中国科学技术大学

软件学院

复试指南（2020 版）

前 言

首先，恭喜大家来到复试阶段，当你进入复试，可以说已经离成功不远了。行百里者半九十，但请不要被初试的胜利冲昏了头脑，还需继续努力，拿下复试。

科大软院是一个温馨的大家庭，有很多热心的学长学姐，他们考完试不忘学弟学妹，会尽量留下自己的经验，供后来人参考，工作了也不忘学弟学妹，会经常给大家推荐一些好的岗位。但是，总有机构对科软这块大蛋糕眼红，想方设法想从中分一杯羹。这些机构拿着一本粗制滥造的资料，动不动卖 99，甚至 199，并时常贩卖焦虑。过去科大软院所有的考研群全由考研机构把持着，这些机构为了卖资料，宣传考研辅导班，会经常宣传一些不实数据。如在 19 年，科软改专业课已是人尽皆知的事了，而这些考研机构还打包票说不会改，只为多卖几本资料。在复试前夕，这些考研机构又在宣传虚假的分数，他们干着这些事，却赚的盆满钵满。

为了给后来的考研人一个可靠的信息交流平台和信息搜集渠道，我建了这个群 243125814，我希望给所有想考科软的人一个公益的考研交流平台。当然，有人会质疑为什么是收费群，我只想说，不是我收费，是腾讯要收费。3000 人群每年需要一共 240+380=620 的会员费。每次进群费一般只有一毛或者两毛。结余的钱也就几十块钱，大部分都以红包的形式重新还给大家了。明年的新群主，我希望你也不要靠这点入群费赚钱，以后有你赚大钱的地方，不要为了蝇头小利丢失科软的光荣传统。

这一年多来，我觉得我做的还可以，尽量为一些想考科大软院的同学答疑解惑。20 年的考研马上就结束了，还剩复试。科软这块蛋糕太大了，总有人盯着，拿着一些粗制滥造的资料，动辄 99，199。要不就是先给一份免费的资料，结果让你报班，这种训练营，那种训练营。

看着 20 的一路走来，还想帮你们最后一程，我将搜集来的各种资料，结合 20 年最新考研形式，去粗取精，加上自己的理解，编辑成了最新的 20 版复试指南，希望对你们复试有所帮助。希望你们上岸后，能有人将这一传统传承下去，给 21 的一些指导，但不要以此盈利。

考研备战是一个辛苦的过程，特别是面对中科大这样一所老牌 985 名校，以及十分不友好的专业课，但是请你相信，风险与收益并存，当你收到中国科大录取通知书时，请欣然接收来自同伴们羡慕的眼神和家长赞许的目光，这份荣誉对得起你付出的辛苦。

整本复试指南将分成以下几个板块

1、软院复试情况介绍

2、经验帖分享

3、机试指南

4、专业面试

5、英语面试

6、科软生活总结

7、科软宿舍图

软院复试情况介绍

1、关键时间节点介绍

初试成绩查询时间：

19 年出分时间是 2 月 15 号，可以大致推测出今年初试成绩查询时间为 2020 年 2 月 10 日。

复试线：

根据往年经验，中科大将在 3 月 5 号左右公布复试线。

志愿填写：

复试前在在信息化平台进行注册，并填写入学方向（非常重要） 地址：http://enroll.sse.ustc.edu.cn/ssers2014/stupages/regeist.aspx，其中合肥只有软件设计方向，苏州会有软件设计，信息安全，嵌入式，大数据人工智能四个方向，大数据是最热门的专业，无论分多高都有调剂的可能，所以请谨慎选择。信息安全和嵌入式选择的人较少。

复试时间：

19 年为 3 月 18 号和 19 号复试。基本上每年都在这几天复试。复试结果一般三天内公布。如果复试通过，会显示计划录取。

通知书：

6 月底发放录取通知书

1. 通知书上写的有关于报道及开学所有注意事项，请务必仔细看。
2. 通知书上写的有你是几等奖学金，没写就是没有

开学：9 月初，可凭录取通知书购买学生票

体检说明：这就是开学后的事了，不要操心。

2、进七年录取情况统计

年份	报考人数	初试通过人数	复试线	一志愿录取人数	复试录取比	总体报录比
2014	未知	约 100	国家线	约 100	约 95%	无法计算
2015	未知	106	国家线	101	96%	无法计算
2016	约 430	162	265	140	86%	33%
2017	约 580	210	295 分	198	95%	34%
2018	约 850	238	310 分	220	92%	26%
2019	约 1300	348	310 分	340	98%	26%

声明：本数据来自学院官网公布及复试名单和录取名单分析，部分来自已录取学长估计，具有一定的真实性和参照性，数据仅供参考。

3、复试地点

复试分为面试与机试，面试在南区，机试往年都在东区。

4、酒店指南

从科大南区东门进的话，会走较长的一段路才能到软件学院大楼，但是东门旅店多，小吃也多，晚上可以选择住在东门。从科大南区南门进的话，比较近，但是南门是科大花园小区，南门附近没有旅店。

当然，也可以选择科大东区附近的酒店。 在合肥上学的话，住本校就可。

5、复试流程(以下为 2019 复试流程，复试前会公布最新流程)

考生需在复试以及录取过程中注意并办理以下事宜：

1、 复试程序：所有考生务必完成以下前 5 项内容

（1） 参加考生动员大会：软件学院大楼 103 教室；

（2） 材料审核、签署协议：软件学院大楼 101，102 教室；

（3） 专业综合面试：软件学院大楼 5 楼各面试室；

（4） 英语听说面试：软件学院大楼 4 楼各面试室；

（5） 专业基础评测（机考）：科大东区教二楼 8 楼，按规定时间提前半小时到达，验身份证入场；

（6） 领取材料、相关咨询等：软件学院大楼一楼大厅；

（7） 填写 2020 级联合培养项目报名表（仅限申请联合培养的考生）：软件学院大楼一楼大厅。

特别说明：报名申请"中科院计算所'寒武纪'"或"中科院电子所"联合培养项目的考生须另行参加所系面试

2、 面试分组序号查询：考生可于复试前一天下午开始在软件学院招生系统，或复试当天在软件学院大门外现场查询面试分组序号。

3、 录取过程：按"计划录取->拟录取->正式录取"的流程确定正式录取名单：

（1） 预计在复试后 3 天内在软件学院招生系统上公布本批次复试的计划录取名单；

（2） 拟录取的考生经审核以及报学校上级主管部门审批通过后，成为 19 级正式录取研究生。预计 6 月中下旬（我校本学期结束前）由我校研究生院统一寄发录取通知书。

下面是 2019 年具体流程，仅供参考

注意:所有考生必须在所安排复试单元的当天完成以下 1-5 项工作(如,复试单元为 3 月 18 日上午和 3 月 18 日下午的所有考生必须在 18 日当天完成以下事项，18 日当天不接待复试单元为 19 日的考生)

分项	时间安排	地点	内容	具体事项
1	3 月 18 日（19 日）7:20-12:00	软件学院大楼一楼大厅	领取材料相关咨询	需领取： 表 3《复试注意事项与复试流程》（本表）
			申请联合培养项目	**（仅限申请联合培养的考生）** 需填写： 表 4 《2019 级联合培养项目报名表》 注：本表为联合培养项目报名表，其他国际合作项目具体申请时间及报名方式将另行通知。
2	3 月 18 日（19 日）7:40-8:10	软件学院大楼 103 教室	考生动员大会	学院领导介绍学院以及本次复试基本情况
3 注：此部分需按流程依次进行	3 月 18 日（19 日）8:10-10:30	软件学院大楼 101 教室 102 教室	材料审核签署协议（按分组进行）	需交验： ①身份证原件与复印件 ②（应届生）学生证/（历届生）本科毕业证和学位证的原件与复印件 ③本科学业成绩单原件 需领取： 表 1《2019 中国科大攻读硕士学位研究生情况表》（装订好相关附件） 需签署： 表 2《自筹经费攻读硕士学位研究生协议书》 **（此协议当场签署，勿带走）**
	3 月 18 日（19 日）8:30-12:00 13:30-17:00	4-5 楼 2、3 号楼梯口	身份认证	需交验：身份证原件；
		5 楼各面试室	专业面试（按分组进行）	需交给专业面试组老师： 已核对并补充完成的《2019 中国科学技术大学攻读硕士学位研究生情况表》及附件；
4		3-4 楼	身份认证	需交验：身份证原件；

	3 月 18 日（19 日）8:30-12:00 13:30-17:00	1 号楼梯口		
		4 楼各面试室	英语面试（不按分组进行）	按身份认证处的安排进入英语面试室进行面试 **（无需材料审核可直接凭身份证参加，进入面试室后须在签到表上签到）**
5	3 月 18 日（19 日）19:00-20:00	科大东区教二楼 8 楼（考场位置见现场通知）	专业基础评测（机试）	需携带：身份证原件； （具体复试名单复试当天可在软件学院一楼大厅或科大东区教二楼 8 楼查看，**请按规定时间提前半小时到达验证入场**）
6	3 月 20 日 8:30-12:00	软件学院大楼 102 教室 103 教室	中科院计算所寒武纪联合培养 中科院电子所联合培养 **所系面试**	**（仅限复试当天提交过“联合培养项目报名表”并申请该项联合培养项目的考生）** 需交验：身份证原件

备注：

（1） 第 3 项的专业面试必须按照“材料审核－>身份认证－>专业面试”的顺序进行；

（2） 第 3、4 项无先后顺序，第 4 项的英语面试无需材料审核即可直接凭身份证参加，考生可根据现场和自身实际情况自行安排先参加哪一项面试；

（3） 材料审核和申请联合培养项目必须在当天上午完成，安排在上午复试单元的考生优先进行材料审核；

（4） 专业面试和英语面试均为安排在本复试单元（以半天为一个复试单元）的考生优先进行。闲暇时方可安排其他复试单元的考生进行。

复试名单及复试分组

所有第一志愿报考我院软件工程硕士（含专业学位和科学学位）并过**复试分数线**的考生均参加本批次复试，具体名单请考生登录我院招生系统查询。达到我院专业学位分数线未达科学学位分数线的科学学位一志愿考生可直接参加专业学位复试。截止到 3 月 15 日下午 17:00 仍未在我院招生系统上注册的第一志愿考生视为放弃复试处理。具体的复试时段及分组安排将于 3 月 17 日下午在招生系统上开放查询，考生也可在复试当天在软件学院大楼内查看。

专业方向调整

考虑到复试当天进行的考生动员大会上，学院领导将介绍各专业方向和联合培养项目，学院招生系统将在复试当天及后 1 天开放本批次复试考生的专业选择权限（2019 年 3 月 20 日下午 5 点关闭本批次复试考生的专业选择权限）。请各位参加复试的考生及时登录学院招生系统调整自己的专业方向。便于后续的录取工作正常进行。

6、最新复试内容与形式

1、专业基础测评（上机 50 分）：专业基础测评主要考察考生有关 C++和数据结构的基础知识及基本运用能力。

2、英语综合面试（满分 25 分）：重点考核考生的英语听说能力。

3、专业综合面试（满分 100 分）：主要考察考生对本学科（专业）理论知识和应用技能的掌握程度，包括考生对本专业基础知识的掌握和理解程度，综合应用所学知识解决实际问题的能力；考察考生在本专业以外的学习、科研、社会实践或实际工作等方面的经历等。

复试成绩：满分 100 分，其中专业基础测评（上机）满分 50 分，英语综合面试满分 25 分，专业综合面试满分 100 分。复试成绩=(专业基础测评成绩+英语综合面试成绩+专业综合面试成绩×1.25)/2。

最终成绩：满分 100 分，初试成绩不计政治，复试成绩占比 50%，即最终成绩=（初试成绩【不计政治】÷4+复试成绩）÷2。

录取：按最终成绩由高到低排序，提出拟录取名单报批。为保证招生质量，报批人数可小于招生计划。

调剂：本专业在生源不足的情况下接受调剂。调剂信息将于复试阶段在中国科大研究生招生在线网站（http://yz.ustc.edu.cn）发布。

7、复试所需携带材料

1、准考证、身份证原件与复印件

2、（应届生）学生证 / （历届生）本科毕业证和学位证的原件与复印件

3、本科学业成绩单原件（教务部门盖章）

8、联培介绍

在科软读书如果你想搞学术，可以选择联合培养，联培是分配导师进实验室的，而且学费全免，每月还有 1200 左右的补助，待遇和 培养方式同学硕一样。科软在苏州和合肥有不少的实验室，只要你有兴趣有能力，导师都是很欢迎同学们的加入的。但如果你想早点就业 还是不要联培，因为联培一般都要 3 年，这就意味着研二你不能自主去自己想要的公司实习。 科软联培项目,包括 2 个和中科院的联培项目、1 个中科大苏研院的联培项目、3 个国外大学的联培项目。

软院复试经验帖

当大家进入复试，说明已经离录取不远了，再次恭喜大家。科软复试还是比较简单的，不用撸代码，也没有笔试。所以大家也不必担心。只有简单的上机和面试，科软上机历来是选择题，今年依然没说上机环境，所以还应该是选择题。虽然简章上写的是 C++,但去年考的还是 C（去年简章上也是写 C++）,故建议两手准备。选择题一般不难，二级难度吧。然后是专业面试，老师一般喜欢问专业相关的问题，比如小编本科网安，问的问题全是密码学，信息论与编码相关的。比如介绍一些**密码学里面的常用算法**，中国的密码学是什么，**海明码原理，海明码距的含义**，为什么考中科大，你咋不去考清华？还有一个印象深刻的问题，计算机与算盘的区别，大家可以探讨一下。工作了的喜欢问一些工作相关的。这一块看一些复试经验贴。

英语面试，这个历来是最水的。现在英语面试一般不会让你自我介绍了，但还是要准备一下。背一些常见话题，可以将话题引到你熟悉的地方。小编当时被问到了家乡，我说安庆，然后老师接着问我喜不喜欢 Huangmei Opera,搞得我一个措手不及，然后还让我唱两段。我哪会唱什么黄梅戏。

特别强调一下，机试一定要带笔进行计算。

下面是一些热心学长与学姐的经验帖，希望大家能有所启发。

1、 某热心学长

我是一志愿报考的考生，初试成绩不是很高 340+，排名大概中中间间，有点偏后。专业分 90+不高。我大三的时候在外实习一年，到了七八月才回到学校完全投入考研备考当中,那段时间是压力很大的，不过大三实习好处也是很多的，可以真的做到很多项目，学到很多学校学不到的知识。我建议在职考研的还是辞职较好，但是如果有毅力，且不愿意放弃工作的话也是可以边工作边考研的。我在这里写下一些心得，供明年考研的学弟学妹们参考参考。

机试：今年是第一年机试改革，考 C++和数据结构，考前复习将群里的二级 C++题库做了挺多，还重头学了一遍 C++,特别是面向对象那几个章节的内容，什么构造函数，友元函数等等一些列知识，结果第一天复试完的老哥们告诉我们没怎么考 C++ ，考得是 C 语言和数据结构。特别是文件和格式化输出，结果面试完的当天下午我看了一下午的 C 语言基础。晚上去考试，打开电脑的考试题，前两题直接懵逼，运算符优先级，没看，浮点数格式没看。最烦的是优先级还考了两题，瞬间觉得自己要挂在最后一关。后来平复了一下心情，发现，前几题有点难度，但是接下的题目还是有很多送分题的。数据结构的题也是做得不太顺，有时间复杂度（这个送分），还有图的遍历，堆排序的手工操作等等，我数据结构学的不是很好，第一遍做完第二遍检查的时候竟然发现了三道做错了的题，最后我在草稿纸上统计了一下不确定的题大概有 12 个，我就拼命检查，检查到最后系统自动交卷。出来之后忧心忡忡，虽然及格应该没问题，但是毕竟很多不会做，还是有些虚，和同学对档案大概错了 7 到 8 题吧。也不知道是多少题算及格，他交卷之后不会显示是否通过。虽然难度不是很大，但是陷阱很多，不过想及格，那些比较难的题做不出倒是影响不会很大，就怕简单的细节题不会。建议跨考的和 c 语言基础不牢的一定要重视机试，不要在这最后一关挂了，那就太可惜了。（不过小编觉得机试还是挺简单的，小编大概花了 20 分钟就交卷了）

英语面试：英语面试强烈建议先去。（小编觉得无所谓，嘻嘻）可以当做一次热身，我觉得面试都是考前紧张，一坐下之后就不会有紧张感了。英语面试排队时真的挺紧张，英语面试完了之后去专业面试就没什么感觉了，所以推荐先去英语面，英语面试的一个房间里有两个老师，一个老师是不说话的。另一个老师负责问你问题，你放轻松回答就行，推荐考前看看常见英语面试问题，自己准备一下，还是问到了挺多原题的。这里有一点要注意的是，你一进去坐下之后要主动自我介绍，不要等老师要你自我介绍，我就是等着老师说 introduce yourself,结果老师等了我一下，看我没说话，就直接发问了，哎，自我介绍我背的滚瓜烂熟，结果白背了.....,还被多问了几个问题，大概问了 8 个问题，有两个问题是常见的（比如你的家乡，你的大学。。）其他只能临场发挥。我虽然听得懂老师问什么，答的也挺流利。但是我觉得老师有点没太懂我的意思，可能我说的比较复杂...最后还问我对现在青少年玩游戏有什么看法，还有我玩什么类型的游戏，是单人还是多人....不过自我感觉英语面试还算挺顺利的。出来了之后就去专业面试，瞬间觉得没什么好紧张的了。这就是热身的效果吧...

专业面试：最重要的一部分来了，专业面试不是按序号进行，先去先排队。我准备了简历，奖状还有证书复印件。（小编注：应届生不用准备简历，没什么用）用夹子弄成三本本子。给老师一人一份（有人说老师不看简历，但是我想就算他不看简历，你拿着简历去，老师也不会扣分吧，还能看出你对这次考试的重视程度，）所以我还是建议大家一定要准备简历，而且是准备多几分，给老师一人一份，我进去面试的时候将简历给老师一人发了一份，老师挺热心地接了过去，翻看我的简历，然后我吸取了之前英语面试的经验，一坐下吸一口气就开始自我介绍，我说我是卓越班的学生，老师就问这个卓越班就是相当于我们这里的实验班是吧，后来他就让我继续介绍。很幸运地将准备的自我介绍完完全全讲完了，然后老师就开始问专业问题了。

老师：“你是本专业的对吧，我问你一些专业性的问题吧”，我惊了，我上面写了这么多实习的时候的项目，准备了这么多的素材，老师不问，直接问专业问题...

老师：“**宏汇编，宏编程，子程序调用**这些你知道吧？”

我瞬间懵逼，心想完了,真的是一个都答不出来：“我不知道，没听说过”

老师：“你没有学过**编译原理**是吧”

我：“是，（松了口气）”

老师：“软件工程**瀑布模型**知道吧”我说知道，然后老师说了一通关于软件工程的原理，说了瀑布模型的几个过程，然后问我**瀑布模型详细设计的文档内容**有哪些?

我：“.....”,然后开始将我在软件工程里学到的什么数据流图，数据字典，uml 类图全部说了出来。

老师：“你这个东扯一点西扯一点，还算是答对了一些，uml 类图，是面向对象的，我要面向过程的”

我：“那就是函数的详细功能介绍文档”

老师:"那你就很不专业了，那叫什么， 伪代码嘛！伪代码会写吧。"

我点点头。心想这应该还算是勉强答出一道题吧。。。。

然后还是那个老师问你问我**软件测试的定义**，这个倒是顺利答出来了，老师点点头，“噢。这个你知道”。

老师：“那我问你，如果我要你纠正软件中百分之九十的错误，你会怎么做？？”

我：“.....（这是想问我什么原理啊），为了表示我还是会的，就说一个软件的已发现错误和他为发现的错误成正比，所以我会用 100 个测试用例，然后如果其中 90 个测试用例出错，我将其改正，这样应该就算完成了吧”

老师：“的确，是有这个结论，已发现的错误和未发现的成正比，但是你这样仅仅是定量上解决了我的问题，而且测试用例也各不相同，我想知道你的思路”

我：“（这到底是想问我什么啊，我晕了，还是认怂吧），老师，我真的不知道了.....”。老师就说你学这个软件工程专业的，这都不知道，那你本科软件工程怎么学的？另一个老师也笑了一下说，对啊，这个问题的确应该知道的，然后那个老师就说你待会去查查文档或者是文献看看吧。

然后就到了另一个老师发问：“我看你的初试成绩也不高啊。”

我：“我大三学年在外实习了一年，到了七八月份才全身心投入到复习当中，所以没考的太好”

那个老师就点点头，然后问计算机网络的问题：“CSMA/cd 中的 cd 是什么意思啊？”

我：“碰撞检测，collision detective”.心想终于有会的了。

老师：“那他是怎么检测碰撞的？”

然后我指手画脚地给他解释了 CSMA/cd 的流程，A 主机向 B 主机发送信息，中间两者的信息发生碰撞，然后 B 主机收到 A 主机的信息发现出现错误，说明发生了碰撞，立刻停止发送信息。

老师：“这是流程，我问的是怎么检测出来的”

我：“载波监听，检测电压？”

老师：“不对，这样吧，我问你最小帧长是用来干嘛的？？”

我：“（恍然大悟，我发现我之前还真没注意到原来是这样检测碰撞的）对对对，就是利用最小帧长来检测争用期内是否发生了碰撞的”

老师点点头：“嗯，就到这吧，你可以出去了”，这样就结束了我的专业面试，整个面试大概 15 分钟左右。其实我出来之后并不是很担心专业面试，毕竟我是科班也有实习经验和项目，建议跨考的学弟学妹们自己准备几个项目，老师都非常厉害，其他专业的知识他都懂，多看看自己本专业的有关计算机的科目。老师人都非常好，非常友善，大家不要太过于紧张。

我一直在担心我的机试会不会过不了，直到看到看到屏幕上显示的计划录取，一年悬着的心终于落下了。感谢老师！

最后祝愿明年考研的学弟学妹们一战成硕，金榜题名！！！

2、热心学长

专业面试比较考究自己的积累，运气和说话方式。。在复习本科课程时候要自己去想，哪些问题可能是以提问的方式问出来的，像**数据库范式**类似的各科基础的东西一定要懂。。还有要自我介绍的时候**引导老师**问，我算法比赛得过奖，我在自我介绍的时候说了，然后老师就问了我 **3 个基础算法**问题。。最后一个老师问了我一道**数据库**的。

英语面试，自己把一些**复试常问的问题**总结有自己的回答，然后背一背。不要担心听不懂。我这 4 级听力全蒙 100 出头的都能听懂老师老师问的问题。。老师一开口你听不懂，但是问问题时候，那个 what,why 的句子才是关键。

3、学霸雨孩子

Emmm 大家好啊~我就那个大水怪雨孩子，本来考差了不太想写经验帖了，但我的经历应该适合更多跨考的小伙伴吧，希望我的经验可以让你少走一些弯路。 考研过程很辛苦，失败不可怕，可怕的是，考完后你就知道你为什么就失败了！ 介绍一下背景： 本科双非，土木跨考一战科软，总分 368，政治 63，英语 78，数学 118，408 109 成功 = 努力+实际吸收+考试状态+细心 ，择校很重要，选择很重要，结合自身经验说说为什么我选择科软呢？ 原因如下：

1、科软名额多，复试不歧视双非跨考，这样可以把更多精力放在初试，给出身不好的我们， 一个公平的机会。

2、科软改了 408，4 门课可以防止扎堆，而且初试认真学完后，对计算机有个深入宏观的把 握。

3、科软联培有导师，不联培第二年可以实习，对于自己，我觉得自习能力还可以，没必要 往某个方向深入研究，我比较喜欢站在巨人的肩膀上学习。

4、科大名气大呀，这学历不会被卡简历了，其他靠你这三年的技术。

复试篇： 英语面试和专业面是分开的。

第一天大帝在 e 组，d 组上下午都被怼得好惨。大家不要慌，只是压力面而已。什么语录都有，如：**你离录取还有很大的距离，努力有什么用**？ 软院不负责预测一波：18 年 310 230 人过，刷了 10 个左右。 19 年 310 3 48 个上线， 过复试 300+，目前不知刷了多少个。 20 年 我估计还是 **310** 的线。（不过小编觉得，今年恐怕形势不妙）

原因：

1、数学变难 了。

2、软院 550 个名额，往年复试线年年涨，只是因为没达到院线。要进行差额复试，毕竟过线人数超过名额才可以实施。所以我猜，20 年还是 310 的线。 另外，炸鱼年年有， 18 年各种群宣传，19 年群人数炸鱼等各种消息，目的就是为了扰乱军心，只要你有实力， 完全不用管那么多炸鱼，一个目标死磕到底，大佬们都去刚浙大了呢。科软近几年性价比还是特别高的。能上岸早点上岸哟。 科班复试多注意自己的项目，本科**学过的一些课程**。因为老师提问很随机，有些组死磕项 目，有些组问的问题很奇葩，比如今年学硕组的一位学硕就只被问了一个问题：**计算器和算盘有什么不同**。跨考的专注 408 就 OK 了，一般不会偏到哪儿去。我模拟

下面试的样子。 我是 18 号下午 E 组第一位进去面试的。 敲门进去。

我： 老师们好（微笑，往办公桌旁边桌子坐下，双手递上材料）

老师 A：这个你拿回去，还有同学，你坐到对面那个沙发上，不是坐在我旁边。（我接过放材料下面的奖学金证书等等，一脸尴尬）

我： 好的，不好意思啊老师，我刚刚不知情。（往沙发那儿坐着，看着老师们）

老师 A：你先自我介绍吧。

我： 好的，我是 balabla。

A： 你这专业课考得挺高的哈。（我心里松了一口气，点头寒暄：还好还好）

A： 你哪门学得不错呢？ 我：数据结构和网络学得比较好。计组学得最坏，考 408 的时候就是因为大题偏编译 原理完全不会了。

A： 那行，数据结构学得不错吧，我问你，**链表怎么建立**呢？

我： 链表啊，头插法和尾插法呀。

A： 完了？ 我： 我想一下哈。（过去了 20S）老师我想不起来了。完了呀。

B: （笑了一下）你想到什么就说什么，一直说就好，说完就说结束了。（我才反应过 来，老师是我要把建立头节点，什么链接操作等一系列说出来。

A： 那你说下**中断和系统调用**吧

我： 系统调用有时候不一定要用到中断，它有子程序调用和系统调用，而如果进去系 统态，就要用到中断，PSW 中状态字置为 1，中断呢，有多重中断和单重中断， 如果多重中断呢，它有两个流程，一个是中断指令完成得 balabla…（说得越多越 好）

B： 你说一下**如何建堆** 我： balala A

B：还有什么要问得吗？

C：你说一下 **cache 命中率**。

我： balala 好了，你可以出去了。（面试得时候，一定不要拘泥于某个问题，一定要发散地把这个问题前因后果全部扯出来，说得越多越好，我差不多每个问题都自己扯出其他两三个知识点，过程大概有 10 分种这样子）

英语面： 大家不用担心，**太简单啦**。老师们很和蔼，不用紧张，注意发音准确就好了。自己准 备好往年那些问题。

上机： 招生简章写的是 C++，而考试上机却是 C 语言。大概题目有很多很**简单**的，有基础的话完全不用担心，40 道选择，1 小时做完。大家差不多 15 到 20 分种就选完了那种。第一 天和第二天的试卷题目是不一样的。

4、18 科大软院一志愿初试及复试经历

一志愿复试是定在 3 月 18 号，计划录取结果 2 天后就出来了。复试之前查到自己被分在了李曦大帝组 F，直到我面试完才知道另外一个面试我的人是陈院长，尴尬，反正过程是一直被怼，自己确实掌握的知识不足也不灵活，感谢老师收留我。相对于来说英语面试就没那么紧张

了，听懂老师的提问，准确充分的回答她就可以了。

专业课面试

一进去给老师道过去成绩单，陈院长把我成绩单拿走了，陈院长坐在左边，李曦老师和另外一位老师坐在桌子右边。因为我本身是在电信行业工作，本科又是通信工程专业，这就为以后的被怼埋下了伏笔（所以建议大家**跨考的把相关专业书核心知识点**稍微看看，老师懂得都是好多的）。

陈院长：电信也在做云，那你可知道哪三朵云？

Loser 我：公有云、私有云，记不得第三朵啥云了。

陈院长：想不起来了？还有马云嘛。然后三个老师笑了，我就更尴尬了

陈院长后面也提到我去云南出差，然后说另外一个老师云南项目做得很好，都不请他们吃饭，弄得我越来越紧张，都不知道说啥，陈院长问我是不是你们工作后的人都喜欢考软院？然后我就顺便夸夸软院的培养模式真的很好。后面李曦老师开始质问我的时刻到来了

李曦：通信专业的，那我问你，你知道哪几种通信方式？

我：曼彻斯特编码

李曦：就这一种么？再想想

我：想不起来了，老师，毕业后太久了，工作中也很少接触到这个

李曦：那我问你 A 到 B 有哪几种传输方式？

我：答错成频分复用、时分复用、码分复用了（估计想问的是单工、双工、半双工吧）

李曦：你说你通信毕业的，又是通信工作，学的东西忘完了，干嘛去了？太可惜了，那行吧我问你 834 吧，计算机那些技术很重要？

我：中断技术及虚拟技术等，把中断流程概念答了下，可以提高计算机更好的并发性，使用到栈，flags 、ip、cs 寄存器入栈等过程，话说我也不知道自己在说什么了，感觉有点乱了。

李曦：那**中断硬件机制**怎么实现？

我默念：蒙了，记不得啊，难道不是用栈实现先进后出么？不过最后还是承认不会，不敢乱答。最后陈院长感觉替我解围了，说好了好了那就问这么多吧，你可以回去了，不过自己当时感觉心凉凉好担心被刷。

英语面试

进去一开始老师让我自我介绍下，我把事先准备好的说了一遍，主要是学习及工作经历，老师对我海外出差比较感兴趣问了我很多关于国外的事情，最后问我为什么选择报考软件学院？英语面一般准备应该都没有问题。复试听说被刷的一般跪在专业面试和机试上。

5、2018 下午一志愿机械跨考科软（注：18 年经验帖仅供参考）

今年复试据说大概刷了 12 人，其中居多在机试和C 组下午（好像是因为下午换了个老师，刷的比较狠）。由于今年过年的原因，出分较早，但是我整个寒假还是没看多少书（尴尬...）

所以过完年很慌，初五就跑去学校了，到学校宿舍不给住，就和同学临时短租了一间房。从两个月的寒假懒散状态转过来真的用了好些时日，一直到最后，效率也大不如初试。

先开始是打算把 834 的书本过掉（注：18 年考 834），再补一本计算机组成原理，其他也就不看了，事实证明，我连计组也没来得及补上。来到学校，先开始的重心放在了机试上，因为是硬指标，再加上 16 年的英文题让人不敢放松，我先从某宝上买了一个二级题库软件，刷了一半后就不刷了，刷了一些复试资料里面的题，中文题还行，英文题做的爆炸，就去原网站看讨论了，有的讨论还挺好的，解决了一部分问题。后来看群里有大佬把 15 年题库也刷了，吓得我赶紧去学校打印店，打印出来贼厚，时间原因我也没刷几道。此时，也边准备英文面试（专业面放到最后了，我是不是傻。。。）

看了宋逸轩的英文复试课，准备了一篇自我介绍和大约十个常见问题，背熟，便于和老师聊起来。专业面准备的也很匆忙，面试相比初试较重于概念我先把 834 又过了一遍。17 号中午到合肥（一个小插曲，宾馆订错了，订在东区了），宾馆是锦江之星，环境和隔音都挺好的。到了之后和研友一起打的去南区踩点，打车大概 10 块钱 10 分钟左右，转回来后又溜溜西区。

18 号早晨合肥下起了小雨，7 点 20 开始动员大会，老师大致介绍了下情况，说今年大数据与人工智能方向有 60%的人选择。。。可怕。我就没和大佬们争，安安静静选择合肥软设。再后来介绍下联培的情况。就开始材料审核，英语面了，人是真多啊，老师先开始说英语面不分上下午，我看材料审核人辣么多就先排英语了，结果排了一上午啊，前面还有 2 个下午的，学姐就把我们赶走了，还是优先面试上午组的，气哭。。。

下午面试是 1 点半开始，但我想去早了还是人多，就不慌不忙才从东区赶过去，到南区才到门口被门卫阿姨训了一顿，说都开始好久了，怎么才来，吓得我赶紧跑到 5 楼，结果还是好长的队伍。排队的时候问群里上午组的有哪些问到的问题，有老哥发了几个，我也都百度了下，不好找。在我前面的也是 D 组的，我们一起在给下午 D 组面试的计时，大概一个人 10 分钟左右。终于排到我了，敲门打招呼，面试宣应该是办公室（当时没仔细观察）两张办公桌拼在一起，三位老师，两男一女，考生也就坐在旁边，氛围挺好的。进来之后我把材料，简历交给老师，坐下后，女老师上来就说

介绍一下你的优点。当时突然有点懵遇，就把英文自我介绍中的优点翻译过来答上去了。右手边的男老师 A 拿着审核材料问

网络的功能是什么。这个问题群里有提到过，可是当时没注意，支支吾吾没回答上来，老师显然开始不满意了，左手边的男老师 B 就说这是网络整本书讨论的，就在绪论上，不应该不会的，我赶紧说，老师，我网络学的比较差，这时，他把手上的材料放下，换成简历，说你是跨考的啊，我嗯了一声，我心想这下应该有转机了。他又问

你哪门学的好，我说数据结构和操作系统，老师 A 冷笑了一声，当时我都慌了...估计是说这两门课的人太多了。又问④操作系统的几大功能是什么，当时由于紧张，说的比较乱，男老师 B 就说你按书本上的顺序答，大脑太混乱了，也没答好。老师 B 好像又问⑤进程管理管理的是什么。然后他就问女老师还有没有其他要问的，女老师就问⑤为什么要跨考啊，我就把准备好的答上去了。又问②栈和队列有什么区别，答上来后又问③生活中有哪些常见栈和队列的应用，我又潜了，我以为她会问在网络和操作系统中有哪些应用，我向老师确认了一下，确定后只答上来队列的应

用，栈我实在没想起来。好像就这么多问题（时间久了，有的都忘了，想起来再补上）然后就让我出去了。

当时内心：what???结束了？？？这么快，虽然没计时，但大概也就 5 分钟吧，阿西吧…面崩了。当时英面已经快结束了，都不用排队，没多想，就赶紧跑过去了。Ps:其实跨考的专业而挺简单的，我就只被问了专业课的内容，科班出身的容被怼。

英语面：

调整一下呼吸，把自我介绍熟悉一下，就进去了，有两个老师，一男一女，男的好像是外教（惊呆）good afternoon，my dear professors!就进去了，女老师让在上面签字，男老师就开始说 introduce yourself，我就把之前准备的说了一下（老师一般不会让你说完，太无聊）老师在我说爱好的时候，因为我提到我的爱好里面有围棋（go）老师先开始没想到。

这个，当我提到李世石和古力，女老师一脸惊讶，然后我就补了句是有关 AlphaGo 的那个 go，然后外教就和女老师讨论最近的 champion，当时提到柯洁的时候，我本来想说是一个非常年轻的棋士，结果脑子一抽，来了一句 he is too young（突然膜了一发...当时吓死...）然后后来老师就没让我继续自我介绍了，老师让我介绍一下你的**母亲**（这个以前学长帖子有提到，但我没提前好好准备）答得不是很好，然后老师问她的**爱好**是什么，我以为还让我说我自己的，就和老师说已经说过了，外教又重复了一遍我才反应过来，我说 she believe Jesus (这个我发音不准，老师先开始没听懂，我解释了好半天）大概就这样结束了，女老师还说了句 good luck 我说了一句 thank you 就走了。总之，**英文面很轻松**，老师人也很好，发音很好听。

(附自己准备的几个问题：①**plans in postgraduate life** ②**introduce youruniversity** ③**why choosethis major** and④为什么跨考⑤**strengthGweakness hometow**n，family Hefei 的印象四为什么来科软读 mse）

面试结束后距离机试还早，就先回宾馆了，回去后，和我住在一起的小伙伴就和我说，别看机试了，今年很简单（他是上午组的），今年的逻辑推理题挺多的（注：现在没有了）。收拾后就去二教 8 楼等了，题目确实不难，而且一场的题目应该是一样的，不是随机抽的，因为有一题错了，老师还过来订正。

面试完，无论过程如何，结束后全身轻松。总之以后的就不是我能左右的了，静等结果，我没报名联培，所以第二天也没过去面试。

机试指南

20 年的招生简章机试部分为 C++和数据结构，与 2019 年一致，但 2019 年仍然是 c 语言，所以建议大家大家稍微准备一下 C 语言部分，这个也很简单。下面是在互联网上搜集的一些 C\C++试题，可能存在少量错误，敬请谅解，大家可以参照这个思路，自己找一些题目刷，比如二级 C 语言/C++。

1、C 语言部分

C 语言理论上机考试选择题部分（共 200 题）

1、下面程序的输出是___D_____

```
#include<stdio.h>
void main()
{ int k=11;
   printf("k=%d,k=%o,k=%x\n",k,k,k);
}
```

oct；
hex；

A) k=11,k=12,k=11 B) k=11,k=13,k=13

C) k=11,k=013,k=0xb D) k=11,k=13,k=b

2、在下列选项中,不正确的赋值语句是__D_____.

