

编译原理第二次实验测试用例：目录

1	A 组测试用例	3
1.1	A-1	3
1.2	A-2	3
1.3	A-3	4
1.4	A-4	5
1.5	A-5	5
1.6	A-6	6
1.7	A-7	7
1.8	A-8	8
1.9	A-9	9
1.10	A-10	9
1.11	A-11	10
1.12	A-12	11
1.13	A-13	12
1.14	A-14	13
1.15	A-15	13
1.16	A-16	14
1.17	A-17	15
1.18	A-18	16
1.19	A-19	17
1.20	A-20	17
2	B 组测试用例	18
2.1	B-1	18
2.2	B-2	20
3	C 组测试用例	22
3.1	C-1	22
3.2	C-2	23
4	D 组测试用例	25
4.1	D-1	25
4.2	D-2	26
4.3	D-3	28

5	E 组测试用例	29
5.1	E-1	29
5.2	E-2	30
5.3	E-3	32
6	结束语	33

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 7, 9, 15。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

1.1.1 输入

```
1 struct Student {
2     int id;
3     float grade;
4 };
5
6 int main() {
7     struct Student student1, student2;
8
9     student1.id = 1;
10    student1.grade = 85.5;
11    student2.id = 2;
12    student2.grade = unknown;
13
14    return 0;
15 }
```

1.1.2 输出

```
1 Error type 1 at Line 12: undefined variable: unknown.
```

1.1.3 说明

第 12 行中，unknown 这个变量没有被定义过。这里可以多报一个 5 型错误。

1.2 A-2

1.2.1 输入

```
1 int main() {
2     int num1, num2, sum;
3     sum = addNumbers(num1, num2);
4     return 0;
5 }
```

1.2.2 输出

```
1 Error type 2 at Line 3: using a undefined function.
```

1.2.3 说明

第 3 行中，函数addNumbers没有被定义过。这里可以多报一个 5 型错误。

1.3 A-3

1.3.1 输入

```
1 struct Car {
2     int year;
3     float price;
4 };
5
6 int main() {
7     struct Car car1;
8     int car1;
9
10    car1.year = 2022;
11    car1.price = 25000.0;
12
13    return 0;
14 }
```

1.3.2 输出

```
1 Error type 3 at Line 8: Redefined variable "car1".
```

1.3.3 说明

第 8 行局部变量的名称car1和第 7 行的重复了。错误也可以报在第 7 行。

1.4 A-4

1.4.1 输入

```
1 int addNumbers(int a, int b) {
2     int result = a + b;
3     return result;
4 }
5
6 int main() {
7     int num1, num2, sum;
8     sum = addNumbers(num1, num2);
9     return 0;
10 }
11
12 int addNumbers(int _a, int _b) {
13     return _a + _b;
14 }
```

1.4.2 输出

```
1 Error type 4 at Line 12: Redefined function.
```

1.4.3 说明

第 12 行定义的函数addNumbers和第 1 行定义的函数重名了。错误也可以报在第 1 行。

1.5 A-5

1.5.1 输入

```
1 struct Student {
2     int id;
3     float grade;
4 };
5
```

```

6  int main() {
7      struct Student student1, student2;
8
9      student1.id = 1;
10     student1.grade = 85.5;
11     student2.id = 2;
12     student2.grade = student1.id;
13
14     return 0;
15 }

```

1.5.2 输出

```

1  Error type 5 at Line 12: Type mismatched for assignment.

```

1.5.3 说明

第 12 行中，赋值表达式两边的变量类型不一致，不能把一个整形变量赋值给一个浮点数变量。

1.6 A-6

1.6.1 输入

```

1  struct Rectangle {
2      float length;
3      float width;
4  };
5
6  float calculateArea(struct Rectangle rect1) {
7      float area = rect1.length * rect1.width;
8      return area;
9  }
10
11 float calculatePerimeter(struct Rectangle rect2) {
12     float perimeter = 2.0 * (rect2.length + rect2.width);
13     return perimeter;
14 }

```

```

15
16 int main() {
17     struct Rectangle myRect;
18
19     myRect.length = 5.0;
20     myRect.width = 3.0;
21
22     myRect.length + myRect.width = 8.0;
23
24     return 0;
25 }

```

1.6.2 输出

```

1 Error type 6 at Line 22: Invalid left value.

