
第一次上机调试实验报告

姓名：刘翰文 学号：522030910109

一、实验目的和要求

- 1.熟练掌握关系表达式和逻辑表达式的使用方法。
- 2.熟练掌握 if 语句和小数设置。
- 3.掌握根据“Build messages”窗口的“error”信息修改程序语法错误的方法。
- 4.掌握单步（Next line）和运行到光标（Run to cursor）的调试方法。
- 5.熟练运用调试方法分析程序运行结果与预期不符的原因并修改程序。
- 6.掌握存储路径的中英文规则。

二、实验内容

- 1、下面程序的目的是判断三条边能否组成三角形、是否为直角三角形及其面积加上学号后三位数的和，修改程序中的语法错误

```

#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    double a, b, c, s, area, tmp, no;
    cin >> a >> b >> c >> d >> no;
    if (a < b)
    {
        tmp = a;
        a = b;
        b = tmp;
    }
    if (a < c) {tmp = a;
    a = c ;
    c = tmp;}
    else if (0 == 1) return 0;

    if (b + c > a);
    {
        if (a*a == b*b+c*c)
            cout << "是三角形且为直角三角形，其面积加学号后三位为: ";
        else
            cout << "是三角形但不是直角三角形，其面积加学号后三位为: ";
        s = (a + b + c) / 2;
        area = sqrt (s * (s - a) * (s - b) * (s - c));
        cout << area << endl;
    }
    else cout << "不能构成三角形" << endl;

    return 0;
}

```

执行上述程序，查看“Build messages”窗口的“error”信息如下图所示：

File	Line	Message
E:\安装\CodeB...	7	error: 'd' was not declared in this scope
E:\安装\CodeB...	20	warning: this 'if' clause does not guard... [-Wmisleading-indentation]
E:\安装\CodeB...	21	note: ...this statement, but the latter is misleadingly indented as if it were guarded...
E:\安装\CodeB...	27	error: expected ')' before ';' token
E:\安装\CodeB...	30	error: 'else' without a previous 'if'

发现有三处语法错误：1、d 没有在先前被定义 2、在；前缺少） 3、else 前少了 if。

修改错误：1、发现 d 为不必要的数，删去 d 2、在；前加上） 3、由于 30 行上面的 if 语句后加了；导致错误，故删去；

再次查看：

File	Line	Message
		=== Build: Debug in homework (compiler: GNU GCC Compiler) ===
E:\安装\CodeB...		In function 'int main()':
E:\安装\CodeB...	27	error: 'sqrt' was not declared in this scope
E:\安装\CodeB...	27	note: suggested alternative: 'qsort'
		=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

又发现一处语法错误：使用 sqrt 时未在开头使用 cmath

修改错误：在开头添加#include <cmath>

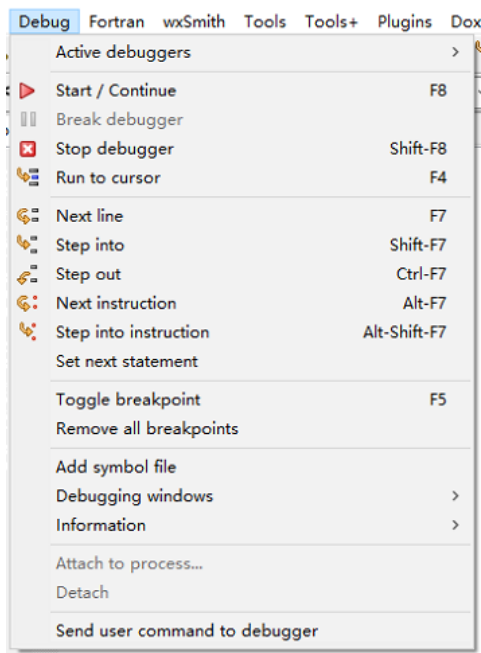
2、下面是修改完上述程序的语法错误后的结果，根据单步调试的方法寻找下面程序的输出与预期结果不符的原因，并修改错误。

```
1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5  int main()
6  {
7      double a, b, c, s, area, tmp, no;
8      cin >> a >> b >> c >> no;
9      if (a < b)
10     {
11         tmp = a;
12         a = b;
13         b = tmp;
14     }
15     if (a < c) {tmp = a;
16     a = c ;
17     c = tmp;}
18     else if (0 == 1) return 0;
19
20
21     if (b + c > a)
22     {
23         if (a*a == b*b+c*c)
24             cout << "是三角形且为直角三角形，其面积加学号后三位为： ";
25         else
26             cout << "是三角形但不是直角三角形，其面积加学号后三位为： ";
27         s = (a + b + c) / 2;
28         area = sqrt (s * (s - a) * (s - b) * (s - c));
29         cout << area << endl;
30     }
31     else cout << "不能构成三角形" << endl;
32
33     return 0;
34 }
```

