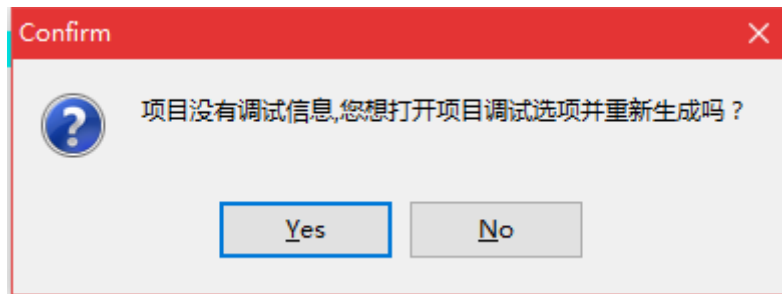


关于如何使用 DEV C++软件进行调试：

| 出错情况 |

1、 点击调试后弹出这样的窗口：

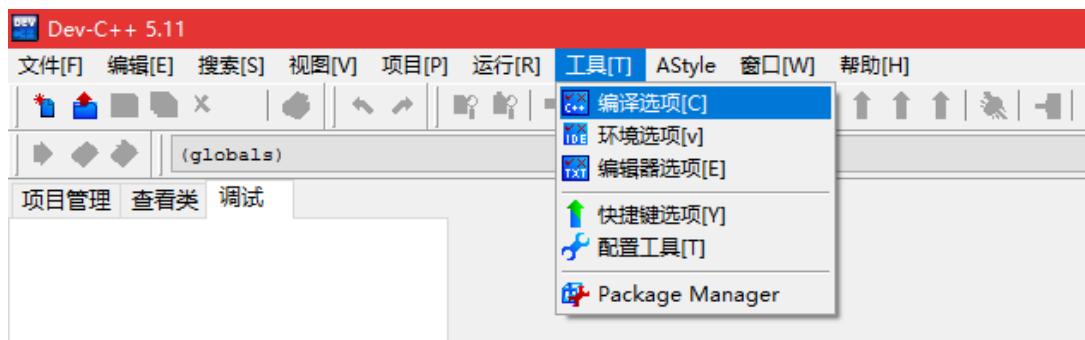


