



# 错误处理作业简介

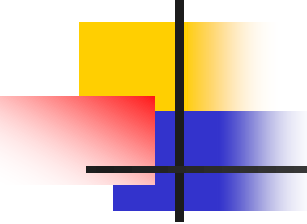
---



## 目标和要求：

---

- 针对常见的错误分类编写错误处理程序，考核学生对错误处理方法的掌握情况，培养学生编写错误处理程序的能力。
- 学生需分析编译过程的每个阶段可能出现的错误，学完“第8章 错误处理”后，对各阶段的错误进行错误局部化处理，并进行补充和完善，按要求输出错误信息。
- 作业提交至教学平台，用5个测试程序进行测试，并进行相似性检查。根据输出结果与预期结果一致部分所占的比例给分。

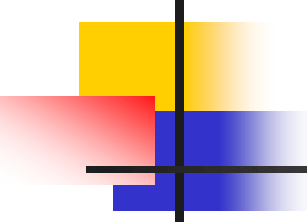


请根据给定的文法设计并实现错误处理程序，能诊察出常见的语法和语义错误，进行错误局部化处理，并输出错误信息。为了方便进行自动评测，输入输出及处理要求如下：

- 错误类别码按下表中的定义输出，行号从**1**开始计数：
- 约定：
  - 输入的被编译源文件统一命名为**testfile.txt**；
  - 错误信息输出的结果文件统一命名为**error.txt**；
  - 结果文件中每行按如下方式组织：

错误所在的行号 错误的类别码

（行号与类别码之间只有一个空格，类别码严格按照表格中的小写英文字母）



错误的类别码请统一按如下形式定义：

| 错误类型   | 错误类别码 |
|--|-------|
| 非法符号或不符合词法   | a     |
| 名字重定义  | b     |
| 未定义的名字   | c     |
| 函数参数个数不匹配  | d     |
| 函数参数类型不匹配  | e     |
| 条件判断中出现不合法的类型                                      | f     |
| 无返回值的函数存在不匹配的 <b>return</b> 语句                     | g     |
| 有返回值的函数缺少 <b>return</b> 语句或存在不匹配的 <b>return</b> 语句 | h     |
| 数组元素的下标只能是整型表达式                                    | i     |
| 不能改变常量的值   | j     |
| 应为分号   | k     |
| 应为右小括号')'  | l     |
| 应为右中括号']'  | m     |
| <b>do-while</b> 语句中缺少 <b>while</b>                 | n     |
| 常量定义中=后面只能是整型或字符型常量                                | o     |



例如对如下程序段：

```
1 const int const1 = 1, const2 = -100;
2 const char const3 = '?';
3 int change1;
4 char change3;
5 int gets1(int var1,int var2){
6     change1 = var1 + var2
7     return (change1);
8 }
9 void main(){
10     change1 = 10;
11     printf("Hello World");
12     printf(gets1(10, 20));
13 }
```

error.txt 中内容为：

```
2 a
6 k
```



## 注意事项:

---

- (1) 本次作业的每个测试程序各包含1-3个错误，均来自前述表格；按报出错误占应报错误的比例得分；
- (2) 表中未包含的错误，需要自行设计，本作业考核不涉及；
- (3) 完成本次作业时，不需要输出词法分析和语法分析作业要求输出的内容；
- (4) 本次考核之外，发现错误时最好直接输出描述信息，而不是仅给出错误类别码；
- (5) 输出到error.txt中的错误信息，每行末尾都需要加一个换行符\n。