

## 编译原理第一次实验测试用例：目录

<b>1</b>	<b>A 组测试用例</b>	<b>2</b>
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
<b>2</b>	<b>B 组测试用例</b>	<b>7</b>
2.1	B-1 . . . . .	7
2.2	B-2 . . . . .	9
<b>3</b>	<b>C 组测试用例</b>	<b>10</b>
3.1	C-1 . . . . .	10
3.2	C-2 . . . . .	27
<b>4</b>	<b>D 组测试用例</b>	<b>40</b>
4.1	D-1 . . . . .	40
4.2	D-2 . . . . .	43
4.3	D-3 . . . . .	45
<b>5</b>	<b>E 组测试用例</b>	<b>47</b>
5.1	E1-1 . . . . .	47
5.2	E1-2 . . . . .	50
5.3	E2-1 . . . . .	51
5.4	E2-2 . . . . .	53
5.5	E3-1 . . . . .	54
5.6	E3-2 . . . . .	56
<b>6</b>	<b>结束语</b>	<b>56</b>

## 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 a = 1;  
3     int b = a#2;  
4 }
```

#### 1.1.2 输出

```
1 Error type A at Line 3: Mysterious characters '#'.  

```

#### 1.1.3 说明

未定义的字符。(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 i = 1;  
3     int 1i = 10 + i;  
4 }
```

#### 1.2.2 输出

```
1 Error type A at Line 3: Illegal ID '1i'.  

```

#### 1.2.3 说明

标识符不能以数字开头。(注：也可以识别为 B 类错误。)

## 1.3 A-3

### 1.3.1 输入

```
1 struct Vector3D {  
2     float x;  
3     float y;  
4     float z;  
5 };  
6  
7 int main() {  
8     Vector3D v;  
9     v.x = 1.5;  
10    v.y = -0.3;  
11    v.z = v.x * 2.0 + v.y;  
12 }
```

### 1.3.2 输出

```
1 Error type B at line 8: Syntax error near 'v'.
```

### 1.3.3 说明

缺少 struct 关键字。

## 1.4 A-4

### 1.4.1 输入

```
1 int matrix[3][3];  
2 int scale = 2;  
3  
4 int main() {  
5     int local = scale * matrix[1][1];  
6     matrix[1][1] = 10;  
7 }
```

### 1.4.2 输出

```
1 Error type B at Line 2: Variable assignment should not be done ahead  
  of a Program.
```

### 1.4.3 说明

全局变量定义时不能初始化。

## 1.5 A-5

### 1.5.1 输入

```
1 int main(){  
2     float f = 1.2.3.4;  
3 }
```

### 1.5.2 输出

```
1 Error type A at Line 2: Illegal FLOAT number '1.2.3.4'.
```

### 1.5.3 说明

错误的浮点数。(注：也可以识别为 B 类错误。)

## 1.6 A-6

### 1.6.1 输入

```
1 struct Point {  
2     float x;  
3     float y;  
4 } points[5.5];  
5  
6 int main() {  
7     points[0].x = 3.14;  
8     points[0].y = points[0].x * 2.0;  
9     return 0;  
10 }
```

### 1.6.2 输出

```
1 Error type B at Line 4: Invalid Array declaration.
```

### 1.6.3 说明

数组的长度必须为整数类型。

## 1.7 A-7

### 1.7.1 输入

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

### 1.7.2 输出

```
1 Error type B at Line 5: Invalid function call.
```

### 1.7.3 说明

逗号后缺少实参。

## 1.8 A-8

### 1.8.1 输入

```
1 struct Data {  
2     int values[10];  
3 };  
4  
5 int main() {  
6     struct Data d;  
7     int i = 0;
```

```

8   while (i < 10) {
9       d.values[i] -= 2;
10      i = i + 1;
11  }
12  return 0;
13 }

```

## 1.8.2 输出

```

1 Error type B at Line 9: '-' is not supported.

```

## 1.8.3 说明

C-不支持 ‘-’

## 1.9 A-9

### 1.9.1 输入

```

1 int main() {
2     int sum;
3     sum = 5 + 3;
4     int j;
5 }

```

### 1.9.2 输出

```

1 Error type B at Line 4: DefList should be put ahead of StmtList.

```

### 1.9.3 说明

语句块的开头才可以定义变量。

## 1.10 A-10

### 1.10.1 输入

```

1 int add(int a, int b) { return; }
2
3 int main() {
4     int x = add(3, 5);
5     return x;
6 }

```

### 1.10.2 输出

```

1 Error type B at Line 1: Invalid expression in RETURN Code block.

```

### 1.10.3 说明

return 语句缺少返回值。

## 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 Car {
2     int color;
3     float weight;
4 };
5
6 struct Fee {
7     struct vehicle;
8     int pass_count;
9     float toll;
10 };
11
12 int calculate(int f,) {
13     struct { int type; };
14     float base = 50.0;

```

