编译原理第一次实验测试用例：目录

1 A 组测试用例

本组测试用例共 9 个，每个仅包含单个的词法或者语法错误。除特殊说明外，不可多报。多报、漏报错误，或者打印语法树都会导致扣分。错误编号和行号之后的说明文字不要求与给出的输出完全一致，仅供助教理解使用，不作为评分依据。

1.1 A-1

输入

```
float _func1(int X1)
{
    float a_1_2;
    int 6_Wrong;
    return 0;
}
```

输出

```
Error type B at line 4: syntax error, near ';'
```

说明：错误类型也可以是 A 类，或者一个 A 一个 B，但是只能在第 4 行。这里有一个非法标识符 6_Wrong，注意标识符可以以下划线开始，所以第 1 行正确。

1.2 A-2

输入

```
int sub(int a, int b)
{
    int c = a + b - a * b / 3;
    int d = !a && b || c >= 4;
    int e = a & b;
    return e;
}
```

输出

```
Error type A at line 5: mysteriously character '&'
```

说明：必须有 type A 错误；可以多报一个 type B 错误。这里有一个非法的符号 &。

1.3 A-3

输入

```
int add()
{
    int a, b, c;
    c = (a + b) - (c * d / (-a));
    b = a || c;
    return b;
}
```

输出

```
Error type B at line 4: syntax error
```

说明：第 4 行缺少匹配的左括号。

1.4 A-4

输入

```
int;
int if()
{
    int a;
    if (a > 0)
        return 1;
    else
        return 0;
}
```

输出

```
Error type B at line 2: syntax error
```

说明：第 2 行保留字 if 不能作为标识符使用。注意第 1 行语句没有意义但是合法。

1.5 A-5

输入

```
int array(int i)
{
    int a[3][3*3];
    a[i][i] = 1;
    while (i < 9)
    {
        a[0][i]= 1;
        i = i + 1;
    }
}
```

输出

```
Error type B at line 3: syntax error, near '*'
```

说明：第 3 行数组定义格式错误，出现表达式。

1.6 A-6

输入

```
int a = 1;
int add(int b)
{
    return a + b;
}
int main()
{
    int sum;
    sum = add(2);
}
```

输出

```
Error type B at line 1: syntax error, the global variable cannot be
    initialized in the definition.
```

说明：第 1 行定义全局变量时进行了初始化。

1.7 A-7

输入

```
struct Date
{
    int year, month;
    int day;
};
int main()
{
    Date d[10];
    d[0].year = 2020;
}
```

输出

```
Error type B at line 8: syntax error, missing 'struct'
```

说明：第 8 行定义结构体时缺少关键字 struct。

1.8 A-8

输入

```
int main()
{
    struct A
    {
        int a;
        float b;
    };
```

```
8       struct A a[100];
9       int i = 0;
10      while (i < 100)
11      {
12          a[i].a = a[i].b = 0;
13          i = i + 1;
14      }
15  }
```

输出

```
1  Error type B at line 7: syntax error, near ';'
```

说明：第 7 行，在方法体内定义结构体时缺少声明的变量名，报在第 3 行也可以。

1.9 A-9

输入

```
1  int max(int x, int y)
2  {
3      int t;
4      if (x > y)
5          t = x;
6      else
7          t = y
8      return t;
9  }
10 int main()
11 {
12     int a = 6, b = 4, c = 5;
13     int maxs = max(a, max(b, c));
14 }
```

输出

```
1  Error type B at line 7: syntax error, missing ';'
```

说明：第 7 行漏写了句尾的分号。也可在第 8 行报错。

2 B 组测试用例

本组测试用例共 2 个，每个用例包含多处不同的错误。除特殊说明外，漏报、多报错误或者打印语法树都会导致扣分。

2.1 B-1

输入

```
1  struct Student
2  {
3      int ID,
4      float score;
5  }class[2];
6
7  struct Student test(struct Student a)
8  {
9      return a;
10 }
11
12 int main()
13 {
14     int i = 0, n = 2;
15     float sum = 0;
16     struct Student classes[n][10];
17     class[0].ID = 1;
18     class[0].score = 90;
19     class[1].ID = 2;
20     class[1].score = .85 * 100;
21     test(class[0]);
22     while (i < 2)
23     {
24         sum += class[i].score;
```

```
25          i = i + 1;
26      }
27      return 0;
28  }
```

