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	6
	1.6 A-6	7
	1.7 A-7	7
	1.8 A-8	8
	1.9 A-9	9
	1.10 A-10	10
	1.11 A-11	11
	1.12 A-12	12
	1.13 A-13	13
	1.14 A-14	14
	1.15 A-15	15
	1.16 A-16	16
	1.17 A-17	16
	1.18 A-18	17
	1.19 A-19	18
	1.20 A-20	19
•		20
2	B 组测试用例	20
	2.1 B-1	
	2.2 B-2	22
3	C 组测试用例	24
	3.1 C-1	24
	3.2 C-2	26
		40
4	D组测试用例	28

6	结束语	37
	5.3 E2.3	35
	5.2 E2.2	33
	5.1 E2.1	32
5	E组测试用例	32
	4.3 D-3	30
	4.2 D-2	29

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

输入

```
int main(int x1, int x2) {
    int p = x1;
    int i = 3;
    int k = x2 + p;
    i = x1 + x3;
    return k;
}
```

输出

```
Error type 1 at Line 5: Undefined variable "x3"
```

说明: 说明: i = x1 + x3 这一句包含未定义的变量 x3,这里也可另外报出错误类型 5 (= 两边类型不匹配)。

1.2 A-2

```
struct Student{
    int id;
    int weight;
    int grades;
};

int newStudent(int a, int b) {
```

```
struct Student st;
            st.id = a;
            st.weight = b;
10
            st.grades = 0;
11
            return 0;
12
13
14
   int main() {
15
            int k = 3;
16
            int p = 14;
17
            newStdent(k, p);
18
            return 0;
19
20
```

```
Error type 2 at Line 18: Undefined function "newStdent"
```

说明: newStdent 未定义。

1.3 A-3

```
struct Danger{
           int what;
2
           int up;
3
  }d;
5
  int main(){
           struct Danger x1;
7
           int x2 = 3;
           float Danger = 1.2;
           x1.what = x2;
10
           x1.up = 0;
11
           return x1.what;
```

```
13 }
```

```
Error type 3 at Line 9: Redefined variable "Danger"
```

说明: 重复定义的变量 Danger, 如果在第一行报错也可以。

1.4 A-4

输入

```
int compare(int a, int b) {
           if(a > b) {
2
                    return 1;
3
           }
           if(a == b) {
                    return 0;
6
           return -1;
10
  float compare(int x1, int x2) {
11
           if (x1 > x2) {
12
                    return 1.0;
13
14
           if (x2 == x1) {
15
                   return 0.0;
           return -1.0;
19
20
  int main() {
21
           return compare(3,4);
```

输出

```
Error type 4 at Line 11: Redefined function "compare"
```

说明: 重复定义的函数 compare。在第一行报错也可以。

1.5 A-5

输入

```
struct Cake{
           int name;
2
           float weight;
3
            int size;
  }c;
  struct Candle{
            struct Cake cc;
            int number;
            float length;
10
  }can;
11
12
   int main() {
13
            int wei = 3;
14
           int s = 2;
15
           int num = 15;
           float len = 1.2;
17
           can.cc = c;
18
           c.name = 0;
19
           c.weight = wei;
20
           c.size = s;
21
           can.number = num;
22
           can.length = len;
23
           return 0;
```

输出

```
Error type 5 at Line 20: Type mismatched for assignment.
```

说明: 第 20 行将 int 型赋值给 float 型。

1.6 A-6

输入

```
int add(int a, int b) {
    return a + b;
}

int main() {
    int i = 10;
    int sum = 0;
    while (i > 0) {
        sum = add(sum, i);
        add(i, -1) = i;
}

return sum;
}
```

输出

Error type 6 at Line 10: The left-hand side of an assignment must be a variable.