2、 点击 YES 后直接停止工作或者闪退。

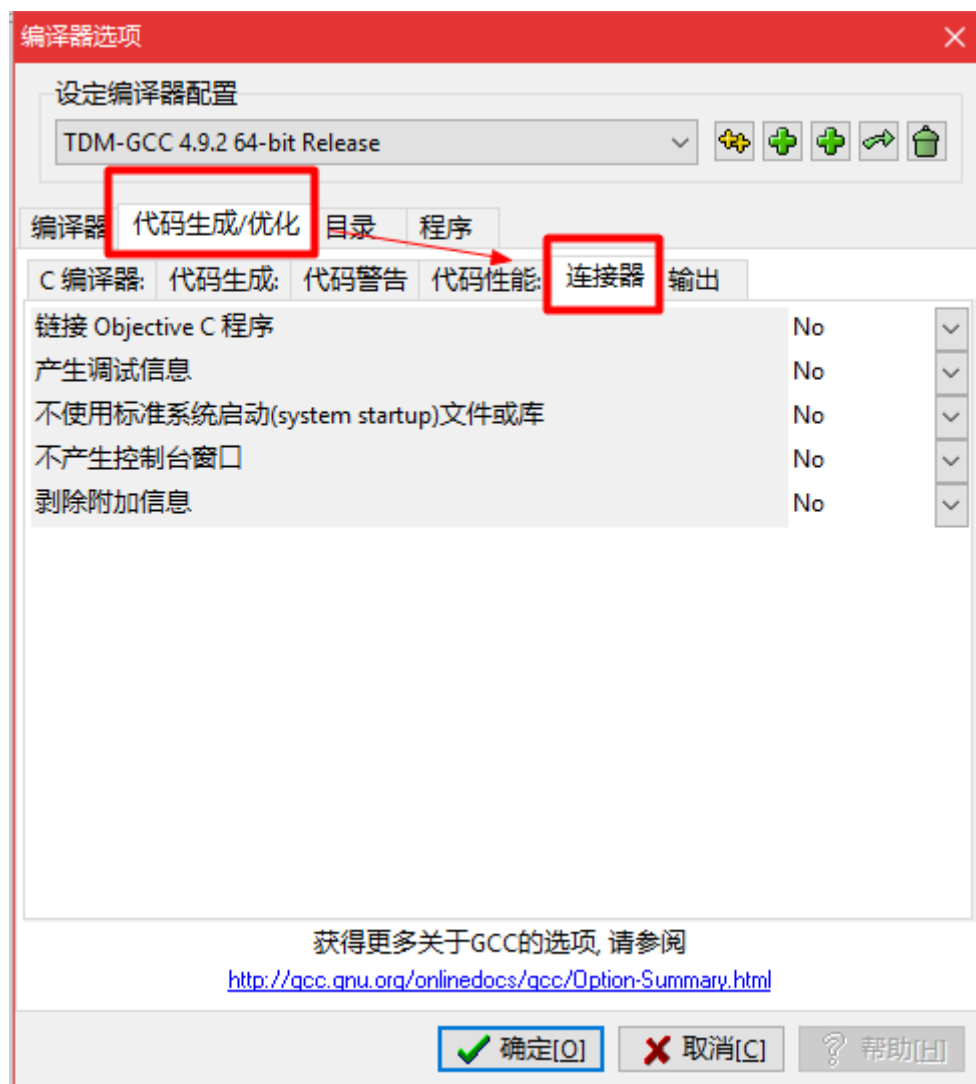
3、 使用的是先导课程上所给地址下载的 DEV C++ 并且没有对编译环境进行过改动（UI 改动、字体改动等除外）