输出

```
1  Error type B at line 3: syntax error, near ','
2  Error type B at line 16: syntax error, near '['
3  Error type A at line 20: illegal float number '.85'
4  Error type B at line 24: syntax error, near '+'
```

说明：第 3 行末尾分号错写成逗号，也可以报错在第 4 行；第 16 行用变量定义数组；第 20 行小数点前必须有数字，也可识别成 B 类型错误，或多报一个 B 类错误；第 24 行使用了未定义的操作符 +=。

2.2 B-2

输入

```
1  int shsort(int s[4], int n)
2  {
3      int i, j, d;
4      d = n / 2;
5      while(d > = 1)
6      {
7          i = d + 1;
8          while (i <= n)
9          {
10             s[0] = s[i];
11             j  = i - d;
12             while((j > 0) && (s[0] < s[j]))
13             {
14                 s[j + d] = s[j];
15                 j = j - d;
16             }
```

```
17              s[j + d] = s[0];
18              i ++;
19          }
20          d = d / 2;
21      }
22      return 0;
23  }
24  int main()
25  {
26      int a[4];
27      a[0] = 2;
28      a[1] = 4;
29      a[2] = 3;
30      a[3]] = 1;
31      int n = 4;
32      shsort(a, n);
33      return 0;
34  }
```

输出

```
1  Error type B at line 5: syntax error, near '>'
2  Error type B at line 18: syntax error, near '+'
3  Error type B at line 30: syntax error, near ']'
4  Error type B at line 31: syntax error, the local variable cannot be
       defined after statements
```

说明：第 5 行 >= 号中间多了一个空格；第 18 行使用了未定义符号 ++；第 30 行多打了一个右括号]；第 31 行语句之后不能再出现变量定义。

3 C 组测试用例

本组测试用例共 2 个，不包含任何错误，需要输出正确的语法树。除特殊说明外，应与给出的语法树完全相同。语法树打印错误酌情扣分。

3.1 C-1

输入

```
struct Student
{
    int SID;
    int age;
    struct Teacher
    {
        int TID;
    } t;
};

int main()
{
    struct Student s1;
    s1.SID = 1;
    s1.age = 18;
    s1.t.TID = 3;
    return s1;
}
```

输出

```
Program (1)
  ExtDefList (1)
    ExtDef (1)
      Specifier (1)
        StructSpecifier (1)
          STRUCT
          OptTag (1)
            ID: Student
          LC
          DefList (3)
```

```
        Def (3)
          Specifier (3)
            TYPE: int
          DecList (3)
            Dec (3)
              VarDec (3)
                ID: SID
          SEMI
        DefList (4)
          Def (4)
            Specifier (4)
              TYPE: int
            DecList (4)
              Dec (4)
                VarDec (4)
                  ID: age
            SEMI
          DefList (5)
            Def (5)
              Specifier (5)
                StructSpecifier (5)
                  STRUCT
                  OptTag (5)
                    ID: Teacher
                  LC
                  DefList (7)
                    Def (7)
                      Specifier (7)
                        TYPE: int
                      DecList (7)
                        Dec (7)
                          VarDec (7)
```

```
                                ID: TID
                            SEMI
                        RC
                    DecList (8)
                      Dec (8)
                        VarDec (8)
                          ID: t
                    SEMI
          RC
      SEMI
    ExtDefList (11)
      ExtDef (11)
        Specifier (11)
          TYPE: int
        FunDec (11)
          ID: main
          LP
          RP
        CompSt (12)
          LC
          DefList (13)
            Def (13)
              Specifier (13)
                StructSpecifier (13)
                  STRUCT
                  Tag (13)
                    ID: Student
              DecList (13)
                Dec (13)
                  VarDec (13)
                    ID: s1
              SEMI
```

```
        StmtList (14)
          Stmt (14)
            Exp (14)
              Exp (14)
                Exp (14)
                  ID: s1
                DOT
                ID: SID
              ASSIGNOP
              Exp (14)
                INT: 1
            SEMI
          StmtList (15)
            Stmt (15)
              Exp (15)
                Exp (15)
                  Exp (15)
                    ID: s1
                  DOT
                  ID: age
                ASSIGNOP
                Exp (15)
                  INT: 18
              SEMI
            StmtList (16)
              Stmt (16)
                Exp (16)
                  Exp (16)
                    Exp (16)
                      Exp (16)
                        ID: s1
                      DOT
```

```
                        ID: t
                      DOT
                        ID: TID
                    ASSIGNOP
                    Exp (16)
                      INT: 3
                  SEMI
                StmtList (17)
                  Stmt (17)
                    RETURN
                    Exp (17)
                      ID: s1
                    SEMI
          RC
```