A) ++t; B) n1=(n2=(n3=0));

C) k=i=j; D) a=b+c=1;

3、下面合法的 C 语言字符常量是_____A____.

A) '\t' B) "A" C) 65 D) A

4、表达式: 10!=9 的值是_____D____.

A) true B) 非零值 C) 0 D) 1

5、C 语言提供的合法的数据类型关键字是_____B____.

A) Double B) short C) integer D) Char

6、字符(char)型数据在微机内存中的存储形式是__D__.

A) 反码 B) 补码 C) EBCDIC 码 D) ASCII 码

7、C 语言程序的基本单位是_____C_____.

A) 程序行 B) 语句 C) 函数 D) 字符

1）先算a*a（结果为144，此时a的值没变，还是3）；然后算 a-=a*a，等效于 a = a - 144;（结果为-132，因为赋值符号，此时a的值为-6，发生了改变）；最后算a = a + a ，结果为-264

8、设 int a=12,则执行完语句 a+=a-=a*a 后,a 的值是____D____

A) 552 B) 264 C) 144 D) -264

9、执行下面程序中的输出语句后,输出结果是____B__.

```
#include<stdio.h>
void main()
{int a;
 printf("%d\n",(a=3*5,a*4,a+5));
}
```

A) 65 B) 20 D) 10

a=3*5，这里给a赋值了，赋值为15；后边的a*4，a+5是表达式，但是都没有给a赋值，所以a仍然是15

(a=3*5,a*4,a+5) 这个逗号表达式运算结果是取最后一个表达式的值a+5，即20，所以最后打印出来是20。

10、下面程序的输出是____B_____.

```
#include<stdio.h>
void main()
{int x=023;
 printf("%d\n",--x);
}
```

A) 17 B) 18 C) 23 D) 24

11、下面程序的输出的是_____C____.

```
#include<stdio.h>
void main()
{int x=10,y=3;
 printf("%d\n",y=x/y);
}
```

A) 0 B) 1 C) 3 D) 不确定的值

12、已知字母 A 的 ASCII 码为十进制的 65,下面程序的输出是_____A_____.

```
#include<stdio.h>
void main()
{char ch1,ch2;
```

```
 ch1='A'+'5'-'3';
 ch2='A'+'6'-'3';
 printf("%d,%c\n",ch1,ch2);
}
```

A的ASCII码：65

A) 67,D　　B) B,C　　C) C,D　　D) 不确定的值

13、若要求在 if 后一对圆括号中表示 a 不等于 0 的关系,则能正确表示这一关系的表达式为____D__.

A) a<>0　　B) !a　　C) a=0　　D) a

14、以下程序的输出结果是____D_____.

```
#include<stdio.h>
void main()
{ int x=10,y=10;
   printf("%d %d\n",x--,--y);
}
```

A) 10 10　　B) 9 9　　C) 9 10　　D) 10 9

15、设有如下定义:

```
   int x=10,y=3,z;
```

则语句

```
   printf("%d\n",z=(x%y,x/y));
```

的输出结果是_____D_____.

A) 1　　B) 0　　C) 4　　D) 3

16、为表示关系 x≥y≥z,应使用 C 语言表达式___A___.

A) (x>=y)&&(y>=z)　　B) (x>=y)AND(y>=z)

C) (x>=y>=z)　　D) (x>=y) & (y>=z)

17、C 语言中非空的基本数据类型包括____B____.

A) 整型,实型,逻辑型　　B) 整型,实型,字符型

C) 整型,字符型,逻辑型

D) 整型,实型,逻辑型,字符型

18、若 x 和 y 都是 int 型变量,x=100,y=200,且有下面的程序片段:

```
   printf("%d",(x,y));
```

上面程序片段的输出结果是____A___.

A) 200　　B) 100　　C) 100 200　　D) 输出格式符不够,输出不确定的值

19、阅读下面的程序

```
#include<stdio.h>
void main()
{
char ch;
scanf("%3c",&ch);
printf("%c",ch);
}
```

如果从键盘上输入

```
      abc<回车>
```

则程序的运行结果是__A_____.

A) a　　B) b　　C) c　　D) 程序语法出错

20、阅读下面的程序

```
#include<stdio.h>
void main()
{
  int i,j;
  i=010;
  j=9;
  printf("%d,%d",i-j,i+j);
 }
```

则程序的运行结果是____D____.

A) 1,19　　B) -1,19　　C) 1,17　　D) -1,17

21、阅读下面的程序

```
#include<stdio.h>
void main()
 {
  int i,j,m,n;
  i=8;j=10;
  m=++i;
  n=j++;
  printf("%d,%d,%
}
```

a+=a-=a*=a计算顺序是1 a*=a;2 a-=a;3 a+=a;也就是1 a=a*a;2 a=a-a;3 a=a+a;可以看到，无论a值是多少，第一步的结果是多少，到第二步的时候，a的值都会是0.所以最终结果，也就是第三步的结果，仍为0。最终a为0。

程序的运行结果是_____C____.

A) 8,10,8,10　　B) 9,11,8,10

C) 9,11,9,10　　D) 9,10,9,11

22、已知 a=12,则表达式 a+=a-=a*=a 的结果是_____A__.

A) 0　　B) 144　　C) 12

C语言中允许的基本数据类型有5种，即字符型char、整型int、浮点型float、双精度型double和无值类型void，而浮点型和双精度型是实型，特别注意的是C语言没有逻辑型。

23、若已定义 int a,则表达式 a=10,a+10,a++的值是__B_.
A) 20 B) 10 C) 21
D) 11

1) a%3=1
2) (int)(x+y)=7
3) 第一步和第二步的答案相乘＝7
4) 7%2=1
5) 1/4=0
*/%同级别，而且结合性从左到右。

24、阅读下面的程序
#include<stdio.h>
void main()
{
int i,j;
scanf("%3d%2d",&i,&
printf("i=%d,j=%d\n",i,j);
}
如果从键盘上输入 1234567<回车>,则程序的运行结果是____D____.
A) i=123,j=4567 B) i=1234,j=567
C) i=1,j=2 D) i=123,j=45

25、下面程序的输出结果是____D____.
#include<stdio.h>
void main()
{
int a=-1, b=4, k;
k=(++a<=0)&&(b--<=0);
printf("%d,%d,%d\n",k,a,b);
}
A) 1,1,2 B) 1,0,3 C) 0,1,2 D) 0,0,3

26、下面程序的输出结果是____A____.
#include<stdio.h>
void main()
{
int a=5,b=3;
float x=3.14, y=6.5;
printf("%d,%d\n",a+b!=a-b,x<=(y-=6.1));
}
A) 1,0 B) 0,1 C) 1,1 D) 0,0

27、执行下面程序段后,输出结果是____A____.
int a;
int b=65536;
a=b;
printf("%d\n",a);
A) 65536 B) 0 C) -1
D) 1

28、若有以下定义和语句:
int a=010, b=0x10, c=10;
printf("%d,%d,%d\n",a,b,c);
则输出结果是____B_____.
A) 10,10,10 B) 8,16,10 C) 8,10,10
D) 8,8,10

29、已知有 double 型变量 x=2.5,y=4.7,整型变量 a=7,
则表达式 x+a%3*(int)(x+y)%2/4 的值是_____B____.
A) 2.4 B) 2.5 C) 2.75
D) 0

30、若已定义 x 和 y 是整型变量,x=2;,则表达式 y=2.75+x/2 的值是____C____.
A) 5.5 B) 5 C) 3 D) 4.0

31、以下程序的输出结果是____D____.
#include<stdio.h>
void main()
{
int a=12, b=12;
printf("%d,%d\n",--a,++b);
}
A) 10,10 B) 12,12 C) 11,10 D) 11,13

32、设有以下语句:int x=10;x+=3+x%(3),则 x 的值是.____A_____
A) 14 B) 15 C) 11
D) 12

33、若 d 为 double 型变量,则表达式 d=1,d+5,d++的值是_____D__.
A) 1 B) 6.0 C) 2.0
D) 1.0

34、表达式 5!=3 的值是____D____.
A) T B) 非零值 C) 0
D) 1

35、若有定义 int a=12,n=5,则表达式 a%=(n%2)运算后,a 的值_____A____.
A) 0 B) 1 C) 12
D) 6

36、若有定义 int x=3,y=2 和 float a=2.5,b=3.5,则表达式:(x+y)%2+(int)a/(int)b 的值是__D__.
A) 0 B) 2 C) 1.5
D) 1

37、在 C 语言中,以下叙述不正确的是_____A____.
A) 在 C 程序中,无论是整数还是实数,都能被准确无误的表示
B) 在 C 程序中,变量名代表存储器中的一个位置
C) 静态变量的生存期与整个程序的生存期相同
D) C 语言中变量必须先定义后引用

38、C 语言中的变量名只能由字母,数字和下划线三种字符组成,且第一个字符____C____.
A) 必须为字母
B) 必须为下划线
C) 必须为字母或下划线
D) 可以是字母,数字或下划线中的任意一种

39、设有说明:char w; int x; float y; double z;则表达式:w*x+z-y 值的数据类型是___D____.
A) float B) char C) int D) double

40、一个 C 语言的执行是从_____A_____.
A) 本程序的主函数开始,到本程序的主函数结束
B) 本程序的第一个函数开始,到本程序的最后一个函数结束
C) 本程序的主函数开始,到本程序的最后一个函数结束
D) 本程序的第一个函数开始,到本程序的主函数结束

41、设 a 为整型变量,不能正确表达数学关系 10<a<15 的 C 语言表达式是____A____.
A) 10<a<15 B) a==11||a==12||a==13||a==14
C) a>10&&a<15 D) !(a<=10)&&!(a>=15)

! 的优先级高于&&

42、下列程序执行后的输出结果是_____C____.

```
#include <stdio.h>
  void main( )
  { int a=5,b=60,c;
    if (a<b)
    {c=a*b;printf("%d*%d=%d\n",b,a,c);}
    else
    {c=b/a;printf("%d/%d=%d\n",b,a,c);}
  }
```

A) 60/5=12 B) 300 C) 60*5=300
D) 12

43、如果 c 为字符型变量,判断 c 是否为空格不能使用____A____.(假设已知空格 ASCII 码为 32)
A) if(c=='32') B) if(c==32)
C) if(c=='\40') D) if(c==' ')

\称为转义字符,可以改变\后面字符原本的含义。\+八进制数字代表字符ASCII值

44、运行下面程序时,若从键盘输入"3,5<CR>",则程序的输出结果是____D____.

```
#include <stdio.h>
  void main( )
  {
    int x,y;
    scanf("%d,%d",&x,&y);
    if (x==y)
        printf("x==y");
    else if (x>y)
        printf("x>y");
        else
            printf("x<y");
  }
```

cr是回车,在C中就是字符'\r'

A) 3<5 B) 5>3 C) x>y
D) x<y

45、运行下面程序时,若从键盘输入数据为"6,5,7<CR>",则输出结果是____C___.

```
#include <stdio.h>
  void main( )
   { int a,b,c;
    scanf("%d,%d,%d",&a,&b,&c);
    if (a>b)
        if (a>c)
            printf("%d\n",a);
        else
            printf("%d\n",c);
    else
        if (b>c)
            printf("%d\n",b);
        else
            printf("%d\n",c);
  }
```

A) 5 B) 6 C) 7 D) 不定值

数据之间不可以用逗号分隔，只能用空白字符（空格或tab键或者回车键）分隔——“2 （空格）3（tab） 4”或“2（tab）3（回车）4”等。若是“%d,%d,%d”，则在输入数据时需要加“,”，如“2,3,4”。

46、执行下面程序时,若从键盘输入"2<CR>"，则程序的运行结果是____A____.

```
#include <stdio.h>
 void main( )
 { int k; char cp;
   cp=getchar( );
   if (cp>='0' && cp<='9')
   k=cp-'0';
   else if (cp>='a' && cp<='f')
                k=cp-'a'+10;
          else   k=cp-'A'+10;
   printf("%d\n",k);
 }
```

A) 2　　B) 4　　C) 1
D) 10

47、运行下面程序时,从键盘输入"2.0<CR>"，则输出结果是___B_____.

```
#include <stdio.h>
 void main( )
 { float a,b;
   scanf("%f",&a);
   if (a<0.0) b=0.0;
   else  if  ((a<0.5)  &&  (a!=2.0))
b=1.0/(a+2.0);
          else if (a<10.0) b=1.0/2;
               else b=10.0;
   printf("%f\n",b);
 }
```

A) 0.000000　　B) 0.500000
C) 1.000000　　D) 0.250000

48、执行下面程序后,运行结果是____A____.

```
#include <stdio.h>
 void main( )
 { int x=41,y=1;
   if (x%3==0 && x%7==0)
     { y+=x;printf("y=%d\n",y);}
   else
      {y=x;printf("y=%d",y);}
  }
```

A) y=41　　B) y=43　　C) y=42
D) y=1

49、运行下面程序时，从键盘输入"12,34,9<CR>"，则输出结果是___A___.

```
#include <stdio.h>
 void main( )
 { int x,y,z;
   scanf("%d,%d,%d",&x,&y,&z);
   if (x<y)
       if (y<z)printf("%d\n",z);
       else printf("%d\n",y);
  else if (x<z)printf("%d\n",z);
        else printf("%d\n",x);
 }
```

A) 34　　B) 12　　C) 9　　D) 不确定的值

50、运行下面程序时，从键盘输入字母 H，则输出结果是_____C____.

当变量表达式所表达的量与其中一个case语句中的常量相符时，就执行此case语句后面的语句，并依次下去执行后面所有case语句中的语句，除非遇到break;语句跳出switch语句为止。如果常量表达式的量与所有case语句的常量都不相符，就执行default语句中的语句。

```
#include <stdio.h>
 void main( )
 { char ch;
   ch=getchar( );
   switch(ch)
   { case 'H':printf("Hello!\n");
     case 'G':printf("Good morning!\n");
     default:printf("Bye_Bye!\n");
   }
 }
```

A) Hello!　　B) Hello!
Good Morning!
C) Hello!　　D) Hello!
Good morning!　　Bye_Bye!
Bye_Bye!

51、执行下列程序段后的输出结果是_____A____.

```
int x=1,y=1,z=1;
  x+=y+=z;
printf("%d\n",x<y?y:x);
```

A) 3　　B) 2　　C) 1
D) 4

52、设 ch 是 char 型变量,值为'A',则表达式 ch=(ch>='A' && ch<='Z')?ch+32:ch 的值是__B___.

A) Z　　B) a　　C) z
D) A

小写字母排在大写字母的后面，一个字母的大小写数值相差32

53、下面程序的输出结果是____C____.

```
#include <stdio.h>
 void main( )
 { int x=8,y=-7,z=9;
   if (x<y)
       if (y<0) z=0;
```

```
        else z-=1;
      printf("%d\n",z);
    }
```

A) 8　　B) 1　　C) 9
D) 0

54、运行下面程序时,若从键盘输入"5 <CR>",则程序的输出结果是_____B___.

```
#include <stdio.h>
  void main( )
  { int a ;
    scanf("%d",&a);
    if (a++>5)printf("%d\n",a);
    else printf("%d\n",a--) ;
  }
```

A) 7　　B) 6　　C) 5
D) 4

55、运行下面程序时，若从键盘输入"3，4 <CR>"，则程序的输出结果是____B___.

```
 #include <stdio.h>
 void main( )
  { int a,b,s;
    scanf("%d,%d",&a,&b);
    s=a;
    if (s<b) s=b;
    s=s*s;
    printf("%d\n",s) ;
  }
```

A) 14　　B) 16　　C) 18　　D) 20

56、下列程序的执行结果是_____D____.

```
 #include <stdio.h>
 void main( )
  { int x=0,y=1,z=0;
    if (x=z=y)
        x=3;
    printf("%d,%d\n",x,z);
  }
```

A) 3,0　　B) 0,0　　C) 0,1
D) 3,1

57、假定等级和分数有以下对应关系：

等级：A　　分数：85～100
等级：B　　分数：60～84
等级：C　　分数：60 以下

对于等级 grade 输出相应的分数区间，能够完成该功能的程序段是____D____.

A)
```
switch (grade)
  {
   case 'A':printf("85--100\n");
   case 'B':printf("60--84\n");
   case 'C':printf("60 以下\n");
   default:printf("等级错误!\n");
  }
```

B)
```
 switch (grade)
  {
   case 'A':printf("85--100\n");break;
   case 'B':printf("60--84\n");
   case 'C':printf("60 以下\n");
   default:printf("  等级错误!\n");
  }
```

C)
```
 switch (grade)
  {
   case 'A':printf("85--100\n");break;
   case 'B':printf("60--84\n");break;
   case 'C':printf("60 以下\n");
   default:printf("等级错误!\n");
  }
```

D)
```
switch (grade)
  {
   case 'A':printf("85--100\n");break;
   case 'B':printf("60--84\n");break;
   case 'C':printf("60 以下  \n");break;
   default:printf("等级错误!\n");
  }
```

58、能够完成如下函数计算的程序段是__B____.

$$y=\begin{cases}-1 & x<0\\ 0 & x=0\\ 1 & x>0\end{cases}$$

A)
```
y=1;
if(x!=0)
  if(x>0) y=1;
  else y=0;
```

B)
```
if (x>=0)
  if(x>0) y=1;
  else y=0;
 else y=-1;
```

C)
```
y=0;
if (x>=0)
  if (x>0) y=1;
  else y=-1;
```

D)
```
y=-1;
if (x>0) y=1;
else y=0;
```

59、有如下程序

```
 #include <stdio.h>
 void main( )
  { float x=5.0,y;
    if(x<0.0) y=0.0;
```

```
    else if (x<10.0) y=1.0/x;
    else y=1.0;
    printf("%f\n",y);
  }
```

该程序的输出结果是____C_____.

A) 0.000000　　B) 0.50000

C) 0.200000　　D) 1.000000

60、以下程序的执行结果是___B_____.

```
#include <stdio.h>
  void main( )
  { int x=1,y=0;
    switch (x)
    {
      case 1:
            switch (y)
            {
              case 0:printf("first\n");break;
              case 1:printf("second\n");break;
            }
      case 2:printf("third\n");
    }
  }
```

A) first　　B) first

second　　third

C) first　　D) second

third

61、以下程序的执行结果是____A____.

```
#include <stdio.h>
void   main( )
  { int a,b,c,d,x;
    a=c=0;
    b=1;
    d=20;
    if (a) d=d-10;
    else if(!b)
            if (!c) x=15;
            else x=25;
    printf("d=%d\n",d);
  }
```

A) d=20　　B) d=10　　C) d=15

D) 25

62、有如下程序:

```
#include <stdio.h>
 void main( )
   { int a=2,b=-1,c=2;
     if (a<b)
       if (b<0) c=0;
       else c++;
     printf("%d\n",c);
   }
```

该程序的输出结果是___C_____.

A) 0　　B) 1　　C) 2

D) 3

63、下列程序执行后的输出结果是____B____.

```
#include <stdio.h>
void main( )
  { int x,y=1,z;
    if ((z=y)<0)  x=4;
    else if (y==0) x=5;
            else x=6;
    printf("%d,%d\n",x,y);
  }
```

A) 4,1　　B) 6,1　　C) 5,0　　D) 出错信息

64、有如下程序

```
#include <stdio.h>
  void main( )
   { int x=1,a=0,b=0;
     switch(x)
     {
     case 0: b++;
     case 1: a++;
     case 2: a++;b++;
     }
     printf("a=%d,b=%d\n",a,b);
   }
```

该程序的输出结果是_____A____.

A) a=2,b=1　　B) a=1,b=1　　C) a=1,b=0　　D) a=2,b=2

65、下面程序的输出结果是____C_____.

```
#include <stdio.h>
void main( )
  { int a=-1,b=1,k;
    if ((++a<0) && (b--<=0))
      printf("%d %d\n",a,b);
    else
      printf("%d %d\n",b,a);
  }
```

A) -1 1　　B) 0 1　　C) 1 0　　D) 0

0

66、假定 w、x、y、z、m 均为 int 型变量，有如下程序段：

```
w=1;x=2;y=3;z=4;
m=(w<x)?w:x;
m=(m<y)?m:y;
m=(m<z)?m:z;
```

则该程序段执行后，m 的值是____D_____.

A) 4 B) 3 C) 2 D) 1

67、以下程序的输出结果是___D_____.

```
main( )
{ int a=100;
  if (a>100) printf("%d\n",a>100);
  else printf("%d\n",a<=100);
}
```

A) a<=100 B) 100 C) 0 D) 1

68、若执行下面的程序从键盘上输入 9,则输出结果是._____B_____

```
#include <stdio.h>
void main( )
 {int n;
  scanf("%d",&n);
  if (n++<10) printf("%d\n",n);
  else printf("%d\n",n--);}
```

A) 11 B) 10 C) 9 D) 8

69、以下程序输出结果是_____D_____.

```
#include <stdio.h>
void main( )
 { int m=4;
   if (++m>5) printf("%d\n",m--);
   else printf("%d\n",--m);
 }
```

A) 7 B) 6 C) 5 D) 4

70、若执行下面的程序从键盘上输入 5,则输出结果是.

```
#include <stdio.h>
void main( )
 {int x;
  scanf("%d",&x);
  if (x++>5) printf("%d\n",x);
  else printf("%d\n",x--);
 }
```

A) 7 B) 6 C) 5 D) 4

71、以下程序段运行结果是____A____.

```
int x=1,y=1,z=-1;
 x+=y+=z;
 printf("%d\n",x<y?y:x);
```

A) 1 B) 2 C) 4 D) 不确定的值

72、有以下程序

```
#include <stdio.h>
 void main( )
 { int a,b,c=246;
   a=c/100%9;
   b=(-1)&&(-1);
   printf("%d,%d\n",a,b);
 }
```

输出结果是____A____.

A) 2,1 B) 3,2 C) 4,3 D) 2,-1

73、运行下面程序时，若从键盘输入数据为"123"，

则输出结果是___C____.

```
#include "stdio.h"
 void main()
 { int num,i,j,k,place;
   scanf("%d",&num);
   if (num>99)
       place=3;
   else if(num>9)
              place=2;
         else
              place=1;
   i=num/100;
   j=(num-i*100)/10;
   k=(num-i*100-j*10);
   switch (place)
   { case 3: printf("%d%d%d\n",k,j,i);
             break;
     case 2: printf("%d%d\n",k,j);
             break;
     case 1: printf("%d\n",k);
   }
 }
```

A) 123 B) 1,2,3 C) 321 D) 3,2,1

74、执行下列程序后的输出结果是___D____.

```
#include <stdio.h>
  void main( )
  { int k=4,a=3,b=2,c=1;
    printf("%d\n",k<a?k:c<b?c:a);
  }
```

A) 4　　B) 3　　C) 2
D) 1

75、以下条件表达式中能完全等价于条件表达式 x 的是____B___.

A) (x==0)　　B) (x!=0)　　C) (x==1)
D) (x!=1)

76、若运行下面程序时,给变量 a 输入 15,则输出结果是___A___.

```
#include <stdio.h>
  void main( )
  { int a,b;
    scanf("%d",&a);
    b=a>15?a+10:a-10;
    printf("%d\n",b) ;
  }
```

A) 5　　B) 25　　C) 15　　D) 10

77、运行下面程序后,输出是___D___.

```
#include <stdio.h>
void main( )
  { int k=-3;
    if (k<=0) printf("****\n");
    else printf("####\n")
  }
```

A) ####
B) ****
C) ########
D) 有语法错误不能通过编译

78、执行下面程序的输出结果是____C____.

```
#include <stdio.h>
void main( )
 { int a=5,b=0,c=0;
    if (a=a+b) printf("****\n");
    else   printf("####\n");
  }
```

A) 有语法错误不能编译
B) 能通过编译，但不能通过连接
C) 输出 ****
D) 输出 ####

79、为了避免嵌套的 if-else 语句的二义性，C 语言规定 else 总是与___C___组成配对关系.

A) 缩排位置相同的 if
B) 在其之前未配对的 if
C) 在其之前尚未配对的最近的 if
D) 同一行上的 if

80、设 x 、y 、z 、t 均为 int 型变量,则执行以下语句后,t 的值为____C_____.

```
  x=y=z=1;
  t=++x || ++y && ++z;
```

A) 不定值　　B) 4　　C) 1
D) 0

81、以下程序段____C_____.

```
 x=-1;
  do
  {
    x=x*x;
}   while (!x);
```

A)是死循环　　B)循环执行两次
C)循环执行一次　　D)有语法错误

82、对下面程序段描述正确的是___B____.

```
  int x=0,s=0;
    while (!x!=0) s+=++x;
    printf("%d",s);
```

A) 运行程序段后输出 0
B) 运行程序段后输出 1
C) 程序段中的控制表达式是非法的
D) 程序段循环无数次

83、下面程序段的输出结果是____C____.

```
  x=3;
    do { y=x--;
         if (!y) {printf("*");continue;}
         printf("#");
       } while(x=2);
```

A) ##　　B) ##*　　C) 死循环　　D)输出错误信息

84、下面程序的运行结果是____B____.

```
#include<stdio.h>
void main( )
```

```
{ int a=1,b=10;
  do
   { b-=a;a++;
   } while(b--<0);
   printf("%d,%d\n",a,b);
}
```

A) 3,11 B) 2,8 C) 1,-1 D) 4,9

85、下面程序段的运行结果是____B_____.

```
int n=0;
 while (n++<=2)
      printf("%d",n);
```

A) 012 B) 123 C) 234 D) 错误信息

86、下面程序段的运行结果是___D_____.

```
int x=0,y=0;
 while (x<15) y++,x+=++y;
 printf("%d,%d",y,x);
```

A) 20,7 B) 6,12 C) 20,8 D)8,20

87、下面程序的运行结果是___B_____.

```
#include<stdio.h>
void main()
 { int s=0,i=1;
   while (s<=10)
    { s=s+i*i;
      i++;
    }
   printf("%d",--i);
 }
```

A) 4 B) 3 C) 5 D) 6

88、函数 pi 的功能是根据以下近似公式求 π 值：____C_____

(π*π)/6=1+1/(2*2)+1/(3*3)+..+1/(n*n)

请填空，完成求 π 的功能。

```
#include <math.h>
 void main( )
 { double s=0.0; int i,n;
   scanf("%ld",&n);
   for(i=1;i<=n;i++)
   s=s+_______ ;
   s=(sqrt(6*s));
   printf("s=%e",s);
 }
```

A) 1/i*i B) 1.0/i*i C) 1.0/(i*i) D) 1.0/(n*n)

89、下面程序段的运行结果是____B____.

```
for(x=10;x>3;x--)
 { if(x%3) x--;
   --x; --x;
   printf("%d ",x);
 }
```

A) 6 3 B) 7 4 C) 6 2 D) 7 3

90、下面程序的运行结果是____D_____.

```
#include<stdio.h>
void main( )
   { int a,b;
     a=-1;
     b=0;
     do {
          ++a;
          ++a;
          b+=a;
        } while(a<9);
     printf("%d\n",b);
   }
```

A) 34 B) 24 C) 26 D) 25

91、下面程序段的运行结果是_____D_____.

```
for(i=1;i<=5;)
   printf("%d",i);
   i++;
```

A) 12345 B) 1234 C) 15 D) 无限循环

92、下面程序的输出结果是_____B_____.

```
#include<stdio.h>
void  main( )
 { int n=4;
   while (n--) printf("%d ",n--);
 }
```

A) 2 0 B) 3 1 C) 3 2 1 D) 2 1 0

93、以下程序运行后的输出结果是___D_____.

```
#include<stdio.h>
void main()
 { int i=10,j=0;
```

```
    do
    { j=j+1;   i--;
    }while(i>2);
    printf("%d\n",j);
  }
```

A) 50　　B) 52　　C) 51
D) 8

94、以下函数的功能是：求 x 的 y 次方，请填空.____C_____

```
#include<stdio.h>
void main()
  { int i,x,y;
    double z;
    scanf("%d %d",&x,&y);
    for(i=1,z=x;i<y;i++)
      z=z*_____ ;
     printf("x^y=%e\n",z);
  }
```

A) i++　　B) x++　　C) x
D) i

95、有如下程序

```
  #include<stdio.h>
  void main()
    { int x=23;
      do
      { printf("%d",x--);
      }while(!x);
    }
```

该程序的执行结果是___B____
A) 321　　B) 23
C) 不输出任何内容　　D) 陷入死循环

96、以下程序段的执行结果是____C____.

```
  int i,j,m=0;
  for(i=1;i<=15;i+=4)
    for(j=3;j<=19;j+=4)
    m++;
    printf("%d\n",m);
```

A) 12　　B) 15　　C) 20　　D) 25

97、下面程序的输出结果是_____A_____.

```
#include<stdio.h>
void main( )
  { int i;
    for(i=1;i<6;i++)
      { if (i%2!=0) {printf("#");continue;}
        printf("*");
      }
    printf("\n");
  }
```

A) #*#*#　　B) #####　　C) *****
D) *#*#*

98、下面程序的输出结果是____D_____.

```
#include<stdio.h>
void   main( )
  { int x=10,y=10,i;
    for(i=0;x>8;y=++i)
        printf("%d %d ",x--,y);
  }
```

A) 10 1 9 2　　B) 9 8 7 6
C) 10 9 9 0　　D) 10 10 9 1

99、执行以下程序后，输出的结果是_____D_____.

```
#include<stdio.h>
void   main( )
  { int y=10;
    do {y--;}
    while (--y);
    printf("%d\n",y--);
  }
```

A) -1　　B) 1　　C) 8
D) 0

100、有如下程序

```
#include<stdio.h>
void main( )
  { int n=9;
    while(n>6) {n--;printf("%d",n);}
  }
```

该程序段的输出结果是_____B_____.
A) 987　　B) 876　　C) 8765
D) 9876

101、有如下程序

```
  #include<stdio.h>
  void main( )
    { int i,sum=0;
      for(i=1;i<=3;sum++) sum+=i;
      printf("%d\n",sum);
    }
```

该程序的执行结果是_____C_____.
A) 6　　B) 3　　C) 死循环
D) 0

102、以下循环体的执行次数是_c___
```
#include<stdio.h>
void main( )
  { int i,j;
    for(i=0,j=1; i<=j+1; i+=2, j--)
      printf("%d \n",i);
  }
```
A) 3　　B) 2　　C) 1
D) 0

103、在执行以下程序时，如果从键盘上输入：ABCdef<回车>，则输出为____B____.
```
#include <stdio.h>
void main( )
  { char ch;
    while ((ch=getchar( ))!='\n')
      { if (ch>='A' && ch<='Z') ch=ch+32;
        else if (ch>='a' && ch<'z') ch=ch-32;
        printf("%c",ch);
      }
    printf("\n");
  }
```
A) ABCdef　　B) abcDEF　　C) abc
D) DEF

104、下面程序的输出结果是____D_____.
```
  main( )
  {
    int i,k=0, a=0, b=0;
    for(i=1;i<=4;i++)
{
 k++;
        if (k%2==0) {a=a+k; continue;}
        b=b+k;
        a=a+k;
    }
    printf("k=%d a=%d b=%d\n",k,a,b);
  }
```
A) k=5 a=10 b=4　　B) k=3 a=6 b=4
C) k=4 a=10 b=3　　D) k=4 a=10 b=4

105、执行下面程序段后,k的值是____D_____.
```
    int i,j,k;
    for(i=0,j=10;i<j;i++,j--)
      k=i+j;
```
A) 9　　B) 11　　C) 8
D) 10

106、下面程序是计算 n 个数的平均值,请填空.____B___
```
#include<stdio.h>
  void main( )
    { int i,n;
      float x,avg=0.0;
      scanf("%d",&n);
      for(i=0;i<n;i++)
        { scanf("%f",&x);
          avg=avg+_____; }
          avg=_____;
      printf("avg=%f\n",avg);
    }
```
A) i　　B) x　　C) x
D) i
  avg/i　　avg/n　　avg/x
avg/n

107、以下程序的功能是:从键盘上输入若干个学生的成绩，统计并输出最高成绩和最低成绩,当输入负数时结束输入。请填空。
__D_____
```
#include<stdio.h>
  void main( )
  { float x,amax,amin;
    scanf("%f",&x);
    amax=x;
    amin=x;
    while (_____ )
      { if (x>amax) amax=x;
        if (_____) amin=x;
        scanf("%f",&x);
      }

printf("\namax=%f\namin=%f\n",amax,amin);
  }
```
A) x<=0　　B) x>0　　C) x>0　　D) x>=0
  x>amin　　x<=amin　　x>amin　　x<amin

108、阅读以下程序，程序运行后的输出结果是__B____.
```
#include<stdio.h>
void   main( )
  { int x;
    for(x=5;x>0;x--)
      if (x--<5) printf("%d,",x);
```

else printf("%d,",x++); }

A) 4,3,2 B) 4,3,1, C) 5,4,2 D) 5,3,1,

109、以下程序段的输出结果是____C____.

```
int k,n,m;
 n=10;m=1;k=1;
 while (k<=n) {m*=2;k+=4;}
 printf("%d\n",m);
```

A) 4 B) 16 C) 8 D) 32

110、下面程序的输出结果是____B_____.

```
#include<stdio.h>
void main( )
{int y=9;
 for(;y>0;y--)
  {if(y%3==0)
    {printf("%d",--y);
      continue;}
  }
}
```

A) 741 B) 852 C) 963 D) 875421

111、下面程序的输出结果是____B____.

```
#include<stdio.h>
void  main( )
 {int x=3;
  do {
    printf("%d ",x-=2);
   }while(!(--x));
 }
```

A) 1 B) 1 -2 C) 3 0 D) 是死循环

112、定义如下变量:

```
int n=10;
```

则下列循环的输出结果是___B____.

```
while(n>7)
 { n--; printf("%d\n",n);}
```

A) 10 B) 9 C) 10 D) 9

9 8 9 8

8 7 8 7

7 6

113、语句 while(!e);中的条件 !e 等价于____A____.

