
208	Exp (22)
209	INT: 1
210	SEMI
211	StmtList (23)
212	Stmt (23)
213	Exp (23)
214	Exp (23)
215	ID: total
216	ASSIGNOP
217	Exp (23)
218	Exp (23)
219	ID: total
220	PLUS
221	Exp (23)
222	Exp (23)
223	Exp (23)
224	ID: st
225	DOT
226	ID: data
227	LB
228	Exp (23)

229		Exp (23)
230		ID: st
231		DOT
232		ID: top
233		RB
234		SEMI
235	RC	
236	StmtList (26)	
237	Stmt (26)	
238	IF	
239	LP	
240	Exp (26)	
241	Exp (26)	
242	ID: total	
243	RELOP	
244	Exp (26)	
245	INT: 30	
246	RP	
247	Stmt (26)	
248	CompSt (26)	
249	LC	
250	StmtList (27)	
251	Stmt (27)	
252	RETURN	
253	Exp (27)	
254	Exp (27)	
255	ID: total	
256	MINUS	
257	Exp (27)	
258	INT: 30	
259	SEMI	
260	RC	
261	StmtList (30)	
262	Stmt (30)	
263	RETURN	
264	Exp (30)	
265	ID: total	

266

SEMI

267

RC

3.1.3 说明

含结构体、数组、双重 while、if-else 的合法程序

3.2 C-2

3.2.1 输入

```
1  int fib[20];
2
3  int build_fib(int n) {
4      int i;
5      fib[0] = 0;
6      fib[1] = 1;
7      i = 2;
8      while (i < n) {
9          fib[i] = fib[i - 1] + fib[i - 2];
10         i = i + 1;
11     }
12 }
13
14 int sum_even(int n) {
15     int i;
16     int total;
17     i = 0;
18     total = 0;
19     while (i < n) {
20         if ((fib[i] - (fib[i] / 2) * 2) == 0) {
21             total = total + fib[i];
22         }
23         i = i + 1;
24     }
25     return total;
26 }
27
28 int main() {
```

```

29  int n;
30  int result;
31  n = 15;
32  build_fib(n);
33  result = sum_even(n);
34  return result;
35  }

```

3.2.2 输出

```

1  Program (1)
2    ExtDefList (1)
3      ExtDef (1)
4        Specifier (1)
5          TYPE: int
6        ExtDecList (1)
7          VarDec (1)
8            VarDec (1)
9              ID: fib
10             LB
11             INT: 20
12             RB
13          SEMI
14        ExtDefList (3)
15          ExtDef (3)
16            Specifier (3)
17              TYPE: int
18            FunDec (3)
19              ID: build_fib
20              LP
21              VarList (3)
22                ParamDec (3)
23                  Specifier (3)
24                    TYPE: int
25                  VarDec (3)
26                    ID: n
27              RP

```

```

28      CompSt (3)
29      LC
30      DefList (4)
31      Def (4)
32      Specifier (4)
33      TYPE: int
34      DeclList (4)
35      Dec (4)
36      VarDec (4)
37      ID: i
38      SEMI
39      StmtList (5)
40      Stmt (5)
41      Exp (5)
42      Exp (5)
43      Exp (5)
44      ID: fib
45      LB
46      Exp (5)
47      INT: 0
48      RB
49      ASSIGNOP
50      Exp (5)
51      INT: 0
52      SEMI
53      StmtList (6)
54      Stmt (6)
55      Exp (6)
56      Exp (6)
57      Exp (6)
58      ID: fib
59      LB
60      Exp (6)
61      INT: 1
62      RB
63      ASSIGNOP
64      Exp (6)

```

65	INT: 1
66	SEMI
67	StmtList (7)
68	Stmt (7)
69	Exp (7)
70	Exp (7)
71	ID: i
72	ASSIGNOP
73	Exp (7)
74	INT: 2
75	SEMI
76	StmtList (8)
77	Stmt (8)
78	WHILE
79	LP
80	Exp (8)
81	Exp (8)
82	ID: i
83	RELOP
84	Exp (8)
85	ID: n
86	RP
87	Stmt (8)
88	CompSt (8)
89	LC
90	StmtList (9)
91	Stmt (9)
92	Exp (9)
93	Exp (9)
94	Exp (9)
95	ID: fib
96	LB
97	Exp (9)
98	ID: i
99	RB
100	ASSIGNOP
101	Exp (9)