```

15     float sensors[10] = {2.1};
16     struct Fee totalFee[10];
17     struct Car truck;
18     truck.weight = 3.8;
19     truck.color = 2;
20
21     while (truck.weight > 3.0);
22
23     while (truck.weight > 3.0) {
24         if (sensors[5] > 3.0) {
25             base = base + 1.111;
26         }
27
28         if (v == 2) {
29             base = base * 1.2;
30         }
31
32         truck.weight = truck.weight - 1.0;
33     }
34
35     if (base >= 100.0) {
36         return base;
37     }
38
39     return 0;
40 }

```

### 2.1.2 输出

```

1 Error type B at Line 7: syntax error, unexpected SEMI, expecting ID.
2 Error type B at Line 12: syntax error, unexpected RP, expecting TYPE
  or STRUCT.
3 Error type B at Line 13: syntax error, unexpected SEMI, expecting ID.
4 Error type B at Line 15: syntax error, unexpected LC.
5 Error type B at Line 21: syntax error, unexpected SEMI.

```



### 2.1.3 说明

第 7 行 `struct vehicle` 缺少结构名 `Car`；第 12 行函数逗号后缺少参数；第 13 行函数中不能单纯声明结构体；第 15 行不支持初始化列表；第 21 行不支持 `while` 循环体为空

## 2.2 B-2

### 2.2.1 输入

```
1 float grades[5];
2
3 float;
4
5 int find_max() {
6     float max = 0;
7     int first = 0;
8     while (first < 5) {
9         if (grades[first] > max) {
10             max = grades[first];
11         }
12         first++;
13     }
14     return max;
15 }
16
17 float calc_avg(int size) {
18     float total = 0.0;
19     if (total >= 0) {
20         return total / size;
21     }
22     return 0.0;
23 }
24
25 int main(void) {
26
27     int max_score = find_max(5);
28     float avg_score = calc_avg(5);
29
30     return 0;
```

31 }

### 2.2.2 输出

```
1 Error type B at Line 1: syntax error, unexpected RB, expecting SEMI.
2 Error type B at Line 12: syntax error, unexpected PLUS.
3 Error type B at Line 19: syntax error, unexpected ASSIGNOP.
4 Error type B at Line 25: syntax error, unexpected ID, expecting TYPE
  or RP or STRUCT.
```

### 2.2.3 说明

第 1 行多了']'; 第 12 行 '+' 后缺少 Exp; 第 19 行 '>' 和 '=' 中间多了空格; 第 25 行不合法的函数定义。

## 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 Point {
2     int x;
3     int y;
4 };
5
6 struct Circle {
7     struct Point center;
8     int radius;
9 };
10
11 int max_points;
12 int PI;
13
14 int distance_sq(struct Point p1, struct Point p2) {
15     int dx = p2.x - p1.x;
```

```

16     int dy = p2.y - p1.y;
17     return (dx * dx + dy * dy);
18 }
19
20 int cir_area(struct Circle c) {
21     return PI * c.radius * c.radius;
22 }
23
24 int main() {
25     int count;
26     int d_sq;
27     int areas[10];
28     struct Point points[2];
29     struct Circle my_cir;
30
31     max_points = 100;
32     PI = 3;
33     count = 0;
34
35     my_cir.center.x = points[0].x;
36     my_cir.center.y = points[0].y;
37     my_cir.radius = 5.0;
38
39     while (count < 10 && count <= max_points) {
40         areas[count] = cir_area(my_cir);
41         my_cir.radius = my_cir.radius + 1;
42         if ((count - (count / 2) * 2) == 0) {
43             areas[count] = count / 2;
44         } else {
45             areas[count] = count * 2;
46         }
47         count = count + 1;
48     }
49
50     if ((d_sq > 10000) || (d_sq < 2500)) {
51         return 1;
52     }

```

```

53
54     return d_sq;
55 }

```

### 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: Point
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: x
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: y
27                SEMI
28              RC
29            SEMI
30          ExtDefList (6)
31            ExtDef (6)

```

```

32     Specifier (6)
33     StructSpecifier (6)
34     STRUCT
35     OptTag (6)
36     ID: Circle
37     LC
38     DefList (7)
39     Def (7)
40     Specifier (7)
41     StructSpecifier (7)
42     STRUCT
43     Tag (7)
44     ID: Point
45     DecList (7)
46     Dec (7)
47     VarDec (7)
48     ID: center
49     SEMI
50     DefList (8)
51     Def (8)
52     Specifier (8)
53     TYPE: int
54     DecList (8)
55     Dec (8)
56     VarDec (8)
57     ID: radius
58     SEMI
59     RC
60     SEMI
61 ExtDefList (11)
62 ExtDef (11)
63     Specifier (11)
64     TYPE: int
65     ExtDecList (11)
66     VarDec (11)
67     ID: max_points
68     SEMI

```

