

FEEL THE MEANING OF THE TRIP

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在



DREAM

MY DREAM WILL NEVER STOP

计算思维与实践

实验一 计算机与程序运行基础

TAKE YOU ON A TRIP
GO!



哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

探索 从未停止



实验室注意事项

- 注意实验室用电安全；
- 请勿携带食品饮料进入实验室；
- 上网需认证；
- 使用自己电脑时，请小心显示器跌落；
- 系统盘安装了还原程序，个人资料/代码请存于非系统盘，避免掉电丢失；

目录

CONTENT

01

实验课程总体介绍

02

实验目的

03

实验环境

04

实验内容



实验课程安排与考核标准

实验课程共有**28个学时**，总成绩为**40分 (40%)**

实验项目 (30分)

题目从单一知识点到综合开发，分数比例根据课程平台设置

期末检查 (5分)

在实验题目中随机抽取2~3个，从“算法设计、调试能力、代码规范、测试方法”四个方面考核实验内容掌握情况，并现场检查程序运行情况。

实验报告 (5分)

需提交一份实验报告，参照提供的报告模板，于实验课结束一周内提交**电子版**实验报告。



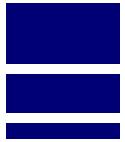
评分标准

编程题目评分项	评分标准
源代码+运行结果 (100%)	OJ课程平台通过 (90%)
	程序符合编码规范 (10%)

	评分项	评分标准
实验报告 (5分)	总体设计	2分
	函数设计	2分
	系统运行结果	1分



- 严禁抄袭，一经发现，双方当次实验记零分；
 - 课后一周内提交，收作业系统、OJ课程平台设置DDL，超时将无法提交；
 - 通过“作业补交通道”提交的作业，按得分80%计算分数。
-



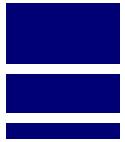
实验目的

- 熟悉**Code::Blocks**集成开发环境
 - 掌握C语言程序的运行过程
 - 了解编码规范的重要性
 - 掌握强制类型转换及输入/输出方法
 - 使用变量、操作符与表达式的运算编程；
-



集成开发环境介绍

IDE	平台	简介
CodeBlocks	Linux、Mac、Windows	开放源码的全功能的跨平台C/C++集成开发环境，开源、免费、轻量级。
Dev-C++	Windows	轻量级 C/C++ 集成开发环境，集成了MinGW中的GCC编译器、GDB调试器和 AStyle格式整理器等众多自由软件。
VSCode	Linux、Mac、Windows	跨平台源代码编辑器，集成了所有现代编辑器所应该具备的特性，插件丰富，调试功能强大，内置Git；初始配置复杂，插件依赖多。
Microsoft visual studio	Windows	Visual Studio 可以用来创建Windows应用程序和网络应用程序，也可以用来创建网络服务、智能设备应用程序。功能强大，安装体积大，占用系统资源多，启动慢。



集成开发环境 Code::Blocks

□ 下载地址(最新版本25.03)

<http://www.codeblocks.org/downloads/>

□ 非常小巧的单文件安装包，安装简单

□ 免费开源的跨平台集成开发环境

□ 支持多种编译器：GCC

□ 强大的调试功能：完整的GDB支持

□ 智能的代码编辑器



Microsoft Windows (64 bit, default)

File	Download from
codeblocks-25.03-setup.exe	Sourceforge.net or dAppCDN.com
codeblocks-25.03-setup-nonadmin.exe	Sourceforge.net or dAppCDN.com
codeblocks-25.03-nosetup.zip	Sourceforge.net or dAppCDN.com
codeblocks-25.03mingw-setup.exe	Sourceforge.net or dAppCDN.com
codeblocks-25.03mingw-nosetup.zip	Sourceforge.net or dAppCDN.com

实验环境

1. 新建项目

2. 编辑代码

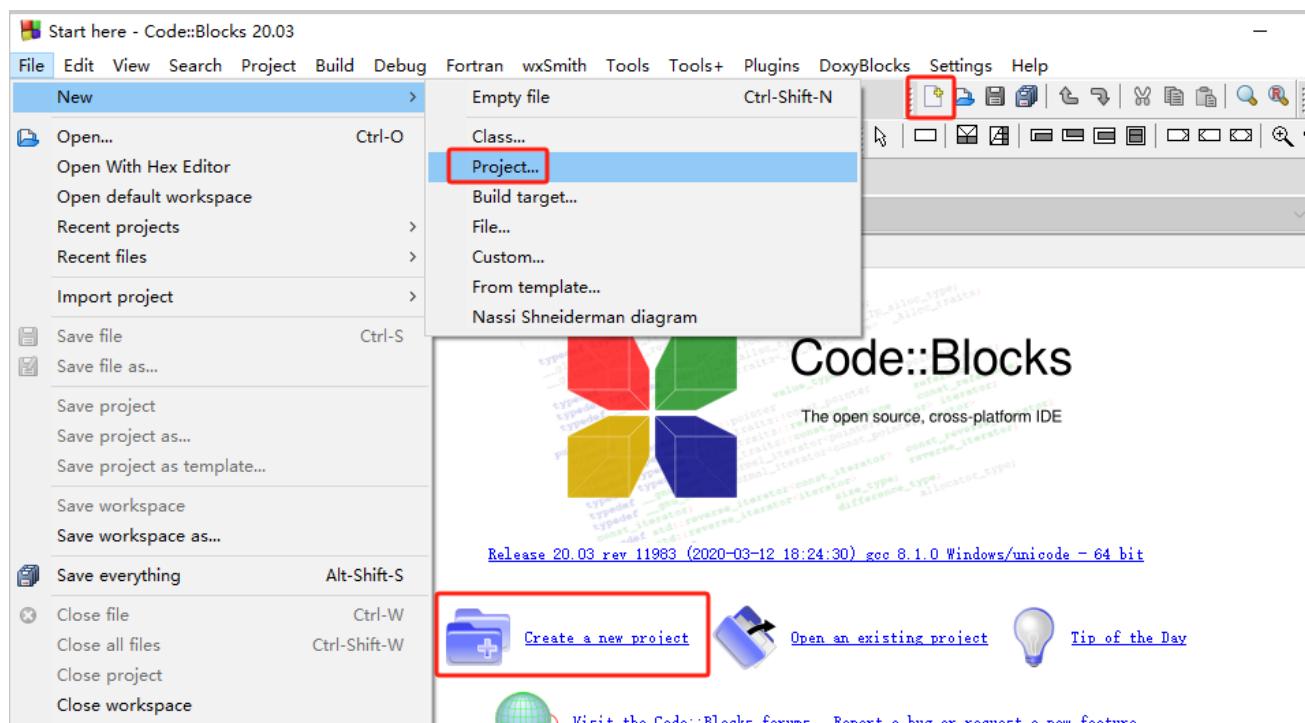
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 点击左上角菜单栏的 “File→New→Project”
- 点击工具栏中的图标选择Project... 
- 在起始界面点击 “Create a new project”