A) e==0 B) e!=1 C) e!=0 D) ~e 每位取反

114、以下叙述正确的是_____B____.

A) continue 语句的作用是结束整个循环的执行

B) 只能在循环体内和 switch 语句体内使用 break 语句

C) 在循环体内使用 break 语句或 continue 语句的作用相同

D) 从多层循环嵌套中退出时,只能使用 goto 语句 因为还可以使用return

115、在下[illegible]循环的程序段是___D_.

A) int i=100;
for(; ;);
while (1)
{ i=i%100+1;
if (i>100) break;
}

B)

C) int k=1000;
do {++k;} while (k>=1000);

D) int s=36;
while (s) --s;

116、下面程序的输出结果是____A_____.

```
#include<stdio.h>
void  main( )
 { int i;
   for(i=1;i<=5;i++)
    { if (i%2) printf("*");
      else continue;
      printf("#");
    }
   printf("$\n");
 }
```

A) *#*#*#$ B) #*#*#*$ C) #*#*$ D) *#*#$

117、下面程序段中,循环体的执行次数是____C_____.

```
int a=10,b=0;
  do {b+=2;a-=2+b;} while(a>=0);
```

A) 4 B) 5 C) 3 D) 2

118、若 i 为整型变量,则以下循环语句的循环次数是_B_.

```
for(i=2;i==0;)
  printf("%d",i--);
```

A) 无限次　　B) 0 次　　C) 1 次
D) 2 次

119、C 语言中 while 和 do-while 循环的主要区别是.__A__

A) do-while 的循环体至少无条件执行一次
B) while 的循环控制条件比 do-while 的循环控制条件严格
C) do-while 允许从外部转到循环体内
D) do-while 的循环体不能是复合语句

120、对于 for(表达式 1;;表达式 3)可理解为_____B____.

A) for(表达式 1;0;表达式 3)
B) for(表达式 1;1;表达式 3)
C) for(表达式 1;表达式 1;表达式 3)
D) for(表达式 1;表达式 3;表达式 3)

121、合法的数组定义是___D_____.

A) int a[6]={"string"};　　B) int a[5]={0,1,2,3,4,5};
C) char a={"string"};　　D) char a[]={0,1,2,3,4,5};

122、要求下面的程序运行后，显示如下结果：

```
2   10   4   6
1    5   2   3
2    4   7   8
5    1   3   2
```

则程序中的划线处应填入___B_____.

```
#include <stdio.h>
void main( )
{ int a[4][4]={ _________ };
  int i,j;
  for(i=0;i<4;i++)
     {for(j=0;j<4;j++)   printf("%4d",a[i][j]);
      printf("\n"); }
}
```

A) {1,5,2,3},{2,4,7,8},{5,1,3,2}
B) {2,10,4,6},{1,5,2,3},{2,4,7,8},{5,1,3,2}
C) {5,1,3,2},{2,4,7,8},{1,5,2,3}
D) {2,1,2,5},{10,5,4,1},{4,2,7,3},{6,3,8,2}

123、给出以下定义:

```
char x[ ]="abcdefg";
char y[ ]={'a','b','c','d','e','f','g'};
```

则正确的叙述为 ___C_____.

A) 数组 x 和数组 y 等价
B) 数组 x 和数组 y 的长度相同
C) 数组 x 的长度大于数组 y 的长度
D) 数组 x 的长度小于数组 y 的长度

前面一个是字符串定义，字符串都有结束符‘\0’

124、定义如下变量和数组:

```
int i;
int
x[4][4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
};
```

则下面语句的输出结果是___C_____.

```
for(i=0;i<4;i++) printf("%3d",x[i][3-i]);
```

A) 1　5　9 13　　B) 1　6 11 16
C) 4　7 10 13　　D) 4　8 12 16

125、下面程序输出的结果是____D____.

```
#include <stdio.h>
void main( )
{ int i,j,x=0;
  static int a[6]={1,2,3,4,5,6};
  for(i=0,j=1;i<5;++i,j++)   x+=a[i]*a[j];
  printf("%d\n",x);
}
```

A) 数组 a 中首尾的对应元素的乘积
B) 数组 a 中首尾的对应元素的乘积之和
C) 数组 a 中相邻各元素的乘积
D) 数组 a 中相邻各元素的乘积之和

126、若希望下面的程序运行后输出 45，程序空白处的正确选择是___C_____.

```
#include <stdio.h>
void main( )
{ int i,j=10,a[ ]={1,3,5,7,9,11,13,15};
  for( _____________ )
     j+=a[i];
  printf("%d\n",j);
}
```

A) i=5;i>1;i--　　B) i=0;i<7;++i
C) i=5;i>=1;--i　　D) i=2;i<6;++i

127、若有以下说明：

```
char s1[ ]={"tree"},s2[]={"flower"};,
```

则以下对数组元素或数组的输出语句中，正确的是__C___.

A) printf("%s%s",s1[5],s2[7]);
B) printf("%c%c",s1,s2);

C) puts(s1);puts(s2);　　D) puts(s1,s2);

128、下列一维数组初始化语句中，正确且与语句 float a[ ]={0,3,8,0,9}; 等价的是_____D___.

A) float a[6]={0,3,8,0,9};　　B) float a[4]={0,3,8,0,9};

C) float a[7]={0,3,8,0,9};　　D) float a[5]={0,3,8,0,9};

129、运行下面程序段的输出结果是___A_____.

```
char s1[10]={'S','e','t','\0','u','p','\0'};
printf("%s",s1);
```

A) Set　　B) Setup　　C) Set up　　D) 'S"e"t'

~~130、以~~下程序段的输出结果是____B____.

```
char s[ ]="an apple";
printf("%d\n",strlen(s));
```

A) 7　　B) 8　　C) 9　　D) 10

131、若有说明:char c[10]={'E','a','s','t','\0'};,则下述说法中正确的是_____D_____.

A) c[7]不可引用　　B) c[6]可引用，但值不确定

C) c[4]不可引用　　D) c[4]可引用，其值为空字符

0x00 NUL(null) 空字符

132、下列初始化语句中，正确且与语句 char c[ ]="string";等价的是____C____.

A) char c[ ]={'s','t','r','i','n','g'};

B) char c[ ]='string';

C) char c[7]={'s','t','r','i','n','g','\0'};

D) char c[7]={'string'};

~~133、若有~~说明 char c[7]={'s','t','r','i','n','g'};则对元素的非法引用是____C____.

A) c[0]　　B) c[9-6]　　C) c[4*2]　　D) c[2*3]

134、如有说明：char s1[5],s2[7];,要给数组 s1 和 s2 整体赋值，下列语句中正确的是____C____.

A) s1=getchar();　s2=getchar();

b

B) scanf("%s%s",s1,s2);

C) scanf("%c%c",s1,s2);

D) gets(s1,s2);

6 2 Z:\未命名1.c [Error] too many arguments to function 'gets'

135、下列一组初始化语句中，正确的是___B_____.

A) int a[8]={　};　　B) int a[9]={0,7,0,4,8};

C) int a[5]={9,5,7,4,0,2};　　D) int a[7]=7*6;

~~136、以~~下程序输出的结果是____D_____.

```
#include <stdio.h>
void main( )
{ char str[ ]="1a2b3c"; int i;
  for(i=0;str[i]!='\0';i++)
    if(str[i]>='0'&&str[i]<='9')
printf("%c",str[i]);
  printf("\n");
}
```

A) 123456789　　B) 1a2b3c　　C) abc　　D) 123

137、以下程序输出的结果是___C_____.

```
#include <stdio.h>
void main( )
{ int a[ ]={5,4,3,2,1},i,j;
  long s=0;
  for(i=0;i<5;i++)　　s=s*10+a[i];
  printf("s=%ld\n",s);
}
```

A) s=12345　　B) s=5 4 3 2 1

C) s=54321　　D) 以上都不对

138、以下程序输出的结果是____A____.

```
#include <stdio.h>
void main( )
{ int a[ ]={1,2,3,4,5},i,j,s=0;
  for(i=0;i<5;i++)　　s=s*10+a[i];
  printf("s=%d\n",s);
}
```

A) s=12345　　B) s=1 2 3 4 5

C) s=54321　　D) s=5 4 3 2 1

~~139、~~在定义 int a[5][6];后，数组 a 中的第 10 个元素是.____C____（设 a[0][0]为第一个元素）

A) a[2][5]　　B) a[2][4]　　C) a[1][3]　　D) a[1][5]

140、当接受用户输入的含有空格的字符串时，应使用____A____函数.

A) gets( )　B) getchar( )　C) scanf( )　D) printf( )

141、以下程序执行时输入 Language Programming<回车>，输出结果是____D____.
#include <stdio.h>
void main( )
{ char str[30];
gets(str);
printf("str=%s\n",str);
}
A) Language Programming B) Language
C) str=Language D) str=Language Programming

142、以下一维数组 a 的正确定义是___D_____ .
A) int a(10); B) int n=10,a[n];
C) int n; D) #define SIZE 10
scanf("%d",&n); int a[SIZE];
int a[n];

143、以下对二维数组 a 进行正确初始化的是__B_____.
A) int a[2][3]={ {1,2},{3,4},{5,6} };
B) int a[ ][3]={1,2,3,4,5,6 };
C) int a[2][ ]={1,2,3,4,5,6};
D) int a[2][ ]={ { 1,2},{3,4}};

144、以下关于数组的描述正确的是____C_____.
A) 数组的大小是固定的，但可以有不同类型的数组元素。
B) 数组的大小是可变的，但所有数组元素的类型必须相同。
C) 数组的大小是固定的，所有数组元素的类型必须相同。
D) 数组的大小是可变的，可以有不同类型的数组元素。

145、以下程序的输出结果是____B____.
#include<stdio.h>
void main()
{ int a[4][4]={{1,3,5,},{2,4,6},{3,5,7}};
printf("%d%d%d%d\n",a[0][0],a[1][1],a[2][2],a[3][3]);
}
A) 0650 B) 1470 C) 5430 D) 输出值不定

146、以下程序的输出结果是___A_____.
#include <stdio.h>
void main()
{ int a[4][4]={{1,3,5,},{2,4,6},{3,5,7}};
printf("%d%d%d%d\n",a[0][3],a[1][2],a[2][1],a[3][0]);
}
A) 0650 B) 1470 C) 5430 D) 输出值不定

147、已知 short int 类型变量占用两个字节，若有定义：short int x[10]={0,2,4}; ，则数组 x 在内存中所占字节数是____D____.
A) 3 B) 6 C) 10 D) 20

148、在定义 int a[5][4]; 之后，对 a 的引用正确的是.____C____
A) a[2][4] B) a[1,3] C) a[4][3] D) a[5][0]

149、以下数组定义中不正确的是____D____.
A) int a[2][3];
B) int b[][3]={0,1,2,3};
C) int c[100][100]={0};
D) int a[3][]={{1,2},{1,2,3},{1,2,3,4}};

150、在执行语句： int a[ ][3]={1,2,3,4,5,6}; 后，a[1][0]的值是____A____.
A) 4 B) 1 C) 2 D) 5

151、以下程序的输出结果是____C____.
#include <stdio.h>
void main()
{ int i,a[10];
for(i=9;i>=0;i--) a[i]=10-i;
printf("%d%d%d",a[2],a[5],a[8]);
}
A) 258 B) 741 C) 852 D) 369

152、以下定义语句中，错误的是___B_____.
A) int a[]={1,2}; B) char a={"test"};
C) char s[10]={"test"}; D) int a[]={'a','b','c'};

153、以下定义语句中，错误的是____D____.
A) int a[]={1,2}; B) char

a[]={"test"};
C) char s[10]={"test"}; D) int n=5,a[n];

154、下列程序的输出结果是____C____.
#include <stdio.h>
void main()
{char b[]="ABCDEFG";
 char p=0;
 while(p<7)
    putchar(b[p++]);
 putchar('\n');
}
A) GFEDCBA B) BCDEFG
C) ABCDEFG D) GFEDCB

155、下述对 C 语言字符数组的描述中错误的是___C__.
A) 字符数组可以存放字符串
B) 字符数组中的字符串可以整体输入、输出
C) 可以在赋值语句中通过赋值运算符"="对字符数组整体赋值
D) 不可以用关系运算符对字符数组中的字符串进行比较

这句不是赋值而是初始化；讲的赋值语句是在程序执行中的赋值。对于字符而言，要赋值就要采用strcpy(s1, s2)这个函数

156、以下程序的输出结果是____A____.
#include<stdio.h>
void main()
{ int i,x[3][3]={1,2,3,4,5,6,7,8,9};
    for(i=0;i<3;i++)
       printf("%d,",x[i][i]);
    printf("\n");
}
A) 1,5,9, B) 1,4,7, C) 3,5,7, D) 3,6,9,

157、如有定义语句 int a[]={1,8,2,8,3,8,4,8,5,8};，则数组 a 的大小是___A_____.
A) 10 B) 11 C) 8
D) 不定

158、有如下程序
#include<stdio.h>
void main()
{ char ch[80]="123abcdEFG*&";
    int j;long s=0;
    puts(ch);
    for(j=0;ch[j]>'\0';j++)
    if(ch[j]>='A'&&ch[j]<='Z')
ch[j]=ch[j]+'e'-'E';
    puts(ch);
}
该程序的功能是____D____.
A) 测字符数组 ch 的长度
B) 将数字字符串 ch 转换成十进制数
C) 将字符数组 ch 中的小写字母转换成大写
D) 将字符数组 ch 中的大写字母转换成小写

puts(ch);puts()函数用来向标准输出设备（屏幕）写字符串并换行，其调用方式为，puts(s);其中s为字符串字符（字符串数组名或字符串指针）。

159、设有如下定义:
char aa[2][20]={ "abcd", "ABCD"};
则以下说法中错误的是____D____.
A) aa 是个二维数组，可以存放 2 个 19 个字符以下的字符串
B) aa 是个二维数组，每行中分别存放了字符串"abcd"和"ABCD"
C) aa[0]可以看作是一维数组名
D) aa[0][0]可以看作是一维数组名

160、以下对 C 语言函数的有关描述中,正确的是___A__.
A) 在 C 中,调用函数时,只能把实参的值传送给形参,形参的值不能传送给实参
B) C 函数既可以嵌套定义又可以递归调用
C) 函数必须有返回值,否则不能使用函数
D) C 程序中有调用关系的所有函数必须放在同一个源程序文件中

```
void foo()
{
   void bar()
   {
// Some code
   }
}
```

C语言可以递归调用，但是(函数)不能嵌套定义

161、有如下说明：int a[10]={0,1,2,3,4,5,6,7,8,9};
则数值不为 9 的表达式是____B____.
A) a[10-1] B) a[8] C) a[9]-0 D) a[9]-a[0]

162、设有数组定义:char array[]="China";则数组 array 所占的存储空间为____C____.
A) 4 个字节 B) 5 个字节
C) 6 个字节 D) 7 个字节

163、下面程序的输出是____B____.
#include <stdio.h>
int m=13;
int fun2(int x, int y)
{ int m=3;
  return(x*y-m);
}
void main()
{ int a=7, b=5;
  printf("%d\n",fun2(a,b)/m);

}
A) 1 B) 2 C) 7
D) 10

164、请读程序:
#include <stdio.h>
f(int b[ ], int n)
{ int i, r=0;
for(i=0; i<=n; i++) r=r+b[i];
return r;
}
void main()
{
int x, a[]={ 2,3,4,5,6,7,8,9};
x=f(a, 3);
printf("%d\n",x); }
上面程序的输出结果是___B_____.
A) 20 B) 14 C) 9
D) 5

165、请读程序:
#include <stdio.h>
f(int b[ ], int n)
{ int i, r=1;
for(i=0; i<=n; i++) r=r*b[i];
return r; }
void main()
{ int x, a[]={ 2,3,4,5,6,7,8,9};
x=f(a, 3);
printf("%d\n",x); }
上面程序的输出结果是___B_____.
A) 720 B) 120 C) 24
D) 6

166、请读程序:
#include<stdio.h>
f(char s[ ])
{ int i,j;
i=0;j=0;
while(s[j]!= '\0') j++;
return (j-i); }
void main()
{printf("%d\n",f("ABCDEF"));}
上面程序的输出结果是____B____.
A) 0 B) 6 C) 7
D) 8

167、对以下程序，正确的说法是__D_____.
sub (char x,char y)
{ int z; z=x%y; return z; }
void main( )
{ int g=5,h=3,k;
k=sub(g,h);
printf("%d\n",k); }
A) 实参与其对应的形参类型不一致，程序不能运行
B) 被调函数缺少数据类型说明，程序不能运行
C) 主函数中缺少对被调函数的说明语句，程序不能运行
D) 程序中没有错误，可以正常运行

168、若已定义实参数组 int a[3][4]={2,4,6,8,10};，则在被调用函数 f 的下述定义中，对形参数组 b 定义正确的选项是_____B___.
A) f(int b[ ][6]) B) f(b) int b[ ][4];
C) f(int b[3][ ]); D) f(b) int b[4][5];

169、若函数调用时用数组名作为函数参数，以下叙述中，不正确的是____C____.
A) 实参与其对应的形参共占用同一段存储空间
B) 实参将其地址传递给形参，结果等同于实现了参数之间的双向值传递
C) 实参与其对应的形参分别占用不同的存储空间
D) 在调用函数中必须说明数组的大小，但在被调函数中可以使用不定尺寸数组

170、以下叙述中，不正确的是____B____.
A) 使用 static float a 定义的外部变量存放在内存中的静态存储区
B) 使用 float b 定义的外部变量存放在内存中的动态存储区
C) 使用 static float c 定义的内部变量存放在内存中的静态存储区
D) 使用 float d 定义的内部变量存放在内存中的动态存储区

171、如果一个函数位于 C 程序文件的上部，在该函数体内说明语句后的复合语句中定义了一个变量，则该变量___C_____.
A) 为全局变量，在本程序文件范围内有效
B) 为局部变量，只在该函数内有效
C) 为局部变量，只在该复合语句中有效
D) 定义无效，为非法变量

172、调用函数时，当实参和形参都是简单变

量时，它们之间数据传递的过程是____D____.
A) 实参将其地址传递给形参，并释放原先占用的存储单元
B) 实参将其地址传递给形参，调用结束时形参再将其地址回传给实参
C) 实参将其值传递给形参，调用结束时形参再将其值回传给实参
D) 实参将其值传递给形参，调用结束时形参并不将其值回传给实参

173、以下叙述中，不正确的是___B_____.
A) 在同一 C 程序文件中，不同函数中可以使用同名变量
B) 在 main 函数体内定义的变量是全局变量
C) 形参是局部变量，函数调用完成即失去意义
D) 若同一文件中全局变量和局部变量同名，则全局变量在局部变量作用范围内不起作用

174、若函数调用时参数为基本数据类型的变量(俗称简单变量)，以下叙述正确的是____C____.
A) 实参与其对应的形参共占存储单元
B) 只有当实参与对应的形参同名时才共占存储单元
C) 实参与其对应的形参分别占用不同的存储单元
D) 实参将数据传递给形参后，立即释放原先占用的存储单元

175、若主调用函数类型为 double，被调用函数定义中没有进行函数类型说明，而 return 语句中的表达式类型为 float 型，则被调函数返回值的类型是____A____。
A) int 型　　B) float 型
C) double 型　　D) 由系统当时的情况而定

~~176、~~在以下叙述中，不正确的选项是____B____。
A) C 语言程序总是从 main( )函数开始执行
B) 在 C 语言程序中，被调用的函数必须在 main( )函数中定义
C) C 程序是函数的集合，在这个函数集中包括标准函数和用户自定义函数
D) 在 C 语言程序中，函数的定义不能嵌套，但函数的调用可以嵌套

177、C 语言中，若未说明函数的类型，则系统默认该函数的类型是___C_____。
A) float 型　　B) long 型　C) int 型　　D) double 型

178、下面函数的功能是____B____。
```
sss(s, t)
char s[ ], t[ ];
{ int i=0;
while((s[i])&&(t[i])&&(t[i]== s[i]))    i++;
return    (s[i]-t[i]);}
```
A) 求字符串的长度
B) 比较两个字符串的大小
C) 将字符串 s 复制到字符串 t 中
D) 将字符串 s 接续到字符串 t 中

~~179、~~设有如下函数定义:
```
 int f(char s[ ])
 { int i=0;
    while(s[i++]!='\0');
    return (i-1);   }
```
如果在主程序中用下面的语句调用上述函数，则输出结果为___C_____.
```
printf("%d\n",f("goodbey!"));
```
A) 3　　B) 6　　C) 8　　D) 9

180、对于 C 语言的函数，下列叙述中正确的是___A___.
A) 函数的定义不能嵌套，但函数调用可以嵌套
B) 函数的定义可以嵌套，但函数调用不能嵌套
C) 函数的定义和调用都不能嵌套
D) 函数的定义和调用都可以嵌套

~~181、以~~下说法中正确的是_____C___.
A) C 语言程序总是从第一个定义的函数开始执行
B) 在 C 语言程序中,要调用的函数必须在 main()函数中定义
C) C 语言程序总是从 main()函数开始执行
D) C 语言程序中的 main()函数必须放在程序的开始部分

~~182、以下程~~序的输出结果是____B_____.
```
#include <stdio.h>
int a,b;
void fun()
{ a=100;   b=200; }
```

```
void main()
{ int a=5,b=7;
    fun();
    printf("%d%d\n",a,b);  }
```

A) 100200 B) 57 C) 200100 D) 75

183、以下函数 func()的功能是：使具有 n 个元素的一维数组 b 的每个元素的值都增加 2，划线处应填入__D_.

```
func(int  b[ ],int n)
{ int ;
for(i=0;i<n;i++)
 _____;  }
```

A) b[i++] B) b[i]++ C) b[i+=2] D) b[i]+=2

184、设有以下函数：

```
#include <stdio.h>
int f(int  a)
{ int b=0,c;
  c=3;
  b++ ;  c++;
  return (a+b+c);  }
```

如果在下面的程序中调用该函数，则输出结果是__B__.

```
void main()
{ int i;
  for(i=0;i<3;i++)  printf("%d\n",f(i));  }
```

A) 5 B) 5 C) 3 D) 3
7 6 4 3
9 7 5 3

185、设有以下函数：

```
#include <stdio.h>
int f(int a)
{ int b=0;
    static  c=3;
    b++ ;  c++;
    return (a+b+c);  }
```

如果在下面的程序中调用该函数，则输出结果是___A_.

```
void main()
{ int a=2,i;
    for(i=0;i<3;i++)  printf("%d\n",f(a));  }
```

A) 7 B) 7 C) 7 D) 7
8 9 10 7
9 11 13 7

186、在调用函数时，如果实参是简单变量，它与对应形参之间的数据传递方式是_____B___.

A) 地址传递 B) 单向值传递
C) 由实参传给形参，再由形参传回实参
D) 传递方式由用户指定

187、C 语言规定，除主函数外，程序中各函数之间__A__.

A) 既允许直接递归调用也允许间接递归调用
B) 不允许直接递归调用也不允许间接递归调用
C) 允许直接递归调用不允许间接递归调用
D) 不允许直接递归调用允许间接递归调用

在函数a(或过程)中直接引用(调用)函数a本身 间接递归调用就是在函数a(或过程)中调用另外一个函数b,而该函数b又引用(调用)了函数a

188、以下函数 fun 形参的类型是____D____.

```
  fun( float  x)
  { float  y;
    y=3*x-4;
    return  y;  }
```

A) int B) 不确定 C) void D) float

189、下面程序的输出是____C____.

```
int fun3(int x)
{static int a=3;
  a+=x;
  return(a);  }
void main()
{int k=2,m=1,n;
  n=fun3(k);
  n=fun3(m);
  printf("%d\n",n);  }
```

A) 3 B) 4 C) 6 D) 9

190、下列程序执行后输出的结果是___C_____.

```
#include<stdio.h>
int f(int a)
{int b=0;
  static c=3;
  a=c++,b++;
```

```
  return(a);  }
void main()
{int a=2,i,k;
  for(i=0;i<3;i++)
     k=f(a++);
  printf("%d\n",k);  }
```

A) 3　　B) 0　　C) 5　　D) 4

~~191~~、C 语言中规定函数的返回值的类型是由__D_____.

A) return 语句中的表达式类型所决定

B) 调用该函数时的主调用函数类型所决定

C) 调用该函数时系统临时决定

D) 在定义该函数时所指定的类型所决定

~~192~~、以下程序的输出结果是_____C_____.

```
  #include <stdi0.h>
  fun(int x,int y,int z)
  { z=x*x+y*y;}
  void main()
  { int a=31;
    fun(5,2,a);
    printf("%d",a);  }
```

A) 0　　B) 29　　C) 31　　D) 无定值

~~193~~、~~以下~~函数调用语句中含有实参个数为____B____.

```
    func((exp1,exp2),(exp3,exp4,exp5));
```

A) 1 个　　B) 2 个　　C) 4 个　　D) 5 个

~~194~~、~~以~~下程序的输出结果是_____D___.

```
  #include <stdio.h>
  void fun()
   { static int a=0;
    a+=2;
    printf("%d",a);  }
  void main()
   { int cc;
    for(cc=1;cc<4;cc++) fun( );
    printf("\n");  }
```

A) 2222　　B) 2468　　C) 222　　D) 246

~~195~~、~~有~~如下程序

```
  int func(int a,int b)
  { return(a+b); }
  void main()
  { int x=2,y=5,z=8,r;
    r=func(func(x,y),z);
    printf("%d\n",r);  }
```

该程序的输出结果是_____D_____.

A) 12　　B) 13　　C) 14　　D) 15

~~196~~、有如下函数调用语句 func(rec1,rec2+rec3,rec4, rec5);该函数调用语句中,含有的实参个数是___B_____.

A) 3　　B) 4　　C) 5　　D) 有语法错误

~~197~~、有如下函数调用语句 func(rec1,rec2+rec3,(rec4, rec5));该函数调用语句中,含有的实参个数是____A___.

A) 3　　B) 4　　C) 5　　D) 有语法错误

198、在一个 C 源程序文件中,要定义一个只允许本源文件中所有函数使用的全局变量,则该变量需要使用的存储类别是:____D____.

A) extern　　B) register　　C) auto　　D) static

199、C 语言中,凡未指定存储类别的局部变量的隐含存储类别是____A____.

A) auto　　B) static　　C) extern　　D) register

200、在 C 语言中,全局变量的存储类别是____A_____.

