



《高级语言程序设计》

VS2022调试工具的使用

学号：2353814

班级：信05

姓名：马小龙

日期：2024年6月12日

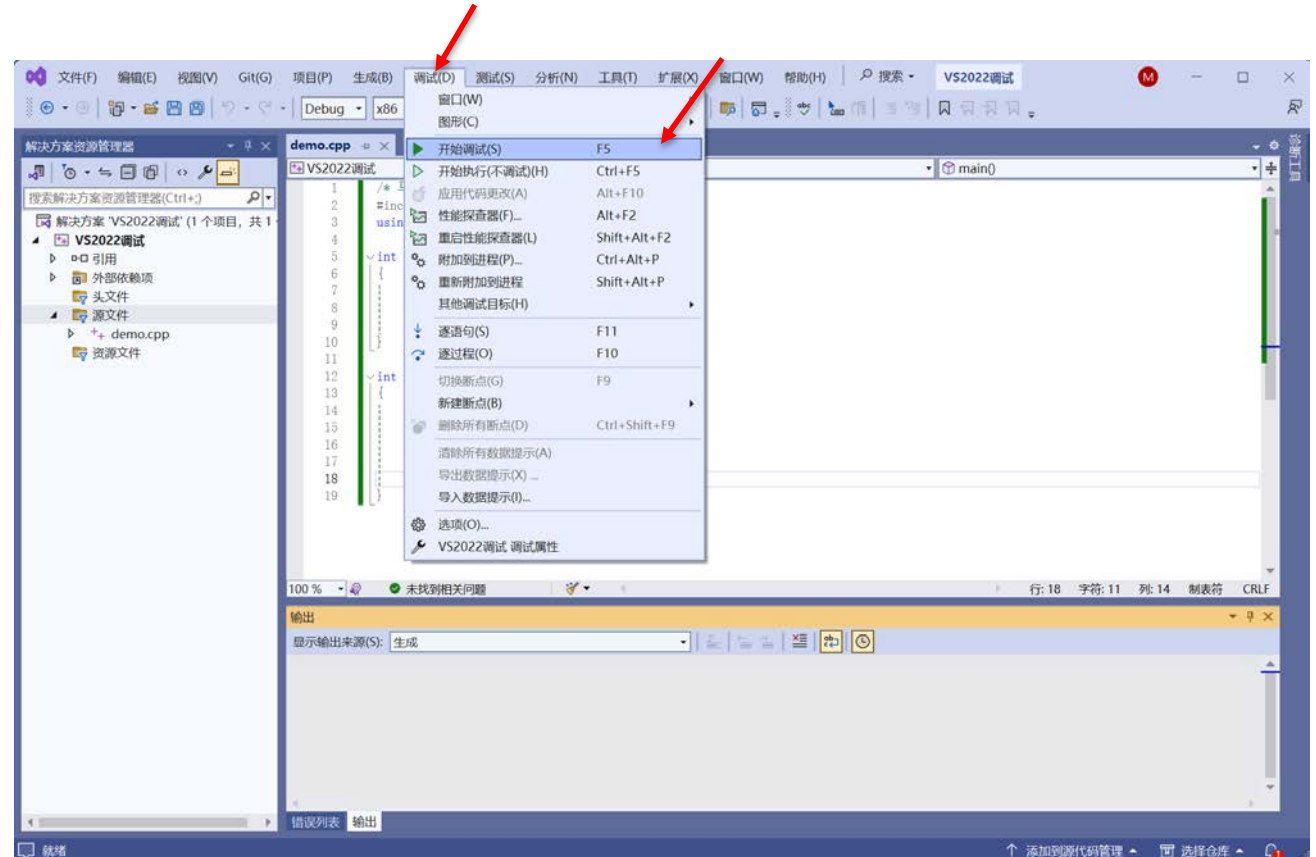


一、基本使用方法

1.开始和调试结束

开始调试:

方法一: 选择菜单-“调试”-“开始调试”



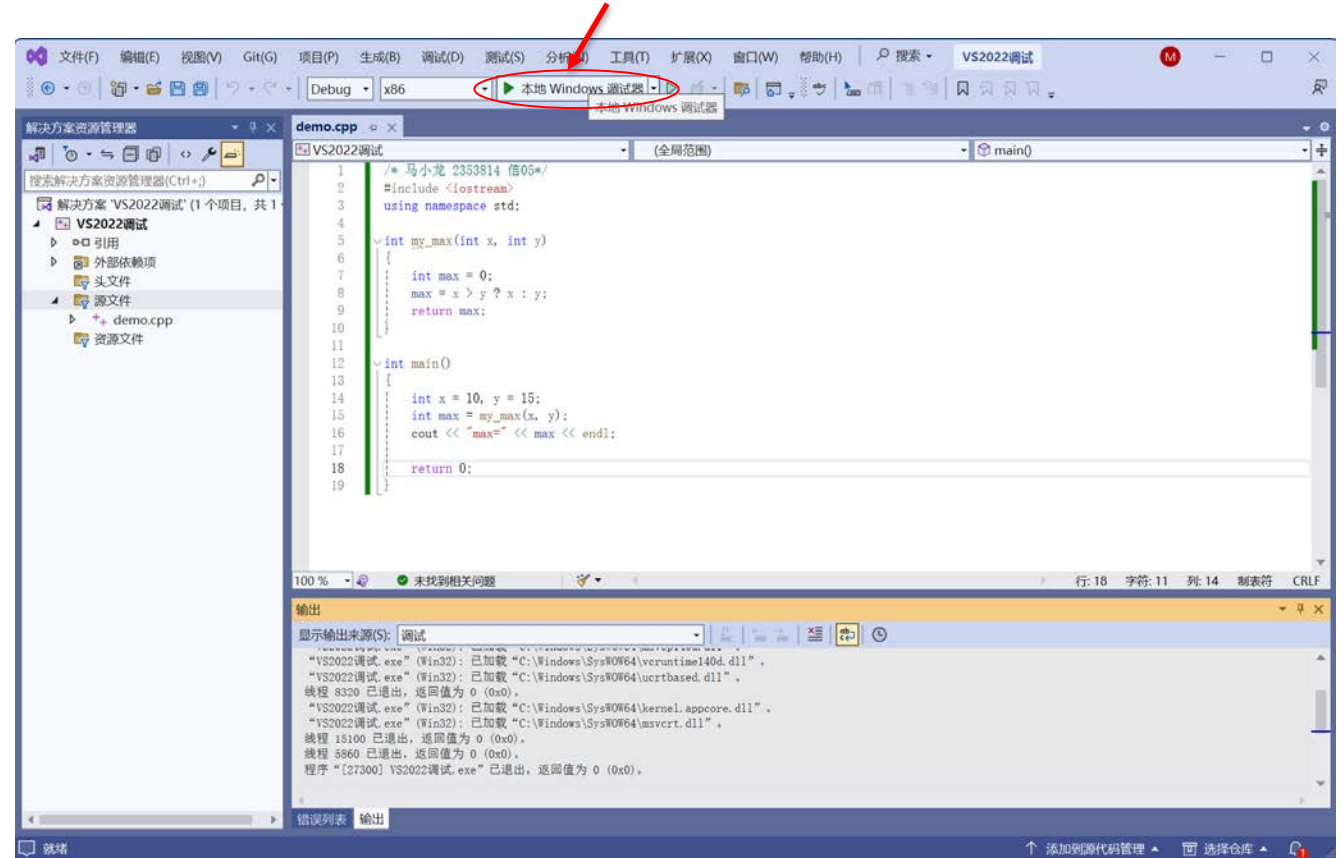
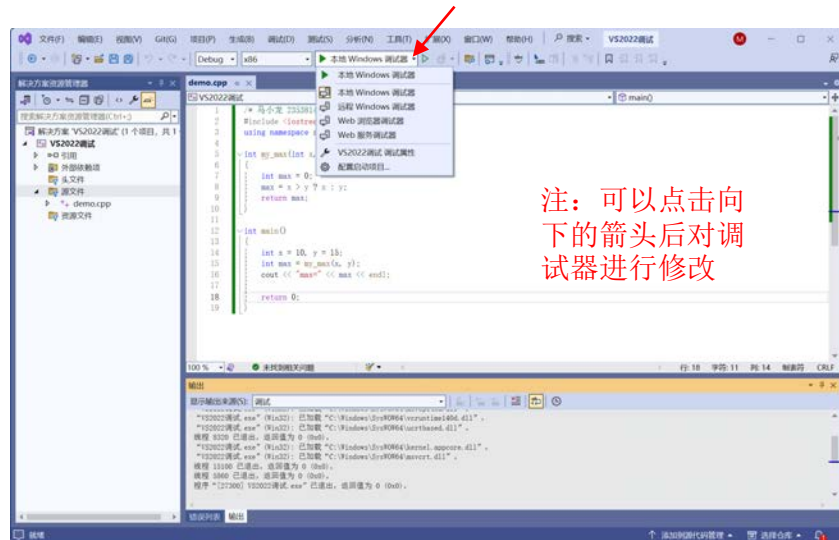


一、基本使用方法

1.开始和调试结束

开始调试:

方法二: 点击深绿色箭头或快捷键F5



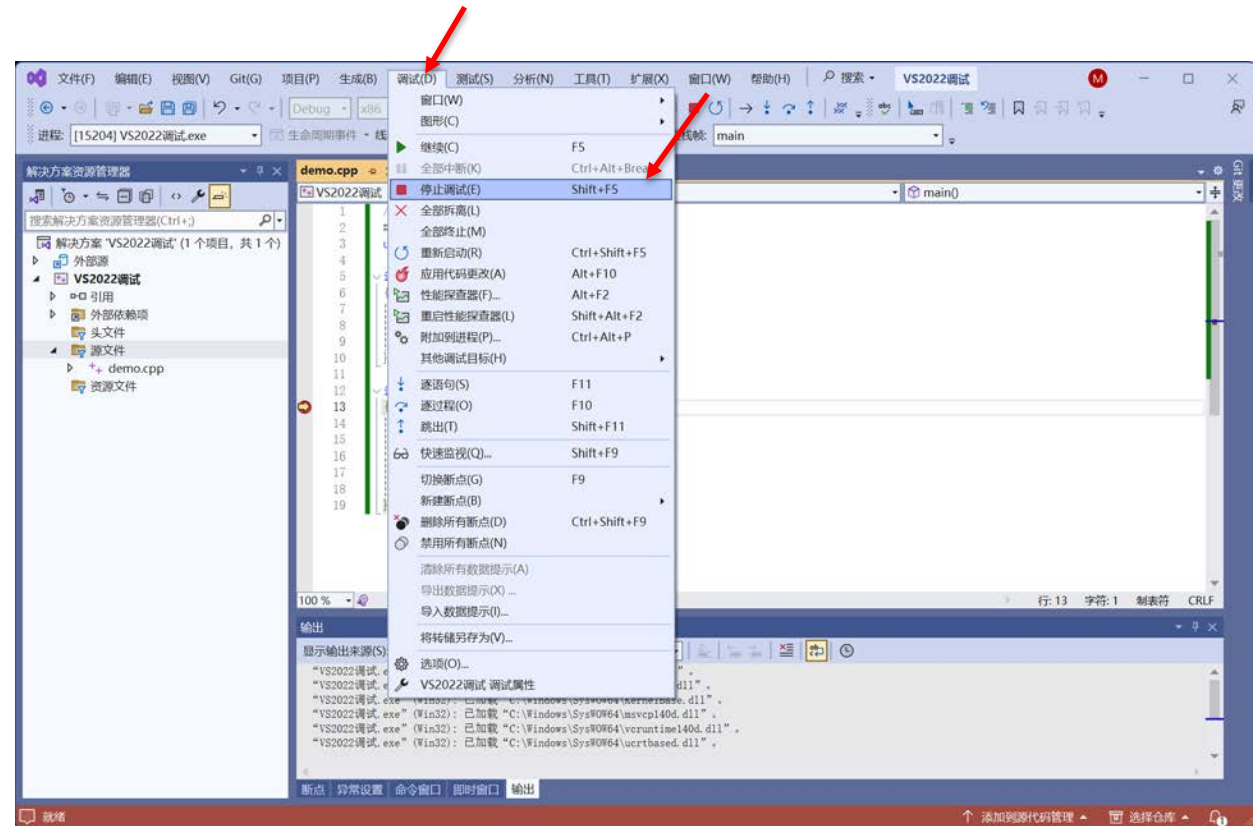


一、基本使用方法

1.开始和结束结束(1.1)

结束调试:

方法一: 选择菜单- “调试” - “停止调试”



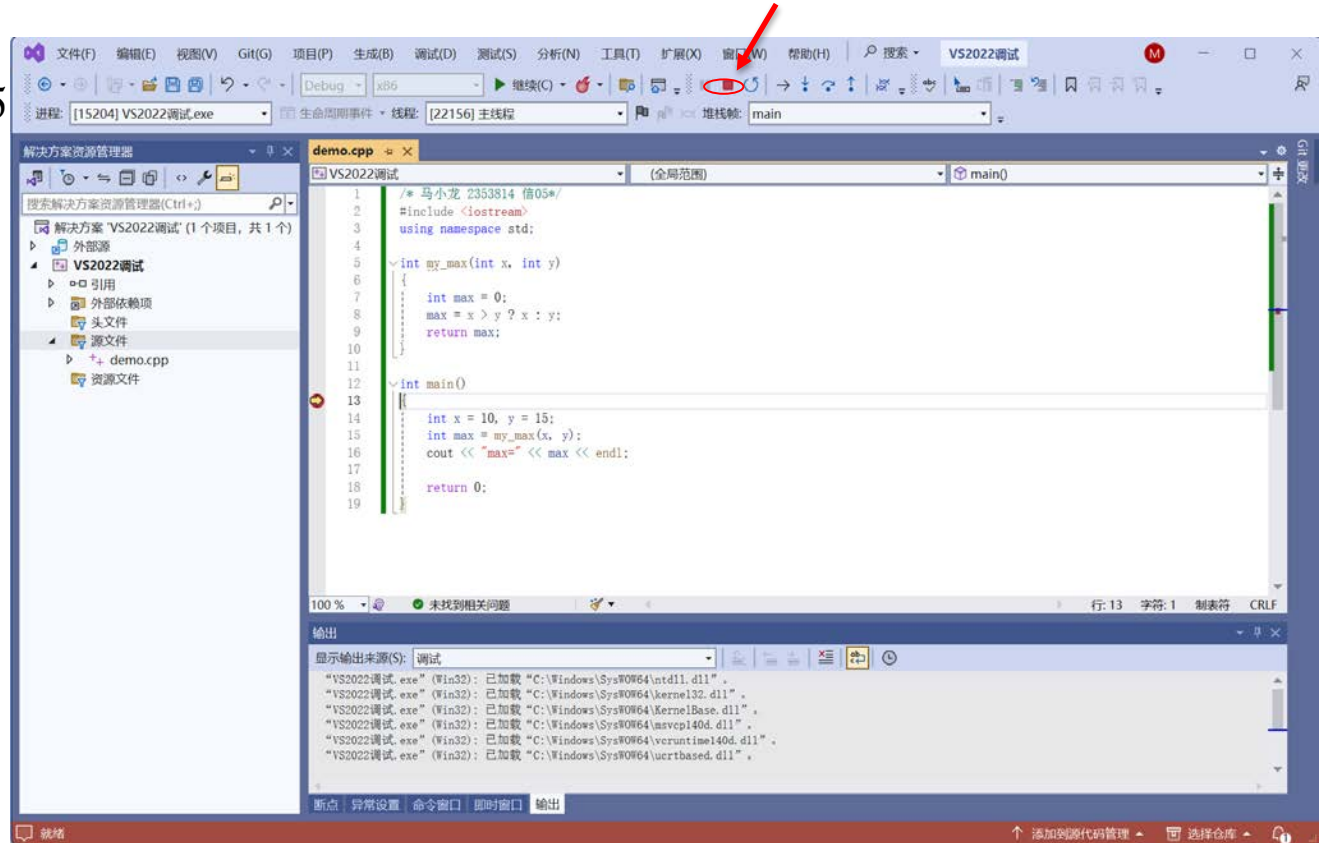


一、基本使用方法

1.开始和结束调试(1.1)

