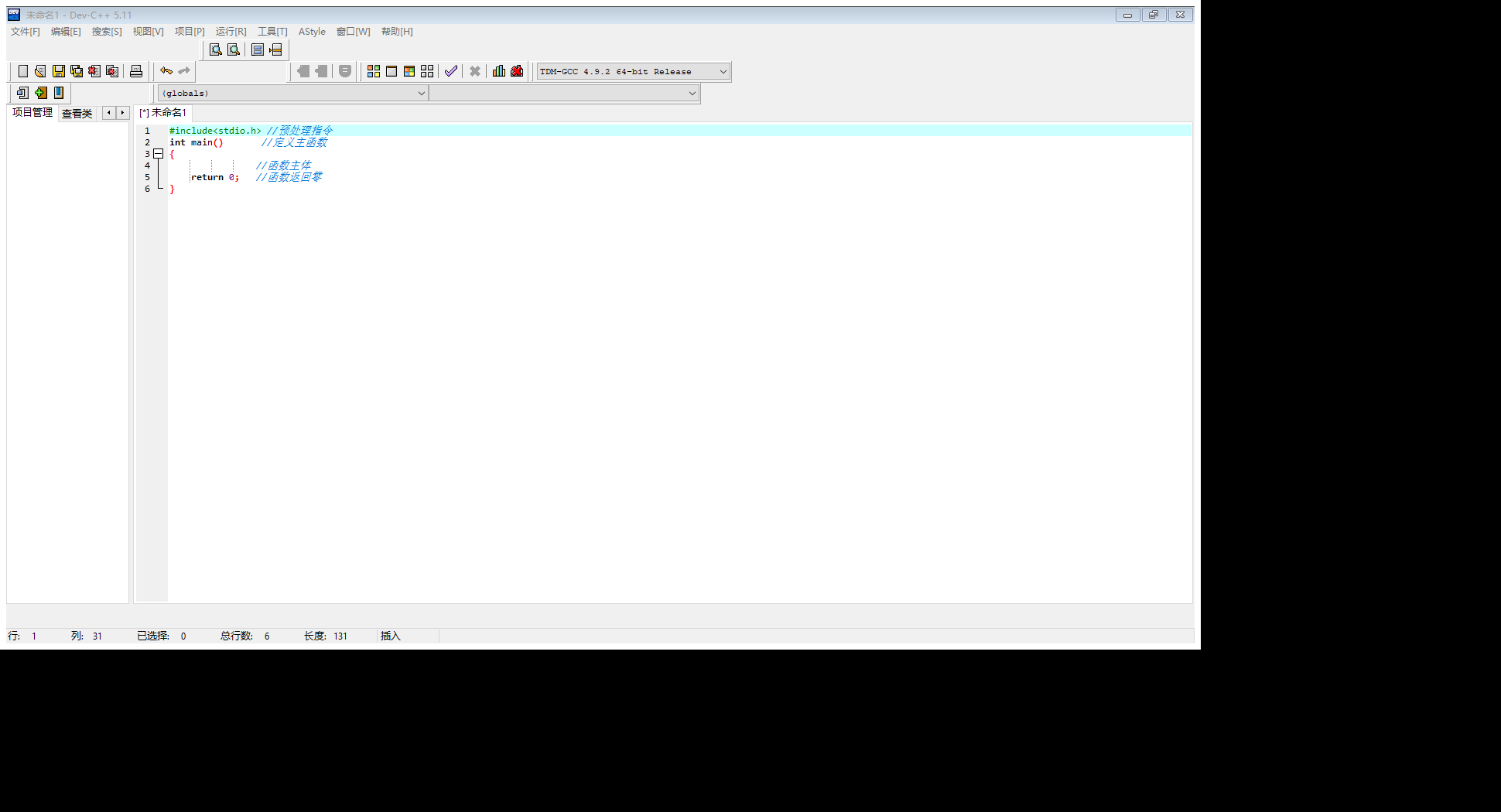
C程序设计学习小结

## 一·简单的程序结构设计——顺序结构

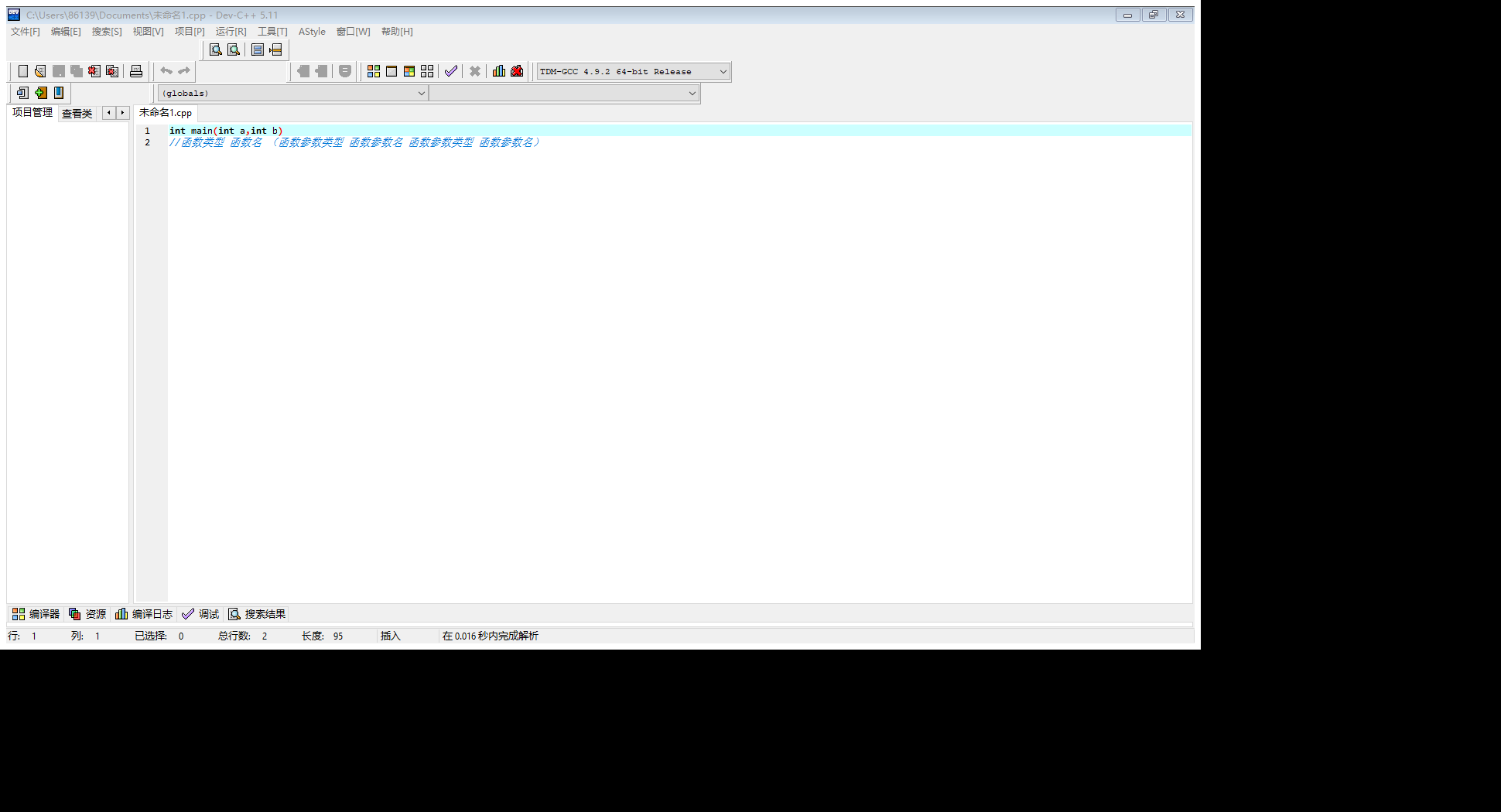


1. 简单结构
2. 预处理指令 eg. #include<stdio.h>

#include<math.h>

#define

1. 全局声明：在函数外进行的数据声明
2. 函数 函数首部



函数体——花括号内：声明部分 执行部分

**易错：**

**每个数据声明和语句后必有分号**

1. 数据的表现形式及其运算
2. 常量 整型常量

实型常量（指数形式AeB/AEB——A非零，B为整数）

字符常量 普通字符-单撇号括起的一个字符

转义字符

|  |  |  |
| --- | --- | --- |
|  |  |  |
| \o | 空字符(NULL) | 00H/0 |
| \n | 换行符(LF) | 0AH/10 |
