# VS2022 调试工具的使用

姓　　名：苗君文

学　　号：2253893

班　　级：信 06

完成日期：2023 年 5 月 30 日

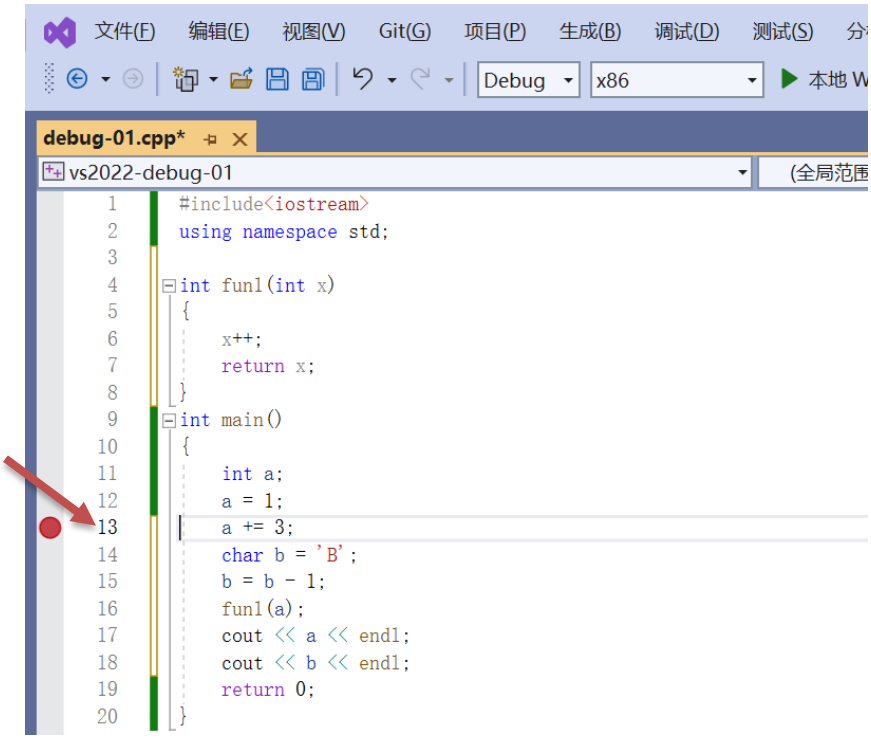
二〇二三年五月

# 1. VS2022下调试工具的基本使用方法

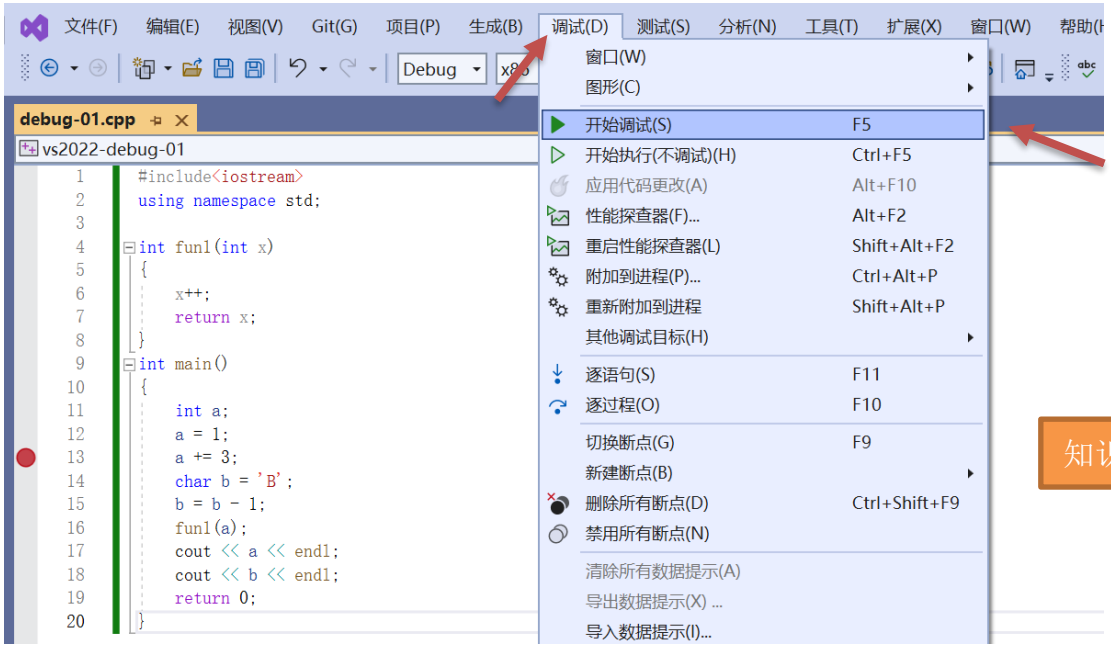
## 1.1. 如何开始调试？如何结束调试？

### 1.1.1. 如何开始调试？

输入代码完成后，在代码左侧灰色区域点亮某一/几行红点，即断点。

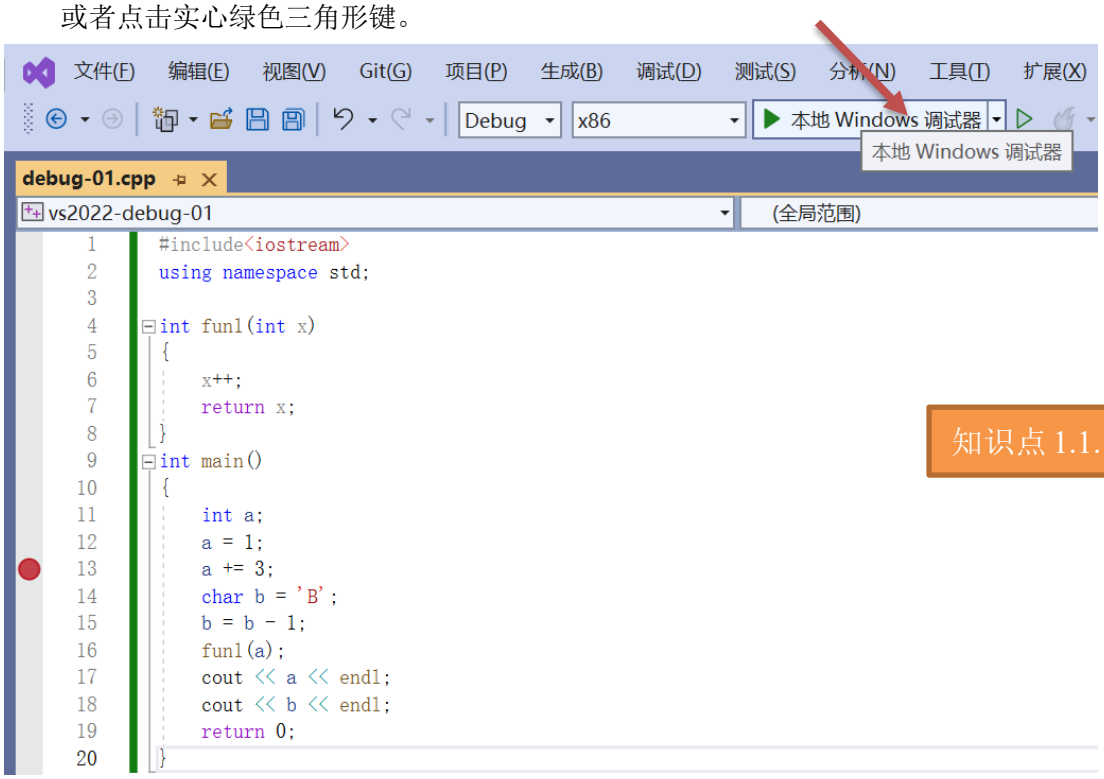



第2步：“调试”-“开始调试”；或者直接按 F5 键。



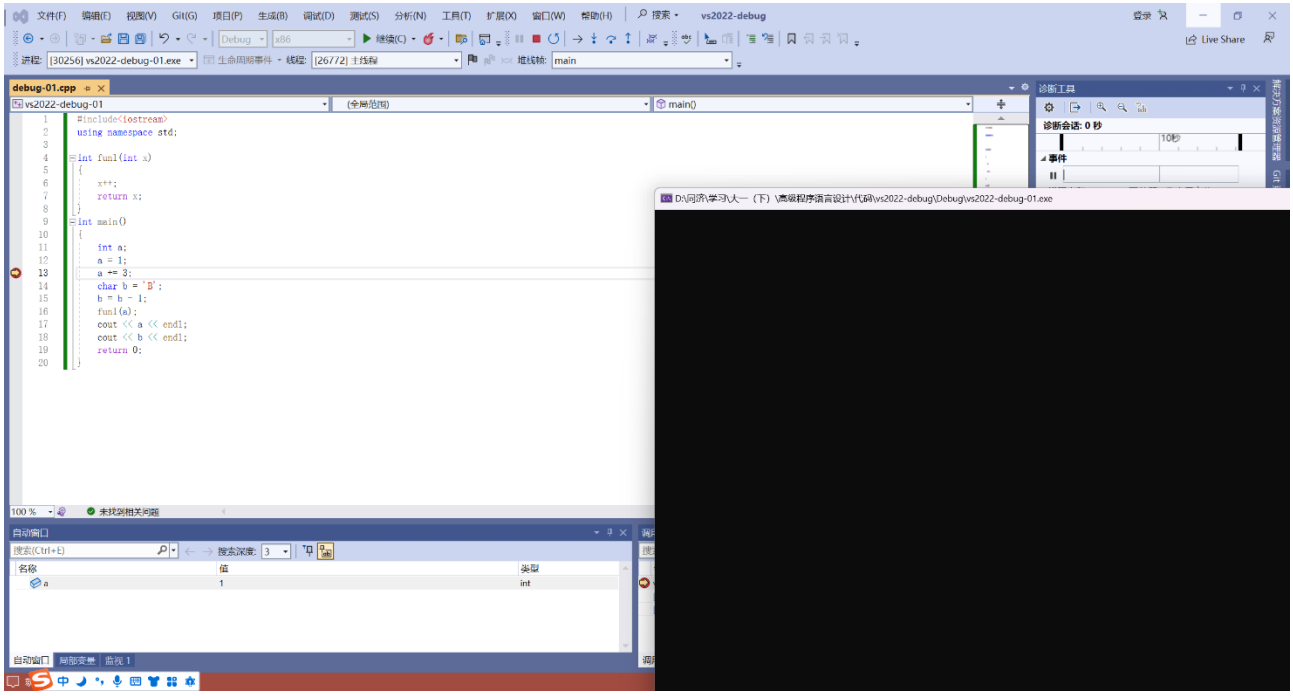

知识点 1.1.1