调试过程如下：

- (1) 输入 3 4 5 235，其结果为“是三角形且为直角三角形，其面积加学号后三位为：6”。结果不符合预期。 输入 1 2 3 235，其结果为“不能构成三角形”。结果符合预期。 输入 3 3 3 235， 其结果为“是三角形但不是直角三角形，其面积加学号后三位为：3.89711”。结果不符合预期
- (2) 单步调试是发现运行错误和逻辑错误的“利器”，可用于：
 - a) 跟踪程序的执行流程，发现错误的线索。
 - b) 跟踪过程中，还可以观察变量的变化，从而发现其中存在的问题。
- (3) 调试菜单。选择“Debug”，观察菜单，了解有哪些功能；灰色的部分，会在“条件”具备时变得可用，如下图所示。左下角的状态栏有各选中功能


的解释，请仔细查看解释，便于学习日常应用的常用工具栏或快捷键。

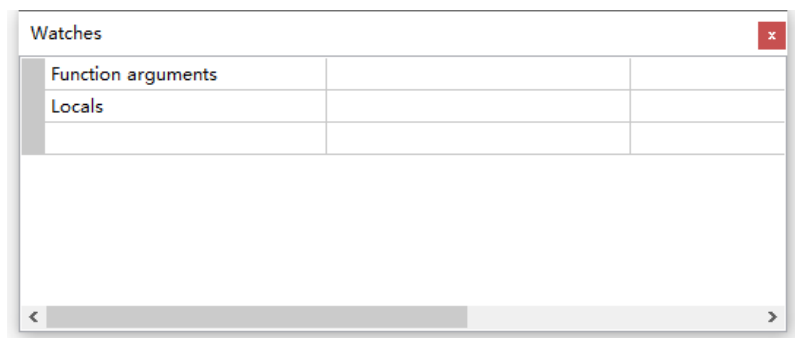


- (4) 调试工具栏。单击工具栏中的按钮，比选择菜单更便捷。将鼠标“悬浮”在工具栏按钮上，会看到“Run to cursor”和“Next line”等提示信息，如下图所示。这些功能“Debug”菜单也有，但这里更便捷。



图 2. 调试工具栏

- (5) 观察窗口，观察窗口一般出现在屏幕的右下角。如果被使用者关闭了，可以通过单击工具栏 （Debugging windows）按钮，弹出菜单，选择“Watches”选项，观察窗口会再次出现在屏幕上，如下图所示。




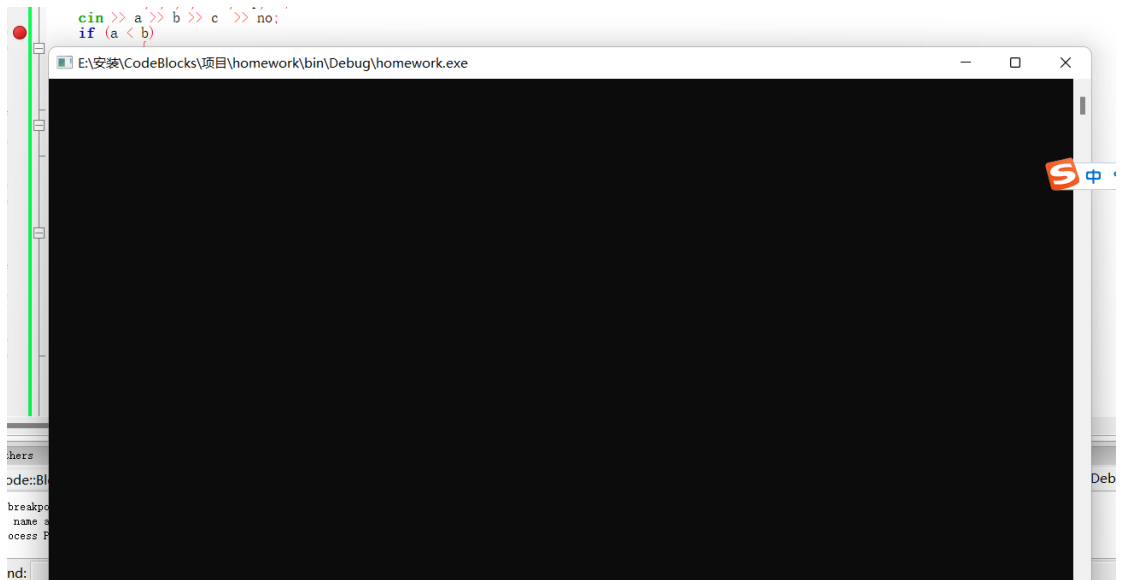
- (6) 接下来进行单步调试，将鼠标移动至程序左侧，当鼠标变成向右倾斜时，单击鼠标左键，程序会在鼠标所在的行生成一个红色的圆，如下图所示，这个步骤叫做设置断点。

```

2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5  int main()
6  {
7      double a, b, c, s, area, tmp, no;
8      cin >> a >> b >> c >> no;
9      if (a < b)
10     {
11         tmp = a;
12         a = b;
13         b = tmp;
14     }
15     if (a < c) {tmp = a;
16     a = c ;
17     c = tmp;}