说明: 第10行赋值语句左值为 int 型(函数调用返回结果),无法作为左值。

1.7 A-7

```
struct IntArray{
    int id;
    int ia[10];

4 };
```

```
struct FloatArray{
            int name;
            float fa[10];
   };
10
   int main(){
            struct IntArray inta;
12
            struct FloatArray floa;
13
            int i = 0;
14
            int k = 1;
            float p = 1.0;
16
            while (i < 10) {
17
                     i = i + k;
18
                     inta.ia[i] = k;
19
                     k = k + 1;
20
                     floa.fa[i] = p;
21
                     p = p + 1;
22
            }
23
            return 0;
24
25
```

```
Error type 7 at Line 22: Type mismathced for operands.
```

说明: 第 22 行 p 为 float 型, 1 为 int 型, 无法匹配。也可以多报一个 5 型错误。

1.8 A-8

```
int compareInt(int a, int b) {
    if (a == b) {
        return 0;
    } else {
        return -1;
}
```

```
}
8
  float compareFloat(float aa, float bb) {
            return aa - bb;
10
12
   int main() {
13
            int i = 3;
14
            int j = 4;
            float p = 1.2;
16
            float k = 2.3;
17
            if(i == 4) {
18
                     return compareInt(i,j);
19
            } else {
20
                     return compareFloat(p,k);
21
            }
22
23
```

```
Error type 8 at Line 21: Type mismatched for return.
```

说明:第 21 行 compareFloat 返回 float 型,与 main 函数定义 return 类型冲突。

1.9 A-9

```
int compareInt(int a, int b) {
    if (a == b) {
        return 0;
    } else {
        return -1;
    }
}
```

```
float compareFloat(float aa, float bb) {
10
            return aa - bb;
11
12
   int main() {
            int i = 3;
14
            int j = 4;
15
            float p = 1.2;
16
            float k = 2.3;
17
            if(i == 4) {
18
                     return compareInt(i,j);
19
            } else {
20
                     return compareInt(j,k);
21
            }
22
23
```

```
Error type 9 at Line 21: Function arguments do not match.
```

说明: 第 21 行函数调用第二个参数类型不符(应为 int, 使用了 float)。

1.10 A-10

```
struct Paper{
    int name;
    int words[100];
}pp;

int main() {
    struct Paper papers[10][10];
    int i = 0;
    int j = 0;
```

```
while (i < 10) {
    pp.words[i+j] = j;
    while (j < 10) {
        papers[i][j].words[i] = i + j;
        j = j + 1;
    }
    i = i + 1;
}
return pp[i+j];
</pre>
```

```
Error type 10 at Line 18: Variable is not an array.
```

说明: 第 18 行错误地对 pp 进行数组寻址。也可以多报一个 8 型错误。

1.11 A-11

```
struct Pad{
           int size;
2
           int storage;
  };
5
  int iPad(int ss, int st) {
           struct Pad ip;
           ip.size = ss;
8
           ip.storage = st;
           return ss;
10
11
  int miPad(int s, int t) {
13
           struct Mi{
14
                    struct Pad core;
15
```

```
int brand;
}myMi;
myMi.core = myMi(s,t);
myMi.brand = 0;
return 0;

int main() {
    return miPad(3,4);
}
```

```
Error type 11 at Line 18: "myMi" is not a function.
```

说明: 第18行对变量进行函数调用。也可以多报一个5型错误

1.12 A-12

```
int main() {
           int fib[100];
2
           int i = 0, a = 1, b = 1;
           int tem;
           float floatFib[100];
5
           float fa = 1.0, fb = 1.0;
           float ftem;
           while (i < 100) {
8
                   fib[i] = a + b;
                   tem = a + b;
10
                   a = b;
11
                   b = tem;
                   i = i + 1;
13
           }
14
           i = 0;
```

```
while (i < 100) {
    ftem = fa + fb;
    floatFib[ftem] = ftem;

fa = fb;

fb = ftem;

i = i + 1;

return i;

// Page 100 |
// Page 100
```

```
Error type 12 at Line 18: Array index is not an integer.
```

说明: 第 18 行使用 ftem 作为数组寻址, 其为 float 型。

1.13 A-13

```
struct Building{
           struct base{
2
                   int size;
                   float height;
           }b;
5
           int bh;
  }Yangzhou;
  int main(){
           struct Building newOne;
10
           Yangzhou.b.size = 1;
11
           Yangzhou.b.height = 3.4;
12
           Yangzhou.bh = 100;
           newOne.b.size = Yangzhou.b.size;
14
           newOne.bh.height = Yangzhou.b.height;
15
           newOne.bh = Yangzhou.bh;
```

```
return 0;
18 }
```

```
Error type 13 at Line 15: Illegal use of "."
```

