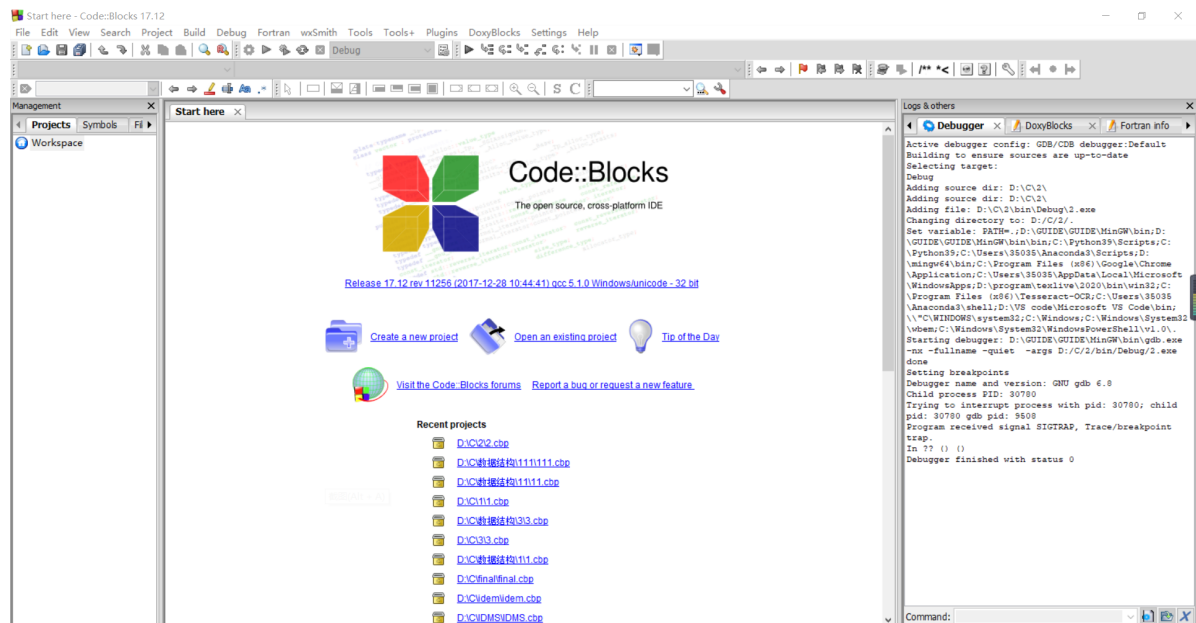
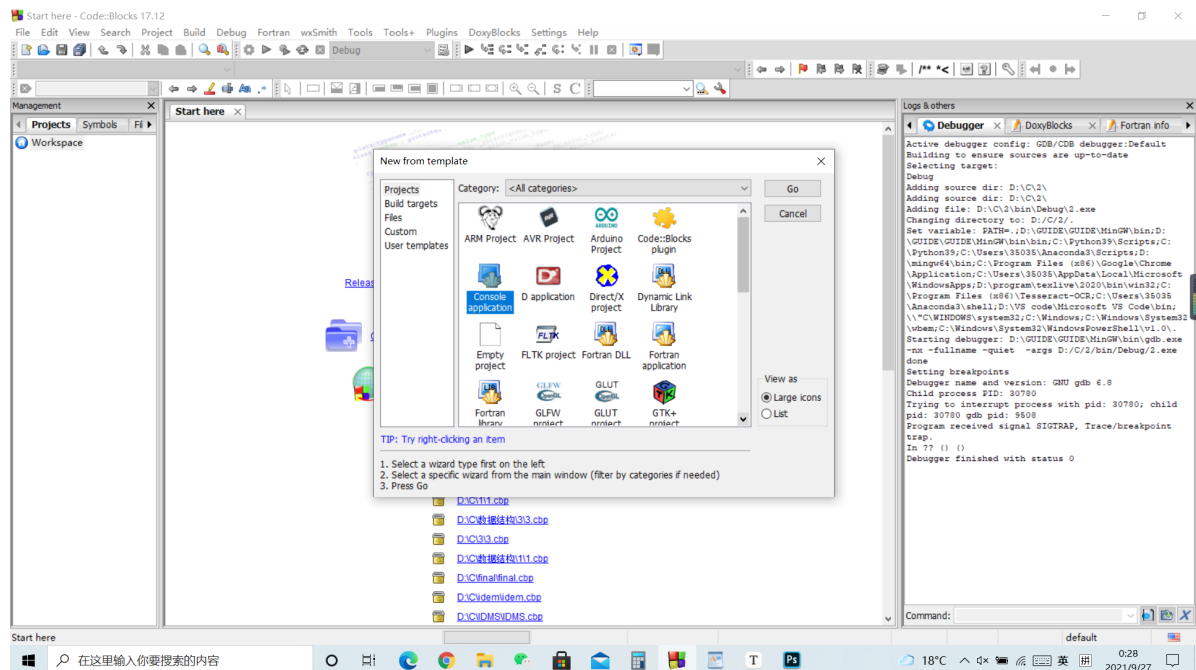


