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

2 B 组测试用例 **20**

2.1 B-1 20

2.2 B-2 22

3 C 组测试用例 **24**

3.1 C-1 24

3.2 C-2 26

4 D 组测试用例 **28**

4.1 D-1 28

4.2 D-2 29
4.3 D-3 30

5 E 组测试用例 **32**
5.1 E2.1 32
5.2 E2.2 33
5.3 E2.3 35

6 结束语 **37**

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

输入

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

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

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

输出

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

说明：newStdent 未定义。

1.3 A-3

输入

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

```
}
```

输出

```
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) {
                return 1;
        }
        if(a == b) {
                return 0;
        }
        return -1;
}

float compare(int x1, int x2) {
        if(x1 > x2) {
                return 1.0;
        }
        if(x2 == x1) {
                return 0.0;
        }
        return -1.0;
}

int main() {
        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;
        float weight;
        int size;
}c;

struct Candle{
        struct Cake cc;
        int number;
        float length;
}can;

int main() {
        int wei = 3;
        int s = 2;
        int num = 15;
        float len = 1.2;
        can.cc = c;
        c.name = 0;
        c.weight = wei;
        c.size = s;
        can.number = num;
        can.length = len;
        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

输入

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

```

```
struct FloatArray{
        int name;
        float fa[10];
};

int main(){
        struct IntArray inta;
        struct FloatArray floa;
        int i = 0;
        int k = 1;
        float p = 1.0;
        while (i < 10) {
                i = i + k;
                inta.ia[i] = k;
                k = k + 1;
                floa.fa[i] = p;
                p = p + 1;
        }
        return 0;
}
```

输出

```
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

输入

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

```
        }
}

float compareFloat(float aa, float bb) {
        return aa - bb;
}

int main() {
        int i = 3;
        int j = 4;
        float p = 1.2;
        float k = 2.3;
        if(i == 4) {
                return compareInt(i,j);
        } else {
                return compareFloat(p,k);
        }
}
```

输出

```
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

输入

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

```
8
9  float compareFloat(float aa, float bb) {
10         return aa - bb;
11 }
12
13 int main() {
14         int i = 3;
15         int j = 4;
16         float p = 1.2;
17         float k = 2.3;
18         if(i == 4) {
19                 return compareInt(i,j);
20         } else {
21                 return compareInt(j,k);
22         }
23 }
```

输出

```
1  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

输入

```
1  struct Paper{
2          int name;
3          int words[100];
4  }pp;
5
6  int main() {
7          struct Paper papers[10][10];
8          int i = 0;
9          int j = 0;
```

```
10          while (i < 10) {
11                  pp.words[i+j] = j;
12                  while (j < 10) {
13                          papers[i][j].words[i] = i + j;
14                          j = j + 1;
15                  }
16                  i = i + 1;
17          }
18          return pp[i+j];
19  }
```

输出

```
1  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

输入

```
1  struct Pad{
2          int size;
3          int storage;
4  };
5
6  int iPad(int ss, int st) {
7          struct Pad ip;
8          ip.size = ss;
9          ip.storage = st;
10         return ss;
11 }
12
13 int miPad(int s, int t) {
14         struct Mi{
15                 struct Pad core;
```

```
16                  int brand;
17          }myMi;
18          myMi.core = myMi(s,t);
19          myMi.brand = 0;
20          return 0;
21  }
22
23  int main() {
24          return miPad(3,4);
25  }
```

输出

```
1  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

输入

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

```
16          while (i < 100) {
17                  ftem = fa + fb;
18                  floatFib[ftem] = ftem;
19                  fa = fb;
20                  fb = ftem;
21                  i = i + 1;
22          }
23          return i;
24  }
```

输出

```
1  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

输入

```
1  struct Building{
2          struct base{
3                  int size;
4                  float height;
5          }b;
6          int bh;
7  }Yangzhou;
8
9  int main(){
10         struct Building newOne;
11         Yangzhou.b.size = 1;
12         Yangzhou.b.height = 3.4;
13         Yangzhou.bh = 100;
14         newOne.b.size = Yangzhou.b.size;
15         newOne.bh.height = Yangzhou.b.height;
16         newOne.bh = Yangzhou.bh;
```

```
17          return 0;
18  }
```