| \r | 回车符(CR) | 0DH/13 |
| \t | 水平制表符(HT) | 09H/9 |
| \v | 垂直制表(VT) | 0B/11 |
| \a | 响铃(BEL) | 07/7 |
| \b | 退格符(BS) | 08H/8 |
| \f | 换页符(FF) | 0CH/12 |
| \’ | 单引号 | 27H/39 |
| \” | 双引号 | 22H/34 |

字符串常量 ——双撇号把若干个字符括起来

符号常量eg #define PI 3.1415

1. 变量 先定义后使用
2. 常变量 eg.const int=3;
3. 标识符 由字母，数字，下划线组成且第一个字符不为数字（大小写区分）
4. 数据类型 基本整型 int 短整型 short int 长整型 long int

单精度浮点型 float 双精度浮点型 double 字符型char

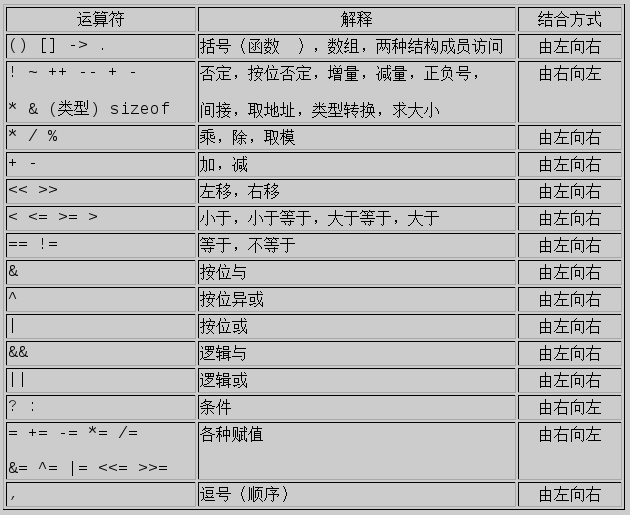
1. 运算符 \*整数/整数=整数（适当添加小数）

\*%要求操作数为整数

\*自增自减运算符 ++i，--i（使用i前先加减1）

i++,i—(使用i后加减i)

\*强制类型转换运算符 （类型名）（表达式）

\*C运算符

1. C语句
2. 分类： 控制语句 函数调用语句 表达式语句 空语句 复合语句
3. 基本语句：赋值语句 \*复合的赋值运算符 += -= \*= /= %=、

\*赋值表达式无分号，赋值表达式有分号

1. 数据的输入输出：printf函数 printf(格式控制，输出表列)

格式控制——格式声明，普通字符

输出表列——需要输出的数据

格式字符：%d有符号的十进制整数

%c字符 %m.nf单精度浮点数

%s字符串 %e指数形式

Scanf函数 scanf(格式控制，地址表列)

\*注意事项

1. 输入时与输入语句的格式对应
2. 输入数值时，在两个数值之间需要插入空格
3. 在连续输入字符时两个字符之间不要插入空格或其他分隔符
4. 在输入数值数据时，如输入空格，回车，Tab键或遇非法字符认为该数据结束