```

1.6.3 说明

第 22 行中，两个浮点数之和不能放在赋值号的左边。

1.7 A-7

1.7.1 输入

```

1 struct Rectangle {
2     float length;
3     float width;
4 };
5
6 float calculateArea(struct Rectangle rect1) {
7     return rect1.length * rect1.width;
8 }
9
10 float calculatePerimeter(struct Rectangle rect2) {
11     return 2.0 * (rect2.length + rect2.width);
12 }
13
14 int main() {

```

```

15     float numbers[5];
16     struct Rectangle myRect;
17
18     myRect.length = numbers[0];
19     myRect.width = numbers[1] + numbers;
20
21     return 0;
22 }

```

1.7.2 输出

```

1 Error type 7 at Line 19: unmatched operands.

```

1.7.3 说明

第 19 行中，不能把一个浮点型变量与一个数组相加。这里可以多报一个 5 型错误。

1.8 A-8

1.8.1 输入

```

1 int increase(float a) {
2     return a + 1.0;
3 }
4
5 int main() {
6     int num1;
7     float num2;
8     num1 = increase(num2);
9     return 0;
10 }

```

1.8.2 输出

```

1 Error type 8 at Line 2: Type mismatched for return.

```

1.8.3 说明

第 2 行中，实际的返回值类型 float 和声明的返回值类型 int 不一致。

1.9 A-9

1.9.1 输入

```
1 struct Pet {
2     int age;
3     float weight;
4 };
5
6 int is_older(struct Pet p1, struct Pet p2) { return p1.age > p2.age;
7     }
8
9 int main() {
10     struct Pet a, b, c;
11
12     a.age = 5;
13     a.weight = 12.5;
14     b.age = 10;
15     b.weight = 0.6;
16     c.age = 7 - 6;
17     c.weight = 3.5;
18
19     is_older(a, b, c);
20
21     return 0;
22 }
```

1.9.2 输出

```
1 Error type 9 at Line 18: Function "is_older(Pet,Pet)" is not
   applicable for arguments "(Pet,Pet,Pet)".
```

1.9.3 说明

第 18 行中，函数 `is_older` 的实参数量与形参数量不符。

1.10 A-10

1.10.1 输入

```

1  int main() {
2      int arr[10];
3      int size = 10;
4      int i = 0;
5      int j = 0;
6
7      while (i < size - 1) {
8          while (j < size - i - 1) {
9              if (arr[j] > arr[j + 1]) {
10                 int temp = arr[j];
11                 size[j] = arr[j + 1];
12                 arr[j + 1] = temp;
13             }
14             j = j + 1;
15         }
16         i = i + 1;
17     }
18
19     return 0;
20 }

```

1.10.2 输出

```

1  Error type 10 at Line 11: using [...] on a non-array variable.

```

1.10.3 说明

第 11 行中，对非数组类型的变量 `size` 使用了数组索引符号 `[]`。这里可以多报一个 5 型错误。

1.11 A-11

1.11.1 输入

```

1  int addNumbers(int a, int b) {
2      int result = a + b;
3      return result;
4  }

```

```

5
6 int main() {
7     int num1, num2, sum;
8     num1(num2);
9     return 0;
10 }

```

1.11.2 输出

```

1 Error type 11 at Line 8: "num1" is not a function.

