

C语言程序设计第二章

——数据类型和表达式

主讲人：李伦彬



CONTENTS

章节复习

引 论

常 量

变 量

运算符和表达式

思维导图



章节复习

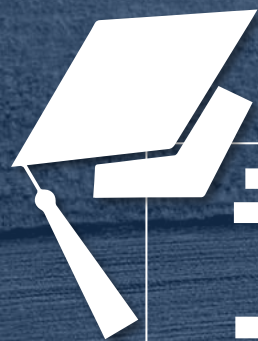
引论

常量

变量

运算符和表达式

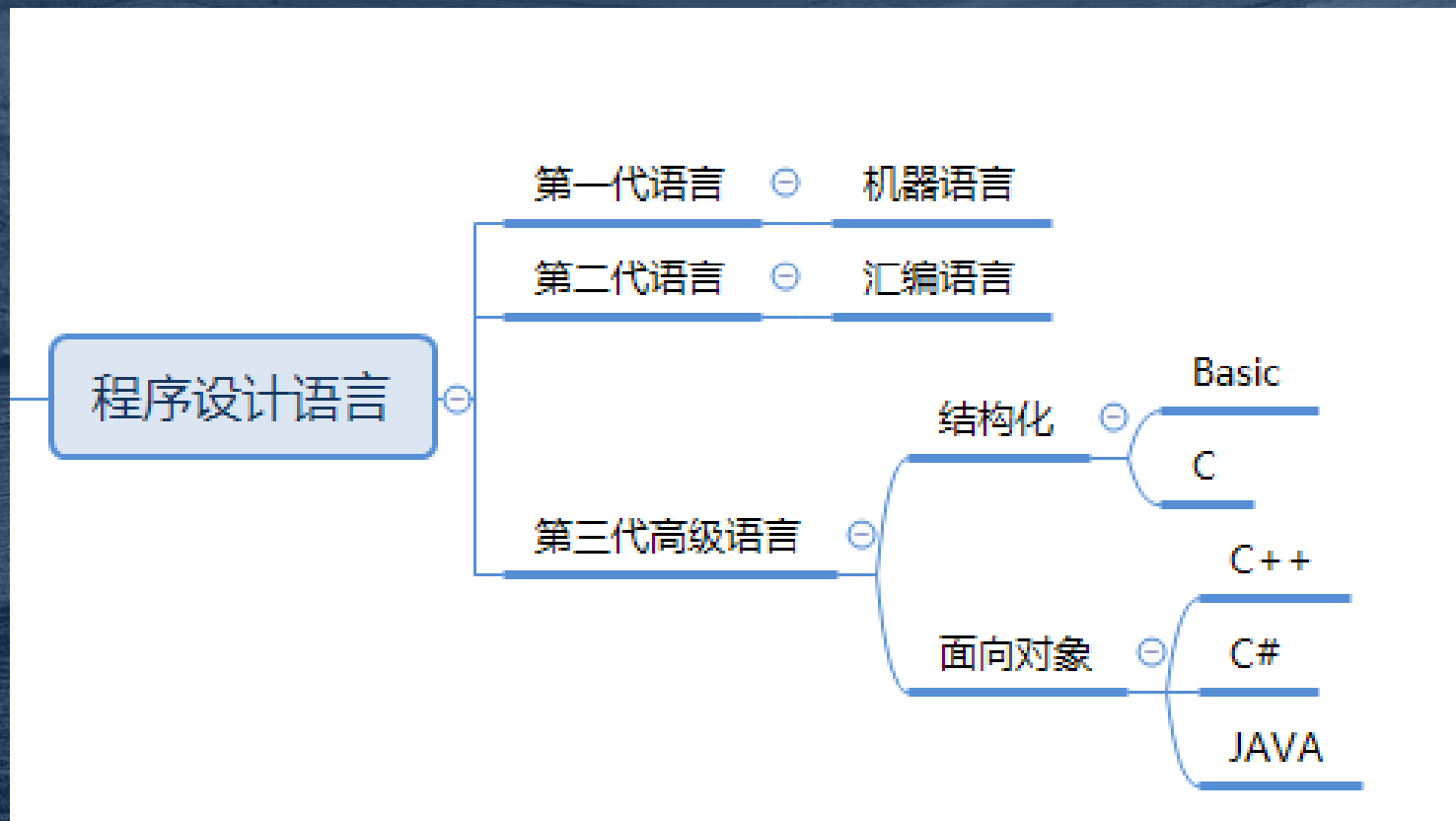
思维导图



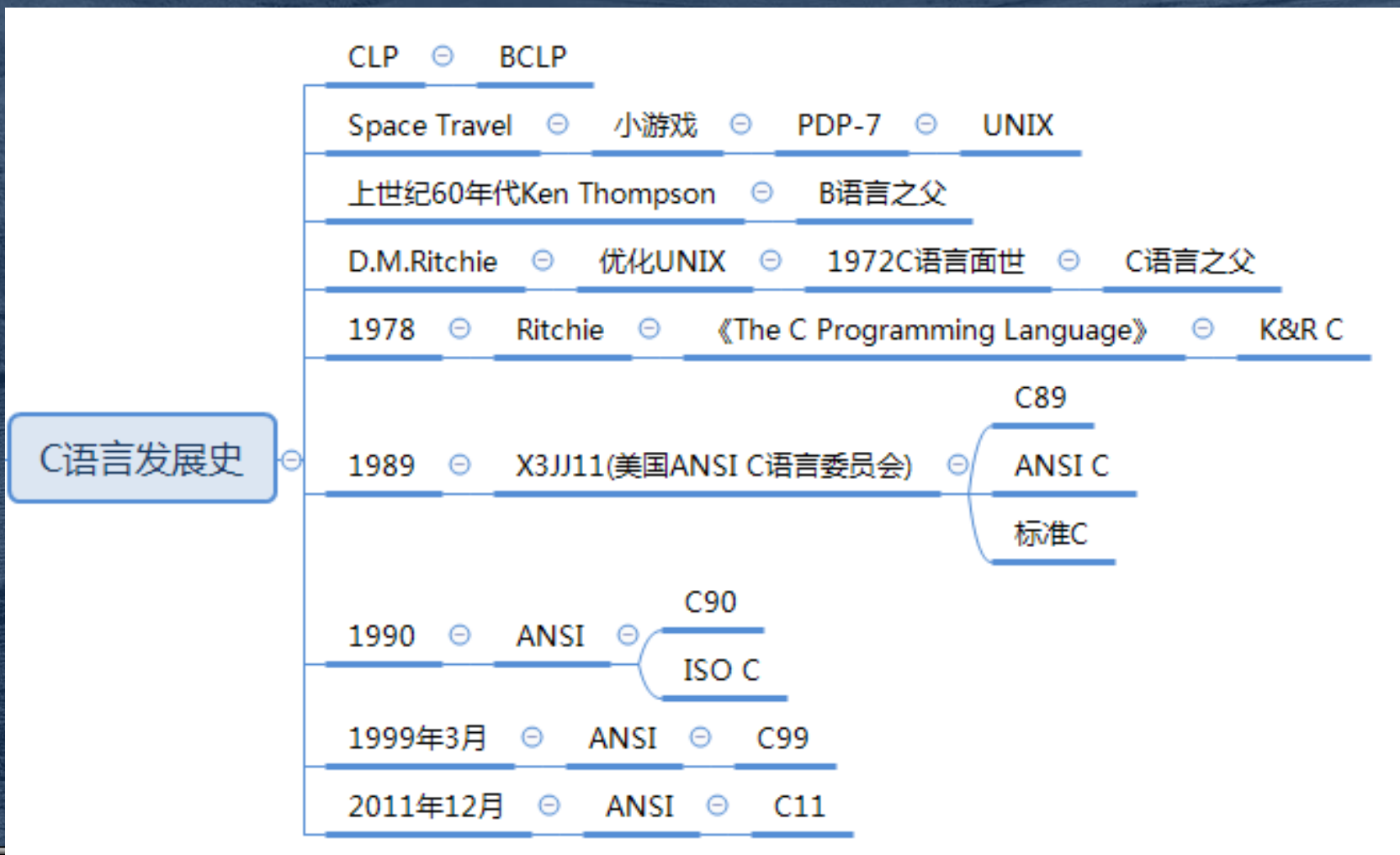
章节复习



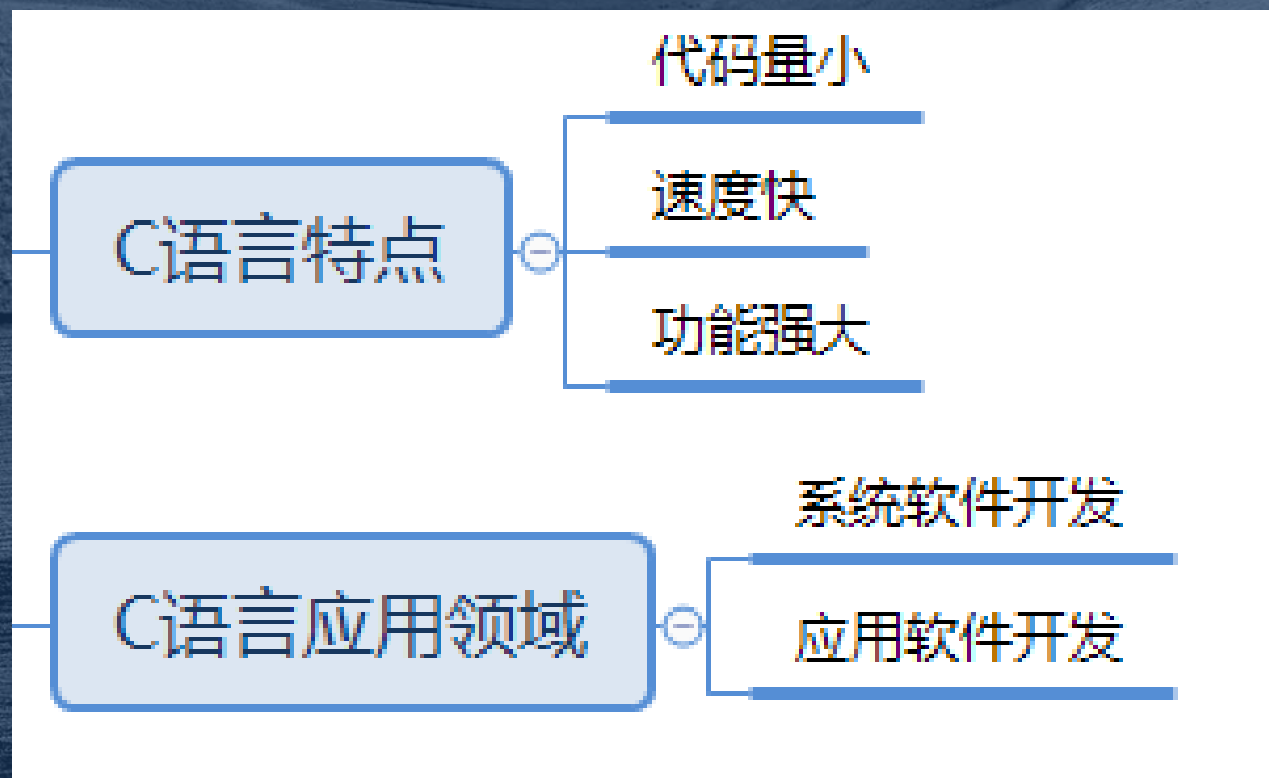
程序设计语言部分思维导图



C语言发展史部分思维导图



C语言特点及应用领域部分思维导图



章节复习

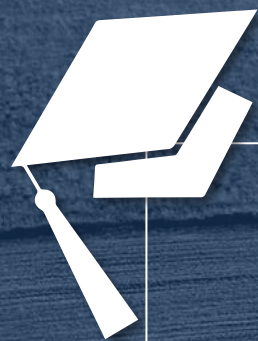
引论

常量

变量

运算符和表达式

思维导图



引论





本节内容

计算机内存

数据类型



计算机内存

内存用来存放数据和程序，一般指**RAM**，断电后信息丢失。

最小单位：**位**

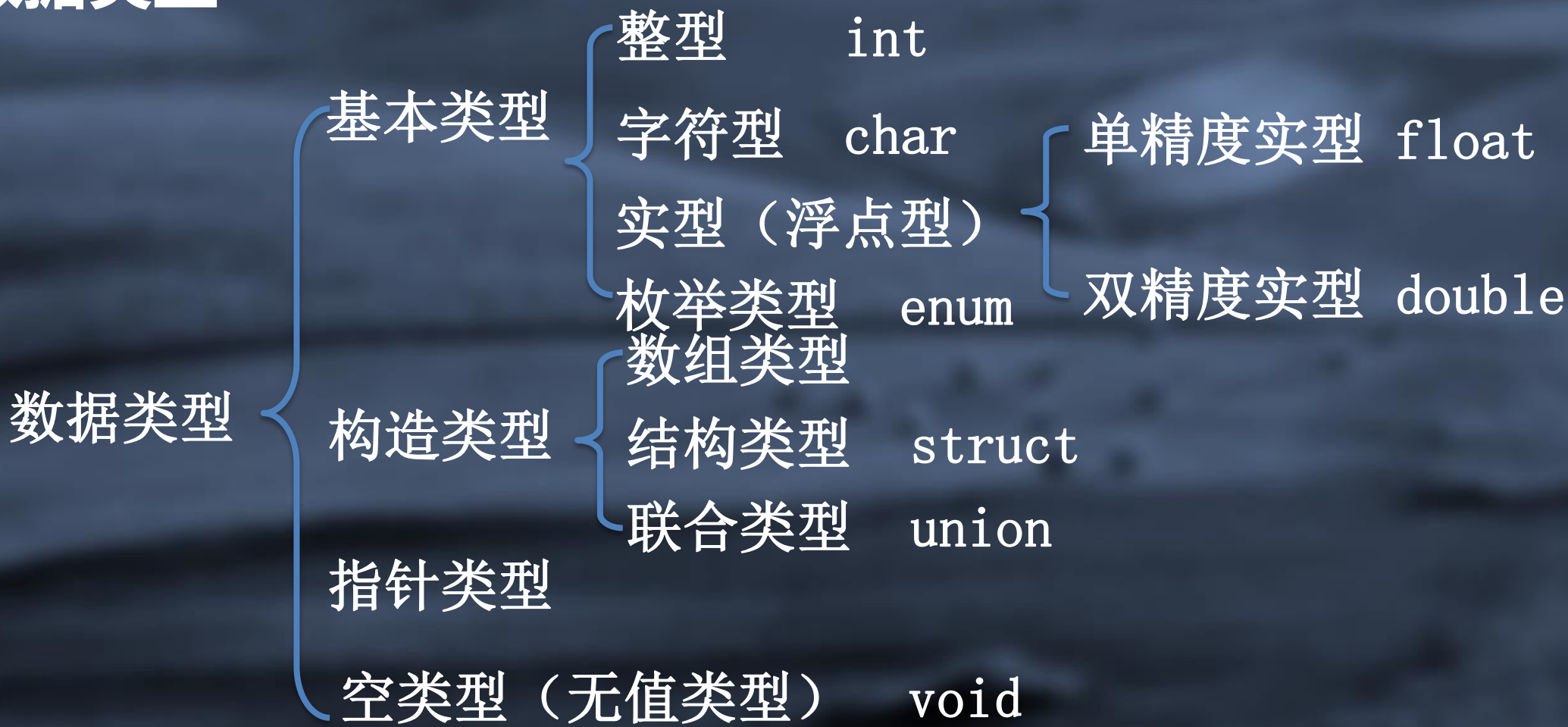
最基本单位：**字节**

1字节 = 8位

内存以**字节**为单位存储和访问



数据类型



章节复习

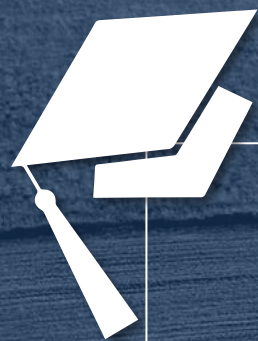
引论

常量

变量

运算符和表达式

思维导图