codeblocks 使用与调试教程(๑•̀ㅂ•́) ✿

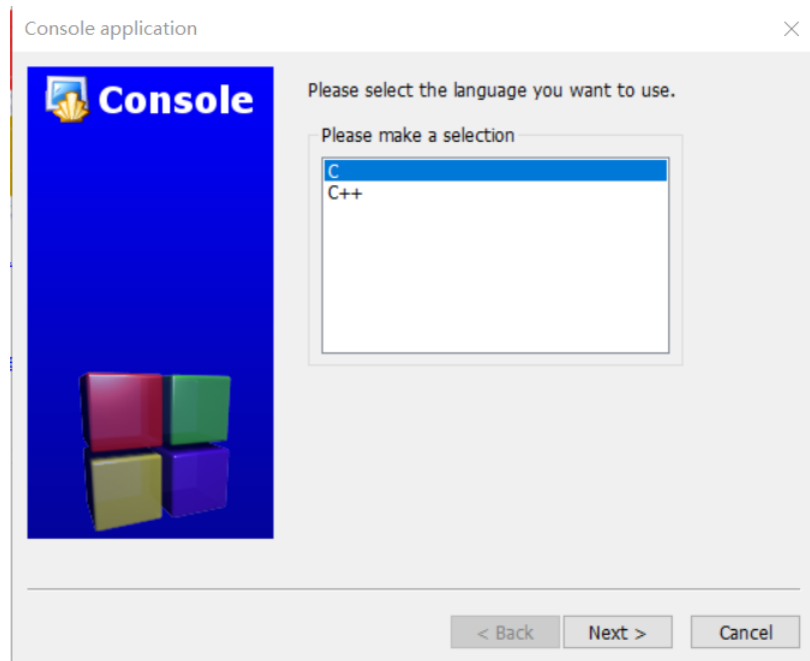
创建项目



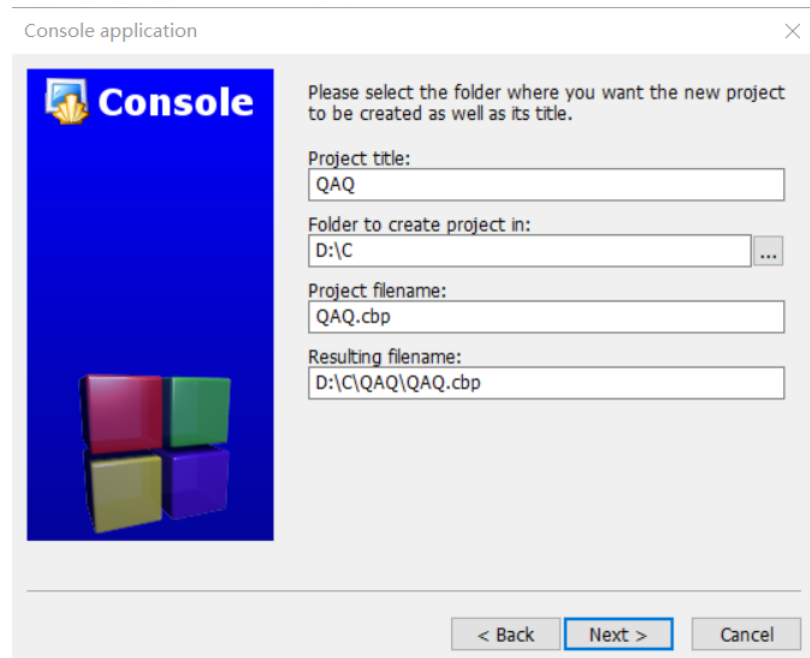
创建一个新的项目点击“Creat a new project”