实验环境

1. 新建项目

2. 编辑代码

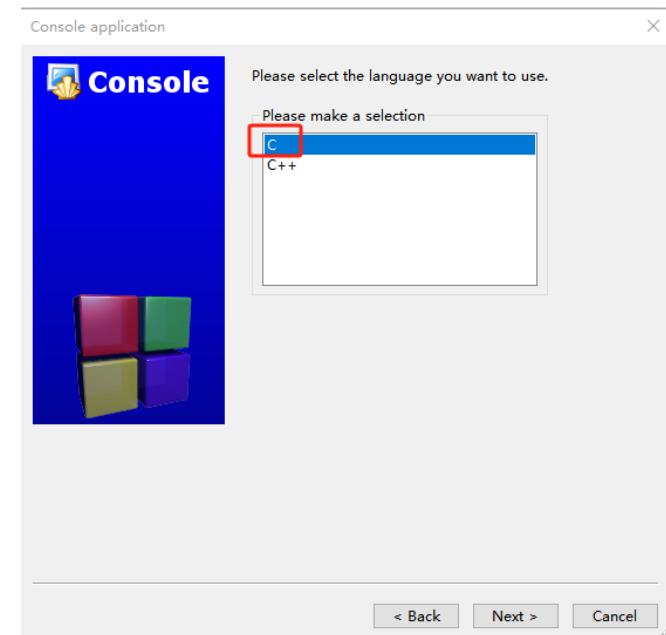
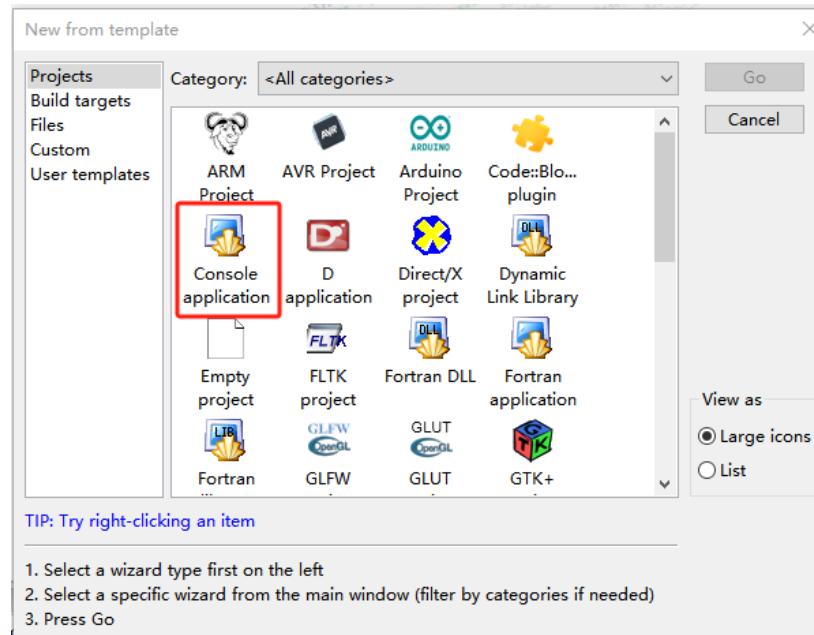
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 选择 “Console application” ；
- 选择 “C” ， 将以C语言的编译环境运行；



