

FEEL THE MEANING OF THE TRIP

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

DREAM
MY DREAM WILL NEVER STOP

高级语言程序设计

实验二 程序调试&基本算术运算

GO!
TAKE YOU ON A TRIP



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

探索 从未停止



实验回顾（一）

➤ %c读入时出错的情况

```
printf("double c=");  
scanf("%lf", &c);  
printf("c=%f, sizeof(c)=%d\n", c, sizeof(c));  
  
printf("char d=");  
scanf("%c", &d);  
printf("d=%c, sizeof(d)=%d\n", d, sizeof(d));
```

```
D:\CodeBlocksWorkspace\Variables\bin\Debug\Variables.exe  
double c=232432.23423  
c=232432.234230, sizeof(c)=8  
char d=d=  
, sizeof(d)=1  
  
Process returned 0 (0x0) executi  
Press any key to continue.
```

➤ 原因:

用%c读入字符时，空格和转义字符（比如回车）都会被当作有效字符读入

➤ 解决方法:

- ① 方法1. 使用函数getchar()将缓冲区的回车符读入;
- ② 方法2. %c前面加一个空格。



实验回顾（二）

➤ short int型、double型变量的输入输出

short int型占2字节，int型占4字节

float型占4字节，double型是8字节

scanf() 函数将输入写入指定地址，因为（通过指针）指向float存储和指向double存储是不一样的，因此，**scanf区别%d和%hd**。

同理，scanf区别%f和%lf。

实验回顾 (二)

main.c x

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int a;
7     printf("Please input int a: ");
8     scanf("%d",&a);
9     printf("int a = %d, size of int is %I64d .\n\n",a,sizeof(int));
10
11     short b;
12     printf("Please input short b: ");
13     scanf("%d",&b);
14     printf("short b = %d, size of short is %I64d .\n\n",b,sizeof(short));
15
16     printf("int a = %d, size of int is %I64d .\n\n",a,sizeof(int));
17
18     return 0;
19 }
20
```

Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Vera++ x

File	Line	Message
=== Build: Debug in experiment1_1 (compiler: GNU GCC Compiler) ===		
In function 'main':		
D:\code\c\e...	13	warning: format '%d' expects argument of type 'int *', but argument 2 has type 'short int *' [-Wformat=]
D:\code\c\e...	13	warning: format '%d' expects argument of type 'int *', but argument 2 has type 'short int *' [-Wformat=]
=== Build finished: 0 error(s), 2 warning(s) (0 minute(s), 0 second(s)) ===		

D:\code\c\experiment1_1\bin\Debug\experiment1_1.exe

Please input int a: 4
int a = 4, size of int is 4 .

Please input short b: 8
short b = 8, size of short is 2 .

int a = 0, size of int is 4 .

Process returned 0 (0x0) execution time : 19.761 s
Press any key to continue.

00	00
00	00
00	00
04	00
初始值	00
初始值	08

目录

CONTENT

01

实验目的

02

调试和排错

03

实验内容

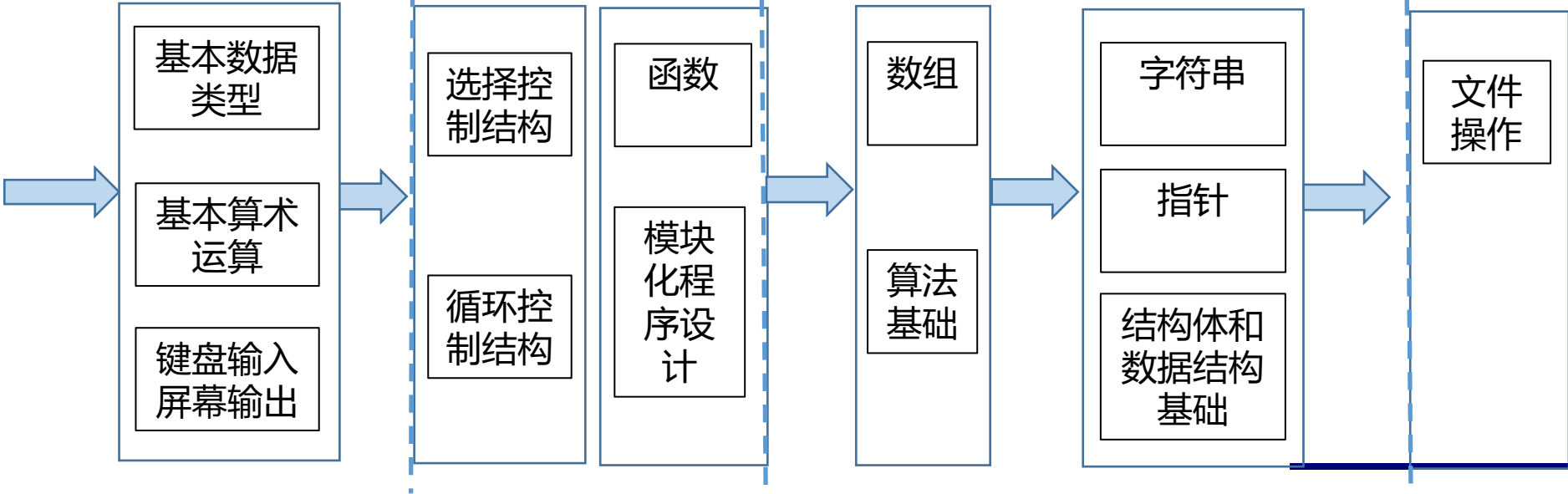
04

OJ课程平台



实验课程安排与考核标准

编号	1	2	3	4	5	6	7	8	9
学时数	2	2	2	2	2	2	2	2	2
实验项目	初级编程	算数运算	循环控制	模块化程序设计	数组算法	字符串二维数组	指针	结构体	文件管理
分数	1	2	2	2	3	3	3	3	3
授课内容	集成开发环境	程序调试	编码规范	程序测试				软件文档	期末检查





实验目的

- 掌握基本的算术运算；
 - 掌握选择控制结构；
 - 掌握关系运算符和逻辑运算符；
 - 掌握常用的程序调试和排错方法。
-



调试和排错—程序的常见错误类型

编译 错误

语法错误，编译器会告知我们。

例如：变量使用前未定义，语句少了分号，括号少了一边等

链接 错误

缺少包含文件，或者包含文件的路径错误等。

例如：使用了math的库函数，没有包含math.h文件等

运行时 错误

程序无法正常运行
运行结果与预期不一致



➤常用方法

- 可以在关键位置增加打印语句和测试代码
- 使用调试（Debug）工具
- 采用注释的办法隐藏一些代码，调试无误后再“打开”隐藏代码

调试和排错

范例程序：判断一个正整数的奇偶性。

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, a;
    printf("请输入一个正整数:");
    scanf("%d", &n);
    a = n%2;

    //判断一个正整数的奇偶性
    if(a == 1)
        printf("这是一个奇数\n");
    else
        printf("这是一个偶数\n");
}
```

```
H:\1-D\code1\OddEven_judge\bin\Debug\OddEven_judge.exe
请输入一个正整数:8
这是一个奇数

Process returned 0 (0x0)   execution time : 1.769 s
Press any key to continue.
```

```
H:\1-D\code1\OddEven_judge\bin\Debug\OddEven_judge.exe
请输入一个正整数:3
这是一个奇数

Process returned 0 (0x0)   execution time : 1.700 s
Press any key to continue.
```

```
H:\1-D\code1\OddEven_judge\bin\Debug\OddEven_judge.exe
请输入一个正整数:4
这是一个奇数

Process returned 0 (0x0)   execution time : 1.723 s
Press any key to continue.
```