说明：使用的空格可以用 Tab 替换，注意缩进

3.2 C-2

输入

```
int MatrixMax(int a[3][4])
{
    int i=0,j=0;
    int max,max_i=0,max_j=0;
    max = a[0][0];
    while(i < 3)
    {
        while(j < 4)
        {
            if(a[i][j] > max)
            {
                max=a[i][j];
                max_i=i;
                max_j=j;
```

```
            }
            j = j + 1;
        }
        i = i + 1;
    }
    return 0;
}
```

输出

```
Program (1)
  ExtDefList (1)
    ExtDef (1)
      Specifier (1)
        TYPE: int
      FunDec (1)
        ID: MatrixMax
        LP
        VarList (1)
          ParamDec (1)
            Specifier (1)
              TYPE: int
            VarDec (1)
              VarDec (1)
                VarDec (1)
                  ID: a
                LB
                INT: 3
                RB
              LB
              INT: 4
              RB
        RP
      CompSt (2)
```

```
        LC
        DefList (3)
          Def (3)
            Specifier (3)
              TYPE: int
            DecList (3)
              Dec (3)
                VarDec (3)
                  ID: i
                ASSIGNOP
                Exp (3)
                  INT: 0
              COMMA
              DecList (3)
                Dec (3)
                  VarDec (3)
                    ID: j
                  ASSIGNOP
                  Exp (3)
                    INT: 0
            SEMI
          DefList (4)
            Def (4)
              Specifier (4)
                TYPE: int
              DecList (4)
                Dec (4)
                  VarDec (4)
                    ID: max
                COMMA
                DecList (4)
                  Dec (4)
```

```
                VarDec (4)
                  ID: max_i
                ASSIGNOP
                Exp (4)
                  INT: 0
              COMMA
              DecList (4)
                Dec (4)
                  VarDec (4)
                    ID: max_j
                  ASSIGNOP
                  Exp (4)
                    INT: 0
        SEMI
    StmtList (5)
      Stmt (5)
        Exp (5)
          Exp (5)
            ID: max
          ASSIGNOP
          Exp (5)
            Exp (5)
              Exp (5)
                ID: a
              LB
              Exp (5)
                INT: 0
              RB
            LB
            Exp (5)
              INT: 0
            RB
```

```
    SEMI
  StmtList (6)
    Stmt (6)
      WHILE
      LP
      Exp (6)
        Exp (6)
          ID: i
        RELOP
        Exp (6)
          INT: 3
      RP
      Stmt (7)
        CompSt (7)
          LC
          StmtList (8)
            Stmt (8)
              WHILE
              LP
              Exp (8)
                Exp (8)
                  ID: j
                RELOP
                Exp (8)
                  INT: 4
              RP
              Stmt (9)
                CompSt (9)
                  LC
                  StmtList (10)
                    Stmt (10)
                      IF
```

```
LP
Exp (10)
  Exp (10)
    Exp (10)
      Exp (10)
        ID: a
      LB
      Exp (10)
        ID: i
      RB
    LB
    Exp (10)
      ID: j
    RB
  RELOP
  Exp (10)
    ID: max
RP
Stmt (11)
  CompSt (11)
    LC
    StmtList (12)
      Stmt (12)
        Exp (12)
          Exp (12)
            ID: max
          ASSIGNOP
          Exp (12)
            Exp (12)
              Exp (12)
                ID: a
              LB
```

```
                    Exp (12)
                      ID: i
                    RB
                  LB
                  Exp (12)
                    ID: j
                  RB
                SEMI
              StmtList (13)
                Stmt (13)
                  Exp (13)
                    Exp (13)
                      ID: max_i
                    ASSIGNOP
                    Exp (13)
                      ID: i
                  SEMI
                StmtList (14)
                  Stmt (14)
                    Exp (14)
                      Exp (14)
                        ID: max_j
                      ASSIGNOP
                      Exp (14)
                        ID: j
                    SEMI
            RC
          StmtList (16)
            Stmt (16)
              Exp (16)
                Exp (16)
                  ID: j
```

```
                                ASSIGNOP
                                Exp (16)
                                  Exp (16)
                                    ID: j
                                  PLUS
                                  Exp (16)
                                    INT: 1
                              SEMI
                        RC
                  StmtList (18)
                    Stmt (18)
                      Exp (18)
                        Exp (18)
                          ID: i
                        ASSIGNOP
                        Exp (18)
                          Exp (18)
                            ID: i
                          PLUS
                          Exp (18)
                            INT: 1
                      SEMI
                RC
            StmtList (20)
              Stmt (20)
                RETURN
                Exp (20)
                  INT: 0
                SEMI
      RC
```

