

编译原理第二次实验测试用例：目录

1	A 组测试用例	2
1.1	A-1	2
1.2	A-2	3
1.3	A-3	4
1.4	A-4	4
1.5	A-5	6
1.6	A-6	7
1.7	A-7	8
1.8	A-8	9
1.9	A-9	10
1.10	A-10	12
1.11	A-11	13
1.12	A-12	14
1.13	A-13	16
1.14	A-14	17
1.15	A-15	18
1.16	A-16	19
1.17	A-17	20
1.18	A-18	20
1.19	A-19	22
1.20	A-20	22
2	B 组测试用例	24
2.1	B-1	24
2.2	B-2	26
3	C 组测试用例	29
3.1	C-1	29
3.2	C-2	31

4	D 组测试用例	34
4.1	D-1	35
4.2	D-2	36
4.3	D-3	37
5	E 组测试用例	39
5.1	E2.1	39
5.2	E2.2	40
5.3	E2.3	44
6	结束语	45

1 A 组测试用例

本组测试用例共 20 个，测试用例 1-17 分别对应语义错误 1-17，之后三个测试用例对应于语义错误 9, 15, 16。每个用例仅在其中一行含有语义错误。某些语义错误可能会产生连锁反应。测试用例 A-i 对应的“本质错误”的错误类型是必须报出来的，如果报出其他错误，只要是由本质错误连带引发的（包括但不限于下面明确给出的情况），我们都不会扣分。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 struct OneDimVector {
2     int id;
3     float x;
4 };
5
6 struct OneDimVector _, _1, _2;
7
8 int foo() {
9     struct OneDimVector __;
10
11     _.id = 0;
12     _1.id = 1;
13     _2.id = 2;
14     __.id = -1;
15
16     _.x = _1.x = _2.x = __.x = 0.1;
17
18     return __.id;
19 }
```

输出

```
1 Error type 1 at line 18: Undefined variable "___"
```

说明: return ____ .id 这一句包含未定义的变量 ____。

1.2 A-2

输入

```
1 int sort(int a[10]) {
2     int i = 0;
3     while (i < 10) {
4         int j = 0;
5         while (j < 10) {
6             if (a[j] <= a[j+1]) {
7                 a[j] = 0;
8             } else {
9                 swap(a, j, j + 1);
10            }
11            j = j + 1;
12        }
13        i = i + 1;
14    }
15    return 1;
16 }
17
18 int swap(int aa[0], int x, int y) {
19     int t = aa[x];
20     aa[x] = aa[y];
21     aa[y] = t;
22     return 0;
23 }
```

输出

```
1 Error type 2 at line 9: Undefined function "swap"
```

说明: swap 函数在第 18 行定义, 而在第 9 行使用时还未定义。

1.3 A-3

输入

```
1 struct Vector {
2     int x, y;
3 };
4
5 int vec_is_zero(struct Vector Vector) {
6     if (Vector.x == 0 && Vector.y == 0) {
7         return 1;
8     } else {
9         return 0;
10    }
11 }
12
13 int vec_is_equal(struct Vector v, struct Vector w) {
14     if (v.x == w.x && v.y == w.y) {
15         return 1;
16     } else {
17         return 0;
18     }
19 }
```

输出

```
1 Error type 3 at line 5: Redefined variable "Vector"
```

说明：第 5 行定义的变量 `Vector` 与第 1 行的结构体重名，也可以报在第 1 行。

1.4 A-4

输入

```
1 struct IntVector {
2     int ix, iy;
3 };
```

```

4
5 struct FloatVector {
6     int fx, fy;
7 };
8
9 int vec_is_equal(struct IntVector iv, struct IntVector iw) {
10     if (iv.ix == iw.ix && iv.iy == iw.iy) {
11         return 1;
12     } else {
13         return 0;
14     }
15 }
16
17 int vec_is_equal(struct FloatVector fv, struct FloatVector fw) {
18     if (fv.fx == fw.fx && fv.fy == fw.fy) {
19         return 1;
20     } else {
21         return 0;
22     }
23 }
24
25 int main() {
26     struct IntVector v1;
27     struct FloatVector v2;
28
29     vec_is_equal(v1, v1);
30     vec_is_equal(v2, v2);
31
32     return 0;
33 }

```

输出

```

1 Error type 4 at line 17: Redefined function "vec_is_equal"

```

说明：第 17 行定义的函数 `vec_is_equal` 与第 9 行重名，也可以报在第 9 行。

1.5 A-5

输入

