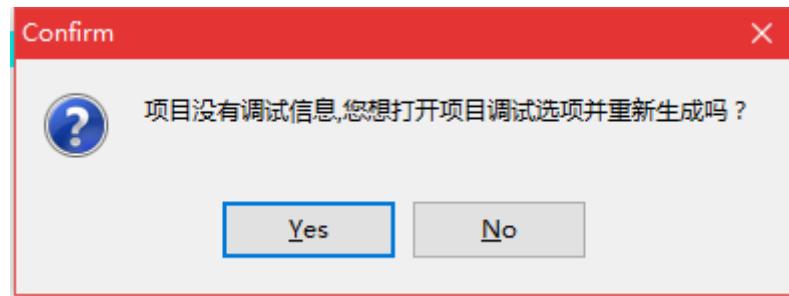


关于如何使用 DEV C++软件进行调试：

### | 出错情况 |

1、点击调试后弹出这样的窗口：

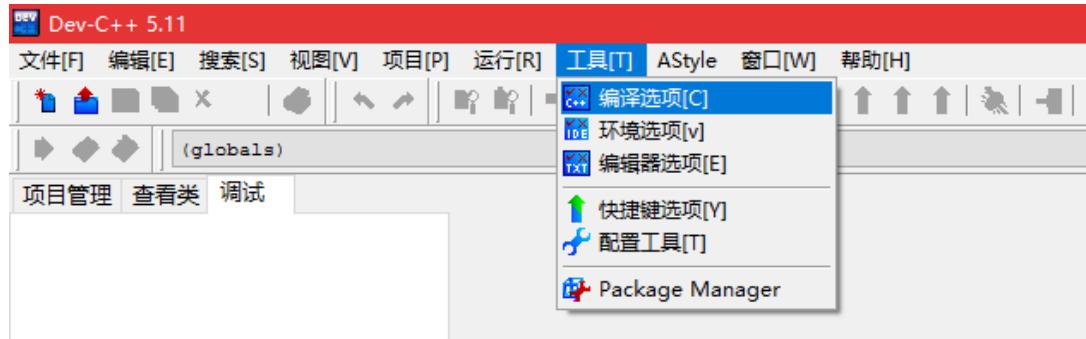


2、点击 YES 后直接停止工作或者闪退。

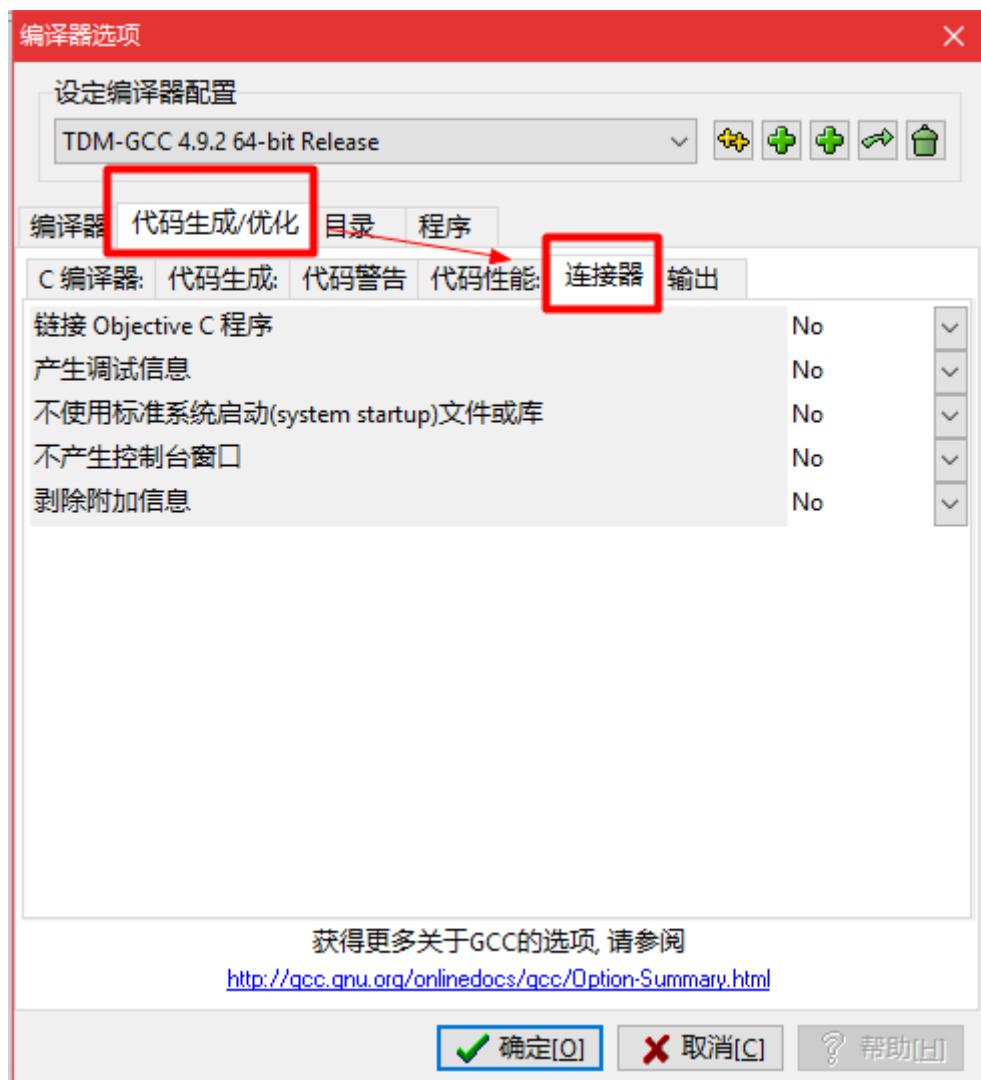
3、使用的是先导课程上所给地址下载的 DEV C++ 并且**没有对编译环境进行过改动** (UI 改动、字体改动等除外)