说明：考察对数组的翻译。

4 D 组测试用例

本组测试用例共 3 个，针对不同分组进行测试。对应分组的同学需要输出语法树，提示错误则不得分；其他分组的同学只需要在对应位置提示错误即可，如果打印了语法树，则将视为违规，将会倒扣分。

4.1 D-1

输入

```
int func_test()
{
    int _dec_ = 947;
    int _oct_ = 0705;
    int _dhex_ = 0xFFaBc - _oct_;
    int _result_ = - _dhex_ + _oct_ * ( _dec_ - 0X23fD );
}
```

输出

```
Program (1)
  ExtDefList (1)
    ExtDef (1)
      Specifier (1)
        TYPE: int
      FunDec (1)
        ID: func_test
        LP
        RP
      CompSt (2)
        LC
        DefList (3)
          Def (3)
            Specifier (3)
              TYPE: int
            DecList (3)
```

```
        Dec (3)
          VarDec (3)
            ID: _dec_
          ASSIGNOP
          Exp (3)
            INT: 947
      SEMI
    DefList (4)
      Def (4)
        Specifier (4)
          TYPE: int
        DecList (4)
          Dec (4)
            VarDec (4)
              ID: _oct_
            ASSIGNOP
            Exp (4)
              INT: 453
        SEMI
      DefList (5)
        Def (5)
          Specifier (5)
            TYPE: int
          DecList (5)
            Dec (5)
              VarDec (5)
                ID: _dhex_
              ASSIGNOP
              Exp (5)
                Exp (5)
                  INT: 1047228
                MINUS
```

```
            Exp (5)
              ID: _oct_
        SEMI
      DefList (6)
        Def (6)
          Specifier (6)
            TYPE: int
          DecList (6)
            Dec (6)
              VarDec (6)
                ID: _result_
              ASSIGNOP
              Exp (6)
                Exp (6)
                  MINUS
                  Exp (6)
                    ID: _dhex_
                PLUS
                Exp (6)
                  Exp (6)
                    ID: _oct_
                  STAR
                  Exp (6)
                    LP
                    Exp (6)
                      Exp (6)
                        ID: _dec_
                      MINUS
                      Exp (6)
                        INT: 9213
                    RP
          SEMI
```

```
81          RC
```

说明：1.1 分组的同学需要输出该语法树，8 进制和 16 进制数必须正确转换（453、1047228 和 9213）；其他分组的同学只要提示相应的错误，而且不输出语法树即可。

4.2 D-2

输入

```
1  int float_test()
2  {
3      float X_1 = 3.80e-7;
4      float X_2 = 345.2e+4;
5      float X_3 = 2.76E+3;
6      float X_4 = .994E-2;
7      float X_5 = 11.e1;
8      float X_6 = -0.4E-03;
9      float result = (15.E-1 + X_3) + X_6;
10 }
```