或者点击实心绿色三角形键。



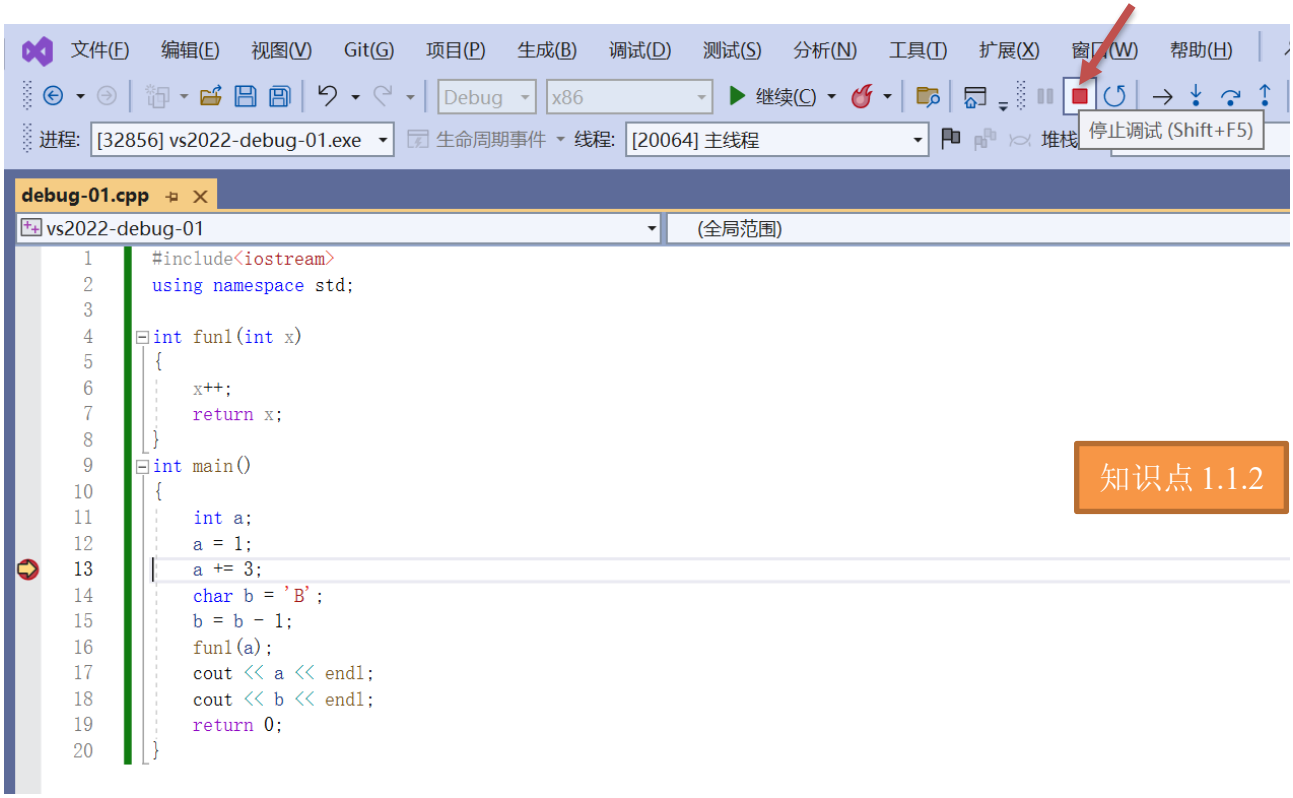

知识点 1.1.1

开始调试后出现以下画面：



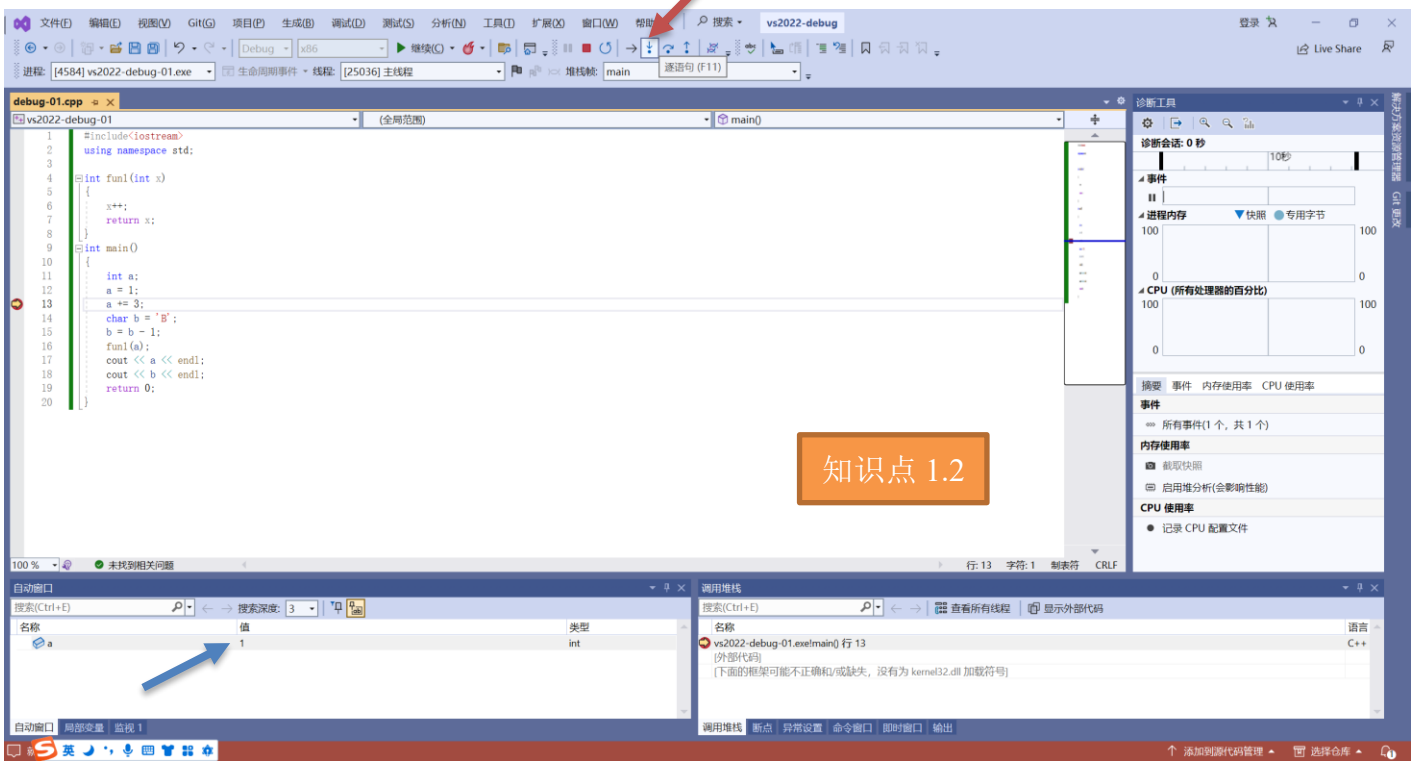
## 1.1.2. 如何结束调试？

在1.1.1的基础上，可以选择点击“停止调试”或者直接按 Shift+F5 组合键。
或者继续调试（点击绿色三角“继续”）直至调试结束。

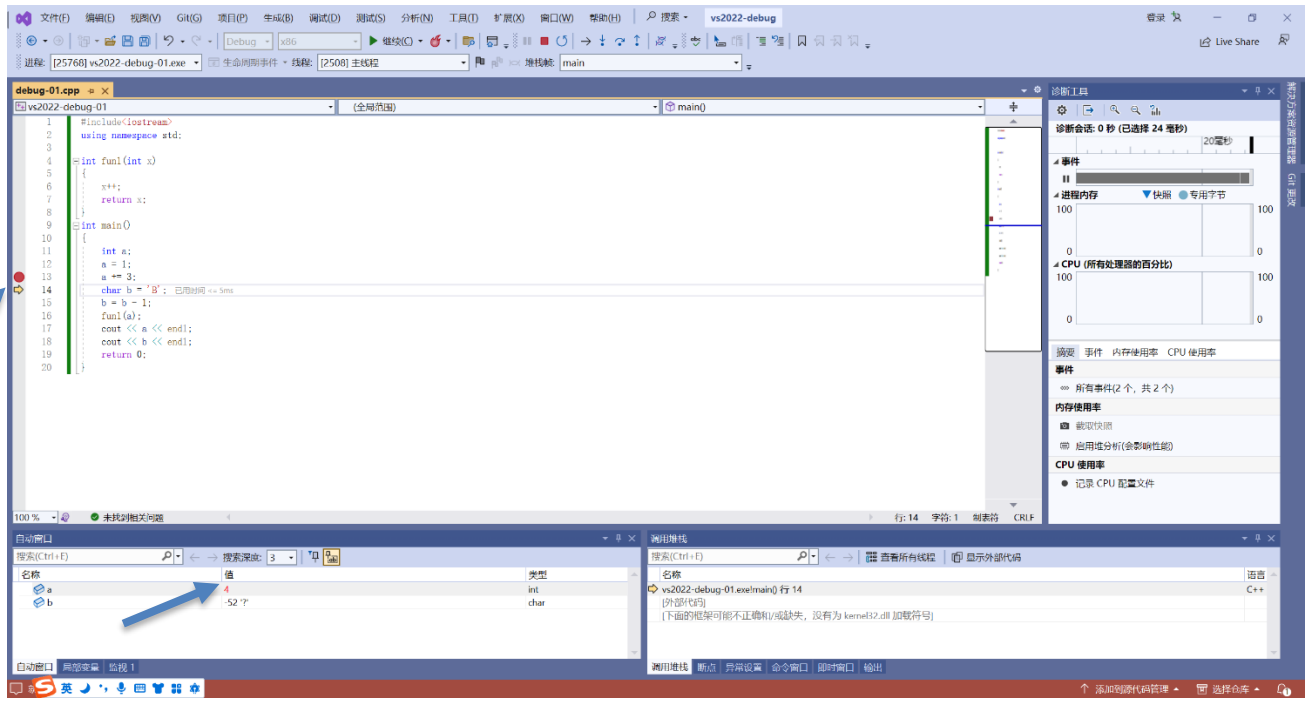



## 1.2. 如何在一个函数中每个语句单步执行？

在开始调试后，点击“逐语句”，或者按 F11 键，或者“调试”—“逐语句”。

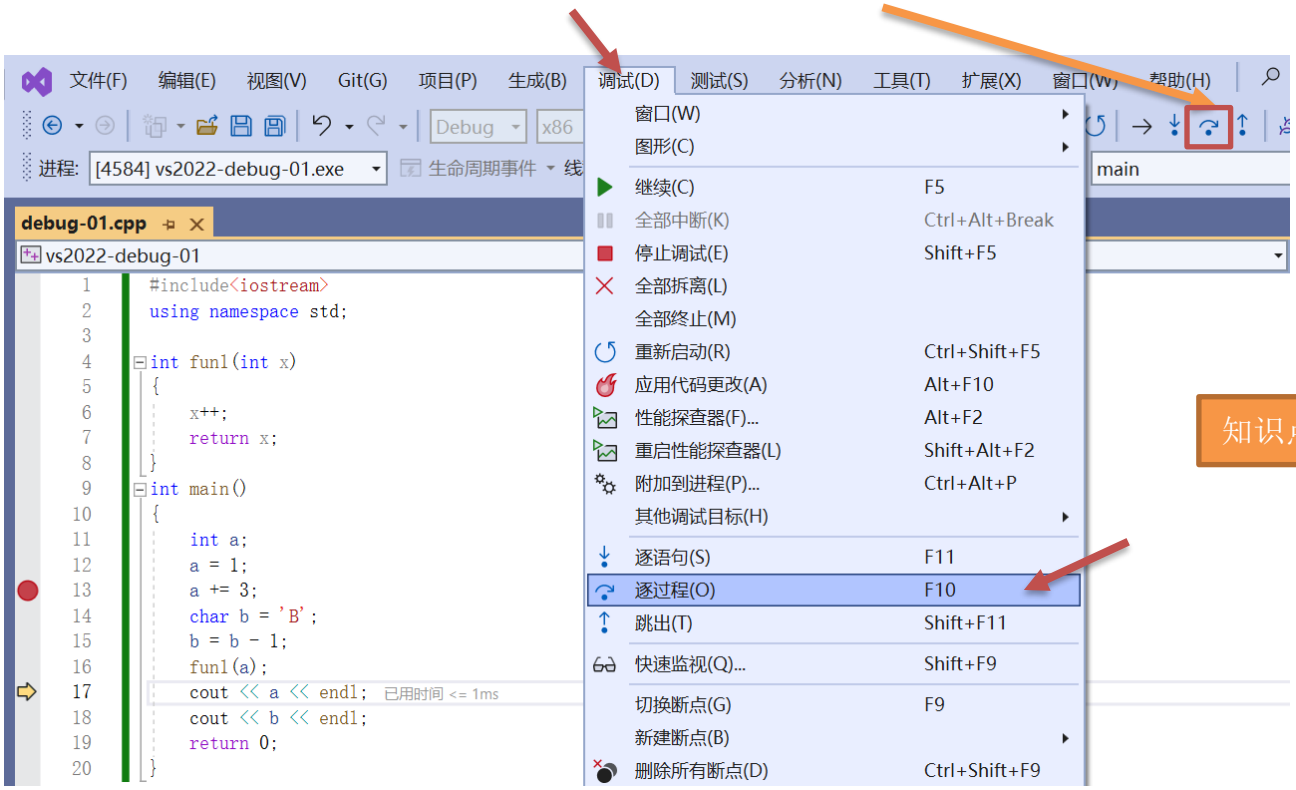


可看出左侧箭头下移，本例中自动窗口中a的值由1变为4.