```
1 struct IntVector {
2     int ix, iy;
3 };
4
5 struct FloatVector {
6     int fx, fy, fz;
7 };
8
9 int ivec_is_equal(struct IntVector iv, struct IntVector iw) {
10     if (iv.ix == iw.ix && iv.iy == iw.iy) {
11         return 1;
12     } else {
13         return 0;
14     }
15 }
16
17 int fvec_is_equal(struct FloatVector fv, struct FloatVector fw) {
18     if (fv.fx == fw.fx && fv.fy == fw.fy && fv.fz == fw.fz) {
19         return 1;
20     } else {
21         return 0;
22     }
23 }
24
25 int main() {
26     struct IntVector iv1, iv2;
27     struct FloatVector fv1, fv2;
```

```

28
29     iv1.ix = 0;
30     iv1.iy = 0;
31
32     fv1.fx = 2;
33     fv1.fy = 4;
34     fv1.fz = 4;
35
36     iv2 = iv1;
37     fv2 = iv1;
38
39     ivec_is_equal(iv1, iv2);
40     fvec_is_equal(fv1, fv2);
41
42     return 0;
43 }

```

输出

```

1 Error type 5 at line 37: Type mismatch

```

说明：第 37 行赋值号两侧类型不一致。

1.6 A-6

输入

```

1 struct IntVector {
2     int ix, iy;
3 };
4
5 int ivec_is_equal(struct IntVector iv, struct IntVector iw) {
6     if (iv.ix == iw.ix && iv.iy == iw.iy) {
7         return 1;
8     } else {
9         return 0;

```



```

10     }
11 }
12
13 int main() {
14     struct IntVector iv1, iv2;
15
16     iv1.ix = 0;
17     iv1.iy = 0;
18
19     iv2.ix = 0;
20     iv2.iy = 1;
21
22     if (ivec_is_equal(iv1, iv2) = 0) {
23         iv2 = iv1;
24     }
25
26     return 0;
27 }

```

输出

```

1 Error type 6 at line 22: Assign a value to a right-hand-only
  expression

```

说明：第 22 行将 0 赋值给了一个右值表达式。

1.7 A-7

输入

```

1 struct IntVector {
2     int ix, iy;
3 };
4
5 struct IntVector subtractX(struct IntVector v1, struct IntVector v2)
6 {

```

```

6  struct IntVector result;
7  result.ix = v1.ix - v2.ix;
8  result.iy = v1.iy;
9  return result;
10 }
11
12 int main() {
13     struct IntVector rv, cv;
14     int i = 0;
15
16     rv.ix = 0;
17     rv.iy = 0;
18     cv.ix = 1;
19     cv.iy = 2;
20
21     while (i < 10) {
22         rv.ix = subtractX(rv, cv) + cv.ix;
23         i = i + 1;
24     }
25
26     return 0;
27 }

```

输出

```

1 Error type 7 at line 22: Operand type mismatch

```

说明：第 22 行加号左右两侧类型不匹配，也可以多报一个 5 型错误。

1.8 A-8

输入

```

1 struct IntVector {
2     int ix, iy;
3 };

```

```

4
5 int diffX(struct IntVector v1, struct IntVector v2) {
6     struct IntVector result;
7     result.ix = v1.ix - v2.ix;
8     result.iy = v1.iy;
9     return result;
10 }
11
12 int main() {
13     struct IntVector rv, cv;
14     int i = 0;
15
16     rv.ix = 0;
17     rv.iy = 0;
18     cv.ix = 1;
19     cv.iy = 2;
20
21     while (i < 10) {
22         rv.ix = diffX(rv, cv) + cv.ix;
23         i = i + 1;
24     }
25
26     return 0;
27 }

```

输出

```

1 Error type 8 at line 9: Return type mismatch

```

说明：第9行返回值类型 struct IntVector 与声明类型 int 不匹配

1.9 A-9

输入

```

1 struct IntVector {

```

```

2     int ix, iy;
3 };
4
5 struct FloatVector {
6     int fx, fy, fz;
7 };
8
9 int ivec_is_equal(struct IntVector iv, struct IntVector iw) {
10     if (iv.ix == iw.ix && iv.iy == iw.iy) {
11         return 1;
12     } else {
13         return 0;
14     }
15 }
16
17 int fvec_is_equal(struct FloatVector fv, struct FloatVector fw) {
18     if (fv.fx == fw.fx && fv.fy == fw.fy && fv.fz == fw.fz) {
19         return 1;
20     } else {
21         return 0;
22     }
23 }
24
25 int main() {
26     struct IntVector iv1, iv2;
27     struct FloatVector fv1, fv2;
28
29     iv1.ix = 0;
30     iv1.iy = 0;
31
32     fv1.fx = 2;
33     fv1.fy = 4;

```