结束调试：

方法二：点击红色暂停键或快捷键Shift+F5





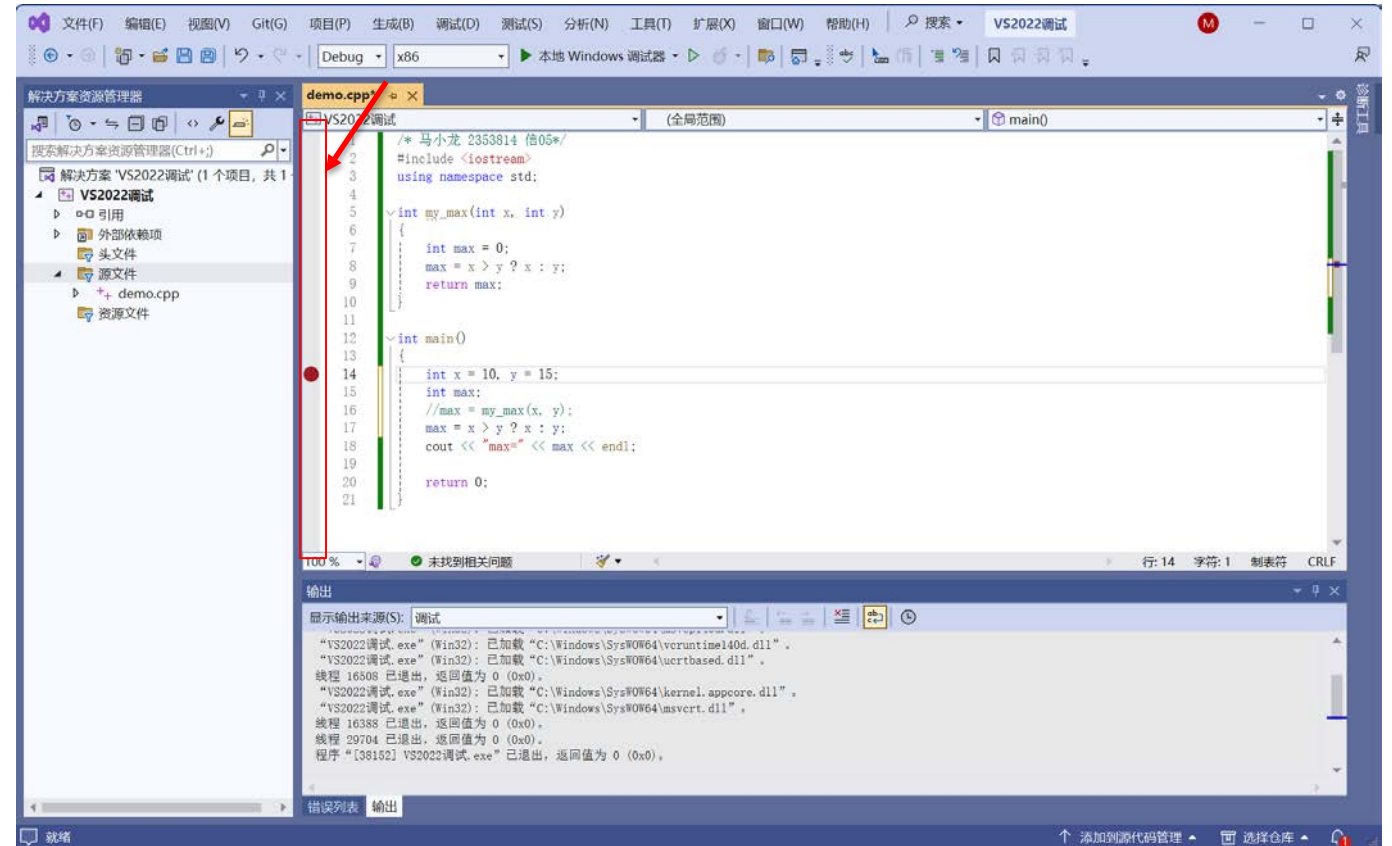
一、基本使用方法

2.单步执行(1.2)

第一步：设置断点

再源码左侧一栏中左键单击即可设置断点，再次单击取消断点

断点的作用：在默认的情况下，开始调试后程序会连续执行到结束或者是等待输出；通过添加断点可以使程序在有断点的语句前停止。

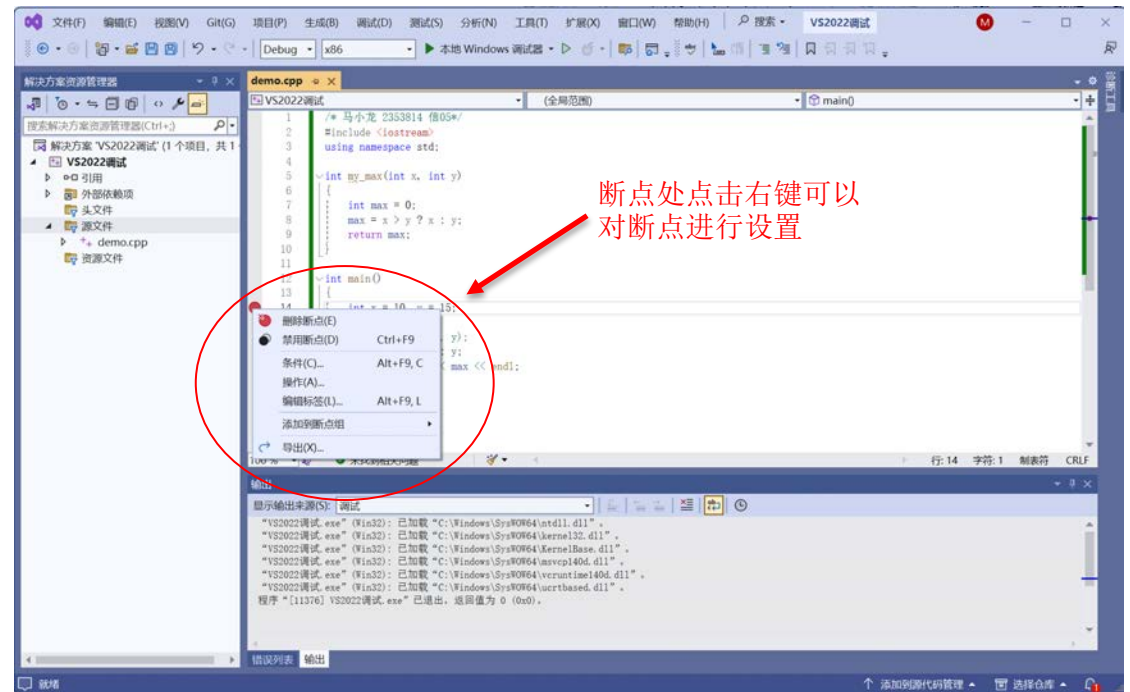
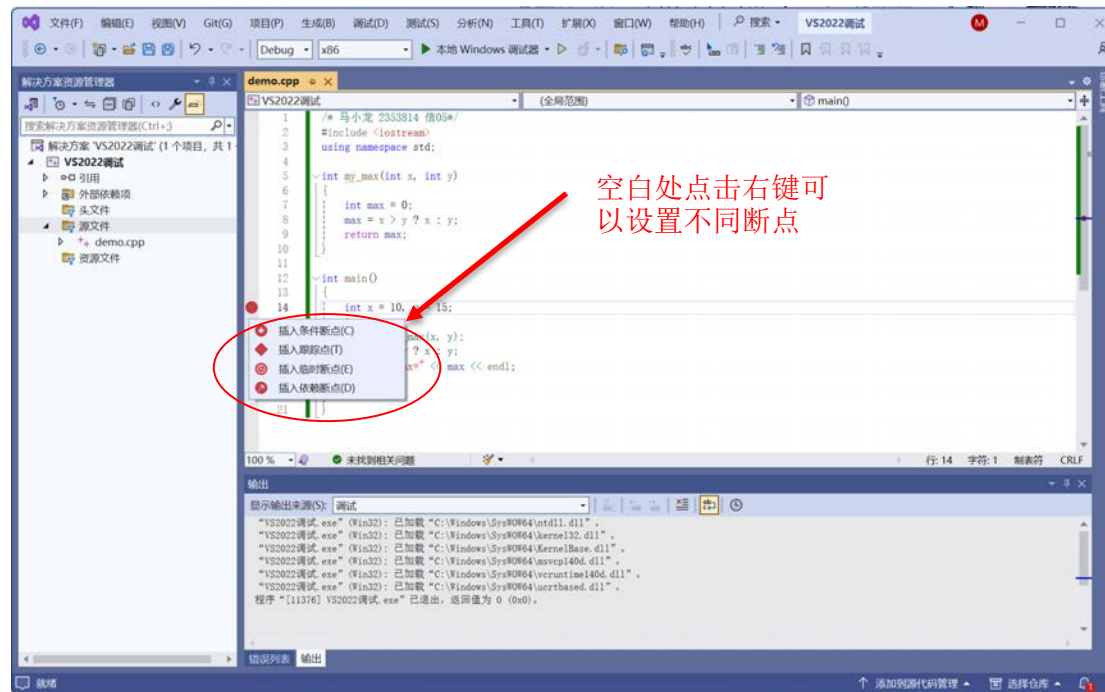




一、基本使用方法

2.单步执行(1.2)

第一步：设置断点



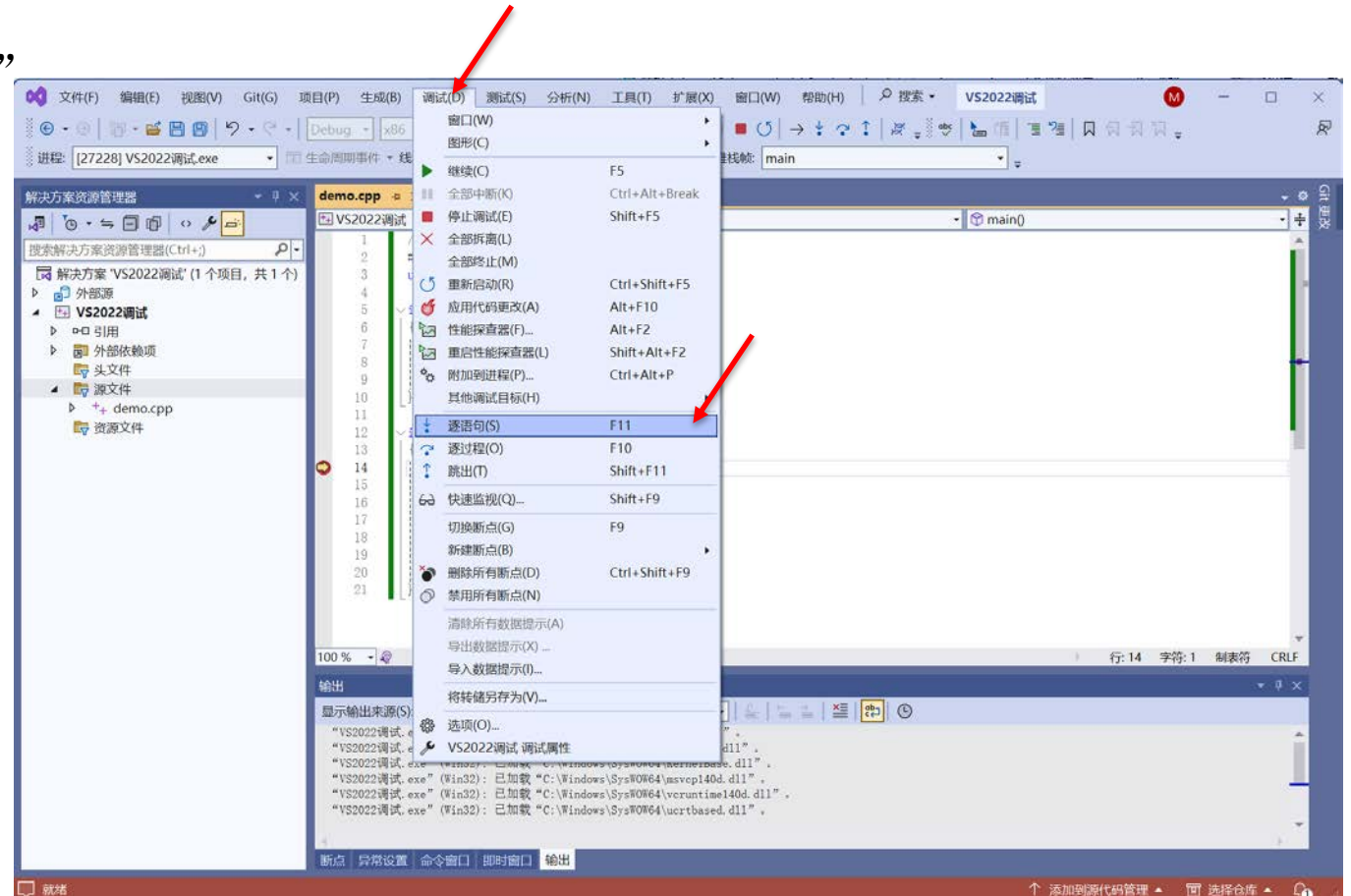


一、基本使用方法

2.单步执行(1.2)

第二步:

方法一: 开始调试后, 选择菜单- “调试”
- “逐语句” 进行单步操作



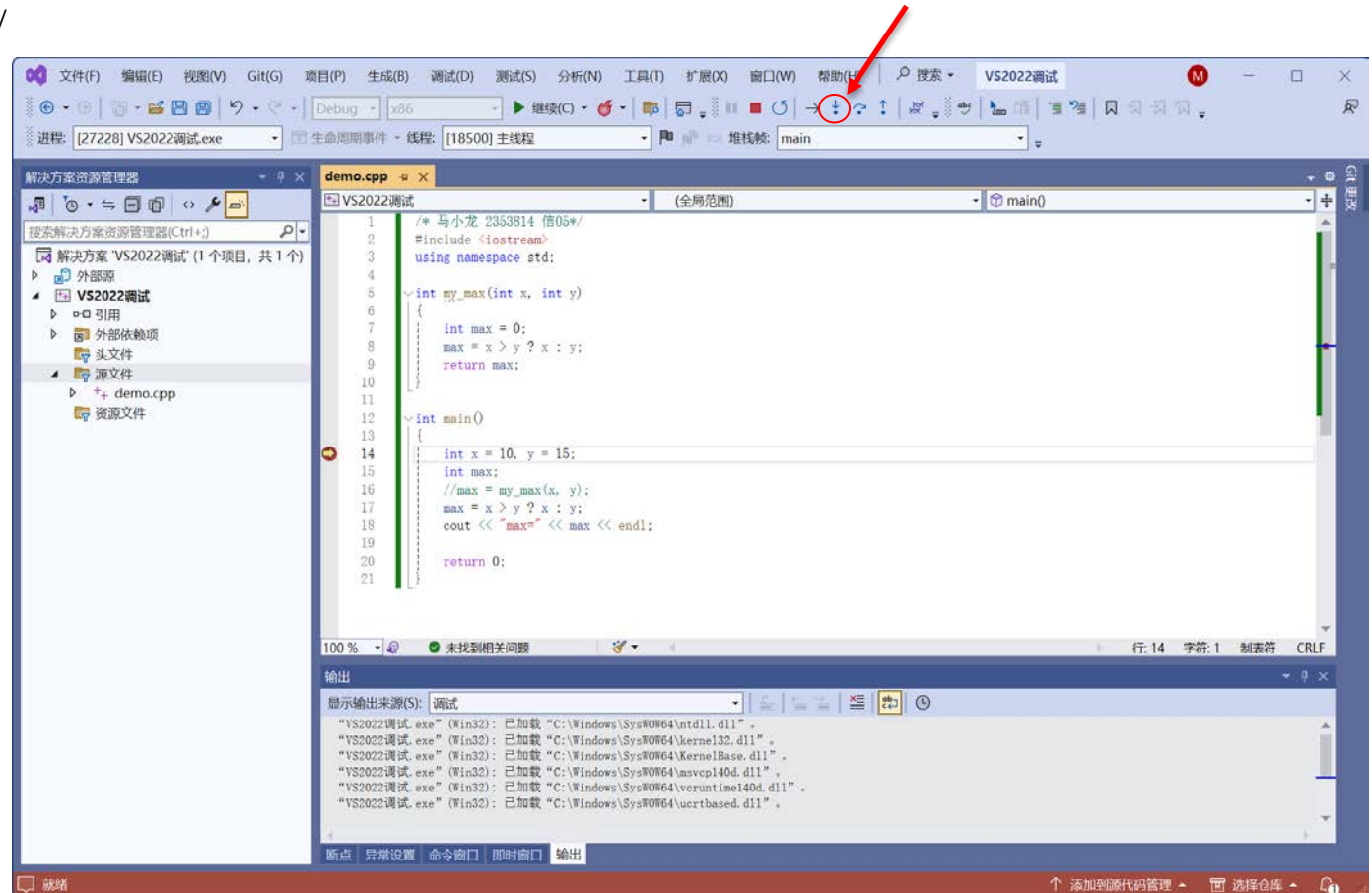


一、基本使用方法

2.单步执行(1.2)