说明: 第15行对变量 bh 进行结构体访问。

1.14 A-14

```
struct Javalin{
           int head;
2
           int shoulder;
           float foot;
           struct Pilot{
5
                    int gender;
                    int age;
           }freelancer;
  };
10
  struct Anthem{
11
           struct Javalin army[100];
12
           int missions[100];
13
  }game;
14
15
  int main() {
16
           int i = 0;
17
           struct Javalin lin;
18
           while(i < 100) {
19
                    game.army[i] = lin;
                    i = i + 1;
21
                    lin.head = i;
22
                    lin.shoulder = i * i;
23
```

```
Error type 14 at Line 25: None-existent field "age".
```

说明: 第 25 行错误访问结构体 Javalin 中不存在的域 age。

1.15 A-15

输入

```
struct Food {
           int cal;
2
           int name, price;
           float name1;
  };
  int main() {
           struct person {
                    struct Food burger;
                    int age;
10
                    float time, burger;
11
           }p;
12
           return 0;
13
14
```

输出

```
Error type 15 at Line 11: Invalid definition in structure.
```

说明: person 结构体定义时存在重名域 burger。

1.16 A-16

输入

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

输出

```
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。

1.17 A-17

```
struct Spider {
            int legs;
            float weight;
  }man;
  struct Venom{
           int height;
           int gender;
8
   }edi;
10
  struct Carnage{
           int number;
12
            float sex;
13
  }end;
14
15
  int main() {
16
           struct Spider p;
17
           struct Venom vv;
18
           struct Cranage cc;
           return 0;
20
21
```

```
Error type 17 at Line 19: Undefined structure "Cranage".
```

说明: 第19行使用了未定义的结构体 Cranage。

1.18 A-18

```
struct Apple {
    int weight;
    float round;
};
```

```
5
  struct Pen {
            int length;
            int used;
  } ;
10
   int add(struct Apple a, struct Pen p) {
11
            return a.weight + p.length;
12
13
14
   int main() {
15
            struct Apple aa;
16
            struct Pen pp;
17
            int sum;
18
            aa.weight = 3;
19
            pp.length = 12;
20
            sum = add(aa, pp);
21
            if(sum > 5) {
22
                     return aa / pp.length;
23
            } else {
24
                     return 0;
25
            }
26
27
```

```
Error type 7 at Line 23: Type mismathced for operands.
```

说明: 第 23 行使用结构体变量 aa 与 int 型进行除法,无法匹配。

1.19 A-19

```
int compare1(int a, int b) {
   if(a==b) {
```

```
return 0;
           } else {
                     return 1;
           }
   int compare2(int aa, int bb, int cc) {
            if (aa == bb) {
10
                     return 0;
11
            } else if (bb == cc) {
12
                     return 1;
13
            } else {
14
                     return -1;
15
           }
17
18
  int main() {
19
           int i = 1, j = 3, k = 12;
20
           return compare1(i,j,k);
21
22
```

```
Error type 9 at Line 21: Function arguments do not match.
```

说明: 第 21 行调用 compare1 是使用了过多的实参。也可以多报一个 8 型错误。

1.20 A-20

```
struct Apple{
    int size;
    int price;
};
```

```
struct Apples{
            struct Apple a[100];
            struct Bucket{
8
                     struct base {
                               int s;
10
                               float t;
                     } bb;
12
                     int p;
13
            }buck;
14
   }lots;
   int main() {
17
            struct Product {
18
                     struct Apples app = lots;
19
            }pro;
20
            return 0;
21
22
```

```
Error type 15 at Line 19: Invalid definition in structure.
```

说明: 第19行结构体定义时对域进行赋值。

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

```
struct Leaf{
int number;
```

```
int isGreen;
  };
5
  struct Tree {
           struct Leaf leaves[100];
7
           int height;
           float weight;
           int hasFruit;
10
  } ;
11
12
  struct AppleTree {
13
           struct Tree t;
14
           int numberApple;
15
           float priceApple;
16
           int highQuality;
17
  };
18
19
  struct AppleTree newAppleTree() {
20
           int i = 0;
21
           int x = 1;
22
           struct AppleTree at = x();
23
           struct Tree tt;
24
           struct Leaf sto[100];
25
           while (i < 100) {
26
                    sto[i].number = x;
27
                    tt.leaves[i].number = sto[i].number;
28
                    sto[i].isGreen = 1;
                    tt.leaves[i].isGreen = sto[i].isGreen;
30
                    i = i + 1;
31
           }
32
           tt.height = 100;
33
           tt.weight = 2.5;
```