常量





本节内容

常量的定义

常量的分类

练习巩固



常量的定义：

- 在程序运行过程中, **其值不能被改变**的量称为常量。
- 常量按表示形式分为**整型常量**、**实型常量**、**字符常量**、**字符串常量**和**符号常量**。

整型常量

- 十进制数：由数字**0~9**和**正负号**表示，如123，-24。
- 八进制数：以**数字0**开头，后跟数字**0~7**，如0123,-011。
- 十六进制数：以**0x或0X**开头，后跟**数字0~9和字母a~f或A~F**，如0x123,0xFF。

实型常量

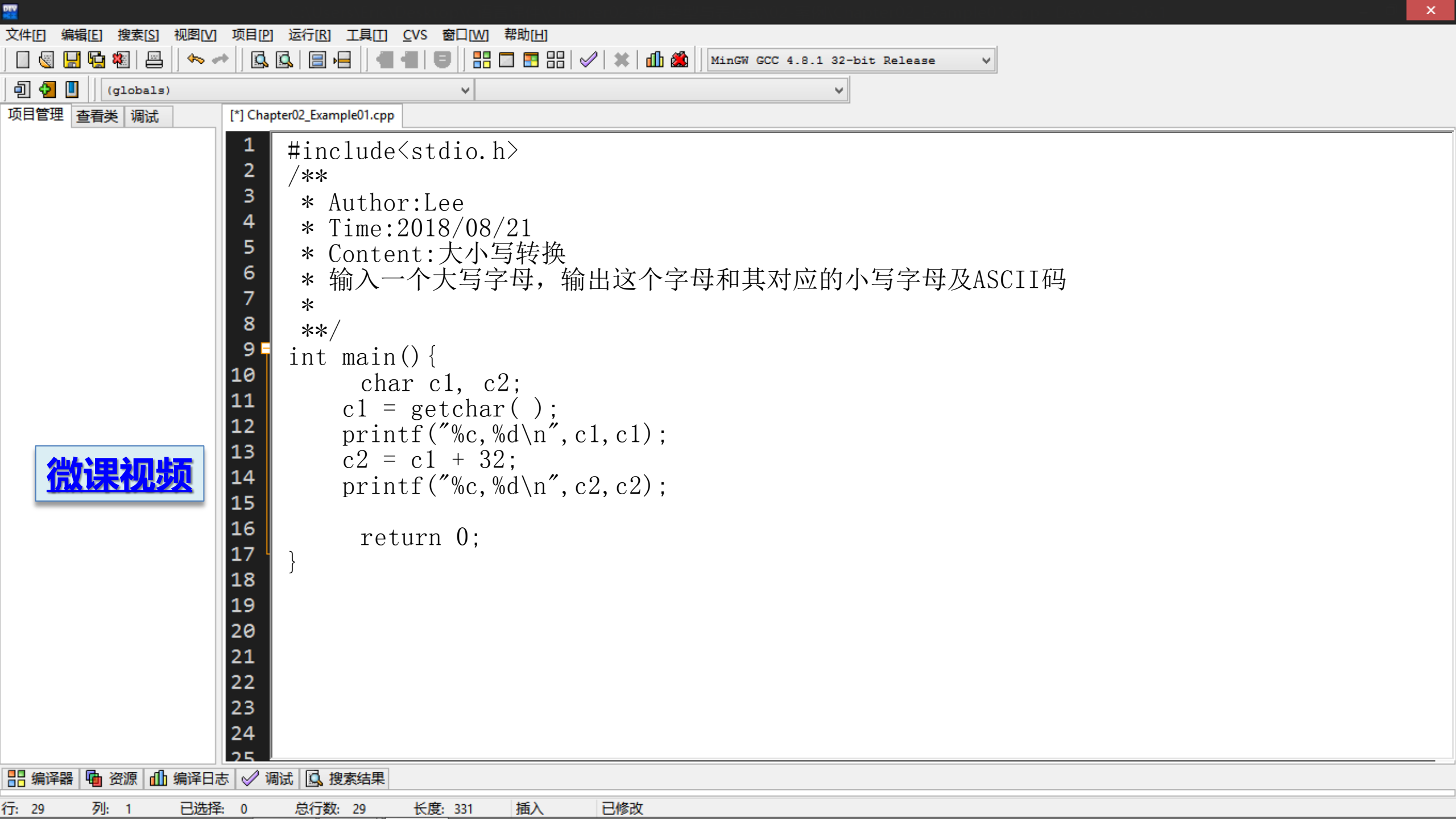
- 十进制小数形式：由数字**0~9**和**小数点**组成（必有**小数点**），如1.23，.4。
- 指数形式：即**科学记数法**，由**十进制数和指数标志e**或**E及指数值**组成，即3.1e4，.34e2。
- E（e）前**必须有数字**，E（e）后**必须是整数**。

字符型常量

- 字符常量只能用**单引号括起来的单个字符**。
- 转义字符，以**“\”**开头，后跟**一个或几个字符**。

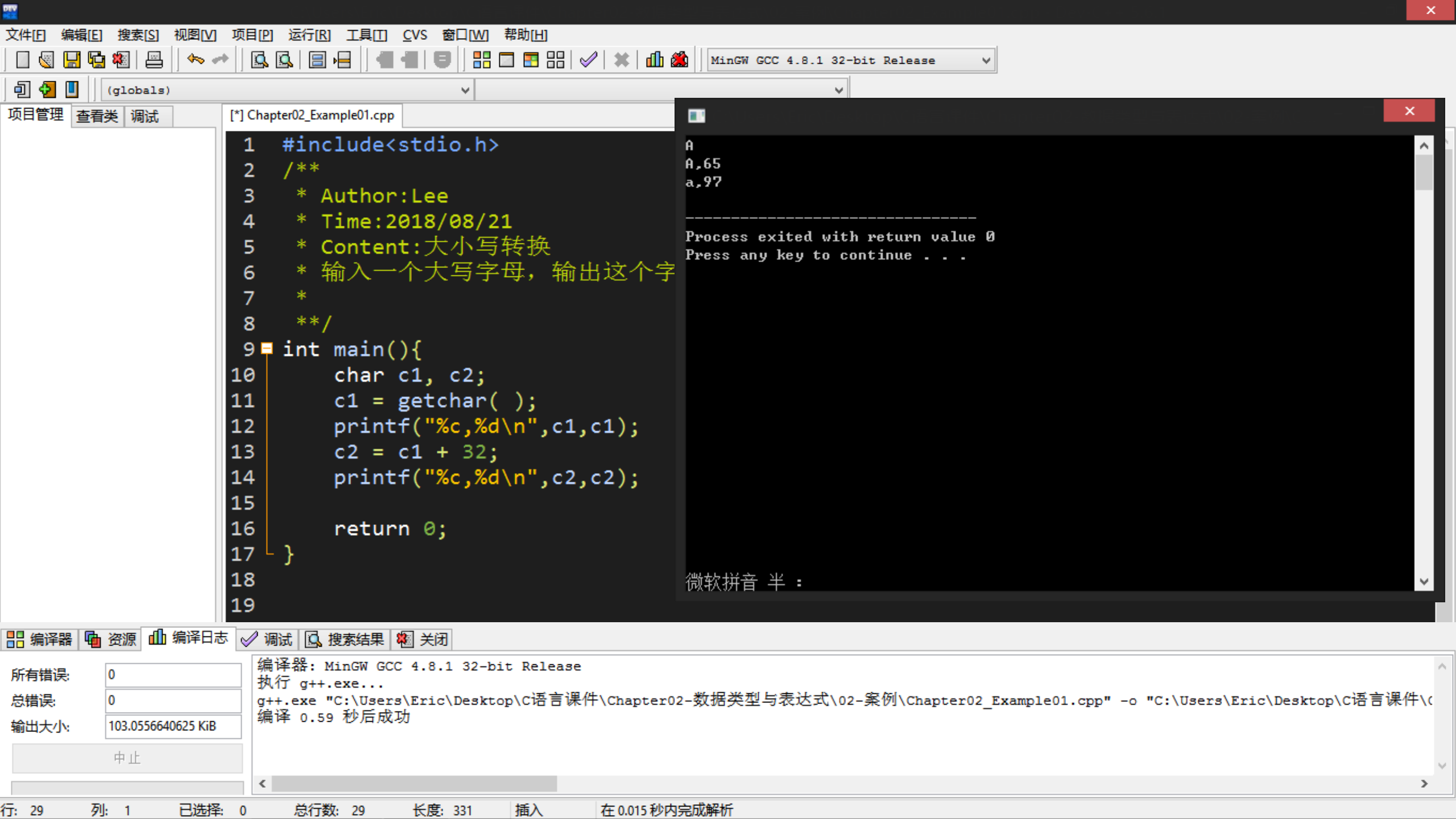
注：

- (1) C语言对字母**大小写敏感**；
- (2) 字符常量在计算机中存储的是**ASCII值**；
- (3) **字符类型的数据可像整型数据一样运算**