第二步：

方法二：开始调试后，单击此按钮/
按F11进行单步操作

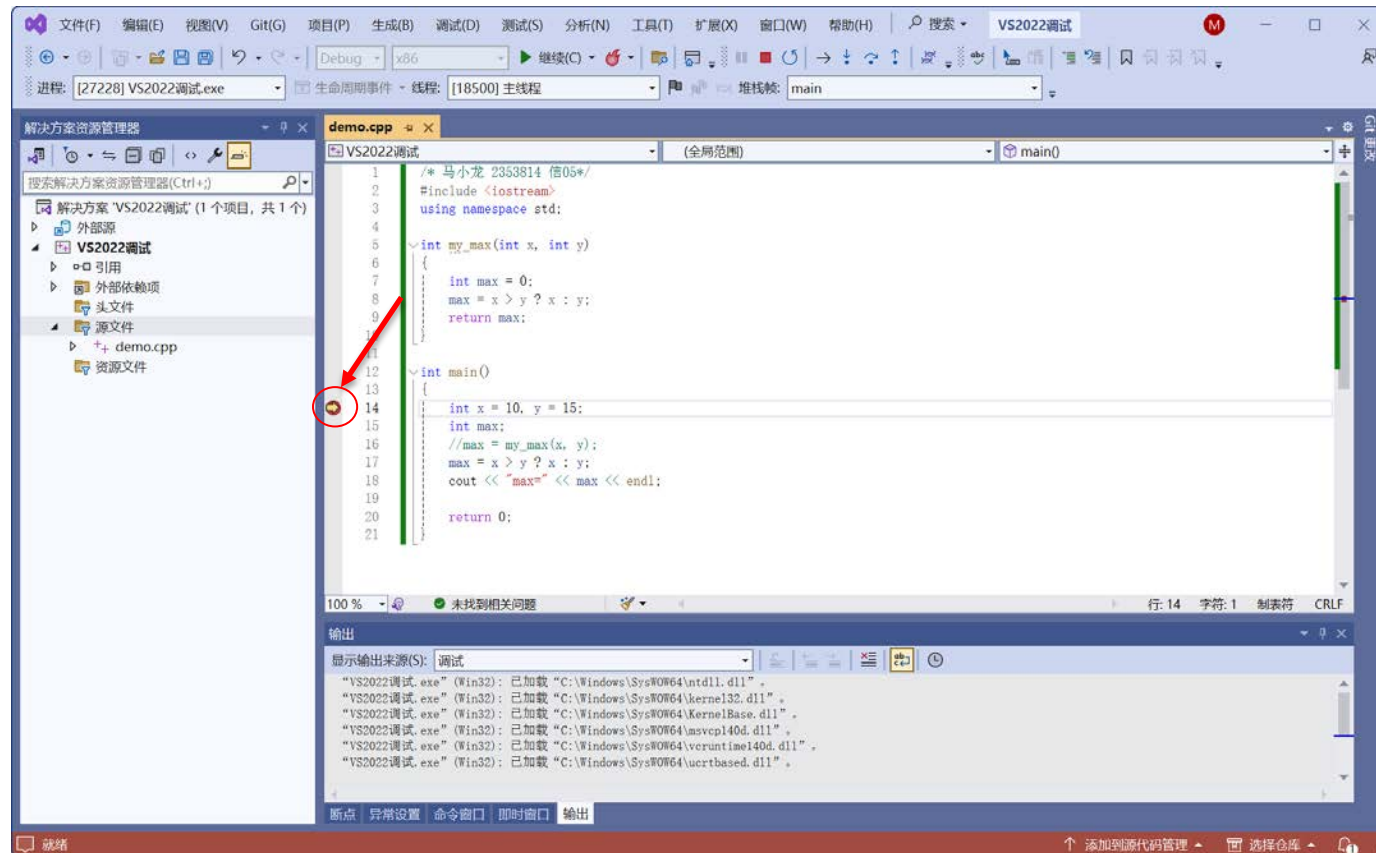




一、基本使用方法

2.单步执行(1.2)

左侧箭头所指，即为下一步需要执行的语句

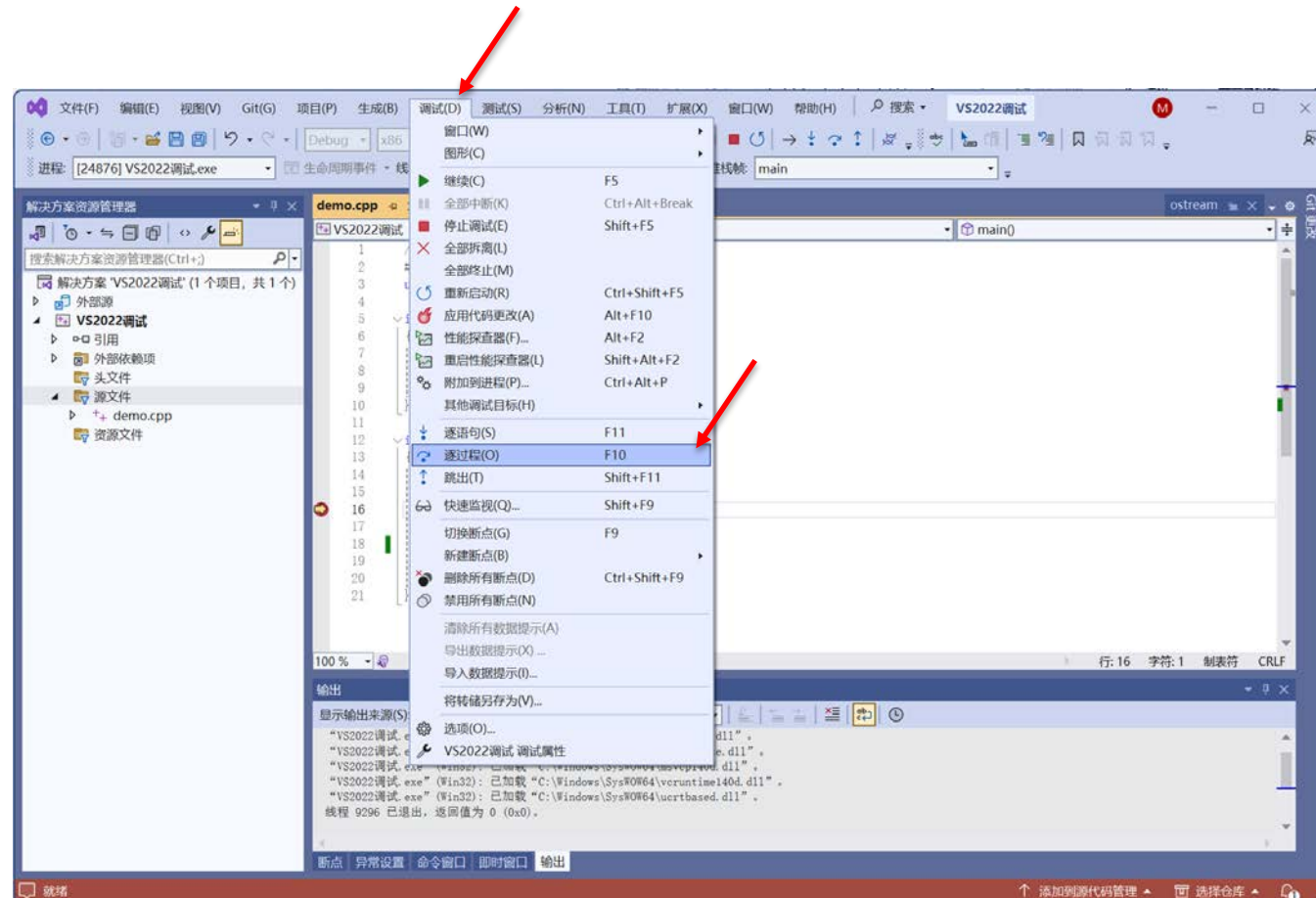




一、基本使用方法

3.一步完成系统类/系统函数或自定义函数(1.3、1.5)

方法一：选择菜单-“调试”-“逐过程”

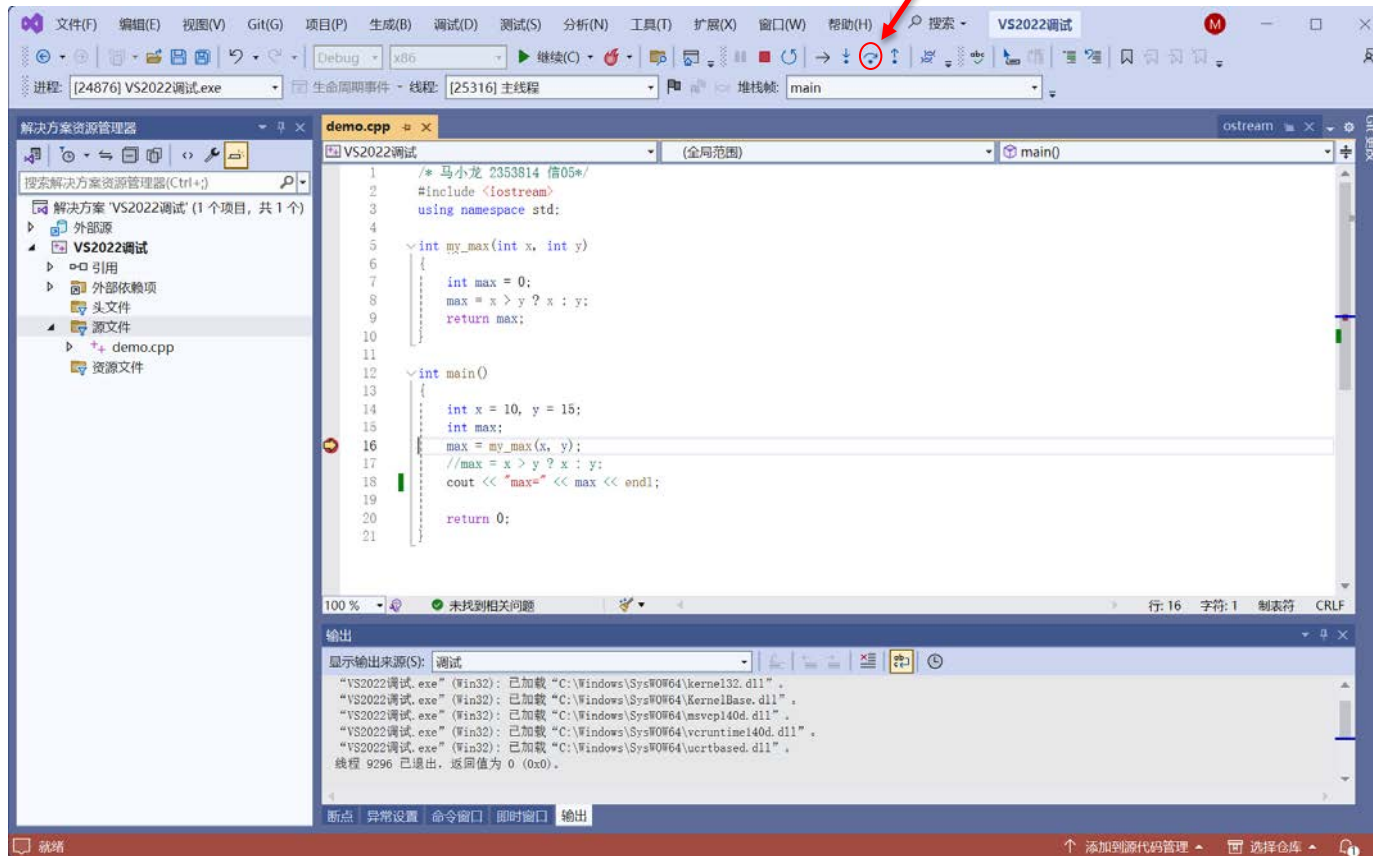




一、基本使用方法

3.一步完成系统类/系统函数或自定义函数(1.3、1.5)

