C语言的复习作用：

是为了帮助同学们回顾之前学习的知识点，辅助我们本次课程中后面所涉及到的内容

C语言的框架主要分为以下十个部分：

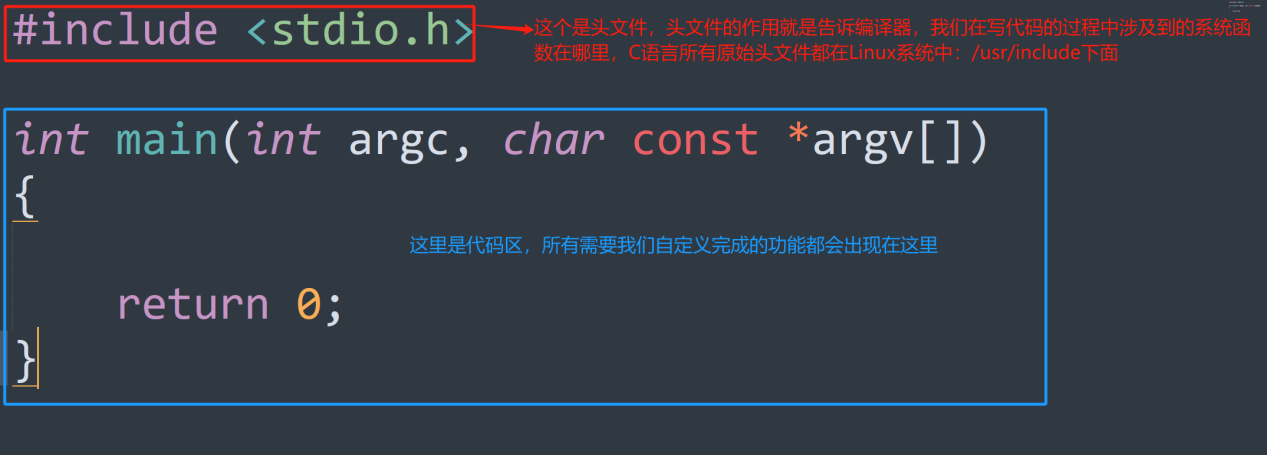
* C语言的框架
* 数据类型
* 变量
* 运算符
* 常量
* 控制流
* 函数
* 指针
* 数组
* 结构体

注意： 除去这十个部分以外，C语言还有很多额外的内容也需要同学们掌握，比如很多关键词的使用、联合体、枚举体、强制转换、隐式转换等内容

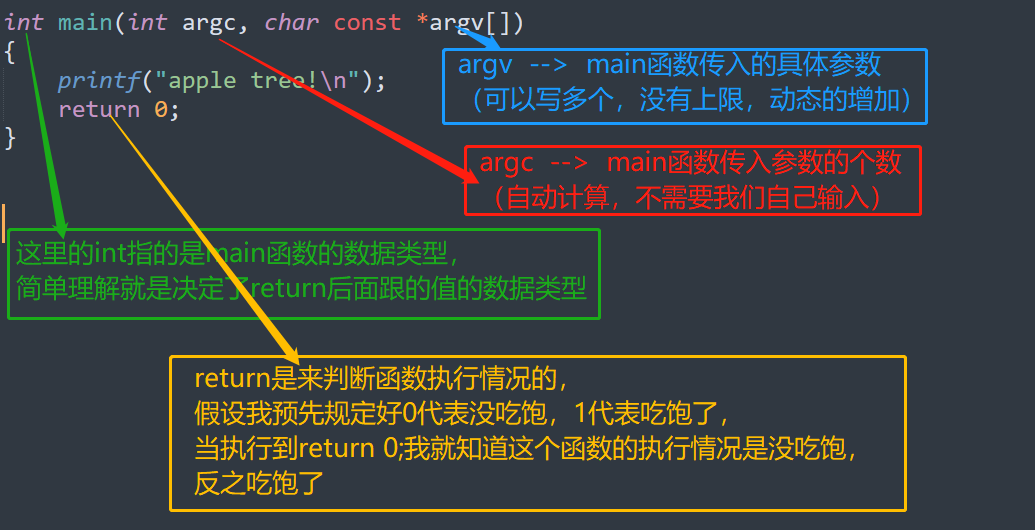


**C语言的框架**

* C语言文件的后缀名： .c .C
* 完整的C语言文件包含的内容：



* 1. main函数：main函数是程序运行时的入口，也是程序结束时的出口



**数据类型：数据类型描述的是一个变量在内存中占用多少个字节的空间**

* 基本数据类型（从小到大）：

char short int long float double bool(true/flase) void

1 2 4 8 4 8 1(所占字节数，以64位系统为例)