调试和排错 (一)

➤ 可以在**关键位置**增加打印语句和测试代码

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, a;
    printf("请输入一个正整数:");
    scanf("%d", &n);
    a = n%2;

    //判断一个正整数的奇偶性
    if(a == 1)
        printf("这是一个奇数\n");
    else
        printf("这是一个偶数\n");
}
```

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, a;
    printf("请输入一个正整数:");
    scanf("%d", &n);
    printf("n=%d\n", n); //输出n
    a = n%2;
    printf("a=%d\n", a); //输出a

    //判断一个正整数的奇偶性
    if(a == 1)
    {
        printf("这是一个奇数\n");
        printf("in if: n=%d, a=%d\n", n, a);
    }

    else
    {
        printf("这是一个偶数\n");
        printf("in else: n=%d, a=%d\n", n, a);
    }
}
```



调试和排错 (一)

➤ 可以在**关键位置**增加打印语句和测试代码

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n, a;
    printf("请输入一个正整数:");
    scanf("%d", &n);
    printf("n=%d\n", n); //输出n
    a = n%2;
    printf("a=%d\n", a); //输出a

    //判断一个正整数的奇偶性
    if(a == 1)
    {
        printf("这是一个奇数\n");
        printf("in if: n=%d, a=%d\n", n, a);
    }