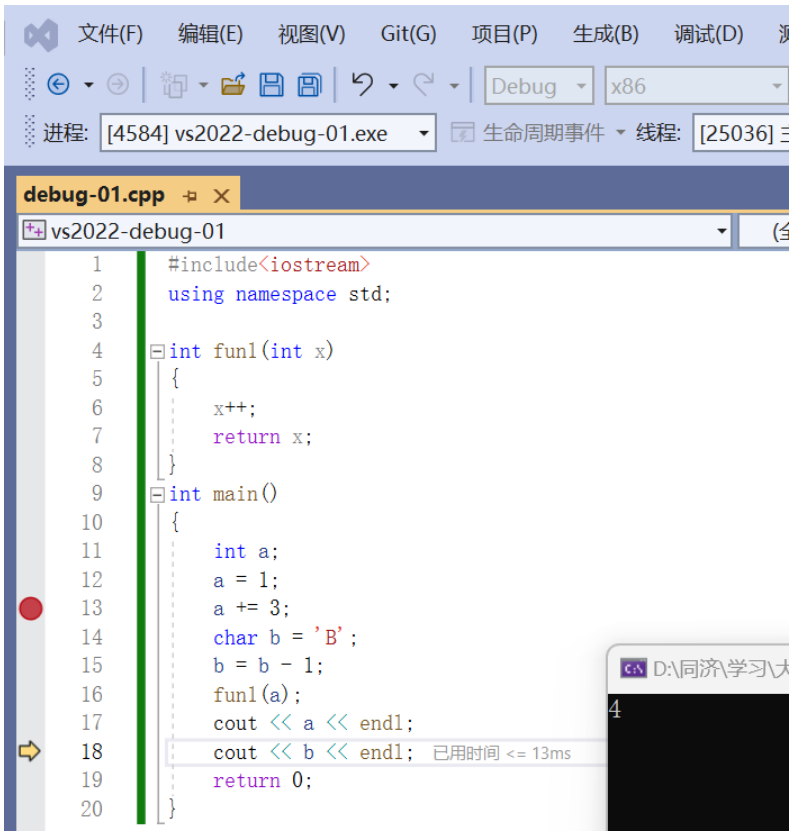
## 1.3. 在碰到 cout/sqrt 等系统类/系统函数时，如何一步完成这些系统类/系统函数的执行而不要进入到这些系统类/函数的内部单步执行?

调试过程中，点击“调试”—“逐过程”，或按 F10 键，或点击此按钮。



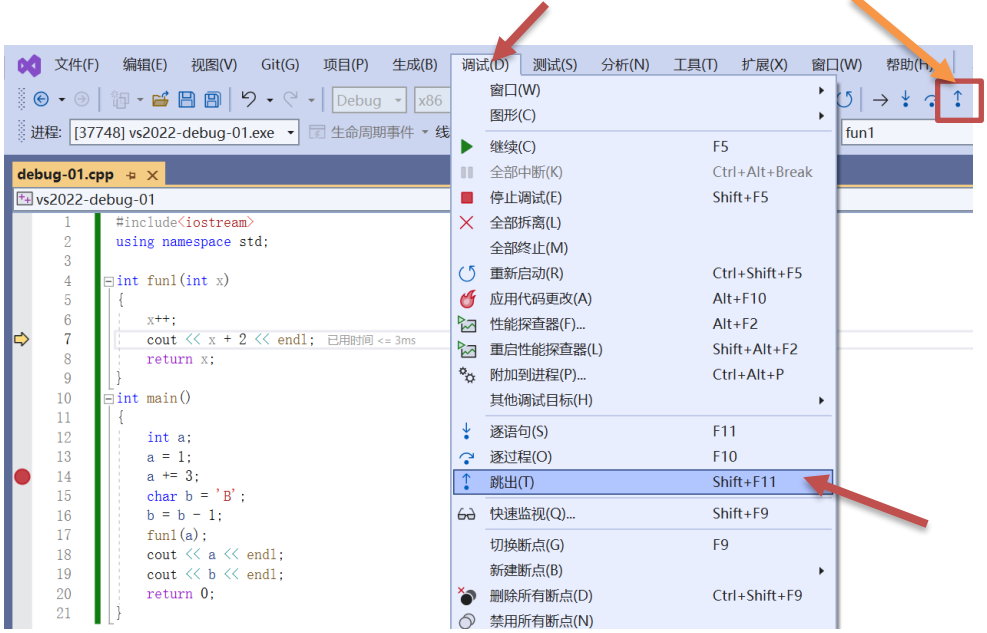

知识点 1.3

本例中结果显示为：输出了a的值，4.




## 1.4. 如果已经进入到 cout/sqrt 等系统类/系统函数的内部，如何跳出并返回自己的函数?

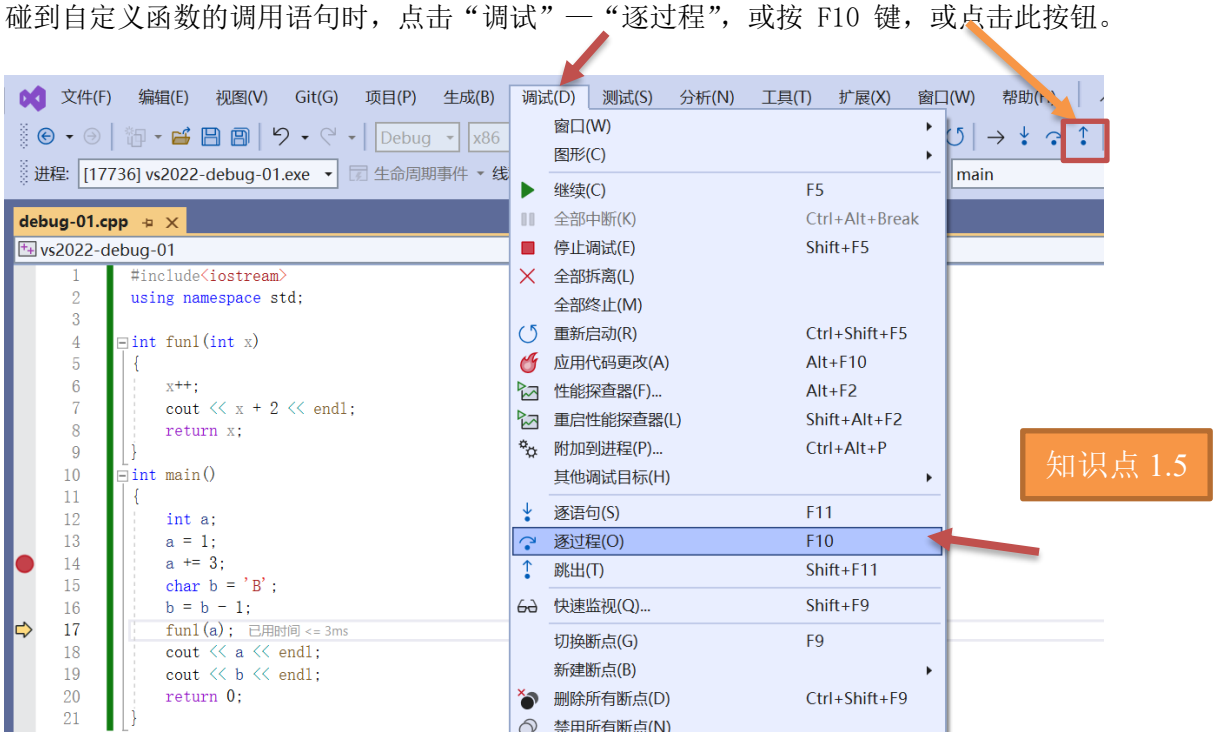
调试过程中，点击“调试”—“跳出”，或按 Shift+F11 键，或点击此按钮。




知识点 1.4

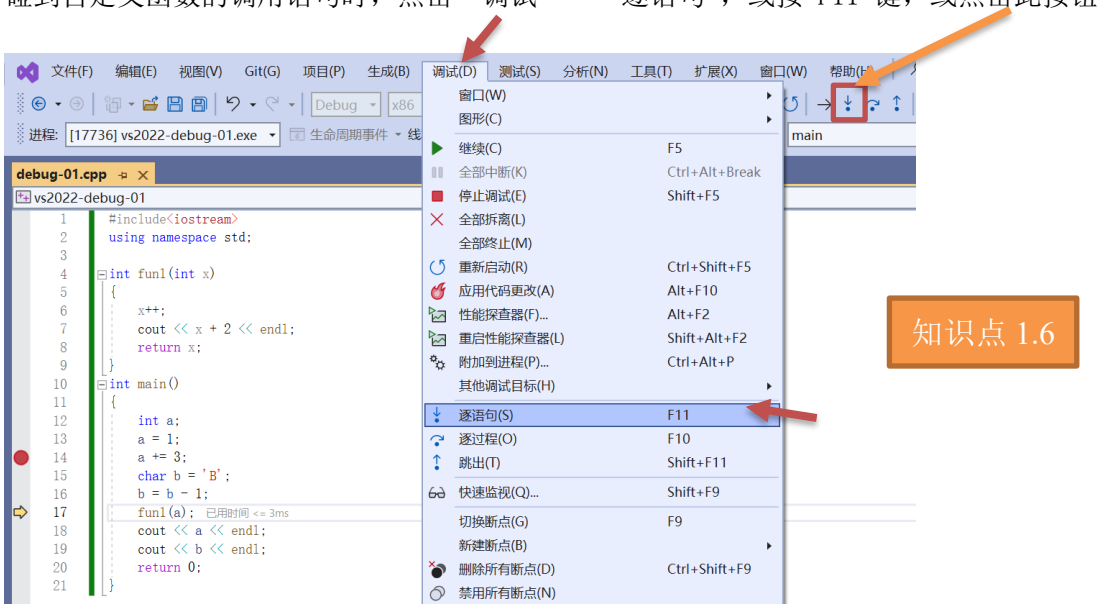
1.5. 在碰到自定义函数的调用语句（例如在 main 中调用自定义的 fun 函数）时，如何一步完成自定义函数的执行而不要进入到这些自定义函数的内部单步执行?

碰到自定义函数的调用语句时，点击“调试”—“逐过程”，或按 F10 键，或点击此按钮。




1.6. 在碰到自定义函数的调用语句（例如在 main 中调用自定义的 fun 函数）时，如何转到被调用函数中单步执行?

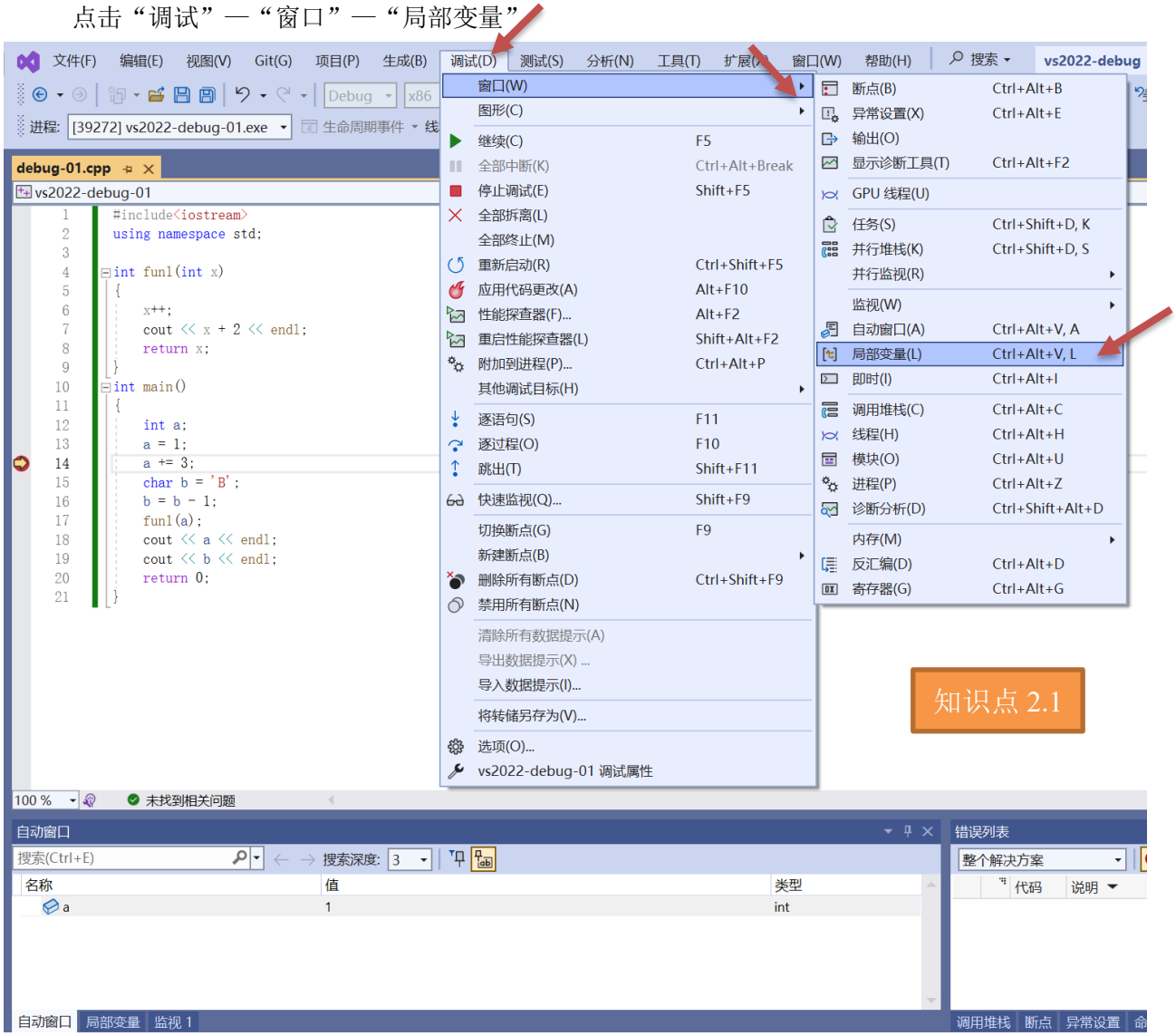
碰到自定义函数的调用语句时，点击“调试”—“逐语句”，或按 F11 键，或点击此按钮。