```
tt.hasFruit = 0;
35
           at.numberApple = 1;
           at.priceApple = 1.2;
37
           i = 0;
38
           while(i < 100) {
39
                    at.t[i].leaves[i].number = tt.leaves[i].number;
                    at.t.leaves[i].isGreen = tt.leaves[i].isGreen;
                    i = i + 1;
42
           }
43
           at.t.height = tt.height;
44
           at.t.weight = tt.weight;
           at.hasFruit = 1;
46
           at.highQuality = at.t.hasFruit * at.t.weight;
47
           return at;
48
50
  int main() {
51
           struct AppleTree aat = newAppleTree();
52
           return 0;
53
54
```

```
Error type 11 at Line 23: "AppleTree" is not a function.

Error type 10 at Line 40: Variable is not an array.

Error type 14 at Line 46: None-existent field "hasFruit".

Error type 7 at Line 47: Type mismathced for operands.
```

说明:第 23 行错误地对结构体进行函数调用;第 40 行对结构体变量进行数组访问;第 46 行访问不属于 AppleTree 结构体地域 hasFruit;第 47 行使用 int 型与 float 型进行乘法操作,也可以多报一个 5 型错误。

2.2 B-2

```
struct Actor{
           int age;
2
           int hair;
  }Di;
  struct Actress{
           int height;
           float cup;
8
  } M;
10
  struct Actor newGuy(int a, int h) {
           struct Actor ac;
12
           ac.age = a;
13
           ac.hair = h;
14
           return ac;
16
17
  struct Actress newGal(int hh, float cc) {
18
           struct Actress as;
           as.height = hh;
20
           as.cup = cc;
21
           return cc;
22
  }
23
24
  struct Actor newGuy(int aa, int hhh, int bbb) {
25
           struct Actor gay;
26
           gay.age = aa + bbb;
27
           gay.hair = hhh;
29
30
  int numberOfHair(struct Actor random) {
31
           return random.hair;
```

```
int main() {
    struct Actor cap = newGuy(1, 2);
    Di.age = cap.age;
    Di.hair = cap.hair;
    numberOfHair(Di) = 3;
    return newGal(Di.age, 15).height;
}
```

```
Error type 8 at Line 22: Type mismatched for return.

Error type 4 at Line 25: Redefined function "newGuy"

Error type 6 at Line 39: The left-hand side of an assignment must be a variable.

Error type 9 at Line 40: Function arguments do not match.
```

说明:第22行 return 语句返回 float型,与方法 newGal 定义不符;第25行函数定义 newGuy 冲突,也可以报在第11行;第39行使用 int 型作为左值;第40行函数调用第二个参数类型不符。

3 C 组测试用例

本组测试用例共2个,不包含任何错误,不需要任何输出。

3.1 C-1

```
struct Leaf{
    int number;
    int isGreen;
};
struct Tree {
```

```
struct Leaf leaves[100];
           int height;
           float weight;
           int hasFruit;
10
  };
11
12
  struct AppleTree {
13
           struct Tree t;
14
           int numberApple;
15
           float priceApple;
           int highQuality;
  };
18
19
  struct AppleTree newAppleTree() {
20
           int i = 0;
21
           int x = 1;
22
           struct AppleTree at;
23
           struct Tree tt;
24
           struct Leaf sto[100];
25
           while (i < 100) {
26
                    sto[i].number = x;
27
                    tt.leaves[i].number = sto[i].number;
28
                    sto[i].isGreen = 1;
29
                    tt.leaves[i].isGreen = sto[i].isGreen;
30
                    i = i + 1;
31
           }
32
           tt.height = 100;
           tt.weight = 2.5;
34
           tt.hasFruit = 0;
35
           at.numberApple = 1;
36
           at.priceApple = 1.2;
37
           i = 0;
```