微课视频

```
1 #include<stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:大小写转换
6  * 输入一个大写字母, 输出这个字母和其对应的小写字母及ASCII码
7  *
8  **/
9 int main() {
10     char c1, c2;
11     c1 = getchar( );
12     printf("%c,%d\n", c1, c1);
13     c2 = c1 + 32;
14     printf("%c,%d\n", c2, c2);
15
16     return 0;
17 }
18
19
20
21
22
23
24
25
```



字符串常量

是由一对**双引号扩起来的字符序列**

'a' 是字符常量, " a" 是字符串常量, 二者不同。

如: 假设 C 被指定为字符变量

```
char c
```

```
c = 'a' ;
```

对

```
c = " a" ; c = " CHINA" ;
```

错

结论: 不能把一个字符串常量赋给一个字符变量。

字符型常量和字符串型常量的区别

- 1. 字符常量是由**单引号**括起来，字符串常量是由**双引号**扩起来。
- 2. 字符常量只表示**1个字符**，字符串常量可以包含**0个或多个字符**。
- 3. 可以将一个字符常量赋给一个字符变量，而不能将字符串常量赋给一个字符变量。
- 4. 字符常量在内存中占**一个字节**，字符串常量占用的存储单元是**字符个数加一**。

符号常量

用一个**标识符**来代表一个常量，称为符号常量

一般定义格式

`#define 标识符 常量`

例：`#define N 10`

习惯上符号常量名用**大写形式**





练习巩固

- Problem description
输入一个圆的半径，要求输出圆的面积，其中圆周率定义为符号常量。
- Input
输入占一行。输入一个实数 r ，代表圆的半径
- Output
输出占一行。输出“ $\text{area} = s$ ”， s 为一个实数，代表圆的面积，保留两位小数


```

1  #include<stdio.h>
2  #define PI 3.14
3  /**
4   * Author:Lee
5   * Time:2018/08/21
6   * Content:常量练习巩固
7   *
8   */
9  int main(){
10     float r,s;
11     scanf("%f",&r);
12     s=PI*r*r;
13     printf("s=%.2f\n",s);
14
15     return 0;
16 }
17
18
19

```

```

2
s=12.56

-----
Process exited with return value 0
Press any key to continue . . .

```

微软拼音 半 :

所有错误:	0
总错误:	0
输出大小:	101.8828125 KiB

中止

```

编译器: MinGW GCC 4.8.1 32-bit Release
执行 g++.exe...
g++.exe "C:\Users\Eric\Desktop\C语言课件\Chapter02-数据类型与表达式\02-案例\Chapter02_homework01.cpp" -o "C:\Users\Eric\Desktop\C语言课件\
编译 0.63 秒后成功

```


章节复习

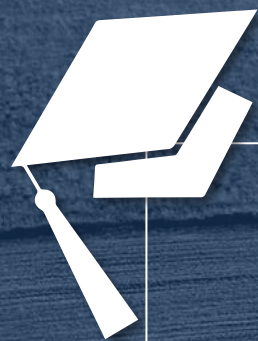
引论

常量

变量

运算符和表达式

思维导图



变量





本节内容

变量的定义

变量的初始化

变量的分类及示例



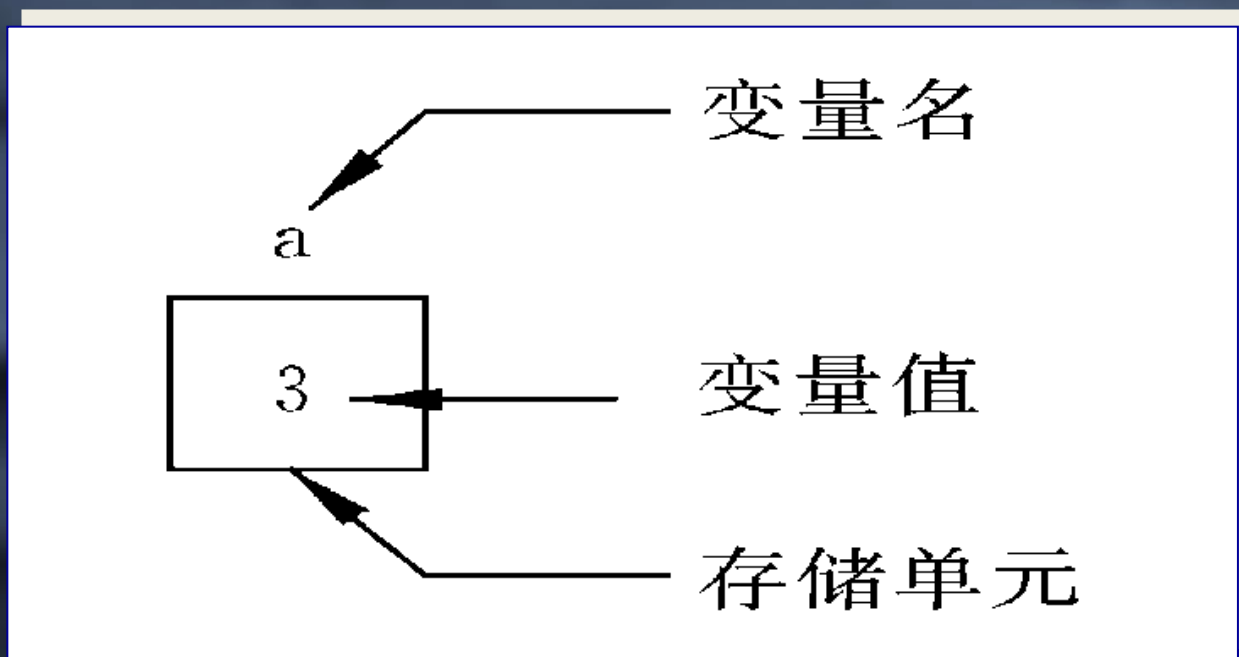
变量的定义：

- 在程序运行过程中, **其值可以改变**的量称为变量。
- 变量代表内存中具有特定属性的一个**存储单元**, 它用来存放数据, 这就是变量的值。

变量的理解：

□ 变量名实际上是一个以一个名字对应代表一个地址。

□ 例 `int a=3;`



变量的命名规则：

C语言规定标识符只能由**字母**、**数字**和**下划线**三种字符组成，
且**第一个字符必须为字母或下划线**。

□ 正确的变量命名：

sum, _total, month, BASIC, Student_name

□ 错误的变量命名：

M. D. John, 123, 3D64, a>b

变量的赋值：

(1) C语言允许在**定义变量的同时使变量初始化**

如：

```
int a=3;           // 指定 a 为整型变量，初值为 3
```

```
float f=3.56;     //指定 f 为浮点型变量，初值为 3.56
```

```
char c= 'a' ;    // 指定 c 为字符变量，初值为 'a'
```

变量的赋值：

(2) 可以使被定义的变量的一部分赋初值。

如：`int a, b, c=5;`

//表示指定 a、b、c 为整型变量，但只对 c 初始化，c 的初值为 5

变量的赋值：

(3) 如果对几个变量赋以同一个初值

应写成：`int a=3, b=3, c=3;` 表示 `a` , `b` , `c` 的初值都是 3。

不能写成：`int a=b=c=3;`

注意：初始化不是在编译阶段完成的而是在程序运行时执行本函数时赋初值的，相当于有一个赋值语句。



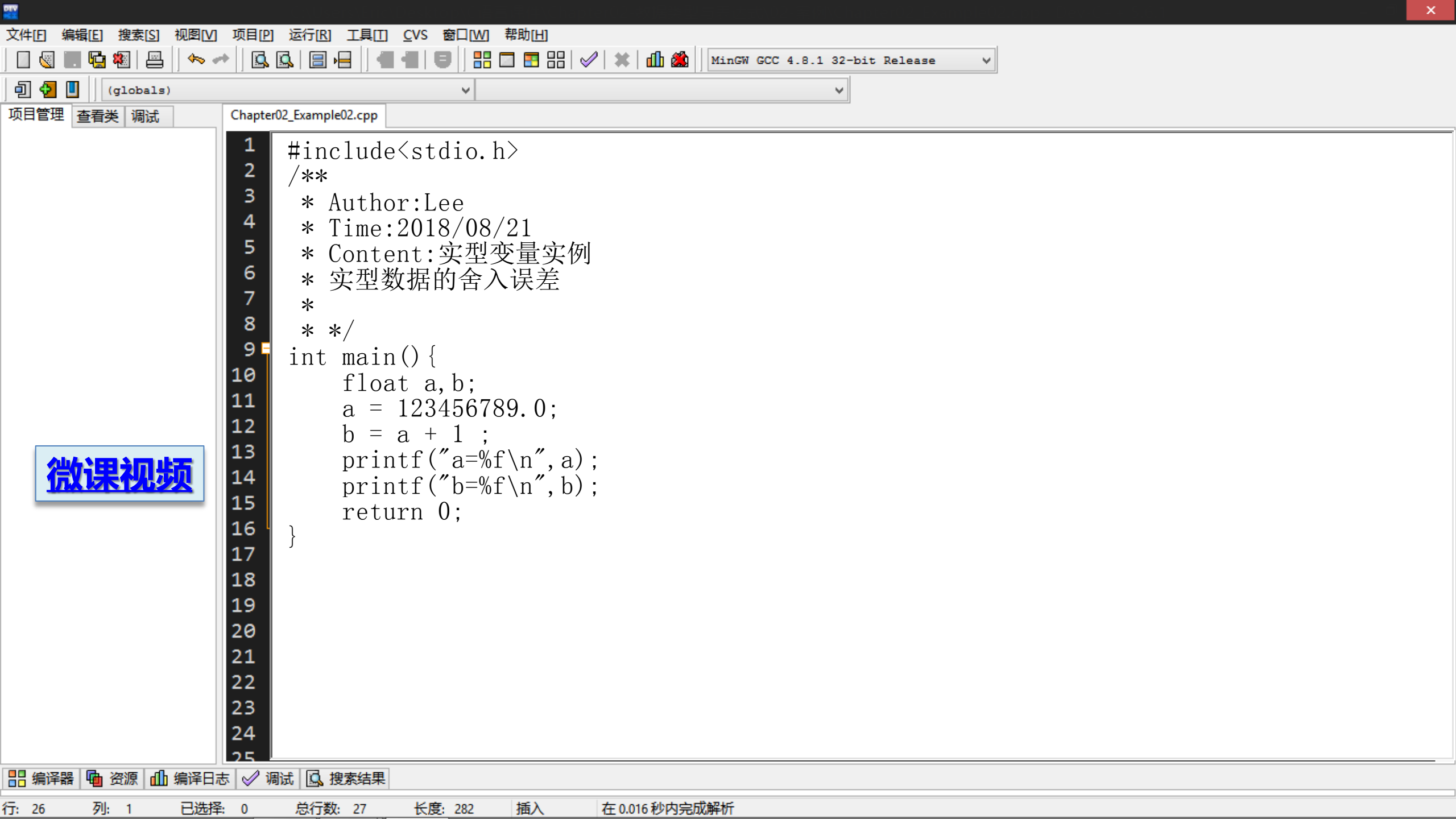
整型变量的定义

- `int a,b` (指定变量 `a`、`b` 为整型)
- `short c,d;` (指定变量 `c`、`d` 为短整型)
- `long e,f;` (指定变量 `e`、`f` 为长整型)