    else
    {
        printf("这是一个偶数\n");
        printf("in else: n=%d, a=%d\n", n, a);
    }
}
```

D:\CodeBlocksWorkspace\2021\2_EvanOrOdd\bin\Debug\EvanOrOdd.exe

请输入一个正整数:8

n=8

a=0

这是一个奇数

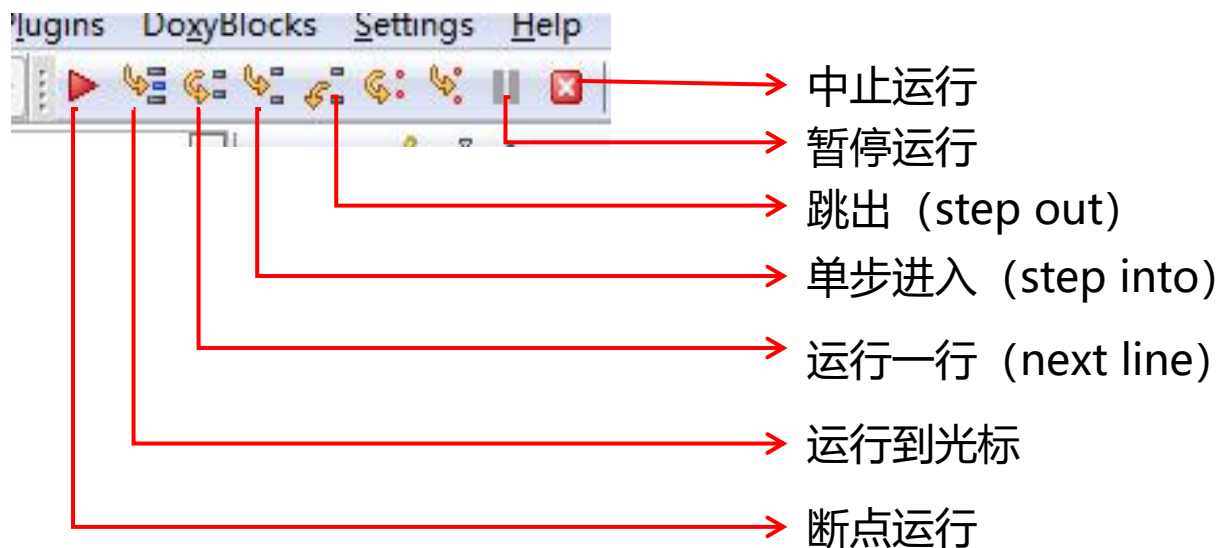
in if: n=8, a=1



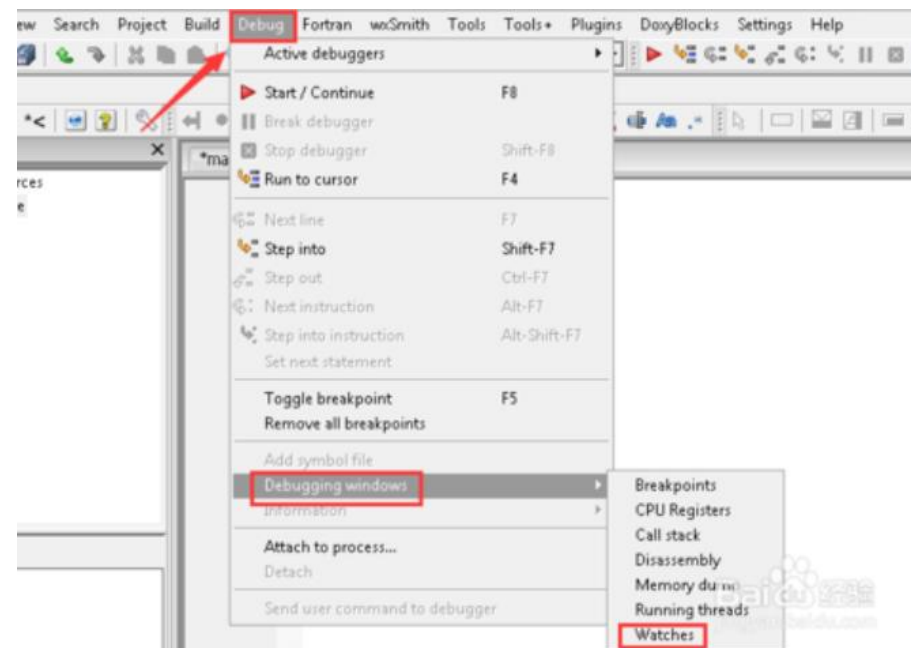
调试和排错 (二)

➤ 使用调试 (Debug) 工具

常用工具栏



打开监视窗



调试和排错 (二)

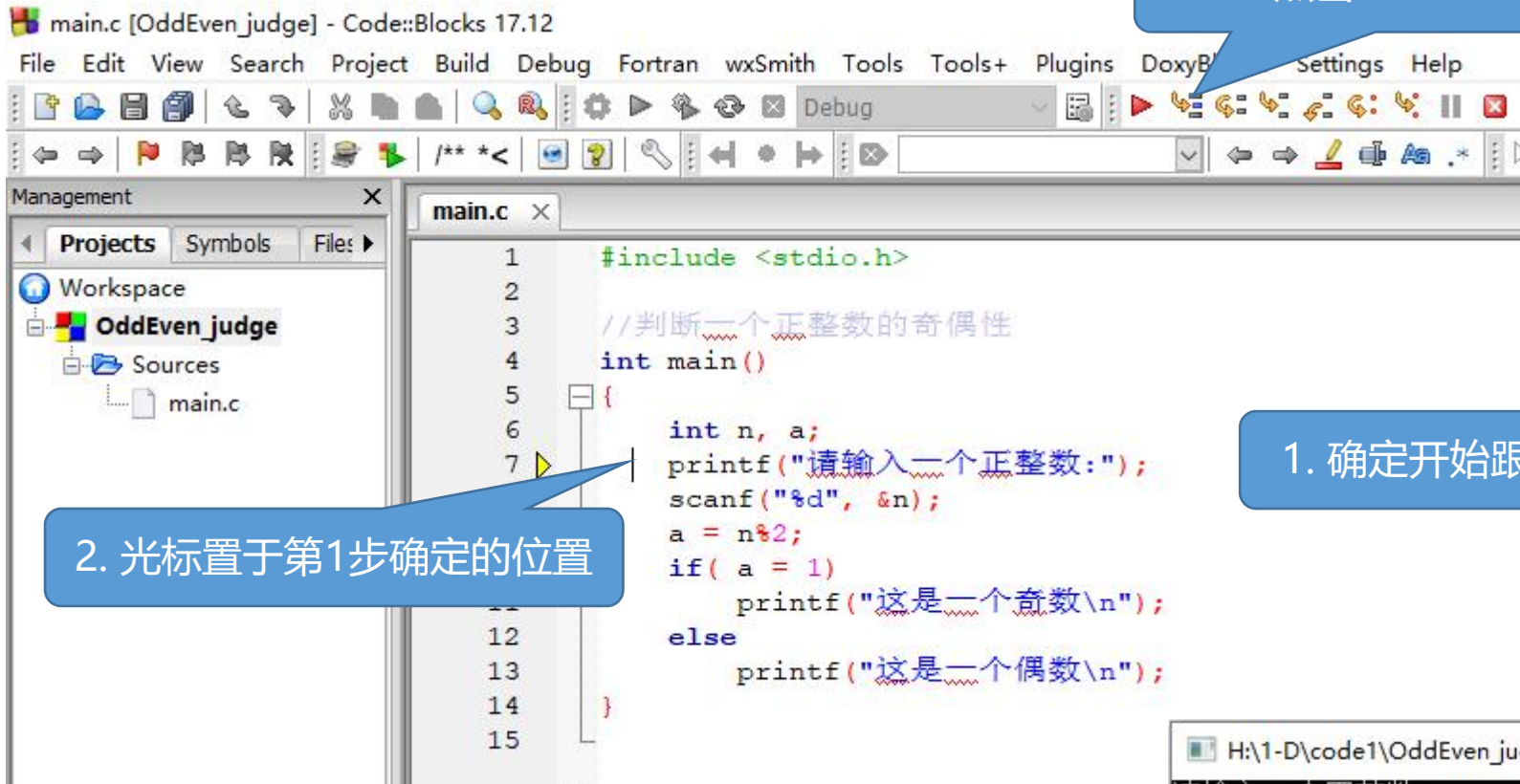
• 单步调试