输出

```
1  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

输入

```
1  struct Javalin{
2          int head;
3          int shoulder;
4          float foot;
5          struct Pilot{
6                  int gender;
7                  int age;
8          }freelancer;
9  };
10
11 struct Anthem{
12         struct Javalin army[100];
13         int missions[100];
14 }game;
15
16 int main() {
17         int i = 0;
18         struct Javalin lin;
19         while(i < 100) {
20                 game.army[i] = lin;
21                 i = i + 1;
22                 lin.head = i;
23                 lin.shoulder = i * i;
```

```
24                  lin.freelancer.gender = i / 2;
25                  lin.age = i * 2;
26          }
27          return i;
28  }
```

输出

```
1  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

```
1  struct Food {
2          int cal;
3          int name, price;
4          float name1;
5  };
6
7  int main() {
8          struct person {
9                  struct Food burger;
10                 int age;
11                 float time, burger;
12         }p;
13         return 0;
14 }
```

输出

```
1  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;
        int weight;
        struct Price{
                int amount;
                float currency;
        }p;
};

struct Thor{
        struct Hammer hh;
        int age;
        struct Hammer{
                int am;
                float current;
        }ww;
}avenger;

int main() {
        avenger.age = 100;
        return avenger.age;
}
```

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

输入

```
1  struct Spider {
2          int legs;
3          float weight;
4  }man;
5
6  struct Venom{
7          int height;
8          int gender;
9  }edi;
10
11 struct Carnage{
12         int number;
13         float sex;
14 }end;
15
16 int main() {
17         struct Spider p;
18         struct Venom vv;
19         struct Cranage cc;
20         return 0;
21 }
```

输出

```
1  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

输入

```
1  struct Apple {
2          int weight;
3          float round;
4  };
```

```
5
6  struct Pen {
7          int length;
8          int used;
9  };
10
11 int add(struct Apple a, struct Pen p) {
12         return a.weight + p.length;
13 }
14
15 int main() {
16         struct Apple aa;
17         struct Pen pp;
18         int sum;
19         aa.weight = 3;
20         pp.length = 12;
21         sum = add(aa, pp);
22         if(sum > 5) {
23                 return aa / pp.length;
24         } else {
25                 return 0;
26         }
27 }
```

输出

```
1  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

输入

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

```
3                   return 0;
4           } else {
5                   return 1;
6           }
7   }
8
9   int compare2(int aa, int bb, int cc) {
10          if(aa == bb) {
11                  return 0;
12          } else if (bb == cc){
13                  return 1;
14          } else {
15                  return -1;
16          }
17  }
18
19  int main() {
20          int i = 1, j = 3, k = 12;
21          return compare1(i,j,k);
22  }
```

输出

```
1   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

输入

```
1   struct Apple{
2           int size;
3           int price;
4   };
5
```

```
struct Apples{
        struct Apple a[100];
        struct Bucket{
                struct base {
                        int s;
                        float t;
                }bb;
                int p;
        }buck;
}lots;

int main() {
        struct Product {
                struct Apples app = lots;
        }pro;
        return 0;
}
```

输出

```
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

输入

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

```
        int isGreen;
};

struct Tree {
        struct Leaf leaves[100];
        int height;
        float weight;
        int hasFruit;
};

struct AppleTree {
        struct Tree t;
        int numberApple;
        float priceApple;
        int highQuality;
};

struct AppleTree newAppleTree() {
        int i = 0;
        int x = 1;
        struct AppleTree at = x();
        struct Tree tt;
        struct Leaf sto[100];
        while (i < 100) {
                sto[i].number = x;
                tt.leaves[i].number = sto[i].number;
                sto[i].isGreen = 1;
                tt.leaves[i].isGreen = sto[i].isGreen;
                i = i + 1;
        }
        tt.height = 100;
        tt.weight = 2.5;
```

```
35          tt.hasFruit = 0;
36          at.numberApple = 1;
37          at.priceApple = 1.2;
38          i = 0;
39          while(i < 100) {
40                  at.t[i].leaves[i].number = tt.leaves[i].number;
41                  at.t.leaves[i].isGreen = tt.leaves[i].isGreen;
42                  i = i + 1;
43          }
44          at.t.height = tt.height;
45          at.t.weight = tt.weight;
46          at.hasFruit = 1;
47          at.highQuality = at.t.hasFruit * at.t.weight;
48          return at;
49  }
50
51  int main() {
52          struct AppleTree aat = newAppleTree();
53          return 0;
54  }
```