选择如图所示的“Console application”后点击右侧的“go”。

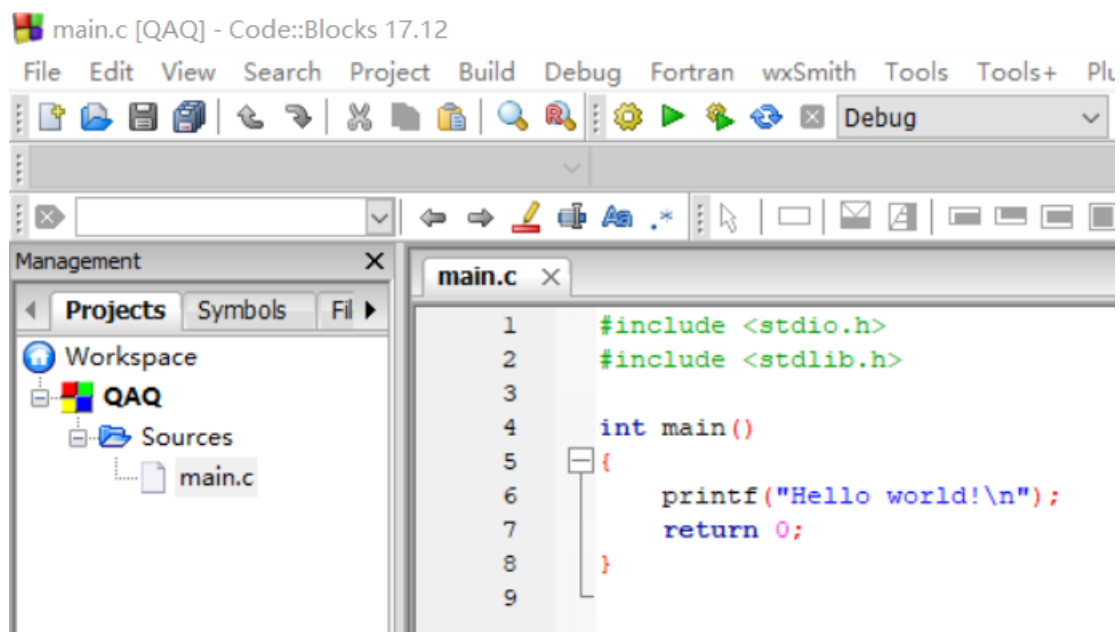


选择语言（一般选C）。



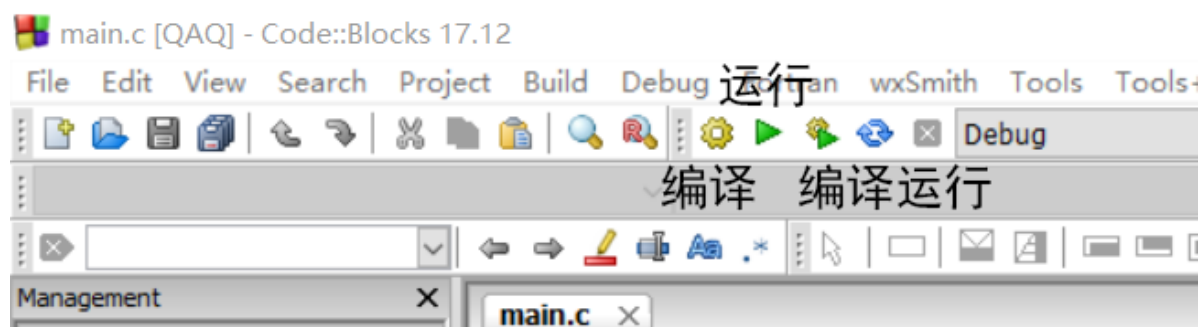
设置项目名称和储存路径，注意第二行储存路径**不要在任何中文目录下**，否则调试功能可能出现问题。