实型变量的分类

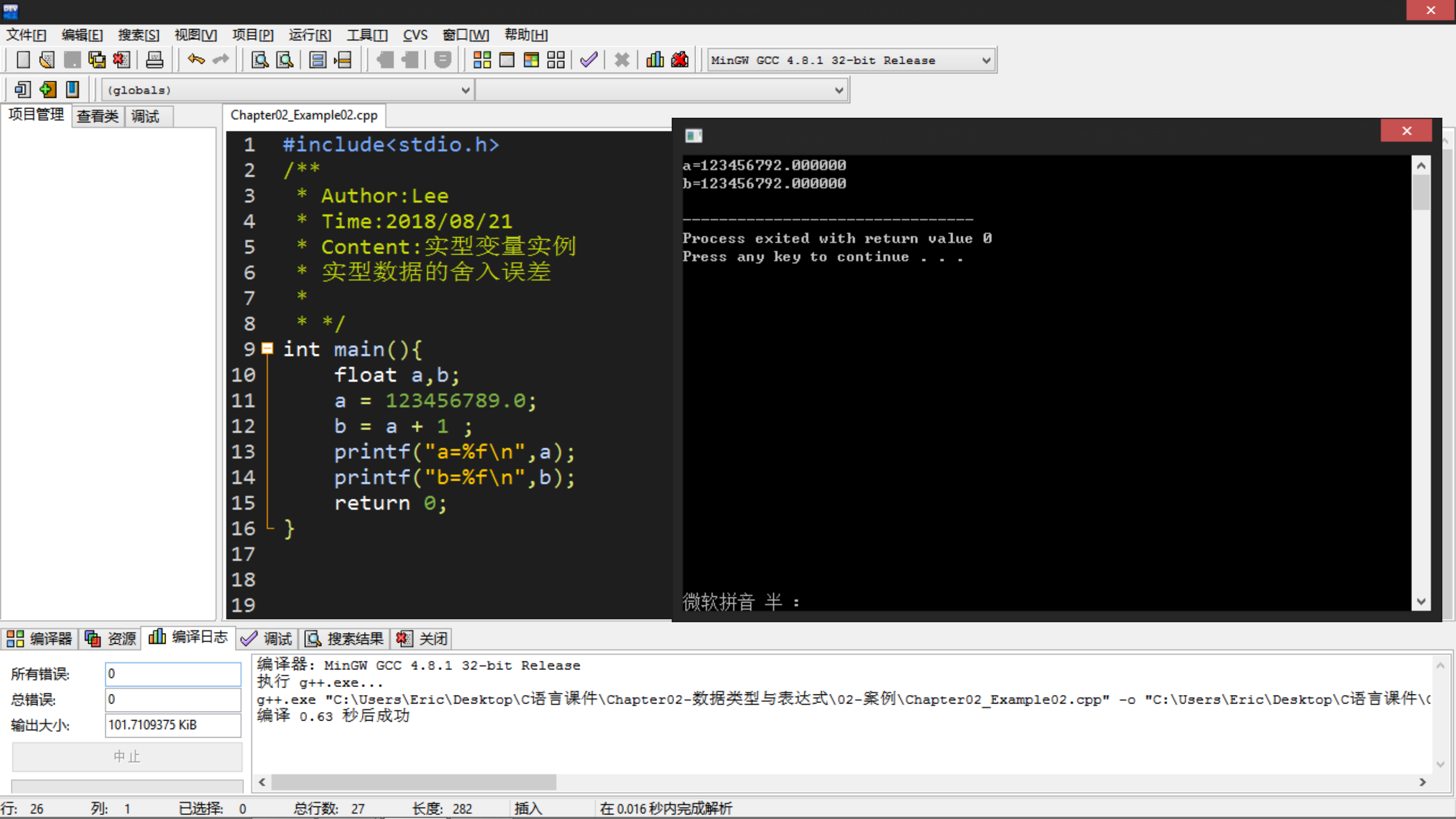
实型变量分**单精度 (float)** 和**双精度 (double)**。

类型	位数	数的范围	有效数字
float	32	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	7位
Double	64	$-1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$	16位



```
1 #include<stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:实型变量实例
6  * 实型数据的舍入误差
7  *
8  * */
9 int main() {
10     float a,b;
11     a = 123456789.0;
12     b = a + 1 ;
13     printf("a=%f\n", a);
14     printf("b=%f\n", b);
15     return 0;
16 }
17
18
19
20
21
22
23
24
25
```

微课视频



```
1 #include<stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:实型变量实例
6  * 实型数据的舍入误差
7  *
8  * */
9 int main(){
10     float a,b;
11     a = 123456789.0;
12     b = a + 1 ;
13     printf("a=%f\n",a);
14     printf("b=%f\n",b);
15     return 0;
16 }
17
18
19
```

```
a=123456792.000000
b=123456792.000000

-----
Process exited with return value 0
Press any key to continue . . .

微软拼音 半 :
```

编译器: MinGW GCC 4.8.1 32-bit Release
执行 g++.exe...
g++.exe "C:\Users\Eric\Desktop\C语言课件\Chapter02-数据类型与表达式\02-案例\Chapter02_Example02.cpp" -o "C:\Users\Eric\Desktop\C语言课件\
编译 0.63 秒后成功

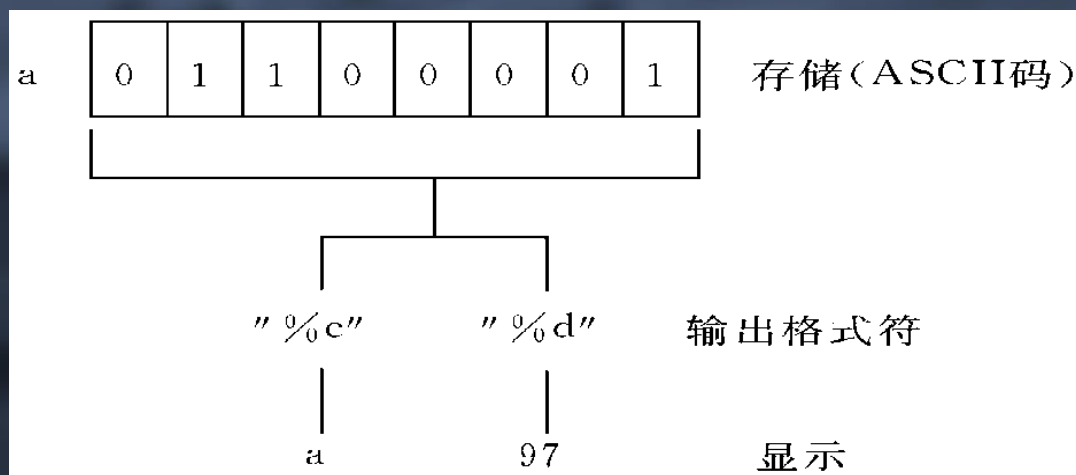
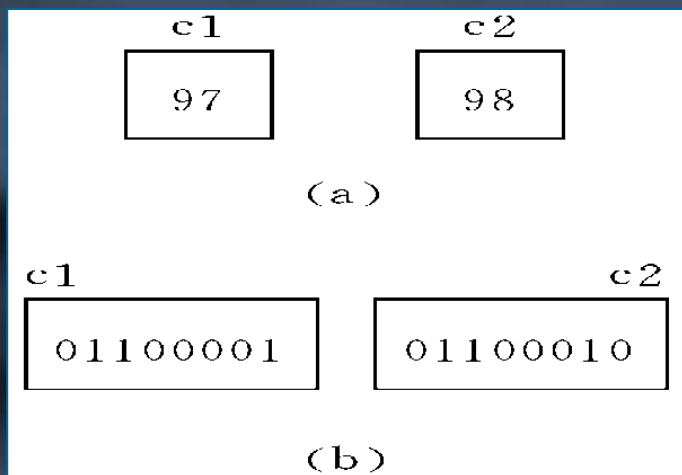
所有错误:	0
总错误:	0
输出大小:	101.7109375 KiB