A) static　　B) extern　　C) void　　D) registe

C语言中，存储类型是指变量存储的内存类型。C语言里面有三种：普通内存、堆栈以及寄存器。（1）普通内存中是静态变量，在代码块之外声明。静态变量在程序运行之前就已经创建好，这程序执行期间一直存在。（2）堆栈中是自动变量，在代码块中缺省的声明就是自动变量。执行执行到代码块的时候才创建出来。（3）寄存器的变量使用register来声明的，存储在硬件寄存器中，而不是内存中。

内存中供用户使用的存储空间分为代码区与数据区两个部分。变量存储在数据区，数据区又可分为静态存储区与动态存储区。
静态存储是指在程序运行期间给变量分配固定存储空间的方式。如全局变量存放在静态存储区中，程序运行时分配空间，程序运行完释放。
动态存储是指在程序运行时根据实际需要动态分配存储空间的方式。如形式参数存放在动态存储区中，在函数调用时分配空间，调用完成释放。

https://blog.csdn.net/Azhanzaitianbian123/article/details/88901640

2、C++选择题集

1.下列的各类函数中，不是类的成员函数。（ C ）

A)构造函数 B)析构函数 C)友元函数 D)拷贝初始化构造函数

2.作用域运算符“::”的功能是:（ B ）

A)标识作用域的级别的 B)指出作用域的范围的

C)给定作用域的大小的 D)标识成员是属于哪个类的

3.下列说明中 const char *ptr；其中 ptr 应该是:（ C ）

A)指向字符常量的指针 B)指向字符的常量指针

C)指向字符串常量的指针 D)指向字符串的常量指针

4.已知：print()函数是一个类的常成员函数，它无返回值，下列表示中，是正确的。（ A ）

不改变对象的成员函数的函数原型中加上const说明。

A)void print()const; B)const void print();

C)void const print() : D)void print(const);

5.在类定义的外部，可以被访问的成员有（ C ）

A)所有类成员 B)private 的类成员

C)public 的类成员 D)public 或 private 的类成员

6.要求打开文件"d:\file.dat"，可写入数据，正确的语句是:（ B ）

ifstream infile("d:\file.dat", ios::in);

ifstream infile("d:\\file.dat", ios::in);

ofstream infile("d:\file.dat", ios::out);

fstream infile("d:\\file.dat", ios::in|ios::out);

7.关于类和对象不正确的说法是:（ C ）

A)类是一种类型，它封装了数据和操作

B)对象是类的实例

C)一个类的对象只有一个

D)一个对象必属于某个类

8.在 C++中，类与类之间的继承关系具有（ C ）

A)自反性 B)对称性 C)传递性 D)反对称性

9．结构化程序设计所规定的三种基本控制结构是:（ C ）

A)输入、处理、输出 B)树形、网形、环形

C)顺序、选择、循环 D)主程序、子程序、函数

10.C++语言是从早期的 C 语言逐渐发展演变而来的.与 C 语言相比，它在求解问题方法上进行的最大改进是:（ B ）

A)面向过程 B)面向对象 C)安全性 D)复用性

11.在 C++中实现封装是借助于(B)

A)枚举 B) 类 C) 数组 D) 函数

12.C++语言是从早期的 C 语言逐渐发展演变而来的与 C 语言相比，它在求解问题方法上进行的最大改进是：（ B ）

A)面向过程 B)面向对象 C)安全性 D)复用性

13.在 C++中用（ D ）能够实现将参数值带回。

a）数组 b）指针 c）引用 d）上述 ABC 都可

14.考虑函数原型 void test(int a,int b=7,char ch="*"),下面的函数调用中，属于不合法调用的是：（ C ）

A)test(5) B)test(5,8) C)test(6,"#") D)test(0,0,"*");

15.下列关于类和对象的叙述中，错误的是：（ A ）

A)一个类只能有一个对象； B)对象是类的具体实例；

C)类是对某一类对象的抽象； D)类和对象的关系是一种数据类型与变量的关系。

16.已知: int m=10；下列表示引用的方法中，正确的是：（ C ）

A)int &Z； B)int &t=10； C)int &X=m； D)float &f=&m；

17.对于 int *pa[5]；的描述中，正确的是:（ D ）

A)pa 是一个指向数组的指针，所指向的数组是 5 个 int 型元素

B)pa 是一个指向某数组中第 5 个元素的指针，该元素是 int 型变量

C)pa[5]表示数组的第 5 个元素的值，是 int 型的值

D)pa 是一个具有 5 个元素的指针数组，每个元素是一个 int 型指针

18.定义析构函数时，应该注意：（ C ）

A)其名与类名完全相同 B)返回类型是 void 类型

C)无形参，也不可重载 D)函数体中必须有 delete 语句

19.在公有继承的情况下，基类的成员（私有的除外）在派生类中的访问权限（ B ）

A)受限制 B)保持不变 C)受保护 D)不受保护

20.下列不是函数重载所要求的条件是:（ D ）

A)函数名相同 B)参数个数不同

C)参数类型不同 D)函数返回值类型不同

21.在 C++中执行以下 4 条语句后输出 rad 值为：(C)

```
static int hot=200;
int &rad=hot;
hot=hot+100;
cout<<rad<<endl;
```

A) 0 B) 100 C) 300 D) 200

22.类 clase CC1 的说明如下，错误的

语句是：（ A ）

C++类中成员变量的初始化有两种方式：

构造函数初始化列表和构造函数体内赋值。

```
clase CC1{
    int a=2;          //(A)
    CC1();            //(B)
public:
    CC1(int val);     //(C)
    ~CC1();           //(D)
};
```

23.下列各类函数中，不是类的成员函数是：（ C ）

A)构造函数　　B)析构函数

C)友元函数　　D)拷贝构造函数

24.类的指针成员的初始化是通过函数完成的，这个函数通常是：(C)

A)析构函数　　B) 构造函数　　C)其它成员函数　　D)友元函数

25.关于构造函数的说法，不正确的是：（ A ）

A)没有定义构造函数时，系统将不会调用它　　B)其名与类名完全相同

C)它在对象被创建时由系统自动调用

D)没有返回值

26.按解释中的要求在下列程序划线处填入的正确语句是：（ C ）

```
#include <iostream.h>
class Base{
public:
   void fun(){cout<<"Base::fun"<<endl；}
};
class Derived:public Base{
public：
    void fun()
    {_____    //在此空格处调用基类的函数 fun()
    cout<<"Derived::fun"<<endl；}
};
```

A)fun() ；　　B)Base.fun() ；

C)Base::fun()；　　D)Base->fun()；

27.在 C++中编译器所生成的文件的扩展名为（ B ）

A) .cpp　　B) .obj　　C) .lib　　D) .exe

28.在保护继承的情况下，基类的成员（私有的除外）在派生类中的访问权限（ C ）

A)受限制　　B)保持不变　　C)受保护　　D)不受保护

29.有如下一段程序：

```
int f1(float);
int f2(char);
int f3(float);
int f4(float);
int（*pf）（float）；
```

则以下不合法的是：（ C ）

A)int(*p)(float)=&f1；　　B)pf=&f4；

C)pf=&f2；　　D)pf=&f3；

30.通常拷贝构造函数的参数是：（ C ）

A)对象名　　B)对象的成员名　　C)对象的引用名　　D)对象的指针名

31.系统在调用重载函数时，不能作为确定哪个重载函数被调用的依据是：

（ D ）

A)参数个数 B)参数类型 C)函数名称 D)函数的返回值类型

32.下列描述中，正确的是：（ D ）

A)所有的运算符号都可以重载。

B)基类类型的指针可以指向子类，子类类型的指针也可以指向基类。

C)可以在类的构造函数中对静态数据成员进行初始化。

D)动态联编要满足两个条件：被调用的成员函数是虚函数；用指针或引用调用虚函数。

33.关于成员函数特征的描述中，错误的是：（ A ）

A) 成员函数一定是内联函数

B)成员函数可以重载

C)成员函数可以设置参数的默认值

D)成员函数可以是静态的

34.下列关于构造函数的描述中，错误的是：（ D ）

A)构造函数可以设置默认参数； B)构造函数在定义类对象时自动执行

C)构造函数可以是内联函数； D)构造函数不可以重载

35.下面描述中，表达错误的是：（ B ）

A)公有继承时，基类中的 public 成员在派生类中仍是 public 的

B)公有继承时，基类中的 private 成员在派生类中仍是 private 的

C)公有继承时，基类中的 protected 成员在派生类中仍是 protected 的

D)私有继承时，基类中的 public 成员在派生类中是 private 的

36.C++语言是从早期的 C 语言逐渐发展演变而来的，与 C 语言相比，C++在求解问题方法上进行的最大改进是：（ B ）

A)面向过程 B)面向对象 C)安全性 D)复用性

37.有如下类声明“class A{int x； ……}；”，则 A 类的成员 x 是：（ B ）

A)公有数据成员 B)私有数据成员

C)公有成员函数 D)私有成员函数

38.有如下程序：执行后的输出结果应该是：（ A ）

```
#include<iostream.h>
class cla{
    static int n;
public:
    cla(){n++；}
    ~cla(){n--；}
    static int get_n(){return n；}
}；
int cla::n=0;
int main()
{
    cla * p=new cla;
    delete p;
    cout<<"n="<<cla::get_n()<<endl;
    return 0;
}
```

A)n=0 B)n=1 C)n=3

D)n=4

39.有如下程序：执行后的输出结果应该是：（ A ）

```
#include <iostream.h>
class A{
public:
    A(){cout<<"A"；}
};
class C:public A{
public:
    C(){cout<<"C"；}
};
void main(){C cobj；}
```

A) AC B) CA C) A D) C

40.有如下程序：执行后的输出结果应该是：（ D ）

```
#include <iostream.h>
class BASE{
public:
    ~BASE(){cout<<"BASE"；}
};
class DERIVED: public BASE{
public:
    ~DERIVED(){cout<<"DERIVED"；}
};
void main(){DERIVED x；}
```

A)BASE B)DERIVED

C)BASEDERIVED

D)DERIVEDBASE

41.类 TM 和类 TN 是两个独立的类，那么类 TM 中哪种形式的数据成员一定是不允许的：(C)

A)TM * a B)TN & a C)TM a

D)TN a；

42.通过下述哪一项调用虚函数时，采用动态联编。（ A ）

A)对象指针 B)对象名 C)成员名限定 D)派生类名

43.在下述哪种情况下适宜采用 inline 定义内联函数。（ C ）

A)函数体含有循环语句 B)函数体含有递归语句

C)函数代码少、频繁调用 D)函数体含有 switch 语句

44.创建动态对象，需要使用操作符（ A ）

A) new B) delete C) . D) *

45.删除动态对象，需要使用操作符（ B ）

A) new B) delete C) . D) *

46.在 C++中不能被重载的一组运算符是：（ A ）

A):: ?: . sizeof ++ ->

C):: ?: new delete D)++ -- ?: sizeof

47.如果类 A 被说明成类 B 的友元，则（ C ）

A)类 A 的成员即是类 B 的成员

B)类 B 的成员即是类 A 的成员

C)类 A 的成员函数可以访问类 B 的所有成员

D)类 B 的成员函数可以访问类 A 的所

有成员

48.下面关于友元的描述中，错误的是：（ D ）

A)友元函数可以直接访问该类的私有成员。

B)一个类的友元类中的所有成员函数都是这个类的友元函数。

C)利用友元可以提高程序的运行效率，但却破坏了封装性。

D)友元关系不能被继承，是双向可交换的。

49.下列虚基类的声明中，正确的是：（ B ）

A)class virtual B: public A B)class B: virtual public A

C)class B: public A virtual D)virtual class B: public A

50.C++设置虚基类的目的是(A)

A)消除二义性 B)简化程序 C)提高运行效率 D)减少目标代码

51.有如下模板定义：

template <class T>T fun(T x,T y){return x*x+y*y; }

函数模版实例时会根据你的参数而自动转换,就好像A中3,5,T会自动变成int.而D,已经说明是int类型,所以5.5会变成5.但是C的话,3和5.5,此时就不知是int或者是double/float了,除非3改写成3.0

在下列对 fun 的调用中，错误的是：（ C ）

A)fun(2,8) B)fun(2.0,8.2)

C)fun(2.3,8) D)fun<float>(2,8.3)

52.通常情况下，函数模板中的类型参数个数不能是(A)

A)0 个 B)1 个 C)2 个 D)3 个

53.以下关于 this 指针的叙述中正确的是：（ D ）

A)任何与类相关的函数都有 this 指针；

B)类的成员函数都有 this 指针；

C)类的友员函数都有 this 指针；

D)类的非静态成员函数才有 this 指针；

54.当一个类的某个函数被说明为 virtual 时，在该类的所有派生类中的同原型函数(A)

A)都是虚函数 B)只有被重新说明时才是虚函数

C)都不是虚函数 D)只有被重新说明为 virtual 时才是虚函数

55.在派生类中重新定义虚函数时，必须在下列哪方面与基类保持一致。(A)

A)参数类型 B)参数名字 C)操作内容 D)赋值

56.关于 const 关键字说法错误的是(D)

A)const 关键字可以修饰对象和成员函数 B)const 对象不能被修改

C)const 成员函数不能修改类数据成员

D)const 可以用于说明类

57.执行如下程序后的输出结果是：（ B ）

```
#include<iostream.h>
class test{
    static int count;
public:
    test(){count++; }
    ~test(){count--; }
```

```
    static int getCount(){return count; }
};
int test::count=0;
int main()
{
    test * p=new test;
    test * q=new test;
    delete p;
    cout<<"count="<<test::getCount()<<
endl;
    return 0;
}
```

A)count=0 B)count=1

C)count=2 D)count=3

58.关于类模板的说法正确的是：

（ B ） 泛型

A)类模板的主要作用是生成抽象类

B)类模板实例化时，编译器将根据给出的模板实参生成一个类

C)在类模板中的数据成员具有同样类型

D)类模板中的成员函数没有返回值

59.有如下说明语句 int a[10]={1,2,3,4,5,6,7,8,9,10}; int *p=a; 则数值为 9 的表达式是：（ B ）。

A)*p+9 B)*(p+8) C)*p+=9

D)p+7

60.以下关键字不能用来声明类的访问权限的是：（ C ）

A)public B)private C)static

D)protected

61.实现运行时的多态性采用(D)

A)重载函数 B)构造函数 C)析构函数 D)虚函数

62.下列说明语句中正确的是：

（ D ）

A)int a,&ra=a,&&ref=ra; B)int &refa[10];

C)int a,&ra=a,&*refp=&ra; D)int *pi, *&pref=pi;

63.进行文件操作时需要包含头文件

（ C ）

A)iostream.h B)stdio.h

C)fstream.h D)stdlib.h

64.使用如 setw()的格式操纵符时需要包含头文件（ C ）

A)iostream.h B)fstream.h

C)iomanip.h D)stdlib.h manipulator(操纵器)的缩写

65.对 C++语言和 C 语言的兼容性，描述正确的是：（ A ）

A)C++兼容 C B)C++部分兼容 C

C)C++不兼容 C D)C 兼容 C++

66. 在 C++中使用流进行输入输出，其中用于屏幕输出的对象是：

（ C ）

A)cerr B)cin C)cout D)cfile

67. 对使用关键字 new 所开辟的动态存储空间，释放时必须使用（ C ）

A)free B)create C)delete

D)realse

68. 如没有使用 private 关键字定义类的数据成员，则默认为（ A ）

A)private B)public C)protected

D)friend

69. 使用值传递方式将实参传给形参，下列说法正确的是：（ A ）

A)形参是实参的备份　　B)实参是形参的备份

C)形参和实参是同一对象　　D)形参和实参无联系

70. 在函数调用时，如某一默认参数要指明一个特定值，则有（ A ）

A)其之前所有参数都必须赋值

B)其之后所有参数都必须赋值

C)其前、后所有参数都必须赋值

D)其前、后所有参数都不必赋值

71. 设存在函数 int max(int，int)返回两参数中较大值，若求 22，59，70 三者中最大值，下列表达式不正确的是：（ C ）

A)int m = max(22，max(59，70))；

B)int m = max(max(22，59)，70)；

C)int m = max(22，59，70)；

D)int m = max(59，max(22，70))；

72. 下列哪个类型函数不适合声明为内联函数（ A ）

A)函数体语句较多　　B)函数体语句较少　　C)函数执行时间较短　　D)函数执行时间过长

内联函数是以内存换取时间，所以选d

73. int Func(int，int)；不可与下列哪个函数构成重载（ B ）

A)int Func(int，int，int)；

B)double Func(int，int)；

C)double Func(double，double)；

D)double Func(int，double)；

74. 类的私有成员可在何处访问（ D ）

A)通过子类的对象访问　　B)本类及子类的成员函数中

C)通过该类对象访问　　D)本类的成员函数中

75. 如果没有为一个类定义任何构造函数的情况下，下列描述正确的是：（ A ）

A)编译器总是自动创建一个不带参数的构造函数　　B)这个类没有构造函数

C)这个类不需要构造函数

D)该类不能通过编译

76. 一个类可包含析构函数的个数是：（ B ）

A)0 个　　B)1 个　　C)0 个或 1 个　　D)0 个或多个

77. 一个类可包含构造函数的个数是：（ D ）

A)0 个　　B)0 个或 1 个　　C)0 个或多个　　D)1 个或多个

78. this 指针存在的目的是：（ B ）

A)保证基类公有成员在子类中可以被访问

B)保证每个对象拥有自己的数据成员，但共享处理这些数据成员的代码

C)保证基类保护成员在子类中可以被访问

D)保证基类私有成员在子类中可以被访问

实际上对象在实例化的时候，体积是非常小的，并没有每一个对象都保存函数，同一个类的对象的函数实际上只有一份副本（二进制层面），那如何来区分到底是谁调用了函数？就是这个this指针。

79．下列关于类的权限的描述错误的是：（ A ）

A)类本身的成员函数只能访问自身的私有成员

B)类的对象只能访问该类的公有成员

C)普通函数不能直接访问类的公有成员，必须通过对象访问

D)一个类可以将另一个类的对象作为成员

80．在编译指令中，宏定义使用哪个指令（ B ）

A)#include B)#define C)#if D)#else

81．设类 A 将其它类对象作为成员，则建立 A 类对象时，下列描述正确的是：（ B ）

A)A 类构造函数先执行 B)成员构造函数先执行

C)两者并行执行 D)不能确定

82．下列描述错误的是：（ A ）

A)在创建对象前，静态成员不存在

B)静态成员是类的成员

C)静态成员不能是虚函数

D)静态成员函数不能直接访问非静态成员

83．对于友元描述正确的是：（ B ）

A)友元是本类的成员函数 B)友元不是本类的成员函数

C)友元不是函数 D)友元不能访问本类私有成员

84．在哪种派生方式中，派生类可以访问基类中的 protected 成员（ B ）

A)public 和 private B)public 和 protected

C)protected 和 private D)仅 protected

85.在每个 C++程序中都必须包含的一个函数是（ B ）

A)main() B)MAIN() C)name() D)function()

86.设 x 和 y 均为 bool 量，则 x&&y 为真的条件是（ A ）

A)它们均为真 B)其中一个为真

C)它们均为假 D)其中一个为假

87.下面的哪个保留字不能作为函数的返回类型（ C ）

A)void B)int C)new D)long

88.假定 a 为一个整型数组名，则元素 a[4]的字节地址为（ C ）

A)a+4 B)a+8 C)a+16 D)a+32

89.假定有类 AB，有相应的构造函数定义，能正确执行“AB a(4),b(5), c[3] , *p[2]={&a,&b};”语句，请问执行完此语句后共调用该类构造函数的次数为（ C ）

A)3 B)4 C)5 D)6

3、C++试题

1.下列的各类函数中，不是类的成员函数。（ C ）

A)构造函数 B)析构函数 C)友元函数 D)拷贝初始化构造函数

2.作用域运算符“::”的功能是：（ B ）

A)标识作用域的级别的 B)指出作用域的范围的

C)给定作用域的大小的 D)标识成员是属于哪个类的

3.下列说明中 const char *ptr；其中 ptr 应该是：（ C ） A)指向字符常量的指针 B)指向字符的常量指针 C)指向字符串常量的指针 D)指向字符串的常量指针

4.已知：print()函数是一个类的常成员函数，它无返回值，下列表示中，是正 确的。（ A ） A)void print()const； B)const void print()； C)void const print()： D)void print(const)；

5.在类定义的外部，可以被访问的成员有（ C ） A)所有类成员 B)private 的类成员 C)public 的类成员 D)public 或 private 的类成员

6.要求打开文件"d:\file.dat"，可写入数据，正确的语句是：（ B ） ifstream infile("d:\file.dat", ios::in)； ifstream infile("d:\\file.dat", ios::in)； ofstream infile("d:\file.dat", ios::out)； fstream infile("d:\\file.dat", ios::in|ios::out)；

7.关于类和对象不正确的说法是：（ C ） A)类是一种类型，它封装了数据和操作 B)对象是类的实例 C)一个类的对象只有一个 D)一个对象必属于某个类

8.在 C++中，类与类之间的继承关系具有（ C ） A)自反性 B)对称性 C)传递性 D)反对称性

9．结构化程序设计所规定的三种基本控制结构是：（ C ）

A)输入、处理、输出 B)树形、网形、环形

C)顺序、选择、循环 D)主程序、子程序、函数

10.C++语言是从早期的 C 语言逐渐发展演变而来的.与 C 语言相比，它在求解问 题方法上进行的最大改进是：（ B ）

A)面向过程 B)面向对象 C)安全性 D)复用性

11.在 C++中实现封装是借助于(B)
A)枚举 B) 类 C) 数组 D)函数

13.在 C++中用（ D ）能够实现将参数值带回。
a）数组 b）指针 c）引用 d）上述 ABC 都可

14.考虑函数原型 void test(int a,int b=7,char ch="*"),下面的函数调用 中，属于不合法调用的是：（ C ）
A)test(5) B)test(5,8) C)test(6,"#") D)test(0,0,"*")；
15.下列关于类和对象的叙述中，错误的是：（ A ）
A)一个类只能有一个对象； B)对象是类的具体实例； C)类是对某一类对象的抽象； D)类和对象的关系是一种数据类型与变量的关系。
16.已知: int m=10；下列表示引用的方法中，正确的是：（ C ） A)int &Z； B)int &t=10； C)int &X=m； D)float &f=&m；

17.对于 int *pa[5]；的描述中，正确的是:（ D ）
A)pa 是一个指向数组的指针，所指向的数组是 5 个 int 型元素 B)pa 是一个指向某数组中第 5 个元素的指针，该元素是 int 型变量 C)pa[5]表示数组的第 5 个元素的值，是 int 型的值 D)pa 是一个具有 5 个元素的指针数组，每个元素是一个 int 型指针

18.定义析构函数时，应该注意：（ C ）
A)其名与类名完全相同 B)返回类型是 void 类型
C)无形参，也不可重载 D)函数体中必须有 delete 语句

19.在公有继承的情况下，基类的成员（私有的除外）在派生类中的访问权限（ B ）
A)受限制 B)保持不变 C)受保护 D)不受保护

20.下列不是函数重载所要求的条件是:（ D ）
A)函数名相同 B)参数个数不同
C)参数类型不同 D)函数返回值类型不同

21.在 C++中执行以下 4 条语句后输出 rad 值为：(C)
static int hot=200； int &rad=hot； hot=hot+100； cout<<rad< class Base{ public: void fun(){cout<<"Base::fun"<<<"Derived::fun"<fun()；

27.在 C++中编译器所生成的文件的扩展名为（ B ）
A) .cpp B) .obj C) .lib D) .exe

28.在保护继承的情况下，基类的成员（私有的除外）在派生类中的访问权限（ C ）
A)受限制 B)保持不变 C)受保护 D)不受保护

29.有如下一段程序： int f1(float)； int f2(char)； int f3(float)； int f4(float)； int（*pf）（float）；则以下不合法的是：（ C ） A)int（*p）（float）=&f1； B)pf=&f4； C)pf=&f2； D)pf=&f3；

30.通常拷贝构造函数的参数是：（ C ） A)对象名 B)对象的成员名 C)对象的引用名 D)对象的指针名

31.系统在调用重载函数时，不能作为确定哪个重载函数被调用的依据是：（ D ）
A)参数个数 B)参数类型 C)函数名称 D)函数的返回值类型

32.下列描述中，正确的是：（ D ）
A)所有的运算符号都可以重载。 B)基类类型的指针可以指向子类，子类类型的指针也可以指向基类。 C)可以在类的构造函数中对静态数据成员进行初始化。 D)动态联编要满足两个条件：被调用的成员函数是虚函数；用指针或引用调用虚函数。

33.关于成员函数特征的描述中，错误的是：（ A ） A)成员函数一定是内联函数 B)成员函数可以重载 C)成员函数可以设置参数的默认值 D)成员函数可以是静态的

4、C++试题

一、单项选择题(本大题共 20 小题，每小题 1 分，共 20 分)在每小题列出的四个备选项中

只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无

分。

1. 编写 C++程序一般需经过的几个步骤依次是（）

A. 编辑、调试、编译、连接

B. 编辑、编译、连接、运行

C. 编译、调试、编辑、连接

D. 编译、编辑、连接、运行

答案：B

解析：经过编辑、编译、连接和运行四个步骤。编辑是将 C++源程序输入计算机的过程，保

存文件名为 cpp。编译是使用系统提供的编译器将源程序 cpp 生成机器语言的过程，目标文件为

obj，由于没有得到系统分配的绝对地址，还不能直接运行。连接是将目标文件 obj 转换为可执行

程序的过程，结果为 exe。运行是执行 exe，在屏幕上显示结果的过程。

2. 决定 C++语言中函数的返回值类型的是（）

A. return 语句中的表达式类型

B. 调用该函数时系统随机产生的类型

C. 调用该函数时的主调用函数类型

D. 在定义该函数时所指定的数据类型

答案：D

解析：函数的返回值类型由定义函数时的指定的数据类型决定的。A 项的表达式的值要转换

成函数的定义时的返回类型。

3. 下面叙述不正确的是（）

A. 派生类一般都用公有派生

B. 对基类成员的访问必须是无二义性的

C. 赋值兼容规则也适用于多重继承的组合

D. 基类的公有成员在派生类中仍然是公有的

答案：D

解析：继承方式有三种：公有、私有和保护。多继承中，多个基类具有同名成员，在它们
的子类中访问这些成员，就产生了二义性，但进行访问时，不能存在二义性。赋值兼容规则是指
派生类对象可以当作基类对象使用，只要存在继承关系，所以单继承或多继承都适用。基类中的
公有成员采用私有继承时，在派生类中变成了私有成员，所以 D 项错误。

4. 所谓数据封装就是将一组数据和与这组数据有关操作组装在一起，形成一个实体，这实体
也就是（）

A. 类

B. 对象

C. 函数体

D. 数据块

答案：A

解析：类即数据和操作的组合体，数据是类的静态特征，操作是类具有的动作。

5. 在公有派生类的成员函数不能直接访问基类中继承来的某个成员，则该成员一定是基类中
的（）

A. 私有成员

B. 公有成员

C. 保护成员

D. 保护成员或私有成员

答案：A

解析：在派生类中基类的保护或者基类公有都可以直接访问，基类的私有成员只能是基类
的成员函数来访问。所以选择 A 项。

6. 对基类和派生类的关系描述中，错误的是（）

A. 派生类是基类的具体化

B. 基类继承了派生类的属性

C. 派生类是基类定义的延续

D. 派生类是基类的特殊化

答案：B

解析：派生类的成员一个是来自基类，一个来自本身，所以派生类是基类的扩展，

也是基
类的具体化和特殊化，派生类是对基类扩展。B 项基类不能继承派生类成员，所以错误。

7. 关于 this 指针使用说法正确的是（）

A. 保证每个对象拥有自己的数据成员，但共享处理这些数据的代码

B. 保证基类私有成员在子类中可以被访问。

C. 保证基类保护成员在子类中可以被访问。

D. 保证基类公有成员在子类中可以被访问。

答案：A

解析：this 指针是隐藏的，可以使用该指针来访问调用对象中的数据。基类的成员在派生
类中能否访问，与继承方式有关，与 this 没有关系。所以选择 A 项。

8. 所谓多态性是指 （）

A. 不同的对象调用不同名称的函数

B. 不同的对象调用相同名称的函数

C. 一个对象调用不同名称的函数

D. 一个对象调用不同名称的对象

答案：B

解析：多态性有两种静态多态性和动态多态性，静态多态性是指调用同名函数，由于参数
的不同调用不同的同名函数；动态多态性是指不同对象调用同名函数时，由于对象不同调用不同
的同名函数。 多态性肯定具有相同的函数名，所以选择 B 项。

9. 一个函数功能不太复杂，但要求被频繁调用，则应把它定义为 （）

A. 内联函数

B. 重载函数

C. 递归函数

D. 嵌套函数

答案：A

解析：内联函数特征代码少，频繁调用，执行效率高。重载函数解决统一接口的问题；递
归是子程序调用，程序调用要耗费很多空间和时间，循环/迭代都比递归有效率得多，递归只是
从形式上，逻辑比较简洁。嵌套函数即反复调用，速度较慢。所以选择 A 项。

10. 下面函数模板定义中不正确的是（）

A. A

B. B

C. C

D. D

答案：A

解析：A 项中 F 是一个返回 Q 类型的值，而 return 中用返回类型作为返回值错误。所以选择

A 项。

11. 假设 ClassY:publicX，即类 Y 是类 X 的派生类，则说明一个 Y 类的对象时和删除 Y 类对象时

，调用构造函数和析构函数的次序分别为（）

A. X,Y；Y,X

B. X,Y；X,Y

C. Y,X；X,Y

D. Y,X；Y,X

答案：A

解析：派生类构造函数必须对这三类成员进行初始化，其执行顺序：调用基类构造函数

；调用子对象的构造函数；派生类的构造函数体。析构函数在执行过程中也要对基类和成员对象

进行操作，但它的执行过程与构造函数正好相反，即对派生类新增普通成员进行清理；调用成员

对象析构函数，对派生类新增的成员对象进行清理；调用基类析构函数，对基类进行清理，所以

选择 A 项。

12. 适宜采用 inline 定义函数情况是（）

A. 函数体含有循环语句

B. 函数体含有递归语句

C. 函数代码少、频繁调用

D. 函数代码多、不常调用

答案：C

解析：内联函数具有程序代码少、频繁调用和执行效率高的特征，所以选择 C 项。

13. 假定一个类的构造函数为 A(int aa,int bb) {a=aa--;b=a*bb;},则执行 A x(4,5);

语句后
，x.a 和 x.b 的值分别为（）
A. 3 和 15
B. 5 和 4
C. 4 和 20
D. 20 和 5
答案：C
解析：a=4,因为后减，b 的值与 a、bb 相关，b＝4*5=20，而与 aa 没有任何关系。
14. 在类中说明的成员可以使用关键字的是（）
A. public
B. extern
C. cpu
D. register
答案：A
解析：extern 用于声明外部变量的。register 声明寄存器类型变量。无 cpu 类型。它们都不能声
明类成员。public 声明为公有访问权限，所以选择 A 项。
15. 下列不能作为类的成员的是（）
A. 自身类对象的指针
B. 自身类对象
C. 自身类对象的引用
D. 另一个类的对象
答案：B
解析：类的定义，如果有自身类对象，使得循环定义，B 项错误。在类中具有自身类的指针，可
以实现链表的操作，当然也可以使用对象的引用。类中可以有另一个类的对象，即成员对象。所
以选择 B 选项。
16. 使用地址作为实参传给形参，下列说法正确的是（）
A. 实参是形参的备份
B. 实参与形参无联系
C. 形参是实参的备份
D. 实参与形参是同一对象
答案：D

解析：地址作为实参，表示实参与形参代表同一个对象。如果实参是数值，形参也是普通
变量，此时形参是实参的备份。所以选择 D 项。

17. 下列程序的输出结果是（）

```
#include <iostream.h>
void main()
{int n［］［3］={10,20,30,40,50,60};
int (*p)［3］;
p=n;
cout<<p［0］［0］<<"，"<<*(p［0］+1)<<"，"<<(*p)［2］<<endl;}
```

A. 10，30，50

B. 10，20，30

C. 20，40，60

D. 10，30，60

答案：B

解析：如果数组元素都是相同类型的指针，则称这个数组为指针数组。指针数组一般用于处理二
维数组。声明的格式为：<数据类型><(*变量名)><［元素个数］>。
p 表示指向数组 n 的行指针。如果将指针的初始化(*p)［3］=b;地址的等价形式：
p+i p［i］*(p+i)都表示 b 数组第 i+1 行的第 1 个元素的首地址。
*(p+i)+jp［i］+j &p［i］［j］都表示 b 数组第 i+1 行、第 j+1 列元素的地址。
值的等价形式：
((p+i)+j) *(p［i］+j) p［i］［j］都表示 b 数组第 i+1、第 j+1 列元素的值。
所以题目分别访问 p［0］［0］，p［0］［1］，p［0］［2］。

18. 在 C++中，使用流进行输入输出，其中用于屏幕输入（）

A. cin

B. cerr

C. cout

D. clog

答案：A

解析：(1)标准输入流 cin：istream 类的对象。(2)标准输出流 cout：ostream 类的对象。
(3)非缓冲型标准出错流 cerr：ostream 类的对象。(4)缓冲型标准出错流 clog：

ostream 类的对象

19. 假定 AA 为一个类，a()为该类公有的函数成员，x 为该类的一个对象，则访问 x 对象中函数

成员 a()的格式为（）

A. x.a

B. x.a()

C. x->a

D. （*x）.a()

答案：B

解析：对象访问成员的方式为：对象名.成员。指针可以有两种：(*对象指针).成员或者对

象指针->成员。A 选项是访问数据成员，B 项是访问成员函数。

20. 关于对象概念的描述中，说法错误的是（）

A. 对象就是 C 语言中的结构变量

B. 对象代表着正在创建的系统中的一个实体

C. 对象是类的一个变量

D. 对象之间的信息传递是通过消息进行的

答案：A

解析：A 对象在 C＋＋中才有，包括数据和操作两项，而 C 中的变量只有数据，没有操作。所

以 A 项错误。

二、填空题(本大题共 20 小题，每小题 1 分，共 20 分)请在每小题的空格中填上正确答案

。错填、不填均无分。

1. C++的流库预定义了 4 个流，它们是 cin、cout、clog 和___。

答案：(P193)cerr

［解析］cin、cout、clog 和 cerr 分别用于标准输入、输出、标准错误流（缓冲）和标准错误流

（非缓冲）。

2. 每个对象都是所属类的一个___。

答案：实例

［解析］类是对象的抽象，对象是类的一个实例。

3. 在已经定义了整型指针 ip 后，为了得到一个包括 10 个整数的数组并由 ip 所指向，应使用语

句___。

答案：int *ip=new int［10］;

［解析］new 用来动态开辟空间。常用来产生动态数组及对象构造函数。

4. 函数模板中紧随 template 之后尖括号内的类型参数都要冠以保留字___。

答案：class

［解析］类模板的使用。template <class T>,也可以引入多参数的如：template <class T1，class T2,...，class Tn>

5. 定义类的动态对象数组时，系统只能够自动调用该类的___构造函数对其进行初始化。

答案：无参

［解析］使用 new 创建对象数组，调用无参构造函数。

6. 表达式 cout<<endl 还可表示为___。

答案：‘\n’

［解析］endl 与字符常量‘\n’等价。

7. 在 C++中，访问一个指针所指向的对象的成员所用的指向运算符是___。

答案：->

［解析］指针使用成员有两种方法：“->”指向运算符和“.”成员访问运算符。

8. 假如一个类的名称为 MyClass，使用这个类的一个对象初始化该类的另一个对象时，可以调

用___构造函数来完成此功能。

答案：复制或拷贝

复制或拷贝构造函数就是用对象初始化新的对象。

9. 对赋值运算符进行重载时，应声明为___函数。

答案：(P183)类成员

［解析］运算符重载的方法有友元或者成员函数两种途径，但是赋值运算符只能使用成员函数的

方法来实现。

10. 如果要把 A 类成员函数 f（）且返回值为 void 声明为类 B 的友元函数，则应在类 B 的定义中加

入的语句___。

答案：(P109)friend void A::f();

［解析］成员函数作为另一个类的友元函数，格式为：friend 返回类型 类名::函数(形参)。

11. 下列程序段的输出结果是___。

for(i=0,j=10,k=0;i<=j;i++,j-=3,k=i+j);cout<<k;

答案：4

［解析］for 循环结构,三个表达式的作用，初始化、循环判断条件和循环变量变化。循环执行了

三次，k 的作用是计算 i、j 的和。

12. String 类的___方法返回查找到的字符串在主串的位置。

答案：(P40)find

［解析］string 类对象方法的 find，查不到字符串，则返回-1。

13. int n=0;

while（n=1）n++;

while 循环执行次数是___。

答案：无限次

［解析］＝是赋值运算符，不是关系运算符，且不等 0，所以死循环。

14. 控制格式输入输出的操作中，函数___是用来设置填充字符。要求给出函数名和参数类型

答案：setfill(char)

［解析］格式控制方法的使用，如 setw，setfill 等等。

15. C++语言支持的两种多态性分别是编译时的多态性和___的多态性。

编译时多态:程序运行前发生的事件——函数重载、运算符重载——静态绑定

运行时多态:程序运行时发生的事件——虚函数机制动态绑定

答案：运行时

［解析］多态性包括静态的（编译时）多态性和动态的（运行时）多态性。

16. 设函数 sum 是由函数模板实现的，并且 sum(3,6)和 sum(4.6,8)都是正确的函数调用，则函

数模板具有___个类型参数。

答案：2

17. 执行下列代码

string str("HelloC++");

cout<<str.substr(5，3);

程序的输出结果是___。

答案：C++

［解析］substr 取子字符串，第 1 个参数表示要截取子串在字符串中的位置，第 2 个表示取多少个

字符。

18. 在面向对象的程序设计中，将一组对象的共同特性抽象出来形成___。

答案：类

［解析］类是相似特征的对象的抽象，对象是类的一个实例。

19. 定义类动态对象数组时，元素只能靠自动调用该类的___来进行初始化。

答案：无参构造函数

［解析］使用 new 创建动态对象数组，不能有参数，所以只能调用无参的构造函数，初始化对象

20. 已知有 20 个元素 int 类型向量 V1，若用 V1 初始化为 V2 向量，语句是___。

答案：ector <int>V2(V1);

［解析］采用向量初始化另一个向量的形式：vector <type> name1(name);

三、程序分析题(本大题共 4 小题，每小题 5 分，共 20 分)

1. 给出下面程序输出结果。

```
#include<iostream.h>
class a
{public:
    virtual void print()
{cout<< "a prog..."<< endl;};
};
    class b:public a
{};
    class c:public b
{public:
    void print(){cout<<"c prog..."<<endl;}
};
    void show(a *p)
{(*p).print();
}
    void main()
{   a a;
    b b;
    c c;
    show(&a);
    show(&b);
    show(&c);
}
```

答案：a prog...

a prog...

c prog...