3. 点击Run to cursor

1. 确定开始跟踪执行的位置

2. 光标置于第1步确定的位置

4. 按要求输入一个正整数，回车

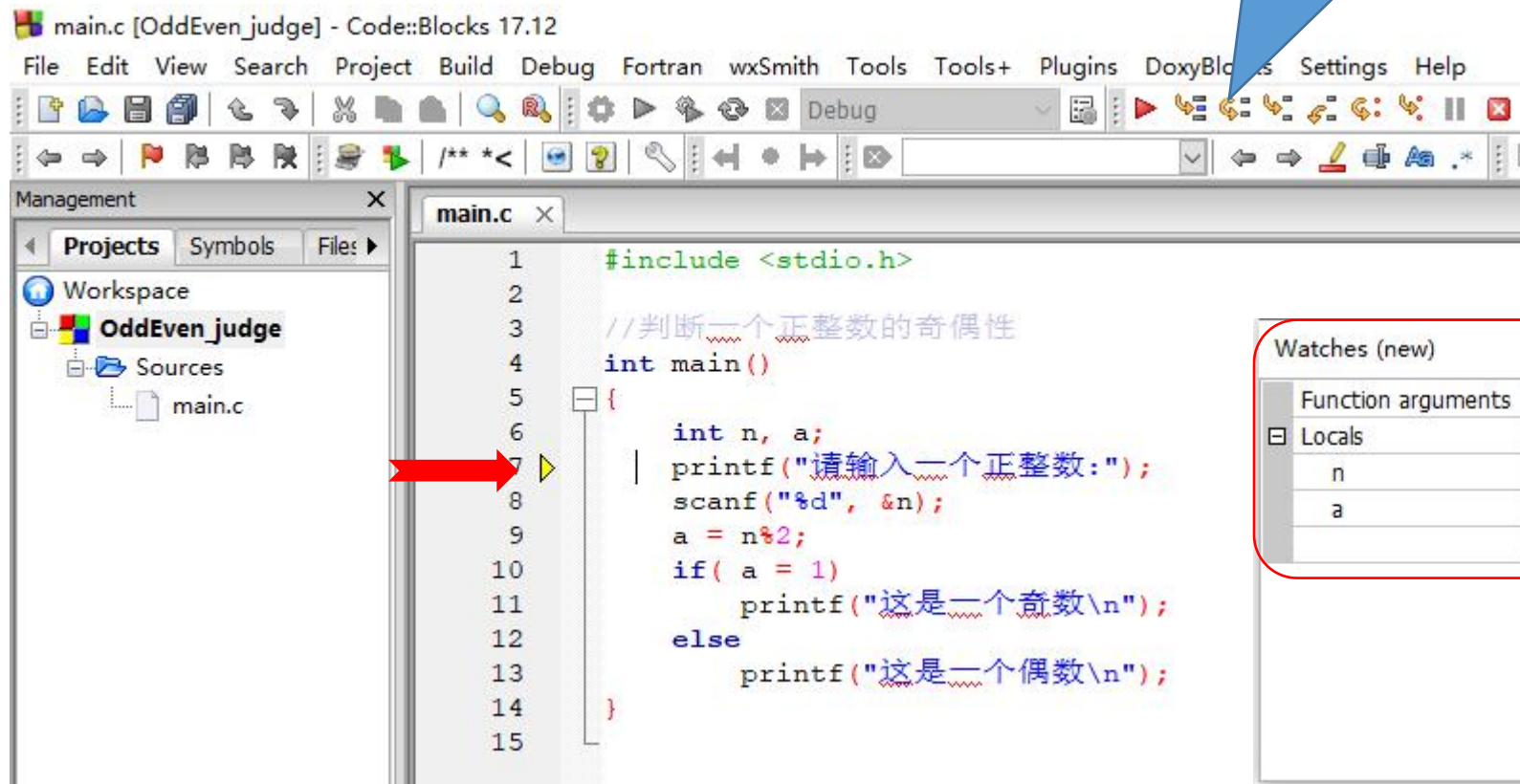


H:\1-D\code1\OddEven_judge\bin\Debug\OddEven_judge.exe

请输入一个正整数:

调试和排错 (二)

• 单步调试



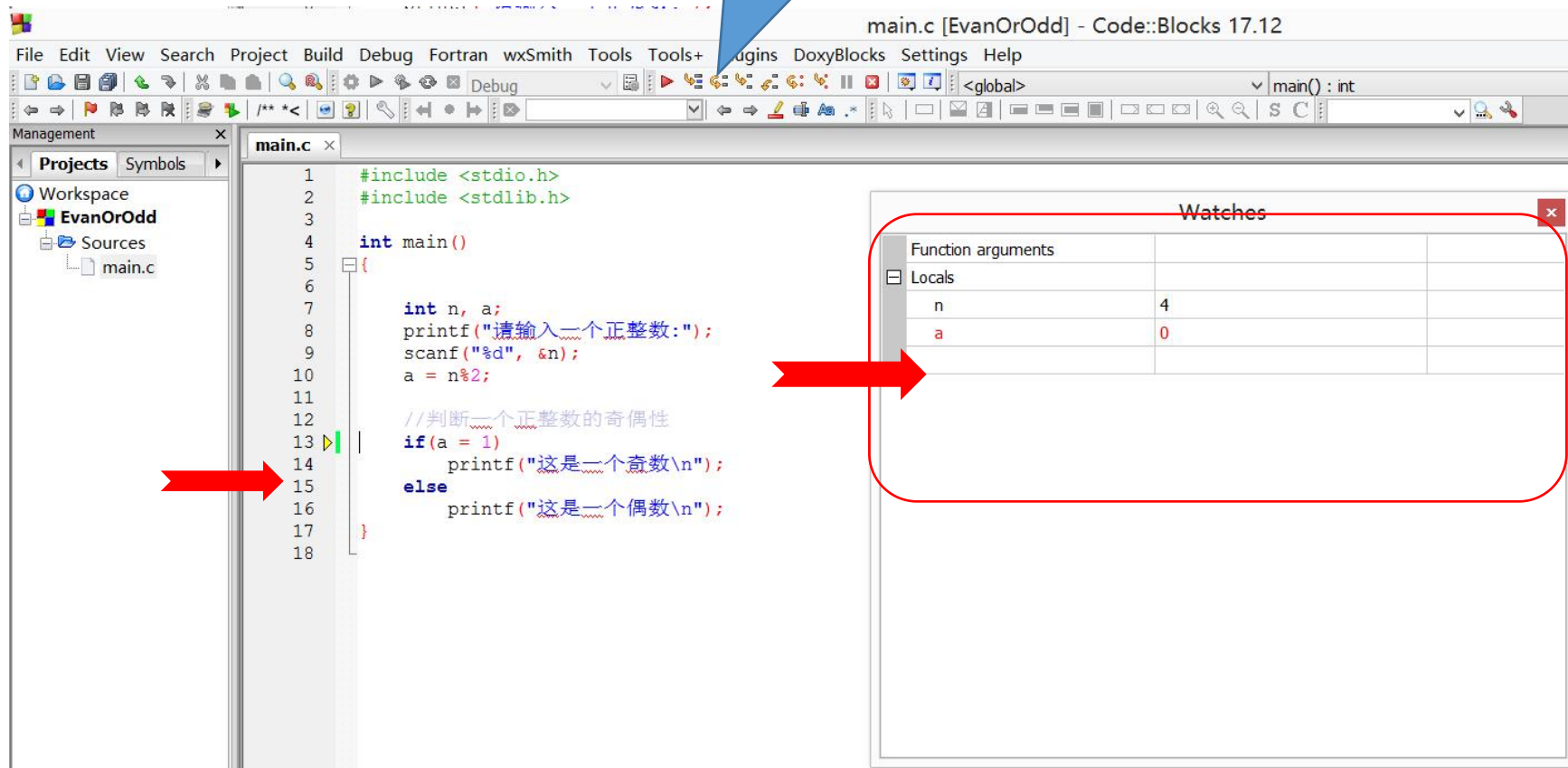
- Function arguments: 函数参数的值
- Locals: 当前有效的局部变量的值
- 空白行: 可添加想要查看的变量, 选中自定义变量所在的行, 右键可删除特定行。



调试和排错 (二)

- 单步调试

7. 点击Next line





调试和排错 (二)

- 单步调试

8. 点击 Stop Debugger

The screenshot shows the Code::Blocks IDE with a C program open. The program is designed to check if a number is odd or even. The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6
7     int n, a;
8     printf("请输入一个正整数:");
9     scanf("%d", &n);
10    a = n%2;
11
12    //判断一个正整数的奇偶性
13    if(a = 1)
14        printf("这是一个奇数\n");
15    else
16        printf("这是一个偶数\n");
17 }
18
```