### | 解决方法 |

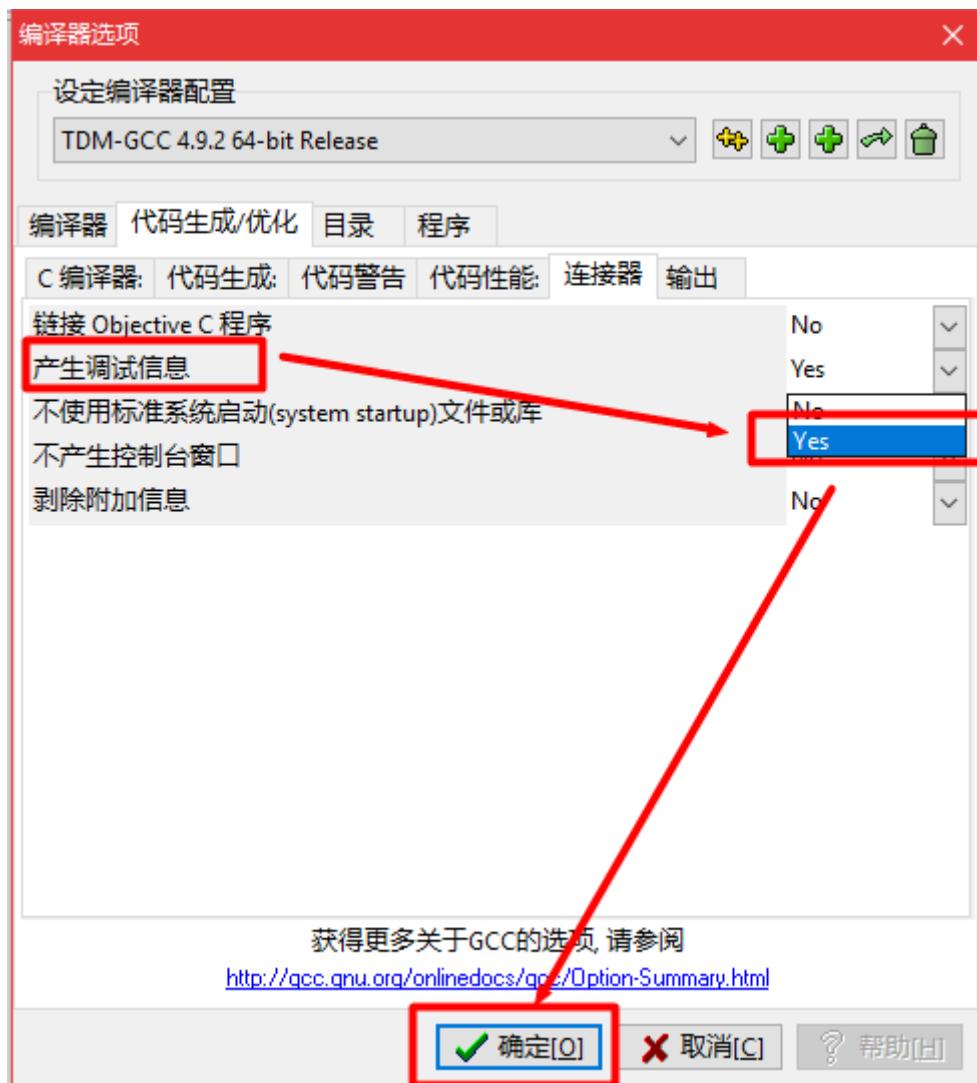
1、打开 DEV C++软件，找到菜单栏中的工具-编译选项。



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 |

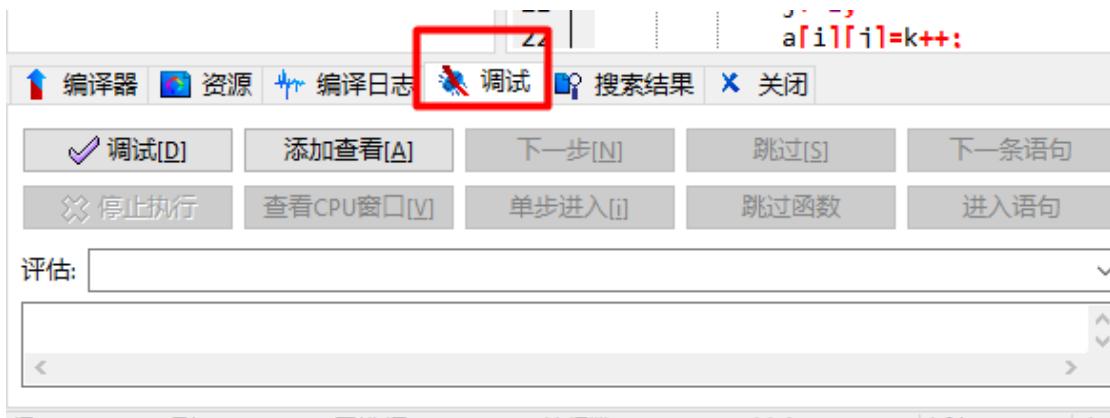
1、调试是一定要加断点的。最好的断点加入是在某一个循环上添加断点。

```
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        }
18    }
19 }
```

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`）

The screenshot shows a debugger interface with a toolbar at the top. The '调试' (Debug) tab is selected, highlighted with a red box. Below it, a list of variables is shown with a red box around them:

- i = Execute to evaluate
- j = Execute to evaluate
- k = Execute to evaluate
- a[1] = Execute to evaluate
- a[2] = Execute to evaluate
- a[3] = Execute to evaluate
- a[4] = Execute to evaluate

A red arrow points from this list down to a text box containing the following explanatory text:

这里就是你添加的变量，可以在调试的过程中得知你的变量值的变化情况。

The main window displays a C program named 'hw7-14-1.c' with line numbers 1 through 22. The code is as follows:

```
#include <stdio.h>
#define N 16
int main(){
    /*TODO*/
    int a[N][N];
    int i,j,k,n=0;
    do{printf("inp");
        scanf("%d",&n);
    }while (n<=0||n>16);
    printf("\n");
    i=j=k=1;
    a[i][j]=k++;
    while(k<=n*n){
        if(j==1&&i==1)
            i+=1;
        a[i][j]=i;
        i-=1;j+=1;
        a[i][j]=i;
        if(i==1&&j==1)
            j+=1;
        a[i][j]=j;
    }
}
```

At the bottom, there are several buttons:

- 编译器 (Compiler)
- 资源 (Resources)
- 编译日志 (Build Log)
- 调试 (Debug) - This button is highlighted.