```

69      ExtDefList (12)
70      ExtDef (12)
71      Specifier (12)
72      TYPE: int
73      ExtDecList (12)
74      VarDec (12)
75      ID: PI
76      SEMI
77      ExtDefList (14)
78      ExtDef (14)
79      Specifier (14)
80      TYPE: int
81      FunDec (14)
82      ID: distance_sq
83      LP
84      VarList (14)
85      ParamDec (14)
86      Specifier (14)
87      StructSpecifier (14)
88      STRUCT
89      Tag (14)
90      ID: Point
91      VarDec (14)
92      ID: p1
93      COMMA
94      VarList (14)
95      ParamDec (14)
96      Specifier (14)
97      StructSpecifier (14)
98      STRUCT
99      Tag (14)
100      ID: Point
101      VarDec (14)
102      ID: p2
103      RP
104      CompSt (14)
105      LC

```

```

106      DefList (15)
107      Def (15)
108          Specifier (15)
109              TYPE: int
110          DecList (15)
111          Dec (15)
112              VarDec (15)
113                  ID: dx
114                  ASSIGNOP
115                  Exp (15)
116                      Exp (15)
117                          Exp (15)
118                              ID: p2
119                              DOT
120                              ID: x
121                              MINUS
122                              Exp (15)
123                                  Exp (15)
124                                      ID: p1
125                                      DOT
126                                      ID: x
127      SEMI
128      DefList (16)
129      Def (16)
130          Specifier (16)
131              TYPE: int
132          DecList (16)
133          Dec (16)
134              VarDec (16)
135                  ID: dy
136                  ASSIGNOP
137                  Exp (16)
138                      Exp (16)
139                          Exp (16)
140                              ID: p2
141                              DOT
142                              ID: y

```

```

143             MINUS
144             Exp (16)
145             Exp (16)
146             ID: p1
147             DOT
148             ID: y
149             SEMI
150 StmtList (17)
151 Stmt (17)
152 RETURN
153 Exp (17)
154 LP
155 Exp (17)
156 Exp (17)
157 Exp (17)
158 ID: dx
159 STAR
160 Exp (17)
161 ID: dx
162 PLUS
163 Exp (17)
164 Exp (17)
165 ID: dy
166 STAR
167 Exp (17)
168 ID: dy
169 RP
170 SEMI
171 RC
172 ExtDefList (20)
173 ExtDef (20)
174 Specifier (20)
175 TYPE: int
176 FunDec (20)
177 ID: cir_area
178 LP
179 VarList (20)

```



```

180         ParamDec (20)
181         Specifier (20)
182         StructSpecifier (20)
183         STRUCT
184         Tag (20)
185         ID: Circle
186         VarDec (20)
187         ID: c
188     RP
189 CompSt (20)
190     LC
191     StmtList (21)
192     Stmt (21)
193     RETURN
194     Exp (21)
195     Exp (21)
196     Exp (21)
197     ID: PI
198     STAR
199     Exp (21)
200     Exp (21)
201     ID: c
202     DOT
203     ID: radius
204     STAR
205     Exp (21)
206     Exp (21)
207     ID: c
208     DOT
209     ID: radius
210     SEMI
211     RC
212 ExtDefList (24)
213     ExtDef (24)
214     Specifier (24)
215     TYPE: int
216     FunDec (24)

```

217	ID: main
218	LP
219	RP
220	CompSt (24)
221	LC
222	DefList (25)
223	Def (25)
224	Specifier (25)
225	TYPE: int
226	DecList (25)
227	Dec (25)
228	VarDec (25)
229	ID: count
230	SEMI
231	DefList (26)
232	Def (26)
233	Specifier (26)
234	TYPE: int
235	DecList (26)
236	Dec (26)
237	VarDec (26)
238	ID: d_sq
239	SEMI
240	DefList (27)
241	Def (27)
242	Specifier (27)
243	TYPE: int
244	DecList (27)
245	Dec (27)
246	VarDec (27)
247	VarDec (27)
248	ID: areas
249	LB
250	INT: 10
251	RB
252	SEMI
253	DefList (28)