最后按提示点击“next”和“finish”，完成项目的创建。



单击左侧的Sources旁边的'+', 双击“main.c”, 开始敲代码(๑_๑)☺

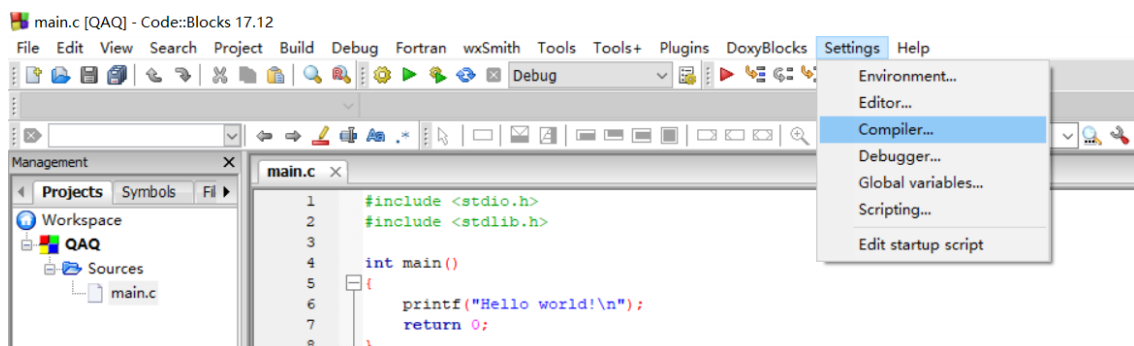
编译与运行



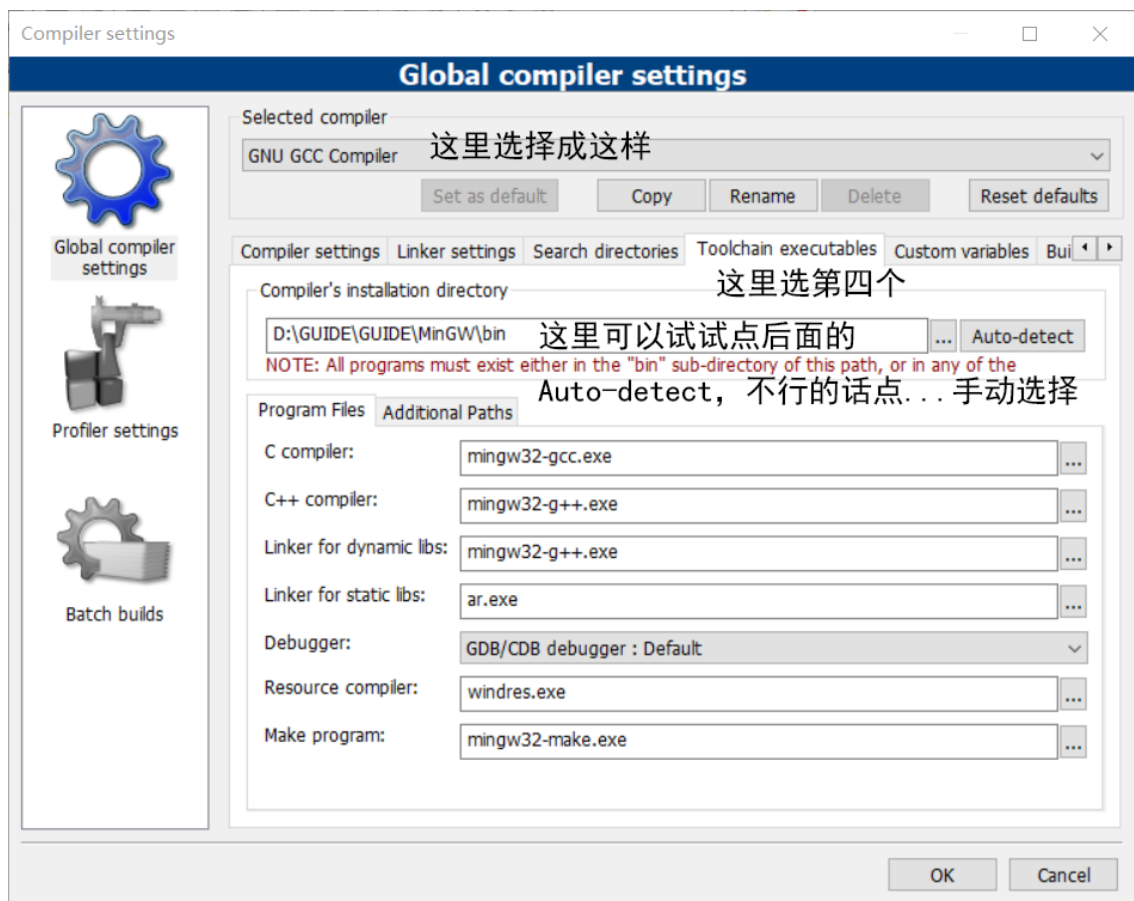
注意运行前，需要编译，或者直接点击“编译运行”。

编译路径设置（如果已经能正常编译&运行就可以忽略了）

- setting -> Compiler...