102	Exp (9)
103	Exp (9)
104	ID: fib
105	LB
106	Exp (9)
107	Exp (9)
108	ID: i
109	MINUS
110	Exp (9)
111	INT: 1
112	RB
113	PLUS
114	Exp (9)
115	Exp (9)
116	ID: fib
117	LB
118	Exp (9)
119	Exp (9)
120	ID: i
121	MINUS
122	Exp (9)
123	INT: 2
124	RB
125	SEMI
126	StmtList (10)
127	Stmt (10)
128	Exp (10)
129	Exp (10)
130	ID: i
131	ASSIGNOP
132	Exp (10)
133	Exp (10)
134	ID: i
135	PLUS
136	Exp (10)
137	INT: 1
138	SEMI

```

139                                     RC
140
141     ExtDefList (14)
142     ExtDef (14)
143         Specifier (14)
144             TYPE: int
145     FunDec (14)
146         ID: sum_even
147         LP
148         VarList (14)
149             ParamDec (14)
150                 Specifier (14)
151                     TYPE: int
152                 VarDec (14)
153                     ID: n
154         RP
155     CompSt (14)
156         LC
157         DefList (15)
158             Def (15)
159                 Specifier (15)
160                     TYPE: int
161                 DecList (15)
162                     Dec (15)
163                     VarDec (15)
164                         ID: i
165             SEMI
166             DefList (16)
167                 Def (16)
168                     Specifier (16)
169                         TYPE: int
170                     DecList (16)
171                         Dec (16)
172                         VarDec (16)
173                             ID: total
174             SEMI
175     StmtList (17)

```

176	Stmt (17)
177	Exp (17)
178	Exp (17)
179	ID: i
180	ASSIGNOP
181	Exp (17)
182	INT: 0
183	SEMI
184	StmtList (18)
185	Stmt (18)
186	Exp (18)
187	Exp (18)
188	ID: total
189	ASSIGNOP
190	Exp (18)
191	INT: 0
192	SEMI
193	StmtList (19)
194	Stmt (19)
195	WHILE
196	LP
197	Exp (19)
198	Exp (19)
199	ID: i
200	RELOP
201	Exp (19)
202	ID: n
203	RP
204	Stmt (19)
205	CompSt (19)
206	LC
207	StmtList (20)
208	Stmt (20)
209	IF
210	LP
211	Exp (20)
212	Exp (20)

213	LP
214	Exp (20)
215	Exp (20)
216	Exp (20)
217	ID: fib
218	LB
219	Exp (20)
220	ID: i
221	RB
222	MINUS
223	Exp (20)
224	Exp (20)
225	LP
226	Exp (20)
227	Exp (20)
228	Exp (20)
229	ID: fib
230	LB
231	Exp (20)
232	ID: i
233	RB
234	DIV
235	Exp (20)
236	INT: 2
237	RP
238	STAR
239	Exp (20)
240	INT: 2
241	RP
242	RELOP
243	Exp (20)
244	INT: 0
245	RP
246	Stmt (20)
247	CompSt (20)
248	LC
249	StmtList (21)

250	Stmt (21)
251	Exp (21)
252	Exp (21)
253	ID: total
254	ASSIGNOP
255	Exp (21)
256	Exp (21)
257	ID: total
258	PLUS
259	Exp (21)
260	Exp (21)
261	ID: fib
262	LB
263	Exp (21)
264	ID: i
265	RB
266	SEMI
267	RC
268	StmtList (23)
269	Stmt (23)
270	Exp (23)
271	Exp (23)
272	ID: i
273	ASSIGNOP
274	Exp (23)
275	Exp (23)
276	ID: i
277	PLUS
278	Exp (23)
279	INT: 1
280	SEMI
281	RC
282	StmtList (25)
283	Stmt (25)
284	RETURN
285	Exp (25)
286	ID: total