```

254         Def (28)
255         Specifier (28)
256         StructSpecifier (28)
257         STRUCT
258         Tag (28)
259         ID: Point
260     DecList (28)
261     Dec (28)
262     VarDec (28)
263     VarDec (28)
264     ID: points
265     LB
266     INT: 2
267     RB
268     SEMI
269     DefList (29)
270     Def (29)
271     Specifier (29)
272     StructSpecifier (29)
273     STRUCT
274     Tag (29)
275     ID: Circle
276     DecList (29)
277     Dec (29)
278     VarDec (29)
279     ID: my_cir
280     SEMI
281 StmtList (31)
282     Stmt (31)
283     Exp (31)
284     Exp (31)
285     ID: max_points
286     ASSIGNOP
287     Exp (31)
288     INT: 100
289     SEMI
290 StmtList (32)

```

291	Stmt (32)
292	Exp (32)
293	Exp (32)
294	ID: PI
295	ASSIGNOP
296	Exp (32)
297	INT: 3
298	SEMI
299	StmtList (33)
300	Stmt (33)
301	Exp (33)
302	Exp (33)
303	ID: count
304	ASSIGNOP
305	Exp (33)
306	INT: 0
307	SEMI
308	StmtList (35)
309	Stmt (35)
310	Exp (35)
311	Exp (35)
312	Exp (35)
313	Exp (35)
314	ID: my_cir
315	DOT
316	ID: center
317	DOT
318	ID: x
319	ASSIGNOP
320	Exp (35)
321	Exp (35)
322	Exp (35)
323	ID: points
324	LB
325	Exp (35)
326	INT: 0
327	RB

328	DOT
329	ID: x
330	SEMI
331	StmtList (36)
332	Stmt (36)
333	Exp (36)
334	Exp (36)
335	Exp (36)
336	Exp (36)
337	ID: my_cir
338	DOT
339	ID: center
340	DOT
341	ID: y
342	ASSIGNOP
343	Exp (36)
344	Exp (36)
345	Exp (36)
346	ID: points
347	LB
348	Exp (36)
349	INT: 0
350	RB
351	DOT
352	ID: y
353	SEMI
354	StmtList (37)
355	Stmt (37)
356	Exp (37)
357	Exp (37)
358	Exp (37)
359	ID: my_cir
360	DOT
361	ID: radius
362	ASSIGNOP
363	Exp (37)
364	FLOAT: 5.000000

365	SEMI
366	StmtList (39)
367	Stmt (39)
368	WHILE
369	LP
370	Exp (39)
371	Exp (39)
372	Exp (39)
373	ID: count
374	RELOP
375	Exp (39)
376	INT: 10
377	AND
378	Exp (39)
379	Exp (39)
380	ID: count
381	RELOP
382	Exp (39)
383	ID: max_points
384	RP
385	Stmt (39)
386	CompSt (39)
387	LC
388	StmtList (40)
389	Stmt (40)
390	Exp (40)
391	Exp (40)
392	Exp (40)
393	ID: areas
394	LB
395	Exp (40)
396	ID: count
397	RB
398	ASSIGNOP
399	Exp (40)
400	ID: cir_area
401	LP

402	Args (40)
403	Exp (40)
404	ID: my_cir
405	RP
406	SEMI
407	StmtList (41)
408	Stmt (41)
409	Exp (41)
410	Exp (41)
411	Exp (41)
412	ID: my_cir
413	DOT
414	ID: radius
415	ASSIGNOP
416	Exp (41)
417	Exp (41)
418	Exp (41)
419	ID: my_cir
420	DOT
421	ID: radius
422	PLUS
423	Exp (41)
424	INT: 1
425	SEMI
426	StmtList (42)
427	Stmt (42)
428	IF
429	LP
430	Exp (42)
431	Exp (42)
432	LP
433	Exp (42)
434	Exp (42)
435	ID: count
436	MINUS
437	Exp (42)
438	Exp (42)

439	LP
440	Exp (42)
441	Exp (42)
442	ID: count
443	DIV
444	Exp (42)
445	INT: 2
446	RP
447	STAR
448	Exp (42)
449	INT: 2
450	RP
451	RELOP
452	Exp (42)
453	INT: 0
454	RP
455	Stmt (42)
456	CompSt (42)
457	LC
458	StmtList (43)
459	Stmt (43)
460	Exp (43)
461	Exp (43)
462	Exp (43)
463	ID: areas
464	LB
465	Exp (43)
466	ID: count
467	RB
468	ASSIGNOP
469	Exp (43)



470		Exp (43)
471	ID:	count
472		DIV
473		Exp (43)
474		INT: 2
475		SEMI
476	RC	
477	ELSE	
478	Stmt (44)	
479	CompSt (44)	
480	LC	
481	StmtList (45)	
482	Stmt (45)	
483	Exp (45)	
484	Exp (45)	
485	Exp (45)	
486	ID:	areas
487		LB
488		Exp (45)
489	ID:	count
490		RB
491		ASSIGNOP
492		Exp (45)
493		Exp (45)
494	ID:	count
495		STAR
496		Exp (45)
497		INT: 2
498		SEMI

499		RC
500		StmtList (47)
501		Stmt (47)
502		Exp (47)
503		Exp (47)
504		ID: count
505		ASSIGNOP
506		Exp (47)
507		Exp (47)
508		ID: count
509		PLUS
510		Exp (47)
511		INT: 1
512		SEMI
513		RC
514		StmtList (50)
515		Stmt (50)
516		IF
517		LP
518		Exp (50)
519		Exp (50)
520		LP
521		Exp (50)
522		Exp (50)
523		ID: d_sq
524		RELOP
525		Exp (50)
526		INT: 10000
527		RP
528		OR
529		Exp (50)
530		LP
531		Exp (50)
532		Exp (50)
533		ID: d_sq
534		RELOP
535		Exp (50)

536	INT: 2500
537	RP
538	RP
539	Stmt (50)
540	CompSt (50)
541	LC
542	StmtList (51)
543	Stmt (51)
544	RETURN
545	Exp (51)
546	INT: 1
547	SEMI
548	RC
549	StmtList (54)
550	Stmt (54)
551	RETURN
552	Exp (54)
553	ID: d_sq
554	SEMI
555	RC

## 3.2 C-2

### 3.2.1 输入

```

1  int data[10];
2
3  int swap(int i, int j) {
4      int temp;
5      temp = data[i];
6      data[i] = data[j];
7      data[j] = temp;
8  }
9
10 int partition(int low, int high) {
11     int pivot;
12     int i;
13     int j;
```