方法二：单击此按钮/按F10

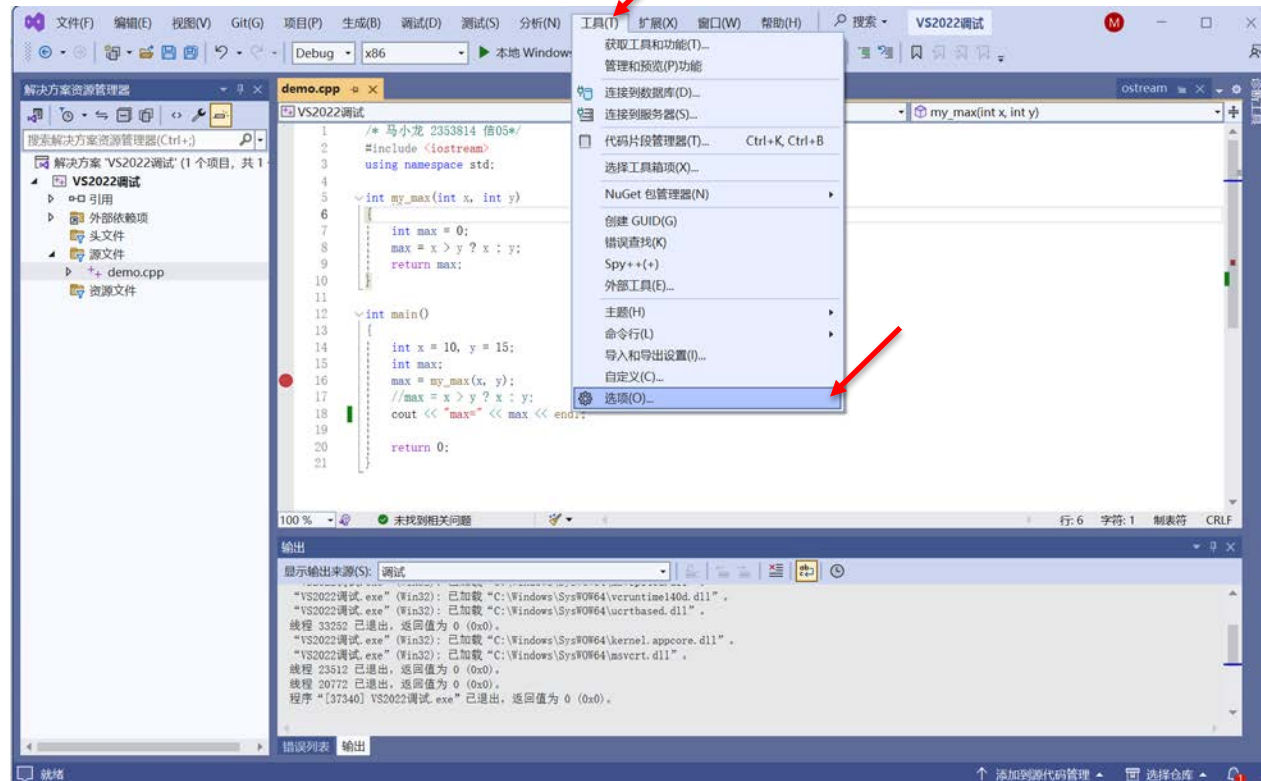




一、基本使用方法

VS2022在默认的情况下调试时只显示“我的代码”，所以无论是“逐语句”还是“逐过程”，都不会进入系统类/系统函数的内部单步执行；因此我们需要取消这一选项。

第一步：选择菜单-“工具”-“选项”，弹出窗口

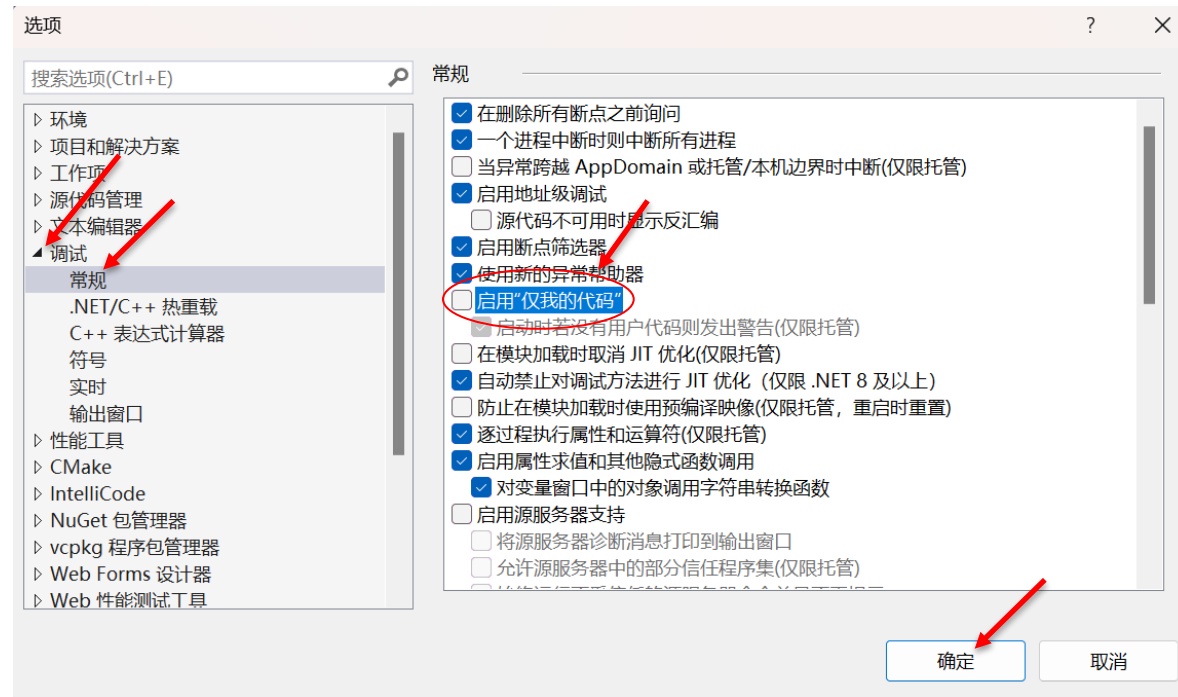




一、基本使用方法

VS2022在默认的情况下调试时只显示“我的代码”，所以无论是“逐语句”还是“逐过程”，都不会进入系统类/系统函数的内部单步执行；因此我们需要取消这一选项。

第二步：在弹出的窗口中选择“调试”“常规”，
取消勾选“启用‘仅我的代码’”，点击“确定”

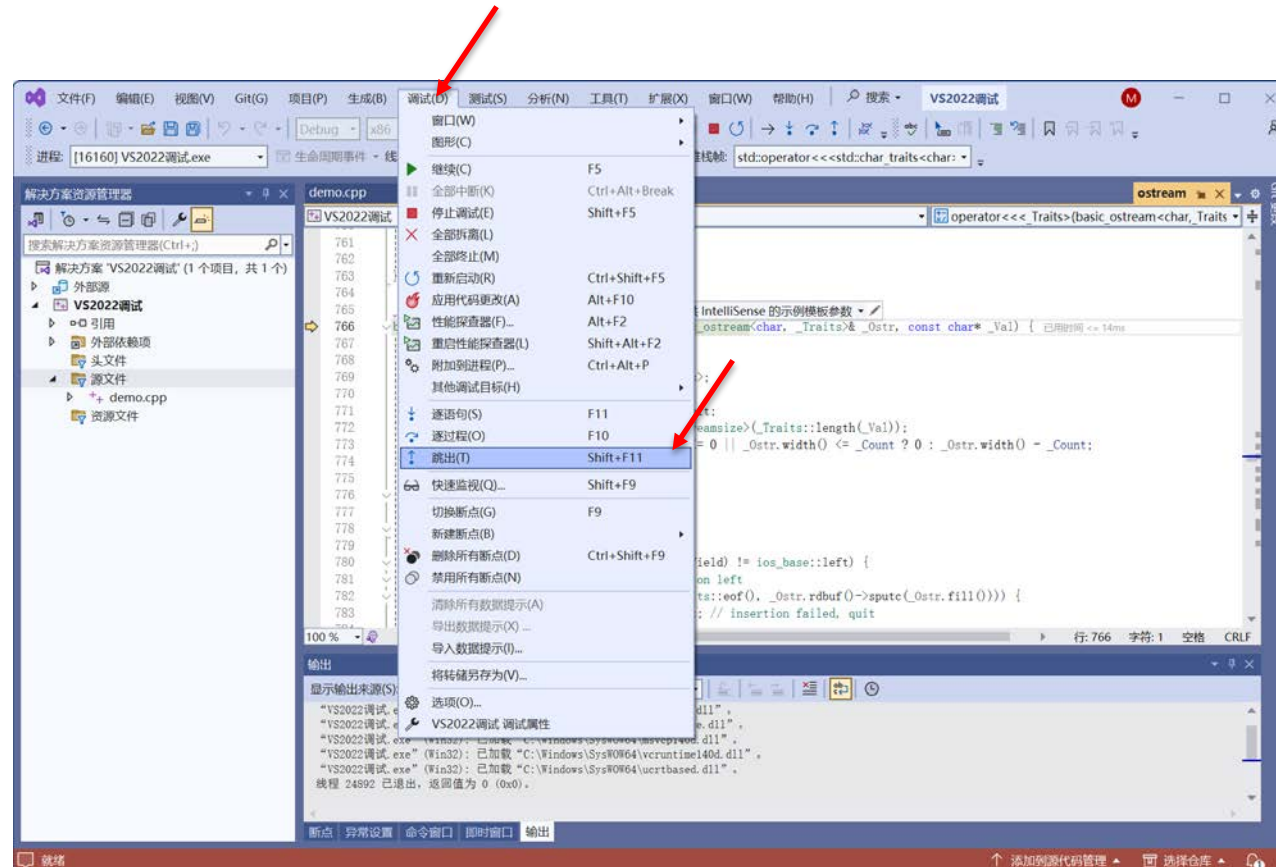




一、基本使用方法

4.跳出系统类/系统函数(1.4)

方法一：选择菜单-“调试”-“跳出”

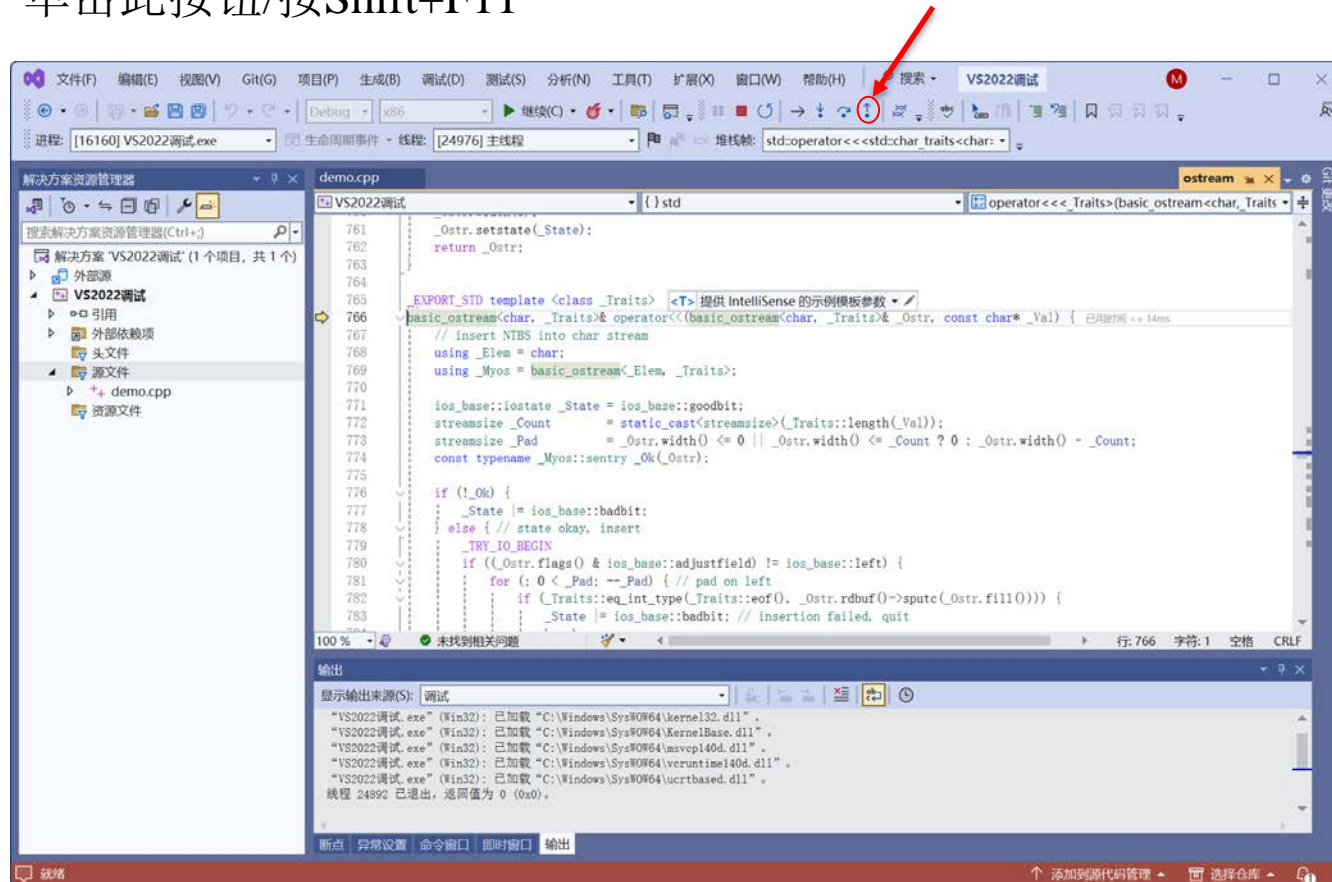




一、基本使用方法

4.跳出系统类/系统函数(1.4)

方法二：单击此按钮/按Shift+F11

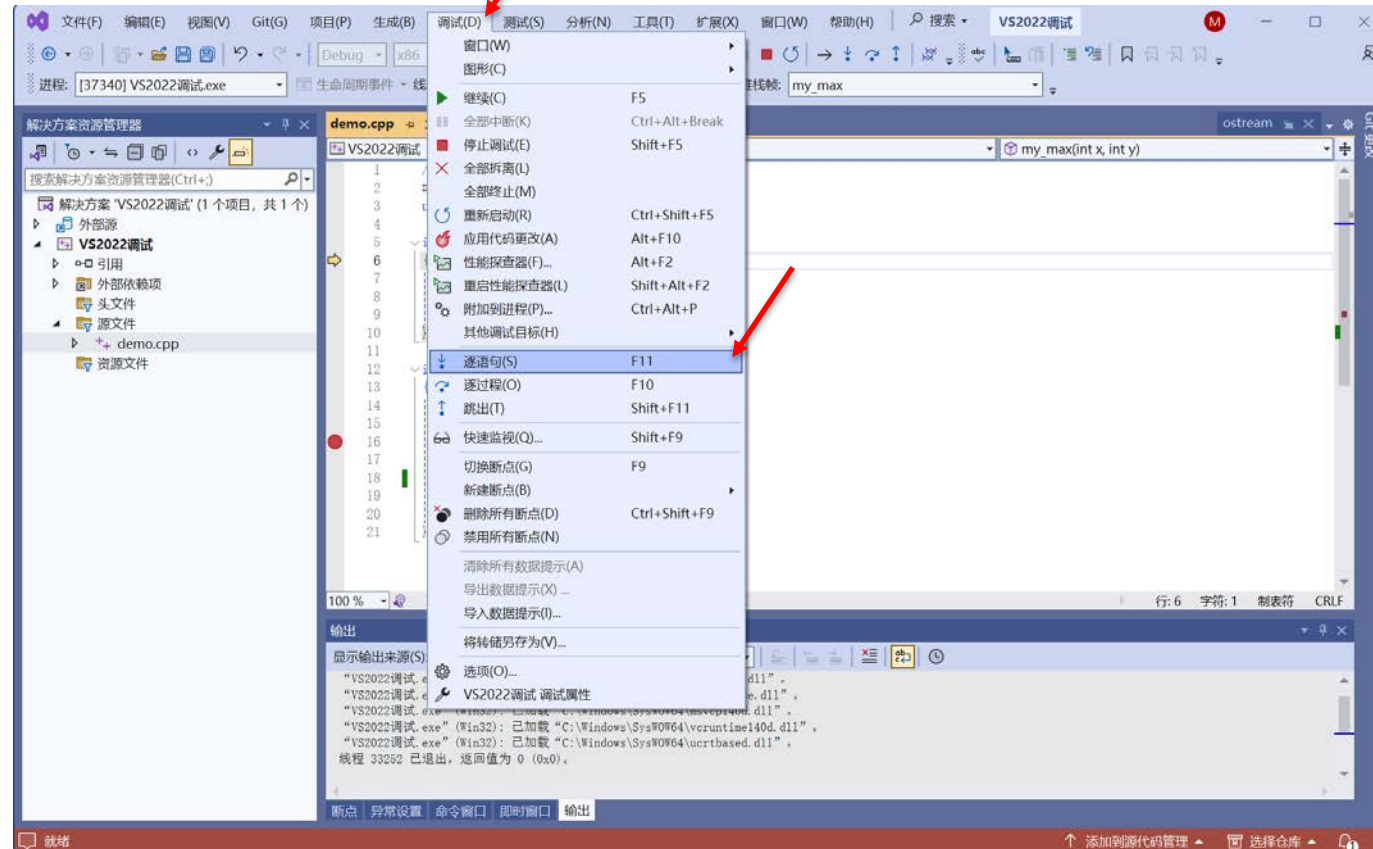




一、基本使用方法

5. 转到被调用自定义函数内部单步中 (1.6)

方法一：选择菜单-“调试”-“逐语句”