287	SEMI
288	RC
289	ExtDefList (28)
290	ExtDef (28)
291	Specifier (28)
292	TYPE: int
293	FunDec (28)
294	ID: main
295	LP
296	RP
297	CompSt (28)
298	LC
299	DefList (29)
300	Def (29)
301	Specifier (29)
302	TYPE: int
303	DecList (29)
304	Dec (29)
305	VarDec (29)
306	ID: n
307	SEMI
308	DefList (30)
309	Def (30)
310	Specifier (30)
311	TYPE: int
312	DecList (30)
313	Dec (30)
314	VarDec (30)
315	ID: result
316	SEMI
317	StmtList (31)
318	Stmt (31)
319	Exp (31)
320	Exp (31)
321	ID: n
322	ASSIGNOP
323	Exp (31)

```

324         INT: 15
325     SEMI
326 StmtList (32)
327     Stmt (32)
328     Exp (32)
329         ID: build_fib
330     LP
331     Args (32)
332     Exp (32)
333         ID: n
334     RP
335 SEMI
336 StmtList (33)
337     Stmt (33)
338     Exp (33)
339     Exp (33)
340         ID: result
341     ASSIGNOP
342     Exp (33)
343         ID: sum_even
344     LP
345     Args (33)
346     Exp (33)
347         ID: n
348     RP
349 SEMI
350 StmtList (34)
351     Stmt (34)
352     RETURN
353     Exp (34)
354         ID: result
355 SEMI
356 RC

```

3.2.3 说明

含全局数组、多函数、复杂下标表达式的合法程序 (Fibonacci)

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；其他分组的同学只需要在对应位置提示错误即可，如果打印了语法树，则将视为违规，将会倒扣分。

4.1 D-1

4.1.1 输入

```
1 int main() {  
2     int a = 0xFF00;  
3     int b = 0x0F0F;  
4     int c = 0644;  
5     int d = 0x3A;  
6     return 0;  
7 }
```

4.1.2 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 (1)  
11        LC  
12      DefList (2)  
13        Def (2)  
14          Specifier (2)  
15            TYPE: int  
16          DecList (2)  
17            Dec (2)  
18              VarDec (2)  
19                ID: a
```

```

20         ASSIGNOP
21         Exp (2)
22         INT: 65280
23     SEMI
24 DefList (3)
25     Def (3)
26         Specifier (3)
27         TYPE: int
28         DecList (3)
29         Dec (3)
30         VarDec (3)
31         ID: b
32         ASSIGNOP
33         Exp (3)
34         INT: 3855
35     SEMI
36 DefList (4)
37     Def (4)
38         Specifier (4)
39         TYPE: int
40         DecList (4)
41         Dec (4)
42         VarDec (4)
43         ID: c
44         ASSIGNOP
45         Exp (4)
46         INT: 420
47     SEMI
48 DefList (5)
49     Def (5)
50         Specifier (5)
51         TYPE: int
52         DecList (5)
53         Dec (5)
54         VarDec (5)
55         ID: d
56         ASSIGNOP

```

```

57             Exp (5)
58             INT: 58
59         SEMI
60     StmtList (6)
61     Stmt (6)
62     RETURN
63     Exp (6)
64     INT: 0
65     SEMI
66 RC

```

4.1.3 说明

第 2 5 行分别赋值十六进制（0xFF00、0x0F0F、0x3A）和八进制（0644）整数：- 分组 0、1（实现了十六/八进制支持）：程序完全合法，应输出语法树 - 分组 2、3（未实现十六/八进制）：每行应各报告 1 个错误（类型 A 或 B 均可），共 4 个错误，分别位于第 2、3、4、5 行

4.2 D-2

4.2.1 输入

```

1 int main() {
2     float p = 3.14e0;
3     float q = 6.02e23;
4     float r = 1.38e-2;
5     float s = 2.72E1;
6     return 0;
7 }

```

4.2.2 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 (1)
11     LC
12     DefList (2)
13     Def (2)
14         Specifier (2)
15             TYPE: float
16         DecList (2)
17             Dec (2)
18                 VarDec (2)
19                     ID: p
20                     ASSIGNOP
21                     Exp (2)
22                         FLOAT: 3.140000
23     SEMI
24     DefList (3)
25     Def (3)
26         Specifier (3)
27             TYPE: float
28         DecList (3)
29             Dec (3)
30                 VarDec (3)
31                     ID: q
32                     ASSIGNOP
33                     Exp (3)
34                         FLOAT: 602000017271895229464576.000000
35     SEMI
36     DefList (4)
37     Def (4)
38         Specifier (4)
39             TYPE: float
40         DecList (4)
41             Dec (4)
42                 VarDec (4)
43                     ID: r
44                     ASSIGNOP