```

14     pivot = data[high];
15     i = low - 1;
16     j = low;
17
18     while (j < high) {
19         if (data[j] < pivot) {
20             i = i + 1;
21             swap(i, j);
22         }
23         j = j + 1;
24     }
25     swap(i + 1, high);
26     return i + 1;
27 }
28
29 int quick_sort(int low, int high) {
30     int pi;
31     if (low < high) {
32         pi = partition(low, high);
33         quick_sort(low, pi - 1);
34         quick_sort(pi + 1, high);
35     }
36 }
37
38 int main() {
39     quick_sort(0, 9);
40     return 0;
41 }

```

### 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: data
10     LB
11     INT: 10
12     RB
13     SEMI
14 ExtDefList (3)
15     ExtDef (3)
16         Specifier (3)
17         TYPE: int
18     FunDec (3)
19         ID: swap
20         LP
21         VarList (3)
22             ParamDec (3)
23                 Specifier (3)
24                 TYPE: int
25                 VarDec (3)
26                 ID: i
27             COMMA
28             VarList (3)
29                 ParamDec (3)
30                     Specifier (3)
31                     TYPE: int
32                     VarDec (3)
33                     ID: j
34         RP
35     CompSt (3)
36         LC
37         DefList (4)
38             Def (4)
39                 Specifier (4)
40                 TYPE: int
41             DecList (4)
42                 Dec (4)
43                 VarDec (4)

```

```

44             ID: temp
45         SEMI
46     StmtList (5)
47     Stmt (5)
48     Exp (5)
49     Exp (5)
50     ID: temp
51     ASSIGNOP
52     Exp (5)
53     Exp (5)
54     ID: data
55     LB
56     Exp (5)
57     ID: i
58     RB
59     SEMI
60     StmtList (6)
61     Stmt (6)
62     Exp (6)
63     Exp (6)
64     Exp (6)
65     ID: data
66     LB
67     Exp (6)
68     ID: i
69     RB
70     ASSIGNOP
71     Exp (6)
72     Exp (6)
73     ID: data
74     LB
75     Exp (6)
76     ID: j
77     RB
78     SEMI
79     StmtList (7)
80     Stmt (7)

```

```

81             Exp (7)
82             Exp (7)
83             Exp (7)
84             ID: data
85             LB
86             Exp (7)
87             ID: j
88             RB
89             ASSIGNOP
90             Exp (7)
91             ID: temp
92             SEMI
93             RC
94 ExtDefList (10)
95     ExtDef (10)
96         Specifier (10)
97             TYPE: int
98         FunDec (10)
99             ID: partition
100        LP
101        VarList (10)
102            ParamDec (10)
103                Specifier (10)
104                    TYPE: int
105                VarDec (10)
106                    ID: low
107            COMMA
108            VarList (10)
109                ParamDec (10)
110                    Specifier (10)
111                        TYPE: int
112                    VarDec (10)
113                        ID: high
114        RP
115 CompSt (10)
116     LC
117     DefList (11)

```

```

118         Def (11)
119             Specifier (11)
120             TYPE: int
121             DecList (11)
122             Dec (11)
123             VarDec (11)
124             ID: pivot
125         SEMI
126     DefList (12)
127         Def (12)
128             Specifier (12)
129             TYPE: int
130             DecList (12)
131             Dec (12)
132             VarDec (12)
133             ID: i
134         SEMI
135     DefList (13)
136         Def (13)
137             Specifier (13)
138             TYPE: int
139             DecList (13)
140             Dec (13)
141             VarDec (13)
142             ID: j
143         SEMI
144     StmtList (14)
145         Stmt (14)
146             Exp (14)
147                 Exp (14)
148                 ID: pivot
149             ASSIGNOP
150             Exp (14)
151                 Exp (14)
152                 ID: data
153             LB
154             Exp (14)

```



155	ID: high
156	RB
157	SEMI
158	StmtList (15)
159	Stmt (15)
160	Exp (15)
161	Exp (15)
162	ID: i
163	ASSIGNOP
164	Exp (15)
165	Exp (15)
166	ID: low
167	MINUS
168	Exp (15)
169	INT: 1
170	SEMI
171	StmtList (16)
172	Stmt (16)
173	Exp (16)
174	Exp (16)
175	ID: j
176	ASSIGNOP
177	Exp (16)
178	ID: low
179	SEMI
180	StmtList (18)
181	Stmt (18)
182	WHILE
183	LP
184	Exp (18)
185	Exp (18)
186	ID: j
187	RELOP
188	Exp (18)
189	ID: high
190	RP
191	Stmt (18)

192	CompSt (18)
193	LC
194	StmtList (19)
195	Stmt (19)
196	IF
197	LP
198	Exp (19)
199	Exp (19)
200	Exp (19)
201	ID: data
202	LB
203	Exp (19)
204	ID: j
205	RB
206	RELOP
207	Exp (19)
208	ID: pivot
209	RP
210	Stmt (19)
211	CompSt (19)
212	LC
213	StmtList (20)
214	Stmt (20)
215	Exp (20)
216	Exp (20)
217	ID: i
218	ASSIGNOP
219	Exp (20)
220	Exp (20)
221	ID: i
222	PLUS
223	Exp (20)
224	INT: 1
225	SEMI
226	StmtList (21)
227	Stmt (21)
228	Exp (21)