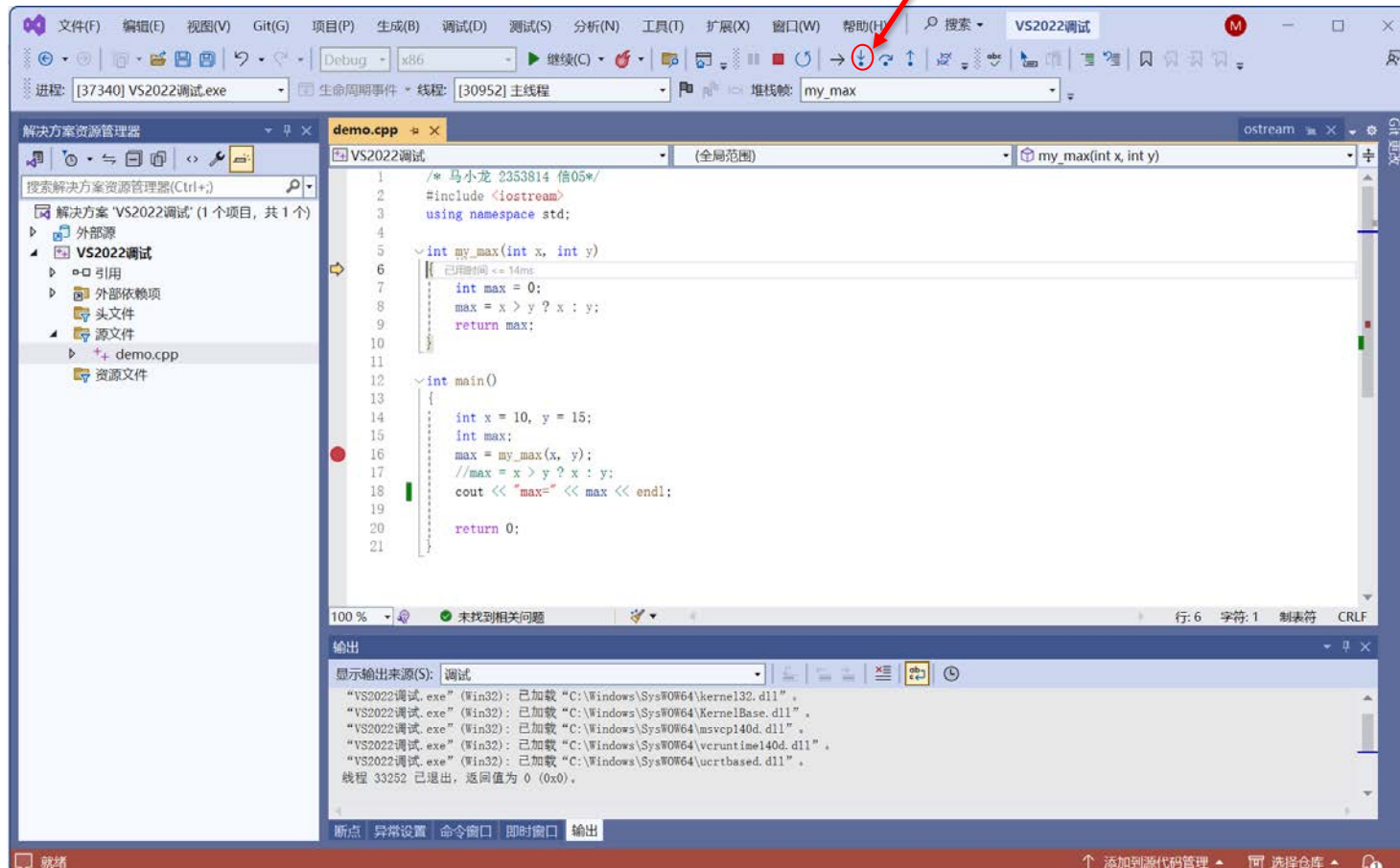




一、基本使用方法

5.转到被调用自定义函数内部单步中(1.6)

方法二：单击此按钮/按F11

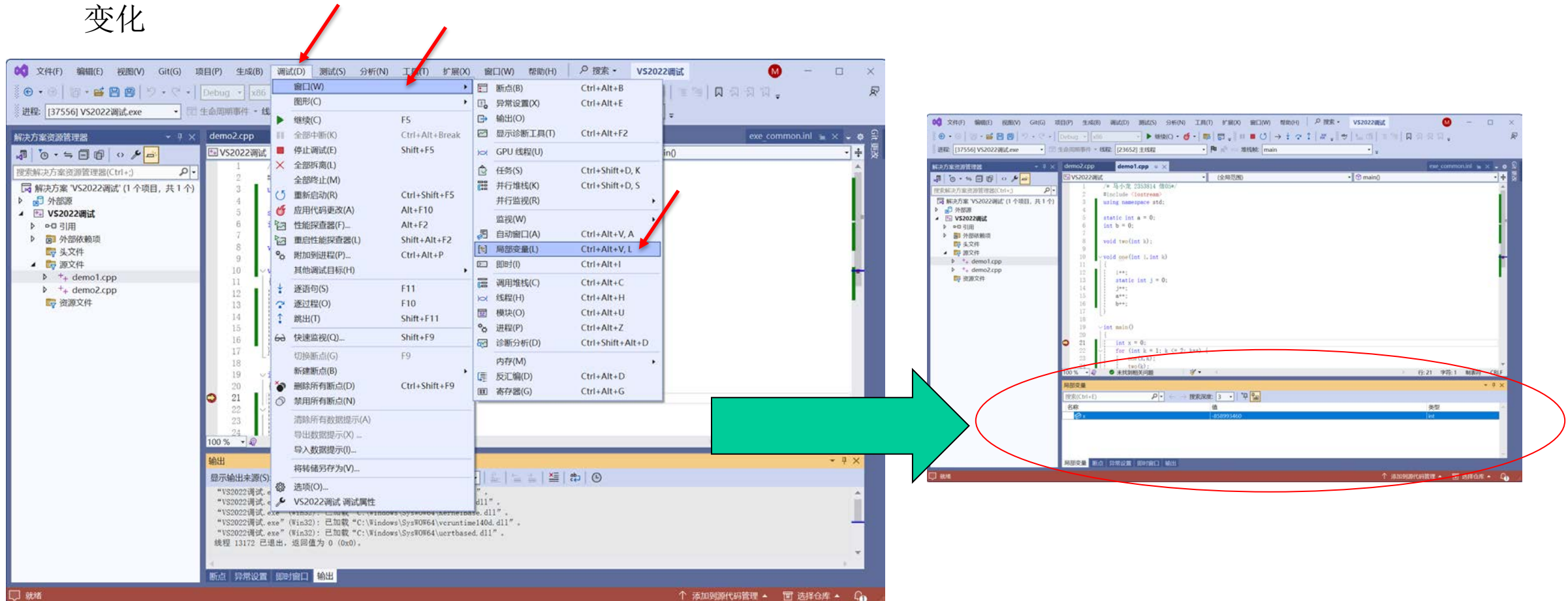




二、查看各种生存期/作用域变量

1.查看形参/自动变量和静态局部变量的变化情况(2.1、2.2)

开始调试，选择菜单-“调试”-“窗口”-“局部变量”，打开局部变量窗口。单步执行可以观察变量变化



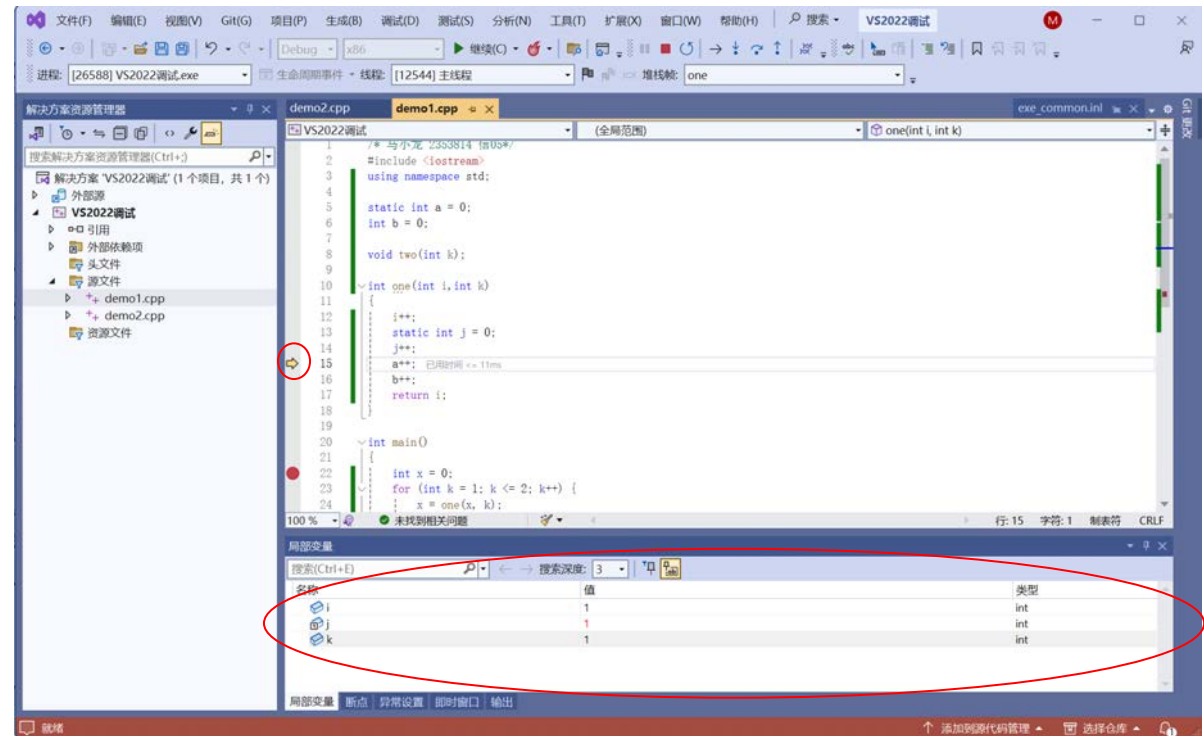
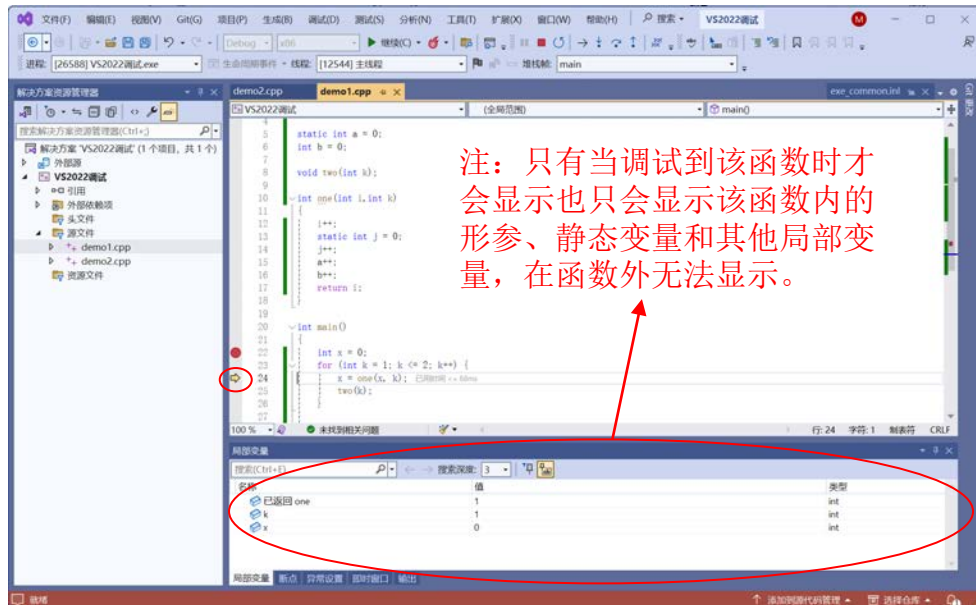


二、查看各种生存期/作用域变量

1.查看形参/自动变量和静态局部变量的变化情况(2.1、2.2)

局部变量窗口内显示当前调试执行的位置所在函数内的形参、静态变量和其他局部变量及其变化。

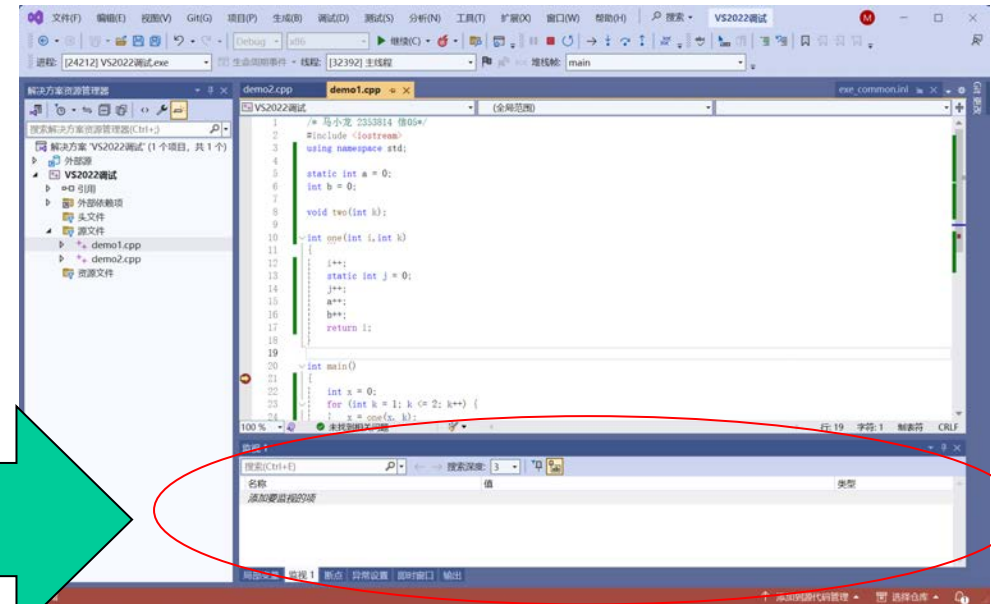
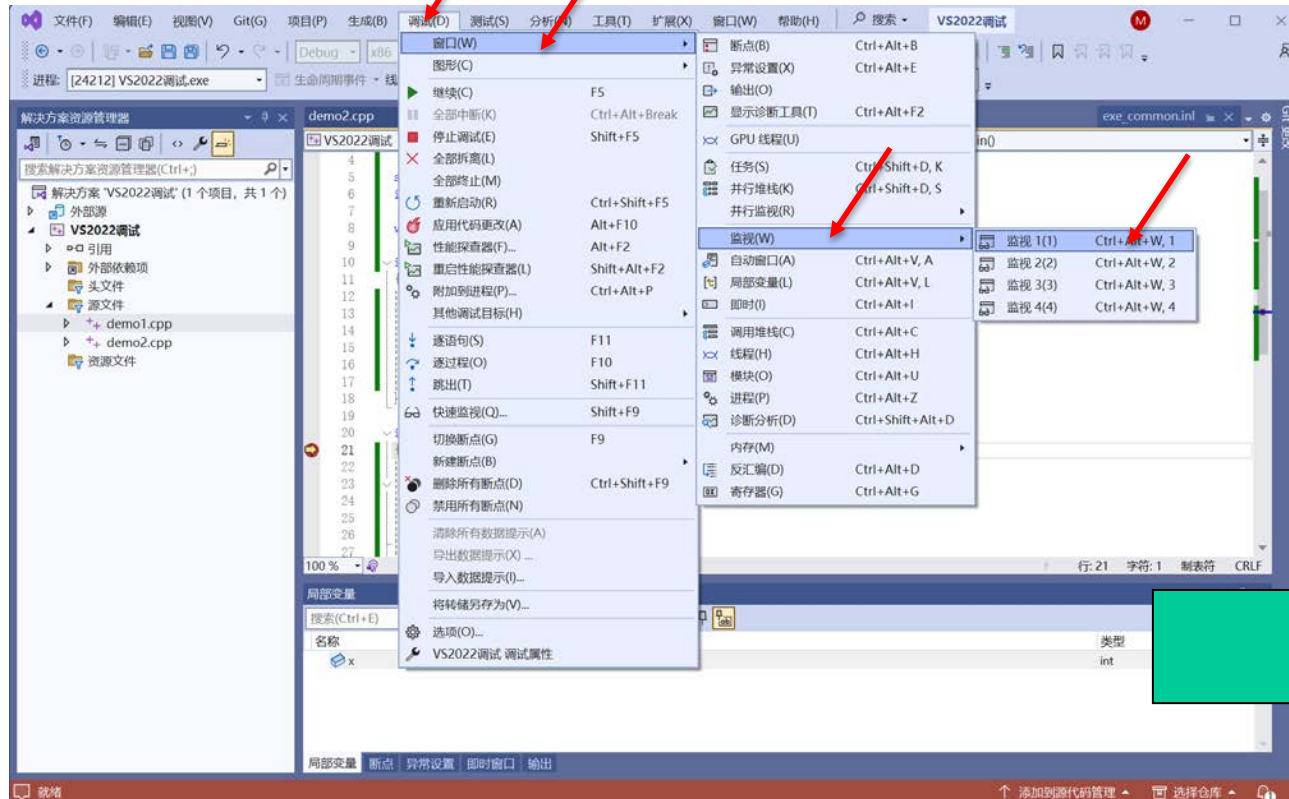
注：只有当调试到该函数时才会显示也只会显示该函数内的形参、静态变量和其他局部变量，在函数外无法显示。



二、查看各种生存期/作用域变量

2. 查看静态全局变量、外部全局变量的变化情况(2.3、2.4)