输出

```
1  Program (1)
2    ExtDefList (1)
3      ExtDef (1)
4        Specifier (1)
5          TYPE: int
6        FunDec (1)
7          ID: float_test
8          LP
9          RP
10       CompSt (2)
11         LC
12         DefList (3)
13           Def (3)
14             Specifier (3)
```

```
      TYPE: float
    DecList (3)
      Dec (3)
        VarDec (3)
          ID: X_1
        ASSIGNOP
        Exp (3)
          FLOAT: 0.000000
    SEMI
  DefList (4)
    Def (4)
      Specifier (4)
        TYPE: float
      DecList (4)
        Dec (4)
          VarDec (4)
            ID: X_2
          ASSIGNOP
          Exp (4)
            FLOAT: 3452000.000000
      SEMI
    DefList (5)
      Def (5)
        Specifier (5)
          TYPE: float
        DecList (5)
          Dec (5)
            VarDec (5)
              ID: X_3
            ASSIGNOP
            Exp (5)
              FLOAT: 2760.000000
```

```
    SEMI
  DefList (6)
    Def (6)
      Specifier (6)
        TYPE: float
      DecList (6)
        Dec (6)
          VarDec (6)
            ID: X_4
          ASSIGNOP
          Exp (6)
            FLOAT: 0.009940
      SEMI
    DefList (7)
      Def (7)
        Specifier (7)
          TYPE: float
        DecList (7)
          Dec (7)
            VarDec (7)
              ID: X_5
            ASSIGNOP
            Exp (7)
              FLOAT: 110.000000
        SEMI
      DefList (8)
        Def (8)
          Specifier (8)
            TYPE: float
          DecList (8)
            Dec (8)
              VarDec (8)
```

```
                    ID: X_6
                  ASSIGNOP
                  Exp (8)
                    MINUS
                    Exp (8)
                      FLOAT: 0.000400
                SEMI
              DefList (9)
                Def (9)
                  Specifier (9)
                    TYPE: float
                  DecList (9)
                    Dec (9)
                      VarDec (9)
                        ID: result
                      ASSIGNOP
                      Exp (9)
                        Exp (9)
                          LP
                          Exp (9)
                            Exp (9)
                              FLOAT: 1.500000
                            PLUS
                            Exp (9)
                              ID: X_3
                          RP
                        PLUS
                        Exp (9)
                          ID: X_6
                    SEMI
    RC
```

说明：1.2 分组的同学需要输出语法树，注意科学计数法浮点数的正确转换。其它分组同学

只需要提示相应错误，而且不输出语法树即可。

4.3 D-3

输入

```
/*
**This is a test for comments
/\\\\\\\\\\\\\\\\\\\/
/
*/
int /*/**/F(int n) // {%^@#~~~ //
{
    /****8
    int i = 0;
    ****\*/
    if(n == 0)
    //
        return 1; /*}  \\\
  \\
  */
    return n * F(n - 1);
}
int main() //b > a comments*//\\//*
{
    int n, r;
    n = 10;
    r = F(n);//\\*//\\*//\\////\\*//\\
    return /*ends\/\/\/\/\/\/*/0;
}
```

输出

```
Program (6)
  ExtDefList (6)
    ExtDef (6)
```

```
    Specifier (6)
      TYPE: int
    FunDec (6)
      ID: F
      LP
      VarList (6)
        ParamDec (6)
          Specifier (6)
            TYPE: int
          VarDec (6)
            ID: n
      RP
    CompSt (7)
      LC
      StmtList (11)
        Stmt (11)
          IF
          LP
          Exp (11)
            Exp (11)
              ID: n
            RELOP
            Exp (11)
              INT: 0
          RP
          Stmt (13)
            RETURN
            Exp (13)
              INT: 1
            SEMI
        StmtList (16)
          Stmt (16)
```

```
            RETURN
            Exp (16)
              Exp (16)
                ID: n
              STAR
              Exp (16)
                ID: F
                LP
                Args (16)
                  Exp (16)
                    Exp (16)
                      ID: n
                    MINUS
                    Exp (16)
                      INT: 1
                RP
            SEMI
      RC
  ExtDefList (18)
    ExtDef (18)
      Specifier (18)
        TYPE: int
      FunDec (18)
        ID: main
        LP
        RP
      CompSt (19)
        LC
        DefList (20)
          Def (20)
            Specifier (20)
              TYPE: int
```

```
        DecList (20)
          Dec (20)
            VarDec (20)
              ID: n
          COMMA
          DecList (20)
            Dec (20)
              VarDec (20)
                ID: r
        SEMI
      StmtList (21)
        Stmt (21)
          Exp (21)
            Exp (21)
              ID: n
            ASSIGNOP
            Exp (21)
              INT: 10
          SEMI
        StmtList (22)
          Stmt (22)
            Exp (22)
              Exp (22)
                ID: r
              ASSIGNOP
              Exp (22)
                ID: F
                LP
                Args (22)
                  Exp (22)
                    ID: n
                RP
```

```
                SEMI
              StmtList (23)
                Stmt (23)
                  RETURN
                  Exp (23)
                    INT: 0
                  SEMI
          RC
```