```

1.11.3 说明

第 8 行中，对非函数类型的变量num1使用了函数调用符号(...)。

1.12 A-12

1.12.1 输入

```

1 int main() {
2     int arr[10];
3     int size = 10;
4     int i = 0;
5     int j = 0;
6     int max;
7
8     while (i < size - 1) {
9         while (j < size - i - 1) {
10             if (arr[j] > arr[j + 1]) {
11                 int temp = arr[j];
12                 arr[j] = arr[j + 1];
13                 arr[j + 1] = temp;
14             }
15             j = j + 1;
16         }
17         i = i + 1;
18     }
19     max = arr[9.0];

```

```
20     return 0;
21 }
```

1.12.2 输出

```
1 Error type 12 at Line 19: Non-integer index.
```

1.12.3 说明

第 19 行中，不能使用float类型的变量作为数组的索引。可以多报一个 5 型错误。

1.13 A-13

1.13.1 输入

```
1 struct Pet {
2     int age;
3     float weight;
4 };
5
6 int is_older(struct Pet p1, int p2) { return p1.age > p2.age; }
7
8 int main() {
9     struct Pet a;
10
11     a.age = 5;
12     a.weight = 12.5;
13
14     return 0;
15 }
```

1.13.2 输出

```
1 Error type 13 at Line 6: using . on a non-structure variable.
```

1.13.3 说明

第 6 行中，对整形变量使用了.操作符。这里可以多报一个 7 型错误和一个 8 型错误。

1.14 A-14

1.14.1 输入

```
1 struct Rectangle {
2     float length;
3     float width;
4 };
5
6 float calculateArea(struct Rectangle rect) {
7     float area = rect.length * rect.width;
8     return area;
9 }
10
11 int main() {
12     struct Rectangle myRect;
13
14     myRect.length = 5.0;
15     myRect.width = 3.0;
16
17     myRect.area = calculateArea(myRect);
18
19     return 0;
20 }
```

1.14.2 输出

```
1 Error type 14 at Line 17: Trying to visit a structure field which is
   undefined.
```

1.14.3 说明

第 17 行中，访问了未定义的域area。这里可以多报一个 5 型错误。

1.15 A-15

1.15.1 输入

```

1 struct Dish {
2     float price;
3     struct {
4         int count;
5         int positive;
6         int negative;
7         int count;
8     } comments;
9 };
10
11 int main() {
12     struct Dish myDish;
13     myDish.price = 12.99;
14
15     return 0;
16 }

```

1.15.2 输出

```

1 Error type 15 at Line 7: Redefined field.

```

1.15.3 说明

第 7 行中，count 与第 4 行重复。该错误可以报在第 4 行。

1.16 A-16

1.16.1 输入

```

1 struct Teacher {
2     int id;
3     int age;
4     float salary;
5 };
6
7 struct Student {
8     int id;
9     int age;

```

```

10     float grade;
11 };
12
13 int main() {
14     struct Student {
15         int id;
16         int age;
17         int year;
18     } students[10];
19
20     return 0;
21 }

```

1.16.2 输出

```

1 Error type 16 at Line 14: Duplicated name: Student.

```

1.16.3 说明

第 14 行中，定义的结构体 `Student` 和已经定义过的结构体重名了，也可以报在第 7 行。可以多报与 `struct Student` 相关的错误。

1.17 A-17

1.17.1 输入

```

1 struct Teacher {
2     int id;
3     int age;
4     float salary;
5 };
6
7 int main() {
8     struct Student students[10];
9
10    return 0;
11 }

```

1.17.2 输出

```
1 Error type 17 at Line 8: Undefined structure "Student".
```

1.17.3 说明

第 8 行中，使用了未被定义的结构体类型Student。

1.18 A-18

1.18.1 输入

```
1 struct Rectangle {
2     float length;
3     float width;
4 };
5
6 float calculateArea(struct Rectangle rect1) {
7     return rect1.length * rect1.width;
8 }
9
10 float calculatePerimeter(struct Rectangle rect2) {
11     return 2 * (rect2.length + rect2.width);
12 }
13
14 int main() {
15     struct Rectangle myRect;
16     return 0;
17 }
```