［解析］考查多态性的。a 类对象调用本身的虚函数，b 类因为没有覆写 print，所以仍然调用基类的虚函数。而 c 类重新定义 print 虚函数，所以调用 c 类的 print。

2. 给出下面程序输出结果。

```
#include <math.h>
#include <iostream.h>
#include <iomanip.h>
bool fun(long n);
void main()
{long a=10,b=30,l=0;
if(a%2==0) a++;
for(long m=a;m<=b;m+=2)
if(fun(m))
{if(l++%10==0)
cout <<endl;
cout <<setw(5) <<m;
}
}
bool fun(long n)
{int sqrtm=(int)sqrt(n);
for(int i=2;i<=sqrtm;i++)
if(n%i==0)
return false;
return true;
}
```

答案：11 13 17 19 23 29

［解析］循环体用来判断 n 是否是质数的函数,在 main 函数判断 10～30 之间质数。

3. 给出下面程序输出结果。

```
#include <iostream.h>
class Test
{int x,y;
```

```
public:
Test(int i,int j=0)
{x=i;y=j;}
int get(int i,int j)
{return i+j;}
};
void main()
{Test t1(2),t2(4,6);
int (Test::*p)(int,int=10);
p=Test::get;
cout<<(t1.*p)(5)<<endl;
Test *p1=&t2;
cout<<(p1->*p)(7,20)<<endl;
}
```

答案：15　27

［解析］指向类成员函数的指针的使用，*p 指向 Test 类中有两个参数的函数的一个指针。

P＝Test::get.这样 p 就和 get 发生了联系。(t1.*p)(5)等价于调用一个参数的 get 函数。

4.

```
#include <iostream.h>
#include <string.h>
#include <iomanip.h>
class student
{char name［8］;
int deg;
char level［7］;
friend class process; //  说明友元类
public:
student(char na［］,int d)
{ strcpy(name,na);
deg=d;
}
};
class process
```

```
{ public:
void trans(student &s)
{int i=s.deg/10;
switch(i)
{case 9:
strcpy(s.level, "优");break;
case 8:
strcpy(s.level,"良");break;
case 7:
strcpy(s.level,"中");break;
case 6:
strcpy(s.level,"及格");break;
default:
strcpy(s.level,"不及格");
}
}
void show(student &s)
{cout<<setw(10)<<s.name<<setw(4)<<s.deg<<setw(8)<<s.level<<endl;}
};
void main()
{ student st［］={student("张三",78),student("李四",92),student("王五
",62),student("孙六",88)};
process p;
cout<<"结  果:"<<"姓名"<<setw(6)<<"成绩"<<setw(8)<<"等级"<<endl;
for(int i=0;i<4;i++)
{ p.trans(st［i］);
p.show(st［i］);}
}
```

答案：结果:姓名成绩等级
张三 78 中
李四 92 优
王五 62 及格
孙六 88 良

5、C++程序设计模拟试卷

一、单项选择题(本大题共 20 小题，每小题 1 分，共 20 分)在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

1. 静态成员函数没有（）

A. 返回值

B. this 指针

C. 指针参数

D. 返回类型

答案：B

解析：静态成员函数是普通的函数前加入 static,它具有函数的所有的特征：返回类型、

形参，所以使用(P107)静态成员函数，指针可以作为形参，也具有返回值。静态成员是类具有的

属性，不是对象的特征，而 this 表示的是隐藏的对象的指针，因此静态成员函数没有 this 指针

。静态成员函数当在类外定义时，要注意不能使用 static 关键字作为前缀。由于静态成员函数在

类中只有一个拷贝（副本），因此它访问对象的成员时要受到一些限制：静态成员函数可以直接

访问类中说明的静态成员，但不能直接访问类中说明的非静态成员；若要访问非静态成员时，必

须通过参数传递的方式得到相应的对象，再通过对象来访问。

2. 假定 AB 为一个类，则执行“AB a(2), b［3］,*p［4］;”语句时调用该类构造函数的次数

为（）

A. 3

B. 4

C. 5

D. 9

答案：B

解析：a(2)调用 1 次带参数的构造函数，b［3］调用 3 次无参数的构造函数，指针没有给它

分配空间，没有调用构造函数。所以共调用构造函数的次数为 4。

3. 有关多态性说法不正确的是（）

A. C++语言的多态性分为编译时的多态性和运行时的多态性

B. 编译时的多态性可通过函数重载实现

C. 运行时的多态性可通过模板和虚函数实现

D. 实现运行时多态性的机制称为动态多态性

答案：C

解析：多态性分为静态的和动态的。静态通过函数的重载来实现，动态是通过基类指针或

基类引用和虚函数来实现的。所以错误的是 C 项。

4. 假定一个类的构造函数为“A(int i=4, int j=0) {a=i;b=j;}”，则执行“A x (1);”语

句后，x.a 和 x.b 的值分别为（）

A. 1 和 0

B. 1 和 4

C. 4 和 0

D. 4 和 1

答案：A

解析：带默认的构造函数，对应实参没有值时就采用形参值。调用构造函数时，i=1,不采

用默认值，而只有一个参数，j 采用默认值 0 即 j=0,因此 a=1,b=0,选择 A 项。

5. 类 MyA 的拷贝初始化构造函数是 （）

A. MyA()

B. MyA(MyA*)

C. MyA(MyA&)

D. MyA(MyA)

答案：C

解析：复制即拷贝构造函数使用对象的引用作形参，防止临时产生一个对象，A 无参构造函

数，B 是指针作为形参，D 项是对象，所以选择 C 项。

6. 在 C++中，函数原型不能标识（）

A. 函数的返回类型

B. 函数参数的个数

C. 函数参数类型

D. 函数的功能

答案：D

解析：函数的声明，说明函数的参数、返回类型以及函数名，函数体即实现部分决定功能。所以

函数的原型不能决定函数的功能。

7. 友元关系不能（）

A. 提高程序的运行效率

B. 是类与类的关系

C. 是一个类的成员函数与另一个类的关系

D. 继承

答案：D

解析：友元可以是函数与类的关系即友元函数，也可以类与类的关系即友元类，但友元不

能继承，是单向性，且不具有传递性。友元可以访问类中所有成员，提高了访问的方便性。因此

选择 D 项。

8. 实现两个相同类型数加法的函数模板的声明是（）

A. add(T x,T y)

B. T add(x,y)

C. T add(T x,y)

D. T add(T x,T y)

答案：D

解析：实现两个相同类型数加法结果应该和操作数具有相同类型。进行加法运算后结果也

是和参数具有相同类型，需要返回值。A 无返回值时要用 void,B 形参无类型，C 形参 y 没有类型

，所以选择 D 项。

9. 在 int a=3,int *p=&a；中，*p 的值是（）

A. 变量 a 的地址值

B. 无意义

C. 变量 p 的地址值

D. 3

答案：D

解析：*p 代表引用 a 变量的值，p 代表 a 的地址值。所以选择 D 项。

10. 下列不是描述类的成员函数的是（）

A. 构造函数

B. 析构函数

C. 友元函数

D. 拷贝构造函数

答案：C

解析：构造函数、析构函数、拷贝构造函数都是特殊的成员函数，友元则不是成员函数。

所以选择 C 项。

11. 如果从原有类定义新类可以实现的是（）

A. 信息隐藏

B. 数据封装

C. 继承机制

D. 数据抽象

答案：C

解析：继承指在原有类的基础上产生新类。数据封装即数据和操作组合在一起，形成类。

信息的隐藏，通过访问权限来实现。数据抽象，将事物的特征抽象为数据成员或服务。因此选择

C 项。

12. 下面有关类说法不正确的是（）

A. 一个类可以有多个构造函数

B. 一个类只有一个析构函数

C. 析构函数需要指定参数

D. 在一个类中可以说明具有类类型的数据成员

答案：C

解析：构造函数可以有参数、可以重载、因此可以有多个，A 项正确。析构函数只有一个不

能重载、不能继承，没有返回值，B 项正确，C 项错误。

13. 在函数定义中的形参属于（）

A. 全局变量

B. 局部变量

C. 静态变量

D. 寄存器变量

答案：B

解析：形参或函数中定义的变量都是局部变量。在函数外定义的变量是全局变量。形参只能用局

部变量，频繁使用的变量可以声明为寄存器变量，形参不能使用静态变量或寄存器变量。

14. 下列有关重载函数的说法中正确的是（）

A. 重载函数必须具有不同的返回值类型

B. 重载函数参数个数必须相同

C. 重载函数必须有不同的形参列表

D. 重载函数名可以不同

答案：C

解析：函数的重载必须函数名相同而形参类型或个数不同，与返回值无关。

15. this 指针存在的目的是（）

A. 保证基类私有成员在子类中可以被访问

B. 保证基类保护成员在子类中可以被访问

C. 保证每个对象拥有自己的数据成员，但共享处理这些数据成员的代码

D. 保证基类公有成员在子类中可以被访问

答案：C

解析：C++要求函数在被调用之前，应当让编译器知道该函数的原型，以便编译器利用函数

原型提供的信息去检查调用的合法性，强制参数转换成为适当类型，保证参数的正确传递。对于

标准库函数，其声明在头文件中，可以用#include 宏命令包含这些原型文件；对于用户自定义函

数，先定义、后调用的函数可以不用声明，但后定义、先调用的函数必须声明。一般为增加程序

的可理解性，常将主函数放在程序开头，这样需要在主函数前对其所调用的函数一一进行声明

，以消除函数所在位置的影响。所以选择 C 项。

16. 关于 new 运算符的下列描述中，错误的是（）

A. 它可以用来动态创建对象和对象数组

B. 使用它创建的对象或对象数组可以使用运算符 delete 删除

C. 使用它创建对象时要调用构造函数

D. 使用它创建对象数组时必须指定初始值

答案：D

解析：new 创建的对象数组不能指定初始值，所以调用无参的构造函数，选择 D 项。

17. 已知：p 是一个指向类 A 数据成员 m 的指针，A1 是类 A 的一个对象。如果要给 m 赋值为 5，正确

的是（）

A. A1.p=5;

B. A1->p=5;

C. A1.*p=5;

D. *A1.p=5;

答案：C

解析：A 中 p 是指针即地址，错误；B 选项中 A1 不是指针不能使用指向运算符->,错误

；“*”比“.”级别要高，所以 D 选项*A1.p=5 相当于(*A1).p=5;错误。另外涉及到指向成员函数

时注意以下几点：

指向成员函数的指针必须于其赋值的函数类型匹配的三个方面：(1)参数类型和个数；(2)返回

类型；(3)它所属的类类型。

成员函数指针的声明：指向 short 型的 Screen 类的成员的指针定义如下：

short Screen::* ps_Screen;

ps_Screen 可以用 _height 的地址初始化如下：short Screen::*ps_Screen=&Screen::_height;

类成员的指针必须总是通过特定的对象或指向改类型的对象的指针来访问。是通过使用两个指

向成员操作符的指针(针对类对象和引用的.*，以及针对指向类对象的指针的->*)。

18. 以下基类中的成员函数表示纯虚函数的是（）

A. virtual void tt()=0

B. void tt(int)=0

C. virtual void tt(int)

D. virtual void tt(int){}

答案：A

解析：当在基类中不能为虚函数给出一个有意义的实现时，可以将其声明为纯虚函数，实
现由派生类完成。格式：virtual<函数返回类型说明符><函数名>(<参数表>)=0;。

19. C++类体系中，不能被派生类继承的有（）

A. 常成员函数

B. 构造函数

C. 虚函数

D. 静态成员函数

答案：B

解析：构造函数不能被继承。缺省构造函数，拷贝构造函数，拷贝赋值函数，以及析构函数这四种成员函数被称作特殊的成员函数。

20. 静态成员函数不能说明为（）

A. 整型函数

B. 浮点函数

C. 虚函数

D. 字符型函数

答案：C

解析：使用关键字 static 声明的成员函数就是静态成员函数，静态成员函数也属于整个类
而不属于类中的某个对象，它是该类的所有对象共享的成员函数。
静态成员函数可以在类体内定义，也可以在类外定义。当在类外定义时，要注意不能使用
static 关键字作为前缀。
由于静态成员函数在类中只有一个拷贝（副本），因此它访问对象的成员时要受到一些限制：静
态成员函数可以直接访问类中说明的静态成员，但不能直接访问类中说明的非静态成员；若要访
问非静态成员时，必须通过参数传递的方式得到相应的对象，再通过对象来访问。虚函数是非静
态的、非内联的成员函数。静态成员函数不能被说明为虚函数。

二、填空题(本大题共 20 小题，每小题 1 分，共 20 分)请在每小题的空格中填上正确答案
。错填、不填均无分。

1. 假设 int a=1,b=2;则表达式(++a/b)*b--的值为___。

答案：2

［解析］前缀＋＋或－－表示先使变量值变化，再使用，这和后缀恰恰相反。但是编译

cout<<(++a/b)*b--时，先++a/b 值为 1，后 1*b--，先取 b＝2，结果为 2，再让 b＝1。

2. 抽象类中至少要有一个___函数。

答案：纯虚

［解析］至少有一个纯虚函数的类就称为抽象类，即不能实例化。

3. 一个抽象类的派生类可以实例化的必要条件是实现了所有的___。

答案：纯虚函数的定义

［解析］抽象类只因有纯虚函数，所以不能被实例化，所以派生类要实例化必须对纯虚函数进行

定义。

4. 下面程序的输出结果为___。

```
#include <iostream.h>
void main()
{int num=2,i=6;
do
{i--;
num++;
}while(--i);
cout<<num<<endl;
}
```


答案：5

［解析］do－while 循环，前缀先使 i 减少 1 后判断是否为零，不为零时再次执行循环，为零退出

循环。循环值执行 3 次就退出，所以结果为 5。

5. 静态成员函数、友元函数、构造函数和析构函数中，不属于成员函数的是___。

答案：友元函数

［解析］友元函数不是类成员，但可以访问类成员。类的封装性保证了数据的安全，但引入友元

，虽然访问类是方便了，但确实破坏类访问的安全性。

6. 在用 C＋＋进行程序设计时，最好用___代替 malloc。

答案：new

［解析］new 与 delete 是 C++语言特有的运算符，用于动态分配和释放内存。new

用于为各种数据
类型分配内存，并把分配到的内存首地址赋给相应的指针。new 的功能类似于 malloc（）函数。
使用 new 的格式为：
<指针变量>new<数据类型>;
其中，<数据类型>可以是基本数据类型，也可以是由基本类型派生出来的类型；<指针变量>取得
分配到的内存首地址。new 有 3 种使用形式。
（1）给单个对象申请分配内存
int *ip;ip=new int;//ip 指向 1 个未初始化的 int 型对象
（2）给单个对象申请分配内存的同时初始化该对象
int *ip;ip=new int(68);//ip 指向 1 个表示为 68 的 int 型对象
（3）同时给多个对象申请分配内存
int *ip;ip=new int［5］;//ip 指向 5 个未初始化的 int 型对象的首地址
for(int i=0;i<5;i++)ip［i］=5*i+1;//给 ip 指向的 5 个对象赋值
用 new 申请分配内存时，不一定能申请成功。若申请失败，则返回 NULL，即空指针。因此，在程
序中可以通过判断 new 的返回值是否为 0 来获知系统中是否有足够的空间供用户使用。
7. 由 const 修饰的对象称为___。
答案：常对象
［解析］使用 const 关键字说明的成员函数称为常成员函数，使用 const 关键字说明的对象称为常
对象。
常成员函数的说明格式如下：<返回类型说明符><成员函数名>(<参数表>)const;
常成员函数不更新对象的数据成员，也不能调用该类中没有用 const 修饰的成员函数。常对象
只能调用它的常成员函数，而不能调用其他成员函数。const 关键字可以用于参与重载函数的区
分。
8. 在 C++程序设计中，建立继承关系倒挂的树应使用___继承。
答案：单
［解析］一个基类可以派生多个子类，一个子类可以再派生出多个子类，这样就形成了一个倒立

的树。

9. 基类的公有成员在派生类中的访问权限由___决定。

答案：访问控制方式或继承方式

10. 不同对象可以调用相同名称的函数，但执行完全不同行为的现象称为___。

答案：多态性

［解析］多态性的概念。虚函数是实现多态的基础，运行过程中的多态需要同时满足 3 个条件

：(1)类之间应满足子类型关系。(2)必须要有声明的虚函数。(3)调用虚函数操作的是指向对象

的指针或者对象引用；或者是由成员函数调用虚函数（如果是在构造函数或析构函数中调用虚函

数，则采用静态联编）。

11. this 指针始终指向调用成员函数的___。

答案：对象

this 指针是隐藏的指针，它指向调用函数的对象。

12. 预处理命令以___符号开头。

答案：operater

［解析］文件包含、预处理和编译都是以＃开头。

13. 类模板用来表达具有___的模板类对象集。

答案：相同处理方法

［解析］模板特点是不同的数据具有相同的处理方法的抽象。

14. C++程序的源文件扩展名为___。

答案：cpp

［解析］源程序*.cpp,目标文件为*.obj,可执行程序*.exe。

15. 在#include 命令中所包含的头文件，可以是系统定义的头文件，也可以是___的头文件。

答案：自定义

［解析］＃include 装入文件有两种方式<>和“ ”，一是系统的，一是自定义文件。

16. vector 类中向向量尾部插入一个对象的方法是___。

答案：push_back

17. C++语言中如果调用函数时，需要改变实参或者返回多个值，应该采取___方式。

答案：传地址或引用

［解析］传地址即指针，在函数中通过指针修改它指向的变量的值时，实参也就

变化了。使用引
用，直接修改变量的别名即引用的值，该变量也就随着变化。

18. 语句序列

ifstream infile;

infile.open(“data.dat”);

的功能可用一个语句实现，这个语句是___。

答案：ifstream infile(“data.dat”);

［解析］void ifstream::open(const char *fname,int mode=ios::in,int
access=filebuf::openprot);

ifstream::ifstream(const char *fname,int mode=ios::in,int access=filebuf::openprot);

其中，第一个参数是用来传递文件名的；第二个参数 mode 的值决定文件将如何被打开；第三个参
数 access 的值决定文件的访问方式，一般取缺省值 filebuf::openprot，表示是普通文件。

mode 的取值如下：(1)ios::in：打开一个文件进行读操作，而且该文件必须已经存在
；(2)ios::nocreate：不建立新的文件。当文件不存在时，导致 open()失败
；(3)ios::noreplace：不修改原来已经存在的文件。若文件已经存在，导致 open()失败
；(4)ios::binary：文件以二进制方式打开，缺省时为文本文件。

19. 如果要把类 B 的成员函数 void fun()说明为类 A 的友元函数，则应在类 A 中加入语句___。

答案：(P111)friend void B::fun();

［解析］声明成员函数作为另外一个类的友元函数时，使用类作用域运算符::。

20. 在编译指令中，宏定义使用___指令。

答案：#define

［解析］静态成员是所有对象共享的特征，也就是类的特征。

6、C++程序设计模拟试卷

一、单项选择题(本大题共 20 小题，每小题 1 分，共 20 分)在每小题列出的四个备选项中只有一个是符合题目要求的，请将其代码填写在题后的括号内。错选、多选或未选均无分。

1. 设有定义 int i;double j＝5;，则 10+i+j 值的数据类型是（）

A. int

B. double

C. float

D. 不确定

答案：B

解析：考察数据的转换，j 是 double 类型，运算只能作同类型的运算，所以要转换，而 int 能自动

转换为 double 类型，所以结果是 double 类型。

2. 要禁止修改指针 p 本身，又要禁止修改 p 所指向的数据，这样的指针应定义为（）

A. const char *p=“ABCD”;

B. char *const p=“ABCD”;

C. char const *p=“ABCD”;

D. const char * const p=“ABCD”;

答案：D

解析：const char *p 说明禁止通过 p 修改所指向的数据。char * const p 则说明不能修改

指针 p 的地址。因此 const char * const p=“ABCD”；它禁止修改指针 p 本身，又禁止修改 p 所指

向的数据。

3. 类的构造函数被自动调用执行的情况是在定义该类的（）

A. 成员函数时

B. 数据成员时

C. 对象时

D. 友元函数时

答案：C

解析：建立对象时，自动构造函数的初始化对象，是系统自动调用的。而成员函数、友元

函数，需要用户直接调用，因此选择 C 项。

4. 已知类 A 是类 B 的友元，类 B 是类 C 的友元，则（）

A. 类 A 一定是类 C 的友元

B. 类 C 一定是类 A 的友元

C. 类 C 的成员函数可以访问类 B 的对象的任何成员

D. 类 A 的成员函数可以访问类 B 的对象的任何成员

答案：C

解析：友元说明方法如下：

friend?<类名>;//友元类类名

使用友元可以访问所有成员：

(1)友元关系不能被继承。

(2)友元关系是单向的，不具有交换性。所以，B 项和 D 项错误。

(3)友元关系不具有传递性。所以，A 项错误。

5. 假定一个类的构造函数为“A(int i=4, int j=0) {a=i;b=j;}”，则执行“A x (1);”语

句后，x.a 和 x.b 的值分别为（）

A. 1 和 0

B. 1 和 4

C. 4 和 0

D. 4 和 1

答案：A

解析：带默认的构造函数，对应实参没有值时就采用形参值。调用构造函数时，i=1,不采

用默认值，而只有一个参数，j 采用默认值 0 即 j=0,因此 a=1,b=0,选择 A 项。

6. 关于 this 指针使用说法正确的是（）

A. 保证每个对象拥有自己的数据成员，但共享处理这些数据的代码

B. 保证基类私有成员在子类中可以被访问。

C. 保证基类保护成员在子类中可以被访问。

D. 保证基类公有成员在子类中可以被访问。

答案：A

解析：this 指针是隐藏的，可以使用该指针来访问调用对象中的数据。基类的成员在派生

类中能否访问，与继承方式有关，与 this 没有关系。所以选择 A 项。

7. 所谓多态性是指 （）

A. 不同的对象调用不同名称的函数

B. 不同的对象调用相同名称的函数

C. 一个对象调用不同名称的函数

D. 一个对象调用不同名称的对象

答案：B

解析：多态性有两种静态多态性和动态多态性，静态多态性是指调用同名函数，由于参数

的不同调用不同的同名函数；动态多态性是指不同对象调用同名函数时，由于对象不同调用不同

的同名函数。 多态性肯定具有相同的函数名，所以选择 B 项。

8. 友元关系不能（）

A. 提高程序的运行效率

B. 是类与类的关系

C. 是一个类的成员函数与另一个类的关系

D. 继承

答案：D

解析：友元可以是函数与类的关系即友元函数，也可以类与类的关系即友元类，但友元不

能继承，是单向性，且不具有传递性。友元可以访问类中所有成员，提高了访问的方便性。因此

选择 D 项。

9. 语句 ofstream f(“TEMP.DAT”,ios::app | ios::binary)?的功能是建立流对象 f，试图打

开文件 TEMP.DAT 并与之连接，并且（）

A. 若文件存在，将文件写指针定位于文件尾；若文件不存在，建立一个新文件

B. 若文件存在，将其置为空文件；若文件不存在，打开失败

C. 若文件存在，将文件写指针定位于文件首；若文件不存在，建立一个新文件

D. 若文件存在，打开失败；若文件不存在，建立一个新文件

答案：A

解析：ios::binary，采用二进制形式，ios::app 定位到文件尾部。

10. 构造函数不具备的特征是（）

A. 构造函数的函数名与类名相同

B. 构造函数可以重载

C. 构造函数可以设置默认参数

D. 构造函数必须指定类型说明

答案：D

解析：构造函数无返回类型不能继承但可以重载，所以选择 D 项。

11. 在公有继承的情况下，基类的公有或保护成员在派生类中的访问权限（）

A. 受限制

B. 保持不变

C. 受保护

D. 不受保护

答案：B

解析：继承方式的不同派生类成员的权限也不同，采用公有继承，除了私有无法访问外

，公有、保护在派生类中保持不变，所以选择 B 项。

12. 假定一个类的构造函数为 A(int aa,int bb) {a=aa--;b=a*bb;},则执行 A x(4,5);语句后

，x.a 和 x.b 的值分别为（）

A. 3 和 15

B. 5 和 4

C. 4 和 20

D. 20 和 5

答案：C

解析：a=4,因为后减，b 的值与 a、bb 相关，b＝4*5=20，而与 aa 没有任何关系。

13. C++对 C 语言做了很多改进，即从面向过程变成为面向对象的主要原因是（）

A. 增加了一些新的运算符

B. 允许函数重载，并允许设置缺省参数

C. 规定函数说明符必须用原型

D. 引进了类和对象的概念

答案：D

解析：C++是一面向对象的语言，面向对象的特征，抽象、多态、继承和封装。

14. 在类中说明的成员可以使用关键字的是（）

A. public

B. extern

C. cpu

D. register

答案：A

解析：extern 用于声明外部变量的。register 声明寄存器类型变量。无 cpu 类型。

它们都不能声
明类成员。public 声明为公有访问权限，所以选择 A 项。

15. C++语言中所有在函数中定义的变量，连同形式参数，都属于（）

A. 全局变量

B. 局部变量

C. 静态变量

D. 函数

答案：B

解析：变量存储类可分为两类：全局变量和局部变量。

（1）全局变量：在函数外部定义的变量称为全局变量，其作用域为：从定义变量的位置开始到
源程序结束。使用全局变量降低了程序的可理解性，软件工程学提倡尽量避免使用全局变量。

（2）局部变量：在函数内部定义的变量称为局部变量，其作用域为：从定义变量的位置开始到
函数结束。局部变量包含自动变量（auto）静态变量（static）以及函数参数。形参不能是静态
的。所以选择 B 项。

16. 在私有继承的情况下，基类成员在派生类中的访问权限（）

A. 受限制

B. 保持不变

C. 受保护

D. 不受保护

答案：A

解析：私有继承下，基类中的公有或保护成员在派生类中也是私有的，所以选择 A 选项。

17. 使用地址作为实参传给形参，下列说法正确的是（）

A. 实参是形参的备份

B. 实参与形参无联系

C. 形参是实参的备份

D. 实参与形参是同一对象

答案：D

解析：地址作为实参，表示实参与形参代表同一个对象。如果实参是数值，形参也是普通

变量，此时形参是实参的备份。所以选择 D 项。

18. C++的继承性允许派生类继承基类的（）

A. 部分特性，并允许增加新的特性或重定义基类的特性

B. 部分特性，但不允许增加新的特性或重定义基类的特性

C. 所有特性，并允许增加新的特性或重定义基类的特性

D. 所有特性，但不允许增加新的特性或重定义基类的特性

答案：A

解析：派生类有两类成员：一是基类，二是自身类。派生类中的成员不能访问基类中的私

有成员，可以访问基类中的公有成员和保护成员。

19. 对于 int *pa［5］;的描述，正确的是（）

A. pa 是一个指向数组的指针，所指向的数组是 5 个 int 型元素

B. pa 是一个指向某个数组中第 5 个元素的指针，该元素是 int 型变量

C. pa［5］表示某个数组的第 5 个元素的值

D. pa 是一个具有 5 个元素的指针数组，每个元素是一个 int 型指针

答案：D

解析：指针数组：数组元素都是相同类型的指针，相同类型的指针是说指针所指向的对象

类型是相同的。例如，语句 int *pa［5］;定义了一个指针数组。在指针数组的定义中有两个运

算符：*和［］，运算符［］的优先级高于*，所以*pa［5］等价于*(pa［5］)，pa［5］表示一

个数组，而*表示后面的对象为指针变量，合在一起*pa［5］表示一个指针数组。该数组包含 5 个

元素，每个元素都是指向 int 型的指针。所以选择 D 选项。

20. 以下基类中的成员函数表示纯虚函数的是（）

A. virtual void tt()=0

B. void tt(int)=0

C. virtual void tt(int)

D. virtual void tt(int){}

答案：A

解析：当在基类中不能为虚函数给出一个有意义的实现时，可以将其声明为纯虚函数，实

现由派生类完成。格式：virtual<函数返回类型说明符><函数名>(<参数表>)=0;。

二、填空题（本大题共 20 小题，每小题 1 分，共 20 分）请在每小题的空格中填上正确答
案。错填、不填均无分。

1. 单目运算符作为类成员函数重载时，形参个数为___个。

答案：(P189)0

［解析］单目运算符使用成员函数重载可以不用形参，双目运算符使用一个参数。

2. 抽象类中至少要有一个___函数。

答案：(P173)纯虚

［解析］至少有一个纯虚函数的类就称为抽象类，即不能实例化。

3. 设类 A 有成员函数 void f(void)；若要定义一个指向类成员函数的指针变量 pf 来指向 f，该
指针变量的声明语句是：___。

答案：(P117)void (A::*pf)(void)=&A::f;

［解析］void(A::*pf)(void)=&A::f;指向成员函数的指针，它相当于两条语句：void(A::*pf)(void);和 pf=&A::f;。

4. 执行下列程序

```
double a=3.,b=3.14;
cout<<setprecision(5)<<a<<", "<<setprecision(5)<<b<<endl;
```

程序的输出结果是___。

答案：3.1416，3.14

［解析］题目设置精度即有效数字都是 5，a 四舍五入是 3.1416，b 是 3.14。

5. vector 类中用于删除向量中的所有对象的方法是___。

答案：clear()

［解析］向量的使用。返回向量中对象的方法有：front()back（）operator［］，在向量中删
除对象的方法 pop_back erase clear。

6. 重载的运算符保持其原有的___、优先级和结合性不变。

答案：操作数

［解析］运算符重载时要遵循以下规则：

(1)除了类属关系运算符“.”、成员指针运算符“.*”、作用域运算符“::”、sizeof 运算符
和三目运算符“?:”以外，C++中的所有运算符都可以重载。

(2)重载运算符限制在 C++语言中已有的运算符范围内的允许重载的运算符之中，不能创建新的

运算符。

(3)重载之后的运算符不能改变运算符的优先级和结合性，也不能改变运算符操作数的个数及

语法结构。

7. 编译时的多态性通过___函数实现。

答案：重载

［解析］编译多态性，实现的方法主要通过函数的重载或运算符的重载。

8. 基类的公有成员在派生类中的访问权限由___决定。

答案：访问控制方式或继承方式

9. 假设类 X 的对象 x 是类 Y 的成员对象，则“Y Obj”语句执行时，先调用类___的构造函数。

答案：X

［解析］派生类中的构造函数的执行顺序，先基类后派生类。

10. 下列程序段的输出结果是___。

```
cout.setf(ios::showpos);
cout<<509.3<<endl;
```

答案：(P193)+509.3

［解析］输入、输出格式 ios::showpos 用于输出数据的符号位。

11. 下列程序段的输出结果是___。

```
for(i=0,j=10,k=0;i<=j;i++,j-=3,k=i+j);cout<<k;
```

答案：4

［解析］for 循环结构,三个表达式的作用，初始化、循环判断条件和循环变量变化。循环执行了

三次，k 的作用是计算 i、j 的和。

12. C＋＋中 ostream 的直接基类___。

答案：ios

［解析］istream 和 ostream 的直接基类是 ios。

13. int n=0;

while（n=1）n++;

while 循环执行次数是___。

答案：无限次

［解析］＝是赋值运算符，不是关系运算符，且不等 0，所以死循环。

14. C++中有两种继承：单继承和___。

答案：多继承

［解析］单继承和多继承，多继承即有多个基类。

15. 在 C++中，利用向量类模板定义一个具有 10 个 int 的向量 A，其元素均被置为 1，实现此操作
的语句是___。

答案：vector<int>A(10,1)

［解析］定义向量列表 vector<int>A(10,1)，使用两个参数，10 表示长度，1 表示数值。

16. vector 类中向向量尾部插入一个对象的方法是___。

答案：push_back

17. C++语言中如果调用函数时，需要改变实参或者返回多个值，应该采取___方式。

答案：传地址或引用

［解析］传地址即指针，在函数中通过指针修改它指向的变量的值时，实参也就变化了。使用引
用，直接修改变量的别名即引用的值，该变量也就随着变化。

18. 若函数的定义处于调用它的函数之前，则在程序开始可以省去该函数的___语句。

答案：声明

［解析］函数使用有两部分：声明和定义。定义在前，可以无声明；但函数定义在后，调用在前
的话，需要先声明函数的原型。

19. 在 C++中有两种参数传递方式：传值和___。

答案：传引用

［解析］（1）传值调用又分为数据传值调用和地址传值调用。（2）引用调用是将实参变量值传
递给形参，而形参是实参变量的引用名。引用是给一个已有变量起的别名，对引用的操作就是对
该引用变量的操作。