1. 新建项目

2. 编辑代码

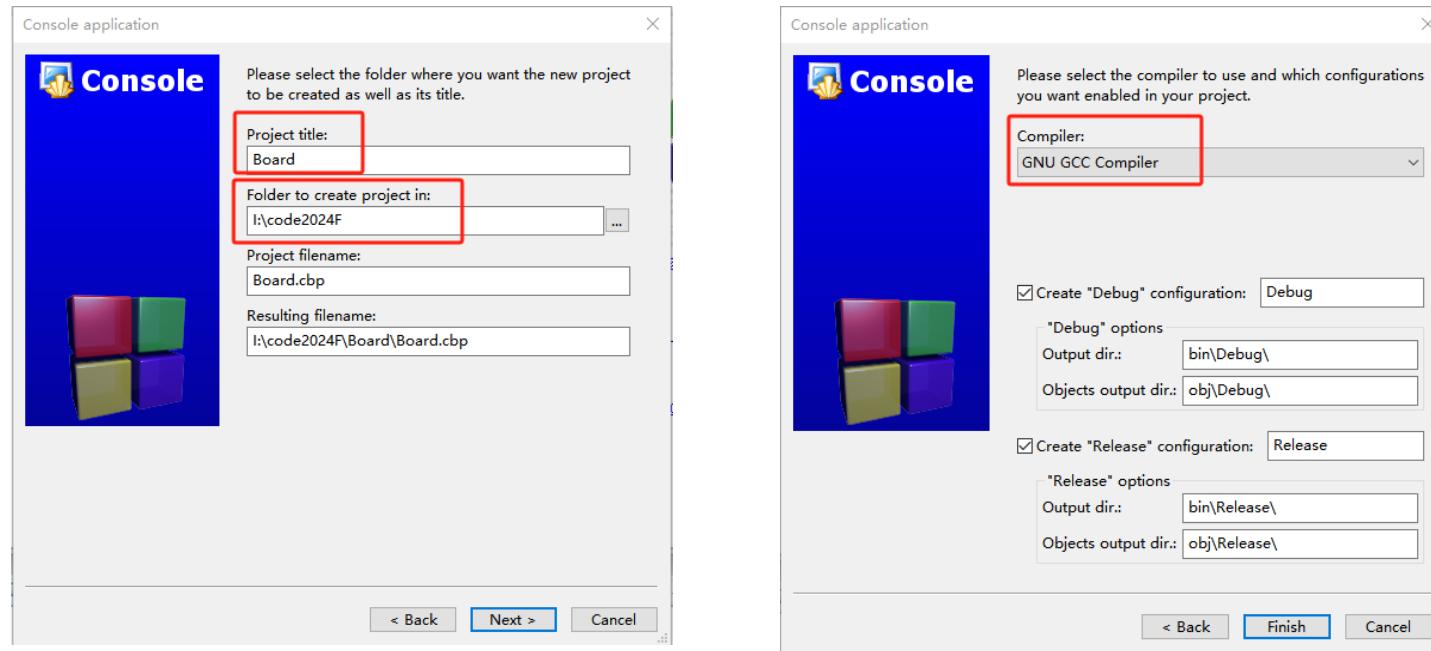
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 填写项目 (project) 名称，选择项目存储位置；
- 选择默认的GCC编译器，点击确定



项目名称和保存路径不要出现中文和空格，否则无法调试

实验环境

1. 新建项目

2. 编辑代码

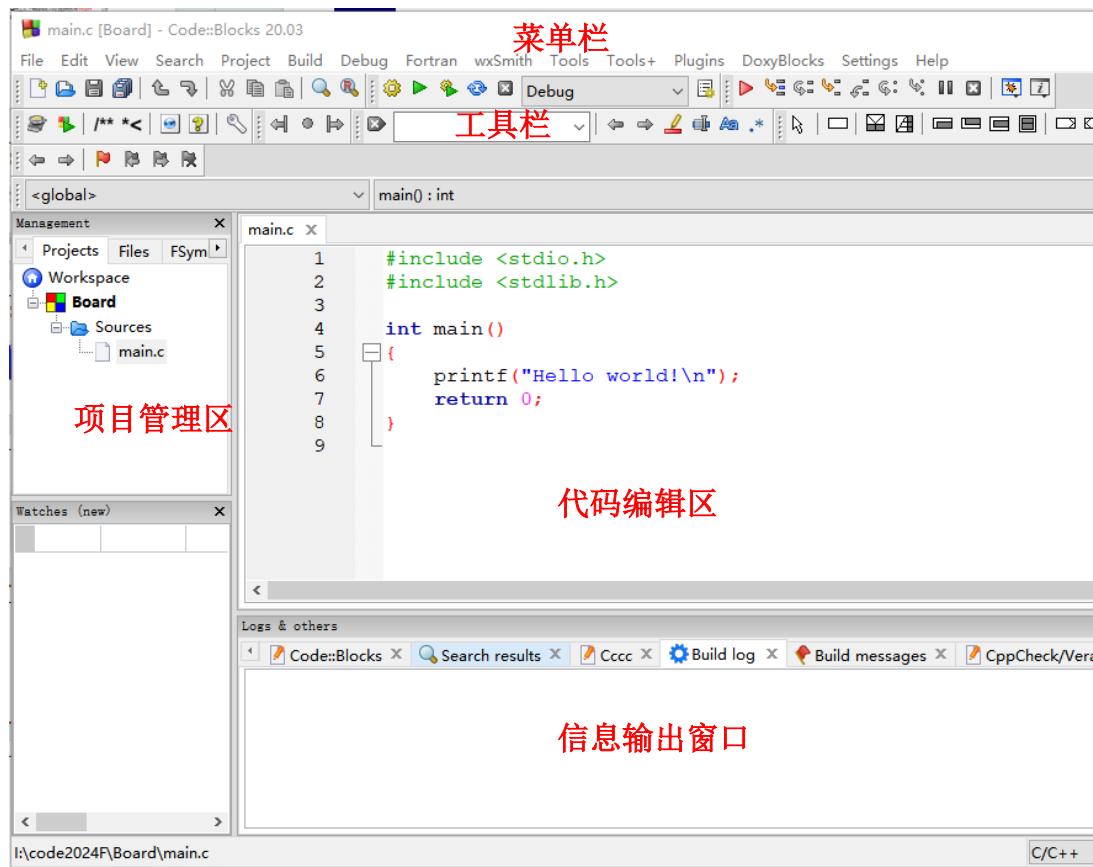
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 双击项目中Sources目录下的main.c文件，在右侧代码编辑框，编辑代码；
- 点击保存按钮，或者使用Ctrl+s快捷键保存