1.18.2 输出

```
1 Error type 7 at Line 11: unmatched operands.
```

1.18.3 说明

第 11 行中，int不能和float相乘。可以多报一个 8 型错误。

1.19 A-19

1.19.1 输入

```
1 struct Pet {
2     int age;
3     float weight;
4 };
5
6 int is_older(struct Pet p1, struct Pet p2) { return p1.age > p2.age;
7     }
8
9 int main() {
10     struct Pet a, b;
11
12     a.age = 5;
13     a.weight = 12.5;
14     b.age = 10;
15     b.weight = 0.6;
16
17     is_older(a, b.age);
18
19     return 0;
20 }
```

1.19.2 输出

```
1 Error type 9 at Line 16: unmatched parameters when calling function:
   is_older .
```

1.19.3 说明

第 16 行中，函数的实参类型与形参类型不匹配。

1.20 A-20

1.20.1 输入

```

1 struct Dish {
2     float price;
3     struct {
4         int count = 0;
5         int positive;
6         int negative;
7     } comments;
8 };
9
10 int main() {
11     struct Dish myDish;
12     myDish.price = 12.99;
13
14     return 0;
15 }

```

1.20.2 输出

```

1 Error type 15 at Line 4: initializing a field in structure.

```

1.20.3 说明

第 4 行中，结构体在定义时不能对域进行初始化。

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

2.1.1 输入

```

1 struct DataPoint {
2     float x;
3     float y = 0.0;
4 };

```

```

5
6 struct DataPoint dataPoints[5];
7 int numDataPoints;
8
9 float calculateAverage() {
10     float sum = 0.0;
11     int i = 0;
12     numDataPoints[0] = 5;
13     while (i < numDataPoints) {
14         sum = sum + dataPoints[i].y;
15         i = i + 1;
16     }
17     return sum / numDataPoints;
18 }
19
20 int main() {
21     float average, maximum;
22     dataPoints[0].x = 1.0;
23     dataPoints[0].y = 2.5;
24
25     dataPoints[1].x = 2.0;
26     dataPoints[1].y = 3.7;
27
28     dataPoints[2].x = 3.0;
29     dataPoints[2].y = 1.8;
30
31     dataPoints[3].x = 4.0;
32     dataPoints[3].y = 4.2;
33
34     dataPoints[4].x = 5.0;
35     dataPoints[4].y = 2.1;
36
37     average = calculateAverage();
38
39     return 0;
40 }

```

2.1.2 输出

```
1 Error type 15 at Line 3: initializing a field in structure.
2 Error type 10 at Line 12: using [...] on a non-array variable.
3 Error type 7 at Line 17: unmatched operands.
```

2.1.3 说明

第 3 行中，在结构体定义时不能对域进行初始化；第 12 行中，不能对非数组类型变量使用数组访问操作符，此处可以多报一个 5 型错误；第 17 行中，操作数类型不匹配，此处可以多报一个 8 型错误；

2.2 B-2

2.2.1 输入

```
1 struct Product {
2     int id;
3     float price;
4     int quantity;
5 };
6
7 struct Product inventory[5];
8 int numProducts;
9
10 float calculateInventoryValue(int numProducts1) {
11     float totalValue = 0.0;
12     int i = 0;
13     while (i < numProducts1) {
14         totalValue = totalValue + inventory[i].price * inventory[i].
            quantity;
15         i = i + 1;
16     }
17     return totalValue;
18 }
19
20 int findProductById(int productId, int numProducts2) {
21     int j = 0;
22     while (j < numProducts2) {
```

```

23     if (inventory[j].id == productId) {
24         return j;
25     }
26     j = j + 1;
27 }
28 return -1;
29 }
30
31 float main() {
32     int productIdToFind = 3;
33     float inventoryValue;
34     numProducts = 5;
35
36     inventory.id = 1;
37     inventory[0].price = 19.99;
38     inventory[0].quantity = 10;
39
40     inventory[1].id = 2;
41     inventory[1].price = 49.99;
42     inventory[1].quantity = 5;
43
44     inventory[2].id = 3;
45     inventory[2].price = 79.99;
46     inventory[2].quantity = 3;
47
48     inventory[3].id = 4;
49     inventory[3].price = 4.99;
50     inventory[3].quantity = 20;
51
52     inventory[4].id = 5;
53     inventory[4].price = 14.99;
54     inventory[4].quantity = 8;
55
56     inventoryValue = calculateInventoryValue(numProducts);
57     findProductById(productIdToFind, numProducts) = productIdToFind;
58     return 0;
59 }