Putchar——输出字符

1. 既可以输出能在显示器上显示的字符也可以，也可以输出屏幕控制字符
2. 他输出的是字符而不能输出整数

Getchar——输入字符

只能接收单个字符

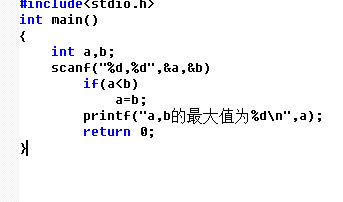
# 二．选择结构程序设计

1. 用if语句实现选择结构
2. if语句的一般形式 if（表达式） 语句1 if(表达式) 语句1

if（表达式） 语句1 else 语句2

else if（表达式2） 语句2

else if（表达式3） 语句3



\*整个if语句可以写在多行上也可以写在一行上

\*复合语句应当用花括号括起来

\*if后的条件不能加分号

1. 关系运算符和关系表达式

——用关系运算符将两个数值或数值表达式连接起来的式子称为关系表达式

\*（<,<=,>,>=）的优先级别相同，（==，!=）优先级别相同，前者的优先级别大于后者

\*关系运算符的优先级低于算术运算符

\*关系运算符的优先级高于赋值运算符

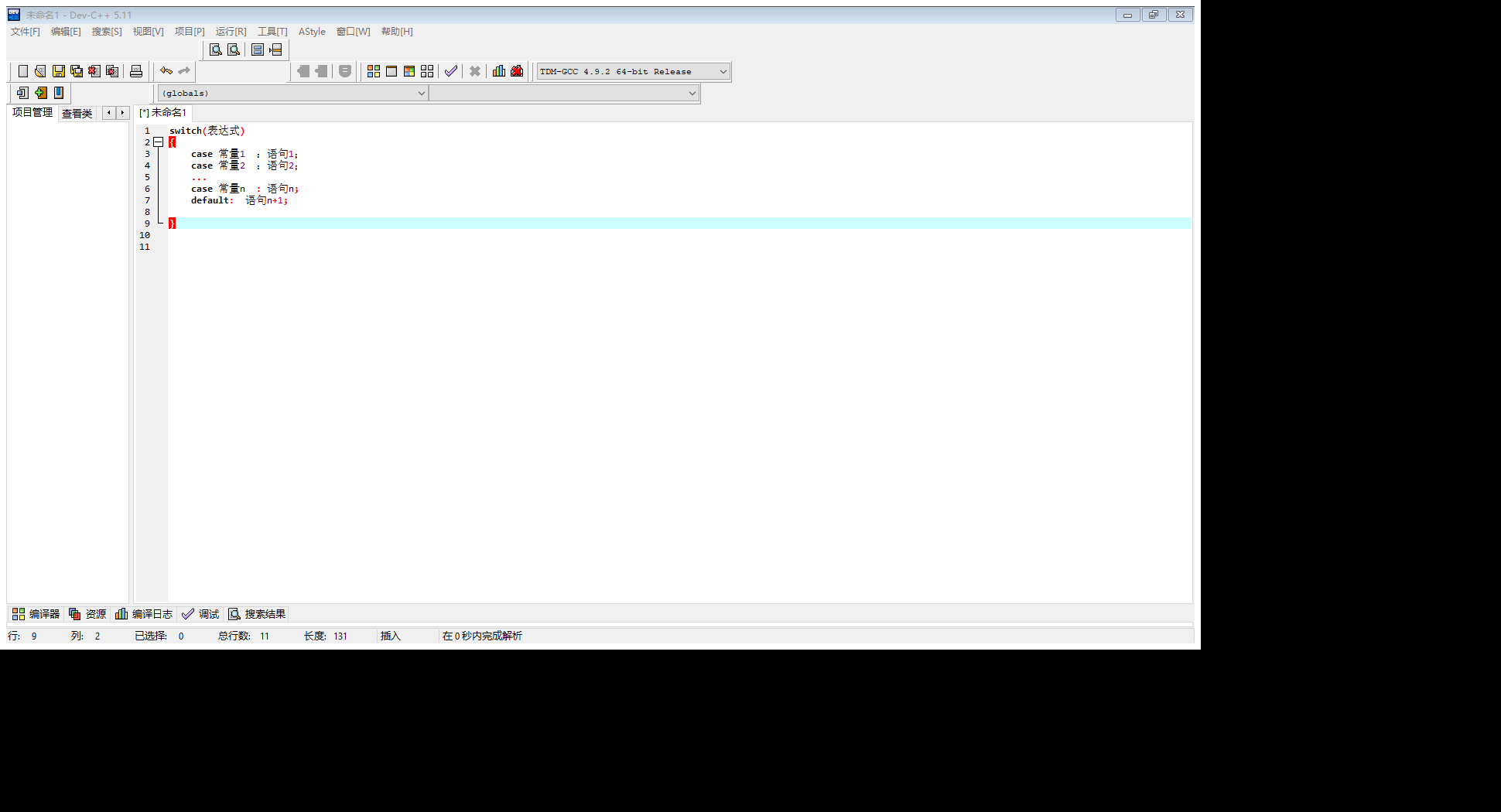
1. 逻辑运算符和逻辑表达式

且&& 或|| 非！

\*优先级 ！>&&>||

\*&&和||优先级低于关系运算符 ！高于算术运算符

1. 逻辑型变量bool true false
2. 用switch语句实现多分支选择结构

一般形式：

\*Switch后面括号内应为整数类型（包括字符型）

\*可以没有default标号

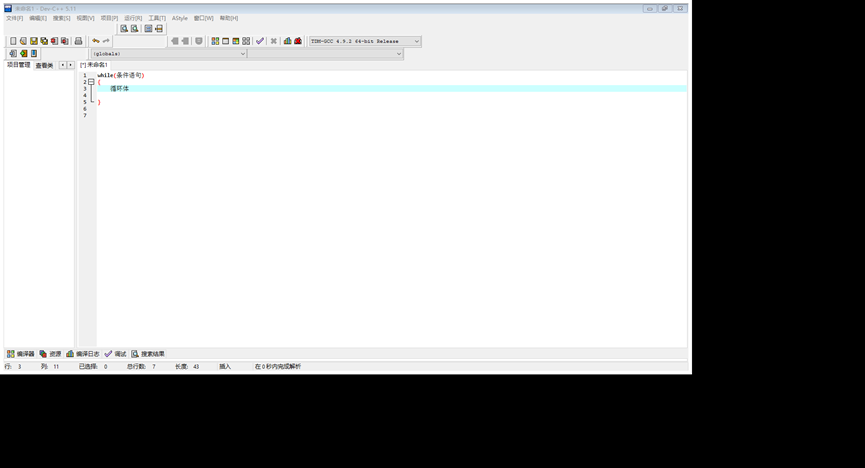
\*各个case标号出现的次序不影响执行结果

\*每一个case常量必须互不相同

\*case标号只起标记作用

\*多个case标号可以共用一组执行语句

# 三．循环结构程序设计

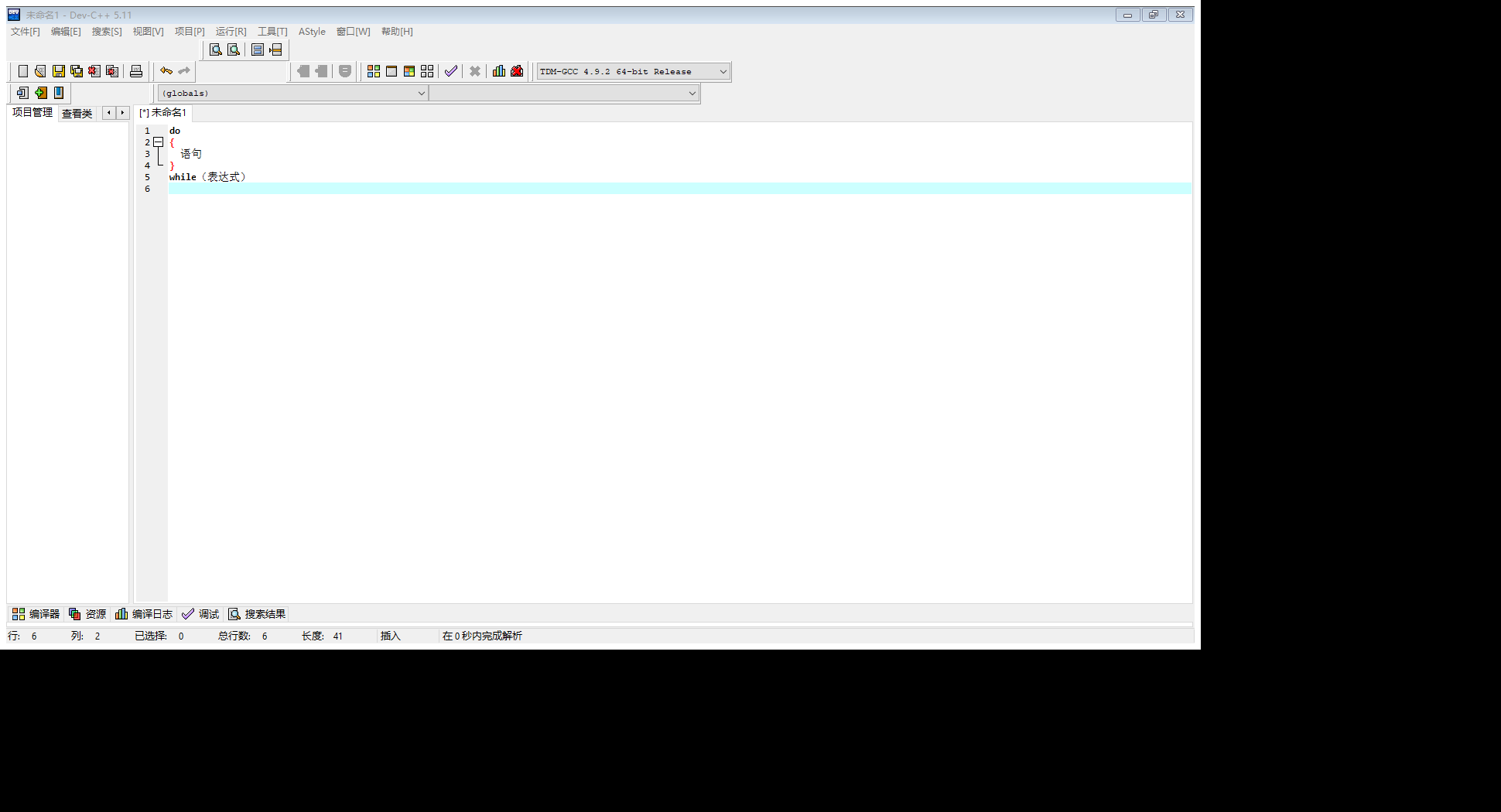
1. 用while语句实现循环——只要当循环条件表达式为真（及给定的条件成立），就执行循环语句体