输出

```
1  Error type 11 at Line 23: "AppleTree" is not a function.
2  Error type 10 at Line 40: Variable is not an array.
3  Error type 14 at Line 46: None-existent field "hasFruit".
4  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

输入

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

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

struct Actor newGuy(int a, int h) {
        struct Actor ac;
        ac.age = a;
        ac.hair = h;
        return ac;
}

struct Actress newGal(int hh, float cc) {
        struct Actress as;
        as.height = hh;
        as.cup = cc;
        return cc;
}

struct Actor newGuy(int aa, int hhh, int bbb) {
        struct Actor gay;
        gay.age = aa + bbb;
        gay.hair = hhh;
}

int numberOfHair(struct Actor random) {
        return random.hair;
```

```
33 }
34
35 int main(){
36         struct Actor cap = newGuy(1, 2);
37         Di.age = cap.age;
38         Di.hair = cap.hair;
39         numberOfHair(Di) = 3;
40         return newGal(Di.age, 15).height;
41 }
```

输出

```
1 Error type 8 at Line 22: Type mismatched for return.
2 Error type 4 at Line 25: Redefined function "newGuy"
3 Error type 6 at Line 39: The left-hand side of an assignment must be
     a variable.
4 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

输入

```
1 struct Leaf{
2         int number;
3         int isGreen;
4 };
5
6 struct Tree {
```

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

```
        while(i < 100) {
                at.t.leaves[i].number = tt.leaves[i].number;
                at.t.leaves[i].isGreen = tt.leaves[i].isGreen;
                i = i + 1;
        }
        at.t.height = tt.height;
        at.t.weight = tt.weight;
        at.t.hasFruit = 1;
        at.highQuality = at.t.hasFruit * at.t.height;
        return at;
}

int main() {
        struct AppleTree aat = newAppleTree();
        return 0;
}
```

输出

```
//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

输入

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

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

```
10
11 struct Actor newGuy(int a, int h) {
12         struct Actor ac;
13         ac.age = a;
14         ac.hair = h;
15         return ac;
16 }
17
18 struct Actress newGal(int hh, float cc) {
19         struct Actress as;
20         as.height = hh;
21         as.cup = cc;
22         return as;
23 }
24
25 struct Actor newGay(int aa, int hhh, int bbb) {
26         struct Actor gay;
27         gay.age = aa + bbb;
28         gay.hair = hhh;
29 }
30
31 int numberOfHair(struct Actor random) {
32         return random.hair;
33 }
34
35 int main(){
36         struct Actor cap = newGuy(1, 2);
37         Di.age = cap.age;
38         Di.hair = cap.hair;
39         return newGal(Di.age, 1.5).height;
40 }
```

输出

```
//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

输入

```
struct Square{
        int length;
        int width;
};

int compare(int a, int b);

int generate(int t) {
        return 5*3;
}

int compare(int a, int b){
        return a * b;
}

struct Square newSquare(int l, int w);

int main(){
        int aaa = 3;
        int bbb = generate(12);
        int tt = compare(aaa,bbb);
```

```
22          struct Square ss = newSquare(tt, aaa);
23          return 0;
24  }
25
26  struct Square newSquare(int l, int w);
27
28  struct Square newSquare(int l, int w){
29          struct Square sq;
30          sq.length = l;
31          sq.width = w;
32          return sq;
33  }
```

输出