* 非基本数据类型：数组、指针、结构体......
* 各种数据类型的输出格式：

%d 十进制有符号int（有正负的）

%u 十进制无符号int（>0）

%hd 十进制有符号shorth：half

%f 单精度浮点数float

%lf 双精度浮点数double l： long

%s 字符串

%c 单个字符--> ascii

**变量：在内存中连续申请n个字节的空间，然后使用变量名间接访问这片空间。**

* 定义变量： 数据类型+自定义的名字
* 自定义的名字就是变量名，自定义的名字不是随便起的，要配合C的一套规则
* 规则：

规则A： 只能使用数字、字母、下划线组成。 例：int a+5int a 6;

规则B： 不能以数字开头。 例：int 5a

规则C： 不能与系统关键词重名。 例：int return(错误) int Return(正确)

**内存分配原则**

* 在分配空间时，内存地址一定是连续的。分配空间时，内存上的空间一定是空闲的（之前没有被使用的），申请下来的空间位置是随机的

**全局变量和局部变量**

* 全局变量：定义在{}外的变量
* 局部变量：定义在{}内的变量
* 两者区别：

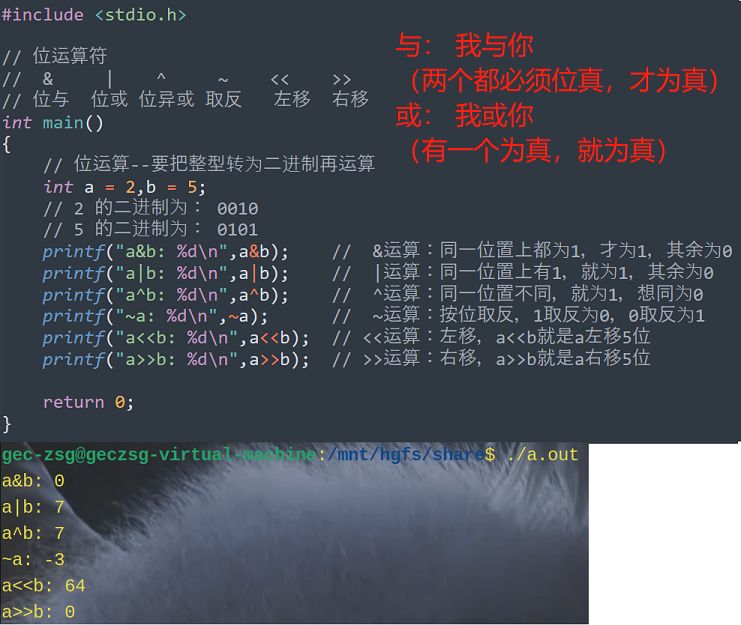
生命周期不同：局部变量由于被{}圈住，出了{}将没有意义，所以局部变量的生命周期是从定义开始，出{}时结束，全局变量由于不在{}内，所以不受{}的限制，所以全局变量的生命周期是从定义开始，程序结束时结束。

初始值不同： 局部变量定义时没有赋初始值，此时局部变量内的存放的值是随机的，全局变量定义时没有赋初始值，此时局部变量内存放的值是0

重复命名的问题： 在一个C语言程序中，可能会出现多个相同名字的局部变量，但是不会出现相同名字的全局变量

**运算符**

* 算术运算： + - \* / %
* 赋值运算： += -= \*= /=
* 关系运算： > < >= <= == !=
* 位运算： & | ^ ~ << >>



* 1. 逻辑运算： && （并且） ||（或者） ！



* 1. 自增、自减： ++ 、 --a=1, a++; ++a;
  2. 三目运算：条件判断?语句1：语句2
  3. 运算符的优先级问题：当你需要把某个运算的优先级提高，可以使用 () 把该运算圈起来，因为（）的运算优先级是最高的