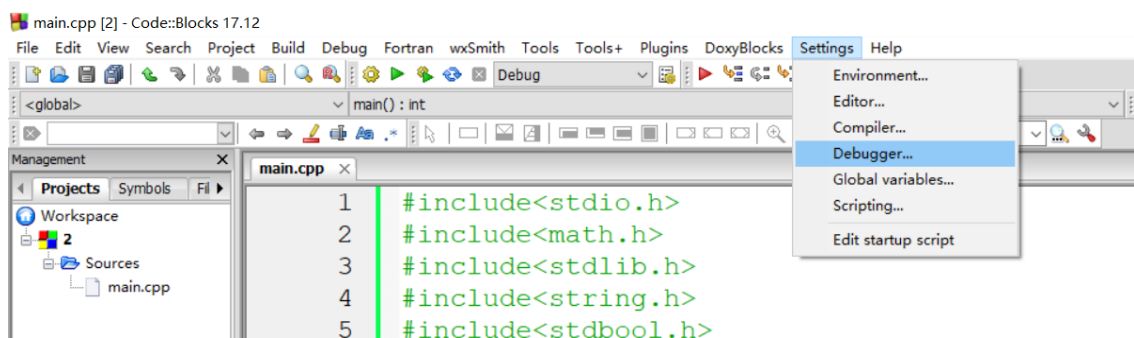
- 然后这样



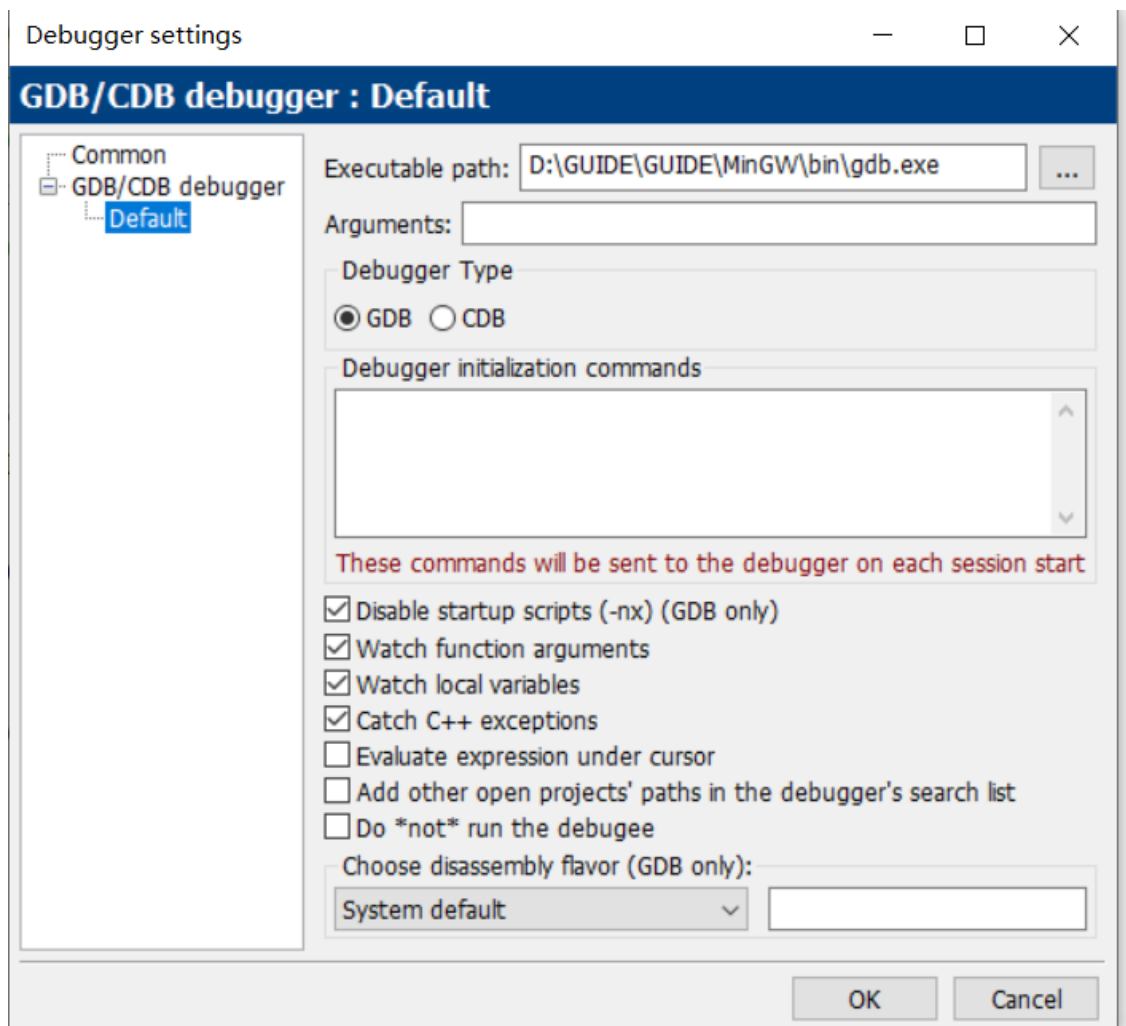
我标注的那一格，要设置到一个bin文件夹，一般就你安装codeblocks的地方（不要看我的我设置到别的地方了呜呜~），可以试试按后面的Auto-detect，不行的话再手动选择。