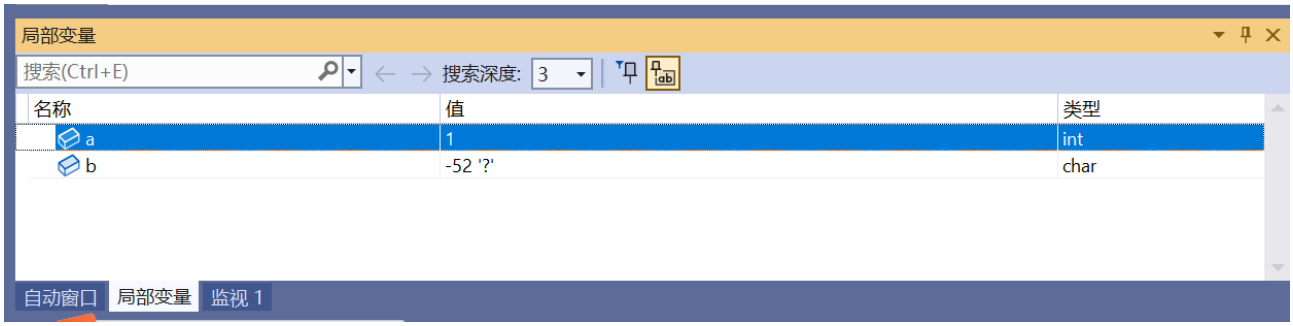
# 2. 掌握用 VS2022 的调试工具查看各种生存期/作用域变量的方法整体

## 2.1. 查看形参/自动变量的变化情况

点击“调试”—“窗口”—“局部变量”