```

- (7) 鼠标点击工具栏的红色三角形  (Debug/Continue) 按钮，会弹出黑色的命令行窗口，如下图所示，此时程序执行到第 8 行并暂停。



点击 (Debug/Continue) 按钮后的命令行窗口

- (8) 在命令行窗口中输入 a,b,c,no 的值，然后按下键盘上的“回车”键，程序会自动执行到断点处并暂停，此时程序左侧会出现黄色的三角形，表示程序运行到当前行并暂停。此时可以在观察窗口查看执行到断点处时，已经定义或运算后各个变量的值，如下图所示。

```

#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main()
{
    double a, b, c, s, area, tmp, no;
    cin >> a >> b >> c >> no;
    if (a < b)
    {
        tmp = a;
        a = b;
        b = tmp;
    }
    if (a < c) {tmp = a;
    a = c;
    c = tmp;}
    else if (0 == 1) return 0;

    if (b + c > a)
    {
        if (a*a == b*b+c*c)
            cout << "是三角形且为直角三角形，其面积加学号后三位为，";
        else
            cout << "是三角形但不是直角三角形，其面积加学号后三位为，";
        s = (a + b + c) / 2;
        area = sqrt(s * (s - a) * (s - b) * (s - c));
        cout << area << endl;
    }
    else cout << "不能构成三角形" << endl;
    return 0;
}

```

Watches		
Function arguments		
Locals		
a	3	
b	4	
c	5	
s	3.5360159697103013e-317	
area	2.0256691479491108e-322	
tmp	7.9050503334599447e-323	
no	235	

(9) 将断点放在输出 area 前，如图调试程序：

Watches

Function arguments		
Locals		
a	5	
b	3	
c	4	
s	6	
area	6	
tmp	4	
no	235	

```

21  if (b + c > a)
22  {
23      if (a*a == b*b+c*c)
24          cout << "是三角形且为直角三角形，其面积加学号后三位为，";
25      else
26          cout << "是三角形但不是直角三角形，其面积加学号后三位为，";
27      s = (a + b + c) / 2;
28      area = sqrt(s * (s - a) * (s - b) * (s - c));
29      cout << area << endl;
30  }
31  else cout << "不能构成三角形" << endl;
32  return 0;
33  }
34
35

```

发现在输出前 area 仍然未加上学号，原因即是未按要求加上 no。错误修改：在 area 的赋值后添加 "+no"

(10) 终止调试。单击工具栏  (Stop debugger)，可以终止程序调试。

(11) 带入第三组数，发现小数精确度不够。 错误原因：程序中未加关于小数位的表达。 解决方法：利用 iomanip 的 setprecision


```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5  int main()
6  {
7      cout<<setprecision(3);
8      double a, b, c, s, area, tmp, no;
9      cin >>a>>b>>c>>no;
10     if (a < b)
11     {
12         tmp = a;
13         a = b;
14         b = tmp;
15     }
16     if (a < c) {tmp = a;
17     a = c;
18     c = tmp;}
19     else if (0 == 1) return 0;
20
21     if (b + c > a)
22     {
23         if (a*a == b*b+c*c)
24             cout << "是三角形且为直角三角形，其面积加学号后三位为：";
25         else
26             cout << "是三角形但不是直角三角形，其面积加学号后三位为：";
27         s = (a + b + c) / 2;
28         area = sqrt (s * (s - a) * (s - b) * (s - c))+no;
29         cout << area << endl;
30     }
31     else cout << "不能构成三角形" << endl;
32     return 0;
33 }
34
35

```

最终结果

三、实验小结/课程建议

学习了调试系统，了解并尝试了  的使用，学会了用 watches 和 next line 排查错误，并学会了在 build 后根据下方报告改正语法错误。在我个人实践中，也因自己的原因在存储路径上出错更加深刻的了解到存储路径全英文的重要性。Debug 和 build 的对错误的排查在程序设计上十分重要，此次试验更让我对它有了更深刻的认识，为今后的程序设计奠定基础。