**常量**

* 常量：常量就是固定的值，在程序运行期间从开始到结束都不会改变的值，就叫常量。常量可以是各种基本数据类型的数据，例如：整数常量，字符常量，浮点常量等等，也有枚举常量等等。常量的值一旦定义好了以后，在程序中任何情况都是不能修改的。
* 定义常量的两种方式：一是使用 宏定义。二是使用 const 关键字。
  1. 宏定义：#define LENGTH 8
  2. const关键词：const int WIDTH = 6;



**控制流**

* 分支控制流： if...else...
* 选择控制流: switch...case...
* 循环控制流: for 、 while 、 do...while

**函数**

* 函数：函数就是一个模块实现过程的封装
* 函数的种类：

main函数

* 系统函数：由系统（Linux）直接提供
* 库函数： 由后来工程师移植过来的函数
* 自定义函数：程序员自己写的函数
* 自定义函数：

步骤：

1. 确定你自己写的这个功能是什么

2. 根据你要写的功能确定：
函数的数据类型+函数的名字(需要尽量贴合功能)+函数的参数

3. 写代码

4. 确定返回值（作用：用来返回函数的执行结果的，辅助调用者来判定函数的执行）

5. 写清楚注释

6. 记得写函数的声明

注意： 自定义函数完成以后，函数的声明是有必要写的，否则会出现waring

**指针**

* 指针：指针是用来存放地址的，所以有句话：地址即指针，指针即地址
* 指针的类型： 指针跟变量其实本质上都是一样的都是用来存储数据的，所以他具备了变量所具备的所有特性，有点特殊的是所有类型的指针变量所占内存的大小均为8字节
* 指针变量的定义：数据类型+指针变量名int \*p;
* 指针变量的赋值： 需要用到取地址符&p = &a;（a是int类型的变量）
* 怎么通过指针去访问地址所指向的空间的值？ 解引用：\*q等价于a；
* 指针变量需要注意的问题：空指针和野指针int \*p; \*p=2;

**数组**

* 数组其实就是一个集合，该集合中可以含有多个相同数据类型的变量。
* 数组中每一个成员既可以是基本数据类型，也可以是非基本数据类型。
* 数组的定义：数据类型 数组名[元素的个数]int arr[10];
* 数组的大小： arr数组的大小4\*10=40字节
* 数组的赋值：

int A[5] = {1,2,3,4,5}; --> 编译通过

int A[5] = {1,2}; --> 编译通过，没有赋值的成员都是0。（不是随机值）

int A[5] = {}; --> 编译通过，没有赋值的成员都是0。（不是随机值）

int A[] = {1,2,3,4,5}; --> 编译通过，初始化列表决定了下标等于5

* 1. 指针数组、数组指针和多级指针

指针数组：本质上是一个数组，成员都是指针

数组指针：本质上是一个指针，存放的是数组的地址

多级指针：就是指针存放着指针的地址

* 1. 数组跟指针的关系：数组的名字就是这个连续空间的起始地址，也就是首元素的地址，arr数据组名等价于&arr[0]

**结构体**

* 结构体其实也是一个集合，该集合中可以含有大量不同数据类型的变量。
* 结构体中每一个成员既可以是基本数据类型，也可以是非基本数据类型。
* 结构体的定义：
* 关键词：struct
* 模型：

struct 结构体名字{

/\* 结构体的组成变量 \*/

....;

}; --> 定义完结构体之后，记住{}后面有一个分号。

* 1. 结构体变量的使用

struct person{

char name[10]; //姓名

int age; //年龄

char tel[12]; //电话号码

};

struct person g; // 该变量由三个数据类型组成。

* 访问结构体变量里的成员需要: . g.age = 18;

结构体变量跟普通变量其实也是一样的，普通变量所具备的特质结构体变量都有。

**练习：**

冒泡排序

使用C语言完成功能： 打印九九乘法表

gcc main.c -o main

gcc: 编译工具包

main.c:需要被编译的对象

-o:output输出二进制文件（程序）

main：二进制文件的名字

流程：

1. 创建.c文件
2. 写代码
3. 保存
4. gcc编译
5. ./运行