```
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

输入

```
1  struct Bio {
2          int id;
3          int list[100];
4          float price;
5  };
6
7  struct Bio newBio(int idd, float p){
8          struct Bio b;
9          int i = 0;
10         b.id = idd;
11         b.price = p;
12         while(i < 0) {
13                 int j = i;
```

```
14                  float i = 1.2;
15                  b.list[j] = 0;
16                  j = j + 1;
17          }
18          return b;
19  }
20
21  float main() {
22          struct Bio b = newBio(12,3.5);
23          float idd = 1.2;
24          int listt[100];
25          int i = 0;
26          while(i < 100) {
27                  listt[i] = b.list[i];
28                  i = i + 1;
29          }
30          return idd;
31  }
```

输出

```
1  //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

输入

```
1  struct A{
2          int a;
3          int aArr[100][12];
4          struct AA{
5                  float aType[12];
6          }aNode;
```

```
};

struct B{
        int b;
        int Barr[12][15];
        struct BB{
                float bType[26];
        }bNode;
};

struct C{
        int c;
        float bArr[10][12][23];
        struct CC{
                int cType;
        }cNode;
};

struct D{
        int d;
        float dArr[100][5][23];
        struct DD{
                int h;
        }dNode;
};

int main(){
        struct A tempA, tempA2;
        struct B tempB, tempB2;
        struct C tempC, tempC2;
        struct D tempD, tempD2;
        tempA = tempB;
```

```
39          tempC = tempD;
40          tempA2 = tempA;
41          tempD = tempD2;
42          return 0;
43  }
```

输出

```
1  //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。

输入

```
1  struct Square{
2          int length;
3          int width;
4  };
5
6  int compare(int a, int b);
7
8  int generate(int t) {
9          return 5*3;
10 }
11
12 int compare(int a, int b){
13         return a * b;
14 }
15
```

```
16 struct Square ChangeSquare(int ttt, int kkk);
17
18 struct Square newSquare(int l, float w);
19
20 int main(){
21         int aaa = 3;
22         int bbb = generate(12);
23         int tt = compare(aaa,bbb);
24         struct Square ss = newSquare(tt, aaa);
25         return 0;
26 }
27
28 int newSquare(int l, int w);
29
30 struct Square newSquare(int l, int w){
31         struct Square sq;
32         sq.length = l;
33         sq.width = w;
34         return sq;
35 }
```

输出

```
1 Error type 19 at Line 28: Inconsistent declaration of function "
    newSquare".
2 Error type 19 at Line 30: Inconsistent declaration of function "
    newSquare".
3 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。

输入

```
struct Bio {
        int id;
        int list[100];
        float price;
};

struct Bio newBio(int idd, float p){
        struct Bio b;
        struct Bio bb;
        int i = 0;
        float bb = 3.0;
        b.id = idd;
        b.price = p;
        while(i < 0) {
                b.list[i] = 0;
                i = i + 1;
        }
        return b;
}

float main() {
        struct Bio b = newBio(12,3.5);
        float idd = 1.2;
        int listt[100];
        int i = 0;
        while(i < 100) {
                float i = 3.0;
                listt[i] = b.list[i];
                i = i + 1;
        }
        return idd;
```

```
}
```

输出

```
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。

输入

```
struct A{
        int a;
        int aArr[100][12];
        struct AA{
                float aType[12];
        }aNode;
};

struct B{
        int b;
        int Barr[100][12];
        struct BB{
                int bType[12];
        }bNode;
};

struct C{
        int c;
        float bArr[10][12][23];
        struct CC{
```

```
                int cType;
        }cNode;
};

struct D{
        int d;
        float dArr[10][12][23];
        struct DD{
                int h;
        }dNode[15];
};

struct E{
        int e;
        float eArr[10][12][23];
        int eNode[15];
};

int main(){
        struct A tempA, tempA2;
        struct B tempB, tempB2;
        struct C tempC, tempC2;
        struct D tempD, tempD2;
        struct E tempE;
        tempA = tempB;
        tempC = tempD;
        tempA2 = tempA;
        tempD = tempD2;
        tempE = tempD;
        return 0;
}
```

输出

```
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，注意同时抄送给许老师。