```

34     fv1.fz = 4;
35
36     iv2 = iv1;
37     fv2 = fv1;
38
39     ivec_is_equal(iv1, fv1);
40     fvec_is_equal(fv1, fv2);
41
42     return 0;
43 }

```

输出

```

1 Error type 9 at line 39: Function arguments do not match

```

说明：第 39 行函数调用时实参与形参不匹配。

1.10 A-10

输入

```

1 struct Array {
2     int elements[10];
3 } global_array;
4
5 int main() {
6     struct Array local_arrays[10];
7     int i = 0;
8
9     while (i < 10) {
10         global_array.elements[i] = i;
11         i = i + 1;
12     }
13
14     i = 0;
15     while (i < 10) {

```

```

16     int j = 0;
17     while (j < 10) {
18         local_arrays[i].elements[j] = global_array.elements[j];
19         j = j + 1;
20     }
21     i = i + 1;
22 }
23
24 return local_arrays[0][0];
25 }

```

输出

```

1 Error type 10 at line 24: Illegal use of of "[" operator

```

说明：第 24 行对结构体变量 `local_arrays[0]` 使用了数组访问操作符，这里可以多报一个 8 型错误。

1.11 A-11

输入

```

1 struct Array {
2     int elements[10];
3 } global_array;
4
5 struct Array newArray(int value) {
6     struct Array result;
7     int ii = 0;
8     while (ii < 10) {
9         result.elements[ii] = value;
10    }
11    return result;
12 }
13
14 int main() {

```

```

15  struct Array local_arrays[10];
16  int i = 0, new_array;
17
18  while (i < 10) {
19      global_array.elements[i] = i;
20      local_arrays[i] = newArray(i);
21      i = i + 1;
22  }
23
24  new_array = local_arrays[0].elements[0];
25
26  return new_array(0);
27 }

```

输出

```

1 Error type 11 at line 26: Illegal use of operator "("

```

说明：第 26 行对整型变量 `new_array` 使用了函数调用操作符，这里可以多报一个 8 型错误。

1.12 A-12

输入

```

1  struct Array {
2      int elements[10];
3      float curr;
4  } global_array;
5
6  struct Array newArray(int value) {
7      struct Array result;
8      int ii = 0;
9      while (ii < 10) {
10         result.elements[ii] = value;
11     }

```

```

12     result.curr = 1.0;
13     return result;
14 }
15
16 float multiplyBy(float m, float n) {
17     return m * n;
18 }
19
20 int main() {
21     struct Array local_arrays[10];
22     int i = 0;
23     float pointer = 1.0;
24
25     global_array.curr = 1.0;
26
27     while (i < 10) {
28         global_array.elements[i] = i;
29         global_array.curr = global_array.curr * pointer;
30
31         local_arrays[i] = newArray(i);
32         local_arrays[i].curr = pointer;
33
34         i = i + 1;
35         pointer = pointer + 1.0;
36     }
37
38     i = local_arrays[0].elements[multiplyBy(global_array.curr,
39         local_arrays[9].curr)];
40
41     return i;
42 }

```

输出


```
1 Error type 12 at line 38: Array index is not an integer
```

说明：第 38 行使用了 float 作为数组索引，这里可以多报一个 5 型错误。

1.13 A-13

输入

```
1 struct Array {
2     int elements[10];
3 } global_array1;
4
5 int global_array2[10];
6
7 struct Array newArray(int value) {
8     struct Array result;
9     int ii = 0;
10    while (ii < 10) {
11        result.elements[ii] = global_array2.elements[ii] * value;
12    }
13    return result;
14 }
15
16 int main() {
17     struct Array local_arrays[10];
18     int i = 0;
19
20     while (i < 10) {
21         global_array1.elements[i] = i;
22         global_array2[i] = i * 2;
23         local_arrays[i] = newArray(i * 3);
24         i = i + 1;
25     }
26 }
```

```
27     return local_arrays[0].elements[0];
28 }
```

输出

```
1 Error type 13 at line 11: Illegal use of operator "."
```

说明：第 11 行对数组变量 `global_array2` 使用了“.”操作符，这里可以多报一个 5 型错误。

1.14 A-14

输入

```
1 struct Array {
2     int elements[10];
3     struct {
4         int curr;
5         int prev;
6     } pointer;
7 } global_array;
8
9 struct Array newArray(int value) {
10     struct Array result;
11     int ii = 0;
12     while (ii < 10) {
13         result.elements[ii] = value;
14     }
15     return result;
16 }
17
18 int main() {
19     struct Array local_arrays[10];
20     int i = 0;
21
22     while (i < 10) {
```

```

23     global_array.elements[i] = i;
24     global_array.pointer.curr = i;
25     global_array.pointer.prev = i - 1;
26     local_arrays[i] = newArray(i * 3);
27     i = i + 1;
28 }
29
30 return local_arrays[0].elements[global_array.curr];
31 }

```

输出