实验环境

1. 新建项目

2. 编辑代码

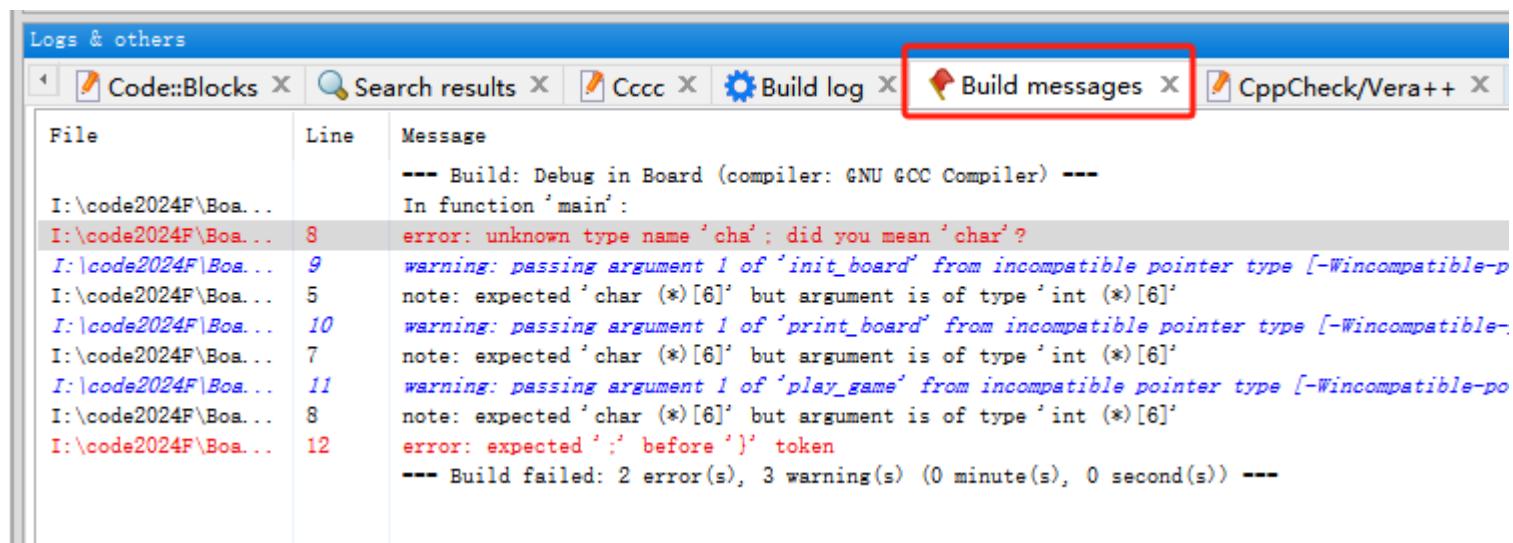
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 点击工具栏里 
- 快捷键方式: Ctrl + F9
- 选择工具栏上的按钮 : Build→Build



File	Line	Message
I:\code2024F\Boa...		--- Build: Debug in Board (compiler: GNU GCC Compiler) ---
I:\code2024F\Boa...	8	In function 'main': error: unknown type name 'cha'; did you mean 'char'?
I:\code2024F\Boa...	9	warning: passing argument 1 of 'init_board' from incompatible pointer type [-Wincompatible-p
I:\code2024F\Boa...	5	note: expected 'char (*)[6]' but argument is of type 'int (*)[6]'
I:\code2024F\Boa...	10	warning: passing argument 1 of 'print_board' from incompatible pointer type [-Wincompatible-
I:\code2024F\Boa...	7	note: expected 'char (*)[6]' but argument is of type 'int (*)[6]'
I:\code2024F\Boa...	11	warning: passing argument 1 of 'play_game' from incompatible pointer type [-Wincompatible-po
I:\code2024F\Boa...	8	note: expected 'char (*)[6]' but argument is of type 'int (*)[6]'
I:\code2024F\Boa...	12	error: expected ';' before ')' token --- Build failed: 2 error(s), 3 warning(s) (0 minute(s), 0 second(s)) ---

1. 新建项目

2. 编辑代码

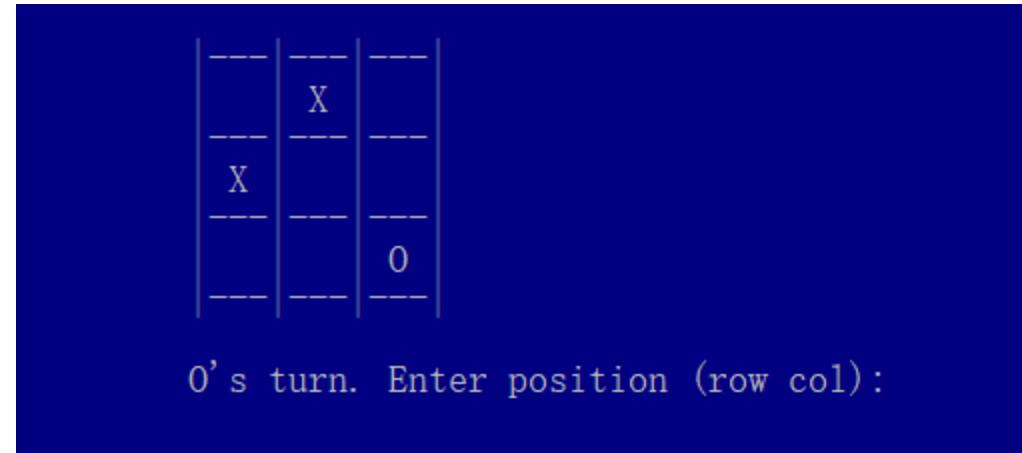
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 点击工具栏里的 
- 快捷键方式: Ctrl + F10
- 选择菜单栏上的按钮 : Build→Run



