参考教材8.2~8.4；6.1~6.2；4.7

本课结束后，基本结构即将完整，后续进行功能加强

（知识比较散）

编程的问题来源于应用问题，题目也偏应用（算法→代码）

思维重于细节（编译器可以检查细节）

第十周五组上机考试，1h内正确10分，1h后每15min扣一分，演示半小时后酌情给分

周二中午、下午、晚上；周三晚上；周五晚上（13：30准备，16：00准备，18：30准备）

准备30min，考试1.5h

可以按照心情/气候好坏选择时间，五个时间题目不同，人少环境好，助教方便辅导

助教如果人少可以按照代码给分

问一次如果错了扣一分，自己一定要反复测试再交！

范围1~6讲（循环结构未知）

不能带U盘，不能联网

函数递归调用；变量存储；文件包含；宏定义

递归调用算法：

斐波那契数列Fn=Fn-1+Fn-2递推法（定义long long）

f1=f2+f1为f3;f2=f1+f2为f4;……只用两个变量，进行重用

递推法：初值出发，新与旧的关系通过循环计算得到最后值

换一个思路！递归调用

循环都可以通过递归调用思想

函数调用过程：主调函数（外面的）、被调函数（里面的）；参数传递（形参、实参）

断点保护，为形参分配存储空间；被调函数执行函数体；运行到return/末行，

函数的形参空间释放

从main开始运行，从main结束运行，通过调用函数串起所有函数

程序：预编译命令、全局声明、主函数（局部声明、语句）、其它函数定义

调用main函数的下节课讲

阅读习惯：先看主函数，所以写在前面（main是领导写的）

保存断点信息（压栈）

栈：一般先进后出、后进先出，特殊的存储空间

恢复断点：弹栈

递归调用：调用自己

直接递归：func函数调用本身

间接递归：f1调用f2，f2调用f1

每次调用做事不同

自己调用自己实参不同

每次调用都会有断点保护

递归比递推简洁，但是不好理解

int fib (int n){

int f;

if (n==1||n==2) f=1;

else f=fib(n-1)+fib(n-2);

return (f);}

自己调用自己不需要再次声明

每次调用实参不一样

递归调用必须有退出机制！否则死循环堆栈溢出（断点存不下了）

条件：边界条件（退出机制）、递归关系

六行代码！定义函数、变量，边界条件（if、结果），递归关系，return

定义：**假设结果出发**，归纳处后一结果与前一结果的关系直到**初值，用断点保护**；

通过断点恢复根据初始值和递推关系，反推出真实结果

递推算法找不到可以进行递归

反解思路！

压栈过程学习递归关系

高级程序用递归代替循环递推（甚至没有循环，全都用递归代替）

生活中很多

主函数里进行声明

回推调用（压栈）→梳理关系：n！→n-1！→……→1！，若1已知→

回代计算（弹栈）→迭代结果：初值1！→2！→……→n！

如果每次实参不变，不断压栈死循环！一定要变化！

一般格式：

if（最简单情况）{问题答案}

else 问题简化为一个或多个子问题，以递归的方式求解，再组合

相比于循环，牺牲内存换取计算效率！

断点和控制转移也有开销！

用断点意识判断输出是什么！

变量存储类型

变量和函数都有两个基本书写：数据类型、存储类型

数据类型：存储大小和编码方式

存储类型：存储区域！代码区、静态数据区、动态数据区

动态区auto：不要了就释放（动态分配、释放）

静态区static：直到程序结束才释放（始终占据）

寄存器register：临时的（现在微机没有）（存取快，但是只有char、int）

完整定义格式：存储类别 数据类型 变量名

缺省的一律auto

外部变量extern：其值可供其它源文件使用，也在静态区

缺省extern是函数里的概念，变量里概念不一样

栈区stack、堆区heap（都叫动态）；

静态区static

文字常量区（常量字符串）

程序代码区

未说明存储类型，函数内变量auto，函数外变量extern

存储类型决定分配再静态还是动态，影响变量的生命周期和初始化值

花括号是auto里变量的作用范围

初始化值：静态区缺省则0，动态去缺省则随机数

生命周期：变量分配存储单元的时间有效性

定义位置决定作用域

动态区的作用域超不过文件，全局变量在其它文件里可以使用，放在静态区

全局变量：函数外定义

局部变量：函数或块内定义（也可以加static放在静态区）

静态区局部变量：生命期按静态区分配，作用范围不变！区域存在就能访问

不能直接使用，可以间接使用

存储位置可以看出存储类型，也可以用存储类别标识符指定

全局变量可以实现函数之间数值的传递，实现通信；但是占地方，容易被误修改

早期规则：函数定义在main之前不用声明

全局变量的生命期：定义点开始到函数结束

如果定义点在后面，需要在前面加入“外部引用性说明”

释放是个重要的美德，内存资源很宝贵

C程序基本结构：编辑→编译→连接→执行

编译有很多子阶段

预处理命令（最开始）（带有#的）——告诉编译器怎么处理（宏定义、文件包含和条件编译）

文件包含：若干源文件组成，把其它文件合成进来

头文件包含函数的声明

#include文件名可使用两种括号：

<>（仅在系统指定目录中寻找）和”” （先在被包含文件目录查找，再到系统指定目录）

#define 宏定义

不带参数的宏：#define 宏名（标识符） 宏体（字符串）

预编译把源程序中的宏名替代为宏体，称为宏展开

#undef 把宏的作用从那之后关掉

宏展开可以嵌套！

（宏不扩展””里的宏名）

带参数的宏：#define S(a,b) 2\*a\*b （展开更加灵活）

计算式直接展开（如S(a) a\*a，S(a+b)运算顺序会错），所以定义的时候打括号！