| 解决方法 |

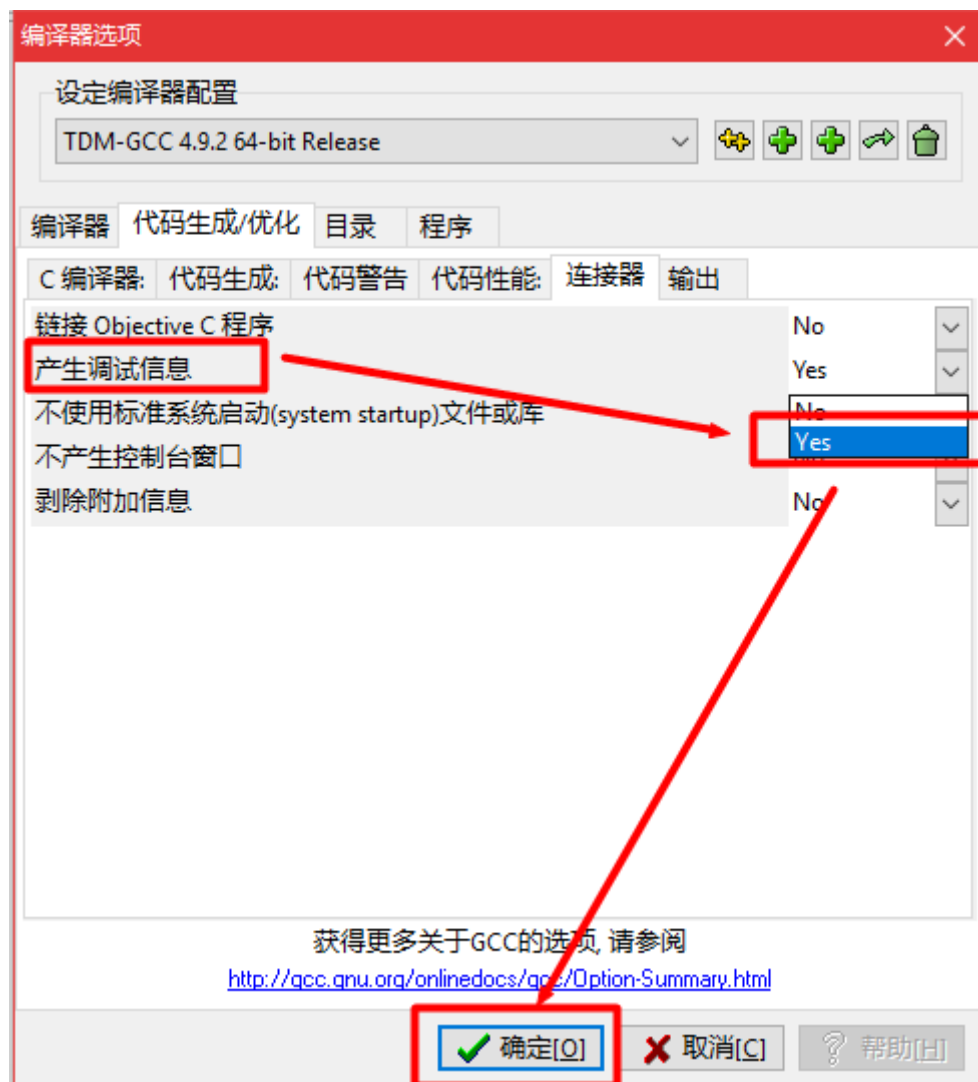
1、 打开 DEV C++软件，找到菜单栏中的工具-编译选项。



2、依次选择代码生成/优化-连接器。



3、将【产生调试信息】后面的按钮调为“yes”，然后确定。



4、可以正常调试了。

| 调试小技巧 |

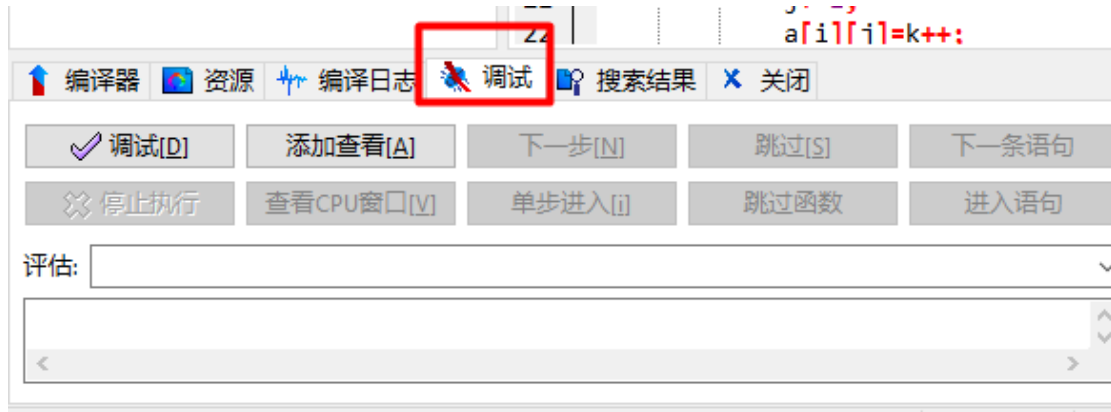
1、调试是一定要加断点的。最好的断点加入是在某一个循环上添加断点。

```
9 | while (n<=0 || n>N-2);  
10 | printf("\n");  
11 | i=j=k=1;  
12 | a[i][j]=k++;  
13 | while(k<=n*n){  
14 |     if(j==1&&i<n){  
15 |         i+=1;  
16 |         a[i][j]=k++;  
17 |     }
```