第一步：开始调试，选择菜单-“调试”-“窗口”-“监视” “监视1(1)”，打开监视窗口

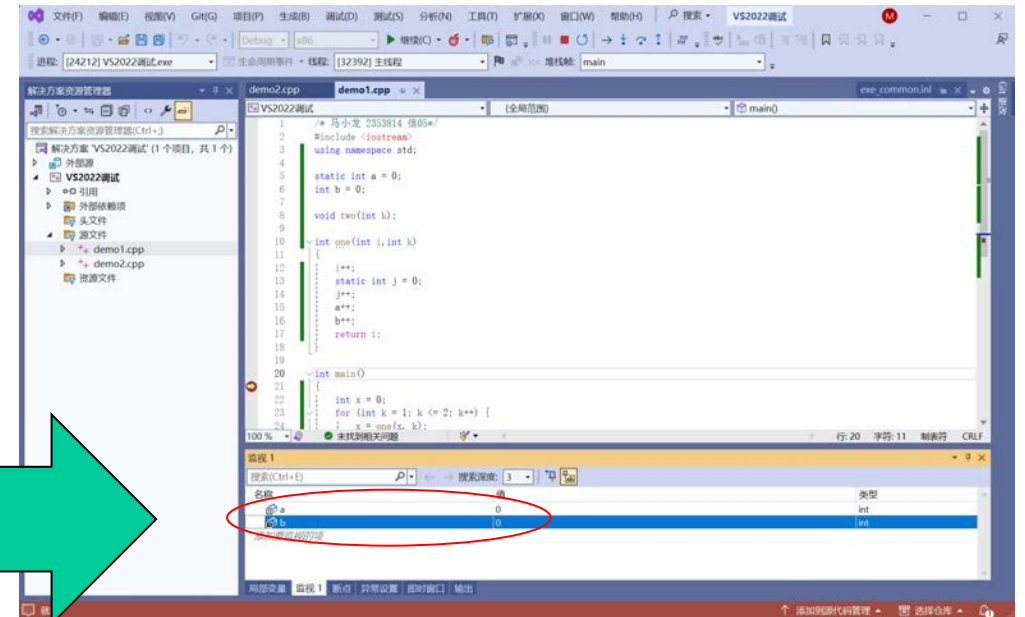
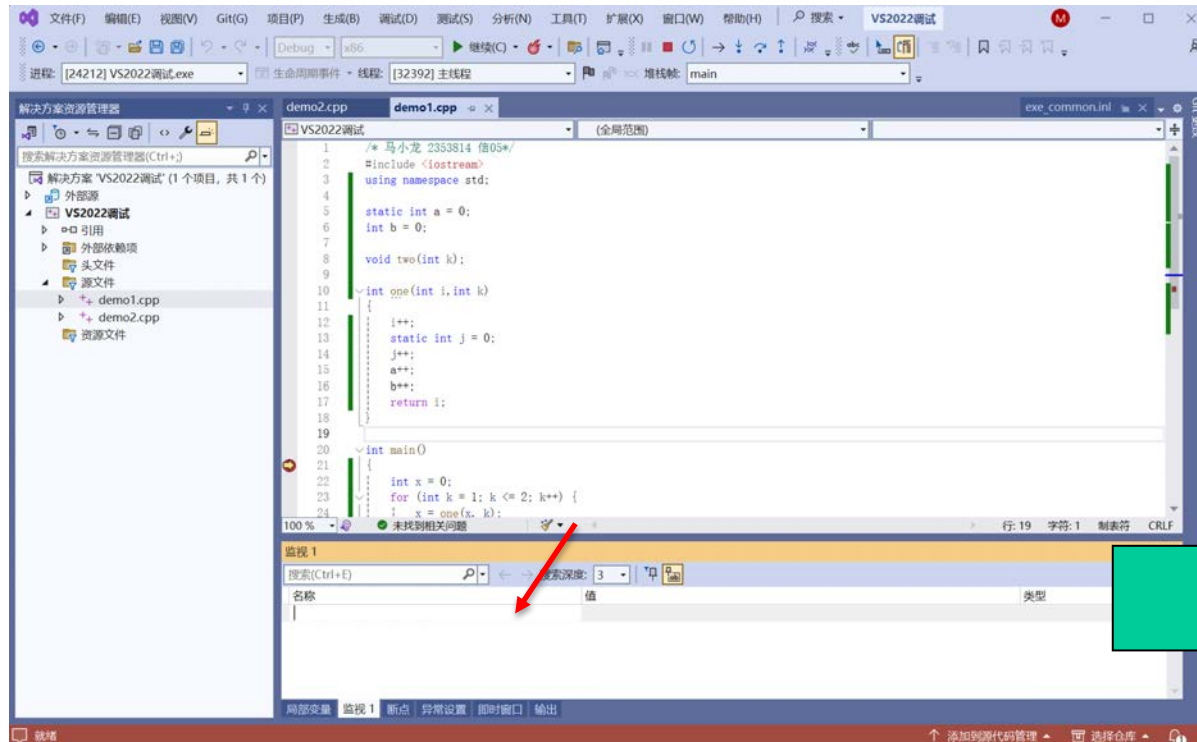




二、查看各种生存期/作用域变量

2. 查看静态全局变量、外部全局变量的变化情况(2.3、2.4)

第二步：在监视窗口中，输入需要监视的外部静态全局变量、外部全局变量。单步执行可以观察变量变化



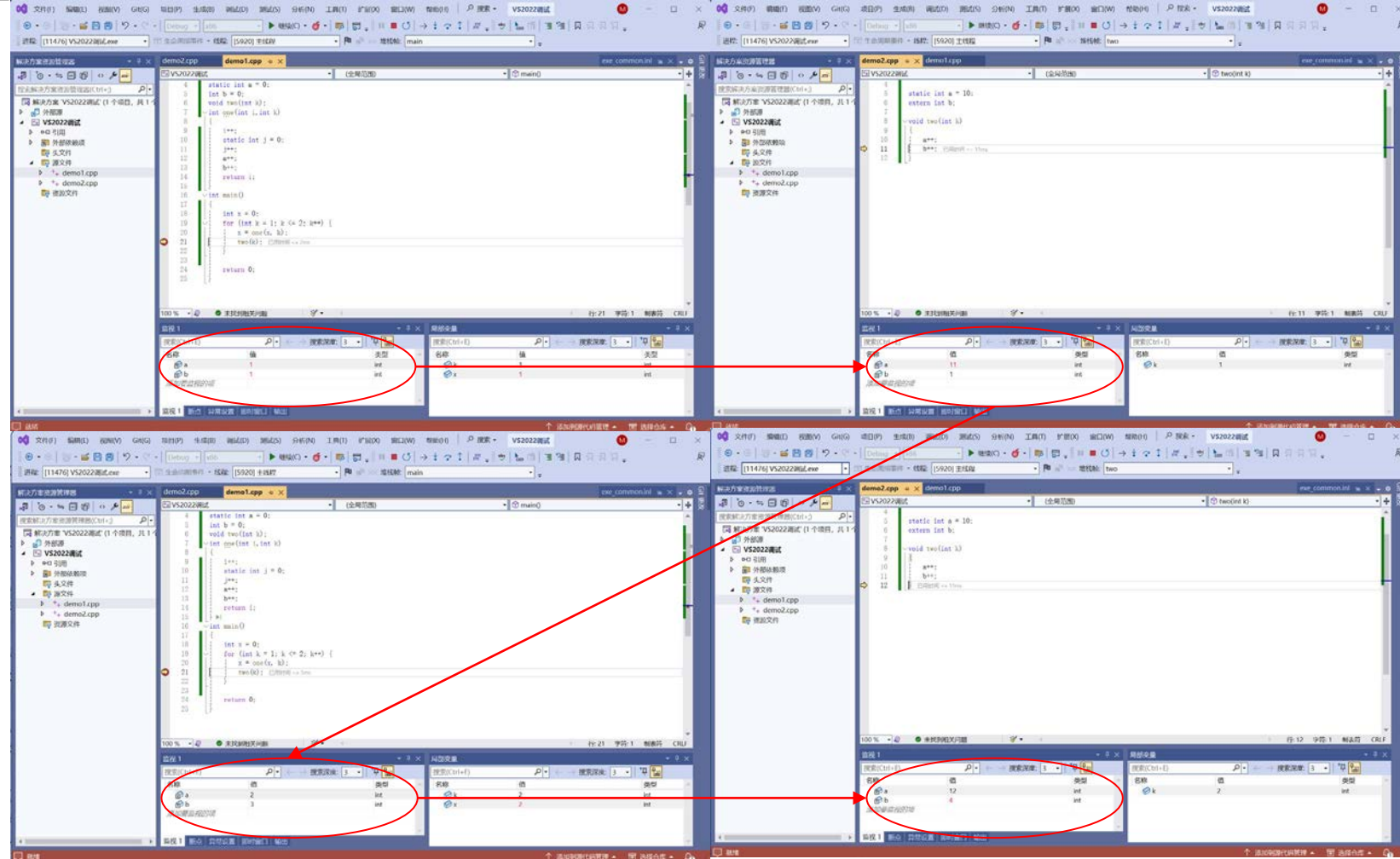


二、查看各种生存期/作用域变量

2. 查看静态全局变量、外部全局变量的变化情况(2.3、2.4)

*对于两个源程序中有同名静态全局变量，监视窗口会显示当前调试执行位置所在源程序中的静态全局变量。

*对于两个源程序，一个定义外部全局变量，一个有extern说明，外部全局变量始终是一个，可以正常查看





二、查看各种生存期/作用域变量

2. 查看静态全局变量、外部全局变量的变化情况(2.3、2.4)

注：1.对于两个源程序中有同名静态全局变量，若只有一个源程序有关于静态全局变量的表达式，则监视窗口只显示该程序中的静态全局变量。

2.若两个程序都只定义，不运算，则显示该静态全局变量未定义



三、查看各种不同类型变量

1.char/int/float等简单变量及指向简单变量的指针变量(3.1、3.2)

对于简单变量及指向简单变量的指针变量，直接在局部变量窗口或监视窗口查看即可

注：对于指针变量，会同时显示指针地址和指向的值，地址在前，值在后

名称	值	类型
quan_ret	false	bool
pquan_ret	0x0071c189 (VS2022调试_exe!bool quan_ret) (false)	bool *
quan_a	0 '\0'	char
pquan_a	0x0071c18a ""	char *
quan_b	65 'A'	char
pquan_b	0x0071c088 "A"	char *
quan_c	2353814	int
pquan_c	0x0071c090 (VS2022调试_exe!int quan_b) (2353814)	int *
quan_d	1.00000000	float
pquan_d	0x0071c098 (VS2022调试_exe!float quan_c) (1.00000000)	float *
quan_d	123.45600000000000	double
pquan_d	0x0071c0a0 (VS2022调试_exe!double quan_d) (123.45600000000000)	double *

注：对于指针变量，会同时显示指针地址和指向的值，地址在前，值在后

名称	值	类型
a	0 '\0'	char
a1	65 'A'	char
b	2353814	int
c	1.00000000	float
d	123.45600000000000	double
pa	0x006ff83f ""	char *
pa1	0x006ff827 "A (VS2022调试_exe!char a)"	char *
pb	0x006ff80c (2353814)	int *
pc	0x006ff7f4 (1.00000000)	float *
pd	0x006ff7d8 (123.45600000000000)	double *
pret	0x006ff7d8 (false)	bool *
ret	false	bool



三、查看各种不同类型变量

2. 一维数组和指向一维数组的指针变量(3.3、3.4)

对于一维数组和指向一维数组的指针变量，直接在局部变量窗口或监视窗口查看即可。

注：一维数组会显示首地址和所有元素；数字指针会显示首地址的当前地址存储的值；字符指针会显示首地址和从该地址开始到'\0'的字符串

类型中可以看到数组大小

名称	值	类型
b	0x012ffc6c "ABCDEFGab"	char[10]
pb	0x0015c0ac [VS2022调试.exe!int a[7]] [2]	int *
p	0x012ffc6c "ABCDEFGab"	char *

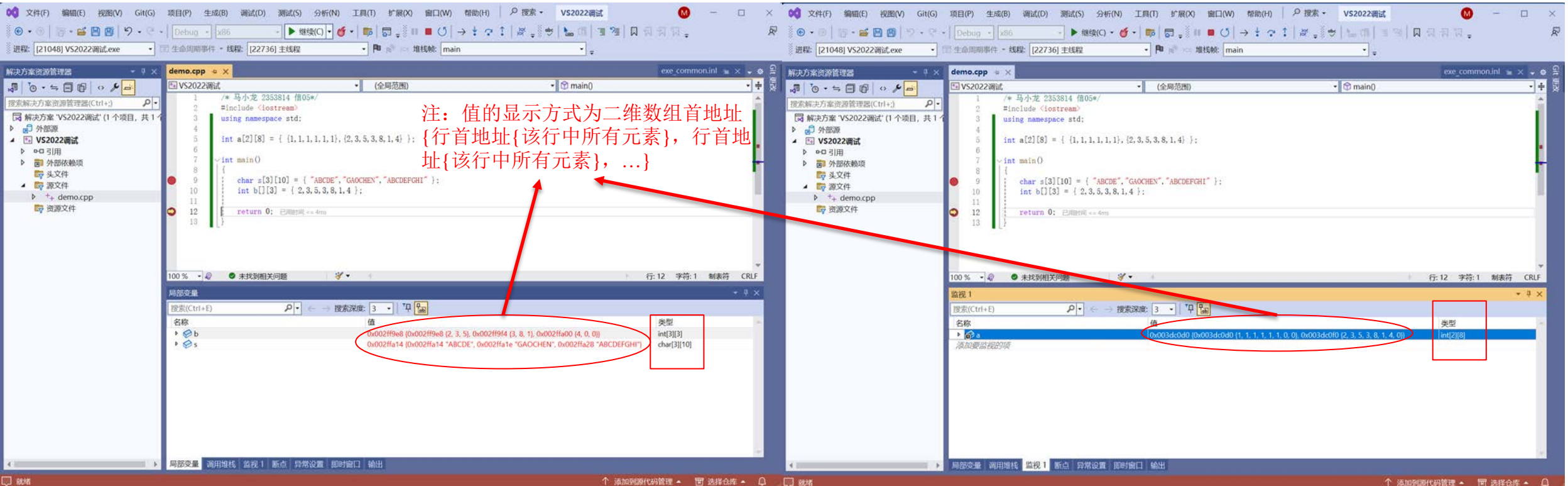
名称	值	类型
a	0x0015c0ac {2, 3, 5, 3, 8, 1, 4}	int[7]
pa	0x0015c0ac [VS2022调试.exe!int a[7]] [2]	int *



三、查看各种不同类型变量

3. 二维数组(3.5)