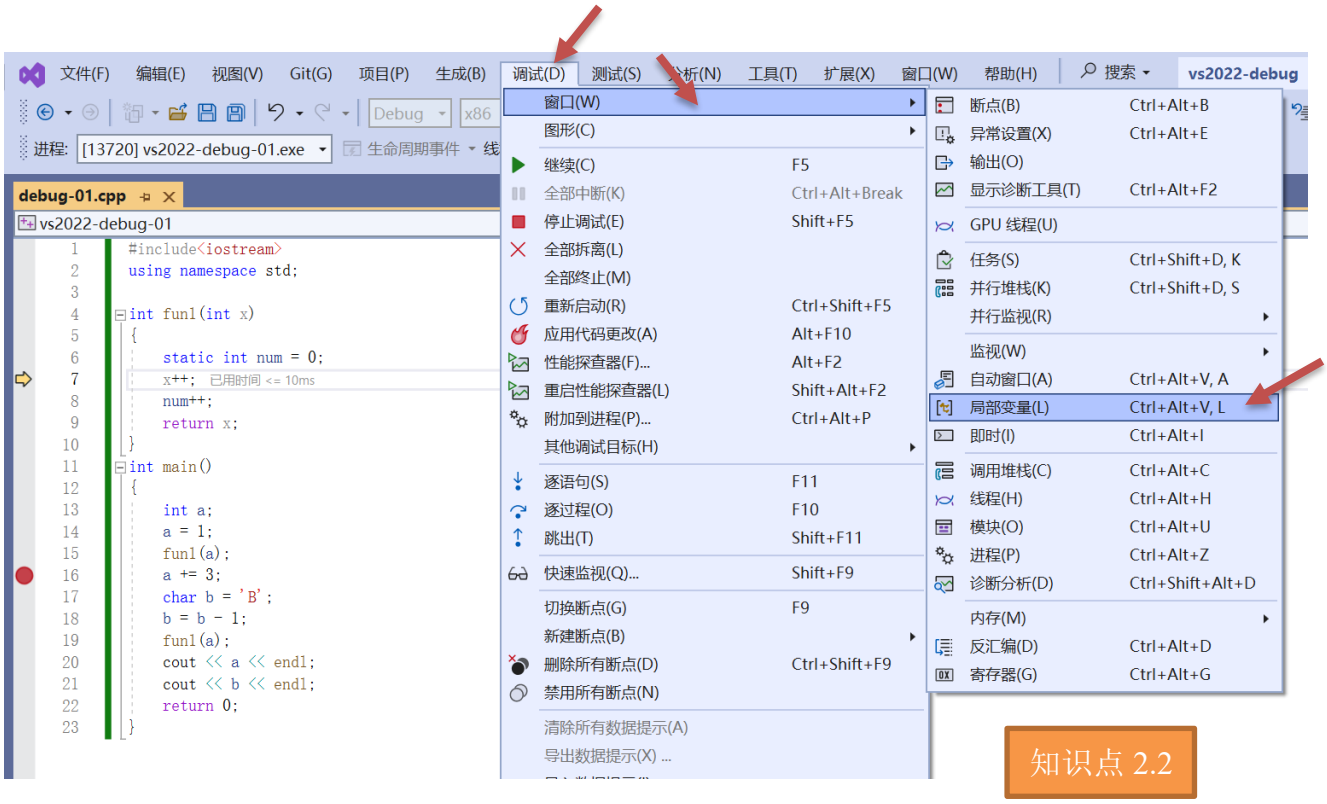



页面左下角显示局部变量，从而可以查看形参/自动变量的情况。

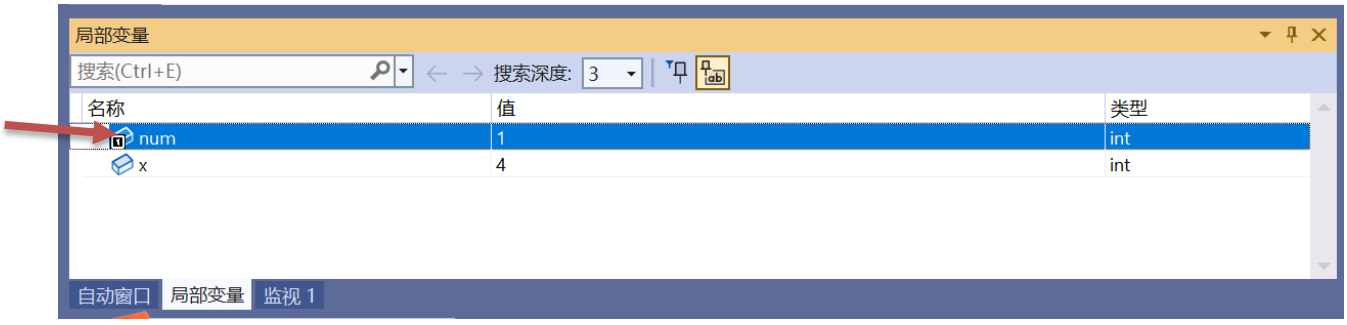


## 2.2. 查看静态局部变量的变化情况(该静态局部变量所在的函数体内/函数体外)

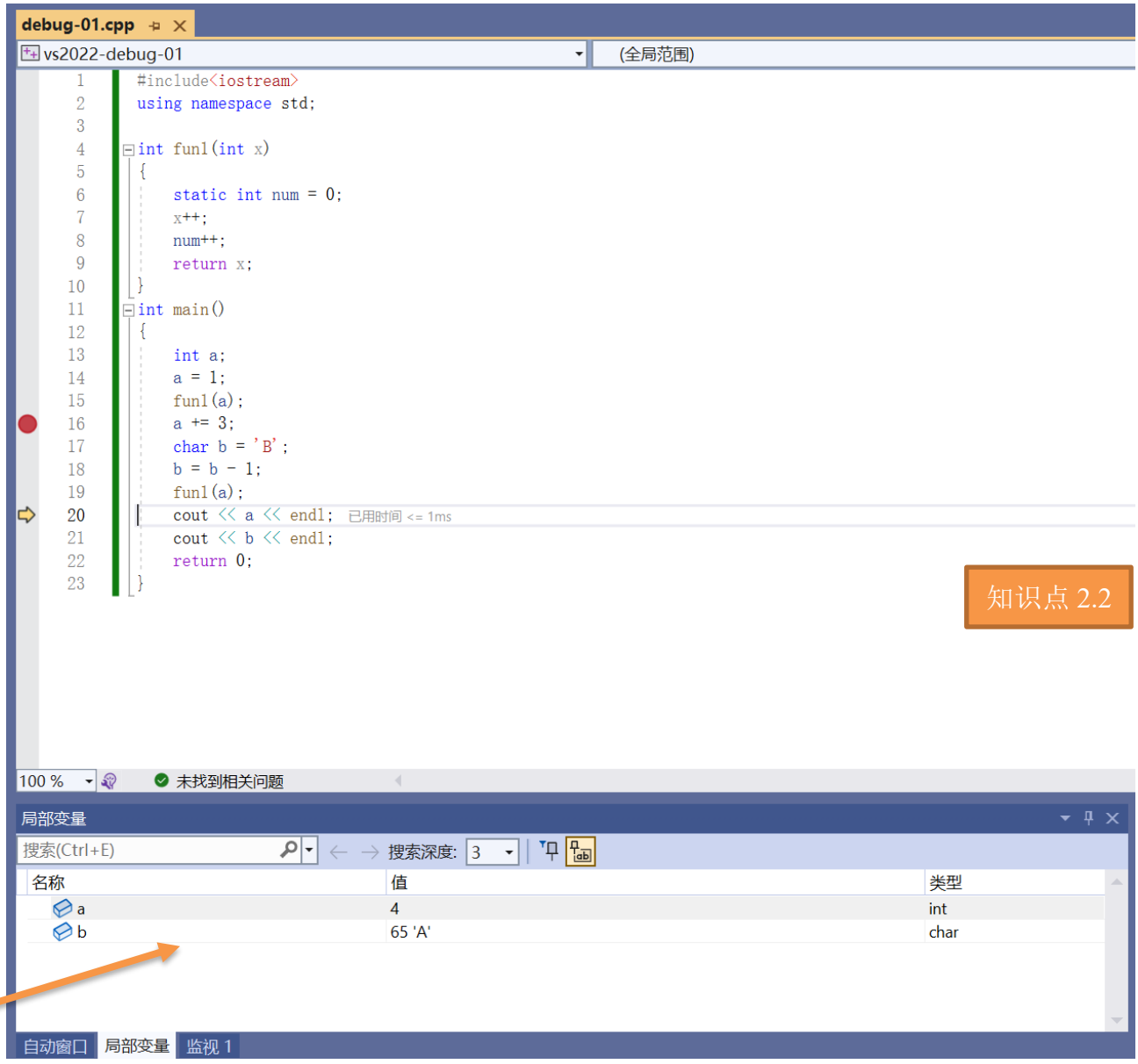
当静态局部变量在函数体内时，点击“调试”—“窗口”—“局部变量”。




则可以在左下角的的“局部变量”处查看静态局部变量在函数体内的变化。

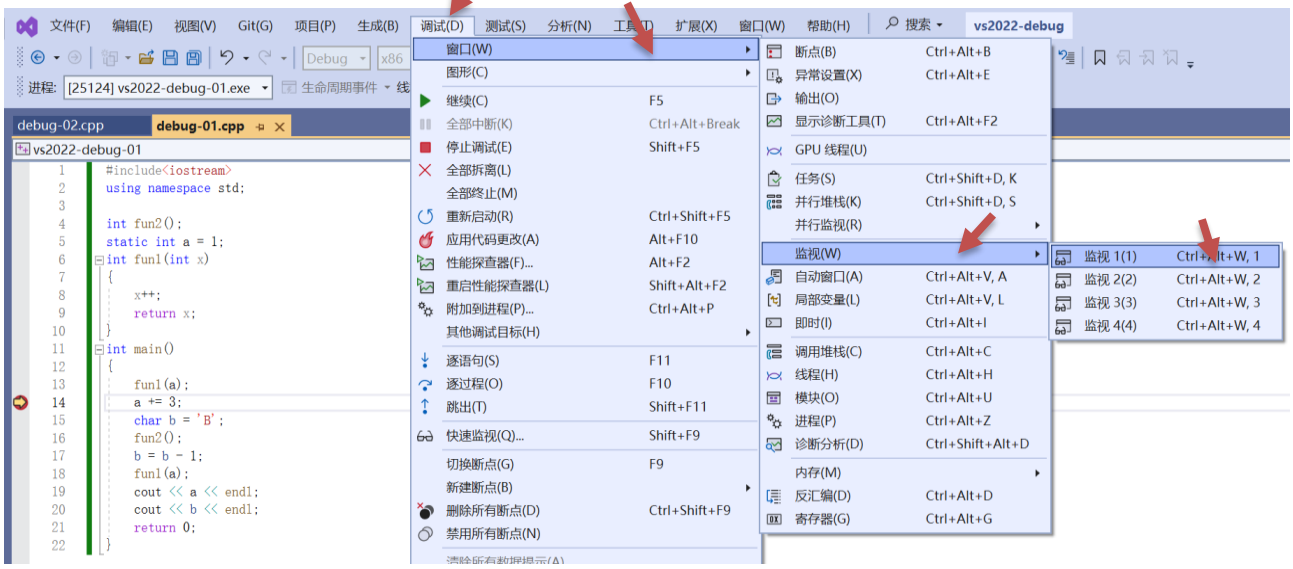



而静态局部变量在函数体外无法查看其变化情况（例如：此处在fun1函数外无法查看num）。

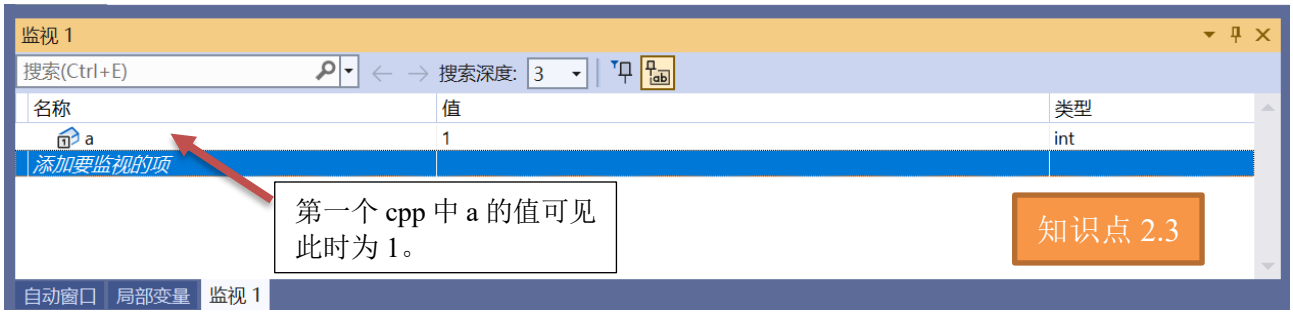



## 2.3. 查看静态全局变量的变化情况(两个源程序文件，有静态全局变量同名)

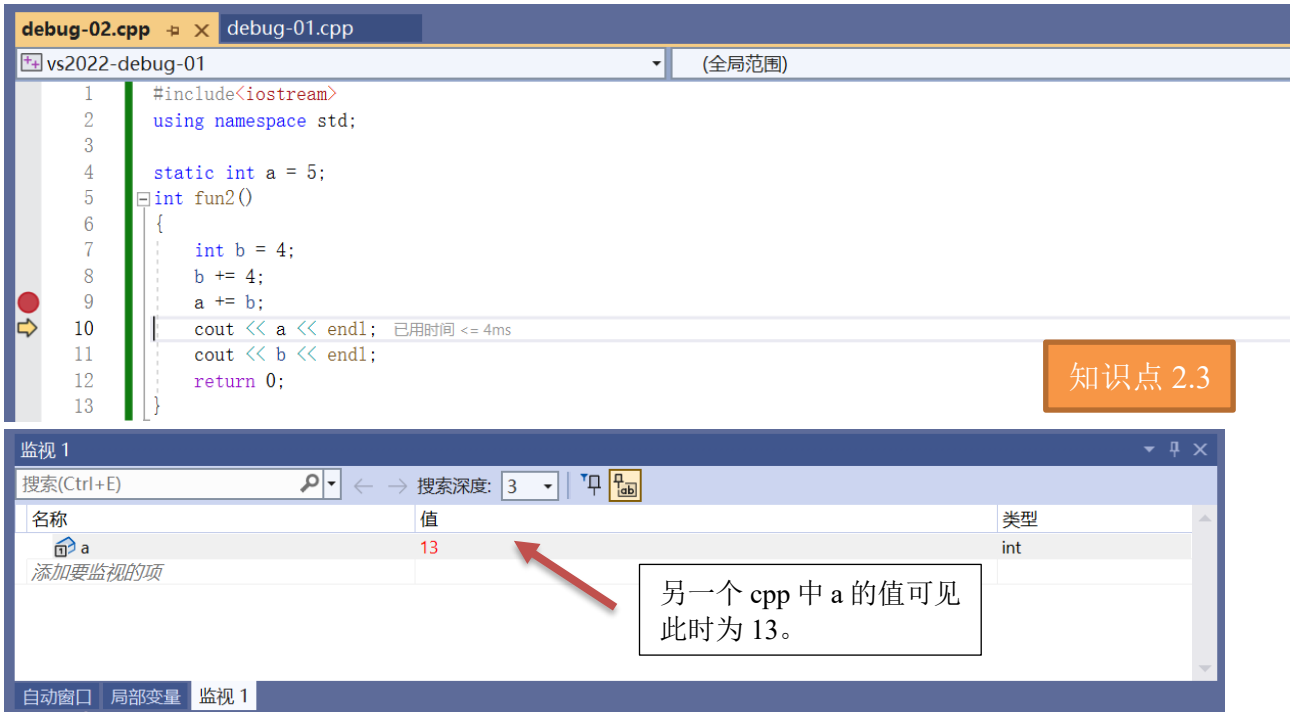
查看静态全局变量时，点击“调试”—“窗口”—“监视”—“监视1”。




在左下角的监视1窗口中输入需要监视的静态全局变量（此处为a），并按回车键。

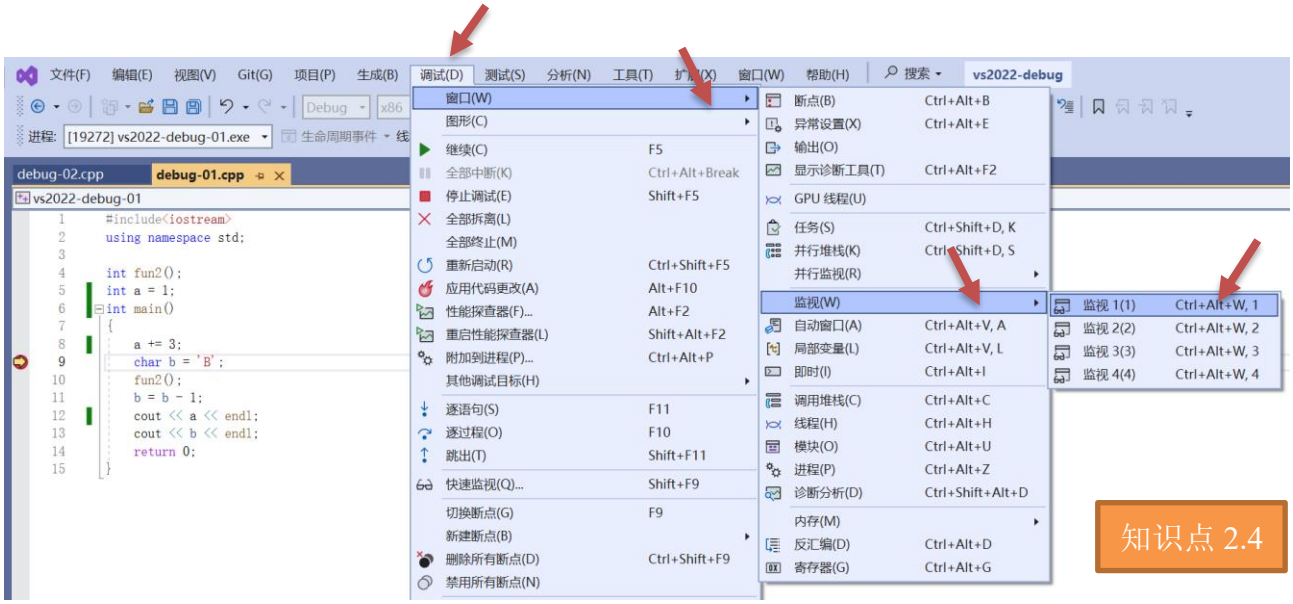



进入fun2函数后，在另一个cpp中的静态全局变量（变量名同样为a）也可以通过监视1的a来查看。

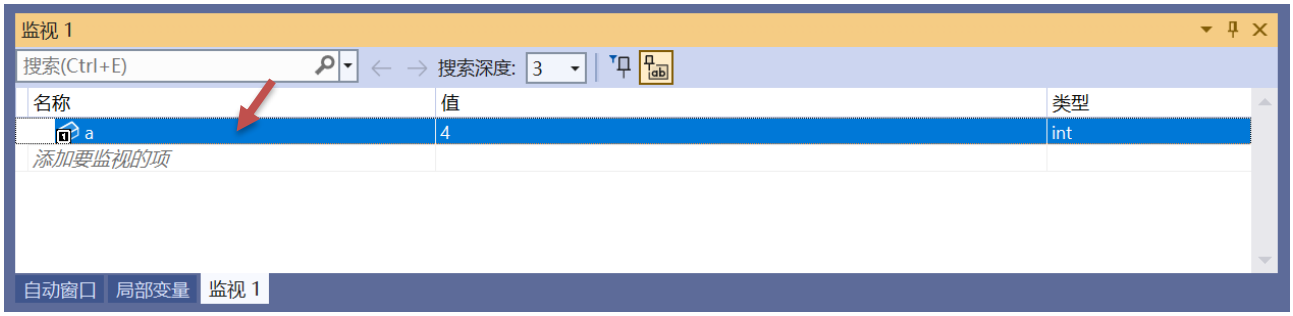


## 2.4. 查看外部全局变量的变化情况(两个源程序文件，一个定义，另一个有 extern 说明)

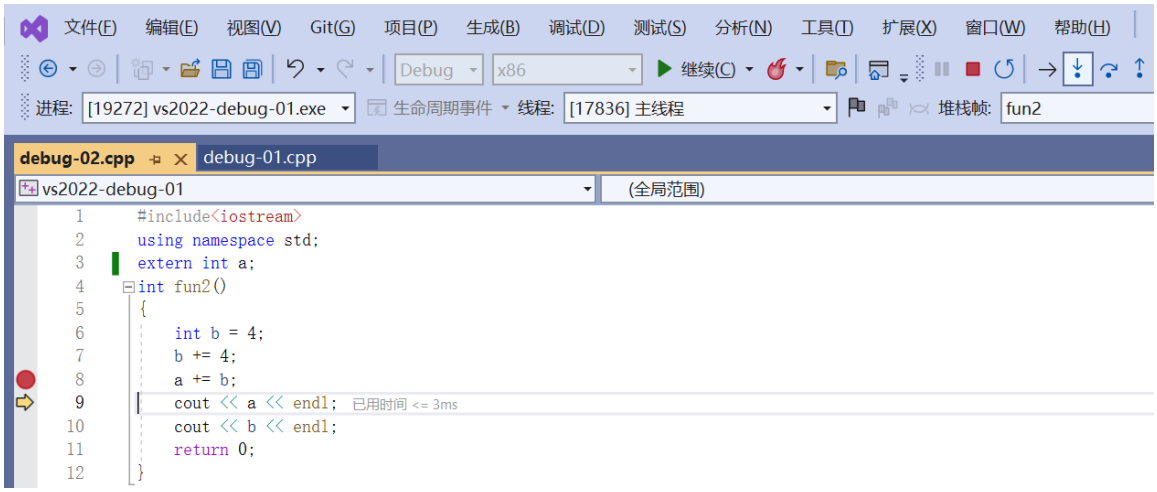
查看外部全局变量时，点击“调试”—“窗口”—“监视”—“监视1”。



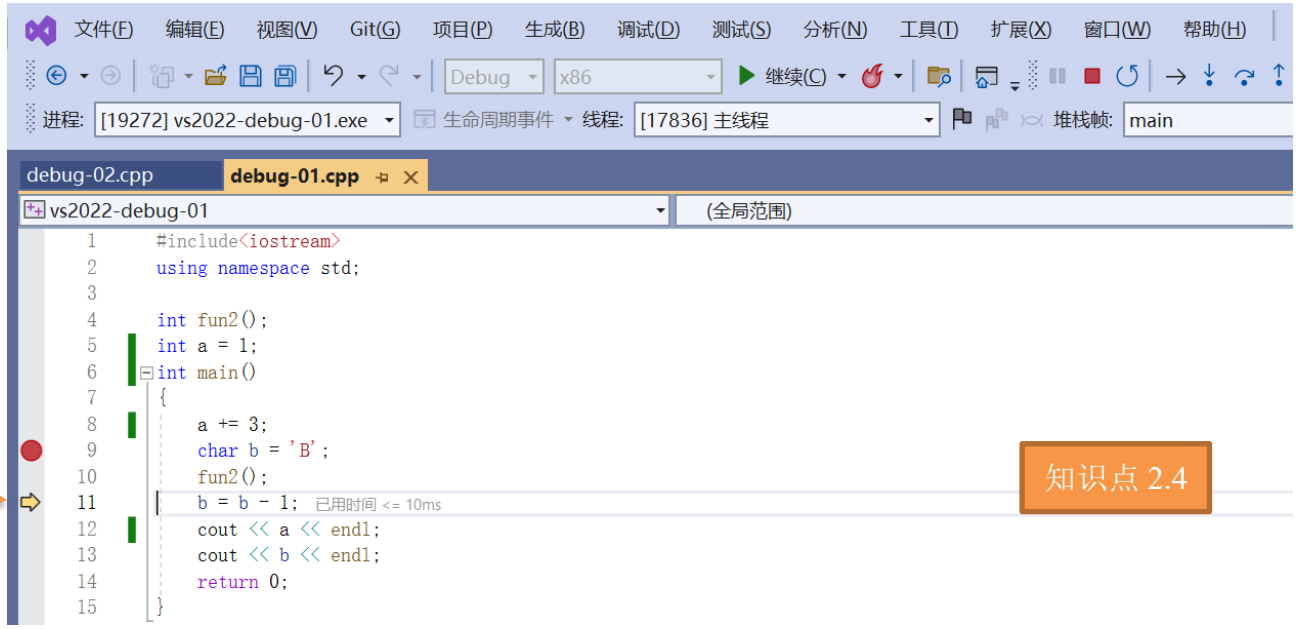

在左下角的监视1窗口中输入需要监视的外部全局变量（此处为a），并按回车键。




在另一个有extern说明的cpp中a值改变：




fun2函数运行结束，回到原本的cpp中，a值依旧为fun2改变后的值：