图为在 While 语句的开始添加断点

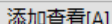
像这样添加断点是非常方便查找出问题的。

2、在调试的时候，建议在画面下方使用调试，如图：

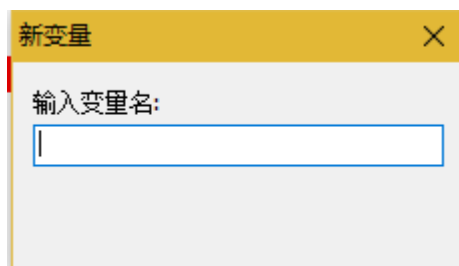


这样可以调出调试工作台（暂且这么称呼），方便操作。

点击按钮  即可开始调试。

3、**（重要）**在调试的时候，点击调试工作台上的按钮  可以添加你要查看的变量，具体步骤如下——

① 点击按钮，弹出一个窗口：



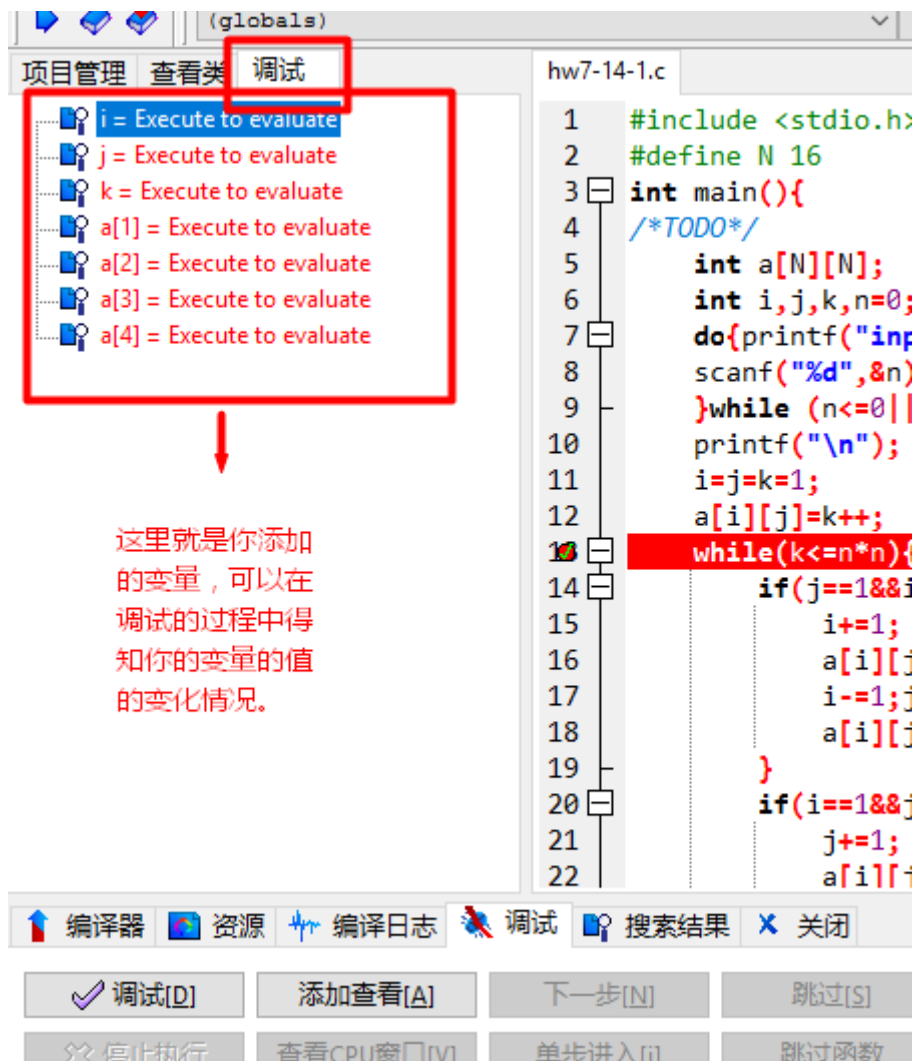
② 输入你要查看的变量名（注意是输入变量名）

你可以输入标准的整型变量和实型变量名，还可以输入数组名。

比如，你用 `int a[10]` 定义了一个数组，你可以通过输入 `a` 来查看整个数组的值，你也可以通过输入 `a[n]` 来查看数组中某个元素的值，这个 `n` 可以是变量也可以是一个指定的常数，比如 `a[n]` 和 `a[3]` 都是可以的。