```



```

45             Exp (4)
46             FLOAT: 0.013800
47         SEMI
48     DefList (5)
49         Def (5)
50             Specifier (5)
51             TYPE: float
52             DeclList (5)
53                 Dec (5)
54                     VarDec (5)
55                     ID: s
56                     ASSIGNOP
57                     Exp (5)
58                     FLOAT: 27.200001
59             SEMI
60 StmtList (6)
61     Stmt (6)
62         RETURN
63         Exp (6)
64         INT: 0
65         SEMI
66 RC

```

4.2.3 说明

第 2 5 行分别赋值科学计数法浮点数 (3.14e0、6.02e23、1.38e-2、2.72E1)：- 分组 0、2 (实现了科学计数法支持)：程序完全合法，应输出语法树 - 分组 1、3 (未实现科学计数法)：每行应各报告 1 个错误 (类型 A 或 B 均可)，共 4 个错误，分别位于第 2、3、4、5 行

4.3 D-3

4.3.1 输入

```

1  int main() {
2      int x /* counter */ = 0;
3      x = x + 1;
4      if /* check */ (x > 0) {
5          return x;
6      }

```

```
7   return 0;
8 }
```

4.3.2 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 (1)
11        LC
12      DefList (2)
13        Def (2)
14          Specifier (2)
15            TYPE: int
16          DecList (2)
17            Dec (2)
18              VarDec (2)
19                ID: x
20                ASSIGNOP
21                Exp (2)
22                  INT: 0
23          SEMI
24      StmtList (3)
25        Stmt (3)
26          Exp (3)
27            Exp (3)
28              ID: x
29              ASSIGNOP
30              Exp (3)
31                Exp (3)
32                  ID: x
```

33	PLUS
34	Exp (3)
35	INT: 1
36	SEMI
37	StmtList (4)
38	Stmt (4)
39	IF
40	LP
41	Exp (4)
42	Exp (4)
43	ID: x
44	RELOP
45	Exp (4)
46	INT: 0
47	RP
48	Stmt (4)
49	CompSt (4)
50	LC
51	StmtList (5)
52	Stmt (5)
53	RETURN
54	Exp (5)
55	ID: x
56	SEMI
57	RC
58	StmtList (7)
59	Stmt (7)
60	RETURN
61	Exp (7)
62	INT: 0
63	SEMI
64	RC

4.3.3 说明

第 2 行 ‘int x /* counter */ = 0;’ 和第 4 行 ‘if /* check */ (x > 0)’ 中将块注释 ‘/* */’ 插入到相邻 token 之间：- 分组 0、3（注释处理实现完善）：注释被词法分析器正确跳过，程序合法，输出语法树 - 分组 1、2（注释处理存在缺陷）：行内块注释导致解析出错，在第 2 行和第 4 行（或第 5

行) 各产生 1 个错误 (A/B 类型均可)

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。其中:

- E1-x 针对 1.1 分组的同学。
- E2-x 针对 1.2 分组的同学。
- E3-x 针对 1.3 分组的同学。

5.1 E1-1

5.1.1 输入

```
1  int check_range(int addr) {
2      int lo = 0x0000;
3      int hi = 0xFFFF;
4      if (addr >= lo) {
5          if (addr <= hi) {
6              return 0755;
7          }
8      }
9      return 0x00;
10 }
11
12 int main() {
13     int base = 0x1A00;
14     int code = check_range(base);
15     return code;
16 }
```

5.1.2 输出

```
1 Program (1)
2   ExtDefList (1)
3     ExtDef (1)
4       Specifier (1)
5         TYPE: int
```