## 3. 掌握用 VS2022 的调试工具查看各种不同类型变量的方法

### 3.1. char/int/float 等简单变量

局部变量（包括静态局部变量、形参、自动变量）通过点击“调试”—“窗口”—“局部变量”查看。

debug-02.cpp | debug-01.cpp

vs2022-debug-01 (全局范围)

```
1   #include<iostream>
2   using namespace std;
3   int fun2();
4   int a = 1;
5   char b = 'B';
6   float c = 3.5;
7   static int d = 2;
8   static char e = 'E';
9   static float f = 4.5;
10  void fun1(int k)
11  {
12      k = 0;
13      static int num = 1;
14      num++;
15      k++;
16  }
17  int main()
18  {
19      int g = 3;
20      char h = 'H';
21      float i = 6.0;
22      a++;  已用时间 <= 3ms
23      fun1(a);
24      fun2();
25      d++;
26      e++;
27      f++;
28      a = a * 10;
29      b = b + 5;
30      c = c + 1;
31      return 0;
32  }
```

知识点 3.1

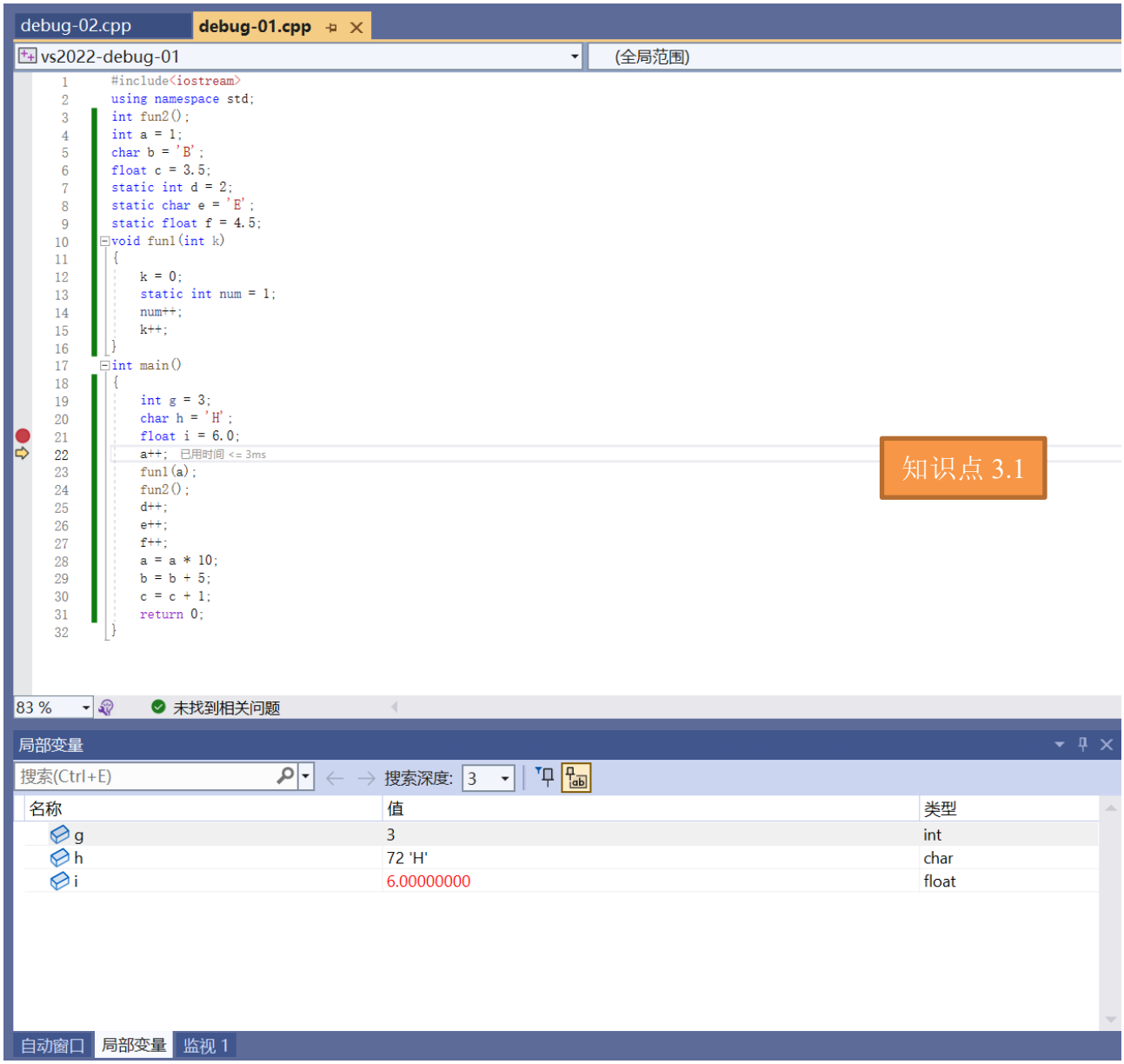
83 % 未找到相关问题

局部变量

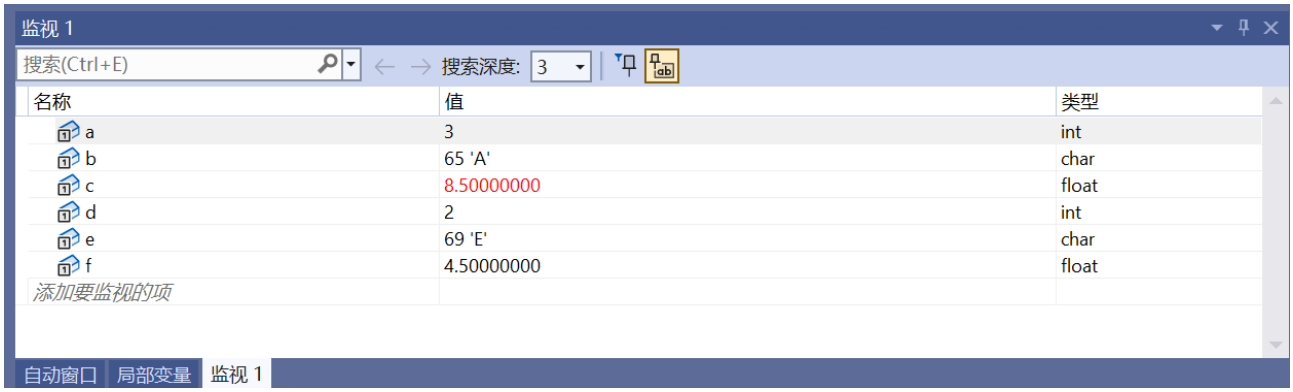
搜索(Ctrl+E) 搜索深度: 3

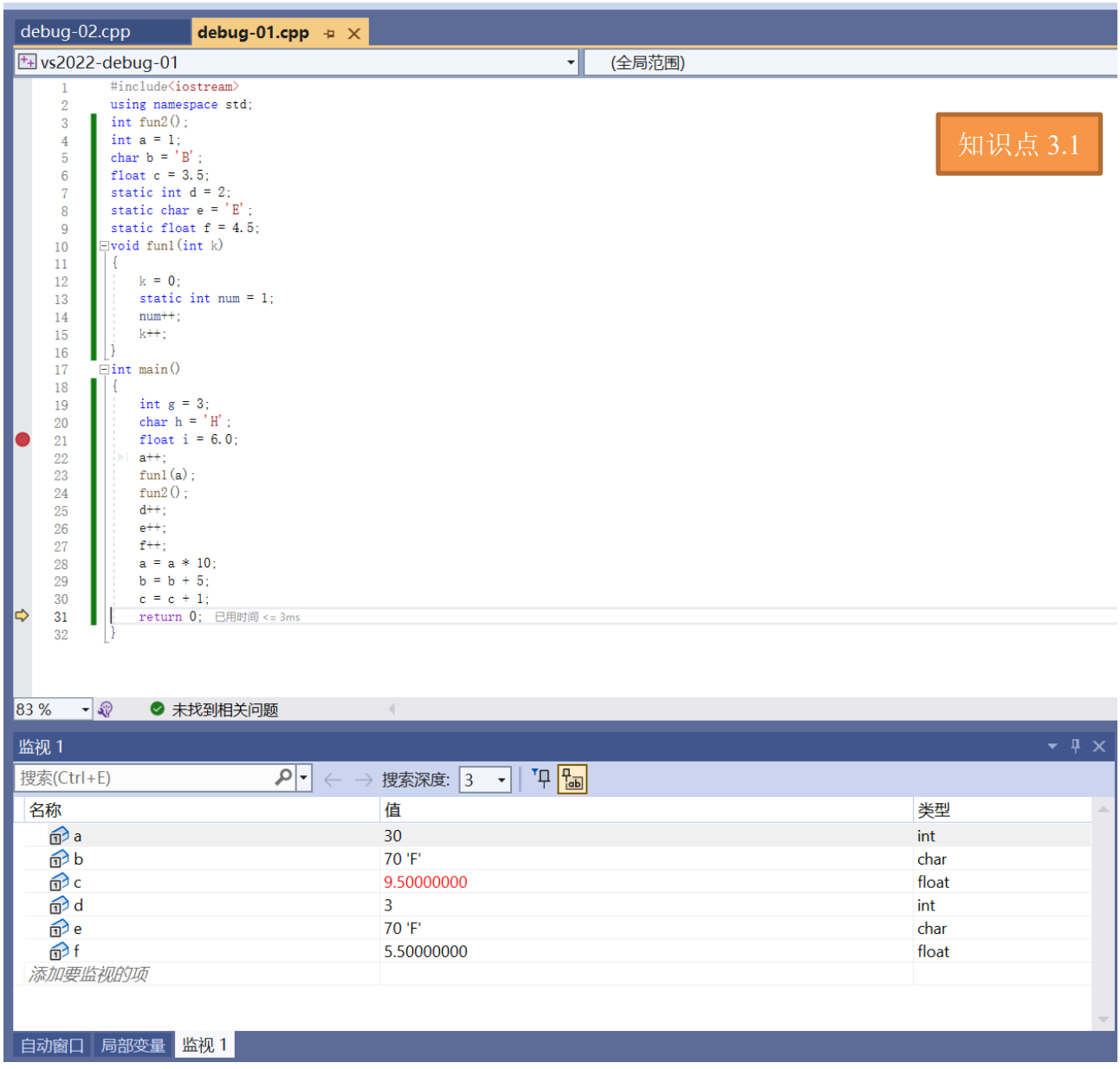
| 名称 | 值 | 类型 |
|---|---|---|
| g | 3 | int |
| h | 72 'H' | char |
| i | 6.00000000 | float |