```
while(i < 100) {
39
                    at.t.leaves[i].number = tt.leaves[i].number;
40
                    at.t.leaves[i].isGreen = tt.leaves[i].isGreen;
                    i = i + 1;
42
           }
43
           at.t.height = tt.height;
           at.t.weight = tt.weight;
45
           at.t.hasFruit = 1;
46
           at.highQuality = at.t.hasFruit * at.t.height;
47
48
           return at;
50
  int main() {
51
           struct AppleTree aat = newAppleTree();
52
           return 0;
53
54
```

1 //正常返回, 没有任何输出。

说明:测试用例 B-1 的正确版。

3.2 C-2

```
struct Actor{
    int age;
    int hair;

Di;

struct Actress{
    int height;
    float cup;
}
```

```
10
  struct Actor newGuy(int a, int h) {
11
12
           struct Actor ac;
           ac.age = a;
13
           ac.hair = h;
14
           return ac;
17
  struct Actress newGal(int hh, float cc) {
18
           struct Actress as;
19
           as.height = hh;
20
           as.cup = cc;
21
           return as;
22
23
24
  struct Actor newGay(int aa, int hhh, int bbb) {
25
           struct Actor gay;
26
           gay.age = aa + bbb;
27
           gay.hair = hhh;
29
30
  int numberOfHair(struct Actor random) {
31
           return random.hair;
32
33
34
  int main(){
35
           struct Actor cap = newGuy(1, 2);
           Di.age = cap.age;
37
           Di.hair = cap.hair;
38
           return newGal(Di.age, 1.5).height;
39
40
```

//正常返回, 没有任何输出。

说明:测试用例 B-2 的正确版。

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

```
struct Square{
           int length;
2
           int width;
  };
5
  int compare(int a, int b);
  int generate(int t) {
           return 5*3;
10
11
  int compare(int a, int b) {
           return a * b;
13
14
15
  struct Square newSquare(int 1, int w);
17
  int main(){
18
           int aaa = 3;
19
           int bbb = generate(12);
20
           int tt = compare(aaa,bbb);
21
```

```
struct Square ss = newSquare(tt, aaa);
22
           return 0;
23
24
25
  struct Square newSquare(int 1, int w);
26
  struct Square newSquare(int 1, int w) {
28
           struct Square sq;
29
           sq.length = 1;
30
           sq.width = w;
31
           return sq;
33
```

1 //正常返回, 没有任何输出。

说明: 2.1 分组的同学应当没有任何输出,其他组同学应当在第6、16、26 行报语法错误。

4.2 D-2

```
struct Bio {
           int id;
2
           int list[100];
3
           float price;
  };
6
  struct Bio newBio(int idd, float p) {
           struct Bio b;
8
           int i = 0;
9
           b.id = idd;
           b.price = p;
11
           while(i < 0) {
12
                    int j = i;
13
```

```
float i = 1.2;
14
                     b.list[j] = 0;
                     j = j + 1;
16
            }
17
           return b;
18
20
   float main() {
21
            struct Bio b = newBio(12,3.5);
22
            float idd = 1.2;
23
           int listt[100];
24
            int i = 0;
25
            while(i < 100) {
26
                     listt[i] = b.list[i];
                     i = i + 1;
28
            }
29
            return idd;
30
31
```

```
//正常返回, 没有任何输出。
```

说明: 2.2 分组的同学应当没有任何输出,其他组同学应当在第 14、22、23、25 行报出 3 型错误。

4.3 D-3

```
struct A{
    int a;
    int aArr[100][12];

struct AA{
    struct AA{
    float aType[12];
}aNode;
```

```
};
  struct B{
            int b;
10
            int Barr[12][15];
11
            struct BB{
12
                     float bType[26];
13
            }bNode;
14
   };
15
   struct C{
            int c;
18
            float bArr[10][12][23];
19
            struct CC{
20
                     int cType;
21
            }cNode;
22
  } ;
23
24
   struct D{
            int d;
26
            float dArr[100][5][23];
27
            struct DD{
28
                     int h;
29
            }dNode;
30
  } ;
31
32
   int main(){
            struct A tempA, tempA2;
34
            struct B tempB, tempB2;
35
            struct C tempC, tempC2;
36
            struct D tempD, tempD2;
37
            tempA = tempB;
```