对于二维数组，直接在局部变量窗口(若为局部变量)或监视窗口(若为全局变量)查看即可。
在类型一栏可以查看数组大小。

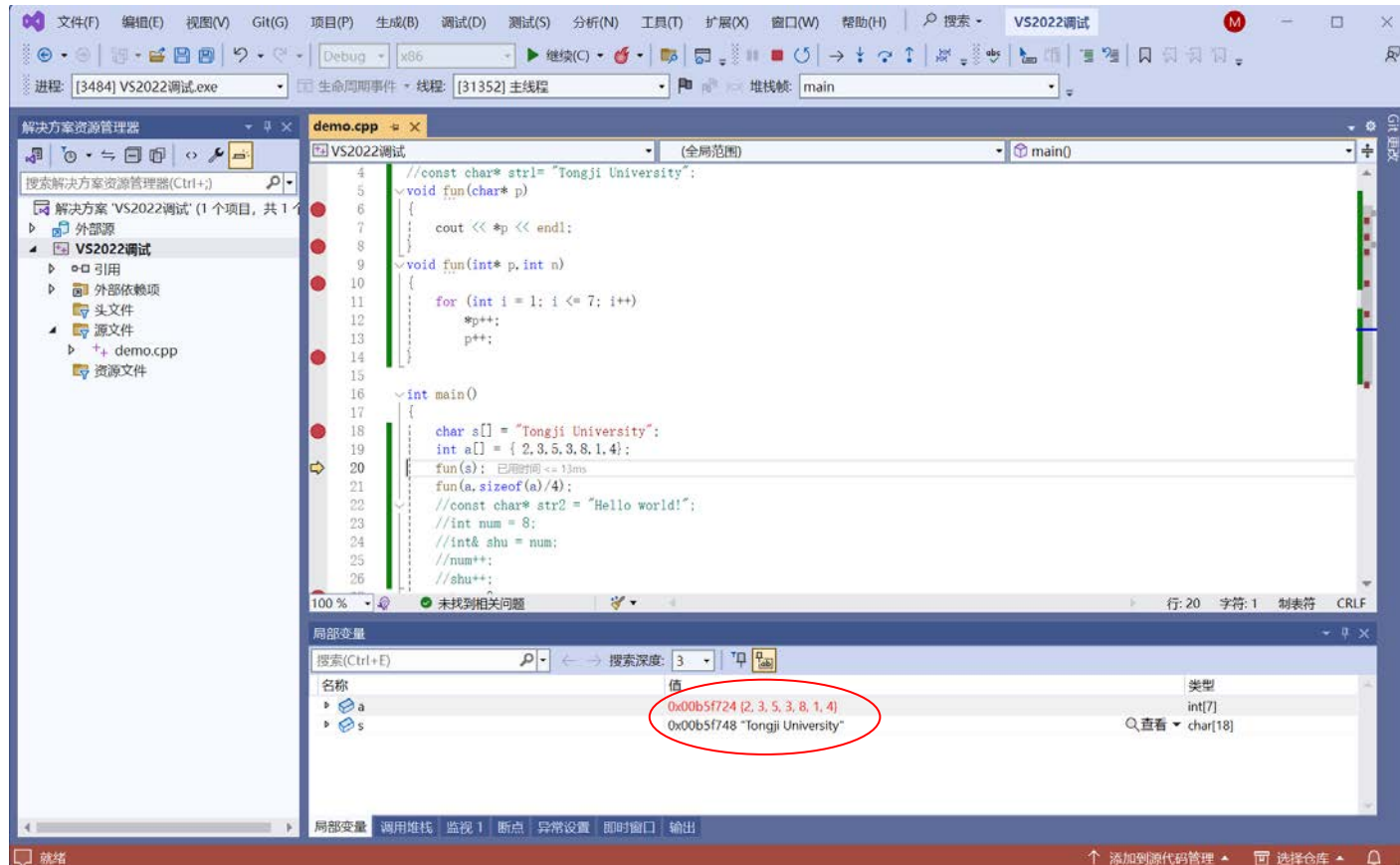




三、查看各种不同类型变量

4.实参是一维数组名，形参是指针(3.6)

可以在局部变量窗口直接查看数组的值



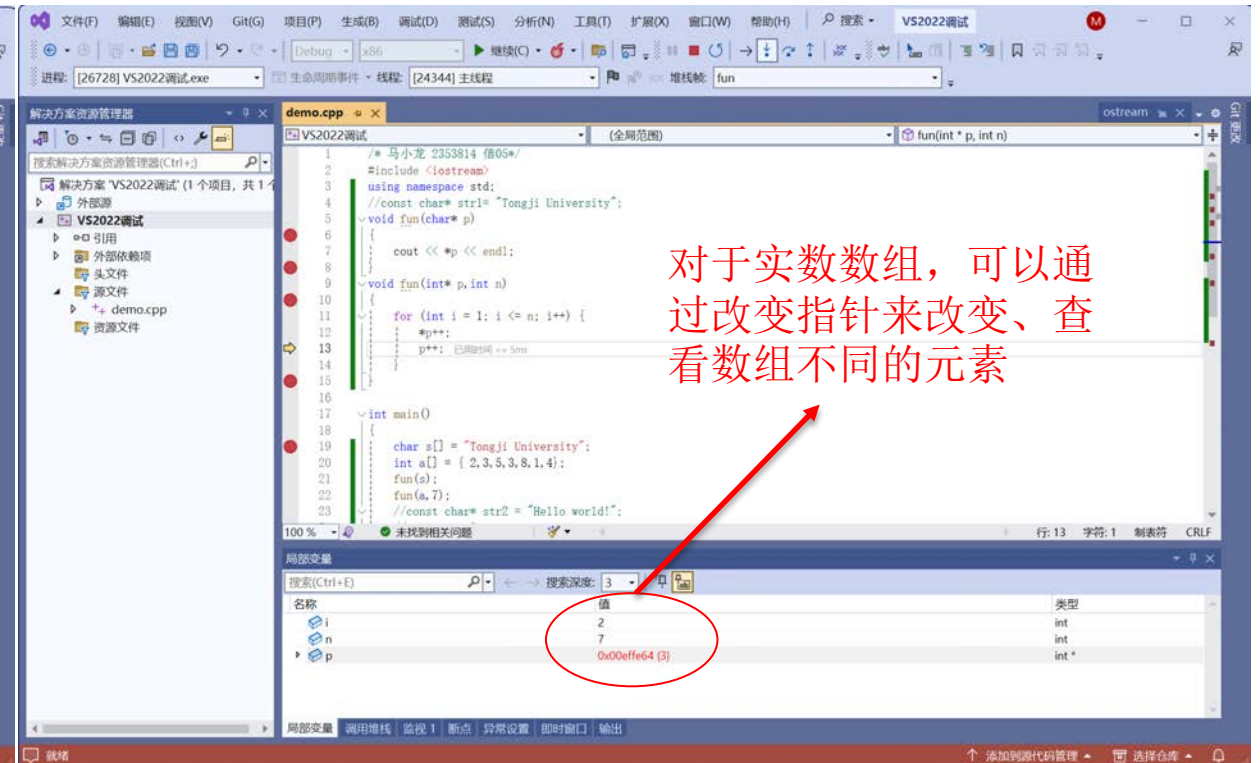
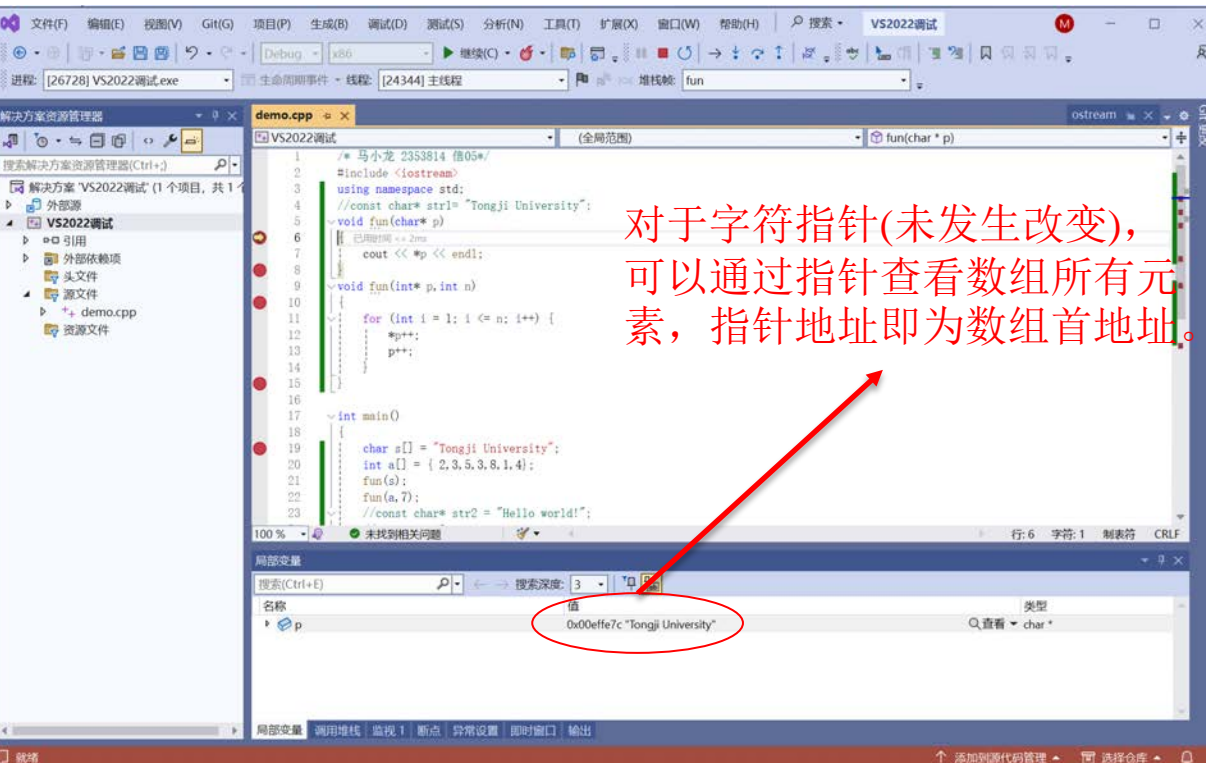


三、查看各种不同类型变量

4.实参是一维数组名，形参是指针(3.6)

方法一：在局部变量窗口直接查看查看

注意：如果调试步骤进行其他函数，则在该窗口无法查看该数组。

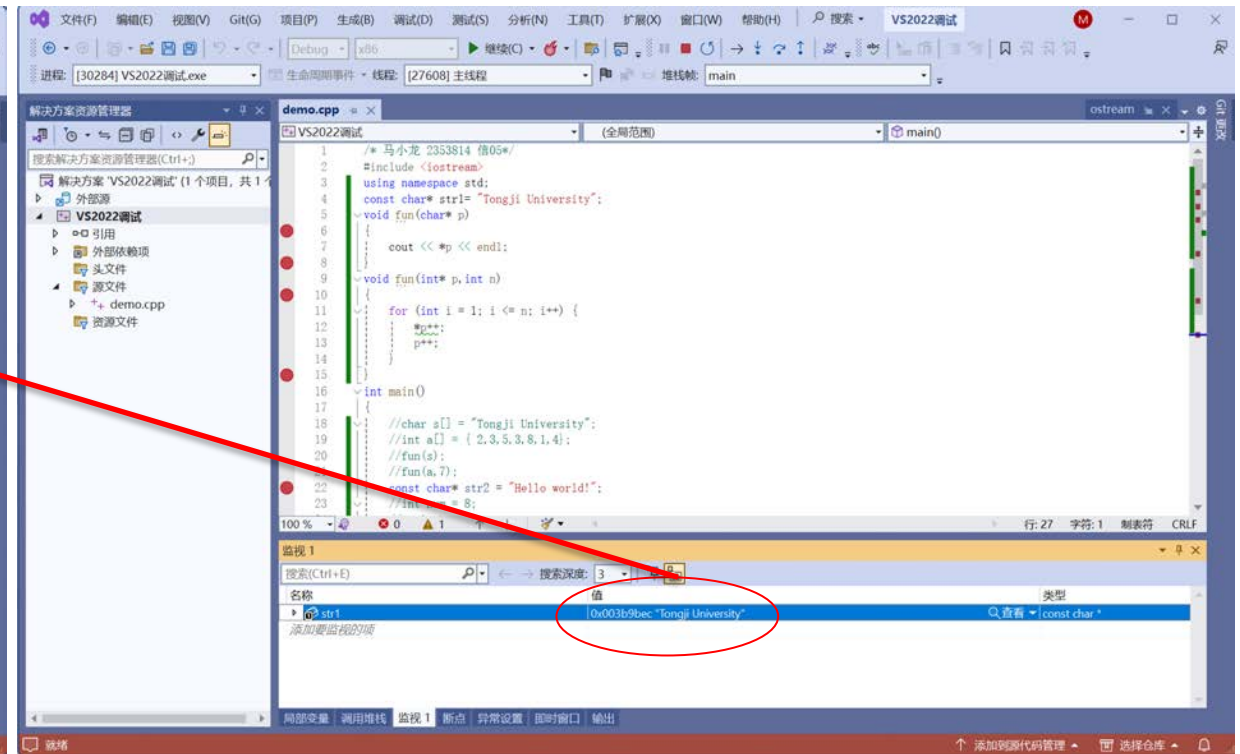
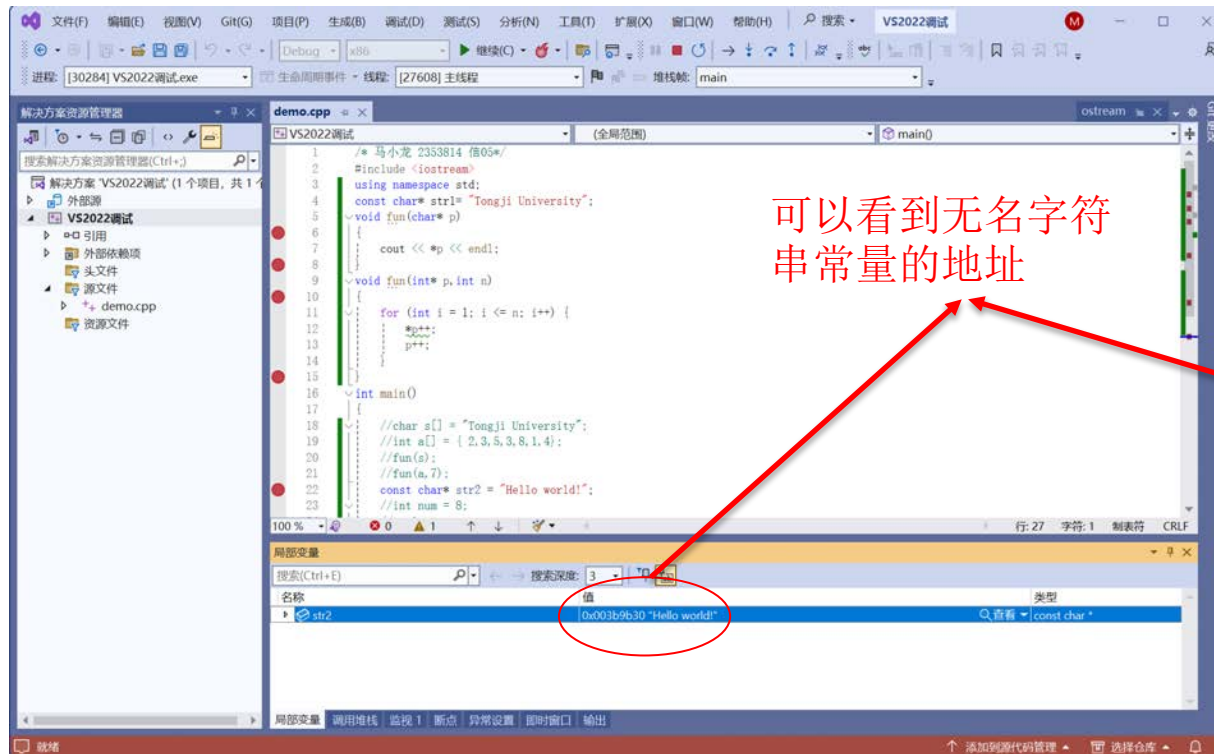




三、查看各种不同类型变量

5. 指向字符串常量的指针变量(3.7)