调试配置

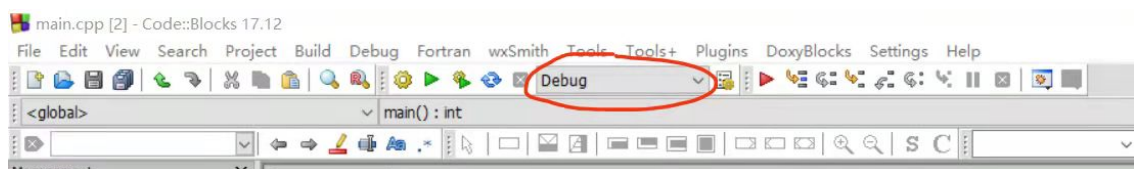
- setting -> Debugger...



- 左侧选择 Default



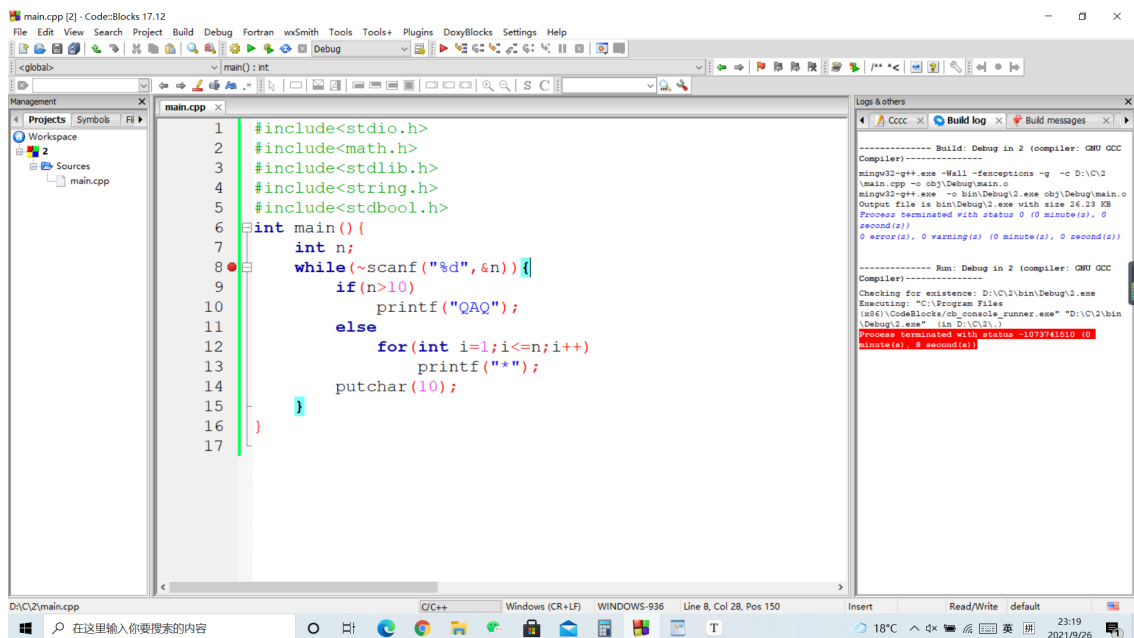
- 检查Executable path，里面是不是设置到了某个位置的gdb.exe，如果没有的话点旁边的"..."选择路径，一般在你安装codeblocks的位置，如果安装codeblocks的文件夹里没有gdb.exe就去网上下一个或者找我拷一个QAQ。
- 以及Debug模式需要启动：



红圈那里要选中Debug

调试

- 设置断点：F5



鼠标单击需要设置断点的位置（行数），按快捷键**F5**设置断点。

- 运行到断点位置：**F8**

设置断点后按快捷键**F8**，程序运行到断点位置：



左侧的黄色箭头表示程序当前运行到的位置，此时程序运行到第八行，正在等待输入。

- 运行到下一行：**F7**

```
1  #include<stdio.h>
2  #include<math.h>
3  #include<stdlib.h>
4  #include<string.h>
5  #include<stdbool.h>
6  int main() {
7      int n;
8      while(~scanf("%d",&n)) {
9          if(n>10)
10             printf("QAQ");
11         else
12             for(int i=1;i<=n;i++)
13                 printf("*");
14         putchar(10);
15     }
16 }
17
```