A yellow star is placed on line 13, and a red arrow points to the condition `if(a = 1)`. Another red arrow points from this condition to the 'Watches' window. The 'Watches' window shows the following data:

Watches	
Function arguments	
<input type="checkbox"/> Locals	
n	4
a	1



调试和排错 (二)

- 单步调试

The screenshot shows the Code::Blocks IDE with the following components:

- Project Explorer:** Shows a workspace named 'EvanOrOdd' with a source file 'main.c'.
- Code Editor:** Displays the source code of 'main.c'. The code includes `<stdio.h>` and `<stdlib.h>`, defines a `main()` function, and prompts the user for a positive integer. It then checks if the number is even or odd using `a = n%2;` and `if(a == 1)`. A red arrow points to the `if(a == 1)` statement.
- Debugger Console:** Shows the output of the program. It displays the prompt '请输入一个正整数:' followed by the user input '4'. The program then outputs '这是一个偶数' (This is an even number). Below this, it shows 'Process returned 0 (0x0) execution time : 2.365 s' and 'Press any key to continue.'.



调试和排错 (二)

- 断点调试

范例程序：
求两个数中的最大值

```
#include <stdio.h>
#include <stdlib.h>

int max(int x, int y); //返回两个int型数中较大的数

int main()
{
    int m, a, b;
    a=100;
    b=80;

    m=max(a, b);
    printf("最大值为:%d\n", m);

    return 0;
}

/*返回两个int型数中较大的数*/
int max(int x, int y)
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    return z;
}
```



调试和排错 (二)

- 断点调试

什么是断点?

在哪里设置断点?