建议先编译，确定没有error和warning后，再运行。

实验环境

1. 新建项目

2. 编辑代码

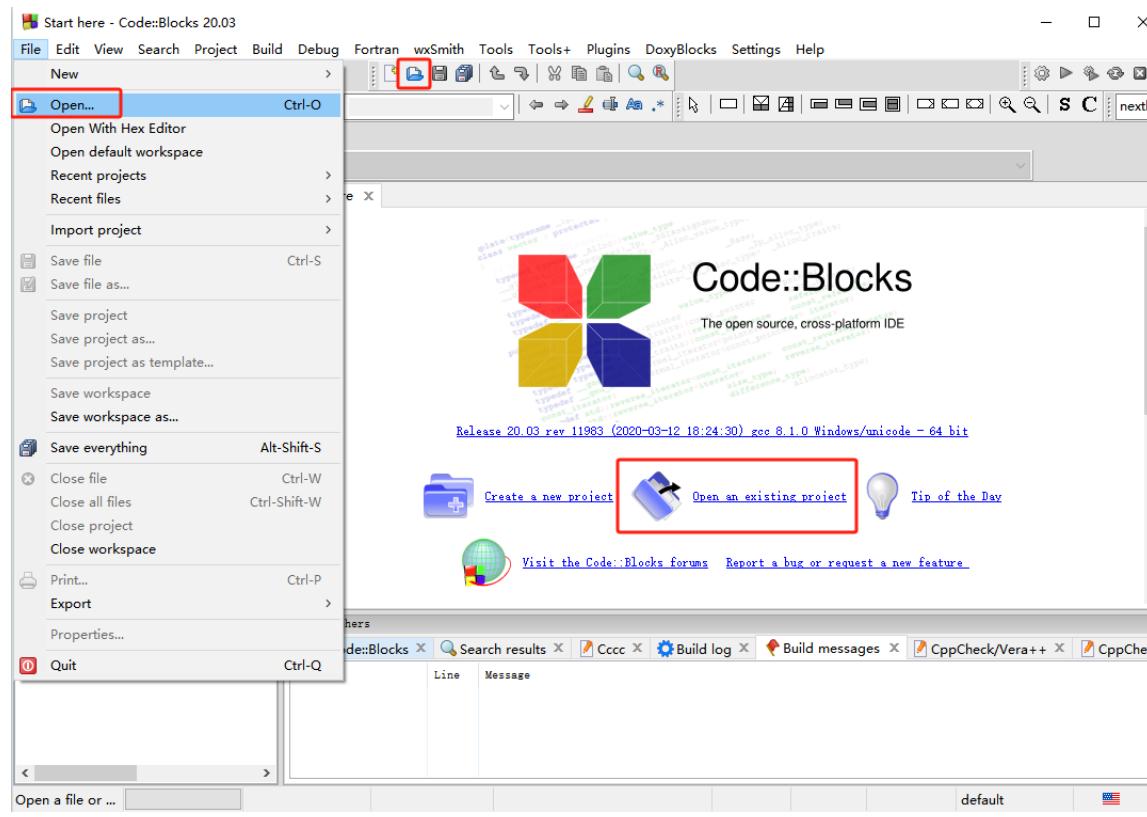
3. 编译

4. 运行

5. 打开新项目

6. 激活项目

- 点击左上角菜单栏的 “File→Open→Project”
- 点击工具栏中的图标选择 
- 在起始界面点击 “Open an existing project”



实验环境

1. 新建项目

2. 编辑代码

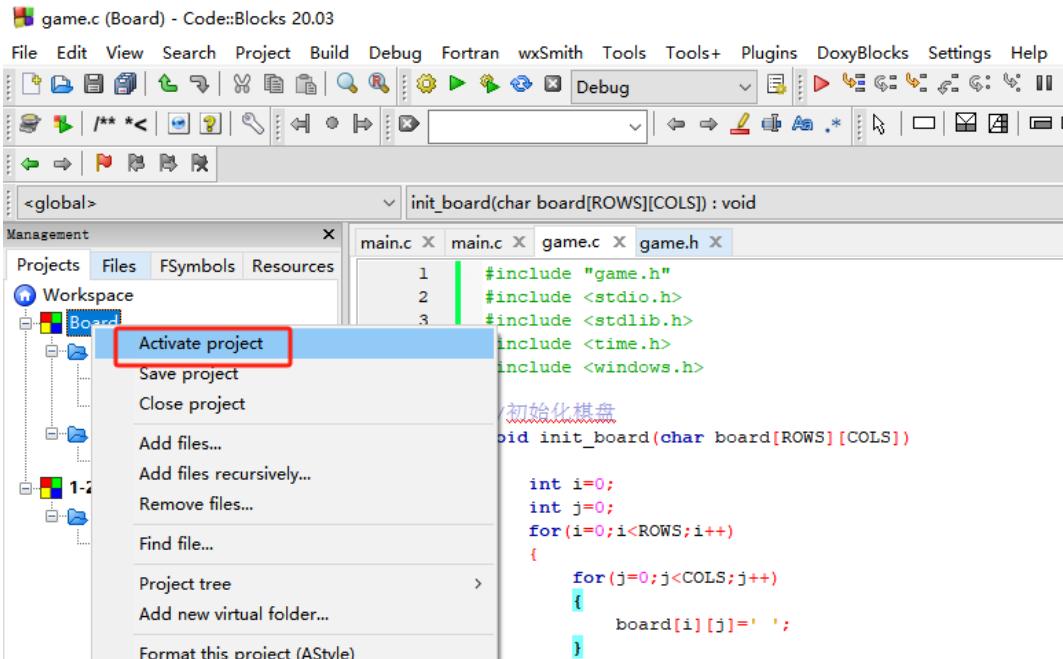
3. 编译

4. 运行

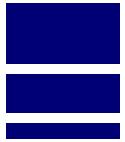
5. 打开新项目

6. 激活项目

- 右键单击，选择“Activate Project”
- 激活的项目呈现黑色粗体
- 可以打开多个项目，但只能有一个项目处于激活状态

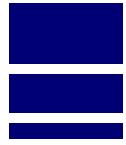


只能有一个被激活的项目



实验目的

- 熟悉Code::Blocks集成开发环境
 - 掌握C语言程序的运行过程
 - 了解编码规范的重要性
 - 掌握强制类型转换及输入/输出方法
 - 使用变量、操作符与表达式的运算编程；
-