```

1 Error type 14 at line 30: Undefined field "curr"

```

说明：第 30 行使用了未定义的域 curr。

1.15 A-15

输入

```

1 struct WeighedFood {
2     float price, weight;
3 } f1;
4
5 int main() {
6     struct CountedFood {
7         struct {
8             int max = 1024, min, bought;
9         } counter;
10        float per_price;
11    } f2;
12
13    return 0;
14 }

```

输出

```
1 Error type 15 at line 8: Initialized field "max"
```

说明：第 8 行对域 max 进行了初始化，结构体名不允许初始化。

1.16 A-16

输入

```
1 struct Hammer {  
2     int size;  
3     int weight;  
4     struct Price {  
5         int amount;  
6         float currency;  
7     } p;  
8 };  
9  
10 struct Thor {  
11     struct Hammer hh;  
12     int age;  
13     struct Hammer {  
14         int am;  
15         float current;  
16     } ww;  
17 } avenger;  
18  
19 int main() {  
20     avenger.age = 100;  
21     return avenger.age;  
22 }
```

输出

```
1 Error type 16 at line 13: Duplicated name "Hammer"
```

说明：结构体名 Hammer 与之前定义的结构体重名，也可以报在第 1 行。

1.17 A-17

输入

```
1 struct Hammer {
2     int size;
3     int weight;
4     struct Price {
5         int amount;
6         float currency;
7     } price;
8 };
9
10 int main() {
11     struct Thor aThor;
12     return 0;
13 }
14
15 struct Thor {
16     struct Hammer hammer;
17     int age;
18 };
```

输出

```
1 Error type 17 at line 11: Undefined structure "Thor"
```

说明：结构体 Thor 在第 15 行定义，第 11 行使用时还未定义。

1.18 A-18

输入

```
1 struct IntVector {
2     int ix, iy;
3 };
4
```

```

5 int ivec_is_equal(struct IntVector iv01, struct IntVector iv02) {
6     if (iv01.ix == iv02.ix && iv01.iy == iv02.iy) {
7         return 1;
8     } else {
9         return 0;
10    }
11 }
12
13 int ivec3_is_equal(struct IntVector iv03, struct IntVector iv04,
14                    struct IntVector iv05) {
15     if (ivec_is_equal(iv03, iv04) == 1 && ivec_is_equal(iv04, iv05) ==
16         1) {
17         return 1;
18     } else {
19         return 0;
20     }
21 }
22
23 int main() {
24     struct IntVector iv1, iv2, iv3;
25
26     iv1.ix = 0;
27     iv1.iy = 0;
28
29     iv2 = iv1;
30     iv2.ix = iv2.ix + 1;
31
32     iv3 = iv1;
33     iv3.iy = iv3.iy + 10;
34
35     ivec3_is_equal(iv1, iv2);

```

```
35     return 0;
36 }
```

输出

```
1 Error type 9 at line 33: Arguments for function "ivec3_is_equal" is
   not match
```

说明：第 33 行函数 `ivec3_is_equal` 需要 3 个参数。

1.19 A-19

输入

```
1 struct WeighedFood {
2     float price, weight;
3 } f1;
4
5 int main() {
6     struct CountedFood {
7         struct {
8             int max, min, bought;
9         } counter;
10        float per_price;
11        int counter;
12    } f2;
13
14    return 0;
15 }
```

输出

```
1 Error type 15 at line 11: Duplicated field name "counter"
```

说明：第 11 行的域名和第 9 行重名，也可以报在第 9 行。

1.20 A-20

输入

```

1 struct Hammer {
2     int size;
3     int weight;
4     struct Price {
5         int amount;
6         float currency;
7     } p;
8 };
9
10 struct Thor {
11     struct Hammer hh;
12     int age;
13     struct Hammers {
14         int am;
15         float current;
16     } ww;
17 } avenger;
18
19 int main() {
20     int Hammer = 12;
21     avenger.age = 100;
22     return avenger.age;
23 }

```

输出

```

1 Error type 3 at line 20: Duplicated name "Hammer"

```

说明：第 20 行定义的变量与第 1 行的结构体重名，也可以报成 16 型错误，但必须在第 1 行。

2 B 组测试用例

本组测试用例共 2 个，其中包含多个语义错误。每一行的语义错误会分别算分，同一个语义错误可能会有连锁反应，其处理方式与 A 类用例相同，只要是合理的（包括但不限于下面明确给出的情况），都不会影响得分。

2.1 B-1

输入

