明年会有免修改革

基础差不是自己的问题

笔试

各有十个程序填空（程序抠空填，不知难易，不唯一）和程序阅读，每题2分，闭卷，90min

往年是20道阅读（15行左右）

填空黄老师出，但是答案不一定唯一（可能是一整行，也有可能是一小部分），给运行结果

阅读题和往年形式一样

13号 19~21 一教101（70人）、201（96人）

位操作没学不考（汇编再学）

考文件操作、结构体、**指针**（高级、数组、行、函数）、字符串、数组、函数（参数、递归）

宏定义、变量存储类型、**命令行参数**、分支、switch、循环、break、continue控制……

死记硬背的东西不考

所有答案都写在试卷上，答题纸是草稿纸，会收上去！

空格只要有就行，不计较几个空格，精细表达也可以

填空语句大小写不计较，但是阅读题计较

大空填在空里，填不下指出去

答疑网络学堂、邮件、62792710、罗姆5-104（预约，中关村国家实验室更多，北京西北角）

助教大多是老师博士生，除了考试题都知道，就在隔壁

条件编译（不考，但是以后会用很多）

编译：代码→机器码

条件编译不是条件执行！

条件代码：控制代码哪些编译哪些不编译

用途：调试，修改一点就改一大片！

原来：注释掉、删掉

条件编译：清除得更简单

格式：（#：预编译——文件包含、宏、pragma warning、……）

#ifdef 标识符

程序段1

#else

程序段2 （可以没有）

#endif

程序段1和程序段2就一段参加编译，看标识符怎么定义

在前面define一个标识符，就程序段1，否则程序段2

或

#if 表达式

程序段1

#else

程序段2

#endif

C语言简洁，习惯阅读和书写，优雅

反复上机练习，上课教方法

编程高手很多职高毕业

逆向工程找漏洞：二进制里看出漏洞，技巧性高

漏洞：不周全的地方，指针引入了很多漏洞（不受编译器影响）

编程=算法+数据结构+语言及开发工具

程序=算法+数据结构（两者不完全独立）

算法对数学思维要求高

语言熟悉后基本靠自学，学起来容易

算法用熟了就快

类型差别：告诉编译器分多大的存储空间（sizeof，编译器不一定）、编码怎么编（编译）

学汇编语言需要学习编码方式

数组：方便寻址（从而方便循环）

结构体：关联在一起的信息（排序）

指针：跨域访问（不受变量名约束，但也带来不确定性）

变量存储类型：动态区变量、静态区变量（局部、外部/全局）

静态区变量只初始化一次，之后不执行！

局部变量加static，改变生命期（到结束）；全局变量加static，改变作用域（限制在文件内）

输入输出：函数实现（细节问题讨厌）

实际应用，标准很重要（不同标准的使用不一样）

程序结构：预编译指令（#）、函数（最少有main）、语句

程序由不同文件组成

文件说明：在开头，说明干什么用

注意编程习惯！

模块化设计思想：逐步细分到叶子结点

实参和形参“双向传递”——共享空间

宏：锦上添花，符号常量

算法：穷举搜素、递推、递归、打擂台、整数分解、素数、孪生素数、排序……

算法是无止境的

算法会了，编程就会了

编程四部曲：编辑→编译→链接→执行

编辑：vs、txt、word、edit……（打字，存储ASCII/二进制）

编译：编译器（代码→机器码）

链接：obj+标准函数→exe

执行：通过控制台命令（dos等）实现

vs里main函数参数传递有界面

学习过程出错好事，见得多才会不怕

第一遍输入不出错概率很小

但是快速找到错误是能力

产品形态：可执行文件/库函数

调试和算法都靠交流

算法设计需要一定灵感，写代码是很机械的活

爱上代码！