229	ID: swap
230	LP
231	Args (21)
232	Exp (21)
233	ID: i
234	COMMA
235	Args (21)
236	Exp (21)
237	ID: j
238	RP
239	SEMI
240	RC
241	StmtList (23)
242	Stmt (23)
243	Exp (23)
244	Exp (23)
245	ID: j
246	ASSIGNOP
247	Exp (23)
248	Exp (23)
249	ID: j
250	PLUS
251	Exp (23)
252	INT: 1
253	SEMI
254	RC
255	StmtList (25)
256	Stmt (25)
257	Exp (25)
258	ID: swap
259	LP
260	Args (25)
261	Exp (25)
262	Exp (25)
263	ID: i
264	PLUS
265	Exp (25)

```

266             INT: 1
267             COMMA
268             Args (25)
269             Exp (25)
270             ID: high
271             RP
272             SEMI
273             StmtList (26)
274             Stmt (26)
275             RETURN
276             Exp (26)
277             Exp (26)
278             ID: i
279             PLUS
280             Exp (26)
281             INT: 1
282             SEMI
283         RC
284     ExtDefList (29)
285     ExtDef (29)
286     Specifier (29)
287     TYPE: int
288     FunDec (29)
289     ID: quick_sort
290     LP
291     VarList (29)
292     ParamDec (29)
293     Specifier (29)
294     TYPE: int
295     VarDec (29)
296     ID: low
297     COMMA
298     VarList (29)
299     ParamDec (29)
300     Specifier (29)
301     TYPE: int
302     VarDec (29)

```

303	ID: high
304	RP
305	CompSt (29)
306	LC
307	DefList (30)
308	Def (30)
309	Specifier (30)
310	TYPE: int
311	DecList (30)
312	Dec (30)
313	VarDec (30)
314	ID: pi
315	SEMI
316	StmtList (31)
317	Stmt (31)
318	IF
319	LP
320	Exp (31)
321	Exp (31)
322	ID: low
323	RELOP
324	Exp (31)
325	ID: high
326	RP
327	Stmt (31)
328	CompSt (31)
329	LC
330	StmtList (32)
331	Stmt (32)
332	Exp (32)
333	Exp (32)
334	ID: pi
335	ASSIGNOP
336	Exp (32)
337	ID: partition
338	LP
339	Args (32)

340	Exp (32)
341	ID: low
342	COMMA
343	Args (32)
344	Exp (32)
345	ID: high
346	RP
347	SEMI
348	StmtList (33)
349	Stmt (33)
350	Exp (33)
351	ID: quick_sort
352	LP
353	Args (33)
354	Exp (33)
355	ID: low
356	COMMA
357	Args (33)
358	Exp (33)
359	Exp (33)
360	ID: pi
361	MINUS
362	Exp (33)
363	INT: 1
364	RP
365	SEMI
366	StmtList (34)
367	Stmt (34)
368	Exp (34)
369	ID: quick_sort
370	LP
371	Args (34)
372	Exp (34)
373	Exp (34)
374	ID: pi
375	PLUS
376	Exp (34)

```

377             INT: 1
378             COMMA
379             Args (34)
380             Exp (34)
381             ID: high
382             RP
383             SEMI
384             RC
385             RC
386 ExtDefList (38)
387     ExtDef (38)
388         Specifier (38)
389             TYPE: int
390         FunDec (38)
391             ID: main
392             LP
393             RP
394         CompSt (38)
395             LC
396             StmtList (39)
397                 Stmt (39)
398                     Exp (39)
399                         ID: quick_sort
400                         LP
401                         Args (39)
402                             Exp (39)
403                                 INT: 0
404                                 COMMA
405                                 Args (39)
406                                     Exp (39)
407                                         INT: 9
408                                         RP
409                                         SEMI
410                                         StmtList (40)
411                                             Stmt (40)
412                                                 RETURN
413                                                 Exp (40)

```

414	INT: 0
415	SEMI
416	RC

## 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 hex_val1 = 0xBd0f34;
3     int hex_val2 = 0X12DeF;
4     int oct_val1 = 0123;
5     int oct_val2 = 0077;
6
7     int result = hex_val1 + hex_val2 + oct_val1 + oct_val2;
8     return result;
9 }
```

#### 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: hex_val1
20              ASSIGNOP
21              Exp (2)
22                  INT: 12390196
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: hex_val2
32              ASSIGNOP
33              Exp (3)
34                  INT: 77295
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: oct_val1
44              ASSIGNOP
45              Exp (4)
46                  INT: 83
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: oct_val2
56         ASSIGNOP
57         Exp (5)
58             INT: 63
59     SEMI
60 DefList (7)
61     Def (7)
62         Specifier (7)
63         TYPE: int
64     DecList (7)
65         Dec (7)
66         VarDec (7)
67             ID: result
68         ASSIGNOP
69         Exp (7)
70             Exp (7)
71                 Exp (7)
72                     Exp (7)
73                         ID: hex_val1
74                     PLUS
75                     Exp (7)
76                         ID: hex_val2
77                     PLUS
78                     Exp (7)
79                         ID: oct_val1
80                     PLUS
81                     Exp (7)
82                         ID: oct_val2
83     SEMI
84 StmtList (8)
85     Stmt (8)
86         RETURN