```
1 struct Rectangle {
2     int tlx, tly;
3     int w, h;
4 };
5
6 struct Circle {
7     int cx, cy;
8     int cr;
9 };
10
11 struct Rectangle makeRect(int etlx, int etly, int ew, int eh) {
12     struct Rectangle erect;
13     erect.tlx = etlx;
14     erect.tly = etly;
15     erect.rw = ew;
16     erect.h = eh;
17     return erect;
18 }
19
20 struct Circle makeCirc(int fcx, int fcy, int fcr) {
21     struct Circle fcirc;
22     fcirc.cx = fcx;
23     fcirc.cy = fcy;
24     fcirc.cr = fcr;
```

```

25     return fcirc(12);
26 }
27
28 int calArea(struct Rectangle arect) {
29     return arect.w * arect.h;
30 }
31
32 int calArea(struct Circle bcirc) {
33     return 3 * bcirc.cr * bcirc.cr;
34 }
35
36 int isRCover(struct Rectangle direct, int dx, int dy) {
37     int dtop = direct.tly;
38     int dleft = direct.tlx;
39     int dbottom = dtop + direct.h;
40     int dright = dleft + direct.w;
41
42     if (dleft <= dx && dx <= dright) {
43         if (dtop <= dy && dy <= dbottom) {
44             return 1;
45         }
46     }
47
48     return 0;
49 }
50
51 int main() {
52     struct Rectangle mr = makeRect(1, 4, 32, 53);
53     struct Circle mc = makeCirc(12.1, 21, 4.3);
54     int mx = 12, my = mc.cx * mc.cy / mc.cr;
55     return isRCover(mr, mx, my);
56 }

```

输出

```
1 Error type 14 at line 15: Undefined field "rw"
2 Error type 11 at line 25: Illegal use of operator "()"
3 Error type 4 at line 32: Redefiend function "calArea"
4 Error type 9 at line 53: Function arguments do not match
```

说明：第 15 行域 `rw` 未定义，这里可以多报一个 5 型错误；第 25 行对结构体类型 `struct Circle` 使用了函数调用操作符，这里可以多报一个 8 型错误；第 32 行函数与 28 行重定义，也可以报在第 28 行；第 53 行函数实参与形参类型不匹配，这里可以多报一个 5 型错误。

2.2 B-2

输入

```
1 struct FibLikeArray {
2     int a0, a1;
3     int _buf[32];
4 };
5
6 int min(int b0, int b1) {
7     if (b0 < b1) {
8         return b0;
9     } else {
10        return b1;
11    }
12 }
13
14 struct FibLikeArray makeFibLikeArray(int c0, int c1) {
15     struct FibLikeArray cfla;
16
17     if (c0 < 0 || c1 < 0 || (c0 + c1 == ZERO)) {
18         cfla.a0 = -1;
19         return -1;
20     }
21 }
```

```

22     cfla.a0 = min(c0, c1);
23     cfla.a1 = max(c0, c1);
24
25     {
26         int ci = 2;
27
28         cfla._buf[0] = cfla.a0;
29         cfla._buf[1] = cfla.a1;
30         cfla._buf[2.1] = -1;
31
32         while (ci < 32) {
33             cfla._buf[ci] = cfla._buf[ci - 1] + cfla._buf[ci - 2];
34             ci = ci + 1;
35         }
36     }
37
38     return cfla;
39 }
40
41 int flaIsGuardian(struct FibLikeArray farray) {
42     if (farray.a0 == -1) {
43         return 1;
44     } else {
45         return 0;
46     }
47 }
48
49 int flaGet(struct FibLikeArray efla, int eindex) {
50     if (flaIsGuardian(efla) = 1) {
51         return -1;
52     }
53

```

```

54  if (eindex <= 32) {
55      return efla._buf[eindex];
56  }
57
58  {
59      int e0 = efla._buf[32 - 2];
60      int e1 = efla._buf[32 - 1];
61      int er;
62      int ei = 32;
63
64      while (ei <= eindex) {
65          er = e0 + e1;
66          e0 = e1;
67          e1 = er;
68          ei = ei + 1;
69      }
70
71      return er;
72  }
73 }
74
75 int main() {
76     struct FibLikeArray mfla = makeFibLikeArray(0, 1);
77     int m0, m1, mr;
78
79     if (flaIsGuardian(mfla)) {
80         return 1;
81     }
82
83     m0 = flaGet(mfla, 10, 80);
84     m1 = mfla[36];
85     mr = m1 * mr;

```