程序是如何运行的

双人对弈的井字棋游戏

【游戏规则】X方先行，在一方棋子三个连成一条直线即为胜出，棋盘中9个格子落子满，且没有一方三子连成直线，则为平局。

【解题思路】首先定义一个 3×3 的二维数组board，将其中全部元素赋值为空格字符，然后两个玩家轮流落子，重绘棋盘，判断胜负，不断重复以上过程，直到出现平局或者一方胜出为止。

程序是如何运行的

1. 预处理

2. 编译

3. 汇编

4. 链接

预处理：在这个步骤中，预处理器将源代码中以 # 开头的指令进行处理，例如 #include 和 #define。预处理器还可以执行条件编译，将一些代码包含或排除在编译过程中。

- 宏定义 #define
- 文件包含 #include
- 条件编译

```
#ifdef <标识符>
    程序段1
#else
    程序段2
#endif
```



去掉#include，会怎样？



程序是如何运行的

1. 预处理

2. 编译

3. 汇编

4. 链接

编译：在这个步骤中，编译器将经过预处理的**源代码**转换成**汇编语言**。编译器会进行词法分析、语法分析和语义分析，生成相应的中间代码。

程序是如何运行的

1. 预处理

2. 编译

3. 汇编

4. 链接

汇编：在这个步骤中，汇编器将**中间代码**转换成可执行的**机器码**。汇编器会将每条汇编指令翻译成一条或多条机器指令，并生成**目标文件**。



程序是如何运行的

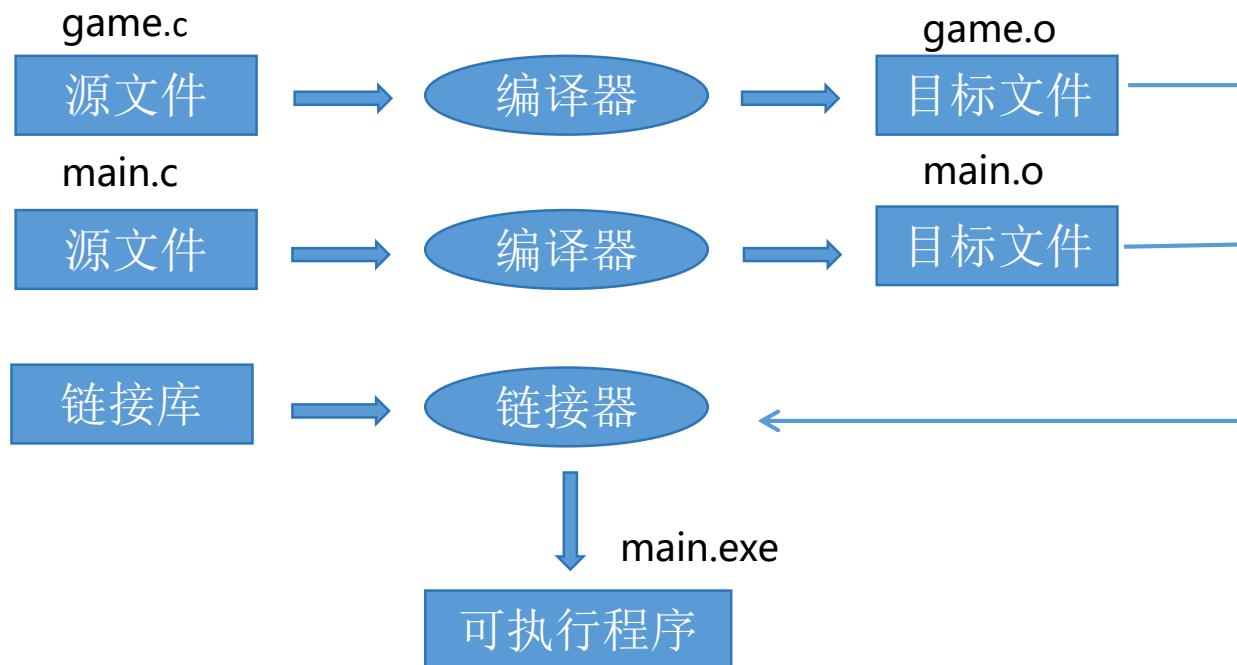
1. 预处理

2. 编译

3. 汇编

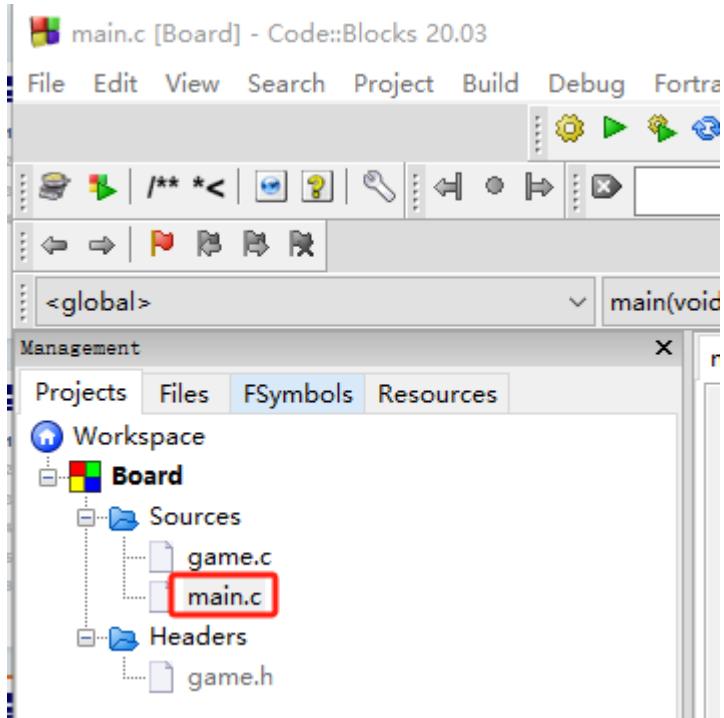
4. 链接

链接：在这个步骤中，链接器将目标文件和库文件组合在一起，生成**可执行文件**。链接器会解决符号引用问题，将代码和数据段放置在内存中的正确位置，并生成可执行文件的元数据（如入口地址和符号表）。



程序是如何运行的

界面显示



文件结构