注意！一次只能输入一个变量！

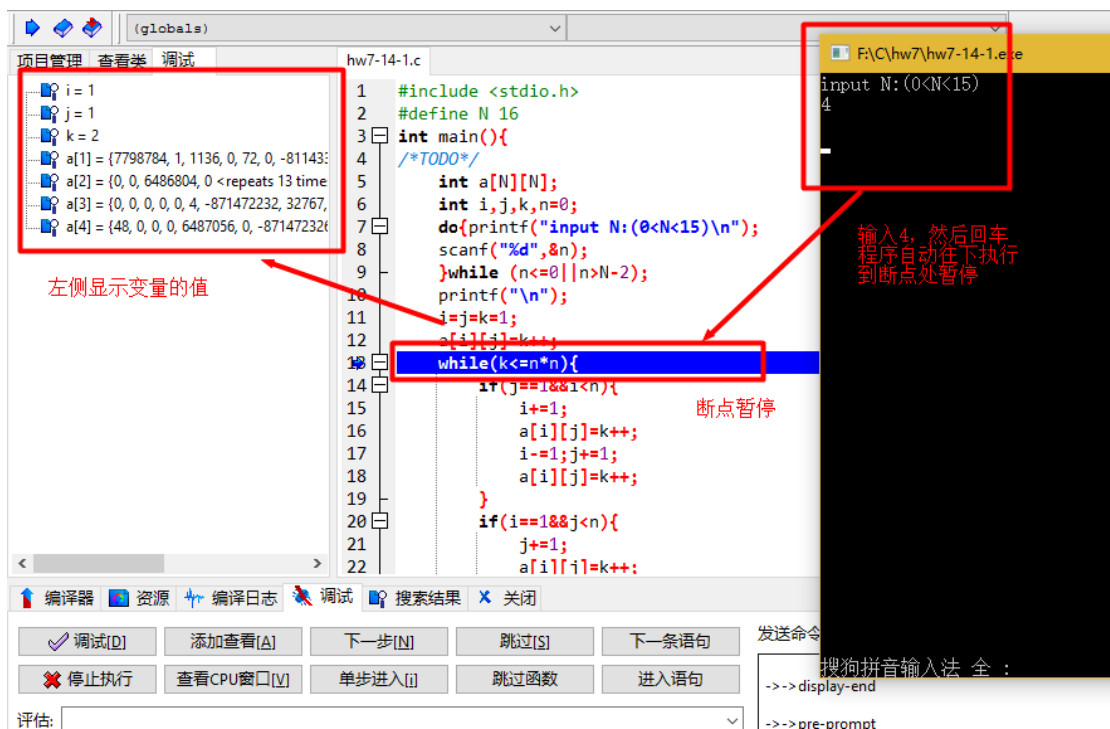
③ 在界面左侧，选择“调试”，就可以看到你添加的变量了（在开始调试之前，暂时都是 `Execute to evaluate`）