可以直接在局部变量或监视窗口查看指向字符串常量的指针变量

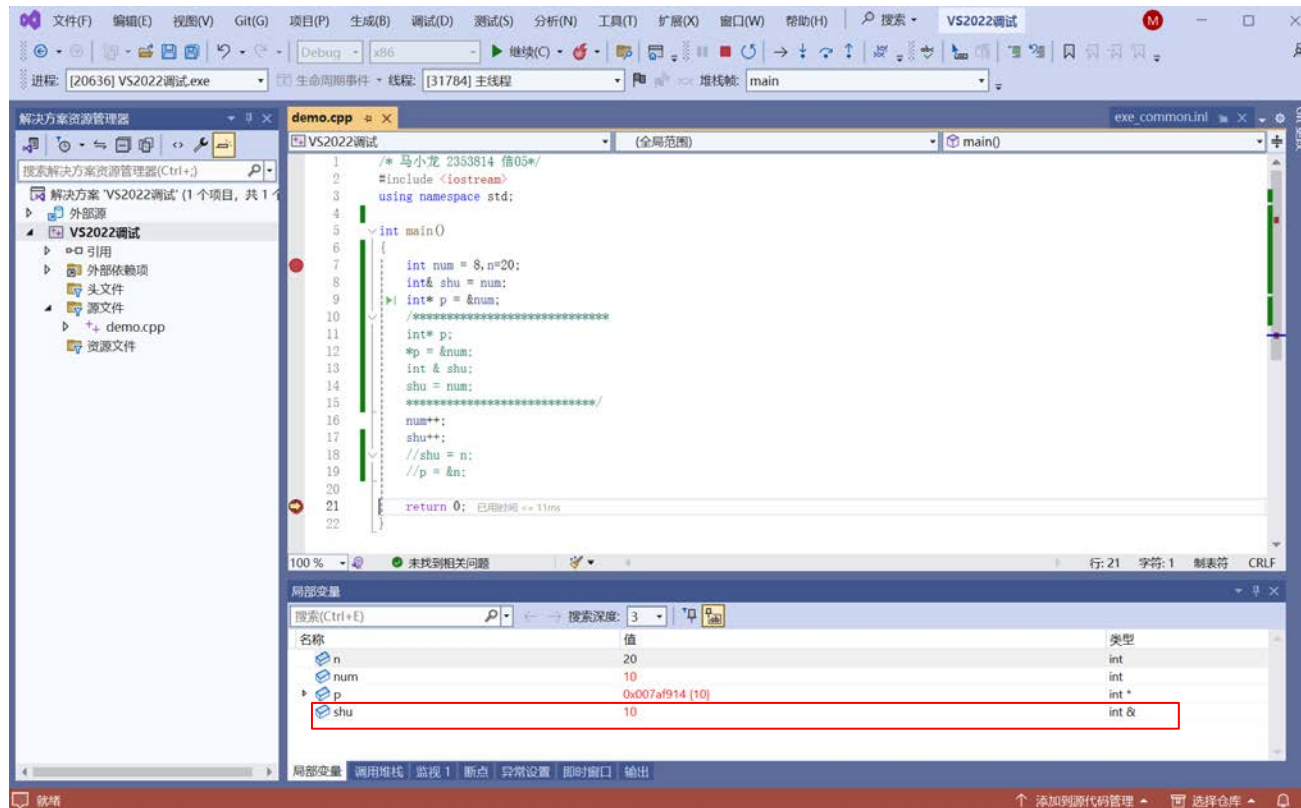




三、查看各种不同类型变量

6. 引用(3.8)

可以直接在局部变量或监视窗口查看引用



引用和指针区别:

- 1.引用是变量的别名，不是新定义的变量；而指针是存储地址的变量
- 2.引用在定义时必须初始化，此后便不能被改变指向另一个对象，而指针不必在定义时初始化，也可以改变指向的对象；

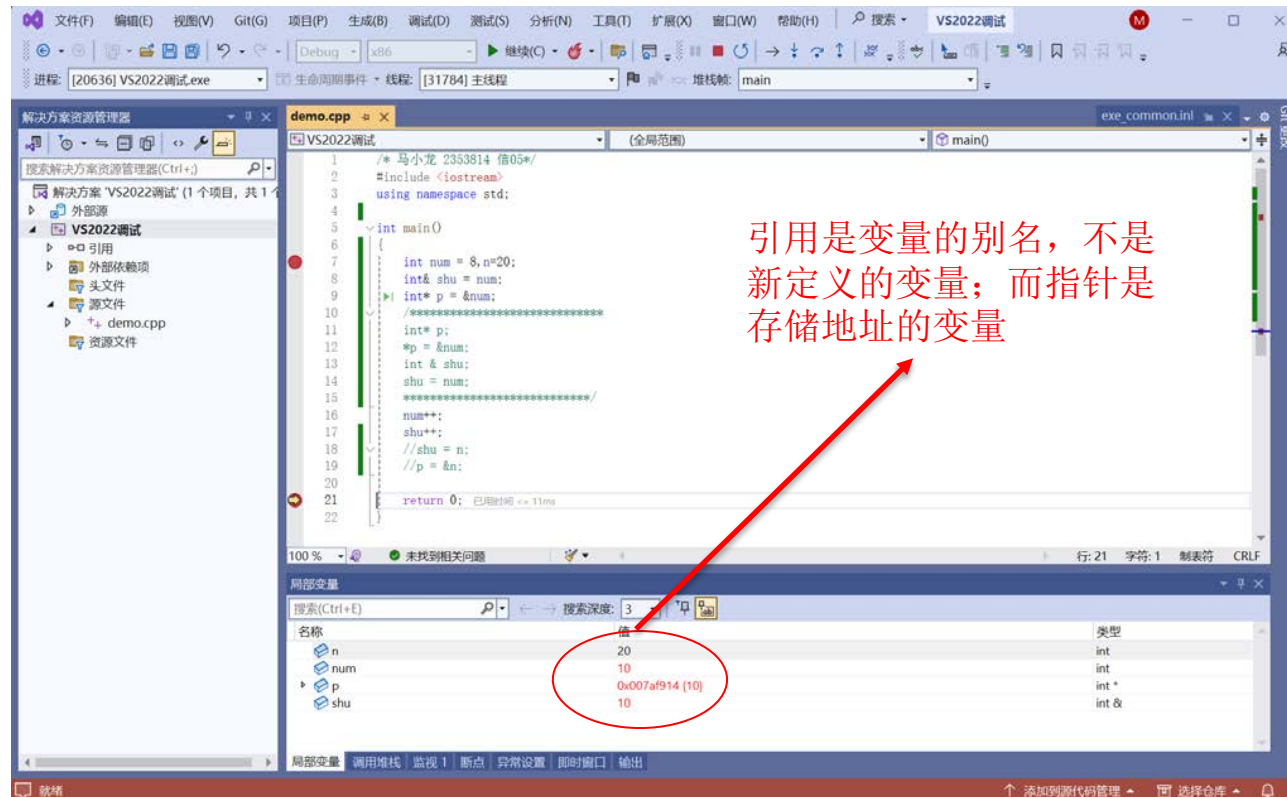


三、查看各种不同类型变量

6. 引用(3.8)

引用和指针区别:

1) 引用是变量的别名, 不是新定义的变量; 而指针是存储地址的变量





三、查看各种不同类型变量

6. 引用(3.8)

引用和指针区别:

2) 引用在定义时必须初始化, 而指针不必在定义时初始化

```
demo.cpp  VS2022调试 (全局范围)
1  /* 马小龙 2353814 信05*/
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7      int num = 8, n=20;
8      //int& shu = num;
9      //int* p = &num;
10
11      int* p;
12      p = &num;
13      int &shu;
14      shu = num;
15
16      num++;
17      shu++;
18      //shu = n;
19      //p = &n;
20
21      return 0;
22  }
```

error C2530: “shu” : 必须初始化引用

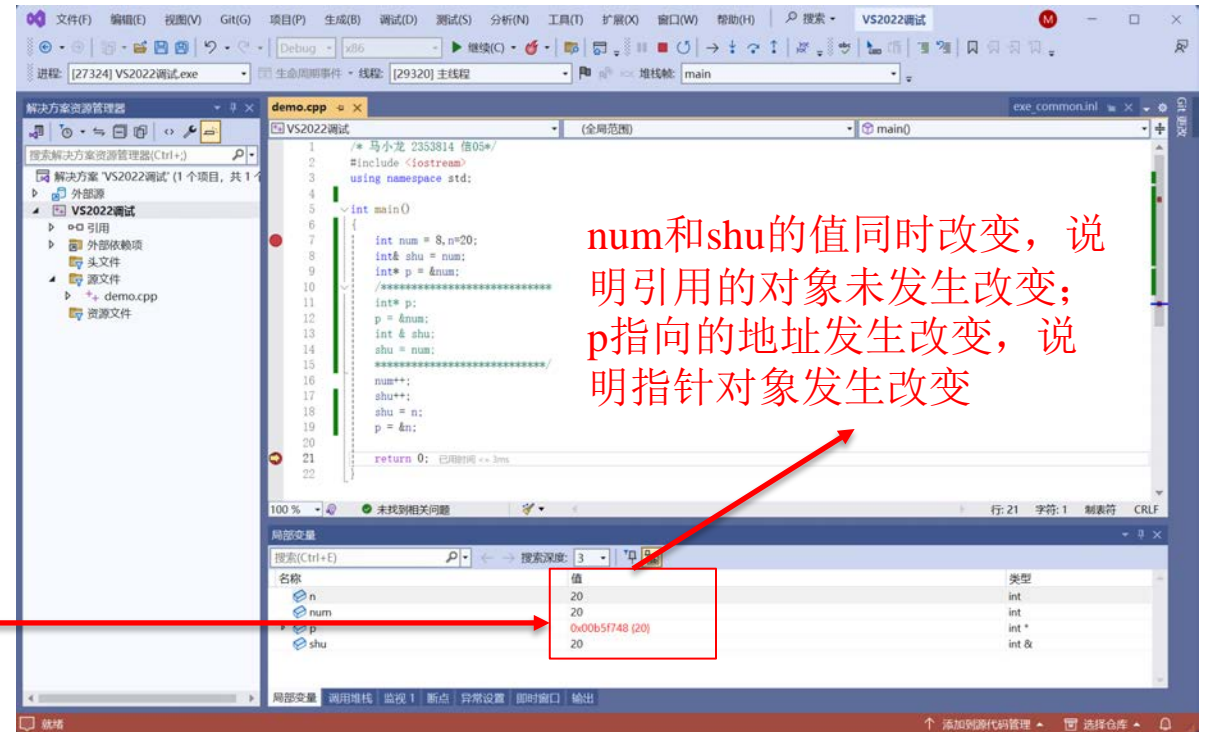
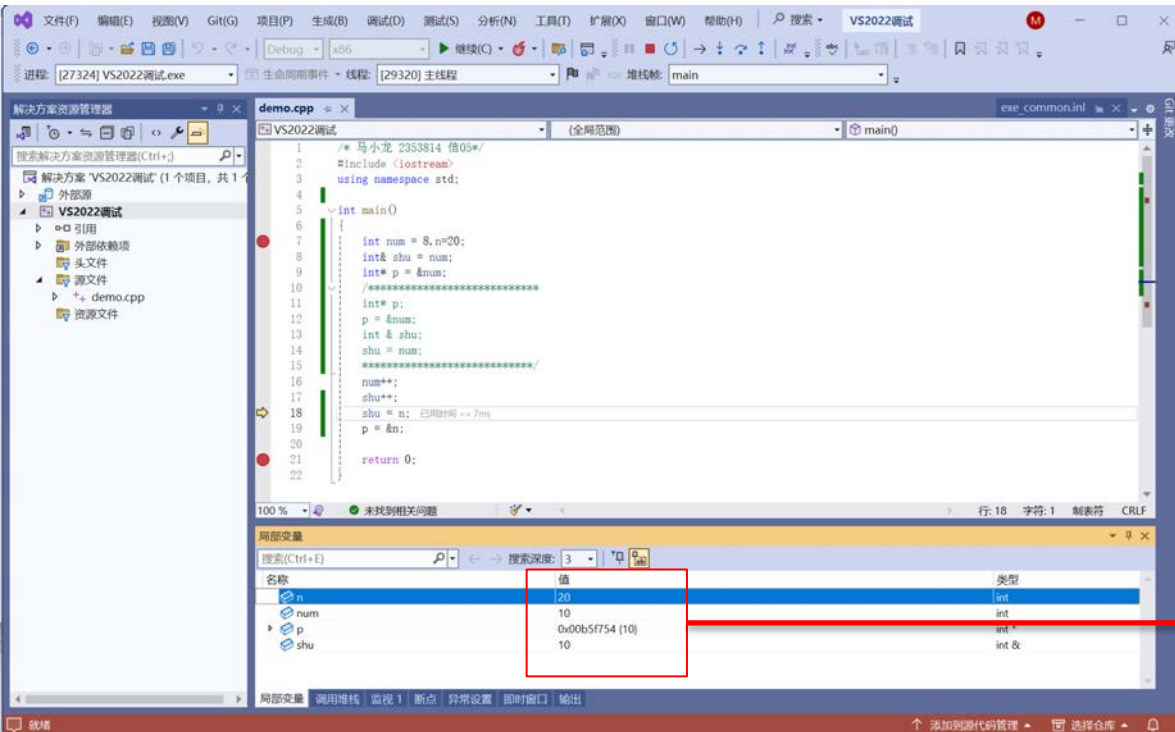


三、查看各种不同类型变量

6. 引用(3.8)

引用和指针区别:

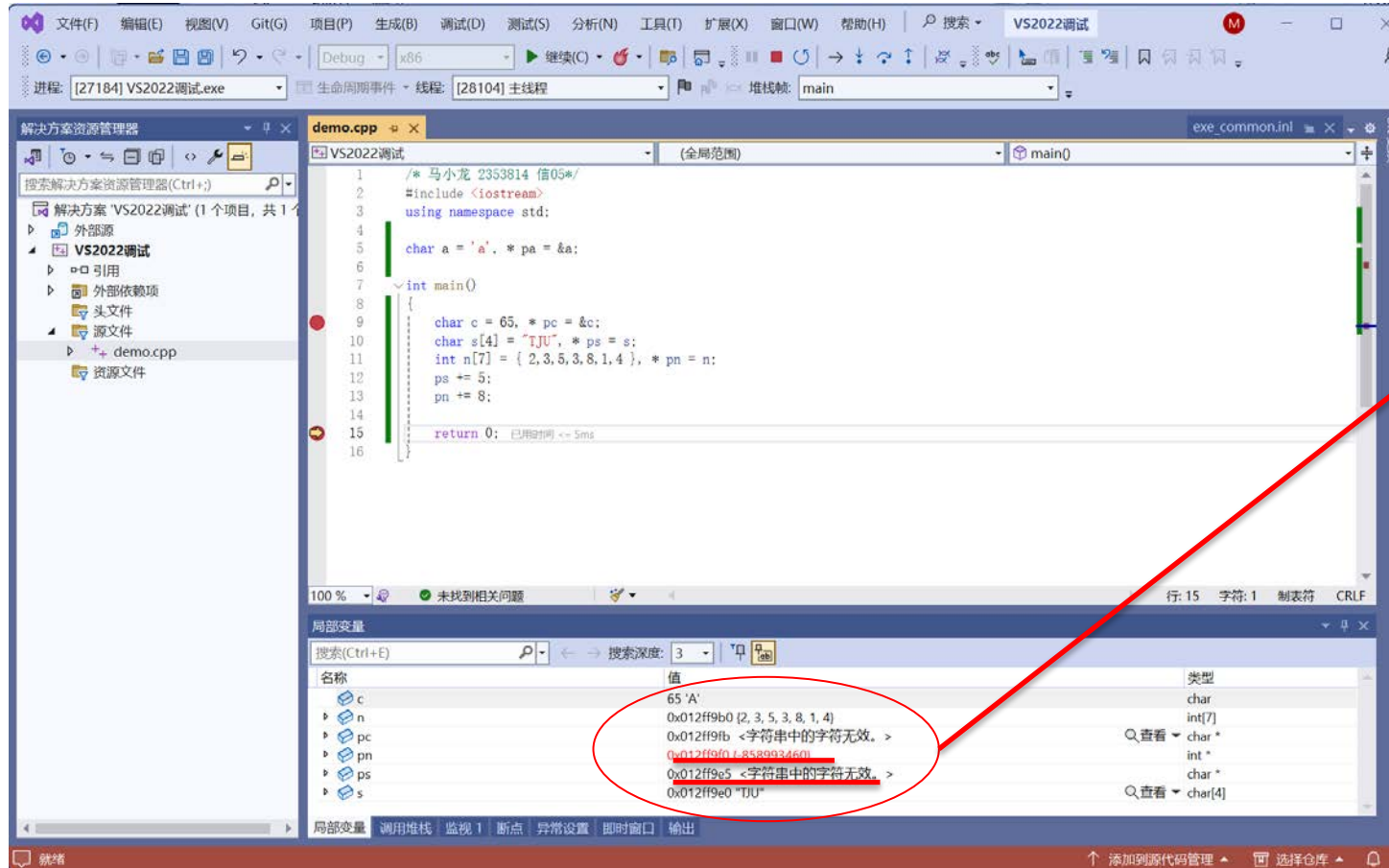
4) 引用不能被改变指向另一个对象, 指针可以改变指向的对象





三、查看各种不同类型变量

7. 使用指针时出现越界访问(3.9)



当实数数组指针越界访问时，
指针会指向不可信的值；
当字符数组指针越界访问时，
会提示字符串中的字符无效



三、查看各种不同类型变量

7. 使用指针时出现越界访问(3.9)

若指针为char类型局部变量指针，指向char类型简单变量，会出现越界访问；若为char类型全局变量指针，则不会出现这一情况。