范例程序:
求两个数中的最大值

```
#include <stdio.h>
#include <stdlib.h>

int max(int x, int y); //返回两个int型数中较大的数

int main()
{
    int m, a, b;
    a=100;
    b=80;

    m=max(a, b);
    printf("最大值为:%d\n", m);

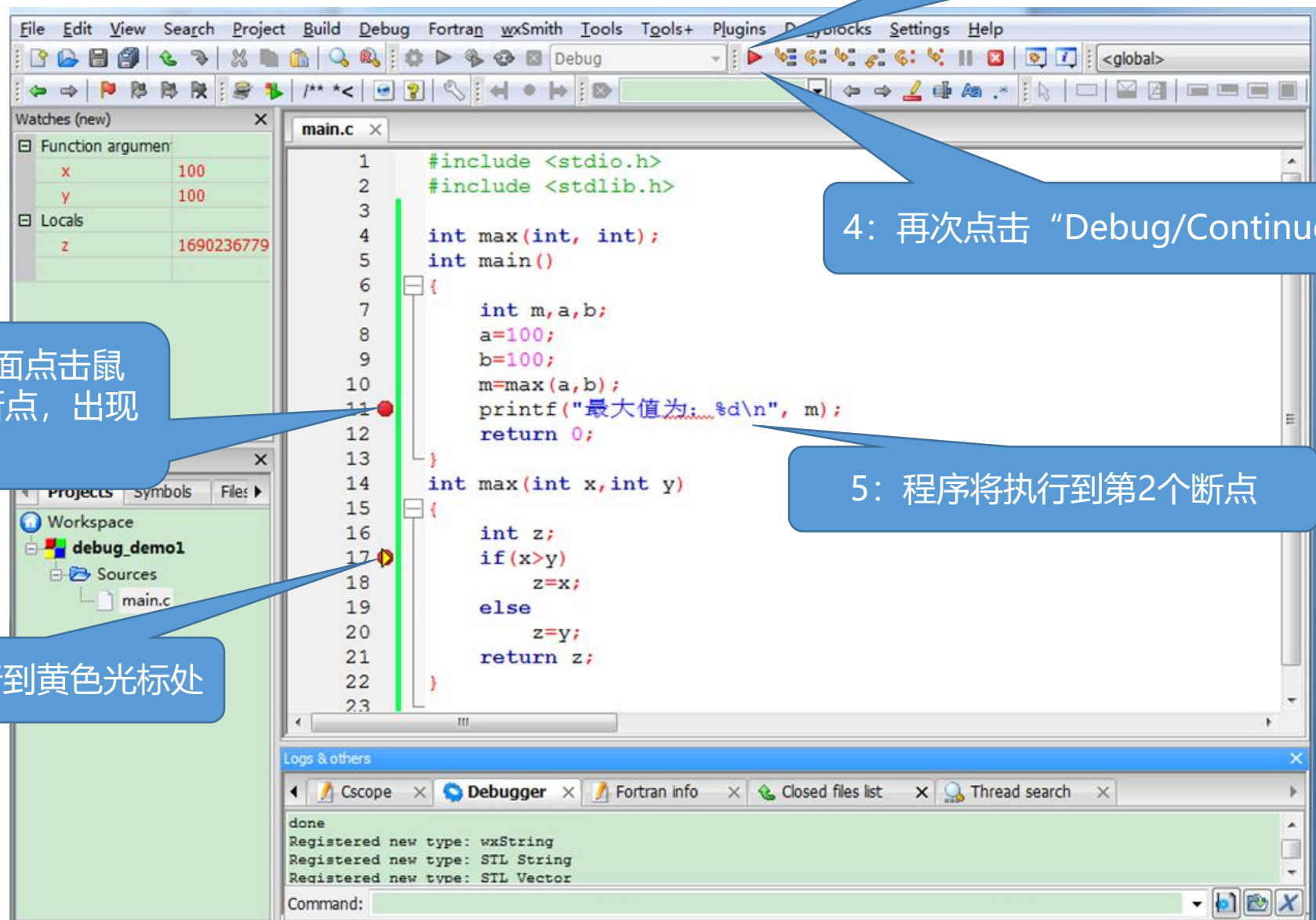
    return 0;
}

/*返回两个int型数中较大的数*/
int max(int x, int y)
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    return z;
}
```



调试和排错 (二)

- 断点调试





➤常用方法

- 可以在关键位置增加打印语句和测试代码
- 使用调试（Debug）工具
- 采用注释的办法隐藏一些代码，调试无误后再“打开”隐藏代码

注意：任何工具都替代不了你的思考



实验内容 – 编程题1

1. 三角形判断

输入三角形的三条边a、b、c，判断它们能否构成一个三角形，如果可以构成三角形，请判断三角形的类型，并输出三角形的面积。

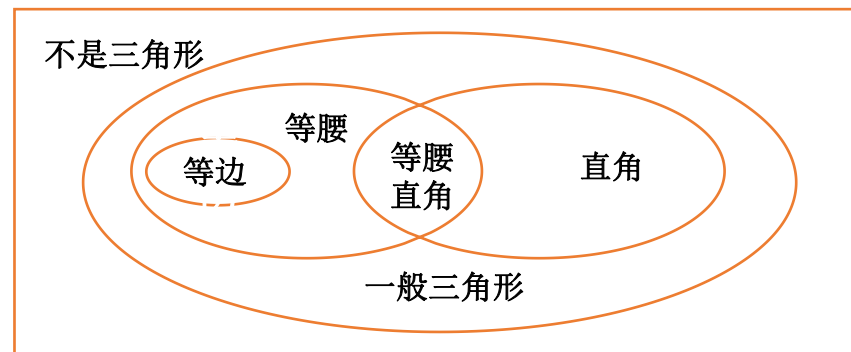
【输入】 3 4 5

【输出】 直角三角形

S=6.00

【说明】

1. 请注意输入输出的空格和换行符；
2. 三角形的三条边数据类型是float；
3. 浮点数精度，小数点后两位；
4. 输入说明：连续输入三个浮点型数据，中间用一个空格间隔
5. 输出的三角形类型，包括等腰直角三角形、直角三角形、等腰三角形、等边三角形、一般三角形、不是三角形。