输入5后，按F7程序运行到下一行。

- 以下功能等讲到函数就用到了：
 - 进入函数：Shift + F7

```
1  #include<stdio.h>
2  #include<math.h>
3  #include<stdlib.h>
4  #include<string.h>
5  #include<stdbool.h>
6  void Test(int n) {
7      printf("TAT\n");
8      printf("%d\n",n);
9  }
10 int main() {
11     int n;
12     while(~scanf("%d",&n)) {
13         Test(n);
14         if(n>10)
15             printf("QAQ");
16         else
17             for(int i=1;i<=n;i++)
18                 printf("*");
19         putchar(10);
20     }
21 }
```

在上图状态Shift + F7，查看Test函数内部的运行过程。

```

1  #include<stdio.h>
2  #include<math.h>
3  #include<stdlib.h>
4  #include<string.h>
5  #include<stdbool.h>
6  void Test(int n){
7      printf("TAT\n");
8      printf("%d\n",n);
9  }
10 int main(){
11     int n;
12     while(~scanf("%d",&n)){
13         Test(n);
14         if(n>10)
15             printf("QAQ");

```

(观察左侧箭头位置)

- 跳出当前函数: **Ctrl + F7**

在上图状态按Ctrl + F7跳出Test函数:

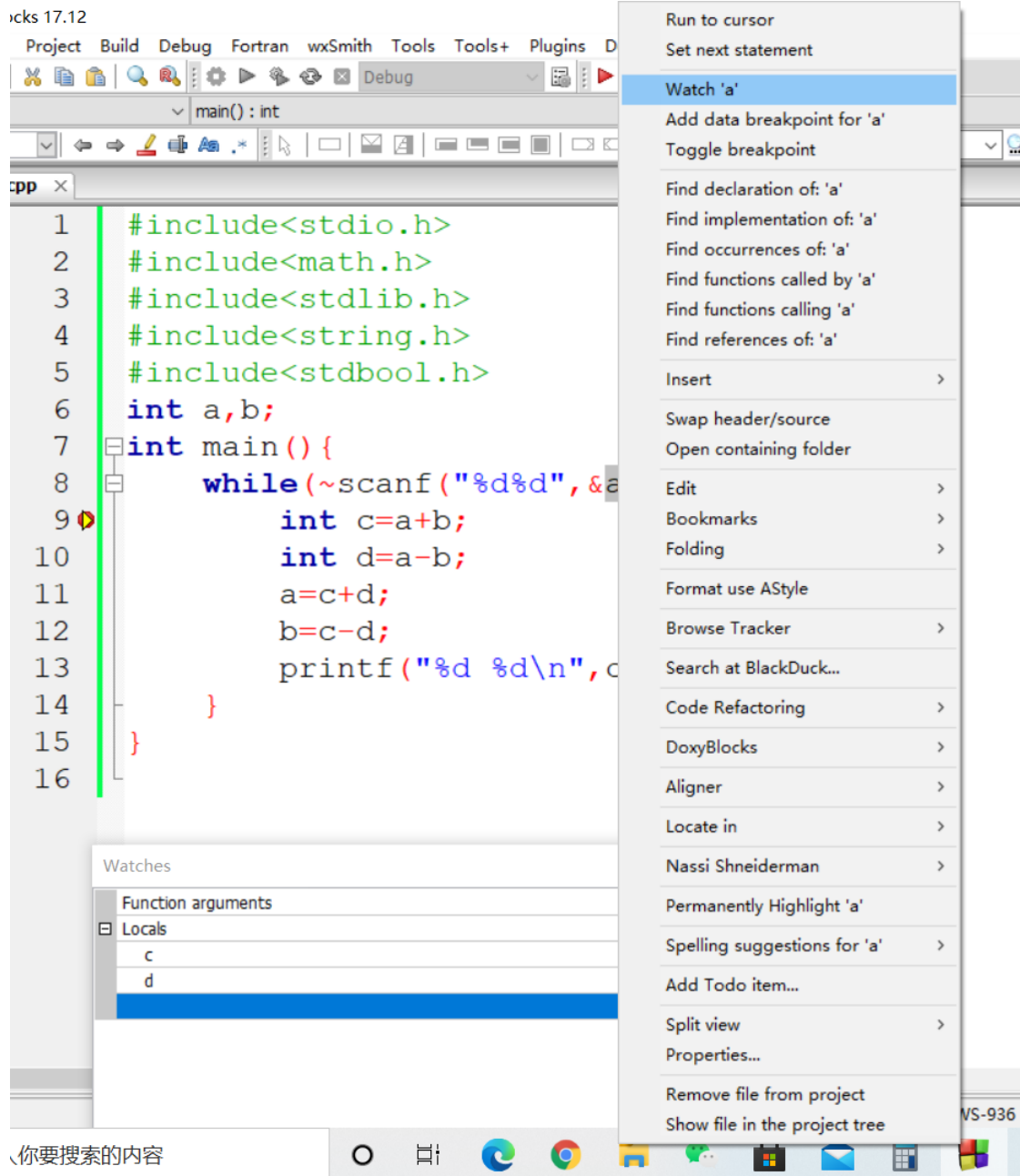
```

1  #include<stdio.h>
2  #include<math.h>
3  #include<stdlib.h>
4  #include<string.h>
5  #include<stdbool.h>
6  void Test(int n){
7      printf("TAT\n");
8      printf("%d\n",n);
9  }
10 int main(){
11     int n;
12     while(~scanf("%d",&n)){
13         Test(n);
14         if(n>10)
15             printf("QAQ");
16         else
17             printf("QAQ");

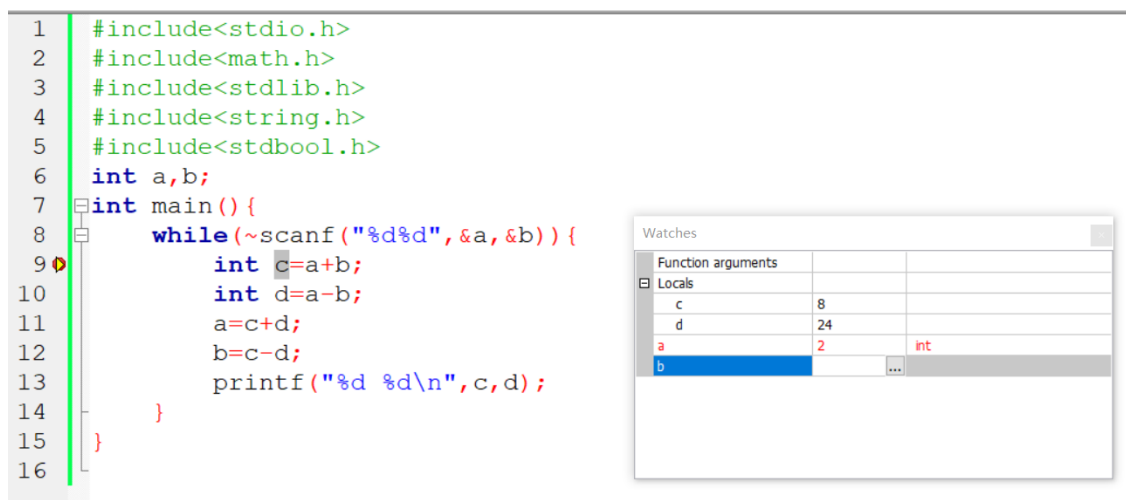
```