```
tempC = tempD;
tempA2 = tempA;
tempD = tempD2;
tempD = tempD = tempD2;
tempD = tempD = tempD2;
tempD = te
```

//正常返回, 没有任何输出。

说明: 3.3 分组的同学应当没有任何输出,其他组同学应当在第38和39行报出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。

```
struct Square{
           int length;
2
           int width;
3
  };
  int compare(int a, int b);
  int generate(int t) {
           return 5*3;
10
11
  int compare(int a, int b) {
12
           return a * b;
  }
14
15
```

```
struct Square ChangeSquare(int ttt, int kkk);
17
   struct Square newSquare(int 1, float w);
18
19
  int main(){
20
           int aaa = 3;
           int bbb = generate(12);
           int tt = compare(aaa,bbb);
23
           struct Square ss = newSquare(tt, aaa);
24
25
           return 0;
27
  int newSquare(int 1, int w);
28
29
  struct Square newSquare(int 1, int w) {
           struct Square sq;
31
           sq.length = 1;
32
           sq.width = w;
33
           return sq;
34
35
```

```
Error type 19 at Line 28: Inconsistent declaration of function "

newSquare".

Error type 19 at Line 30: Inconsistent declaration of function "

newSquare".

Error type 18 at Line 16: Undefined function "changeSquare".
```

说明: 2.1 分组同学需要测试,在第 16、28、30 行报错,16 行错也可以报在最后一行,也可以多报一个 newSquare 的 18 型错误。

5.2 E2.2

这组测试用例针对 2.2 分组的同学。

```
struct Bio {
           int id;
2
           int list[100];
3
           float price;
  };
  struct Bio newBio(int idd, float p) {
           struct Bio b;
           struct Bio bb;
           int i = 0;
10
           float bb = 3.0;
11
           b.id = idd;
           b.price = p;
13
           while(i < 0) {
14
                    b.list[i] = 0;
15
                    i = i + 1;
17
           }
           return b;
18
19
  float main() {
           struct Bio b = newBio(12,3.5);
22
           float idd = 1.2;
23
           int listt[100];
24
           int i = 0;
25
           while(i < 100) {
26
                    float i = 3.0;
27
                    listt[i] = b.list[i];
28
                    i = i + 1;
30
           return idd;
31
```

```
32 }
```

```
Error type 3 at Line 11: Redefined variable "bb"

Error type 12 at Line 28: Not an integer

Error type 7 at Line 29: Type mismatched for operands
```

说明: 仅 2.2 分组的同学需要测试这个用例,需在第 11 行报 3 型错误,第 27 行不报错,但第 28 与 29 行报相应错误。

5.3 E2.3

这组测试用例针对 3.3 分组的同学。

```
struct A{
           int a;
2
           int aArr[100][12];
3
           struct AA{
                     float aType[12];
           }aNode;
  };
  struct B{
           int b;
10
           int Barr[100][12];
11
           struct BB{
12
                     int bType[12];
13
           }bNode;
14
  };
15
  struct C{
           int c;
18
           float bArr[10][12][23];
19
           struct CC{
```

```
int cType;
21
            }cNode;
22
   };
23
24
   struct D{
25
            int d;
            float dArr[10][12][23];
27
            struct DD{
28
                     int h;
29
            }dNode[15];
30
   };
31
32
  struct E{
33
            int e;
34
            float eArr[10][12][23];
35
            int eNode[15];
36
   };
37
38
   int main(){
39
            struct A tempA, tempA2;
40
            struct B tempB, tempB2;
41
            struct C tempC, tempC2;
42
            struct D tempD, tempD2;
43
            struct E tempE;
44
            tempA = tempB;
45
            tempC = tempD;
46
            tempA2 = tempA;
47
            tempD = tempD2;
48
            tempE = tempD;
49
            return 0;
50
51
```

```
Error type 5 at Line 45: Type mismatched.

Error type 5 at Line 46: Type mismatched.

Error type 5 at Line 49: Type mismatched.
```

说明: 仅 2.3 分组的同学需要测试这个用例,须在第 45、46、49 行报 5 型错误。

6 结束语

如果对本测试用例有任何疑议,可以写邮件与王珏助教联系,注意同时抄送给<mark>许老师</mark>。