20. 将指向对象的引用作为函数的形参，形参是对象的引用，实参是 ___。

答案：对象名

［解析］实参与形参类型要一致，形参是对象的引用，实参应该是对象名。

软院面试指南

专业面试的话还是比较轻松的，一个人平均也就十分钟，如果是学硕的话，时间相对就久一点。不过有一些问题也是经常被问到的，比如**快排的原理，时间复杂度，数据库里面的范式，进程、线程的区别**，什么是优先级队列？老师们特别喜欢问一些专业相关的问题，通信专业要关注信息论编码相关问题，软件工程专业要熟悉软工常见概念，信息专业密码学一定要十分了解。有一组老师特别喜欢问一些**硬件相关**的问题，所以大家对硬件这一块应给予重视，下面是一些常见问题总结。

好多问题我并没有列出答案，希望大家自己总结答案。

1. 好多人会问到时间复杂度

   各种排序的时间复杂度，尤其是快排的复杂度，原理，应能做到**手撸快排**。

2. 各种排序的时间复杂度和性能比较

   请自己查找答案

3. 什么叫堆排序？与快速排序有神马不同？

   请自己查找答案

4、循环队列的顺序表示中，为什么要空一个位置？

   队列那个为了辨别对空和队满

5. 什么是二叉查找树，原理

   二叉排序树（Binary Sort Tree）又称二叉查找树。 它或者是一棵空树；或者是具有下列性质的二叉树：（1） 若左子树不空，则左子树上所有结点的值均小于它的根结点的值； （2）若右子树不空，则右子树上所有 结点的值均大于它的根结点的值； （3）左、右子树也分别为二叉排序树；

   步骤：若根结点的关键字值等于查找的关键字，成功。 否则，若小于根结点的关键字值，递归查左子树。 若大于根结点的关键字值，递归查右子树。若子树为空，查找不成功。

6. 排序算法最优的时间复杂度

   O(logn)

7. 哈夫曼树以及哈夫曼树的构造

8. 什么是哈希冲突，及**如何解决**

哈希表：散列表（Hash table，也叫哈希表），是根据关键码值(Key value)而直接进行访问的数据结构。 也就是说，它通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。这个映射函数叫做散 列函数，存放记录的数组叫做散列表。

常用散列函数：

1、 直接寻址法

2、 数字分析法

3、 平方取中法

4、 折叠法

5、 随机数法

6、 除留余数法 处理冲突的方法： 1、 开放寻址法 2、 再散列法 3、 链地址法 4、 建立一个公共溢出区 散列因子： 散列表的装填因子定义为：α= 填入表中的元素个数 / 散列表的长度 α 是散列表装满程度的标志因子。由于表长是定值，α 与“填入表中的元素个数”成正比，所以，α 越大， 填入表中的元素较多，产生冲突的可能性就越大；α 越小，填入表中的元素较少，产生冲突的可能性就越小。

9. 深度、广度搜索的过程

图的深度优先遍历： 假设给定图 G 的初态是所有顶点均未曾访问过。在 G 中任选一顶点 v 为初始出发点(源点)，则深度优 先遍历可定义如下：首先访问出发点 v，并将其标记为已访问过；然后依次从 v 出发搜索 v 的每个邻接点 w。 若 w 未曾访问过，则以 w 为新的出发点继续进行深度优先遍历，直至图中所有和源点 v 有路径相通的顶点 (亦称为从源点可达的顶点)均已被访问为止。若此时图中仍有未访问的顶点，则另选一个尚未访问的顶点 作为新的源点重复上述过程，直至图中所有顶点均已被访问为止。 图的深度优先遍历类似于树的前序遍历。采用的搜索方法的特点是尽可能先对纵深方向进行搜索。这 种搜索方法称为深度优先搜索(Depth-First Search)。相应地，用此方法遍历图就很自然地称之为图的深度 优先遍历。 使用数据结构：堆栈 图的广度优先遍历： 设 x 是当前被访问顶点，在对 x 做过访问标记后，选择一条从 x 出发的未检测过的边(x，y)。若发现 顶点 y 已访问过，则重新选择另一条从 x 出发的未检测过的边，否则沿边(x，y)到达未曾访问过的 y，对 y 访问并将其标记为已访问过；然后从 y 开始搜索，直到搜索完从 y 出发的所有路径，即访问完所有从 y 出 发可达的顶点之后，才回溯到顶点 x，并且再选择一条从 x 出发的未检测过的边。上述过程直至从 x 出发 的所有边都已检

测过为止。此时，若 x 不是源点，则回溯到在 x 之前被访问过的顶点；否则图中所有和源 点有路径相通的顶点(即从源点可达的所有顶点)都已被访问过，若图 G 是连通图，则遍历过程结束，否则 继续选择一个尚未被访问的顶点作为新源点，进行新的搜索过程。
使用数据结构：队列

10. 迪杰斯克拉算法的过程

Dijkstra(迪杰斯特拉)算法是典型的单源最短路径算法，用于计算一个节点到其他所有节点的最短路 径。主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。Dijkstra 算法是很有代表性的最短路 径算法，在很多专业课程中都作为基本内容有详细的介绍，如数据结构，图论，运筹学等等。Dijkstra 一般 的表述通常有两种方式，一种用永久和临时标号方式，一种是用 OPEN, CLOSE 表的方式，这里均采用永 久和临时标号的方式。注意该算法要求图中不存在负权边。 算法步骤如下： 1、初使时令 S={V0},T={其余顶点}，T 中顶点对应的距离值若存在，d(V0,Vi)为弧上的权 值若不存在，d(V0,Vi)为∝ 2、从 T 中选取一个其距离值为最小的顶点 W 且不在 S 中，加入 S 3、对 T 中顶点的距离值进行修改：若加进 W 作中间顶点，从 V0 到 Vi 的距离值比不加 W 的路径要短， 则修改此距离值重复上述步骤 2、3，直到 S 中包含所有顶点，即 S=T 为止

11. 链表查询某个元素，平均时间复杂度是多少？ O(n)

12. 图的存储方式

图没有顺序映像的存储结构 1、 邻接矩阵：用两个数组分别存储数据元素（顶点）的信息和数据元素之间的关系（边或 弧）的信息。图的邻接矩阵表示是唯一的，且无向图的邻接矩阵一定是一个对称矩阵。 2、 邻接表：是图的链式存储结构。3、 十字链表：有向图的另一种链式存储结构 4、 邻接多重表：无向图的链式存储结构

13. 图的深度遍历是否唯一

不唯一

14. 图相关概念

有向图 无向图 弧，弧头，弧尾 边

有向完全图：n*(n-1)条弧的有向图

完全图：1/2*n*(n-1)条边的无向图

稀疏图：很少条边或弧（如 enlogn 权：有时图的边或弧具有与它相关的数，这种与图的边或弧相关的数叫做权 网：带权图称为网 子图：假设有两个图 G=(V, {E})和 G'=(V', {E'})，如果 V'是 V 的子集，E'是 E 的子集，则称 G'为 G 的子 图。 路径：顶点序列，路径的长度就是路径上的边或弧的数目。

回路/环：第一个顶点和最后一个顶点相同的路径称为回路/环 简单路径：序列中顶点不重复出现的路径称为简单路径

简单回路/简单环：除了第一个顶点和最后一个顶点之外，其余顶点不重复出现的回路，称为简单回路/简单 环。

出度： 入度： 度： 连通：无向图中从顶点 A 到顶点 B 有路径，则称两 个顶点连通。 强连通图：有向图中任意两个不同顶点 A、B，从顶 点 A 到顶点 B 和从顶点 B 到顶点 A 都存在路径，则 称为强连通图。 连同图：图中任意两个顶点都是连通的，则称为连 通图。

强连通分量：有向图中的极大强连通子图 连通分量：无向图中的极大连通子图

生成森林：一个有向图的生成森林由若干棵有向树 组成，含有图中全部顶点，但只有足以构成若干棵 不相交的有向树的弧。生成树：极小连通子图，含有图中全部顶点，但只 有足以构成一棵树的（n-1）条边。如果在一棵生成树上添加一条边，必定构成一个环。 有向树：如果一个有向图恰好有一个顶点的入度为 0，其余顶点的入度均为 1，则是一棵有向树。

15. 连通图的概念

在图论中，连通图基于连通的概念。在一个无向图 G 中，若从顶点 vi 到顶点 vj 有路径相连(当然从 vj 到 vi 也一定有路径)，则称 vi 和 vj 是连通的。如果 G 是有向图，那么连接 vi 和 vj 的路径中所有的边都 必须同向。如果图中任意两点都是连通的，那么图被称作连通图。图的连通性是图的基本性质。

16. 解释下最小生成树

极小连通子图，含有图中全部顶点，但只有足以构成一棵树的(n-1)条边。如果在一棵生成树上添 加一条边，必定构成一个环。

17. n 个节点的图的最小生成树有几个节点，几条边 n 个顶点，n-1 条边。

18. 平衡二叉树

平衡二叉树（Balanced Binary Tree）又被称为 AVL 树（有别于 AVL 算法），且具有以下性质：它是一 棵 空树或它的左右两个子树的高度差的绝对值不超过 1(-1,0,1)，并且左右两个子树都是一棵平衡二叉树。 AVL 树是最先发明的自平衡二叉查找树。AVL 树得名于它的发明者 G.M. Adelson-Velsky 和 E.M. Landis， 他们在 1962 年的论文 "An algorithm for the organization of information" 中发表了它。 查找、插入和删除在平均和最坏情况下都是 O（logn）。增加和删除可能需要通过一次或多次树旋转来重新 平衡这个树。

19. 二叉树怎么存储

1、顺序存储结构：用一组地址连续的存储单元依次自上而下、自左向右存储完全二叉树上的结点信息。这 种顺序存储结构仅适用于完全二叉树。否则其他形式的二叉树用顺序存储，会浪费不少存储空间。

2、链式存储结构：二叉链表、三叉链表。含有 n 个结点的二叉链表中有 n+1 个空链域。

20. 单链表就地逆置

所谓 “就地”是指算法的辅助空间为 O(1)

思路：用指针 p 扫描原单链表，先将头节点 L 的 next 域设置为 NULL 而变成一个空链表，然后将*p 节点 采用头插法插入到 L 中。

21. 各种查找总结

22. m 阶的 B-树和 m 阶的 B+树主要区别

m 阶的 B-树和 m 阶的 B+树主要区别

1、 B+树所有有效数据全在叶子节点，而 B-树所有节点分散在树中，B-树中的关键字不重复。

2、 B+树种有几个关键字就有几个子树，B-树中具有 n 个关键字的节点含有(n+1)棵子树。

3、 B+树有两个指针，根指针和只想最小节点的指针，叶子节点连接成一个不定长的线性链表

4、 B+树中，每个节点（除根节点外）中的关键字个数 n 的取值范围是 [m/2]<=n<=m，根节点 n 的取值 范围是 2<=n<=m。B-树中，每个节点（除根

节点外的所有最底层非叶子节点）中的关键字取值范围是 [m/2]-1<=n<=m-1，根节点 n 的取值范围是 1<=n<=m-1。

5、 B+树中的所有非叶子节点仅仅起到索引的作用，节点中的每个索引项只包含对应子树的最大关键字和 指向该子树的指针，不含有该关键字对应记录的存储地址。而在 B-树中，每个关键字对应记录的存储 地址。

23. 折半查找，适用范围和时间复杂度

适用范围：顺序存储结构且要求元素按关键字有序排列

时间复杂度：o(log 2 n)

24. 计算机和计算器的区别(这个问题去年再次被问到)

计算器：具有简单计算功能，有些具有简单存储功能，不能自动工作，而计算机可以通过编程实现程序自 动运行。价位便宜。 计算机：高速计算的电子计算机，可进行数值计算，逻辑计算，还具有存储记忆功能。自动化程度高于计 算器。 实际上二者还有另一个本质性的区别。计算器使用的是固化的处理程序，只能完成特定的计算任务； 而计算机借助操作系统平台和各类应用软硬件，可以无限扩展其应用领域。也就是说，是否具有扩展性是 二者的本质区别。

25. 线程/进程空间是什么

线程运行所需要的内存空间，比如线程程序的存储空间，数据空间，运行空间等。 逻辑地址和物理地址之间的切换，是由 CPU 的内存管理单元 MMU 完成。Linux 内核维护每个进程的逻辑地 址到物理地址的对照表。当用户空间进程切换时，内核会更新对照表，用户空间也跟随变。 每个进程的用户空间都是相互独立的，把同一个程序同时运行 10 次，会看到 10 个进程使用的线性地址一 模一样但物理内存不一样。

26. 硬实时和软实时

在实时操作系统中，系统必须在特定的时间内完成指定的应用，具有较强的“刚性”，而分时操作系 统则注重将系统资源平均地分配给各个应用，不太在意各个应用的进度如何，什么时间能够完成。不过， 就算是实时操作系统，其“刚性”和“柔性”的程度也有所不同，就好像是系统的“硬度”有所不同，因 而有了所谓的“硬实时(hard real-time)”和“软实时 (soft real-time)”。 硬实时系统有一个刚性的、不可改变的时间限制，它不允许任何超出时限的错误。超时错

误会带来损 害甚至导致系统失败、或者导致系统不能实现它的预期目标。 软实时系统的时限是一个柔性灵活的，它可以容忍偶然的超时错误。失败造成的后果并不严重，例如 在网络中仅仅是轻微地降低了系统的吞吐量。

27. 进程和程序的区别

进程：程序的一次执行过程，一个动态的过程。存在生命周期，包括进程的创建、进程运行、进程挂起、 进程结束。 程序：代码+数据的一个集合，一个静态的表示。

28. 进程和线程的区别

进程：资源分配和调度的基本单位，进程切换耗费资源比线程切换大。进程有独立的进程地址空间。 线程：线程是进程中的一个实体，是 CPU 调度的基本单位，是比进程更小的能独立运行的基本单位，线程 自己不拥有系统资源，只拥有运行中必不可少的资源（如程序计数器、寄存器、栈），在同一个线程内，可 以有多个线程，多个线程共享同一个进程的资源，每个线程具有自己的线程栈。线程没有独立的线程地址 空间，多个线程共享同一个进程地址空间。

29. 什么是微内核

微内核是一种能够提供必要服务的操作系统内核；其中这些必要的服务包括任务，线程，交互进程通信 （IPC，Inter－Process Communication）以及内存管理等等。所有服务（包括设备驱动）在用户模式下运 行，而处理这些服务同处理其他的任何一个程序一样。因为每个服务只是在自己的地址空间运行。所以这 些服务之间彼此之间都受到了保护。 微内核是提供操作系统核心功能的内核的精简版本，如 Windows Linux 是一个单内核，也就是说 Linux 内核运行在单独的内核地址空间。不过，Linux 汲取了微内核的精华， 其引以为豪的是模块化设计、抢占式内核、支持内核线程已经动态装载内核模块的能力。

30. call 和 return 具体做了哪些工作（硬件问题给予重视）

call：参数压栈、返回地址压栈、保护现场 return：返回地址、恢复现场

31. 什么是 DMA，什么是中断。DMA 和中断有什么区别？（硬件问题给予重视）

DMA：Direct Memory Access 直接内存访问，为了实现外设<->CPU<->存储器

之间的不足，从而实现<-> 存储器。CPU 释放总线，由 DMA 控制器管理。中断：是指在 CPU 正常运行程序时，由于内部/外部事件引起 CPU 暂时停止正在运行的程序，转而去执行 请求 CPU 服务的内部事件或外部事件的服务子程序，待该服务子程序处理完毕后又返回到被中止的程序继 续运行，这一过程叫做中断。 区别：首先概念功能就不一样，另外在 DMA 的过程中需要用到中断系统。

32. **硬中断和软中断是什么，区别是什么？ （硬件问题给予重视）**

软中断： 1、编程异常通常叫做软中断 2、软中断是通讯进程之间用来模拟硬中断的 一种信号通讯方式。 3、中断源发中断请求或软中断信号后,CPU 或接收进程在适当的时机自动进行中断处理或完成软中断信号 对应的功能 4、软中断是软件实现的中断,也就是程序运行时其他程序对它的中断;而硬中断是硬件实现的中断,是程序运 行时设备对它的中断。 硬中断： 1、硬中断是由外部事件引起的因此具有随机性和突发性；软中断是执行中断指令产生的，无外部施加中断 请求信号，因此中断的发生不是随机的而是由程序安排好的。2、硬中断的中断响应周期，CPU 需要发中断回合信号（NMI 不需要），软中断的中断响应周期，CPU 不 需发中断回合信号。 3、硬中断的中断号是由中断控制器提供的（NMI 硬中断中断号系统指定为 02H）；软中断的中断号由指 令直接给出，无需使用中断控制器。 4、硬中断是可屏蔽的（NMI 硬中断不可屏蔽），软中断不可屏蔽。 区别： 1、软中断发生的时间是由程序控制的,而硬中断发生的时间是随机的 2、软中断是由程序调用发生的,而硬中断是由外设引发的 3、硬件中断处理程序要确保它能快速地完成它的任务,这样程序执行时才不会等待较长时间

33. 页面置换算法有哪些？什么是 LRU

在地址映射过程中，若在页面中发现所要访问的页面不再内存中，则产生缺页中断。当 发生缺页中断时操作系统必须在内存选择一个页面将其移出内存，以便为即将调入的页面让 出空间。而用来选择淘汰哪一页的规则叫做页面置换算法 常见的置换算法有： 01．最佳置换算法（OPT）（理想置换算法） 02．先进先出置换算法（FIFO）： 03．最近最久未使用（LRU）算法 04．Clock 置换算法（LRU 算法的近似实现） 05．最少使用（LFU）置换算法 06．工作集算法 07．工作集时钟算法 08．老化算法（非常类似 LRU 的有效算法） 09．NRU(最近未使用）算法 10．第二次机会算法 LRU 是 Least Recently Used 最近最少使用算法。 为了尽量减少与理想算法的差距，产生了各种精妙的算

法，最近最少使用页面置换算法 便是其中一个。LRU 算法的提出，是基于这样一个事实：在前面几条指令中使用频繁的页 面很可能在后面的几条指令中频繁使用。反过来说，已经很久没有使用的页面很可能在未来 较长的一段时间内不会被用到。这个，就是著名的局部性原理 ——比内存速度还要快的 cache，也是基于同样的原理运行的。因此，我们只需要在每次调换时，找到最近最少使用 的那个页面调出内存。这就是 LRU 算法的全部内容。

34. 操作系统中磁盘调度算法

磁盘调度的目标是，使磁盘的平均寻道时间最少。 调度算法名称 基本原理 特点 先来先服务（FCFS：First Come First Serve） 仅适用于请求磁盘 IO 的进程 数目较少的场合 最简单的一种磁盘调度算法， 根据进程请求访问磁盘的先 后次序进行调度。 优点：简单，公平，每一个请 求的进程都能够得到处理，不 会出现长期得不到处理的进 程。 缺点：未对寻道进行优化，致 使平均寻道时间可能较长。 最短寻道时间优先 (SSTF ： Shorted Seek Time First) 该算法选择这样的进程，其要 求访问的磁道，与当前磁头所 在的磁道距离最近，以使每次 的寻道时间最短，但这种算法 不能保证平均寻道时间最短 优点：较 FCFS 有更好的寻道 性能 缺点：可能导致老进程饥饿， 可能会出现磁臂黏着 扫描算法(SCAN)==电梯调度 算法 在 SSTF 算法基础上优化而得到，可以防止老进程饥饿。 该算法不仅考虑到被访问磁 道和当前磁头之间的距离，更 优先考虑的是磁头当前的移 动方向。(应该选取同方向且距 离最近的磁道访问。) 优点：磁头双向移动，不会产 生饥饿，平均寻道时间短。 缺点：可能会出现磁臂黏着 循环扫描算法(CSCAN) 为了减少延迟，CSCAN 规定 磁头单向移动，当从里到最外 后，磁头立即返回到最里的欲 访问磁道，然后继续向外。 优点：磁头单向移动，不会产 生饥饿，平均寻道时间短。 N-Step-SCAN 算法，对 SCAN 算法的优化。 将磁盘请求队列分成若干个 长度为 N 的子队列，磁盘调度 将按照 FCFS 依次处理这些子 队列，而每处理一个队列时又 是按照 SCAN 算法，对一个队 优点：无磁臂黏着。 列处理后再处理其他队列，将 新请求队列放入新队列。FSCAN 算法，对 SCAN 算法 的优化。 将请求队列分成两个子队列， 将新出现请求磁盘 IO 的进程 放入另一个子队列。 优点：无磁臂黏着。

35. 操作系统中的信号量

信号量是一种实现进程同步和互斥的工具。 1、 整型信号量：所谓整型信号量就是一个用于表示资源个数的整型量 2、 记录性信号量：就是用一个

结构体实现，里面包含了表示资源个数的整型量和一个等待 队列。 Linux 内核代码对 semaphore 的实现 struct semaphore { raw_spinlock_t lock; unsigned int count; struct list_head wait_list; };

36. 什么是 Pv 操作

P、V 操作以原语形式实现，信号量的值仅能由这两条原语加以改变。 P 操作相当于申请资源，V 操作相当于释放资源。

37. 什么是操作系统

操作系统是管理电脑硬件与软件资源的程序，同时也是计算机系统的内核与基石。操作系统是控制其他程 序运行，管理系统资源并为用户提供操作界面的系统软件的集合。操作系统身负诸如管理与配置内存、决 定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。操作系统的型 态非常多样，不同机器安装的 OS 可从简单到复杂，可从手机的嵌入式系统到超级电脑的大型操作系统。 目前微机上常见的操作系统有 DOS、OS/2、UNIX、XENIX、LINUX、Windows、Netware 等。

38. 简述操作系统中系统调用过程

系统调用把应用程序的请求传给内核，调用相应的的内核函数完成所需的处理，将处理结果返回给应用程 序，如果没有系统调用和内核函数，用户将不能编写大型应用程序。 Linux 系统调用，包含了大部分常用系统调用和由系统调用派生出的的函数。

39. 虚拟存储器，虚存，问有啥相关算法

虚拟存储器：由于常规内存的一次性（要求将作业全部装入内存后才能运行）和驻留性（作业装入内存后， 就一直驻留在内存中，直到作业运行结束。）特点，难以满足作业很大和有大量作业要求运行的情况。虚 拟存储器是一种借助于外存空间，从而允许一个进程在其运行过程中部分地装入内存的技术。之所以引入虚拟存储管理方式，是因为程序执行时呈现局部性规律。 局部性原理： 1、 时间局部性：一条指令的一次执行和下次执行，一个数据的一次执行和下次执行，都集中在一个较短 时间内。 2、 空间局部性：指当前指令和邻近的几条指令，当前访问的数据和邻近的数据，都集中在一个较小区域。 实现虚拟存储器的硬件支持： 1、 相当数量的外存 2、 相当数量的内存 3、 地址变换机构：以动态实现虚拟地址到实地址的地址变换。 替换算法：替换规

则用来确定替换主存中哪一部分，以便腾空部分主存，存放来自辅存要调入的那部分内 容。 1、 随机算法：用软件或硬件随机数产生器确定替换的页面。2、 先进先出：先调入主存的页面先替换。 3、 LRU 算法：替换最长时间不用的页面。 4、 最优算法：替换最长时间以后才使用的页面。这是理想化的算法，只能作为衡量其他各种算法优劣的 标准。 5、 CLOCK 置换算法

40. 存储器管理应具有的功能？

1、内存的分配和回收 2、地址变换：在多道程序环境下，程序的逻辑地址与内存的物理地址不可能一致，因此存储管理器必须提 供地址变换机构，将逻辑地址装换为物理地址。 3、扩充内存容量：从逻辑上扩充内存容量 4、存储保护：保护进入内存的各道作业在自己的存储空间运行，互补干扰。 地址重定位：将目标程序中的逻辑地址转换为可执行代码的实际内存物理地址

41. 什么是 TLB cache

是一种高速缓存存储器，用于保存 CPU 频繁使用的数据。在使用 Cache 技术的处理器上，当一条 指令要访问内存的数据时，首先查询 cache 缓存中是否有数据以及数据是否过期，如果数据未过期则从 cache 读出数据。处理器会定期回写 cache 中的数据到内存。根据程序的局部性原理，使用 cache 后可以 大大加快处理器访问内存数据的速度。 TLB 的作用是在处理器访问内存数据的时候做地址转换。TLB 的全称是 Translation Lookaside Buffer，可 以翻译做旁路缓冲。TLB 中存放了一些页表文件，文件中记录了虚拟地址和物理地址的映射关系。当应用 程序访问一个虚拟地址的时候，会从 TLB 中查询出对应的物理地址，然后访问物理地址。TLB 通常是一 个分层结构，使用与 Cache 类似的原理。处理器使用一定的算法把最常用的页表放在最先访问的层次。 TLB 加速了查表速度，TLB（块表）放在 cache 中，用于存放慢表中常用的部分内容，专用、快速 的硬件缓冲。

42. 程序连接方式有哪些？

1、静态链接： 2、装入时动态链接：边装入边链接 3、运行时动态链接：方便对目标模块的共享

43. 程序的装入方式有哪些？

1、绝对装入： 2、可重定位装入：也叫静态重定位，地址变换通常是装入时一次性完成的。 3、动态运行装入：程序运行时可在内存中移动位置，只有

到程序需要真正执行时才把相对地址转换为绝对 地址。

44. 什么是交换技术？什么是覆盖技术？及其区别

覆盖技术：把一个大的程序划分为一系列的覆盖，每个覆盖就是一个相对独立的程序单元，把程序执行时 并不要求同时装入内存的覆盖组成一组，称为覆盖段。将一个覆盖段分配到同一个存储区域，这个存储区 域就称为覆盖区。 交换技术：就是把暂时不用的某个程序及数据部分或全部从内存移到外存中去，以便腾出更多的内存空间。 主要区别： 1、交换主要是在进程和作业之间进行 2、覆盖主要在同一个进程或作业中进行 3、打破了一个程序一旦进入主存便一直运行到结束的限制 4、打破了必须将一个进程的全部信息装入主存后才能运行的限制

45. 内存连续分配管理方式有哪几种？

1、单一连续：适合单道程序 2、固定分区分配：适合多道程序，简单。 3、动态分区分配：根据内存实际需要，动态地为进程分配空间。

46. 动态分区分配算法有哪些？

动态分区分配将内存的空闲分区单独构成一个空间分区表或空闲分区链

算法名称	原理	优点	缺点	要求
首次适应算法	从链首开始循序 查找，找到第一个 满足空间要求的 空闲分区为止。	优先利用低地址部 分，无内部碎片	低地址内存被无 限划分，有外部碎 片。	空闲分区链以地址 递增的次序连接
循环首次适应算法	从上次找到空闲 分区的下一个空 间分区开始查找。	空闲分区分布均匀， 无内部碎片。	缺乏大的空闲分 区，有外部碎片。	空闲分区链以地址 递增的次序连接
最佳适应算法	把满足要求又是 最小的空闲分区分配给作业	产生的外部碎片很 小，无内部碎片。	产生需要无法利 用的外部小碎片。	空闲分区链以容量 从小到大的次序连 接
最坏适应算法	把满足要求又是 留下来的空闲空间 大空间不易被保	空闲分区链以容量 最大的空闲分区	分配给作业 比较大，适合下次使 用，无内部碎片。	留 从大到小的次序连 接

47. 什么是拼接技术？

拼接技术/紧凑技术：移动存储中所有已分配区到内存的一端，将其余空闲分区合并为一个大的空闲分区。

48. 什么是原子操作

"原子操作(atomic operation)是不需要 synchronized"，这是 Java 多线程编程的老生常谈了。所谓原子操作 是指不会被线程调度机制打断的操作；这种操作一旦开始，就一直运行倒结束，中间不会有任何 context switch (切换到另一个线程)。

49. 什么内部碎片？什么是外部碎片？

内部碎片：分配给作业的存储空间中未被利用的部分 外部碎片：系统中无法利用的小存储块，比如通过动态内存分配技术从空闲内存区上分配内存后剩下的那 部分内存块。

50. 常用存储保护方法有哪些？

1、界限寄存器 上下界寄存器方法 基址、限长寄存器方法

2、存储保护键：给每个存储块分配一个单独的存储键，它相当于一把锁。

51. 连续分区分配 vs 非连续分区分配

连续内存分配方式 非连续内存分配方式：允许将程序分散装载 到很多个不相邻的小分区中。

基本分页存储管理方式 请求分页存储管理方式

52. 什么是页面？什么是块或物理块？

页面：作业地址空间被划分为若干个大小相等的区域，页面大小应该是 2 的整数幂，通常为 4KB。 块/物理块：将内存存储空间也分为和页大小相等的区域，这些区域被称为块。

53. 什么是页表？及页表的作用？

页表：为了便于在内存中找到进程的每个页面所对应的物理块，系统为每个进程建立一张页面映射表。

54. 段寄存器

段寄存器：在 8086 系统中，访问存储器的地址码由段地址和段内偏移地址两部分组成。段寄存器用来存 放各分段的逻辑基值，并指示当前正在使用的 4 个逻辑段，包括代码段寄存器 CS、堆栈段寄存器 SS、数 据段寄存器 DS 和附加段数据寄存器 ES。 1、代码段寄存器 CS：存放当前正在运

行的程序代码所在段的段基值，表示当前使用的指令代码可以 从该段寄存器指定的存储器段中取得，相应的偏移值则由 IP 提供。 2、数据段寄存器 DS：指出当前程序使用的数据所存放段的最低地址，即存放数据段的段基值。3、堆栈段寄存器 SS：指出当前堆栈的底部地址，即存放堆栈段的段基值。4、附加段寄存器 ES：指出当前程序使用附加数据段的段基址，该段是串操作指令中目的串所在的段。

55. 进程线程树图

进程树是一个形象化的比喻，比如一个进程启动了一个程序，而启动的这个进程就是原来那个进程的 子进程，依此形成的一种树形的结构，我们可以在进程管理器选择结束进程树，就可以结束其子进程和派 生的子进程。

56. 作业与进程的区别

一个进程是一个程序对某个数据集的执行过程，是分配资源的基本单位。作业是用户需要计算机完成的某 项任务，是要求计算机所做工作的集合。一个作业的完成要经过作业提交、作业收容、作业执行和作业完 成 4 个阶段。而进程是对已提交完毕的程序所执行过程的描述，是资源分配的基本单位。其主要区别如下。

(1)作业是用户向计算机提交任务的任务实体。在用户向计算机提交作业后，系统将它放入外存中的作业等 待队列中等待执行。而进程则是完成用户任务的执行实体，是向系统申请分配资源的基本单位。任一进程， 只要它被创建，总有相应的部分存在于内存中。

(2) 一个作业可由多个进程组成，且必须至少由一个进程组成，反过来则不成立。

(3) 作业的概念主要用在批处理系统中，像 UNIX 这样的分时系统中就没有作业的概念。而进程的概念则 用在几乎所有的多道程序系统中进程是操作系统进行资源分配的单位。在 Windows 下,进程又被细化为线 程,也就是一个进程下有多个能独立运行的更小的单位。

57. 进程的三种状态，以及之间转换的过程

58. 进程调度算法

先来先服务，短作业，高优先权，时间片，高响应比，多级反馈队列，这些是进程调度算法吧。

59. 死锁

死锁： 是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作 用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称 为死锁进程。

产生死锁的必要条件：

1、 互斥条件：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。

2、 请求和保持条件：指进程已经保持至少一个资源，但又提出了新的资源请求，而该资源已被其它进程 占有，此时请求进程阻塞，但又对自己已获得的其它资源保持不放。

3、 不剥夺条件：指进程已获得的资源，在未使用完之前，不能被剥夺，只能在使用完时由自己释放。

4、 环路等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0，P1，P2，…， Pn}中的 P0 正在等待一个 P1 占用的资源；P1 正在等待 P2 占用的资源，……，Pn 正在等待已被 P0 占用的资源。

处理死锁的基本方法： 1、 预防死锁：这是一种较简单和直观的事先预防的方法。方法是通过设置某些限制条件，去破坏产生死 锁的四个必要条件中的一个或者几个，来预防发生死锁。预防死锁是一种较易实现的方法，已被广泛 使用。但是由于所施加的限制条件往往太严格，可能会导致系统资源利用率和系统吞吐量降低。 2、 避免死锁：该方法同样是属于事先预防的策略，但它并不须事先采取各种限制措施去破坏产生死锁的 的四个必要条件，而是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免 发生死锁。 3、 检测死锁：这种方法并不须事先采取任何限制性措施，也不必检查系统是否已经进入不安全区，此方 法允许系统在运行过程中发生死锁。但可通过系统所设置的检测机构，及时地检测出死锁的发生，并精确地确定与死锁有关的进程和资源，然后采取适当措施，从系统中将已发生的死锁清除掉。 4、 解除死锁：这是与检测死锁相配套的一种措施。当检测到系统中已发生死锁时，须将进程从死锁状态 中解脱出来。常用的实施方法是撤销或挂起一些进程，以便回收一些资源，再将这些资源分配给已处 于阻塞状态的进程，使之转为就绪状态，以继续运行。

60. 分页和分段的区别

61. 死锁及死锁原因

产生死锁的原因： 1、 竞争资源引起的进程死锁：当系统中供多个进程共享的资源如打印机、公用队列的等，其数目不足以 满足诸进程的需要时，会引起诸进程对资源的竞争而产生死锁。 2、 进程推进顺序不当引起的死锁：由于进程在运行中具有异步性特征，这可能使 P1 和 P2 两个进程推进顺序出现问题。

62. 介绍下银行家算法

避免死锁算法中最有代表性的算法是 Dijkstra E.W 于 1968 年提出的银行家算法： 该算法需要检查申请者对资源的最大需求量，如果系统现存的各类资源可以满足申请者的请求，就满足申 请者的请求。 这样申请者就可很快完成其计算，然后释放它占用的资源，从而保证了系统中的所有进程都能完成，所以 可避免死锁的发生。