…

\*循环体如果包含一个以上的语句，应该用花括号括起来

\*不要忽略给i和sum赋初值

\*在循环体中应有使循环体趋于结束的语句

1. 用do…while语句实现循环——无条件地执行循环体，然后判断循环条件是否成立
2. 用for语句实现循环

For(表达式1；表达式2；表达式3)

表达式1：设置初始条件，只执行一次，可以为零个，一个或多个变量设置初值。

表达式2：是循环条件表达式，用来判断是否继续循环。在每次执行循环体前先执行此表达式，决定是否继续执行循环。

表达式3：作为循环的调整，例如使循环变量增值，他是在执行循环体后才进行的

\*表达式1可以省略，既不设置初始值，但表达式1后的分号不能省略

\*表达式2也可以省略，既不用表达式2作为循环条件表达式，不设置和检查循环条件此时循环无终止的循环下去

\*表达式3也可以省略，但此时程序设计者应另外设法保证循环能正常结束

\*表达式1可以设置循环变量初值的赋值表达式，也可以是与循环变量无关的其他表达式。

\*表达式1和表达式3可以是一个简单的表达式，也可以是逗号表达式。

\*表达式2一般是关系表达式或逻辑表达式，但也可以是数值表达式或字符表达式，只要其值为非零，就执行循环体。

1. 改变循环执行的状态

用break语句提前终止循环——使流程跳到循环体之外，接着执行循环体下面的语句

（只能用于循环语句和switch语句之中，而不能单独使用）

用continue语句提前结束本次循环——结束本次循环，既跳过循环体中下面未执行的 语句，跳到循环体结束点之前，接着执行for语句中的表达式3，然后进行下一次是否执行循环的判断

# 四．利用数组处理批量数据

1. 定义和引用一维数组

类型名 数组名[ 常量表达式 ]

\*方括号内的常量表达式用来表示元素的个数，既数组长度(下标是从零开始的)

\*常量表达式中可以包括常量和符号常量，不能包含变量

\*只能引用数组元素而不能一次性调用整个数组全部元素的值。（利用循环语句处理整个数组）

\*在定义数组是可对全部或部分数组赋值，未赋值的数组元素默认为零

全部赋值时可省略方括号内长度

1. 定义和使用二维数组

类型说明符 数组名[ 常量表达式] [常量表达式]

（与一维数组类似）

\*内层花括号内为行，外层花括号内为列