```

86
87     return 0;
88 }

```

输出

```

1 Error type 1 at line 17: Undefined variable "ZERO"
2 Error type 8 at line 19: Return type mismatch
3 Error type 2 at line 23: Undefined function "max"
4 Error type 12 at line 30: Array index is not an integer
5 Error type 6 at line 50: Assign a value to a right-hand-only
  expression
6 Error type 9 at line 83: Function arguments do not match
7 Error type 10 at line 84: Illegal use of of "[]" operator

```

说明：第 17 行出现未定义变量 ZERO，这里可以多报一个 7 型错误；第 19 行返回值类型 int 与声明类型 struct FibLikeArray 不匹配；第 23 行 max 函数未定义，这里可以多报一个 5 型错误；第 30 行数组索引应该为整数；第 50 行赋值常数 1 给右值表达式；第 83 行函数实参与形参数目不一匹配，这里可以多报一个 5 型错误；第 84 行对结构体变量使用了数组访问操作符，这里可以多报一个 5 型错误。

3 C 组测试用例

本组测试用例共 2 个，不包含任何错误。

3.1 C-1

输入

```

1 struct Rectangle {
2     int tlx, tly;
3     int w, h;
4 };
5
6 struct Circle {
7     int cx, cy;

```

```

8   int cr;
9   };
10
11  struct Rectangle makeRect(int etlx, int etly, int ew, int eh) {
12      struct Rectangle erect;
13      erect.tlx = etlx;
14      erect.tly = etly;
15      erect.w = ew;
16      erect.h = eh;
17      return erect;
18  }
19
20  struct Circle makeCirc(int fcx, int fcy, int fcr) {
21      struct Circle fcirc;
22      fcirc.cx = fcx;
23      fcirc.cy = fcy;
24      fcirc.cr = fcr;
25      return fcirc;
26  }
27
28  int calRArea(struct Rectangle arect) {
29      return arect.w * arect.h;
30  }
31
32  int calCArea(struct Circle bcirc) {
33      return 3 * bcirc.cr * bcirc.cr;
34  }
35
36  int isRCover(struct Rectangle direct, int dx, int dy) {
37      int dtop = direct.tly;
38      int dleft = direct.tlx;
39      int dbottom = dtop + direct.h;

```

```

40  int dright = dleft + drect.w;
41
42  if (dleft <= dx && dx <= dright) {
43      if (dtop <= dy && dy <= dbottom) {
44          return 1;
45      }
46  }
47
48  return 0;
49 }
50
51 int main() {
52     struct Rectangle mr = makeRect(1, 4, 32, 53);
53     struct Circle mc = makeCirc(12, 21, 4);
54     int mx = 12, my = mc.cx * mc.cy / mc.cr;
55     return isRCover(mr, mx, my);
56 }

```

输出

```

1 // 正常返回，没有任何输出

```

3.2 C-2

输入

```

1 struct FibLikeArray {
2     int a0, a1;
3     int _buf[32];
4 };
5
6 int min(int b0, int b1) {
7     if (b0 < b1) {
8         return b0;
9     } else {

```



```

10     return b1;
11 }
12 }
13
14 int max(int d0, int d1) {
15     if (d0 > d1) {
16         return d0;
17     } else {
18         return d1;
19     }
20 }
21
22 struct FibLikeArray makeFibLikeArray(int c0, int c1) {
23     struct FibLikeArray cfla;
24
25     if (c0 < 0 || c1 < 0 || (c0 + c1 == 0)) {
26         cfla.a0 = -1;
27         return cfla;
28     }
29
30     cfla.a0 = min(c0, c1);
31     cfla.a1 = max(c0, c1);
32
33     {
34         int ci = 2;
35
36         cfla._buf[0] = cfla.a0;
37         cfla._buf[1] = cfla.a1;
38         cfla._buf[2] = -1;
39
40         while (ci < 32) {
41             cfla._buf[ci] = cfla._buf[ci - 1] + cfla._buf[ci - 2];

```

```

42     ci = ci + 1;
43 }
44 }
45
46 return cfla;
47 }
48
49 int flaIsGuardian(struct FibLikeArray farray) {
50     if (farray.a0 == -1) {
51         return 1;
52     } else {
53         return 0;
54     }
55 }
56
57 int flaGet(struct FibLikeArray efla, int eindex) {
58     if (flaIsGuardian(efla)) {
59         return -1;
60     }
61
62     if (eindex <= 32) {
63         return efla._buf[eindex];
64     }
65
66     {
67         int e0 = efla._buf[32 - 2];
68         int e1 = efla._buf[32 - 1];
69         int er;
70         int ei = 32;
71
72         while (ei <= eindex) {
73             er = e0 + e1;

```

```

74     e0 = e1;
75     e1 = er;
76     ei = ei + 1;
77 }
78
79     return er;
80 }
81 }
82
83 int main() {
84     struct FibLikeArray mfla = makeFibLikeArray(0, 1);
85     int m0, m1, mr;
86
87     if (flaIsGuardian(mfla)) {
88         return 1;
89     }
90
91     m0 = flaGet(mfla, 10);
92     m1 = flaGet(mfla, 37);
93     mr = m1 * mr;
94
95     return 0;
96 }

```

输出