- 变量查看: (变量、数组、函数等都能查看)

选中需要查看的变量, 单击右键, 选择“watch ...”, 如下图中选中变量a:

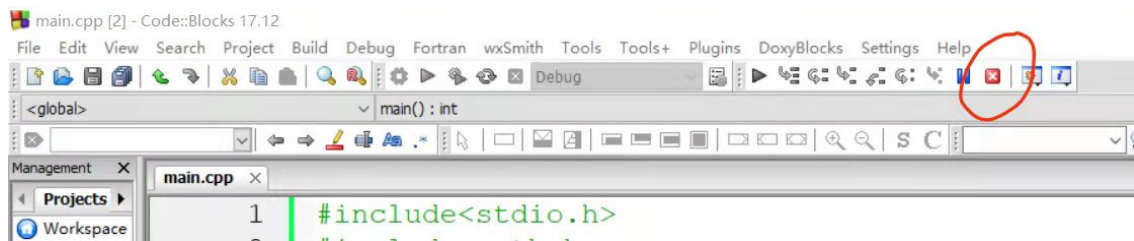


会弹出变量查看的窗口，在该窗口可以查看当前变量的值，也可以在该窗口输入新增或删除可查看的变量



Watches窗口显示了各个变量，在程序运行到左侧黄色箭头时的值（黄色箭头的位置如何变化看前面的部分）。

- 结束调试



点击上图标注的叉就能结束调试。

- 其它

codeblocks的调试功能非常强大，还有许多其它实用的功能，就就就大家自行探索吧(๑•̀ㅂ•́)و✧。

备注

平时做题如果样例都过不了的时候，可以试试用调试功能自行检查，题解代码看不懂的时候也可以用调试功能查看一下代码的运行过程。

后面讲到递归的时候应该会有部分同学觉得不太好理解，那时候也可以翻出调试模拟一下QAQ。

Author YUKILSY (๑•̀ㅂ•́)و✧