```

6      FunDec (1)
7          ID: check_range
8          LP
9          VarList (1)
10             ParamDec (1)
11                 Specifier (1)
12                     TYPE: int
13             VarDec (1)
14                 ID: addr
15             RP
16      CompSt (1)
17          LC
18          DefList (2)
19              Def (2)
20                  Specifier (2)
21                      TYPE: int
22                  DeclList (2)
23                      Dec (2)
24                          VarDec (2)
25                              ID: lo
26                              ASSIGNOP
27                              Exp (2)
28                                  INT: 0
29              SEMI
30          DefList (3)
31              Def (3)
32                  Specifier (3)
33                      TYPE: int
34                  DeclList (3)
35                      Dec (3)
36                          VarDec (3)
37                              ID: hi
38                              ASSIGNOP
39                              Exp (3)
40                                  INT: 65535
41              SEMI
42      StmtList (4)

```

```

43      Stmt (4)
44      IF
45      LP
46      Exp (4)
47      Exp (4)
48      ID: addr
49      RELOP
50      Exp (4)
51      ID: lo
52      RP
53      Stmt (4)
54      CompSt (4)
55      LC
56      StmtList (5)
57      Stmt (5)
58      IF
59      LP
60      Exp (5)
61      Exp (5)
62      ID: addr
63      RELOP
64      Exp (5)
65      ID: hi
66      RP
67      Stmt (5)
68      CompSt (5)
69      LC
70      StmtList (6)
71      Stmt (6)
72      RETURN
73      Exp (6)
74      INT: 493
75      SEMI
76      RC
77      RC
78      StmtList (9)
79      Stmt (9)

```

```

80             RETURN
81             Exp (9)
82             INT: 0
83             SEMI
84         RC
85     ExtDefList (12)
86     ExtDef (12)
87     Specifier (12)
88     TYPE: int
89     FunDec (12)
90     ID: main
91     LP
92     RP
93     CompSt (12)
94     LC
95     DefList (13)
96     Def (13)
97     Specifier (13)
98     TYPE: int
99     DeclList (13)
100    Dec (13)
101    VarDec (13)
102    ID: base
103    ASSIGNOP
104    Exp (13)
105    INT: 6656
106    SEMI
107    DefList (14)
108    Def (14)
109    Specifier (14)
110    TYPE: int
111    DeclList (14)
112    Dec (14)
113    VarDec (14)
114    ID: code
115    ASSIGNOP
116    Exp (14)

```

```

117             ID: check_range
118             LP
119             Args (14)
120             Exp (14)
121             ID: base
122             RP
123             SEMI
124         StmtList (15)
125             Stmt (15)
126             RETURN
127             Exp (15)
128             ID: code
129             SEMI
130         RC

```

5.1.3 说明

使用十六进制地址（0x0000、0xFFFF、0x1A00）和八进制权限码（0755、0x00）进行范围检查。

5.2 E1-2

5.2.1 输入

```

1  int main() {
2      int a = 0;
3      int b = 0xQQQ;
4      int c = 099;
5      return a + b + c;
6  }

```

5.2.2 输出

```

1  Error type B at Line 3: Syntax error near 'Q'
2  Error type B at Line 4: Syntax error near '9'

```


5.2.3 说明

程序包含两处针对分组 1 实现的非法字面量：Line 3: ‘int b = 0xQQQ;’ —十六进制前缀后含非法字符 Q，不是合法十六进制数 Line 4: ‘int c = 099;’ —八进制前缀后含非法数字 9（八进制只允许 0-7）应精确报告 2 个错误，分别位于第 3 行和第 4 行，类型 A 或 B 均可。

5.3 E2-1

5.3.1 输入

```
1 int main() {
2     float mass = 9.1e-2;
3     float speed = 2.99e8;
4     float gravity = 9.8E0;
5     float energy = 1.23e-4;
6     float result;
7     result = mass + speed + gravity + energy;
8     return 0;
9 }
```

5.3.2 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 (1)
11        LC
12      DefList (2)
13        Def (2)
14          Specifier (2)
15            TYPE: float
16          DeclList (2)
17            Dec (2)
```

```

18         VarDec (2)
19             ID: mass
20         ASSIGNOP
21         Exp (2)
22             FLOAT: 0.091000
23     SEMI
24 DefList (3)
25     Def (3)
26         Specifier (3)
27             TYPE: float
28         DecList (3)
29             Dec (3)
30                 VarDec (3)
31                     ID: speed
32                 ASSIGNOP
33                 Exp (3)
34                     FLOAT: 299000000.000000
35     SEMI
36 DefList (4)
37     Def (4)
38         Specifier (4)
39             TYPE: float
40         DecList (4)
41             Dec (4)
42                 VarDec (4)
43                     ID: gravity
44                 ASSIGNOP
45                 Exp (4)
46                     FLOAT: 9.800000
47     SEMI
48 DefList (5)
49     Def (5)
50         Specifier (5)
51             TYPE: float
52         DecList (5)
53             Dec (5)
54                 VarDec (5)