实验内容 – 编程题1

【提示1】浮点数的比较

1. 浮点数比较大小由于精度问题，如果直接比较可能会出错，所以在比较的时候可以设定一个很小的数值（精度），当二者差小于设定的精度时，就认为二者是相等的。
2. 精度的设置由计算过程中需求而定。

比如：设置精度为1E-2，也就是0.01。对于两个浮点数a、b，如果 $\text{fabs}(a-b) \leq 1\text{E}-2$ ，那么就是相等了；类似的判断大于的时候，就是 $\text{if}(a > b \ \&\& \ \text{fabs}(a-b) > 1\text{E}-2)$ ；判断小于的时候，就是 $\text{if}(a < b \ \&\& \ \text{fabs}(a-b) > 1\text{E}-2)$ 。

【提示2】设三角形三边分别为a、b、c，三角形的面积有多种计算方式，以下提供三种作为参考：

- ① 底×高÷2 [$S = ah \div 2$]
- ② 任意两边之积×这两边的夹角 α 的正弦值÷2 [$S = ab \times \sin\alpha \times 1/2$]
- ③ 海伦公式： $S = \sqrt{p(p-a)(p-b)(p-c)}$ ， $p = (a+b+c)/2$



实验内容 – 编程题2

2. 比较两个分数的大小

人工比较分数大小的常见方法是：对分数进行通分后比较分子的大小。请编程模拟人工比较两个分数的大小。首先输入两个分数分子分母的值，例如11/13,17/19，比较分数大小后，输出相应的提示信息。

【输入】 11/13,17/19

【输出】 11/13 < 17/19

【说明】

1. 请注意输入输出的空格和换行符；
2. 分子和分母均为正整数，分母非零；
3. 输出的大小关系符，前后各有一个空格。



OJ课程平台

登录网址: <http://10.249.176.82:9000/#/login>, 推荐浏览器: Chrome

⚠ 不安全 | 10.249.176.82:9000/#/home

首页 作业 实验题目 用户指南 常见问题和反馈 更新日志











22级电信5班

 实验

实验二 程序调试&基本算术运算

未提交

剩余 8 天

描述:

初始用户名、密码均为学号，登录后请修改



实验二 程序调试&基本算术运算

 完成下列题目

2001.三角形判断

点击题目进入答题页面

2002.比较两个分数大小

2 提交实验报告

下一步



📖 题目描述

🕒 提交记录

🔍 16 px

2001. 三角形判断

输入三角形的三条边a、b、c，判断它们能否构成一个三角形，如果可以构成三角形，请判断三角形的类型，并输出三角形的面积。

【示例1】

输入：

3 4 5

输出：

直角三角形

S=6.00

【示例2】

严格按照输入输出格式才能提交成功，注意大小写、空格、换行等。

输入：

2 2 3

输出：

等腰三角形

S=1.98

【示例3】

输入：

2 4 8

输出：

提交记录需刷新页面才能看到

1 // code here

代码编辑区，按照题目的给定的输入输出格式在本地IDE编写程序，调试运行成功后再复制到OJ平台提交。

测试用例

代码执行结果

提交运行结果

3 4 5

修改代码后需限制性代码再提交

📄 控制台

执行代码

提交



- ❑ 截止时间内，可以重新提交作业，**不限次数**；
- ❑ 只要有一**次提交通过**，即使重新提交代码未通过，也视为**access**状态；
- ❑ 目前仅支持C99，请**在本地IDE完成调试后再提交**课程平台；
- ❑ 代码**点击“提交”**才会保存内容；
- ❑ 代码“提交”后，需**刷新**页面，才能查看“提交记录”和“已提交代码”；
- ❑ 只支持**单文件**。

✓ 不仅要关注是否通过，还需要注意编码规范，写“好的代码”



实验作业提交要求

- **截止时间**

实验课后**一周内**（如周一上课，截止日期为下周二23：59）

- **提交内容**

- ① **截至日期内请务必在OJ课程平台完成代码提交和评测**
- ② 按照实验一要求打包所有项目的工程文件夹，提交至HITsz Grader 作业提交平台备份（除实验九外，不需要再提交word版实验作业）

- **补交**

OJ课程平台暂无补交功能，补交提交至HITsz Grader 作业提交平台。

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

FEEL THE MEANING OF THE TRIP

DREAM

MY DREAM WILL NEVER STOP

请同学们开始实验



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

GO!
TAKE YOU ON A TRIP

探索 从未停止