```

2.2.2 输出

```
1 Error type 7 at Line 14: unmatched operands.
2 Error type 13 at Line 36: using . on a non-structure variable.
3 Error type 6 at Line 57: LHS are a right-value-only Expression.
4 Error type 8 at Line 58: the return value contradicts the difinition
  of the function.
```

2.2.3 说明

第 14 行中，操作数类型不匹配，可以多报一个 5 型错误；第 36 行中，不能直接对数组类型的变量赋值，可以多报一个 5 型错误；第 57 行中，函数调用不能作为左值；第 58 行中，返回值类型与声明的类型不匹配。

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

3.1.1 输入

```
1 struct Coefficients {
2     float c1, c2, c3, c4;
3 } coefficients;
4
5 float evaluatePolynomial(struct Coefficients c, int degree, float x)
6 {
7     float res[1], _c;
8     int i = degree - 1;
9
10    if (degree == 1)
11        res[0] = c.c1;
12    else if (degree == 2)
13        res[0] = c.c2;
14    else if (degree == 3)
15        res[0] = c.c3;
16    else
17        res[0] = c.c4;
```

```

17
18     while (i >= 0) {
19         if (i == 1)
20             _c = c.c1;
21         else if (i == 2)
22             _c = c.c2;
23         else if (i == 3)
24             _c = c.c3;
25         else
26             _c = c.c4;
27         res[0] = res[0] * x + _c;
28     }
29     return res[0];
30 }
31
32 int main() {
33     int d = 4 / 4 - 1;
34     float f = 2.5;
35
36     float result = evaluatePolynomial(coefficients, d, f);
37
38     coefficients.c1 = 2.0;
39     coefficients.c2 = 3.0;
40     coefficients.c3 = 4.0;
41     coefficients.c4 = 5.0;
42
43     return d >= 1;
44 }

```

3.1.2 输出

```

1 // 正常返回，没有任何输出。

```

3.2 C-2

3.2.1 输入

```

1 int factorial(int n) {

```

```

2   if (n == 0)
3       return 1;
4   else
5       return n * factorial(n - 1);
6   }
7
8   int power(int base, int exponent) {
9       int result = 1;
10      int i = 0;
11      while (i < exponent) {
12          result = result * base;
13          i = i + 1;
14      }
15      return result;
16  }
17
18  int sinApproximation(int x, int terms) {
19      int result2 = 0;
20      int sign = 1, _n = 0;
21      while (_n < terms) {
22          int numerator = power(x, (2 * _n) + 1);
23          int denominator = factorial((2 * _n) + 1);
24          result2 = result2 + (sign * numerator) / denominator;
25          sign = (-1) * sign;
26          _n = _n + 1;
27      }
28      return result2;
29  }
30
31  int main() {
32      int value = 1;
33      int t = 10;
34
35      int sinValue = sinApproximation(value, t);
36
37      return 0;
38  }