名称	修改日期	类型
bin	2024/10/12 14:27	文件夹
obj	2024/10/12 14:27	文件夹
Board.cbp	2024/10/12 14:27	CBP 文件
Board.depend	2024/10/12 15:22	DEPEND 文件
Board.layout	2024/10/12 16:19	LAYOUT 文件
game.c	2024/10/12 14:27	C 文件
game.h	2024/10/12 14:27	C Header File
main.c	2024/10/12 16:07	C 文件

- bin目录为最终生成的可执行文件路径
- obj目录为编译过程中的临时文件路径

long (l:) > code2024F > Board > obj > Debug

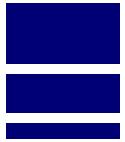
名称	修改日期	类型
game.o	2024/10/12 14:27	O 文件
main.o	2024/10/12 16:07	O 文件

long (l:) > code2024F > Board > bin > Debug

名称	修改日期	类型
Board.exe	2024/10/12 16:07	应用程序



每个project下只能有一个main函数

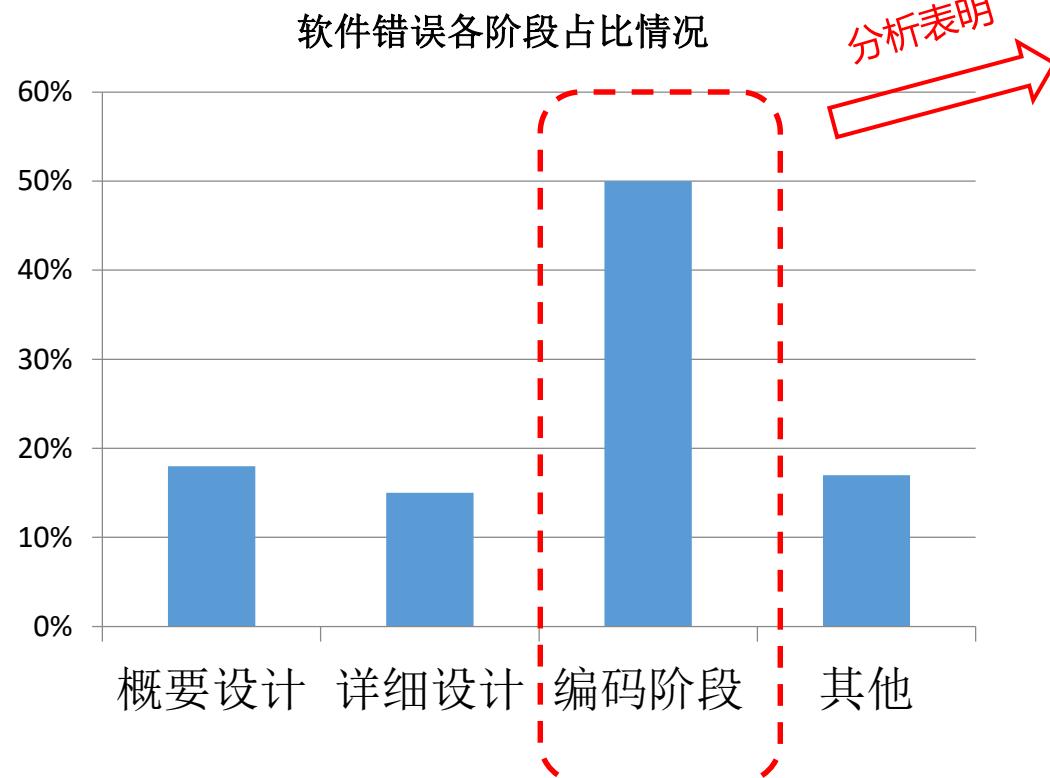


实验目的

- 熟悉Code::Blocks集成开发环境
 - 掌握C语言程序的运行过程
 - **了解编码规范的重要性**
 - 掌握强制类型转换及输入/输出方法
 - 使用变量、操作符与表达式的运算编程；
-

编码规范

Bell实验室的研究资料表明，软件错误中，编码阶段产生的错误占的比例接近50%。



“编码阶段” 错误种类	比例
语法错误	20%
函数(模块)之间接口错误 代码可理解度低导致优化维护阶段对代码的错误修改引起的错误 未严格检查软件逻辑导致的错误	50%以上
其他	10~30%