```

```

55         ID: energy
56     ASSIGNOP
57     Exp (5)
58         FLOAT: 0.000123
59     SEMI
60     DefList (6)
61         Def (6)
62             Specifier (6)
63                 TYPE: float
64             DecList (6)
65                 Dec (6)
66                 VarDec (6)
67                     ID: result
68     SEMI
69 StmtList (7)
70     Stmt (7)
71         Exp (7)
72             Exp (7)
73                 ID: result
74     ASSIGNOP
75     Exp (7)
76         Exp (7)
77             Exp (7)
78                 Exp (7)
79                     ID: mass
80                 PLUS
81                 Exp (7)
82                     ID: speed
83                 PLUS
84                 Exp (7)
85                     ID: gravity
86                 PLUS
87                 Exp (7)
88                     ID: energy
89     SEMI
90 StmtList (8)
91     Stmt (8)

```

92	RETURN
93	Exp (8)
94	INT: 0
95	SEMI
96	RC

5.3.3 说明

程序使用多种科学计数法浮点数：9.1e-2（小负指数）、2.99e8（大正指数）、9.8E0（大写 E 零指数）、1.23e-4（小负指数）

5.4 E2-2

5.4.1 输入

```
1 int main() {
2     float a = 0.0;
3     float b = 2.5e;
4     float c = 1.0E+;
5     return 0;
6 }
```

5.4.2 输出

```
1 Error type B at Line 3: Syntax error near ';'
2 Error type B at Line 4: Syntax error near ';'

```

5.4.3 说明

程序包含两处针对分组 2 实现的非法科学计数法字面量：Line 3: ‘float b = 2.5e;’ 一科学计数法指数部分缺失（e 后无数字）Line 4: ‘float c = 1.0E+;’ 一科学计数法指数部分缺失（E+ 后无数字）应精确报告 2 个错误，分别位于第 3 行和第 4 行，类型 A 或 B 均可。

5.5 E3-1

5.5.1 输入

```
1 /* Matrix utility: supports init and trace */
2 /* All values default to zero on init */
3
```

```

4  int mat[3][3];
5
6  /* Set all elements to val */
7  int init_mat(int val) {
8      int i;
9      int j;
10     i = 0;
11     while (i < 3) {
12         j = 0;
13         while (j < 3) {
14             mat[i][j] = val; /* assign element */
15             j = j + 1;
16         }
17         i = i + 1;
18     }
19 }
20
21 int main() {
22     init_mat(0);          /* zero out matrix */
23     mat[1][1] = 1;        /* set diagonal */
24     return mat[1][1];     // return center value
25 }

```

5.5.2 输出

```

1  Program (4)
2      ExtDefList (4)
3          ExtDef (4)
4              Specifier (4)
5                  TYPE: int
6              ExtDecList (4)
7                  VarDec (4)
8                      VarDec (4)
9                          VarDec (4)
10                              ID: mat
11                              LB
12                              INT: 3

```

```

13         RB
14     LB
15     INT: 3
16     RB
17     SEMI
18 ExtDefList (7)
19     ExtDef (7)
20         Specifier (7)
21             TYPE: int
22     FunDec (7)
23         ID: init_mat
24         LP
25         VarList (7)
26             ParamDec (7)
27                 Specifier (7)
28                     TYPE: int
29                 VarDec (7)
30                     ID: val
31     RP
32 CompSt (7)
33     LC
34     DefList (8)
35         Def (8)
36             Specifier (8)
37                 TYPE: int
38             DecList (8)
39                 Dec (8)
40                 VarDec (8)
41                     ID: i
42     SEMI
43     DefList (9)
44         Def (9)
45             Specifier (9)
46                 TYPE: int
47             DecList (9)
48                 Dec (9)
49                 VarDec (9)