```


3.2.2 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

4.1.1 输入

```
1 struct Point {
2     int x;
3     int y;
4 };
5
6 struct Point shift_point(struct Point c, int d, int e);
7
8 struct Location {
9     int a;
10    int b;
11 };
12
13 struct Rectangle {
14     struct Point topLeft;
15     struct Point bottomRight;
16 };
17
18 struct Point shift_point(struct Point p, int plus_x, int plus_y) {
19     struct Point _p;
20     _p.x = p.x + plus_x;
21     _p.y = p.y + plus_y;
22     return _p;
23 }
24
25 struct Rectangle shift_rectangle(struct Rectangle r, int to_east,
```

```

26         int to_north) {
27     struct Rectangle _r;
28     _r.topLeft = shift_point(r.topLeft, to_east, to_north);
29     _r.bottomRight = shift_point(r.bottomRight, to_east, to_north);
30     return _r;
31 }
32
33 int main() { return 0; }
34
35 struct Rectangle shift_rectangle(struct Rectangle f, int g, int h);

```

4.1.2 输出

```

1 // 正常返回，没有任何输出。

```

4.1.3 说明

3.1 分组的同学没有任何输出，其它分组的同学在第 6 行、第 35 行报语法错误。

```

1 Error Type B at Line 6: syntax error, unexpected SEMI, expecting LC.
2 Error Type B at Line 35: syntax error, unexpected SEMI, expecting LC.

```

4.2 D-2

4.2.1 输入

```

1 int factorial(int n) {
2     if (n == 0)
3         return 1;
4     else
5         return n * factorial(n - 1);
6 }
7
8 int power(int base, int exponent) {
9     int result = 1;
10    int i = 0;
11    while (i < exponent) {
12        result = result * base;
13        i = i + 1;

```

```

14     }
15     return result;
16 }
17
18 int sinApproximation(int x, int terms) {
19     int result = 0;
20     int sign = 1, n = 0;
21     while (n < terms) {
22         int numerator = power(x, (2 * n) + 1);
23         int denominator = factorial((2 * n) + 1);
24         result = result + (sign * numerator) / denominator;
25         sign = (-1) * sign;
26         n = n + 1;
27     }
28     return result;
29 }
30
31 int main() {
32     int x = 1;
33     int terms = 10;
34     int sinValue = sinApproximation(x, terms);
35
36     return 0;
37 }

```

4.2.2 输出

```

1 // 正常返回，没有任何输出。

```

4.2.3 说明

3.2分组的同学没有任何输出。其它分组的同学应该识别出对于变量result, n, x, terms的重复定义。

```

1 Error type 3 at Line 19: Redefined variable "result".
2 Error type 3 at Line 20: Redefined variable "n".
3 Error type 3 at Line 32: Redefined variable "x".
4 Error type 3 at Line 33: Redefined variable "terms".

```

4.3 D-3

4.3.1 输入

```
1 struct Point {
2     int x;
3     int y;
4 };
5
6 struct Location {
7     int a;
8     int b;
9 };
10
11 struct Rectangle {
12     struct Point topLeft;
13     struct Point bottomRight;
14 };
15
16 struct Location shift_point(struct Point p, int plus_x, int plus_y) {
17     struct Point _p;
18     _p.x = p.x + plus_x;
19     _p.y = p.y + plus_y;
20     return _p;
21 }
22
23 struct Rectangle shift_rectangle(struct Rectangle r, int to_east,
24                                 int to_north) {
25     struct Rectangle _r;
26     _r.topLeft = shift_point(r.topLeft, to_east, to_north);
27     _r.bottomRight = shift_point(r.bottomRight, to_east, to_north);
28     return _r;
29 }
30
31 int main() { return 0; }
```

4.3.2 输出

```
1 // 正常返回，没有任何输出。
```

4.3.3 说明

3.3 分组的同学没有任何输出，其它分组的同学应该在第 20 行、第 26 行、第 27 行报错。

```
1 Error type 8 at Line 20: the return value contradicts the definition  
  of the function.  
2 Error type 5 at Line 26: Type mismatched for assignment.  
3 Error type 5 at Line 27: Type mismatched for assignment.
```