解决办法

Bell实验室的研究人员制定了**详细的软件编程规范**，并培训每一位程序员，最终的结果把编码阶段的错误率降低至10%左右。

编码规范-命名规范

➤ 项目命名

- 选择有实际意义的单词
- 单词与单词之间应使用下划线分开或首字符大写
- Project、*.c文件、及保存文件的路径，必须具有一定的实际意义；严禁出现中文和空格

experiment3_task1_MoneyChanger
Experiment3Task1MoneyChanger

➤ 变量命名

- 命名形式为xAbcFgh，x由变量类型确定，Abc、Fgh表示连续意义字符串；

内容	类型	不规范的变量名	规范的变量名
身高	float	h	fHeight
年龄	int	a	iAge
数学成绩	float	s	fMathScore

➤ 函数命名

- 必须具有一定的实际意义；
- 第一个字母必须使用大写字母，要求用大小写字母组合规范函数命名，用下划线间隔。

float Male_HeightCompute (float fFatherHeight, float fMotherHeight);

➤ 宏和常量的命名

- 必须具有一定的实际意义；
- 常量和宏定义必须全部以大写字母来撰写，中间可根据意义的连续性用下划线连接。

#define PI 3.14159265

编码规范-程序版式

➤ 缩进

缩进以 Tab 为单位；建议以下情况比上一行缩进一个Tab：

- 函数体相对函数名及{、}
- if、else 等之后的代码

➤ 空行

两段代码，分别完成不同的功能，这两段代码之间可使用一空行，完成逻辑上的代码区分。

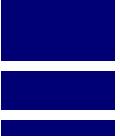
```
float Male_HeightCompute (float fFaHeight, float fMoHeight)
{if(sex== 'm' ) myHeight=(faHeight + moHeight)*0.54;
else myHeight = (faHeight*0.923 + moHeight)/2.0;
if(sports== 'y' ) myHeight*=(1+0.02);
if(diet== 'y' ) myHeight *= (1+0.015);
else myHeight *=0.9;}
```

```
float Male_HeightCompute (float fFaHeight, float fMoHeight)
{
    if ( sex == 'm' )
        myHeight=(faHeight + moHeight)*0.54;
    else
        myHeight = (faHeight*0.923 + moHeight)/2.0;

    if ( sports == 'y' )
        myHeight*=(1+0.02);

    if ( diet == 'y' )
        myHeight *= (1+0.015);
    else
        myHeight *=0.9;
}
```

缩进
空行



编码规范-程序注释

- 文件开头的注释模板

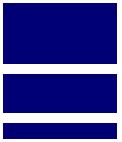
```
*****  
** 文件名:  
** 创建人:  
** 日期:  
** 修改人:  
** 日期:  
** 描述:  
** 版本:  
**-----  
***** /
```

- 函数开头的注释模板

```
*****  
** 函数名:  
** 输入: a,b,c  
** a---  
** b---  
** c---  
** 输出: x---  
** x 为 1, 表示...  
** x 为 0, 表示...  
** 功能描述:  
** 全局变量:  
** 调用模块:  
** 作者:  
** 日期:  
** 修改:  
** 日期:  
** 版本:  
***** /
```

- 程序中的注释模板

```
/*-----*/  
/* 注释内容 */  
/*-----*/  
  
或者  
  
//注释内容
```



实验目的

- 熟悉Code::Blocks集成开发环境
- 掌握C语言程序的运行过程
- 了解编码规范的重要性
- 掌握强制类型转换及输入/输出方法
- 使用变量、操作符与表达式的运算编程；



实验内容

一、代码改错

1. 将输入的华氏温度转换成摄氏温度，要求输出小数点后**两位小数**

```
#include<stdio.h>

int main()
{
    double c;
    int f;
    printf("Input Fahrenheit:") ;
    scanf("%d", &f) ;
    c = 5/9*(f-32) ;
    printf( "\n %d (F) = %2f (C)\n\n ", f, c ) ;
    return 0;
}
```



实验内容

二、编程实现： 1001 模拟运算

输入一个算式（没有空格），遇等号“=”表示输入结束，输出结果。假设计算器只能进行加、减、乘、除运算，运算数和结果都是整数，4种运算符的优先级相同，按“从左到右”的顺序计算。

【输入】一个算式（**没有空格**），等号“=”表示结束输入

【输出】一行

①运算结果 或者 ②“错误的运算符:%c”

【示例1】

1+2*10-10/2=

10

【示例2】

3+2(5=

错误的运算符:(



实验内容

1002 三天打渔两天晒网

某人三天打渔两天晒网，假设他从1990年1月1日开始打渔三天，然后晒网两天，请编程回答任意的一天他在打渔还是晒网。

【输入】 %4d-%2d-%2d

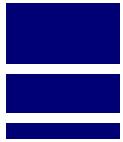
【输出】 一行

`rest` 或者 `working` 或者 `Invalid input`

【示例】

1990-01-05

rest



课程平台使用指南

- 登录网址: <http://10.249.41.7:9000/>, 推荐浏览器: Chrome;
- 初始用户名、密码均为学号, 登录后请修改;
- 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;
- 代码没有点击 “**提交**” 之前, 所有内容没有保存, 请做好本地备份;
- 代码 “**提交**”, 需刷新之后, 才可以查看 “**提交记录**” 和 “**已提交代码**”;
- 只支持单文件。

□ 常见oj错误提示：

- ① Compile Error 编译错误，不符合语法规范
- ② Wrong Output 表示运行结果不正确
- ③ Representation error 一般表示输出格式不正确
- ④ Runtime error 请检查是否有除零、数组越界、eles语句块没有用{}括起来等
- ⑤ Nonzero Exit Status 一般表示主函数返回非0值

FEEL THE MEANING OF THE TRIP

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在



DREAM

MY DREAM WILL NEVER STOP

TAKE YOU ON A TRIP
GO!

请同学们开始实验



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

探索 从未停止