自动窗口 局部变量 监视 1



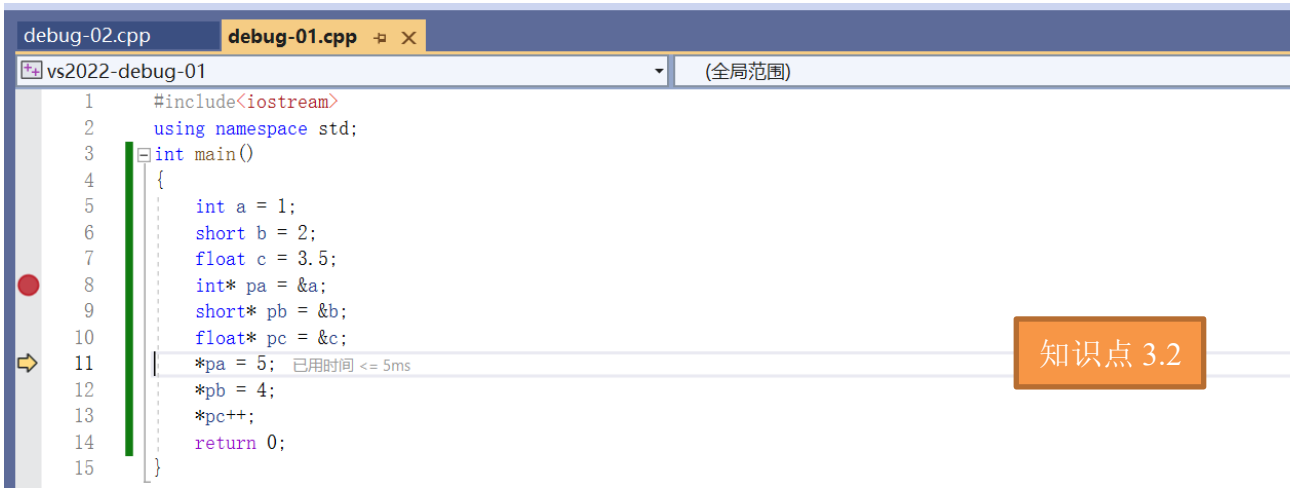

全局变量（包括静态全局变量、外部全局变量），通过点击点击“调试”—“窗口”—“监视”—“监视1”来查看。

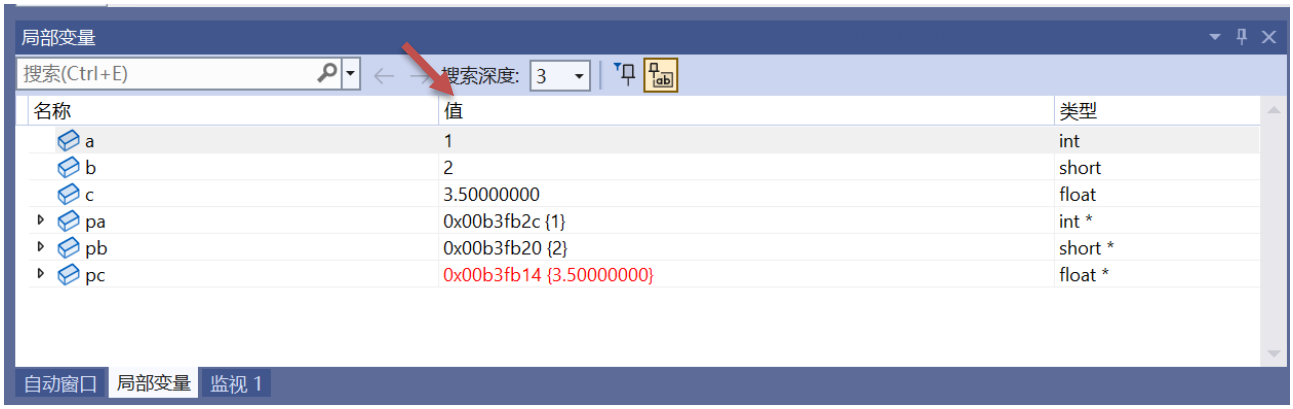



## 3.2. 指向简单变量的指针变量(如何查看地址、值？)

指向简单变量的指针变量，可以通过左下窗口的“值”来查看地址和值，其中地址在前，后面的{}内为值。



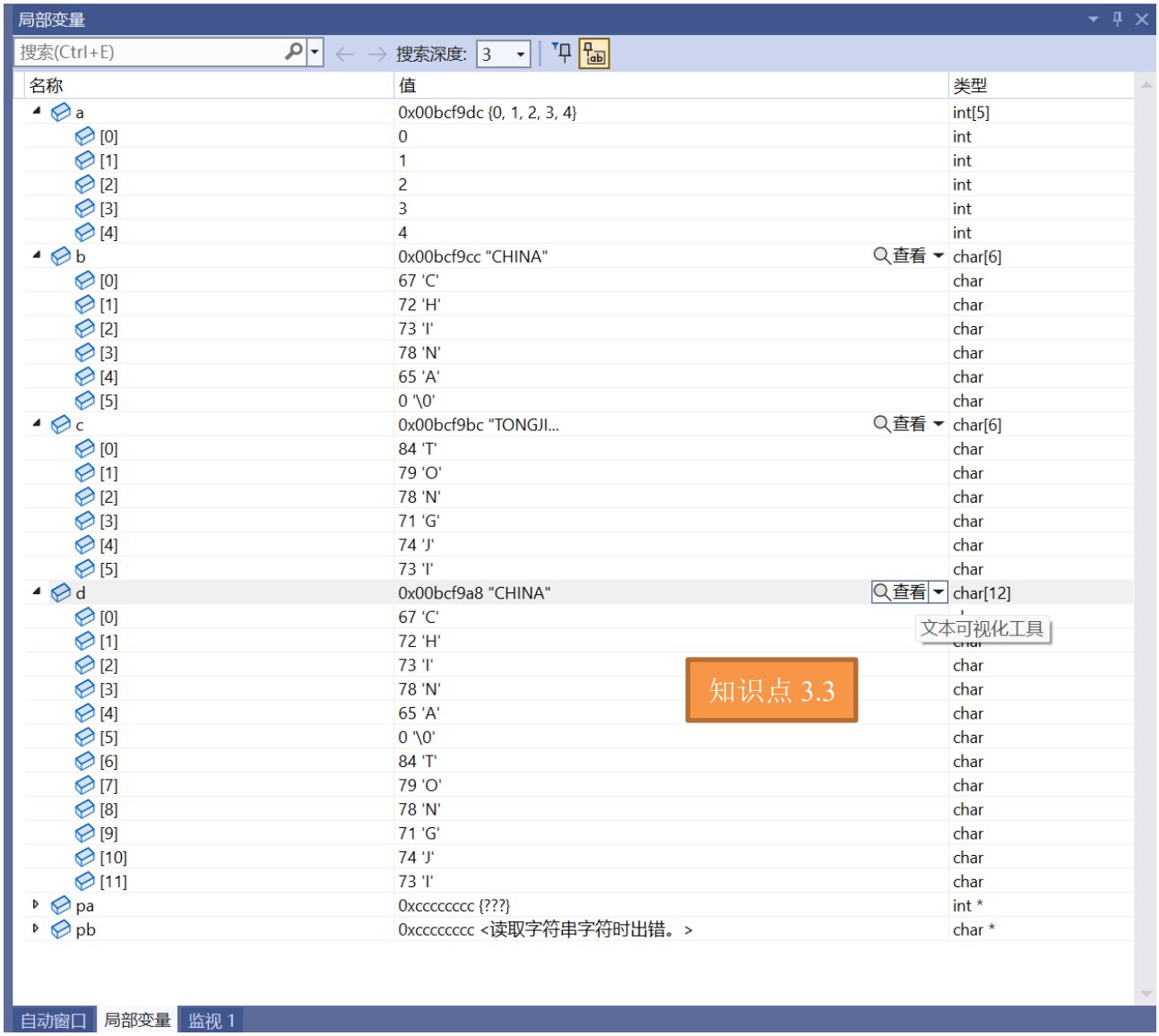

## 3.3. 一维数组

在左下角窗口可以查看一维数组，若为局部变量，则在局部变量查看，若为全局数组，则在监视1中添加后查看。

可以在数组名称的右侧看到数组的地址以及数组的内容，int型数组则用“{}”来描述其中的数值；char型数组用字符串来描述数组里的内容，无尾零则用“...”显示。

可以通过点击数组名称左侧的三角来查看数组中各元素所对应的值。

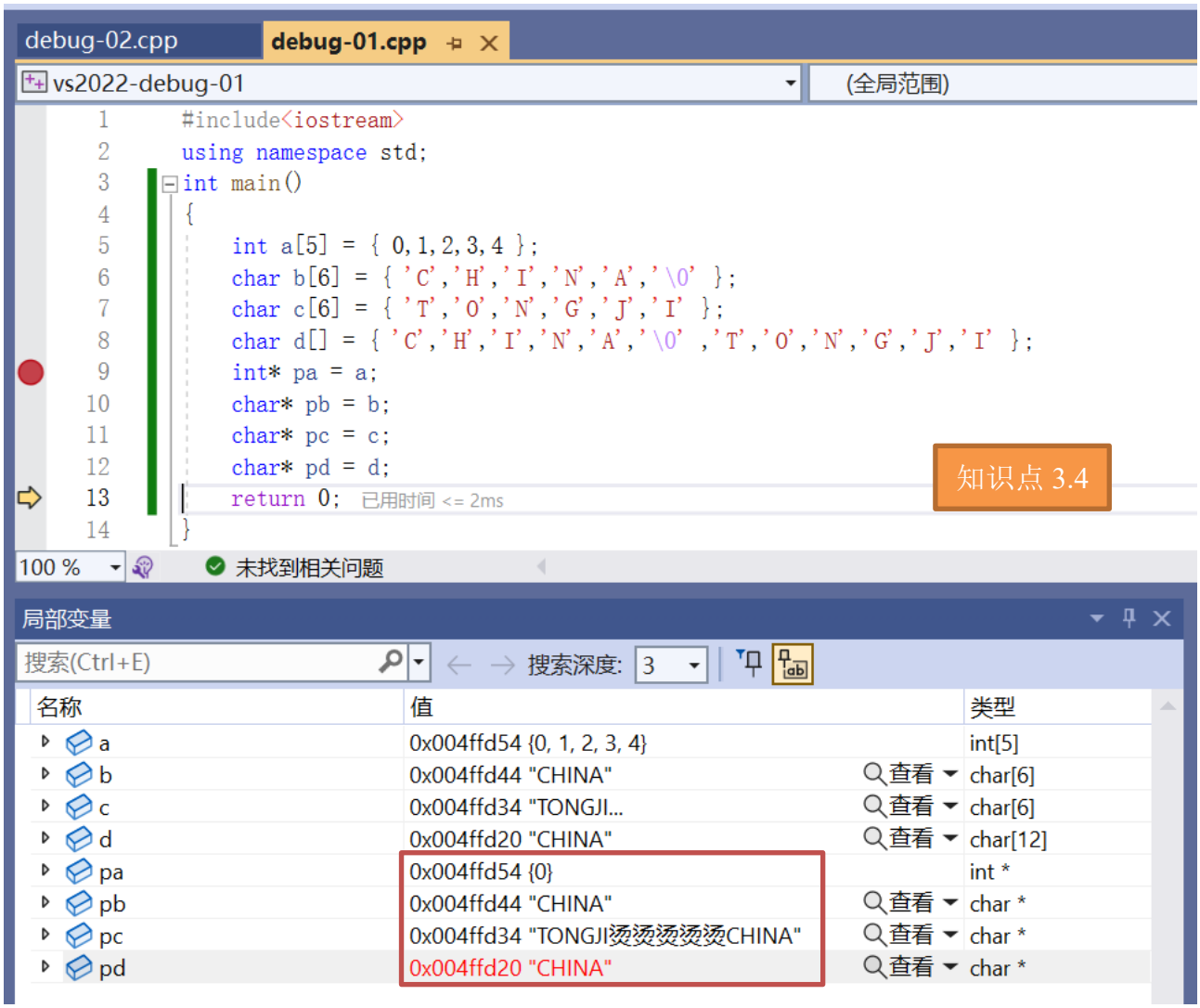
```
#include<iostream>
using namespace std;
int main()
{
    int a[5] = { 0,1,2,3,4 };
    char b[6] = { 'C','H','I','N','A','\0' };
    char c[6] = { 'T','O','N','G','J','I' };
    char d[] = { 'C','H','I','N','A','\0' ,'T','O','N','G','J','I' };
    int* pa = a;
    char* pb = b;
    return 0;
}
```