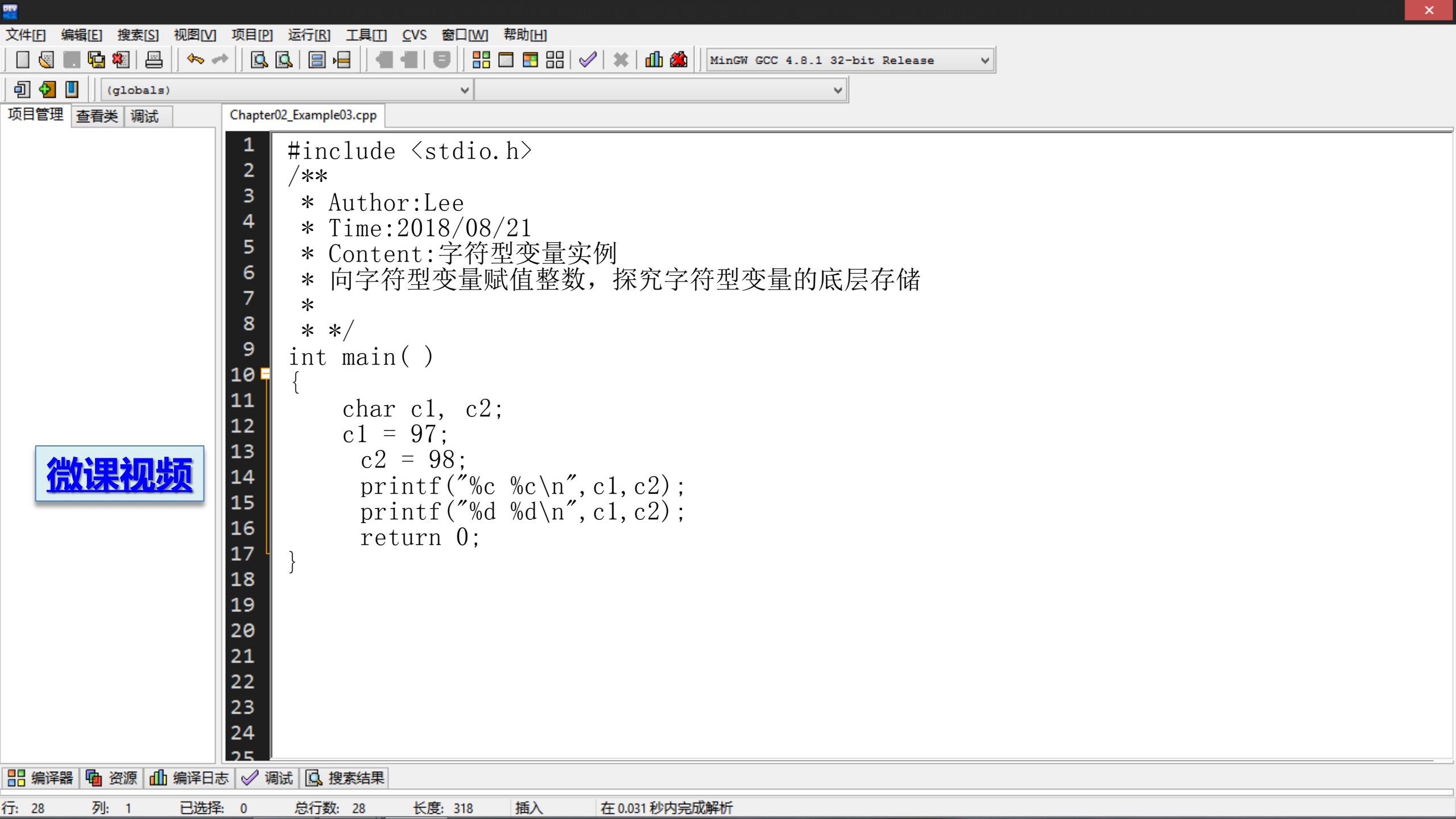
中止

行: 26 列: 1 已选择: 0 总行数: 27 长度: 282 插入 在 0.016 秒内完成解析

字符型变量在内存中的存储

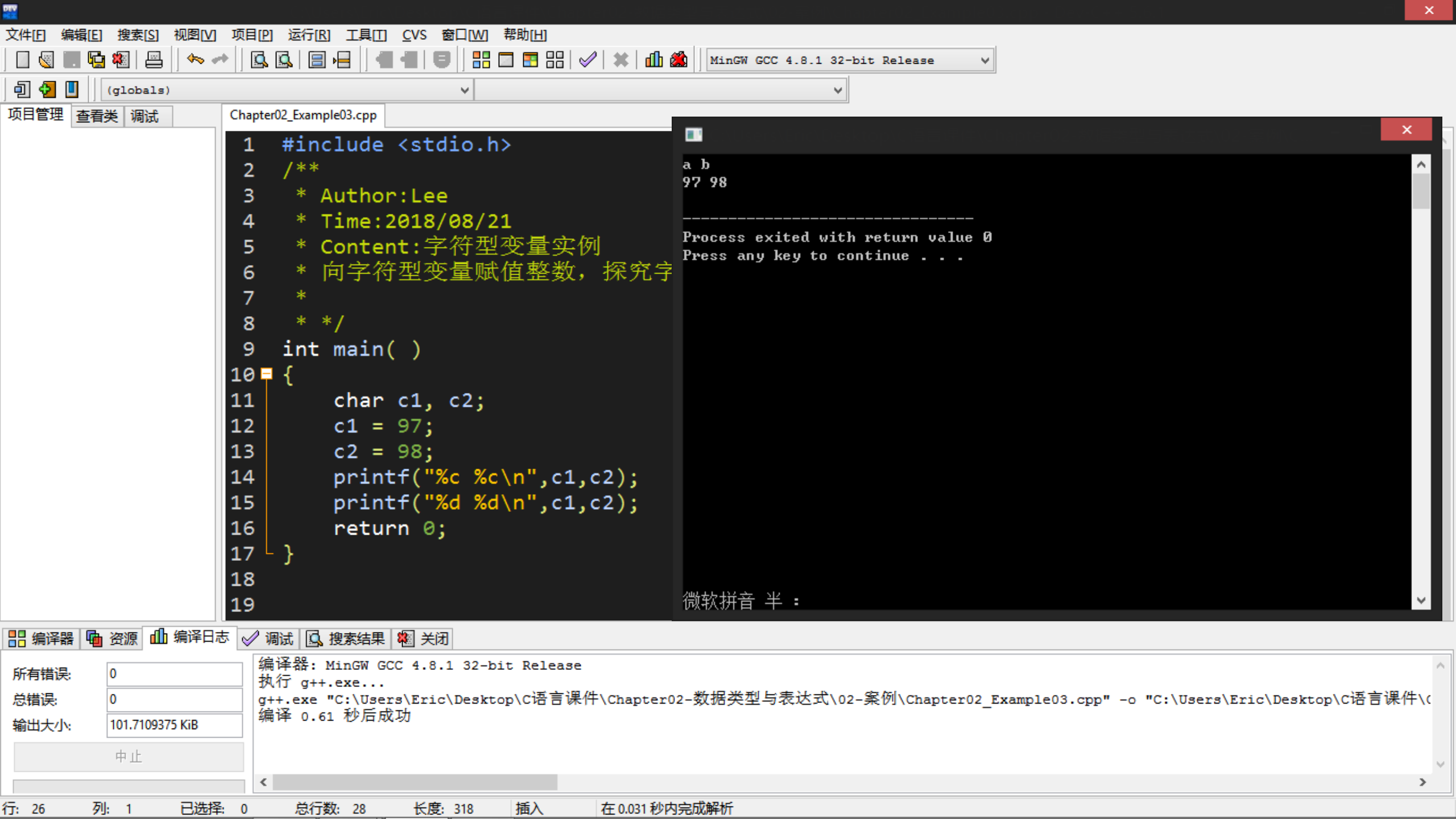
- 字符在内存中是将相应的**ASCII码**放到存储单元中。
- 一个字符数据**既可以以字符形式输出，也可以以整数形式输出。**





```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:字符型变量实例
6  * 向字符型变量赋值整数，探究字符型变量的底层存储
7  *
8  * */
9 int main( )
10 {
11     char c1, c2;
12     c1 = 97;
13     c2 = 98;
14     printf("%c %c\n", c1, c2);
15     printf("%d %d\n", c1, c2);
16     return 0;
17 }
18
19
20
21
22
23
24
25
```

微课视频



```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:字符型变量实例
6  * 向字符型变量赋值整数，探究字
7  *
8  * */
9 int main( )
10 {
11     char c1, c2;
12     c1 = 97;
13     c2 = 98;
14     printf("%c %c\n",c1,c2);
15     printf("%d %d\n",c1,c2);
16     return 0;
17 }
18
19
```

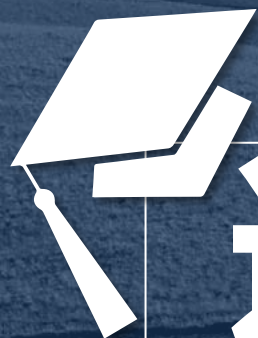
```
a b
97 98

-----
Process exited with return value 0
Press any key to continue . . .
```

微软拼音 半 :

所有错误:	0
总错误:	0
输出大小:	101.7109375 KiB
中止	

编译器: MinGW GCC 4.8.1 32-bit Release
执行 g++.exe...
g++.exe "C:\Users\Eric\Desktop\C语言课件\Chapter02-数据类型与表达式\02-案例\Chapter02_Example03.cpp" -o "C:\Users\Eric\Desktop\C语言课件\
编译 0.61 秒后成功



运算符和 表达式



本节内容

相关概念

算数运算符

条件运算符

逻辑运算符

赋值运算符

逗号运算符

条件运算符

练习巩固



相关概念：

- **运算符**：在C语言中用来表示某种计算的符号
- **操作数**：运算符操作的对象，可以为变量（已赋值）、常量或其他有确切值的表达式。
- **表达式**：运算符和操作数组成的具有确切值的符合C语言语法的式子称为表达式。