④ 全部添加好后，设置好断点，点击调试按钮，开始调试。

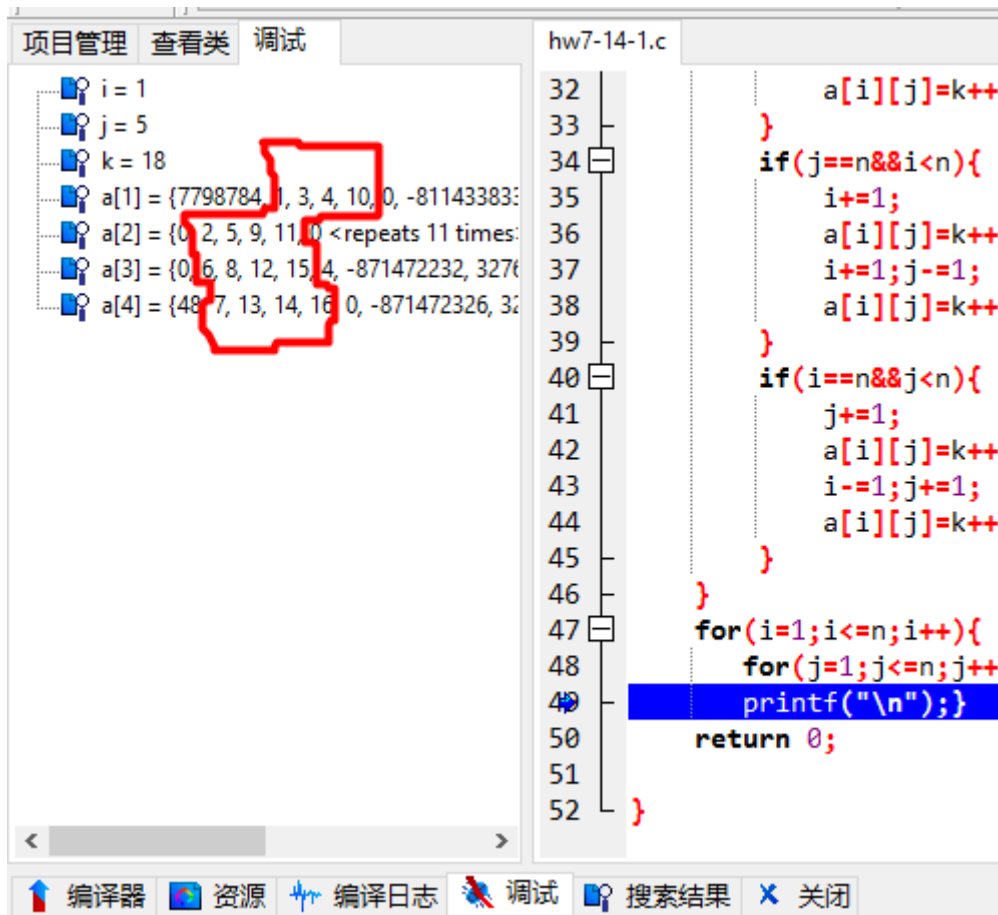
⑤ 用我的这个程序举例（课本第 4 章课后习题 14 题第一问）

我的目的是，输入一个整数 N ($0 < N < 15$)，然后按照题目要求输出 $1-N^2$ 的排列矩阵。按照课本要求，我选择 4 为例。

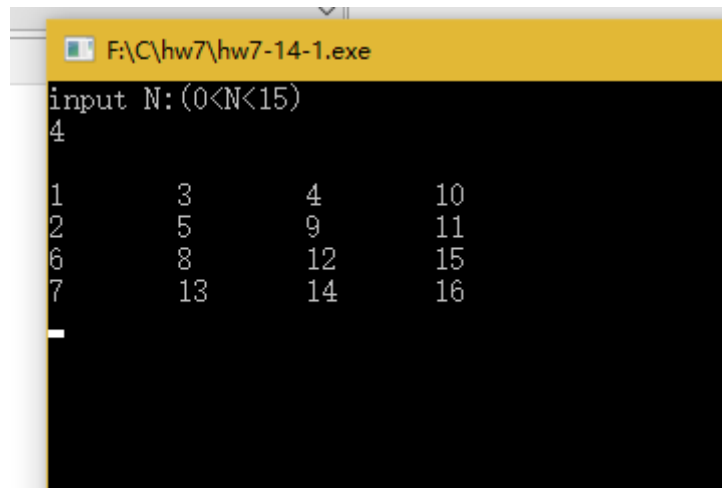


如图，可以看到，我的程序在执行到断点处的时候自动暂停，并且左侧显示出了变量的即时值。

- ⑥ 点击按钮 **下一步[N]** 让程序继续往下进行，观察变量的变化。
- ⑦ 重复步骤 6
- ⑧ 通过观察变量的变化，就可以得知程序的问题了。在观察的时候你自己也要思考，按照你的想法这个变量应该是什么值，实际上是什么值，为什么会与你的想法不一样。
- ⑨ 可以看到，我的数组已经形成了我想要的矩阵。



继续进行下一步，可以看到事实上也的确输出了我想要的矩阵。



调试完成。（正常而言都是会有问题的，我拿了个调试好的程序举例而已。）

- 4、调试之前**一定要编译！一定要编译！一定要编译!!!**重说三!!!!
- 5、尽量添加一些比较关键的变量，添加的变量最好都是会影响到你想观察的循环的变量，不要添加一些无谓的你根本不会去看的变量，那只会让你看不过来。