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 E-1

5.1.1 输入

```
1 struct Point {  
2     int x;  
3     int y;  
4 };  
5  
6 struct Point shift_point(struct Point c, int d, int e);  
7  
8 struct Location {  
9     int a;  
10    int b;  
11 };  
12  
13 struct Rectangle {  
14     struct Point topLeft;  
15     struct Point bottomRight;  
16 };  
17  
18 struct Point shift_point(struct Point p, int plus_x, int plus_y) {  
19     struct Point _p;
```

```

20  _p.x = p.x + plus_x;
21  _p.y = p.y + plus_y;
22  return _p;
23  }
24
25  struct Rectangle shift_rectangle(struct Rectangle f, int g, int h);
26
27  int main() {
28      struct Rectangle r1, r2;
29      r2 = shift_rectangle(r1, 1, 2);
30
31      return 0;
32  }
33
34  struct Location shift_point(struct Point c, int d, int e);

```

5.1.2 输出

```

1 Error type 19 at Line 34: Inconsistent declaration of function "
  shift_point".
2 Error type 18 at Line 25: Undefined function "shift_rectangle".

```

5.1.3 说明

仅 3.1 分组的同学需要测试这个用例，并且报出错误。错误 19 也可以报在第 18 行。
若选做内容全部完成，则不需要报错误类型 19，这种情况的报错信息如下：

```

1 Error type 18 at Line 25: Undefined function "shift_rectangle".

```

5.2 E-2

5.2.1 输入

```

1 int factorial(int n) {
2     if (n == 0)
3         return 1;
4     else
5         return n * factorial(n - 1);
6 }

```

```

7
8 int power(int base, int exponent) {
9     int result = 1;
10    int n = 0;
11    while (n < exponent) {
12        result = result * base;
13        n = n + 1;
14    }
15    return result;
16 }
17
18 int sinApproximation(int x, int terms) {
19     int result = 0;
20     int sign = 1, n = 0;
21     while (n < terms) {
22         int numerator = power(x, (2 * n) + 1);
23         int denominator = factorial((2 * n) + 1);
24         result = result + (sign * numerator) / denominator;
25         sign = (-1) * sign;
26         n = n + 1;
27     }
28     return result;
29 }
30
31 int main() {
32     int x = 1;
33     int terms = 10;
34
35     int sinValue = sinApproximation(x, terms);
36
37     return 0;
38 }

```

5.2.2 输出

```

1 // 正常返回，没有任何输出。

```

5.2.3 说明

仅 3.2 分组的同学需要测试这个用例。

5.3 E-3

5.3.1 输入

```
1 struct Avg {
2     int r1;
3     struct {
4         int v2;
5         int weight;
6     } items[10];
7 };
8
9 struct Sum {
10    struct {
11        int v1;
12        int coefficient;
13    } parts[10];
14    int r2;
15 };
16
17 int calculate_sum(struct Sum sum) {
18     int i = 0;
19     sum.r2 = 0;
20     while (i < 10) {
21         sum.r2 = sum.r2 + sum.parts[i].coefficient * sum.parts[i].v1;
22         i = i + 1;
23     }
24     return sum.r2;
25 }
26
27 int calculate_avg(struct Avg avg) {
28     int s = calculate_sum(avg);
29     avg.r1 = s / 10;
30     return avg.r1;
31 }
```



```
32  
33 int main() {  
34     struct Avg a1;  
35     calculate_avg(a1);  
36     return 0;  
37 }
```

5.3.2 输出

```
1 Error type 9 at Line 28: Arguments are not applicable to function "  
    calculate_sum".
```

5.3.3 说明

仅 3.3 分组的同学需要测试这个用例，并且报出错误。

6 结束语

若对本文档有任何疑议，可写邮件与周意可助教联系，注意同时抄送给许畅老师。