```

87	Exp (8)
88	ID: result
89	SEMI
90	RC

### 4.1.3 说明

说明：1.1 分组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；其他分组的同学只要提示相应的错误（不输出语法树即可）。（注：识别为 A、B 类错误均可。）

## 4.2 D-2

### 4.2.1 输入

```

1 int main() {
2     float num1 = 01.23E-04;
3     float num2 = 13.34e-01;
4     float num3 = 123.45e2;
5     float num4 = .13E+06;
6 }
```

### 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: num1
20         ASSIGNOP
21         Exp (2)
22         FLOAT: 0.000123
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: num2
32         ASSIGNOP
33         Exp (3)
34         FLOAT: 1.334000
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: num3
44         ASSIGNOP
45         Exp (4)
46         FLOAT: 12345.000000
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: num4
56             ASSIGNOP
57             Exp (5)
58             FLOAT: 130000.000000
59             SEMI
60 RC

```

### 4.2.3 说明

1.2 分组的同学需要输出正确的语法树，注意科学计数法浮点数的正确转换。其它分组的同学要提示相应的错误，识别为 A、B 类错误均可。

## 4.3 D-3

### 4.3.1 输入

```

1 int main() {
2     int num /* Declare a int variable named num */;
3     num = 1;
4     if (num == 1) { // Check if num is equal to 1
5         return 0;
6     }
7     return 1;
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: num
20              SEMI
21      StmtList (3)
22          Stmt (3)
23              Exp (3)
24                  Exp (3)
25                      ID: num
26                  ASSIGNOP
27                  Exp (3)
28                      INT: 1
29              SEMI
30      StmtList (4)
31          Stmt (4)
32              IF
33              LP
34              Exp (4)
35                  Exp (4)
36                      ID: num
37                  RELOP
38                  Exp (4)
39                      INT: 1
40              RP
41              Stmt (4)
42                  CompSt (4)
43                      LC
44                      StmtList (5)
45                          Stmt (5)
46                              RETURN

```

```

47             Exp (5)
48             INT: 0
49             SEMI
50             RC
51         StmtList (7)
52         Stmt (7)
53         RETURN
54         Exp (7)
55         INT: 1
56         SEMI
57     RC

```

### 4.3.3 说明

1.3 分组的同学需要输出正确的语法树，不能提示有语法错误；其它分组的同学只要提示相应的错误（不输出语法树）即可。（注：识别为 A、B 类错误均可。）

## 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 cal_permission(int base_addr) {
2     if (base_addr < 0x2000) {
3         return 0777;
4     }
5     return 0x00;
6 }
7
8 int main() {
9

```

```

10     int permission = 0644;
11     int base_addr = 0x1FF0;
12
13     return cal_permission(base_addr);
14 }

```

### 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: cal_permission
8         LP
9         VarList (1)
10          ParamDec (1)
11            Specifier (1)
12              TYPE: int
13            VarDec (1)
14              ID: base_addr
15          RP
16        CompSt (1)
17          LC
18          StmtList (2)
19            Stmt (2)
20              IF
21              LP
22              Exp (2)
23                Exp (2)
24                  ID: base_addr
25              RELOP
26              Exp (2)
27                INT: 8192
28              RP
29            Stmt (2)

```



```

30         CompSt (2)
31         LC
32         StmtList (3)
33         Stmt (3)
34         RETURN
35         Exp (3)
36         INT: 511
37         SEMI
38         RC
39         StmtList (5)
40         Stmt (5)
41         RETURN
42         Exp (5)
43         INT: 0
44         SEMI
45         RC
46     ExtDefList (8)
47     ExtDef (8)
48     Specifier (8)
49     TYPE: int
50     FunDec (8)
51     ID: main
52     LP
53     RP
54     CompSt (8)
55     LC
56     DefList (10)
57     Def (10)
58     Specifier (10)
59     TYPE: int
60     DecList (10)
61     Dec (10)
62     VarDec (10)
63     ID: permission
64     ASSIGNOP
65     Exp (10)
66     INT: 420

```

```

67         SEMI
68     DefList (11)
69         Def (11)
70             Specifier (11)
71                 TYPE: int
72             DecList (11)
73                 Dec (11)
74                     VarDec (11)
75                         ID: base_addr
76                     ASSIGNOP
77                     Exp (11)
78                         INT: 8176
79         SEMI
80     StmtList (13)
81         Stmt (13)
82             RETURN
83             Exp (13)
84                 ID: cal_permission
85             LP
86             Args (13)
87                 Exp (13)
88                     ID: base_addr
89             RP
90         SEMI
91     RC

```

### 5.1.3 说明

## 5.2 E1-2

### 5.2.1 输入