运算符的分类：

□ 单目运算符（一元运算符）

只有1个操作数。

□ 双目运算符（二元运算符）

具有2个操作数。（主要类型）

□ 三目运算符（三元运算符）

需要3个操作数。

（唯一：条件运算符 `?` `:`）



运算符的优先级和结合性：

□ 运算符优先级：

某些运算符先于其他运算符被执行。

例如， $x + y * 4$ ，先乘除后加减

□ 运算符的结合性：

当出现并列的运算优先级别相同的运算符时，由运算符的结合性决定计算的次序。

例如： $x * y / 4$ ；从左往右依次进行计算

算数运算符

❖ 基本算术运算符： $+$ $-$ $*$ $/$ $\%$

结合方向：从左向右（左结合）

优先级： $-$ 高于 $*$ $/$ $\%$ 高于 $+$ $-$

❖ 说明：

“-” 可为单目运算符时,右结合性

两整数相除，结果为整数，如 $1/2$ ，结果0

$\%$ 要求两侧均为整型数据，余数符号与被除数一致

$+$ $-$ $*$ $/$ 中任意数据类型为实型，结果均为实型

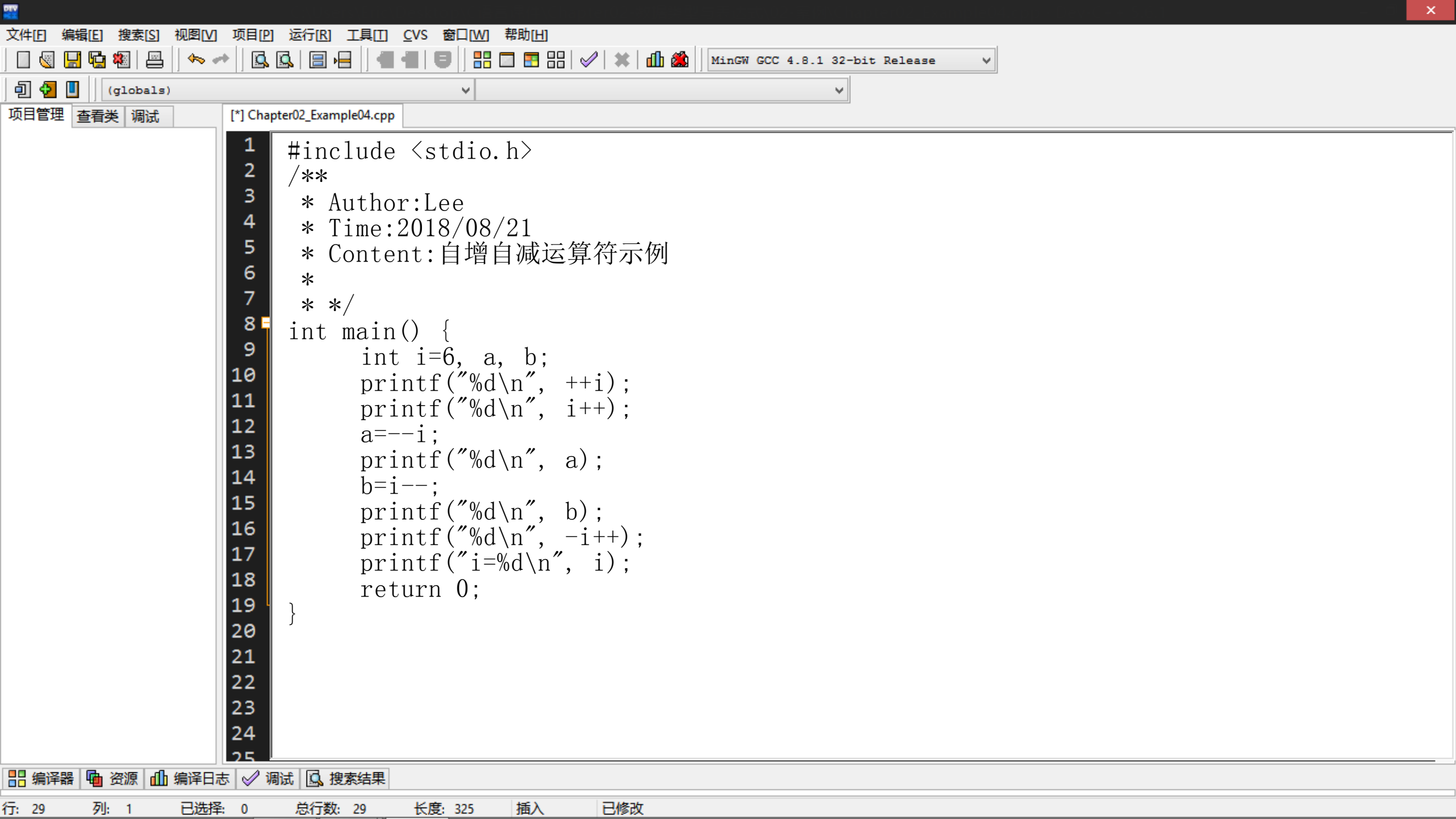
自增自减运算符 $--$, $++$

❖ 作用：使变量值加1或减1 优先级与 $-$ 相同，右结合

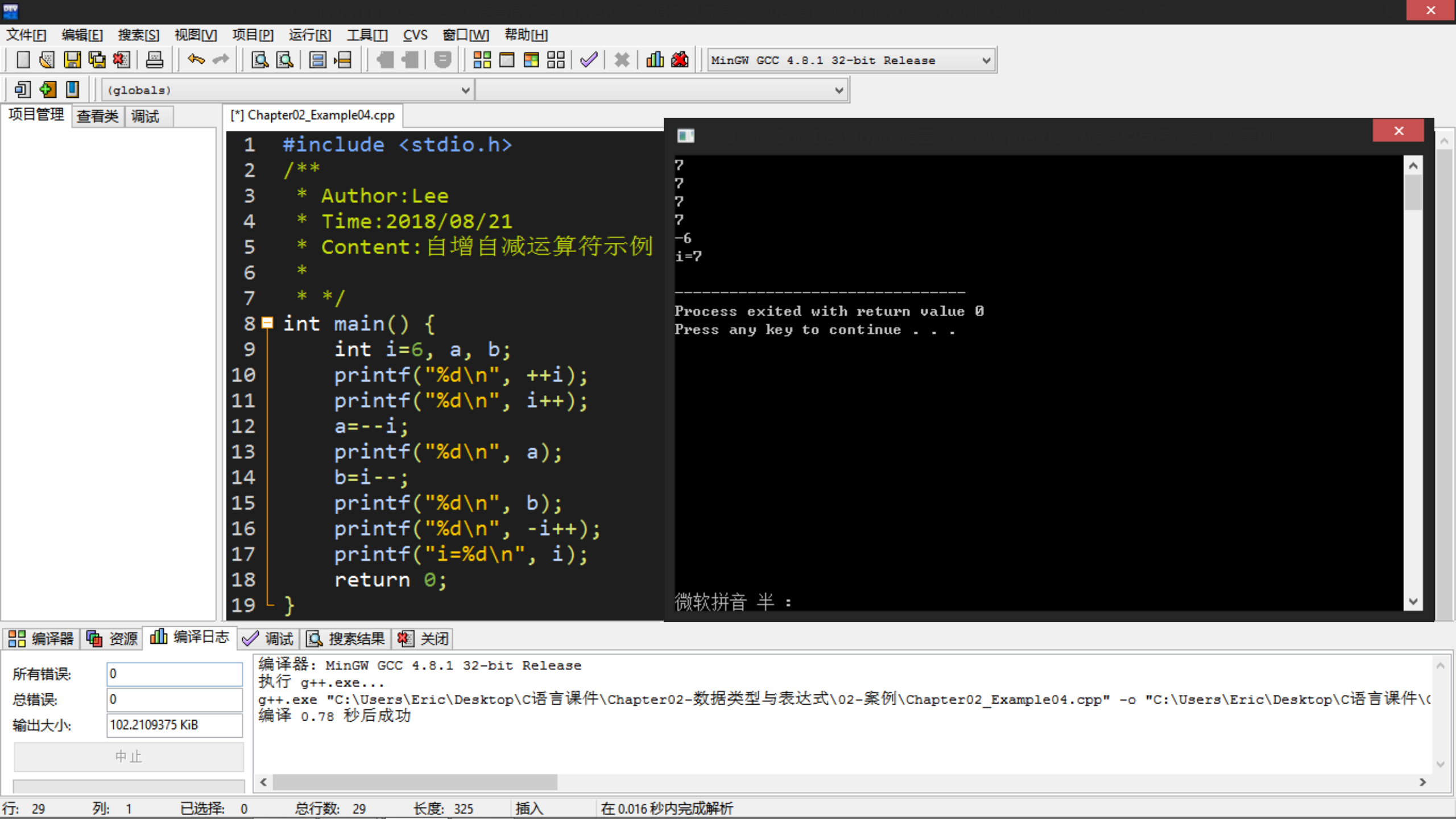
❖ 种类：

前置 $++i, --i$ (先执行 $i+1$ 或 $i-1$ ，再使用 i 值相当于 $i=i+1$)

后置 $i++, i--$ (先使用 i 值,再执行 $i+1$ 或 $i-1$)



```
#include <stdio.h>
/**
 * Author:Lee
 * Time:2018/08/21
 * Content:自增自减运算符示例
 *
 * */
int main() {
    int i=6, a, b;
    printf("%d\n", ++i);
    printf("%d\n", i++);
    a=--i;
    printf("%d\n", a);
    b=i--;
    printf("%d\n", b);
    printf("%d\n", -i++);
    printf("i=%d\n", i);
    return 0;
}
```



```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:自增自减运算符示例
6  *
7  * */
8 int main() {
9     int i=6, a, b;
10    printf("%d\n", ++i);
11    printf("%d\n", i++);
12    a--i;
13    printf("%d\n", a);
14    b=i--;
15    printf("%d\n", b);
16    printf("%d\n", -i++);
17    printf("i=%d\n", i);
18    return 0;
19 }
```

```
?
?
?
?
-6
i=7

-----
Process exited with return value 0
Press any key to continue . . .
```

编译器: MinGW GCC 4.8.1 32-bit Release
执行 g++.exe...
g++.exe "C:\Users\Eric\Desktop\C语言课件\Chapter02-数据类型与表达式\02-案例\Chapter02_Example04.cpp" -o "C:\Users\Eric\Desktop\C语言课件\
编译 0.78 秒后成功

所有错误:	0
总错误:	0
输出大小:	102.2109375 KiB

中止

行: 29 列: 1 已选择: 0 总行数: 29 长度: 325 插入 在 0.016 秒内完成解析

关系运算符

运算符	含义	优先级	结合性
<	小于	10	左结合
<=	小于或等于		
>	大于		
>=	大于或等于		
==	等于	9	
!=	不等于		



关系运算和关系表达式

❖ 关系表达式：关系运算表示一个条件，根据其结果判断条件是否满足关系表达式

❖ 关系表达式形式

<表达式1> <关系运算符> <表达式2>

P.S. 表达式1和表达式2可以是任何表达式

❖ 举例： $a > 3$

如果 $a=8$ ，结果为“真”，即条件满足

如果 $a=1$ ，结果为“假”，即条件不满足

逻辑运算符

□ 种类：**!** (逻辑非) , **&&** (逻辑与) , **||** (逻辑或)