## 3.4. 指向一维数组的指针变量(如何查看地址、值？)

在左下角窗口可以查看一维数组的指针变量，若为局部变量，则在局部变量查看，若为全局变量，则在监视1中添加后查看。

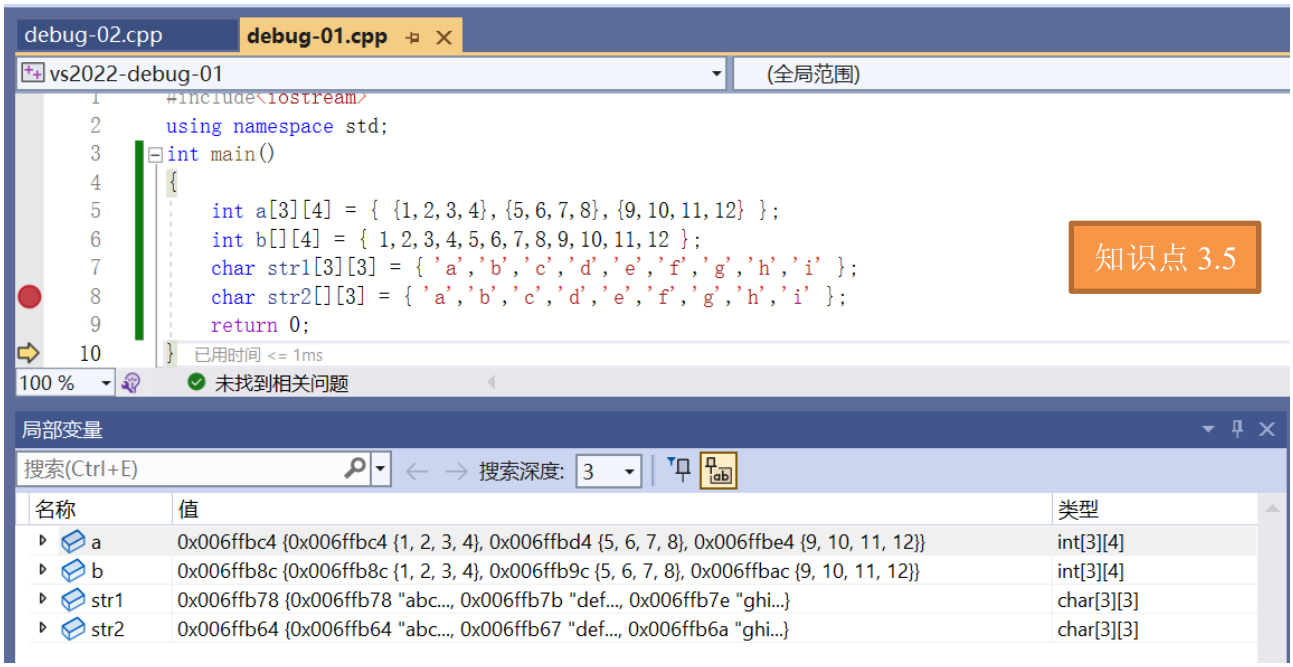
一维数组指针变量的“值”的一栏先显示数组的首地址，若为字符型数组，则显示字符串，若为整型数组，则显示第0号元素的值，但都不显示各个数组元素的地址。




## 3.5. 二维数组(包括数组名仅带一个下标的情况)

在左下角窗口可以查看一维数组，若为局部变量，则在局部变量查看，若为全局数组，则在监视1中添加后查看。

在“值”这一栏中可查看二维数组的信息，大括号前先显示二维数组的首地址，大括号内为各一维数组的首地址和一维数组中各元素的值。(以a[3][4]为例，大括号外先显示二维数组a的首地址，大括号内分别显示一维数组a[0],a[1],a[2]的首地址，并用大括号来显示每个一维数组中的值)，若为字符型数组，则用字符串来显示数组中的值。



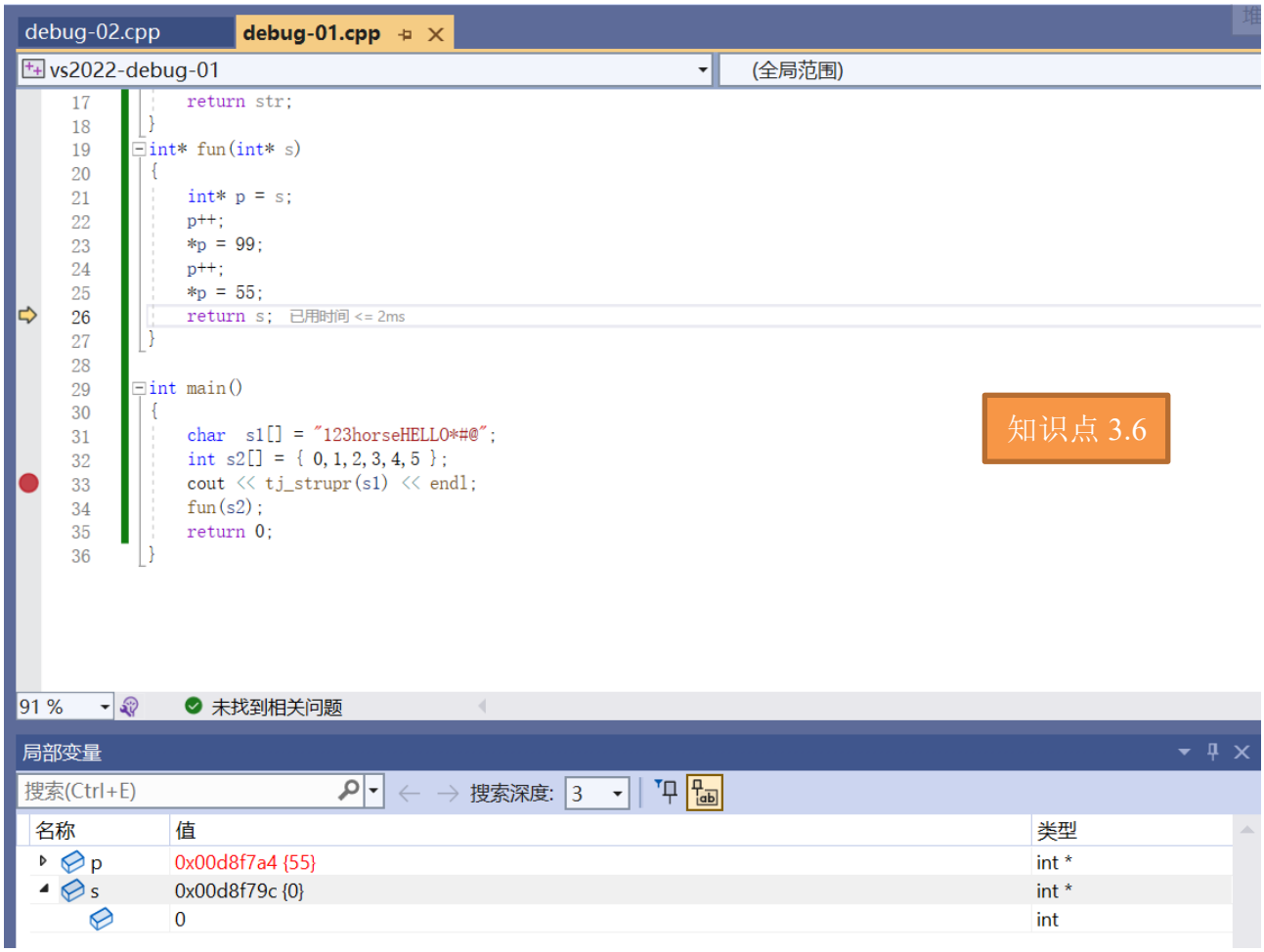

知识点 3.5

## 3.6. 实参是一维数组名，形参是指针的情况，如何在函数中查看实参数组的地址、值?

对于字符型一维数组，可以通过查看形参指针的地址和值来查看实参数组的地址和值。


知识点 3.6


对于整型一维数组，无法在函数内查看实参数组的地址、值，只有函数结束运行时才能查看其地址和值。

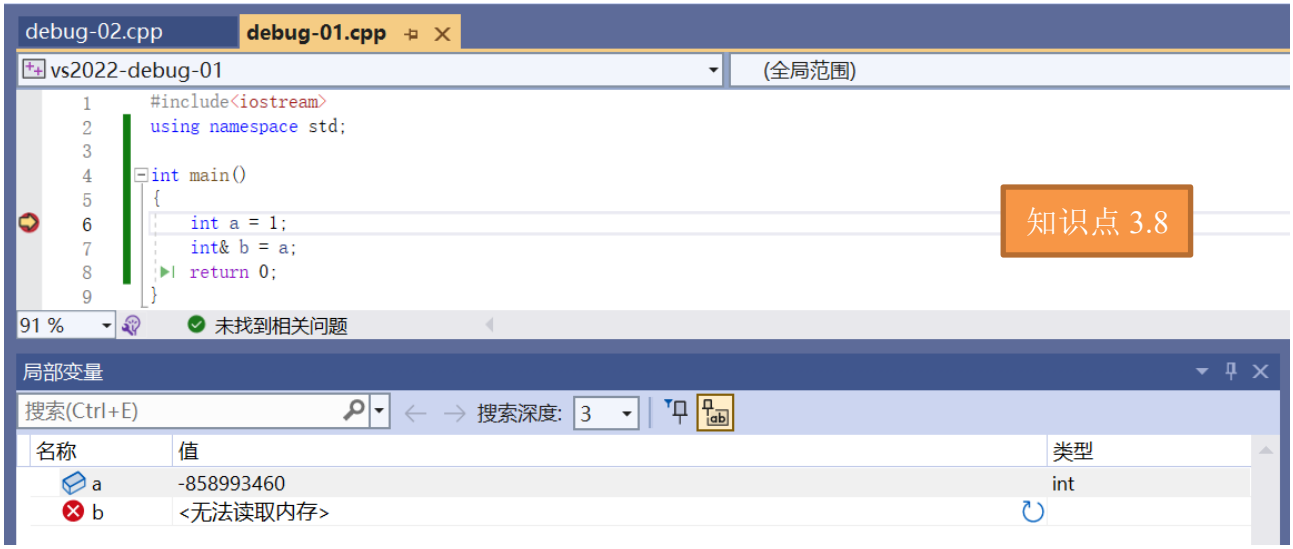



## 3.7. 指向字符串常量的指针变量（能否看到无名字符串常量的地址？）

可以看到无名字符串常量的地址。


## 3.8. 引用（引用与指针是否有区别？有什么区别？）

引用与指针有区别。引用不分配单独的空间，而指针变量有单独的空间。




## 3.9. 使用指针时出现了越界访问

虽然没有报错，但指针所指的变量的值是随机的，不可信的。


debug-02.cpp
debug-01.cpp
vs2022-debug-01
(全局范围)
#include<iostream>
using namespace std;
int main()
{
int s[3] = { 0,1,2 };
int* p = s;
p = p + 4;
return 0;
已用时间 <= 2ms
}
91 %
未找到相关问题
知识点 3.9
局部变量
搜索(Ctrl+E)
搜索深度: 3
名称
值
类型
p
0x00bef8c8 {-1431003977}
int *
s
0x00bef8b8 {0, 1, 2}
int[3]