```

1  int main() {
2
3      int mode = 089778;
4      int code = 0xGHda;
5
6      code = sqrt(pow(mode, 2) + pow(code, 2)) / (mode + code) - pow(
          mode * code, 2);

```

```

7     return code;
8 }

```

### 5.2.2 输出

```

1 Error type B at Line 3: Syntax error near '89778'
2 Error type B at Line 4: Syntax error near 'xGHda'

```

### 5.2.3 说明

识别为 A、B 类错误均可。

## 5.3 E2-1

### 5.3.1 输入

```

1 int main() {
2     float charge, speed, mass, gravity, energy;
3
4     speed = 2.9979e+6;
5     mass = 9.1e-2;
6     gravity = 9.8E0;
7     energy = 123.456e-1;
8
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          DecList (2)
17              Dec (2)
18                  VarDec (2)
19                      ID: charge
20          COMMA
21          DecList (2)
22              Dec (2)
23                  VarDec (2)
24                      ID: speed
25          COMMA
26          DecList (2)
27              Dec (2)
28                  VarDec (2)
29                      ID: mass
30          COMMA
31          DecList (2)
32              Dec (2)
33                  VarDec (2)
34                      ID: gravity
35          COMMA
36          DecList (2)
37              Dec (2)
38                  VarDec (2)
39                      ID: energy
40      SEMI
41      StmtList (4)
42          Stmt (4)
43              Exp (4)
44                  Exp (4)
45                      ID: speed
46          ASSIGNOP
47          Exp (4)

```

```

48             FLOAT: 2997900.000000
49         SEMI
50     StmtList (5)
51     Stmt (5)
52     Exp (5)
53     Exp (5)
54     ID: mass
55     ASSIGNOP
56     Exp (5)
57     FLOAT: 0.091000
58     SEMI
59     StmtList (6)
60     Stmt (6)
61     Exp (6)
62     Exp (6)
63     ID: gravity
64     ASSIGNOP
65     Exp (6)
66     FLOAT: 9.800000
67     SEMI
68     StmtList (7)
69     Stmt (7)
70     Exp (7)
71     Exp (7)
72     ID: energy
73     ASSIGNOP
74     Exp (7)
75     FLOAT: 12.345600
76     SEMI
77 RC

```

### 5.3.3 说明

## 5.4 E2-2

### 5.4.1 输入

```

1 int main() {
2     float a, b, c, d, e;

```

```

3     a = 1.6e;
4     b = 7e-;
5
6     return 0;
7 }

```

### 5.4.2 输出

```

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

```

### 5.4.3 说明

识别为 A、B 类错误均可。

## 5.5 E3-1

### 5.5.1 输入

```

1  /* 测试1：普通多行注释（合法） */
2  /* 内容包含 * 和 / 符号，但无嵌套 */
3  /* 特殊符号：/ * \ 的组合测试 */
4
5  /* 测试2：边界注释闭合（合法） */
6  /*.\//\//\*/ // 含转义符的注释闭合测试
7  /*****合法注释*(*\/*\**\)* */
8
9  // 测试3：单行注释内的特殊符号（合法）
10 // /* 伪多行注释开始（仅作为文本）
11 //     伪嵌套注释：/* 内部内容 */
12 //     伪闭合符号：*/
13
14 int main() {
15     int x = 10; // 单行注释：包含 /* 和 */ 符号（合法文本）
16     int y = 20; /* 多行注释内的字符串：（合法） */
17
18     /* 测试4：注释与代码混合（合法） */
19     return

```

```

20     /* 注释分割return和0 */ 0;
21 }

```

### 5.5.2 输出

```

1 Program (14)
2   ExtDefList (14)
3     ExtDef (14)
4       Specifier (14)
5         TYPE: int
6       FunDec (14)
7         ID: main
8         LP
9         RP
10      CompSt (14)
11        LC
12        DefList (15)
13          Def (15)
14            Specifier (15)
15              TYPE: int
16            Declist (15)
17              Dec (15)
18                VarDec (15)
19                  ID: x
20                  ASSIGNOP
21                  Exp (15)
22                    INT: 10
23              SEMI
24            DefList (16)
25              Def (16)
26                Specifier (16)
27                  TYPE: int
28                Declist (16)
29                  Dec (16)
30                    VarDec (16)
31                      ID: y
32                      ASSIGNOP

```

```

33             Exp (16)
34             INT: 20
35             SEMI
36         StmtList (19)
37         Stmt (19)
38         RETURN
39         Exp (20)
40         INT: 0
41         SEMI
42     RC

```

### 5.5.3 说明

## 5.6 E3-2

### 5.6.1 输入

```

1  /*/* 内层注释尝试嵌套（错误！） */*/
2
3  int main() {
4      float value = 3.14;
5      return 0;
6  }

```

### 5.6.2 输出

```

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

```

### 5.6.3 说明

## 6 结束语

若对本文档有任何疑问，可写邮件与[孙伟杰](#)助教联系，注意同时抄送给[许畅老师](#)。