63. RAID 磁盘阵列（Redundant Arrays of Inexpensive Disks，RAID）， 原理是利用数组方式来作磁盘组，配合数 据分散排列的设计，提升数据的安全性。

RAID 0：RAID 0 连续以位或字节为单位分割数据，并行读/写于多个磁盘上，因此具有很高的数据传输率， 但它没有数据冗余，因此并不能算是真正的 RAID 结构。RAID 0 只是单纯地提高性能，并没有为数据的可 靠性提供保证，而且其中的一个磁盘失效将影响到所有数据。因此，RAID 0 不能应用于数据安全性要求高 的场合。

RAID 1：它是通过磁盘数据镜像实现数据冗余，在成对的独立磁盘上产生互为备份的数据。当原始数据繁 忙时，可直接从镜像拷贝中读取数据，因此 RAID 1 可以提高读取性能。RAID 1 是磁盘阵列中单位成本最 高的，但提供了很高的数据安全性和可用性。当一个磁盘失效时，系统可以自动切换到镜像磁盘上读写， 而不需要重组失效的数据。

RAID 0+1: 也被称为 RAID 10 标准，实际是将 RAID 0 和 RAID 1 标准结合的产物，在连续地以位或 字节为单位分割数据并且并行读/写多个磁盘的同时，为每一块磁盘作磁盘镜像进行冗余。它的优点是同时 拥有 RAID 0 的超凡速度和 RAID 1 的数据高可靠性，但是 CPU 占用率同样也更高，而且磁盘的利用率比 较低。

RAID 2：将数据条块化地分布于不同的硬盘上，条块单位为位或字节，并使用称为“加重平均纠错码（海 明码）”的编码技术来提供错误检查及恢复。

这种编码技术需要多个磁盘存放检查及恢复信息，使得 RAID 2 技术实施更复杂，因此在商业环境中很少使用。

RAID 3：它同 RAID 2 非常类似，都是将数据条块化分布于不同的硬盘上，区别在于 RAID 3 使用简 单的奇偶校验，并用单块磁盘存放奇偶校验信息。如果一块磁盘失效，奇偶盘及其他数据盘可以重新产生 数据；如果奇偶盘失效则不影响数据使用。RAID 3 对于大量的连续数据可提供很好的传输率，但对于随机 数据来说，奇偶盘会成为写操作的瓶颈。

RAID 4：RAID 4 同样也将数据条块化并分布于不同的磁盘上，但条块单位为块或记录。RAID 4 使用 一块磁盘作为奇偶校验盘，每次写操作都需要访问奇偶盘，这时奇偶校验盘会成为写操作的瓶颈，因此 RAID 4 在商业环境中也很少使用。

RAID 5：RAID 5 不单独指定的奇偶盘，而是在所有磁盘上交叉地存取数据及奇偶校验信息。在 RAID 5 上，读/写指针可同时对阵列设备进行操作，提供了更高的数据流量。RAID 5 更适合于小数据块和随机读 写的数据。RAID 3 与 RAID 5 相比，最主要的区别在于 RAID 3 每进行一次数据传输就需涉及到所有的阵 列盘；而对于 RAID 5 来说，大部分数据传输只对一块磁盘操作，并可进行并行操作。在 RAID 5 中有“写 损失”，即每一次写操作将产生四个实际的读/写操作，其中两次读旧的数据及奇偶信息，两次写新的数据及 奇偶信息。

RAID 6：与 RAID 5 相比，RAID 6 增加了第二个独立的奇偶校验信息块。两个独立的奇偶系统使用不 同的算法，数据的可靠性非常高，即使两块磁盘同时失效也不会影响数据的使用。但 RAID 6 需要分配给 奇偶校验信息更大的磁盘空间，相对于 RAID 5 有更大的“写损失”，因此“写性能”非常差。较差的性能和复 杂的实施方式使得 RAID 6 很少得到实际应用。

RAID 7：这是一种新的 RAID 标准，其自身带有智能化实时操作系统和用于存储管理的软件工具，可 完全独立于主机运行，不占用主机 CPU 资源。RAID 7 可以看作是一种存储计算机（Storage Computer）， 它与其他 RAID 标准有明显区别。除了以上的各种标准（如表 1），我们可以如 RAID 0+1 那样结合多种 RAID 规范来构筑所需的 RAID 阵列，例如 RAID 5+3（RAID 53）就是一种应用较为广泛的阵列形式。用户一般 可以通过灵活配置磁盘阵列来获得更加符合其要求的磁盘存储系统。

64. 数据库里面的： 什么分级？什么是 ER 图？

（数据库里面的一些概念同样是重点）

实体-联系图(Entity-Relation Diagram)用来建立数据模型,在数据库系统概论中属于概念设计阶段，形 成一个独立于机器，独立于 DBMS 的 ER 图模型。 通常将它简称为 ER 图，相应地可把用 ER 图描绘的数 据模型称为 ER 模型。ER 图提供了表示实体（即数据对象）、属性和联系的方法，用来描述现实世界的概 念模型。 构成 E-R 图的基本要素是实体、属性和联系，其表示方法为：

实体型：用矩形表示，矩形框内写明实体名；

属性：用椭圆形或圆角矩形表示，并用无向边将其与相应的实体连接起来；多值属性由双线连 接；主属性名称下加下划线；

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向 边旁标上联系的类型 在 E-R 图中要明确表明 1 对多关系，1 对 1 关系和多对多关系。

1 对 1 关系在两个实体连线方向写 1；

1 对多关系在 1 的一方写 1，多的一方写 N；

多对多关系则是在两个实体连线方向各写 N,M

65. 数据库的三级模式结构

数据库领域公认的标准结构是三级模式结构，它包括外模式、模式和内模式，有效地组织、管理数据，提 高了数据库的逻辑独立性和物理独立性。用户级对应外模式 概念级对应模式 物理级对应内模式 视图，就是指观察、认识和理解数据的范围、角度和方法，是数据库在用户“眼中”的反映，很显然，不同 层次(级别)用户所“看到”的数据库是不相同的。 使不同级别的用户对数据库形成不同的视图。 数据库的三级模式分别为外模式、模式、内模式。

1、 模式 模式又称概念模式或逻辑模式，对应于概念级。它是由数据库设计者综合所有用户的数据，按照统一 的观点构造的全局逻辑结构，是对数据库中全部数据的逻辑结构和特征的总体描述，是所有用户的公共数 据视图(全局视图)。它是由数据库管理系统提供的数据模式描述语言(Data Description Language，DDL)来描 述、定义的，体现、反映了数据库系统的整体观。

2、 外模式 外模式又称子模式或用户模式，对应于用户级。它是某个或某几个用户所看到的数据库的数据视图， 是与某一应用有关的数据的逻辑表示。外模式是从模式导出的一个子集，包含模式中允许特定用户使用的 那部分数据。用户可以通过外模式描述语言来描述、定义对应于用户的数据记

录(外模式)，也可以利用数 据操纵语言(Data Manipulation Language，DML)对这些数据记录进行。外模式反映了数据库的用户观。

3、 内模式 内模式又称存储模式，对应于物理级，它是数据库中全体数据的内部表示或底层描述，是数据库最低 一级的逻辑描述，它描述了数据在存储介质上的存储方式和物理结构，对应着实际存储在外存储介质上的 数据库。内模式由内模式描述语言来描述、定义，它是数据库的存储观。 在一个数据库系统中，只有唯一的数据库， 因而作为定义 、描述数据库存储结构的内模式和定义、 描述数据库逻辑结构的模式，也是唯一的，但建立在数据库系统之上的应用则是非常广泛、多样的，所以 对应的外模式不是唯一的，也不可能是唯一的。 用户应用程序根据外模式进行数据操作，通过外模式一模式映射，定义和建立某个外模式与模式 间的对应关系，将外模式与模式联系起来，当模式发生改变时，只要改变其映射，就可以使外模式保持不变，对应的应用程序也可保持不变；另一方面，通过模式一内模式映射，定义建立数据的逻辑结构 (模式) 与存储结构(内模式)间的对应关系，当数据的存储结构发生变化时，只需改变模式一内模式映射，就能保 持模式不变，因此应用程序也可以保持不变。

66. 视图在数据库的第几层

数据库有三层模式：外模式（用户模式或子模式）、模式（逻辑模式）内模式（物理模式）。视图属于外模式范畴。

67. 数据库的两级映像是什么，作用

数据库的二级映像功能与数据独立性 为了能够在内部实现数据库的三个抽象层次的联系和转换，数据 库管理系统在这三级模式之间提供了两层映像。

（1）外模式 / 模式映像 对应于同一个模式可以有任意多个外模式。对于每一个外模式，数据库系统都有 一个外模式 / 模式映像，它定义了该外模式与模式之间的对应关系。 当模式改变时，由数据库管理员对各 个外模式 / 模式映像作相应的改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，从而 应用程序可以不必修改，保证了数据与程序的逻辑独立性。

（2）模式 / 内模式映像 数据库中只有一个模式，也只有一个内模式，所以模式 / 内模式映像是惟一的， 它定义了数据库的全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时，由数据库管理 员对模式

/内模式映像做相应改变，可以使模式保持不变，从而应用程序也不必修改。保证了数据与程序的 物理独立性。

68. 数据库，ER 模型转换成关系模型是数据库设计的第几个阶段。（重点）

第四步骤

第一步，规划。规划阶段的主要任务是进行建立数据库的必要性及可行性分析。如系统调查（即对企业全 面调查，画出组织层次图，以了企业组织结构），可行性分析，确定 DBS（数据库系统）的总目标和制定 项目开发计划。

第二步，需求分析。需求分析阶段应该对系统的整个应用情况作全面的、详细的调查，确定企业组织的目 标，收集支持系统总的设计目标的基础数据和对这些数据的要求，确定用户的需求，并把这些要求写成用 户和数据库设计者都能够接受的需求分析报告。这一阶段的工作只要有，分析用户活动，产生业务流程图； 确定系统范围，产生体统范围图；分析用户活动涉及的数据，产生数据流程图；分析系统数据，产生数据 字典。

第三步，概念设计。概念设计的目标是产生反应企业组织信息需求的数据库概念结构,即设计出独立与计算 机硬件和 DBMS（数据库管理系统）的概念模式。E-R 模型是主要设计工具。

第四步，逻辑结构设计。其目的是把概念设计阶段设计好的全局 E-R 模式转换成与选用的具体机器上的 DBMS 所支持的数据模型相符合的逻辑结构（包括数据库模式和外模式）。

第五步，数据库的物理设计。对于给定的数据模型选取一个最适合应用应用环境的物理结构的过程。数据 库的物理结构主要指数据库的存储记录格式、存储记录安排和存取方法，完全依赖于给定的硬件环境赫尔 数据库产品。

第六步，数据库的实现。该阶段主要有 3 项工作：1 建立实际数据库结构 2 装入试验数据对应用程序进行 调试 3 装入实际数据，进入试运行状态。

第七步，数据库的运行与维护。数据库系统的正式运行，标志着数据库设计与应用开发工作的结束和维护 阶段的开始，该阶段有 4 项任务：1 维护数据库的安全性与完整性 2 监测并改善数据库运行性能 3 根据用 户要求对数据库现有功能进行扩充 4 及时改正运行中发现的系统错误。

69. 数据库，数据模型有哪几种，说出至少两种的特征

1、非关系模型

层次模型：记录之间的联系通过指针实现，查找效率高。

网状模型：一个结点可以有多于一个的双亲，允许一个以上的结点无双亲。

2、关系模型：概念简单，结构清晰，用户易学易用

3、面向对象模型

4、对象关系模型

70. 数据库中什么叫主码

主关键字(primary key)是表中的一个或多个字段，它的值用于惟一地标识表中的某一条记录。

71. 数据库的表怎样分级

72. 事务

数据库事务(Database Transaction) ，是指作为单个逻辑工作单元执行的一系列操作。 事务处理可以确保除 非事务性单元内的所有操作都成功完成，否则不会永久更新面向数据的资源。 通过将一组相关操作组合为 一个要么全部成功要么全部失败的单元，可以简化错误恢复并使应用程序更加可靠。一个逻辑工作单元 要成为事务，必须满足所谓的 ACID(原子性、一致性、隔离性和持久性)属性。 数据库事务的 ACID 性 原子性：事务必须是原子工作单元 一致性：事务在完成时，必须使所有的数据都保持一致状态。 隔离性：由并发事务所作的修改必须与任何其它并发事务所作的修改隔离。 持久性：事务完成之后，它对于系统的影响是永久性的。

73. 数据库操纵语言，定义语言，定义、操作、查询、控制 数据操纵语言 DML（Data Manipulation Language），用户通过它可以实现对数据库的基本操作。例如， 对表中数据的插入、删除和修改。 插操作：数据操纵语言 DML（Data Manipulation Language），用户通过它可以实现对数据库的基本操作。 例如，对表中数据的插入、删除和修改。 删操作：删除数据库中不必再继续保留的一组记录，如 DELETE 对数据库中记录作删除标志。PACK 是将 标有删除标志的记录彻底清除掉。ZAP 是去掉数据库文件的所有记录。 改操作： 排序操作 检索操作

74. 数据库中怎样预防死锁

死锁发生在当多个进程访问同一数据库时，其中每个进程拥有的锁都是其他进程所需的，由此造成每个进 程都无法继续下去。 理解了死锁的原因，尤其是产生死锁的四个必要条件，就可以最大可能地避免、预防和解除死锁。下列方法有助于最大限度地降低死锁： （1）按同一顺序访问对象。 （2）避免事务中的用户交互。 （3）保持事务简短并在一个批处理中。 （4）使用低隔离级别。 （5）使用绑定连接。

75. 并发控制是为了保证事务的？

当多个用户并发地存取数据库时可能出现多个事务同时存取同一数据的情况，并发控制机制将对这些并发 操作加以控制以保证每个事务的 ACID 性，确保数据库的一致性。

76. 在数据库中什么是关系，它和普通二维表啥区别 二维表可以看作是一种关系，关系是二维表的超集

77. 视图、索引 视图概念：计算机数据库中的视图是一个虚拟表，其内容由查询定义。同真实的表一样，视图包含一系列 带有名称的列和行数据。但是，视图并不在数据库中以存储的数据值集形式存在。行和列数据来自由定义视图的查询所引用的表，并且在引用视图时动态生成。 视图作用： 1、 为用户集中数据，简化用户的数据查询和处理。 2、 屏蔽数据库的复杂性 3、便于数据共享 索引的概念：数据库中的索引和书籍中的索引类似。在数据库中，索引使数据库程序无需对整个表进行扫 描。就可以在其中找到所需要数据。 索引的作用： 1、 通过创建唯一索引，可以保证数据记录的唯一性。2、 可以大大加快数据检索速度。 3、 加速表与表之间的连接。

79. 数据库里的读锁、写锁

读写锁实际是一种特殊的自旋锁，它把对共享资源的访问者划分成读者和写者，读者只对共享资源进 行读访问，写者则需要对共享资源进行写操作。这种锁相对于自旋锁而言，能提高并发性，因为在多处理 器系统中，它允许同时有多个读者来访问共享资源，最大可能的读者数为实际的逻辑 CPU 数。写者是排他 性的，一个读写锁同时只能有一个写者或多个读者（与 CPU 数相关），但不能同时既有读者又有写者。

80. 数据库里，如何解决数据冗余问题？

81. 关系范式

第一范式：满足最低程度要求的范式，简称 1NF。 定义 设 R 是一个关系模式，如果 R 中的每一个属性 A 的值域中的每个值都是不可分解的，则称 R 是 属于第一范式的，记作 R ∈ 1NF。 例如：在关系 SA（姓名，工资）中，属性“工资”还可再分为基本工资，奖金还有补贴 3 个数据项，这违 背了第一范式中元组的每个属性不可再分的原则，所以它不满足第一范式。

第二范式：在第一范式中进一步满足一些要求的关系，简称 2NF。 定义如果关系 R ∈ 1NF，并且 R 中每一个非主属性完全函数依赖于任一个候选码，则 R ∈ 2NF。 从定义可以看出，若某个 1NF 的关系的主码只由一个列组成，那么这个关系就是 2NF 关系。但是， 如果主码是由多个属性列共同组成的复合主码，并且存在非主属性对属性的部分函数依赖，则这个关系不 是 2NF 关系。 例如：在关系 SB（学号，姓名，系名，系主任，课号，成绩）中，、 非主属性“姓名”仅函数依赖于“学号”，也就是“姓名”部分函数依赖于主码（学号，课号）而不是完全依 赖； 非主属性“系名”仅函数依赖于“学号”，也就是“系名”部分函数依赖于主码（学号，课号）而不是完全依 赖； 非主属性“系主任”仅函数依赖于“学号”，也就是“系主任”部分函数依赖于主码（学号，课号）而不是完 全依赖

第三范式： 对关系模式的属性间的函数依赖加以不同的限制就形成了不同的范式。 定义 如果关系 R ∈ 2NF，并且 R 中每一个非主属性对任何候选码都不存在传递函数依赖，则 R ∈ 3NF 。 从定义中可以看出，如果存在非主属性对主码的传递依赖，则相应的关系模式就不是 3NF。 接着上面的例子，关系模式 SC 和 SD 均是 2NF 的，但在关系 SD（学号，姓名，系名，系主任） 中，存在如下函数依赖： 学号 → 系名 系名 → 系主任 系名 —\→ 学号 那么，存在着一个传递函数依赖“学号 → 系主任”成立。

82. 断点之类的问题

所谓断点就是程序被中断的地方，这个词对于解密者来说是再熟悉不过了。那么什么又是中断呢？中断就 是由于有特殊事件（中断事件）发生，计算机暂停当前的任务（即程序），转而去执行另外的任务（中断服 务程序），然后再返回原先的任务继续执行。

83. 关于显卡的，显卡作用，原理。

显卡：显卡全称显示接口卡（Video card，Graphics card），又称为显示适配器（Video adapter），显示器 配置卡简称为显卡，是个人电脑最基本组成部分之一。 显卡作用：是将计算机系统所需要的显示信息进行转换驱动，并向显示器提供行扫描信号，控制显示器的 正确显示，是连接显示器和个人电脑主板的重要元件，是“人机对话”的重要设备之一。

工作原理： 1、 将 CPU 送来的数据送到北桥（主桥）再送到 GPU（图形处理器）里面进行处理。 2、 将芯片处理完的数据送到显存。 3、 从显存读取出数据再送到 RAM DAC 进行数据转换的工作（数字信号转模拟信号）。 4、 将转换完的模拟信号送到显示屏。

84. VGA VGA(Video Graphics Array)是 IBM 在 1987 年随 PS/2 机一起推出的一种视频传输标准，具有分辨率高、 显示速率快、颜色丰富等优点，在彩色显示器领域得到了广泛的应用。

85. FPGA FPGA（Field－Programmable Gate Array），即现场可编程门阵列，它是在 PAL、GAL、CPLD 等可编程 器件的基础上进一步发展的产物。它是作为专用集成电路（ASIC）领域中的一种半定制电路而出现的，既 解决了定制电路的不足，又克服了原有可编程器件门电路数有限的缺点。

86. 算数移位、逻辑移位、循环移位

算术移位：算术移位指令对带符号数进行移位。 逻辑移位：逻辑移位指令对无符号数进行移位。 这里有一个进位位 C，它就是标志寄存器（即状态寄存器，也称程序状态寄存器 PSW）中的那个进位位， 指示是否有进位或借位，若有则该位为 1，否则为 0。逻辑左移跟算术左移完全一样。

而逻辑右移跟算术右移则不一样，逻辑右移的最高位在移出后补 0，而在算术右移中，最高位不变（这里的最高位指整个编码 的最高位。即有符号数的符号位。），其他跟逻辑右移一样。 循环移位：分为带进位 C 和不带进位 C 两种 注意，在循环移位中没有算法移位、逻辑移位之分，只有是否带进位位之分，不要搞混淆了。 在循环移位中，只有“带进位的循环移位”这种方式中，进位位 C 才对移位后的结果产生 影响，其他饿进位位都是受影响（被新移入的二进制位覆盖）

91. cache 和虚拟存储的功能不同 cache 是为了提高常访问的内存块儿的速

度，虚拟存储是为了扩展空间。

95. 编译:如何把一个机器的语言拿到另一台机器语言机器上执行

96. 编译原理语法分析句法分析

词法分析（英语：lexical lexical lexical lexical analysis analysis analysis analysis）是计算机科学中将字符序列转换为单词（Token）序列的过程。 进行语法分析的程序或者函数叫作词法分析器（Lexical analyzer，简称 Lexer），也叫扫描器（Scanner）。 词法分析器一般以函数的形式存在，供语法分析器调用。 词法分析阶段是编译过程的第一个阶段，是编译的基础。这个阶段的任务是从左到右一个字符一个字 符地读入源程序，即对构成源程序的字符流进行扫描然后根据构词规则识别单词(也称单词符号或符号)。 词法分析程序实现这个任务。

语法分析是编译过程的一个逻辑阶段。语法分析的任务是在词法分析的基础上将单词序列组合成各类 语法短语，如“程序”，“语句”，“表达式”等等.语法分析程序判断源程序在结构上是否正确.源程序的结构由上 下文无关文法描述.语法分析程序可以用 YACC 等工具自动生成。 句法分析(Parsing)就是指对句子中的词语语法功能进行分析，比如“我来晚了”，这里“我”是主语，“来” 是谓语，“晚了”是补语。句法分析现在主要的应用在于中文信息处理中，如机器翻译等。

97. 编译的过程

(1) 词法分析程序 (也称为扫描器)；
(2) 语法分析程序 (有时简称为分析器)；
(3) 语义分析程序；
(4) 中间代码生成程序；
(5) 代码优化程序；
(6) 目标代码生成程序；
(7) 错误检查和处理程序；
(8) 各种信息表格的管理程序。

98. 路由协议有哪些？

100. 比特率波特率

比特率：在数字信道中，比特率是数字信号的传输速率，它用单位时间内传输的二进制代码的有效位(bit) 数来表示，其单位为每秒比特数 bit/s(bps)、每秒千比特数(Kbps)或每秒兆比特数(Mbps)来表示(此处 K 和 M 分别为 1000 和 1000000，而不是涉及计算机存储器容量时的 1024 和 1048576)。

波特率：波特率指数据信号对载波的调制速率，它用单位时间内载波调制状态改变次数来表示，其单位为 波特(Baud)。波特率与比特率的关系为：比特率=波特率 X 单个调制状态对应的二进制位数。

101. 网络服务质量包括哪些方面？

QoS（Quality of Service）服务质量，是网络的一种安全机制， 是用来解决网络延迟和阻塞等问题的一种 技术。 在正常情况下，如果网络只用于特定的无时间限制的应用系统，并不需要 QoS，比如 Web 应用， 或 E-mail 设置等。但是对关键应用和多媒体应用就十分必要。当网络过载或拥塞时，QoS 能确保重要业 务量不受延迟或丢弃，同时保证网络的高效运行。

QoS 关键指标： 可用性、吞吐量、时延、时延变化(包括抖动和漂移)和丢失 可用性：是当用户需要时网络即能工作的时间百分比。 吞吐量：是在一定时间段内对网上流量(或带宽)的度量。 时延：指一项服务从网络入口到出口的平均经过时间。 时延变化：是指同一业务流中不同分组所呈现的时延不同。 丢包：不管是比特丢失还是分组丢失，对分组数据业务的影响比对实时业务的影响都大。

102. 什么是信道

信道：表示向某一方向传送信息的媒体

103. 信道分类？

模拟信道：传送模拟信号的信道 数字信道：传送数字信号的信道

104. 什么是模拟信号？什么是数字信号？

模拟信号：连续信号，例如：话音信号和广播信号 数字信号：离散信号，二进制代码 0、1 组成的信号

105. 什么是基带信号？什么是宽带信号？

基带信号：将数字信号 1 或 0 直接用不同的电压来表示，然后送到电

路上去传输。

宽带信号：将基带信号调制后形成的频分复用模拟信号。由于基带信号经过调制，其频谱移动到较高的频 率处。由于每一路基带信号的频谱都被移动到不同的频段上，因此合在一起后并不会互相干扰，这样可以 在一条电缆中传送多路的数字信号，因而提高了线路的利用率。

106. java 与 C++区别他说他想问的是地址方面的.

1．指针 JAVA 语言让编程者无法找到指针来直接访问内存无指针，并且增添了自动的内存管理功能，从而有效地防 止了 c / c++语言中指针操作失误，如野指针所造成的系统崩溃。但也不是说 JAVA 没有指针，虚拟机内部 还是使用了指针，只是外人不得使用而已。这有利于 Java 程序的安全。

2．多重继承 c++支持多重继承，这是 c++的一个特征，它允许多父类派生一个类。尽管多重继承功能很强，但使用复杂， 而且会引起许多麻烦，编译程序实现它也很不容易。Java 不支持多重继承，但允许一个类继承多个接口 (extends+implement)，实现了 c++多重继承的功能，又避免了 c++中的多重继承实现方式带来的诸多不 便。

3．数据类型及类 Java 是完全面向对象的语言，所有函数和变量部必须是类的一部分。除了基本数据类型之外，其余的都作 为类对象，包括数组。对象将数据和方法结合起来，把它们封装在类中，这样每个对象都可实现自己的特 点和行为。而 c++允许将函数和变量定义为全局的。此外，Java 中取消了 c / c++中的结构和联合，消除了 不必要的麻烦。

4．自动内存管理 Java 程序中所有的对象都是用 new 操作符建立在内存堆栈上，这个操作符类似于 c++的 new 操作符。下面 的语句由一个建立了一个类 Read 的对象，然后调用该对象的 work 方法：Read r＝new Read(); r.work(); 语句 Read r＝new Read()；在堆栈结构上建立了一个 Read 的实例。Java 自动进行无用内存回收操作， 不需要程序员进行删除。而 c 十十中必须由程序贝释放内存资源，增加了程序设计者的负扔。Java 中当一 个对象不被再用到时，无用内存回收器将给它加上标签以示删除。JAVA 里无用内存回收程序是以线程方式 在后台运行的，利用空闲时间工作。

5．操作符重载 Java 不支持操作符重载。操作符重载被认为是 c 十十的突出特征，在 Java 中虽然类大体上可以实现这样 的功能，但操作符重载的方便性仍然丢失了不少。Java 语言不支持操作符重载是为了保持 Java 语言尽可 能简单。

6．预处理功能 Java 不支持预处理功能。c / c++在编译过程中都有一个

预编泽阶段，即众所周知的预处理器。预处理器为 开发人员提供了方便，但增加丁编译的复杂性。JAVA 虚拟机没有预处理器，但它提供的引入语句(import) 与 c 十十预处理器的功能类似。

7. Java 不支持缺省函数参数，而 c++支持 在 c 中，代码组织在函数中，函数可以访问程序的全局变量。c 十十增加了类，提供了类算法，该算法是与类相连的函数，c 十十类方法与 Java 类方法十分相似，然而，由于 c 十十仍然支持 c，所以不能阻止 c 十十开发人员使用函数，结果函数和方法混合使用使得程序比较混乱。 Java 没有函数，作为一个比 c 十十更纯的面向对象的语言，Java 强迫开发人员把所有例行程序包括在类中， 事实上，用方法实现例行程序可激励开发人员更好地组织编码。

8 字符串 c 和 c 十十不支持字符串变量，在 c 和 c 十十程序中使用 Null 终止符代表字符串的结束，在 Java 中字符 串是用类对象(strinR 和 stringBuffer)来实现的，这些类对象是 Java 语言的核心，用类对象实现字符串 有以下几个优点：(1)在整个系统中建立字符串和访问字符串元素的方法是一致的； (2)J3 阳字符串类是作为 Java 语言的一部分定义的，而不是作为外加的延伸部分； (3)Java 字符串执行运行时检空，可帮助排除一些运行时发生的错误； (4)可对字符串用“十”进行连接操作。

9“goto 语句 “可怕”的 goto 语句是 c 和 c++的“遗物”，它是该语言技术上的合法部分，引用 goto 语句引起了程序 结构的混乱，不易理解，goto 语句子要用于无条件转移子程序和多结构分支技术。鉴于以广理由，Java 不提供 goto 语句，它虽然指定 goto 作为关键字，但不支持它的使用，使程序简洁易读。

l0. 类型转换 在 c 和 c 十十中有时出现数据类型的隐含转换，这就涉及了自动强制类型转换问题。例如，在 c 十十中可 将一浮点值赋予整型变量，并去掉其尾数。Java 不支持 c 十十中的自动强制类型转换，如果需要，必须由 程序显式进行强制类型转换。

11.异常 JAVA 中的异常机制用于捕获例外事件，增强系统容错能力 try{ / / 可能产生例外的代码 }catch(exceptionType name){ //处理 } 其中 exceptionType 表示异常类型。而 C++则没有如此方便的机制。

107. 传送介质和无线网络协议

网络传输介质是网络中发送方与接收方之间的物理通路，它对网络的数据通信具有一定的影响。常用的传 输介质有：双绞线、同轴电缆、光纤、无线传输媒介。

108. 什么是 SHELL

在计算机科学中，Shell 俗称壳（用来区别于核），是指“提供使用者使用界面”的软件（命令解析器）。 它类似于 DOS 下的 command.com。它接收用户命令，然后调用相应的应用程序。同时它又是一种程序设 计语言。作为命令语言，它交互式解释和执行用户输入的命令或者自动地解释和执行预先设定好的一连串 的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高阶语言中才具有的控制结构， 包括循环和分支。

109. 网络里的时延、带宽

时延是指一个报文或分组从一个网络的一端传送到另一个端所需要的时间。它包括了发送时延，传播时延， 处理时延，排队时延。（时延=发送时延+传播时延+处理时延+排队时延）一般，发送时延与传播时延是我 们主要考虑的。对于报文长度较大的情况，发送时延是主要矛盾；报文长度较小的情况，传播时延是主要 矛盾。（计算机网络方面的时延概念） 计算机网络带宽：比特/秒 电信带宽：频带的两个端频率之间的差值

110. 网络拥塞

拥塞现象是指到达通信子网中某一部分的分组数量过多，使得该部分网络来不及处理，以致引起这部分乃 到整个网络性能下降的现象，严重时甚至会导致网络通信业务陷入停顿即出现死锁现象。 拥塞产生原因： （1）多条流入线路有分组到达，并需要同一输出线路，此时，如果路由器没有足够的内存来存放所有 分组，那么有的分组就会丢失。 （2）路由器的慢速处理器的缘故，以至于难以完成必要的处理工作（如缓冲区排队、更新路由表等）。那么，即使有多余的线路容量，分组也需要进入到队列中。 防止拥塞的方法：（1）传输层可采用：重传策略、乱序缓存策略、确认策略、流控制策略和确定超时策略。 （2）网络层可采用：子网内部的虚电路与数据报策略、分组排队和服务策略、分组丢弃策略、路由算 法和分组生存管理。 （3）数据链路层可采用：重传策略、乱序缓存策略、确认策略和流控制策略。

111. CSMA/CD CSMA/CD CSMA/CD CSMA/CD 的原理

CSMA/CD（Carrier Sense Multiple Access/Collision Detect）即载波监听多路访问/冲突检测方法 在以太 网中，所有的节点共享传输介质。如何保证传输介质有序、高效地为许多节点提供传输服务，就是以太网 的介质访问控制协议要解决的问题。 CSMA/CD 是一种争用型的介质访问控制协议。它起源

于美国夏威夷大学开发的 ALOHA 网所采用的争用 型协议，并进行了改进，使之具有比 ALOHA 协议更高的介质利用率。 CSMA/CD 应用在 OSI 的第二层 数据链路层 工作原理：发送数据前，先侦听信道是否空闲，若空闲，则立即发送数据，在发送数据时，边发送边继续 侦听，若侦听到冲突，则立即停止发送数据，等待一段随机时间，再重新尝试。 先听后发，边发边听，冲突停发，随机延迟后重发。

112. 数据缓存 cache 的基本概念

高速缓冲存储器（Cache）其原始意义是指存取速度比一般随机存取记忆体（RAM）来得快的一种 RAM， 一般而言它不像系统主记忆体那样使用 DRAM 技术，而使用昂贵但较快速的 SRAM 技术，也有快取记忆 体的名称。 高速缓冲存储器是存在于主存与 CPU 之间的一级存储器， 由静态存储芯片(SRAM)组成，容量比较小但 速度比主存高得多， 接近于 CPU 的速度。 主要由三大部分组成： Cache 存储体：存放由主存调入的指令与数据块。 地址转换部件：建立目录表以实现主存地址到缓存地址的转换。 替换部件：在缓存已满时按一定策略进行数据块替换，并修改地址转换部件。[1]

113. 应用层有什么协议，作用 简单电子邮件传输（SMTP）、文件传输协议（FTP）、网络远程访问协议（TELNET）、DNS

114. 网络各层的设备是什么和工作原理

传输层：四层交换机、也有工作在四层的路由器 网络层：路由器、三层交换机 数据链路层：网桥、以太网交换机（二层交换机）、网卡（其实网卡是一半工作在物理 层、一半工作在数据链路层） 物理层：中继器、集线器、还有我们通常说的双绞线也工作在物理层

115. 传统的搜索引擎基本原理？
基于内容的搜索？原理及实现？

117. 数据传输方式
串行传输 并行传输

118. 数据链路层有哪些协议，举 1~2 例

PPP 协议：当前世界使用最多的数据链路层协议 SLIP 协议：
HDLC

119. 电路交换和分组交换的区别及联系

电路交换： 1、 基于位置，即在某一位置的比特经交换后变更到另一个位置上。 2、 面向连接的，电路交换必定是面向连接的，但面向连接的却不一定是电路交换。 3、 通话过程中始终占用端到端的固定传输带宽。

分组交换： 1、 采用存储转发技术 2、 基于标记的，将欲发送的整块数据称为报文(message)，将较长的报文划分为一个个更小 等长数据段（比如 1024bit），在每个数据段前面加上首部(header)，后就构成一个分组(packet)，分组又称为包，分组的头部称为包头。分组首部包含了目的地址和源地址等重 要的控制信息，每一个分组才能在分组交换网中独立地选择路由。 3、 无连接的