```
1 // 正常返回，没有任何输出
```

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。需要能够识别其语言特性，如果提示错误则不得分；其他分组的同学需要识别出其中的错误，如果没有报错，则将视为违规，将会倒扣分。

4.1 D-1

输入

```
1 int equal(int ax, int ay);
2 int equal(int zx, int zy) {
3     if (zx == zy) {
4         return 1;
5     } else {
6         return 0;
7     }
8 }
9
10 int add(int dx, int dy);
11
12 int subtract(int cx, int cy) {
13     return cx - cy;
14 }
15
16 float subtractf(float qx, float qy) {
17     return qx - qy;
18 }
19
20 int main() {
21     int mix = 0, miy = 1;
22     float mfx = 1.2, mfy = 4.2;
23
24     int mir;
25     float mfr;
26
27     {
28         mir = subtract(mix, miy);
29         if (mir == 0) {
30             return 1;
```

```

31     }
32
33     mfr = subtractf(mfx, mfy);
34 }
35
36 return 0;
37 }
38
39 int add(int ex, int ey) {
40     return ex + ey;
41 }

```

输出

```

1 // 正常返回， 没有任何输出。

```

说明：对于 2.1 分组的同学没有任何输出，其他同学在 1，10 行报语法错误。

4.2 D-2

输入

```

1 int res;
2
3 int sumIfGT(int a[10], int b) {
4     int res = 0;
5     int i = 0;
6     while(i < 10) {
7         res = res + a[i];
8         if (a[i] > b) {
9             int t = 3;
10            res = res - b;
11        }
12    }
13    return res;
14 }

```

```

15
16 int prodIfGT(int a[10], int b2) {
17     int res1 = 1;
18     int i = 0;
19     while(i < 10) {
20         if (a[i] > b2) {
21             res1 = res1 * (a[i] - b2);
22         } else {
23             int t = 123;
24             res1 = res1 * a[i];
25         }
26     }
27     return res;
28 }
29
30 int main() {
31     int aa[10];
32     res = sumIfGT(aa, 5);
33     res = res + prodIfGT(aa, 7);
34     return 0;
35 }

```

输出

```

1 // 正常返回，没有任何输出。

```

说明：对于 2.2 分组的同学，没有任何输出；其他同学应该识别到针对变量 `res` , `a` , `i` , `t` 的重复定义。

4.3 D-3

输入

```

1 struct IntArray10 {
2     int iaa10[10];
3 };

```

```

4
5 struct IntArray100 {
6     int iaa100[100];
7 };
8
9 struct IntVector {
10     int ix, iy;
11     int ia[2];
12     struct {
13         float ic;
14         struct IntArray10 id;
15         struct IntArray100 ie[38];
16     } ib;
17 };
18
19 struct FloatVector {
20     int fx, fy;
21     int fa[39];
22     struct {
23         float fc;
24         struct IntArray100 fd;
25         struct IntArray10 fe[38];
26     } fb;
27 };
28
29 int vec_is_equal(struct IntVector iv, struct IntVector iw) {
30     if (iv.ix == iw.ix && iv.iy == iw.iy) {
31         return 1;
32     } else {
33         return 0;
34     }
35 }

```

```

36
37 int main() {
38     struct IntVector v1;
39     struct FloatVector v2;
40
41     if (!vec_is_equal(v1, v2)) {
42         v2 = v1;
43     }
44
45     return 0;
46 }

```

输出

```

1 // 正常返回，没有任何输出

```

说明：对于 2.3 分组的同学，没有任何输出；其他同学应该在 41 行和 42 行分别报出 9 型和 5 型错误。

5 E 组测试用例

本组测试用例共 3 个，针对不同分组进行测试。

5.1 E2.1

这组测试用例针对 2.1 分组的同学。

输入

```

1 int equal(int ax, int ay);
2
3 int subtract(int cx, int cy);
4 int subtract(int fx, float fy);
5
6 int main() {
7     int mix = 0, miy = 1;
8     float mfx = 1.2, mfy = 4.2;

```



```

9
10 int mir;
11 float mfr;
12
13 {
14     mir = subtract(mfx, miy);
15     if (mir == 0) {
16         return 1;
17     }
18
19     mfr = subtract(mfx, mfy);
20 }
21
22 return 0;
23 }

```

输出