- 搜索结果 (Search Results)
- 关闭 (Close)

Below these are more buttons:

- 调试 (Debug)
- 添加查看 (Add Watch)
- 下一步 (Next)
- 跳过 (Skip)
- 停止执行 (Stop Execution)
- 查看CPU窗口 (View CPU Window)
- 单步进入 (Step Into)
- 跳过函数 (Skip Function)

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。

左侧显示变量的值

输入4, 然后回车  
程序自动往下执行  
到断点处暂停

```

1 #include <stdio.h>
2 #define N 16
3 int main(){
4     /*TODO*/
5     int a[N][N];
6     int i,j,k,n=0;
7     do{printf("input N:(0<N<15)\n");
8         scanf("%d",&n);
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            i-=1;j+=1;
18            a[i][j]=k++;
19        }
20        if(i==1&&j<n){
21            j+=1;
22            a[i][j]=k++;
}

```

评估: | ->-> ore-promot

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。

项目管理 查看类 调试 hw7-14-1.c

```
i = 1  
j = 5  
k = 18  
a[1] = {7798784, 1, 3, 4, 10, 0, -81143383;...  
a[2] = {0, 2, 5, 9, 11, 0 <repeats 11 times;...  
a[3] = {0, 6, 8, 12, 15, 4, -871472232, 3276,...  
a[4] = {48, 7, 13, 14, 16, 0, -871472326, 3276,...  
32 } a[i][j]=k++  
33 }  
34 if(j==n&&i<n){  
35 i+=1;  
36 a[i][j]=k++  
37 i+=1;j-=1;  
38 a[i][j]=k++  
39 }  
40 if(i==n&&j<n){  
41 j+=1;  
42 a[i][j]=k++  
43 i-=1;j+=1;  
44 a[i][j]=k++  
45 }  
46 }  
47 for(i=1;i<=n;i++){  
48 for(j=1;j<=n;j++)  
49 printf("\n");}  
50 return 0;  
51 }  
52 }
```

编译器 资源 编译日志 调试 搜索结果 关闭

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。

```
E:\C\hw7\hw7-14-1.exe  
input N:(0<N<15)  
4  
1 3 4 10  
2 5 9 11  
6 8 12 15  
7 13 14 16
```

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。