□ 逻辑运算**真值表**

a	b	!a	!b	a&& b	a b
真	真	假	假	真	真
真	假	假	真	假	真
假	真	真	假	假	真
假	假	真	真	假	假

□ C语言中, 运算量: 0表示**“假”**, 非0表示**“真”**。

运算结果: 0表示**“假”**, 1表示**“真”**。



逻辑运算符的短路特性

❖ 逻辑表达式求解时，并非所有的逻辑运算符都被执行，只是在必须执行下一个逻辑运算符才能求出表达式的解时，才执行该运算符

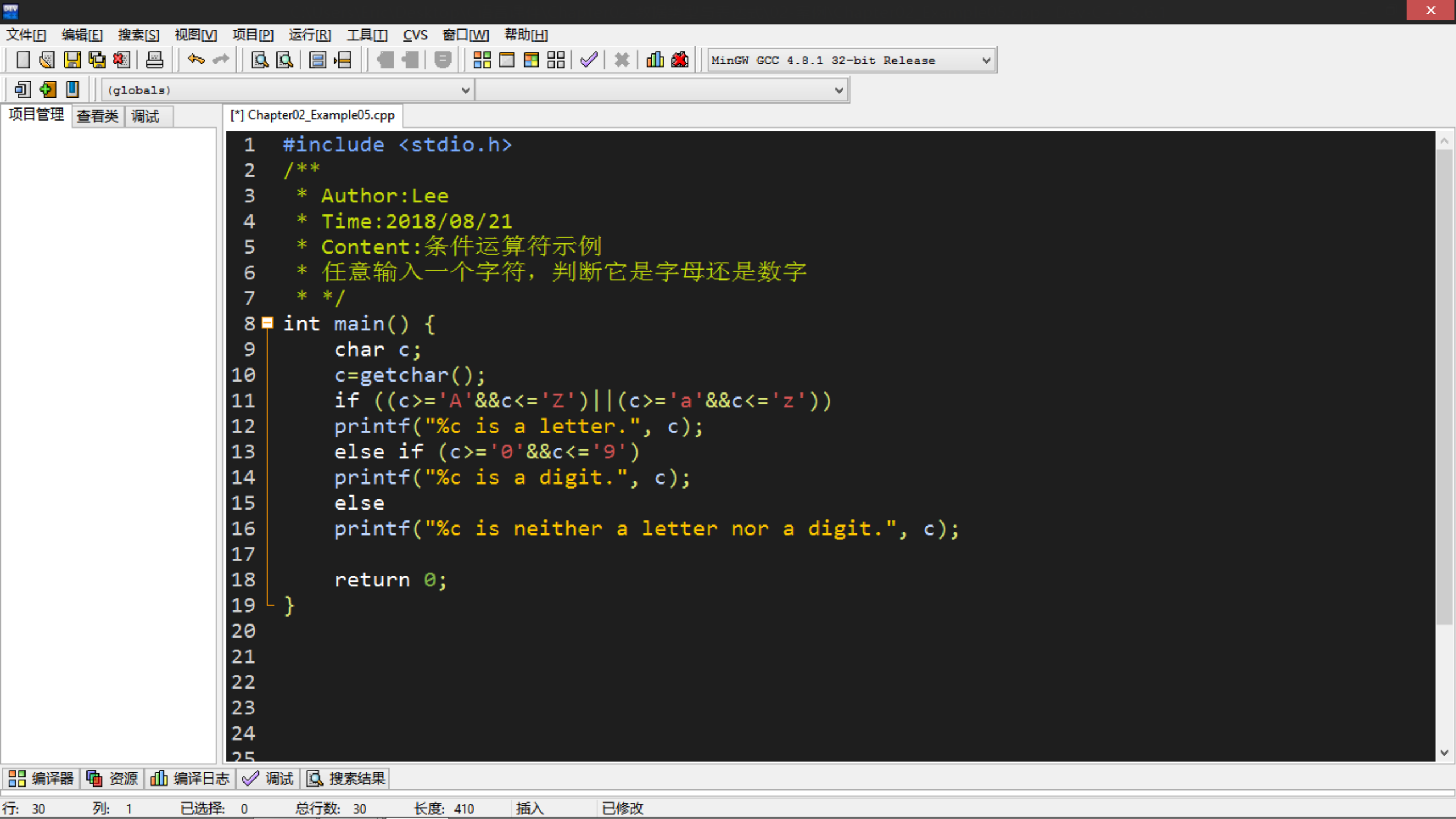
❖ 例如：

1. 表达式1 && 表达式2

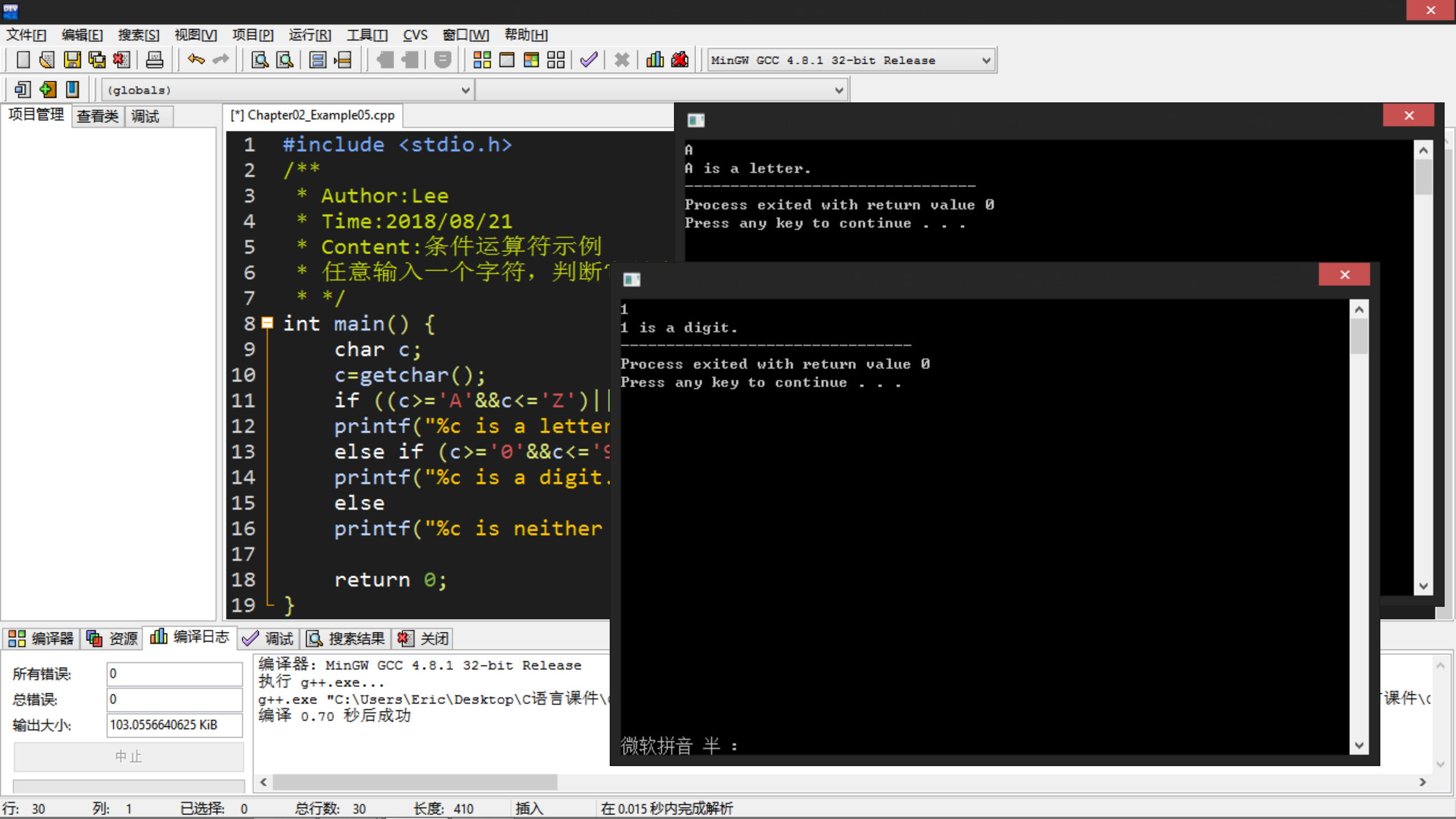
若表达式1为“假”，那么表达式2不执行

2. 表达式1 || 表达式2

若表达式1为“真”，那么表达式2不执行



```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:条件运算符示例
6  * 任意输入一个字符，判断它是字母还是数字
7  * */
8 int main() {
9     char c;
10    c=getchar();
11    if ((c>='A'&& c<='Z') || (c>='a'&& c<='z'))
12        printf("%c is a letter.", c);
13    else if (c>='0'&& c<='9')
14        printf("%c is a digit.", c);
15    else
16        printf("%c is neither a letter nor a digit.", c);
17
18    return 0;
19 }
```



[*] Chapter02_Example05.cpp

```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/21
5  * Content:条件运算符示例
6  * 任意输入一个字符,判断
7  * */
8 int main() {
9     char c;
10    c=getchar();
11    if ((c>='A'&&c<='Z')||
12        printf("%c is a letter\n",c);
13    else if (c>='0'&&c<='9')
14        printf("%c is a digit.\n",c);
15    else
16        printf("%c is neither\n",c);
17
18    return 0;
19 }
```

```
A
A is a letter.
-----
Process exited with return value 0
Press any key to continue . . .
```

```
1
1 is a digit.
-----
Process exited with return value 0
Press any key to continue . . .
```

编译器 资源 编译日志 调试 搜索结果 关闭

所有错误: 0
总错误: 0
输出大小: 103.0556640625 KiB

编译器: MinGW GCC 4.8.1 32-bit Release
执行 g++.exe...
g++.exe "C:\Users\Eric\Desktop\C语言课件\
编译 0.70 秒后成功

中止

赋值运算符和赋值表达式

□ 赋值运算符

赋值符号“=”就是赋值运算符，它的作用是将一个数据赋给一个变量。如“a = 3”的作用是执行一次赋值操作（或称赋值运算）。把常量3赋给变量a。也可以将一个表达式的值赋给一个变量。

□ 类型转换

如果赋值运算符两侧的类型不一致，在赋值时要进行类型转换，将右边的类型转换为左边类型。

运算结果的类型，即左边变量的数据类型。

赋值运算注意事项

- I. 将浮点型数据（包括单、双精度）赋给整型变量时，**舍弃**浮点数的小数部分。
- II. 将整型数据赋给单、双精度变量时，数值不变，但**以浮点数形式**存储到变量中
- III. 字符型数据赋给整型变量时，由于字符只占1个字节，而整型变量为4个字节，因此将字符数据（8个二进制位）放到整型变量存储单元的**低8位**中。
- IV. 将一个int、short、long型数据赋给一个char型变量时，只将其**低8位**原封不动地送到char型变量（即**截断**）。
例如：`int i=289;char c=' a' ;c=i;`
- V. 有符号整数和无符号整数之间相互赋值时，是**按存储单元中的存储形式直接传送**。

复合的赋值运算

在赋值符 “=” 之前加上其他运算符，可以构成复合的赋值运算符。

例如：

$a += 3$ 等价于 $a = a + 3$

$x *= y + 8$ 等价于 $x = x * (y + 8)$

$x \% = 3$ 等价于 $x = x \% 3$



逗号运算符

□ 逗号运算符:将两个表达式连接起来,又称为“**顺序求值运算符**”

如: $3 + 5, 6 + 8$

□ **一般形式**: 表达式1, 表达式2, …… , 表达式 n

□ 求解过程:

先求解表达式1, 再求解表达式2 ……

整个逗号表达式的值是表达式 n 的值。

□ 整个逗号表达式的值是**最后一个表达式的值**

条件运算符

□ 条件运算符形式：

表达式1 ? 表达式2 : 表达式3 ;

如：

$2 > 1 ? 2 : 3$

若表达式1为真，那么执行表达式2，否则执行表达式3

□ 条件运算符结合性为右结合

□ 条件运算符为三目运算符





练习巩固

- Problem description
编写一个判断是否为闰年的程序
- Input
输入占一行。输入一个整数 y ，代表年份
- Output
输出占一行。“yes” 或者 “no”


```

1  #include <stdio.h>
2  /**
3   * Author:Lee
4   * Time:2018/08/21
5   * Content:运算符和表达式练习巩固
6   * 判断闰年
7   * */
8  int main() {
9      int year;
10     scanf("%d",&year);
11     if((year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0)){
12         printf("yes\n");
13     }else{
14         printf("no\n");
15     }
16
17     return 0;
18 }
19

```

微软拼

微软拼音 半 :

所有错误: 0
 总错误: 0
 输出大小: 102.380859375 KiB

中止

编译器: MinGW GCC 4.8.1 32-bit Release

执行 g++.exe...

g++.exe "C:\Users\Eric\Desktop\C语言课件\Chapter02-数据类型与表达式\02-案例\Chapter02_homework02.cpp" -o "C:\Users\Eric\Desktop\C语言课件\