说明：1.3 分组的同学需要输出语法树，不能提示有语法错误；其他分组同学只需要提示相应错误，且不输出语法树即可。

5 E 组测试用例

本组测试用例共 6 个，针对不同分组进行测试。

5.1 E1.1

这组测试用例针对 1.1 分组的同学。

输入（E1-1）

```c
int test_for_wrong_oct_number()
{
    int _correct_oct_number_ = 003456;
    int _decimal_number = 748;
    int _wrong_oct_number_ = 0748;
    return 0;
}
```

输出

```
Error type A at Line 5: Illegal octal number "0748"
```

说明：仅 1.1 分组的同学需要测试这个用例，针对错误的 8 进制数 0748，识别成错误类型 B 也可以。

输入（E1-2）

```
int test_for_wrong_dhex_number()
{
    int _correct_dhex_number_ = 0xf29C0;
    int _wrong_dhex_number_ = 0xCd0G;
    int _correct_dhex_number2_ = 0x000abCD;
    int _wrong_dhex_number2_ = 0xx346f;
}
```

输出

```
Error type A at Line 4: Illegal hexadecimal number "0xCd0G"
Error type A at Line 6: Illegal hexadecimal number "0xx346f"
```

说明：仅 1.1 分组的同学需要测试这个用例，针对错误的 16 进制数 0xCd0G 与 0xx346f，识别成错误类型 B 也可以。

5.2 E1.2

这组测试用例针对 1.2 分组的同学。

输入（E2-1）

```
float function()
{
    float x1 = 002.45E10;
    float x2 = 45.e-1.1;
    float x3 = .45e-3;
    float x4 = 34.34E+1.2;
    return x1 + x3;
}
```

输出

```
Error type A at Line 4: Illegal float number "45.e-1.1"
Error type A at Line 6: Illegal float number "34.34E+1.2"
```

说明：仅 1.2 分组的同学需要测试这个用例，针对错误浮点数 45.e-1.1 和 34.34E+1.2，识别成错误类型 B 也可以。

输入（E2-2）

```
float function()
{
    float a = 12.34e+;
    float b = .e;
    float c = .e-2;
}
```

输出

```
Error type A at Line 3: Illegal float number "12.34e+"
Error type B at line 4: syntax error, unexpected DOT
Error type A at Line 5: Illegal float number ".e-2"
```

说明：仅 1.2 分组的同学需要测试这个用例，针对错误浮点数 12.34e+、.e 和.e-2，识别成错误类型 B 和 A 都可以。

5.3 E1.3

这组测试用例针对 1.3 分组的同学。

输入（E3-1）

```
/****
* @param int a
*/
int main()
{
    /*   TODO
                definition 12345
        */
    int day,x1,x2;
    day = 9 /* this is okay*/;
    x2 = 1;
    // \\\\\////////\\\\\\~~////////\\\\\\*/
    while(day> /*>=<*/ 0)
    {
```

```
        x1=(x2+1)*2;
        /*
            //////\\\\\\\\\/////>
            /* \\\\\\\\''''''' this is not okay!! */
        */
        x2=x1;
        day = day - 1;
    }// assert(day == ?);
    return /*-1;*/0;
}
```

输出

```
Error type B at Line 19: syntax error, near '*'
```

说明：仅 1.3 分组的同学需要测试这个用例，针对嵌套的多行注释。19 行出现了多余的注释符号。识别成类型 A 也可以，需要合理的错误提示。

输入（E3-2）

```
/***comment
still
*/
/* ////\\asfdlkajhsldf\\\\<>@#$
*/
int main() //a function
{
    float h, s; int i;
    h = s = 100 //;
    h = h / 2; // first
    i = 2;
    /* a new comment*/
    while (i <= 10) //\\\//\\/*
    {
        s = s + 2 * h;
        h = h / 2 /*\/\/\/\/\*\/\/\\****\/*/;
```

```
17          i = i + 1;
18      }
19      return 0;
20  }
21  /**end ~??
```

输出

```
1  Error type B at Line 9: missing ';'
2  Error type B at Line 21: syntax error, no match comment "/*"
```

说明：仅 1.3 分组的同学需要测试这个用例。第 9 行语句漏写了分号，也可以报在第 10 行；第 21 行第缺少结束的注释，如果打印了语法树，或者程序异常终止、死循环无法退出等，则该用例不得分。不限定错误类型以及提示方式，但是出错位置必须限定在 21 行或者以后的位置；直接提示“未终止的注释”也可以。

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。