```

50	ID: j
51	SEMI
52	StmtList (10)
53	Stmt (10)
54	Exp (10)
55	Exp (10)
56	ID: i
57	ASSIGNOP
58	Exp (10)
59	INT: 0
60	SEMI
61	StmtList (11)
62	Stmt (11)
63	WHILE
64	LP
65	Exp (11)
66	Exp (11)
67	ID: i
68	RELOP
69	Exp (11)
70	INT: 3
71	RP
72	Stmt (11)
73	CompSt (11)
74	LC
75	StmtList (12)
76	Stmt (12)
77	Exp (12)
78	Exp (12)
79	ID: j
80	ASSIGNOP
81	Exp (12)
82	INT: 0
83	SEMI
84	StmtList (13)
85	Stmt (13)
86	WHILE

87	LP
88	Exp (13)
89	Exp (13)
90	ID: j
91	RELOP
92	Exp (13)
93	INT: 3
94	RP
95	Stmt (13)
96	CompSt (13)
97	LC
98	StmtList (14)
99	Stmt (14)
100	Exp (14)
101	Exp (14)
102	Exp (14)
103	Exp (14)
104	ID: mat
105	LB
106	Exp (14)
107	ID: i
108	RB
109	LB
110	Exp (14)
111	ID: j
112	RB
113	ASSIGNOP
114	Exp (14)
115	ID: val
116	SEMI
117	StmtList (15)
118	Stmt (15)
119	Exp (15)
120	Exp (15)
121	ID: j
122	ASSIGNOP
123	Exp (15)


```

124                                     Exp (15)
125                                     ID: j
126                                     PLUS
127                                     Exp (15)
128                                     INT: 1
129                                     SEMI
130                                     RC
131                               StmtList (17)
132                               Stmt (17)
133                               Exp (17)
134                               Exp (17)
135                               ID: i
136                               ASSIGNOP
137                               Exp (17)
138                               Exp (17)
139                               ID: i
140                               PLUS
141                               Exp (17)
142                               INT: 1
143                               SEMI
144                               RC
145                               RC
146       ExtDefList (21)
147       ExtDef (21)
148       Specifier (21)
149       TYPE: int
150       FunDec (21)
151       ID: main
152       LP
153       RP
154       CompSt (21)
155       LC
156       StmtList (22)
157       Stmt (22)
158       Exp (22)
159       ID: init_mat
160       LP

```

161	Args (22)
162	Exp (22)
163	INT: 0
164	RP
165	SEMI
166	StmtList (23)
167	Stmt (23)
168	Exp (23)
169	Exp (23)
170	Exp (23)
171	Exp (23)
172	ID: mat
173	LB
174	Exp (23)
175	INT: 1
176	RB
177	LB
178	Exp (23)
179	INT: 1
180	RB
181	ASSIGNOP
182	Exp (23)
183	INT: 1
184	SEMI
185	StmtList (24)
186	Stmt (24)
187	RETURN
188	Exp (24)
189	Exp (24)
190	Exp (24)
191	ID: mat
192	LB
193	Exp (24)
194	INT: 1
195	RB
196	LB
197	Exp (24)

198	INT: 1
199	RB
200	SEMI
201	RC

5.5.3 说明

程序覆盖以下合法注释场景：- 文件顶部多行独立块注释（/* ... */）- 函数前独立块注释说明 - 代码行末尾行内块注释（mat[i][j] = val; /* assign element */）- 代码行末尾单行注释（return mat[1][1]; // return center value）- 函数调用行末尾块注释（init_mat(0); /* zero out matrix */）

5.6 E3-2

5.6.1 输入

```

1  /*/* 嵌套注释不被支持，此处为错误示例 */*/
2
3  int main() {
4      int x = 1;
5      return x;
6  }
```

5.6.2 输出

```

1  Error type B at Line 1: unexpected right comment '*/'.
```

5.6.3 说明

第 1 行 ‘/*/* 嵌套注释不被支持，此处为错误示例 */*/’ 尝试嵌套块注释，应在第 1 行报告语法错误（类型 B）

6 结束语

若对本文档有任何疑问，可写邮件与陈泰霖助教联系，注意同时抄送给许畅老师。