编译 0.59 秒后成功

章节复习

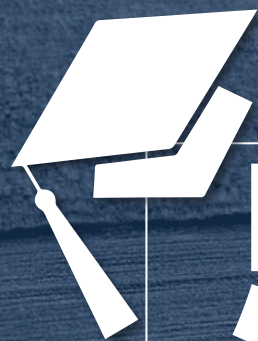
引论

常量

变量

运算符和表达式

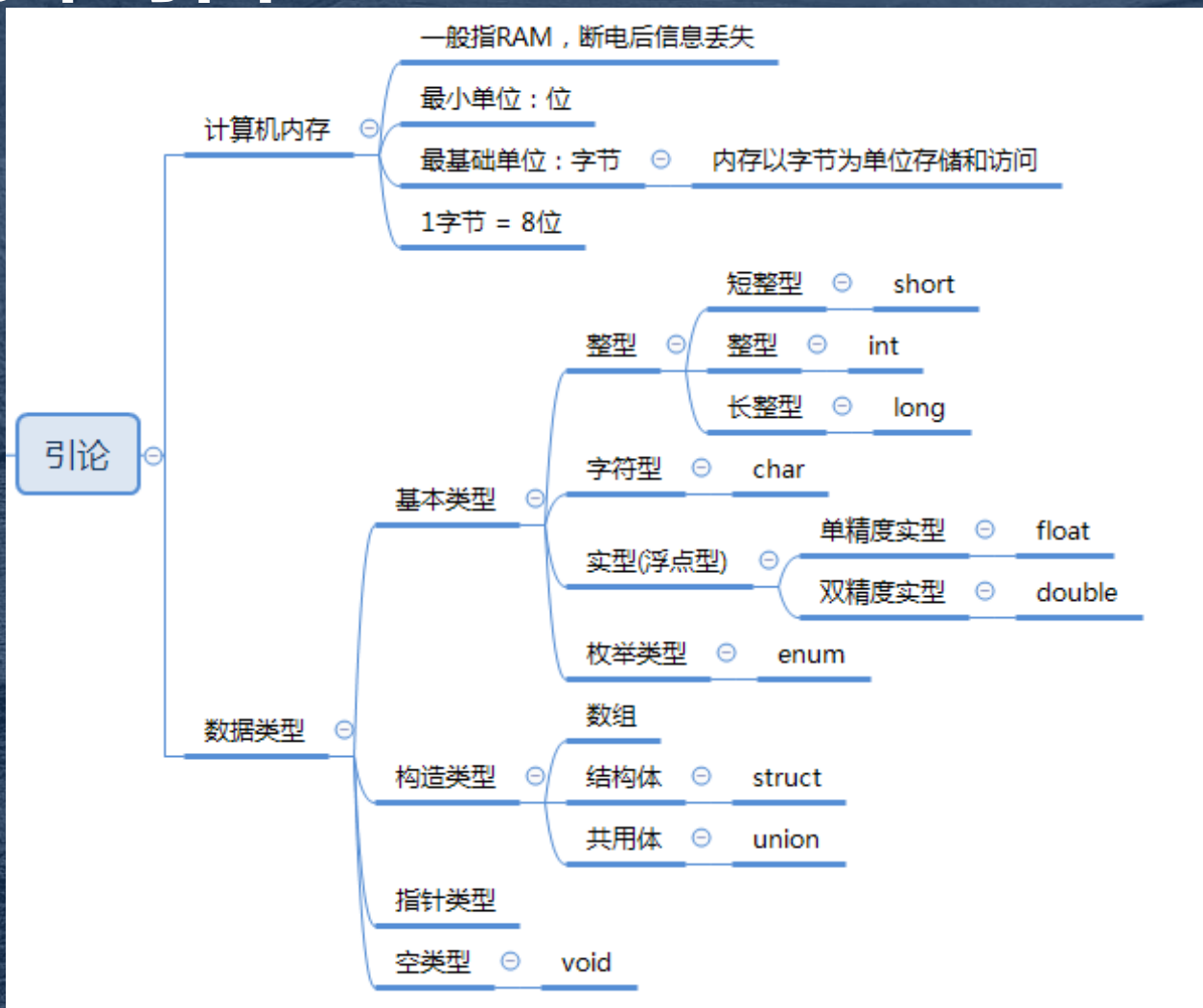
思维导图



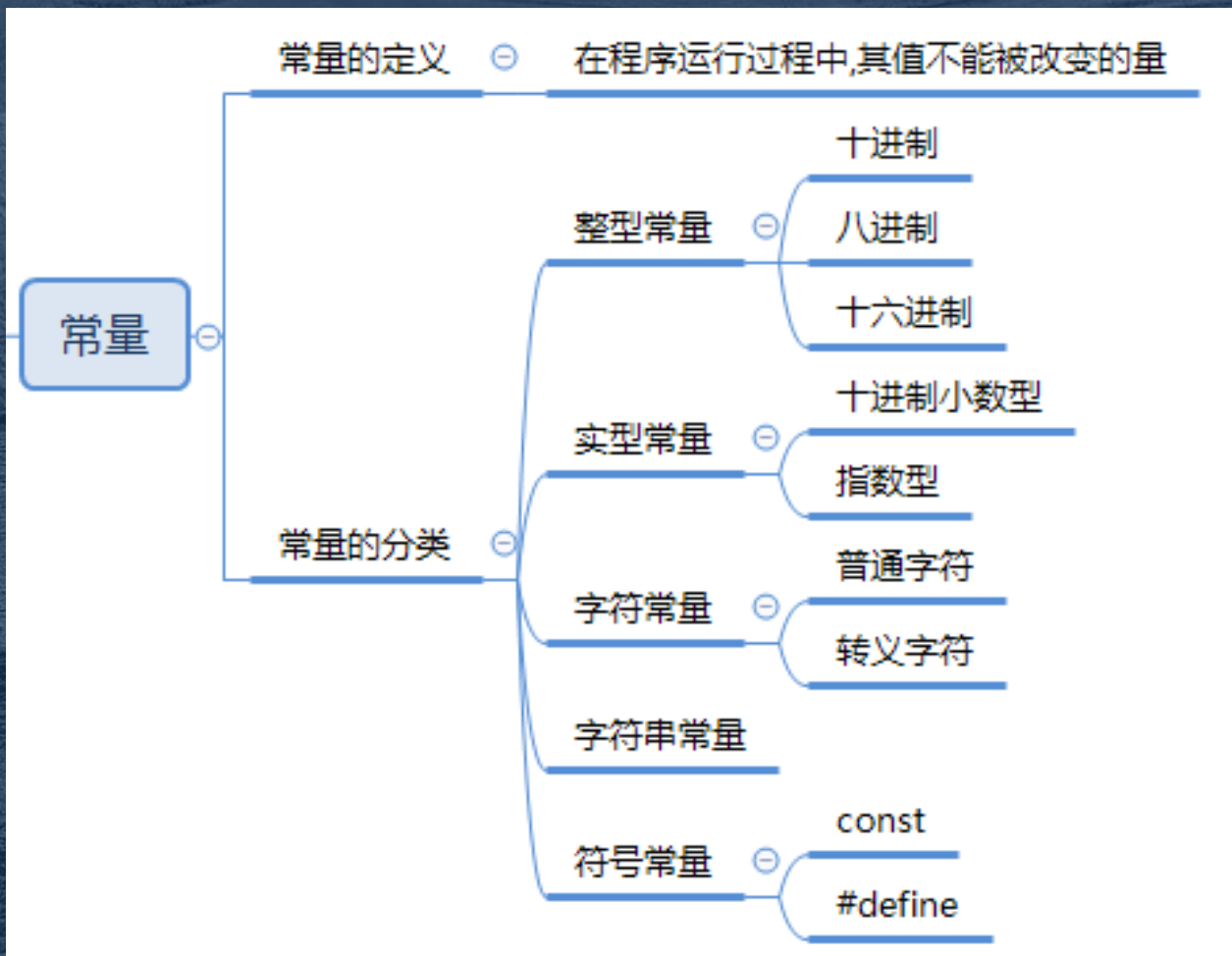
思维导图



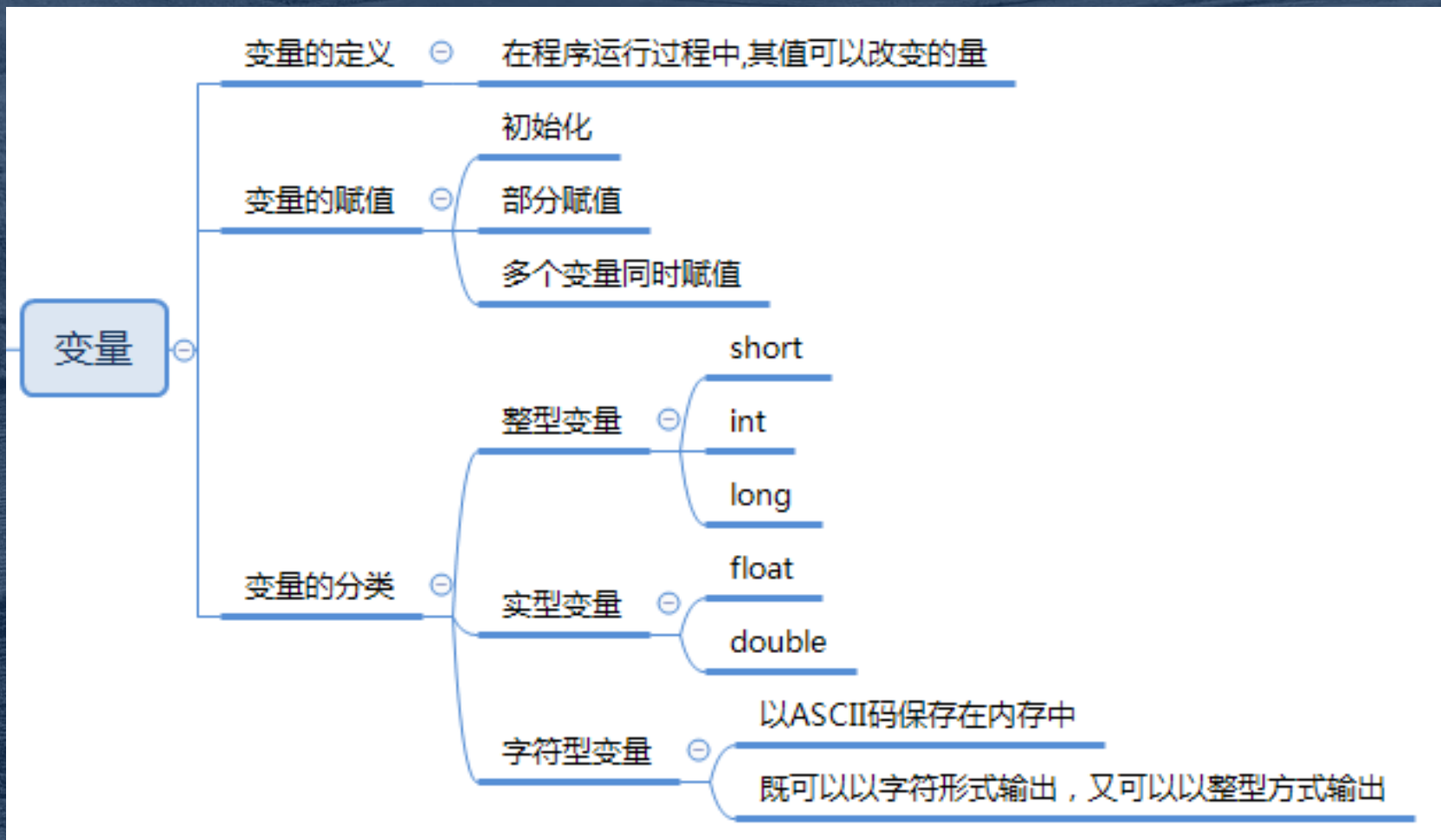
引论部分思维导图



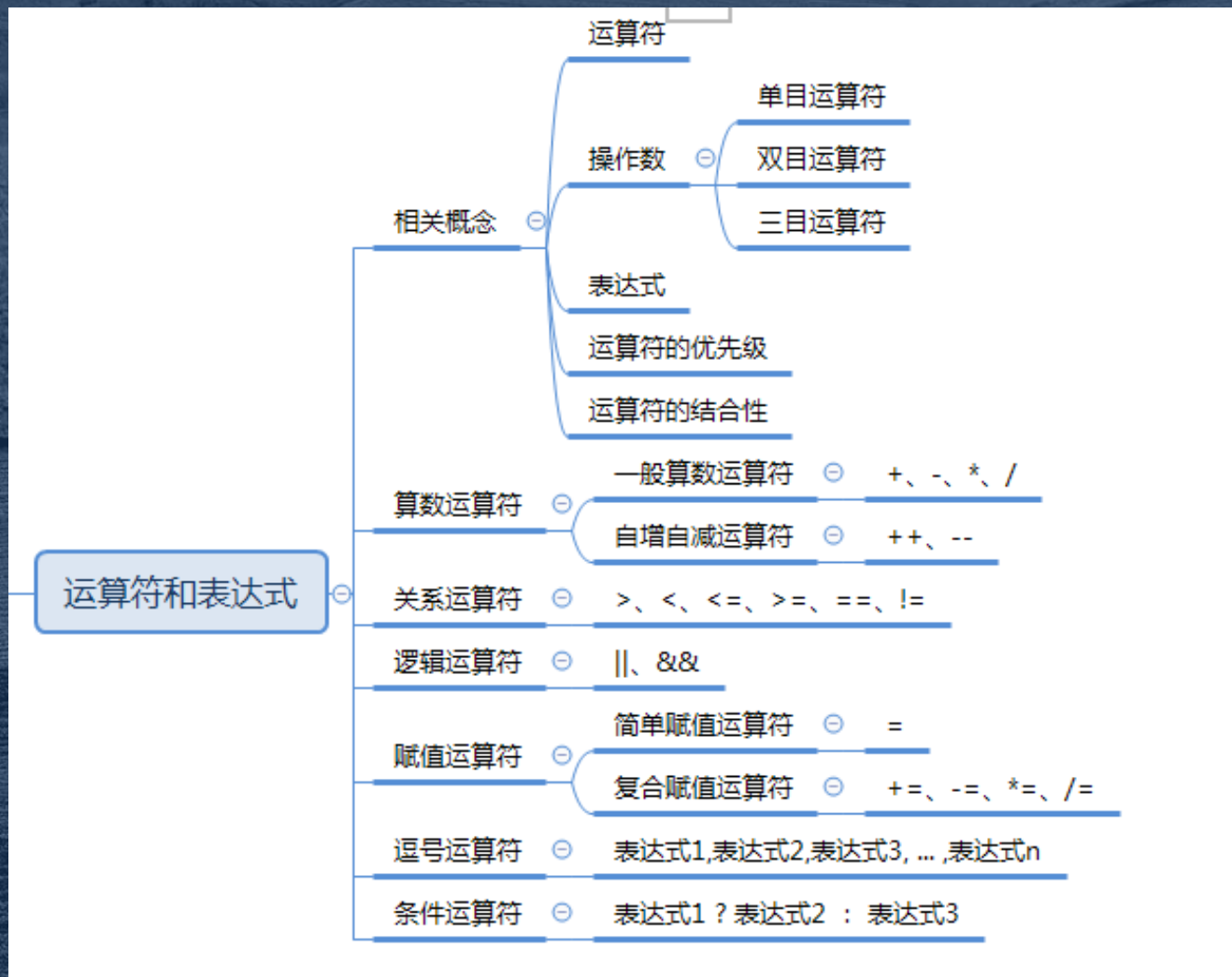
常量部分思维导图



变量部分思维导图



运算符和表达式部分思维导图



教学评价

完成本章的作业和练习题后，请在QQ群中根据自己对本章知识的掌握情况，进行投票，我将根据同学们的投票，及时了解同学们对本章的掌握情况，并做出教学调整。

< 返回

投票

发布投票

进行中



★李伦彬135552966

24秒前

完成本章作业后，认真复习和总结，你掌握本章知识了吗？

选项预览

共0票

- A. 已掌握，可独立完成作业。
- B. 基本掌握，参考课件和微视频，可完成作业...
- C. 没有掌握。

立即投票



练习题库-笔试

FSCapture

练习题库-上机

Dev-Cpp编程工具

自测系统

ZoomIt

Xmind-8



课件下载网址：

<http://www.jsjx.com/index.aspx>

[推荐资料1 - C语言经典编程282例](#)

[推荐资料2 - 华为C语言编程规范](#)

[推荐资料3 - 嗨翻C](#)

[推荐资料4 - C&C++ API.chm](#)



程序社团-技术交流群 : 540653133



Programming Association
540653133

发消息

首页 成员 设置 [编辑资料](#)



群介绍 本群创建于2017/10/17: 群主很懒,什么都没有留下

群标签 [行业交流](#) [IT/互联网](#) [程序设计](#) [IT](#) [计算机](#)

群主/管理员



成员分布 (136/200)

66%	13%	84%	16%
男-91人	黑河-19人	90后-115人	单身-23人

本群星级  该群一切正常

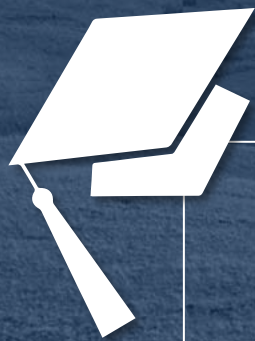


课件下载网址：

<http://jsjxy.hhhxy.cn/info/1074/2602.htm>

课件下载二维码：





感谢聆听

下一章内容：程序控制流程

主讲人：李伦彬

请同学们提前预习课本相关知识点