```

1 Error type 19 at Line 4: Inconsistent declaration of function "
    subtract"
2 Error type 9 at Line 14: Function arguments do not match
3 Error type 9 at Line 19: Function arguments do not match
4 Error type 18 at Line 1: Undefined function "equal"
5 Error type 18 at Line 3: Undefined function "subtract"

```

说明：仅 2.1 分组的同学需要测试这个用例，需要报出以上错误，其中与函数 `subtract` 相关的 19 型错误也可以报在 3 行，但同时其 18 型错误就要报在第 4 行，与其相关的两个 9 型错误必须报出，也可以同时多报 2 个 5 型错误。

5.2 E2.2

这组测试用例针对 2.2 分组的同学。

输入

```

1 struct FibLikeArray {
2     int a0, a1;

```

```

3   int _buf[32];
4   };
5
6   int min(int a, int b) {
7       if (a < b) {
8           return a;
9       } else {
10          return b;
11      }
12  }
13
14  int max(int a, int b) {
15      if (a > b) {
16          return a;
17      } else {
18          return b;
19      }
20  }
21
22  struct FibLikeArray makeFibLikeArray(int a0, int a1) {
23      struct FibLikeArray fla;
24
25      if (a0 < 0 || a1 < 0 || (a0 + a1 == 0)) {
26          fla.a0 = -1;
27          return fla;
28      }
29
30      fla.a0 = min(a0, a1);
31      fla.a1 = max(a0, a1);
32
33      {
34          fla._buf[0] = fla.a0;

```

```

35     fla._buf[1] = fla.a1;
36     fla._buf[2] = -1;
37
38     while (ci < 32) {
39         fla._buf[ci] = fla._buf[ci - 1] + fla._buf[ci - 2];
40         ci = ci + 1;
41     }
42 }
43
44 return fla;
45 }
46
47 int flaIsGuardian(struct FibLikeArray fla) {
48     if (fla.a0 == -1) {
49         return 1;
50     } else {
51         return 0;
52     }
53 }
54
55 int flaGet(struct FibLikeArray fla, int index) {
56     if (flaIsGuardian(fla)) {
57         return -1;
58     }
59
60     if (index <= 32) {
61         return fla._buf[index];
62     }
63
64     {
65         int e0 = fla._buf[32 - 2];
66         int e1 = fla._buf[32 - 1];

```

```

67     int er;
68     int ei = 32;
69     int er = ei;
70
71     while (ei <= index) {
72         er = e0 + e1;
73         e0 = e1;
74         e1 = er;
75         ei = ei + 1;
76     }
77
78     er = e0 + e1;
79     er = er * 193;
80
81     return er;
82 }
83 }
84
85 int main() {
86     struct FibLikeArray fla = makeFibLikeArray(0, 1);
87     int m0, m1, mr;
88
89     if (flaIsGuardian(fla)) {
90         return 1;
91     }
92
93     m0 = flaGet(fla, 10);
94     m1 = flaGet(fla, 37);
95     mr = m1 * mr;
96
97     return 0;
98 }

```

输出

```
1 Error type 1 at line 38: Undefined variable "ci"
2 Error type 1 at line 39: Undefined variable "ci"
3 Error type 1 at line 40: Undefined variable "ci"
4 Error type 3 at line 69: Redefined variable "er"
```

说明：仅 2.2 分组同学需要测试该用例，需要输出上述的错误信息，其中 69 行的错误也可以报在 67 行。

5.3 E2.3

这组测试用例针对 2.3 分组的同学。

输入

```
1 struct IntArray10 {
2     int iaa10[10];
3 };
4
5 struct IntArray100 {
6     int iaa100[100];
7 };
8
9 struct IntVector {
10    int ix, iy;
11    int ia[2];
12    struct {
13        struct IntArray10 id[10];
14        struct IntArray100 ie;
15    } ib;
16 };
17
18 struct FloatVector {
19     int fx, fy;
20     int fa[39];
21     struct {
```

```

22     struct IntArray100 fd;
23     struct IntArray10 fe[10];
24 } fb;
25 };
26
27 int vec_is_equal(struct IntVector iv, struct IntVector iw) {
28     if (iv.ix == iw.ix && iv.iy == iw.iy) {
29         return 1;
30     } else {
31         return 0;
32     }
33 }
34
35 int main() {
36     struct IntVector v1;
37     struct FloatVector v2;
38
39     if (!vec_is_equal(v1, v2)) {
40         v2 = v1;
41     }
42
43     return 0;
44 }

```

输出

```

1 Error type 9 at line 39: Function arguments do not match
2 Error type 5 at line 40: Type mismatch

```

说明：仅 2.3 分组同学需要测试该用例，需要输出上述的错误信息。

6 结束语

如果对本测试用例有任何疑议，可以写邮件与李聪助教或陈紫琦助教联系，注意同时抄送给许老师。