120. 电路交换、报文交换、分组交换主要的区别

1、 电路交换，需要先建立链接，才能进行通信。而报文交换和分组交换不需要先建立链接。

2、 若连续传送大量数据，传送时间大于链路建立时间，比较适合用电路交换。

3、 报文交换和分组交换不需要预先分配传输带宽，在传送突发数据时，可提高整个网络的 信道利用率。

4、 分组交换比报文交换时延小，但结点交换机必须具备更强的处理能力。

122. CDMA 全称及原理

CDMA：Code Division Multiple Access，码分复用。 抗干扰性强，信号类似白噪声，用于军事，现在也用于民用。 原理： 码片：每个比特时间再划分为 m 个短的间隔，称为码片。 码片序列： CDMA 系统中，每个站被指定一个 m bit 的码片序列，当站点用发送 1 时，就发送自己的码片序列，当要 发送 0 时，就发送自己码片序列的二进制反码。CDMA 系统中的各个站点要求码片序列满足正交性，规格 化内积为 1，且一个码片向量和该码片反码向量的内积为-1。 若 A 站要接受 B 站的信息，首先 A 站需要知道 B 站的码片序列，A 站将此码片序列和接收到的码片序列 进行内积，当非 B 站发来的码片序列，内积全为 0，全部过滤掉了。剩下的内积为 1 或-1 的

信息，就是 B 发给 A 的信息。

123. ICMP

ICMP 是（Internet Control Message Protocol）Internet 控制报文协议。它是 TCP/IP 协议族的一个子协议， 用于在 IP 主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网 络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。 ICMP 协议是一种面向无连接的协议，用于传输出错报告控制信息。它是一个非常重要的协议，它对 于网络安全具有极其重要的意义。我们在网络中经常会使用到 ICMP 协议，比如我们经常使用的用于检查网络通不通的 Ping 命令（Linux 和 Windows 中均有），这个“Ping”的过程实际上就是 ICMP 协议工作的过程。还有其他的网络命令如跟踪 路由的 Tracert 命令也是基于 ICMP 协议的。

124. 问我什么是非对称加密？什么是数据安全的特征？
（信息安全专业请重视安全方面的问题）

1976 年，美国学者 Dime 和 Henman 为解决信息公开传送和密钥管理问题，提出一种新的密钥交换协议， 允许在不安全的媒体上的通讯双方交换信息，安全地达成一致的密钥，这就是“公开密钥系统”。相对于“对 称加密算法”这种方法也叫做“非对称加密算法”。 与对称加密算法不同，非对称加密算法需要两个密钥：公开密钥（publickey）和私有密钥（privatekey）。 公开密钥与私有密钥是一对，如果用公开密钥对数据进行加密，只有用对应的私有密钥才能解密；如果用 私有密钥对数据进行加密，那么只有用对应的公开密钥才能解密。因为加密和解密使用的是两个不同的密 钥，所以这种算法叫作非对称加密算法。

125. 保护频带

在给定信道的上下限处留出的未占用窄频带。其目的是确保信道间有足够隔离，防止相邻信道干扰。

126. 问到 PPP 协议

点对点协议（PPP）为在点对点连接上传输多协议数据包提供了一个标准方法。PPP 最初设计是为两 个对等节点之间的 IP 流量传输提供一种封装协议。在 TCP-IP 协议集中它是一种用来同步调制连接的数 据链路层协

议（OSI 模式中的第二层），替代了原来非标准的第二层协议，即 SLIP。除了 IP 以外 PPP 还 可以携带其它协议，包括 DECnet 和 Novell 的 Internet 网包交换（IPX）。 PPP 协议是一种点——点串行通信协议。PPP 具有处理错误检测、支持多个协议、允许在连接时刻协 商 IP 地址、允许身份认证等功能，还有其他。PPP 提供了 3 类功能：成帧；链路控制协议 LCP；网络控制 协议 NCP。PPP 是面向字符类型的协议。

127. 流量控制在哪些层实现

流量控制：实质就是限制发送方的数据流量，使其发送的速率不要超过接收方处理的速率。

1、 传输层：端到端的流量控制

2、 数据链路层：相邻结点间的流量控制

128. 二层交换机是哪一层的设备，与三层交换机之间的区别？

二层交换技术是发展比较成熟，二层交换机属数据链路层设备，可以识别数据包中的 MAC 地址信息，根据 MAC 地址进行转发，并将这些 MAC 地址与对应的端口记录在自己内部的 一个地址表中。 三层交换机就是具有部分路由器功能的交换机 三层交换技术就是二层交换技术＋三层转发技术。传统交换技术是在 OSI 网络标准模型第 二层——数据链路层进行操作的，而三层交换技术是在网络模型中的第三层实现了数据包的 高速转发，既可实现网络路由功能，又可根据不同网络状况做到最优网络性能。

129. 三网，指哪三网

电信网络：主要业务是电话，传真等。 有线电视网络：单向电视节目传输网络 计算机网络：我们现在用的很多的局域网和 Internet 等。

130. 分组交换的优点及缺点

优点： 1、 高效：逐段占用通信链路，动态分配资源。 2、 迅速：通信前，不用首先建立链接，可以直接发送。 3、 可靠：完善的网络协议；分布式路由 4、 灵活：每个分组可以独立路由

缺点： 1、时延：分组在各个结点存储转发时因要排队，造成一定的时间延迟。 2、开销：因为包头要携带控制信息，造成一定的网络开销。

131. 组成网络协议的三个要素

1、 语法：数据和控制信息的结构或格式

2、 语意：需要发出何种控制信息，完成何种动作以及做出何种应答。

3、 同步：事件实现顺序的详细说明

132. DNS and DHCP

DNS 是计算机域名系统 (Domain Name System 或 Domain Name Service) 的缩写，它是由解析器和域 名服务器组成的。域名服务器是指保存有该网络中所有主机的域名和对应 IP 地址，并具有将域名转换为 IP 地址功能的服务器。

动态主机设置协议（Dynamic Host Configuration Protocol, DHCP）是一个局域网的网络协议，使用 UDP 协议工作，主要有两个用途：给内部网络或网络服务供应商自动分配 IP 地址，给用户或者内部网络管理员 作为对所有计算机作中央管理的手段。

133. 网络安全有哪些方面

网络安全是指网络系统的硬件、软件及其系统中的数据受到保护，不因偶然的或者恶意的原 因而遭受到破坏、更改、泄露，系统连续可靠正常地运行，网络服务不中断。 网络安全从 其本质上来讲就是网络上的信息安全。从广义来说，凡是涉及到网络上信息的保密性、完整 性、可用性、真实性和可控性的相关技术和理论都是网络安全的研究领域。网络安全是一门 涉及计算机科学、网络技术、通信技术、密码技术、信息安全技术、应用数学、数论、信息 论等多种学科的综合性学科。

134. P2P 协议

135. 停止等待协议

136. ipv4 地址匮乏解决办法

IP 地址的编址共经过了三个基本阶段：分类的 IP 地址、子网的划分、构造超网 分类的 IP 地址二层结构：IP 地址::={,} 子网划分的三层结构 IP 地址：IP 地址::{,,} CIDR 地址：IP 地址::{, } 子网划分：在 ip 地址中添加一个“子网号字段”，使两级的 ip 地址变成三级的 ip 地址，这种做法叫做划分 子网。 VPN 一般采用隧道技术 A：1~126 B：128~191 C：192~223 D：224~239 E：240~255

1、IPv6：根本解决办法。 ipv4:32 位 ipv6:128 位 IPv6 分组由基本首部，扩展首部，数据三部分构成。加快了路由器处理数据报的速度。 IPv6 数据报的目的地址可以是以下三种基本类型的地址之一 单播(unicast)：传统的点对点通信 组播(multicast)：一点对多点的通信 任播(anyeast)：IPv6 新增类型，任播的目的站是一组计算机，但数据报在交付时只交付给其中一个，通常 是距离最近的一个。

2、NAT 技术：将专用网络内部使用的本地 ip 地址转换成有效的外部使用的全球 ip 地址，使得整个专用网 只需要一个全球 ip 地址就可以与英特网联通。此方法可以解决 ip 地址紧缺的问题。 静态 NAT：将内部网络中的每个主机都永久映射成外部外部网络中的某个合法的地址。 动态地址：在外部网络中定义了一系列的合法地址，采用动态分配的方法映射到内部网络。 网络地址端口转换：把内部地址映射到外部网络的一个 ip 地址的不同端口上。

3、CIDR：Classless Inter-Domain Routing，消除了传统的 A 类、B 类和 C 类、地址以及划分子网的概念， 因而可以更加有效地分配 IPv4 的地址空间。这是一种为解决地址耗尽而提出的一种措施。 最长前缀匹配==最长匹配==最佳匹配：网络前缀越长，其地址块就越小，因为路由就越具体。

137. 单工、半双工、全双工

单工：单向通信，只能有一个方向的通信，而没有反方向的交互。（例如：无线/有线电广播，电视广播）

半双工：双向交替通信，通信的双方都可以发送信息，但不能同时发送，也不能同时接收。一方发送一方 接收，多段时间，再反过来。

全双工：双向同时通信，通信的双方可以同时发送和接收信息。 半双工和全双工通信都需要两条信道。

138. TCP/IP 模型分层 OSI 中的七层，每层完成的功能

139. IPv4 和 IPv6 怎么相互通信?

从 IPv4 向 IPv6 过渡可以采用双协议栈和隧道技术

双协议栈：在完全过渡到 IPv6 之前，使一部分主机（或路由器）装有两个协议栈。即一个 IPv4 和一个 IPv6， 通过双协议栈进行转换。

隧道技术：是将整个 IPv6 数据报封装到 IPv4 数据报的数据部分，这样，使得 IPv6 数据报可以在 IPv4 网 络的隧道中传输。

140. IPv4 的替代方案

IPv4 的替代方案是 IPv6，IPv6 共 128 位。 相对 IPv4 的主要改进

1、 地址变长了，从 32 位变成了 128 位

2、 简化了 IP 分组的基本首部，包含 8 个段（IPv4 包含 12 个段），这一改变使得路由器可以尽快地处理 分组，从而可以改善吞吐率。

3、 IPv6 更好地支持选项，这一特征加快了分组处理速度。 IPv6 分组由基本首部，扩展首部，数据三部分构成。加快了路由器处理数据报的速度。

IPv6 数据报的目的地址可以是以下三种基本类型的地址之一

单播(unicast)：传统的点对点通信

组播(multicast)：一点对多点的通信

任播(anyeast)：IPv6 新增类型，任播的目的站是一组计算机，但数据报在交付时只交付给其中一个，通常 是距离最近的一个。

IPv6 的地址采用冒号分隔每个 16 位，3CD4:BA23:2254:GFFA:CB4D:674D:DFA5:2345

141. 从网络的作用范围进行分类

1、 广域网： 2、 局域网： 3、 城域网：作用范围在局域网和广域网之间

142. 从网络的使用范围分类

1、 公用网：一般是国家的邮电部门建造的网络

2、 专用网：一般一个单位专用的网络

143. 从网络的拓扑结构分类

1、 星型网络 2、 网状网络 3、 总线网络 4、 令牌环网络 5、 树形网络

144. 信道划分，及其典型应用

信道根据频率来划分

145. 复用的相关概念（频分、时分、码分等）

FDM：Frequency Division Multiplexing， 频分复用

TDM：Time Division Multiplexing，时分复用（同步时分复用），有利于数字信号传输。

STDM：Statistical Time Division Multiplexing，统计时分复用（异步时分复用）

DWDM：Dense Wavelength Division Multiplexing，密集波分复用

CDMA：Code Division Multi Access，码分多址

1、 频分复用的所有用户，在同样的时间占用不同的带宽资源。 2、 时分复用的所有用户，在不同的时间占用同样的频带宽度。 3、 码分多址，每个用户在同一时间使用同样的频带进行通信。 4、 波分复用就是光的频分复用

146. 计算机网络中使用的信道共享技术有哪些？

1、 随机接入 2、 受控接入 3、 信道复用

常用的信道复用（multiplexing）技术

FDM：Frequency Division Multiplexing， 频分复用

TDM：Time Division Multiplexing，时分复用（同步时分复用），有利于数字信号传输。

STDM：Statistical Time Division Multiplexing，统计时分复用（异步时分复用）

DWDM：Dense Wavelength Division Multiplexing，密集波分复用

CDMA：Code Division Multi Access，码分多址

5、 频分复用的所有用户，在同样的时间占用不同的带宽资源。

6、 时分复用的所有用户，在不同的时间占用同样的频带宽度。

7、 码分多址，每个用户在同一时间使用同样的频带进行通信。

8、 波分复用就是光的频分复用 频分复用和时分复用技术比较成熟，缺点是不灵活。 TDM 帧和 STDM 帧都是在物理层中传送的比特流中所划分的帧。这种帧和我们在数据链路 层讨论的帧是完全不同的概念，不可混淆。

软院生活学习总结

本部分共收录五篇学长学姐的研一生活学习总结，其中有四篇来自苏州的学习生活总结，有一篇来自合肥的学习生活总结。该部分内 容旨在帮助你了解科大研一的生活、学习、活动、实习，提前摆脱不适感，在机会来临前，做好相应的准备，祝各位在科大度过一个充实 愉快的研一。

5.1 合肥篇

你在桥上看风景，看风景的人在看你

距离 2018 年考研快过去整整一年，来到肥科也快一学期了，回想合肥学习这半年，学霸过，也学渣过，差点挂过科，却也始终渴望 着拿奖学金。这里没有繁华，没有喧嚣，妹子不多，基友很多，总之， 是个能够静下心来学习的好地方。。。。。。 “合肥，居皖之中，可为省会。”这个地方说实话是挺尴尬，二 线的城市，三线的工资，一线的房价。想必报考科软的多为安徽学子， 这里对合肥的介绍就简而言之了。土肥没啥玩的，但一般城市该有的都有，冬暖夏凉，四季分明，所以外地外省的小伙伴不用太担心这里 的生活。下面首先重点谈谈位于中科大南区的软院生活情况。

中科大在合肥一共有（东西南北）四个主要校区，无论生活还是 学习南区跟东西这两个主校区肯定是都不能比的。西区靠近安大老区 （你懂得）和繁华的三里庵商圈，科大的计算机学院也就位于西区。 东区有免费的健身房（周一闭馆），而且，科大 99%的活动都在东西 区举办。（比如说每年双十一著名的“美丽邂逅”（相亲大会）） 相比 之下，南区会显得有些简陋，换句话说，有点不像是科大亲生的。。。。。

在南区，主要是软件学院，MBA 管理学院以及一些中科院的代 培生。除了我们这帮叔叔阿姨之外，每天见的最多的就是在南区生活 的那帮科大附小附中的“侠们”。（合肥话，小屁孩的意思）如果早上 没课的话，周一到中午吃饭一定要赶在 12 点之前，否则，食堂里充 满了一帮在讨论王者荣耀的小学生，连个坐的地方都没得。 南区的食堂还是很给力的，饭菜普遍不贵，一般 6，7 元吃饱， 9,10 元吃好。但是饭菜的种类比较单一，不如东西区那么丰富。反正吃腻了食堂还是可以叫外卖的，周围各式各样的小餐馆多得很。印象中一般大学附近都有个小吃街吧，很可惜，南区这没有。附近到了晚 上就有几家烧烤，有个叫罍（lei）街的小吃一条街还不错，就是有那么丢丢贵那么丢丢远。 说完了吃我们再谈谈住行。

南区的宿舍**四人间**，有**空调有暖气**， 缺点在我看来就是**没有独立卫生间**，不过这样一来不用自己打扫厕所 了也还是蛮不错。洗澡的话每一层都有个公共浴室，有隔间，有门， 不用担心捡肥皂，另外在食堂后面还有一个澡

堂，总之洗澡也挺方便。 南区的体育设施还是不错的，篮球场的筐基本都有球网这点真心赞，羽毛球馆乒乓球台也都有室内的，足球场跑道啥的自然也不用说。 缺点是没有网球场，没有健身房，想要健身的小哥哥小仙女只能去东 区了，不过也不是很远，4.5 公里左右，骑个小黄车十多分钟就到了。

科大的校车也很频繁，基本一两个小时内在三个校区间都有来往。南 区面积不大，不过小也有小的好吧，下楼就是食堂，走五分钟就是图 书馆和教学楼，校园内出门基本靠走，交流基本靠吼。这里再夸下科 大合肥这边的网络通（WIFI），一次可以连接十个设备，4 号线日常上 网，9 号线下载一步电影分分中，而且，钱可以拖着毕业再交！在校 园内还可以连上谷歌，都省去了翻墙的麻烦。 说完了吃住行，下面回归主题，谈谈科软合肥的学习生活。合肥的选课每个人都是学校固定的，不用像在苏州那样还得抢， 看群里感觉苏州选课就跟打仗一样，这点觉得合肥还是好很多。不过 选课那天大家还是早点选，不然个别同学还是会选不到的。这里贴张 合肥的课表，大家考上后可以看着先自己学学，尤其是跨考生，不然 到时候很累！我再来给大家介绍介绍几位上课老师的特点。 石竹老师：比较幽默，但是幽默属性比较硬，上课风格严谨中夹 带着风趣气息，喜欢开玩笑，但大家往往不笑。。。。。。石老师真的是 好人，考试一般都是给过的那种； 丁箐老师：丁老师的，一杯茶一把椅，谈笑风生，他的课感觉挺 难，偏架构方向，感觉没个一两年相关的工作经验很难真正听懂，班 上的人数也是随着课程的推进逐渐减少。。。。。有一天早上貌似只有 十多个人； 徐云老师：大佬，真的是大佬。计算机学院过来的老师，《算法 导论》的翻译者之一，算法教的很好很负责，人也很风趣，不过考试 有点难吧，给分普遍不高； 白天老师：靠谱，很靠谱。白老师上课真的很认真负责，以前貌 似在网易工作过，讲的课还是比较接地气，老师人也非常好。 李金龙老师：也是计算机学院那边过来的，教的数据挖掘，课程 讲的还是不错，涉及到了一些人工智能的概念。 写到此，也回想到了自己去年从初试考完后的失落（当时估分 290+，结果 340+，欧气满满有木有），到看到分数后的喜悦，再到准 备复试阶段时候的不安，再到复试结果出来后的欣慰。

每个考研人都 是不容易，都是有梦想的，在最后，送大家一首苏轼的定风波： 莫听穿林打叶声，何妨吟啸且徐行。竹杖芒鞋轻胜马，谁怕？一 蓑烟雨任平生。 料峭春风吹酒醒，微冷，山头斜照却相迎。回首向来萧瑟处，归去， 也无风雨也无晴。 如今，你们已经走到了复试这一阶段，离最后的胜利也不远了。 这一路，也许会有许多旁人的说笑，也许会有对自己选择的怀疑，也 许会有对未来的迷茫。此词作于苏轼黄州之贬后的第三个春天，此

时 的你们离春天也不远了，愿你们能像词中所说“莫听穿林打叶声”不 要太关注外界的声音，坚持自己的初心，待到来年山花烂漫时，你们 也能“回首向来萧瑟处，归去，也无风雨也无晴。” 最后的最后，还是祝福各位都能实现自己的梦想吧！（此处双手 抱头举过头顶）

5.2 苏州篇（1）

苏州科软杂谈

首先恭喜学弟学妹们，当你们看到这篇文档的时候应该已经顺利地过了初试，到了复试阶段。来苏州三个多月了，想要写下一些有关 苏州科软的故事。 学校里面大部分是软院的学生，也有纳米学院和 MBA 管理学院。 校园不大，但风景很好，小桥流水，杨柳依依。建筑相对旁边的人大有点旧，但设备很完善，有健身房、食堂、图书馆等等，健身房虽然 不是免费的，不过办卡很便宜，如果不是经济特别紧张的话建议办一 张，毕竟适当的健身可以释放压力有助于学习；食堂吃饭的话 8-10 元 可以吃饱，10-15 元可以吃好；宿舍不在学校里面，校园里除了教学楼就是实验室，大部分同学下完课会呆在实验室继续学习，晚上再搭 乘公交车回宿舍（宿舍距离学校有一段路程，需要乘坐公交车。）二 人间的宿舍有独卫、热水器和空调，很安静，也有人选择在宿舍学习。 考上之后选课一定是大家最关心的问题，课程完全由设定，所以 建议提前做好规划，好课往往都是僧多粥少，如果不提前做好规划很 可能会选不上心仪的课程。

在这里学习说实话压力挺大，整个学习节 奏是紧张的，真正在校的有效学习时间是一年，但是科大把这一年当 成两年来用，每个学期分为上半学期和下半学期，每个小学期结束都 会有期末考试，有些课还有期中考试。所以如果你想好好学习的话还 是非常忙碌的。当然，这里的大神也很多，有不懂的问题随时可以问。 希望大家珍惜时间好好把握。 再给大家谈谈我遇到的几位老师 吴俊敏老师很儒雅，课上干货很多，计算机学院过来的老师，上他的课需要提前预习，老师讲的很快，一不小心就听不懂了。不过老 师还是很 nice 的，考试不会为难你。

郭艳老师的课挺值得选，老师很可爱，学生们都喜欢她，她的课 互动性很强，课堂氛围非常好，上课会带你现场操作，同学们对她的 实验课积极性非常高，经常有人做实验到十一点都不肯离开实验室， 上过他的信息安全，很值得选，与之对应的还有信息安全实践，据说 反响也不错。

朱洪军老师是一个特别严谨的人，教的课程也多，有 Andriod、JavaEE、软件设计模式，他的课任务多，作业一定要按照他的方式来， 要不然就没有

分数，所以选了他的课的人基本上会很辛苦，但是上完 了就会发现学到了很多东西，基本上在目前遇到的老师当中，朱洪军老师的课上学到的东西是最多的。如果你不怕累倒是可以试试。

赵振刚老师是一个非常认真的人，上课偶尔会讲笑话活跃气氛， 老师从来不用助教，班上一百多人上实验课，他一个一个检查，而且问题都问的很仔细，同学们都佩服他。在他的课上也能学到很多东西， 如果你选嵌入式方向，可以考虑报他的工程实践。

汪炀老师是从计算机学院过来的重量级老师，副院长，教的是新 开的课程数据挖掘（也就是现在很火的机器学习的前身）汪老师上课 很富有激情，不容易走神，板书很多，任务量适中，总体不错。 当然，除了学习，学校偶尔也会有活动，“迎新晚会”，“姑苏行， 三校情”是两个参与度很高的活动。“迎新晚会”主要是认识自己学 校的人，自己组织活动欢迎一下自己。“姑苏行，三校情”是科大、 人大、东南三校联谊，会有很多小哥哥小姐姐参加，主要是认识外校 的人。 以上差不多就是苏州科软的生活啦，最后的最后，祝愿各位学弟 学妹们达成所愿，前程似锦！

5.3 苏州篇（2）

中科大软院的生活

从去年八月份来苏州，到现在也差不多快一年了吧，时间过 的可真快。今天特意写个博文来记录下这大半年的情况吧。木有啥文 笔，想到什么说什么吧。

首先挺感谢科大软件学院，在我考研失败的情况下收留了我，中 科大的牌子提供了很多机会，感谢这个平台。有些人觉得软院收这么 多人没有导师比较水，其实我觉得主要看自己，有利也有弊吧。我自 己而言的话倒是觉得这样挺好的，没人管我可以做自己喜欢的事情 （玩=。=），反正没有学术追求，第二年可以出去实习跟给老板干活 差不多嘛。如果想配导师的话可以选择联合培养，貌似有奖学金免住 宿费。如果有学术追求的话，情况我就不太了解了。总之，中科大的 牌子是好使的~ 生活方面，我是挺喜欢苏州这个城市的，以后还挺想在苏州定居 的(￣▽￣)，不过貌似园区房价涨的有点快啊==虽然涨不涨都买不起， 说不定以后回家养猪呢。。学校这边环境蛮好的，宿舍在文汇，坐公 交要好几站，走着快的话要半个多小时吧。每天晚上十点多快 2 都 是科大的学霸们。。每天晚上十点半文汇的小摊小贩们准时出来，各 种垃圾食品应有尽有。学校这边思贤楼有个食堂，难吃的一米。旁边 有个诺亚餐厅，除了黄焖鸡米饭就是砂锅，吃吐了。 还有个巴菲，吃不起。。。

文星那边吃的比较多~思贤楼一楼有个健 身房，免费的，不过貌似里面没啥可以用的器械了=。=不过可以打打 乒乓球~明德楼旁边有个健身房，办个学生年卡也不是很贵，和文星 那边的通用，还可以（我这里还有一张卡 8 月份到期，有想要的我 送他了）。想游泳的话，附近有个独墅湖游泳馆，可以办学生次卡， 600 元 30 次。吃过晚饭去独墅湖溜达溜达也是 nice 的，学校后面 有个土地庙我最近才发现，好神奇(￣▽￣)。。旁边的人大居然还养了 孔雀，听说过一段时间要引进梅花鹿。。。动物园么。。。当然妹子也多， 虽然我不看~ 学校的活动还是蛮丰富的，迎新晚会，这种帅哥靓女(￣▽￣)，和其他学校的联（相）谊（亲）以及院篮球赛，吊的飞起的嵌入式 1 班啊哈哈哈哈~~研究生会自己组织的活动也挺多的， 反正没事多玩玩嘛，开心最重要~ 学习方面，我是嵌入式专业的，貌似很多同学是从软设调剂过来 的，不过从就业情况来说，还是看自己咯，转行要趁早！

嵌入式的话 就业方向一般有两个：嵌入式 linux，比如内核，驱动，BSP 等；底 层芯片设计，FPGA，EDA 等。课程的话，我自己比较懒，几乎没有去旁听过其他的课。 关于选课，我比较推荐李春杰老师的课，李老师慷慨激昂，重要 的事情说八遍=。=还教我们做人的道理，真是赚呢~~（Ps:有同学说李 老师的课有点坑，怎么说呢，从考试看的话貌似分数的确会偏低一点，但是我自身的话确实收获不小，如果想搞嵌入式 linux 的话，上完李 老师的几门课会挺有帮助的~）另外一门比较推荐的课是程序设计与 计算机系统，这门课以经典的 CSAPP 作为教材，还有，一定要好好 做下实验！收获会很大！听说华保健老师的课也不错，可惜我太懒了 没去听过。当然，也有一些水课，上课念 PPT，讲的也差，还堂堂课 点名=。=工程实践的话，如果你本科不是相关专业，没做过什么项目 的话，一定要好好做！一定要好好做！一定要好好做！说三遍。工程 实践抱个大腿是重要的，如果你自制力差，找个负责任的老师起码能 每周找你开会也是好事。

实习方面，嵌入式方向的实习相对软设确实会少一点，但是机会 还是很多的。像 Intel，IBM，Cisco，Qualcomm，Marvell，NXP，NI， Tplink，华为，计算所，电子所等等都有嵌入式的岗位。首先是简历 的问题，简历挺重要的，早点准备好，别人家都招聘了你才写。简历 改个十几遍不为过，最好准备份英文简历。没有项目经验的把工程实 践好好做。时间的话，我是三月份写的简历，主要四五月份找的。可 以去各大高校 BBS，大街拉钩等网站获得招聘信息，以及你的各种师 兄师姐七大姑八大姨资源==。学校五月份会有个实习招聘周，会有很 多企业来招聘，我没参加但是听说今年好公司比较少。 说说我自己的情况吧（我是学渣，仅供参考==）我本科是学物理 的，

跨专业，投的主要是设备驱动，bsp 方向的实习。其实之前还是 挺没底的，一是自己比较渣，二是没有特别拿的出手的项目经验，但 是最后挺幸运，拿到了目标几家公司的 offer，岗位也是比较满意的。

总结下的话两方面： 1.目标要明确，尽早准备 2.基础要扎实。我面试问的题目都不是 很难，70%都是基础的东西。很多都想不起来了，说说大概吧：1. C 是 必须掌握的技能，语法这种东西记住大家都会用，还要理解更深层次 的东西。 2.硬件相关：iic spi pci 总线啦，uart 啦，中断啦，指令集啦等等 微机原理的知识，以及 ARM 的基本知识。 3.操作系统工作原理，linux 内核的一些机制和原理要理解 4.linux 驱动程序原理，相关的一些机制，如自旋锁互斥锁，tasklet， 内核中断等等 5.linux 环境要熟悉，基本的命令，vim，makefile 等等。 6.基本的算法，链表二叉树什么的，写代码要注意鲁棒性！ 7.项目经验，一般都是从项目开始聊起的，所以一定要把你的项 目做了啥搞清楚，别吹牛逼，给自己挖坑会死的很惨。。。。 8.一般面试官会看重解决问题的思路，如果遇到不了解的问题， 也要试着解决，根据自己对已有知识的了解大致推断出解决思路。 9.招聘信息里都会有要求，如果你不了解的话可以提前学习下。 10.拿到 offer 一定要问清楚自己要做什么，不喜欢的话就不要去 了。。大概就是这么多吧，暂时想不起来了，想起来再补充。

5.5 苏州篇（3）

苏州学习生活有感

今天 12 月 10 日，苏州天气晴朗，但是丝毫感受不到暖意，这里的冷是冷的无处可藏，周末待在文汇公寓的我，穿的厚厚的长款棉袄 坐在电脑前瑟瑟发抖。在电脑前码着字上着网，看到今天是国考的日 子，一组数据是我有些惶恐不安，2.6 万的岗位，166 万人报考，几 乎要有 164 万人要被淘汰，心情复杂，现实的残酷犹如千军万马过独 木桥，要想出人头地，除了变得更强别无选择。

这样的不安和惶恐让 我想起了我的考研之路，去年的今天，距离考研不到俩周的时间，每 天重复几件事情，背英语作文，背专业课错题，背政治….这样的重复 让我产生了复习的倦怠，那时心里想着无论结果如何都不想再考了， 只想快快结束这样暗无天日的生活，虽然每次和家人同学通话都会吐槽一番，但是行动上却不敢有丝毫怠慢，每天早晨依然七点起床去图 书馆，晚上十点回去睡觉，我庆幸我没有在最后因为情绪的不稳定而 放弃考研，我依然坚持了下来。是坚持和信念让我来了科大，来到了 苏州这个美丽的城市。 复试结果出来之后，科大让我们选择方向和要去的城市，我没有 犹豫的选择了苏州，一直想要感受一下苏杭的天堂般的生活。依稀记 得和父母一

起在 9 月 5 号来科大苏州研究院报道，火车缓缓的驶进了 苏州站，怀着欣喜和激动的心情迫不及待的出了站，苏州火车站还真 是漂亮，但是天公不作美，阴沉的乌云伴着淅淅沥沥的小雨作为苏州 欢迎我们到来的方式。

这样的欢迎方式几乎持续了整个九月，对于我 这样一个北方汉子，缠缠绵绵的小雨还是多少有些不适用。之后我们 坐着快线 2 号公共交通到达了科大苏州研究院，没有想像中的大操场 和漂亮的图书馆，但是有一条河流穿过校园，将仅有的几座教学楼连 接在了一起，再加上校园中树木花草交相辉映，一个小巧精致的江南 园林展现我的眼前。这种婉约含蓄的美，不身在其中是无法感受到的。 在学校报到完之后，我们又坐上公车去了文汇公寓，人民大学的同学， 东南大学研究生院的同学也在这里住，我们是俩人一间，感觉和本科 完全不一样，自己需要独立面对很多事情。我选宿舍选在了二层的南 北方向的一间房间，想说一下用户感受，二层比较潮，南北方向常年 不见太阳就更潮了，不建议选楼层低阴面。

接下来，我们去了文汇公 寓旁的配套餐厅吃饭，米饭菜，这应该是全部大学食堂都有的食物， 这里的菜种类还是蛮多的，当时还觉得很好吃而且又便宜，俩个菜加 一份米 9，汤还免费喝，简直就是棒棒哒啊，但是吃过不久之后， 发现所有菜都是一个味道，于是，爱上了科大校园里面的餐厅，价格一样便宜，但是质量很好，唯一的缺点就是给的少，不过，对于我这 个资深吃货，到现在三个月，这些已经都不能满足我了，我现在只想 吃家乡的面，但是我还没有发现一个好吃的山西面馆，只能用兰州拉 面来满足我的胃了，一说到吃，刚才营造出的艺术气氛都消失了，文 笔质量直线下降，哈哈，各位读者见谅。 不知不觉来了苏州三个月了，很少有时间去城里玩耍，守在独墅 湖旁学习，还是别有一番情趣，苏州是一座古城，有着悠久的历史， 无论是人文景观还是自然景观都是值得一去的，希望学弟学妹们考上 之后，来发现苏州的美。 接下来，讲讲科大的具体的学习生活和相关老师的情况。快开学那会要进行选课，学弟学妹们一定要重视选课，当时开放 选课的时候，我正在火车上，然而。。。想必大家都猜到了结局，那个 网连系统都进不去，最后上去只能选一些别人选剩下的课，这点苏州 和合肥有很大的区别，虽然合肥也要选课，但是合肥的学生比苏州的 学生要少，他们基本都可以选到想选的课。 我最后选到了信息安全，信息安全实践，程序设计与计算机系统， 组合数学，设备驱动程序设计，设计模式。（我是软件设计方向的， 选了一堆其他方向的课）， 后来在上的过程中，我感觉我选的这几门课的老师都挺不错的。

特别是信息安全和信息安全实践课的郭燕老师，没有讲一些高深 的理论，从实践和例子着手，使得上课的气氛特别好，郭老师的攻防 案例成功的吸引了我们，虽然大多数的案例都很老了，但是不妨碍我 们在课堂的积极性。最后关键的是，老师考试划的重点在卷子上全部 体现出来了，真是良心。接下来，我觉得程序设计和计算机系统也不错，这门课可以让我 们深入到计算机底层去理解程序是如何运行的，吴老师每一点讲的都 很细致，突出重点，如果有 C 语言基础的同学，这门课还是值得选的。

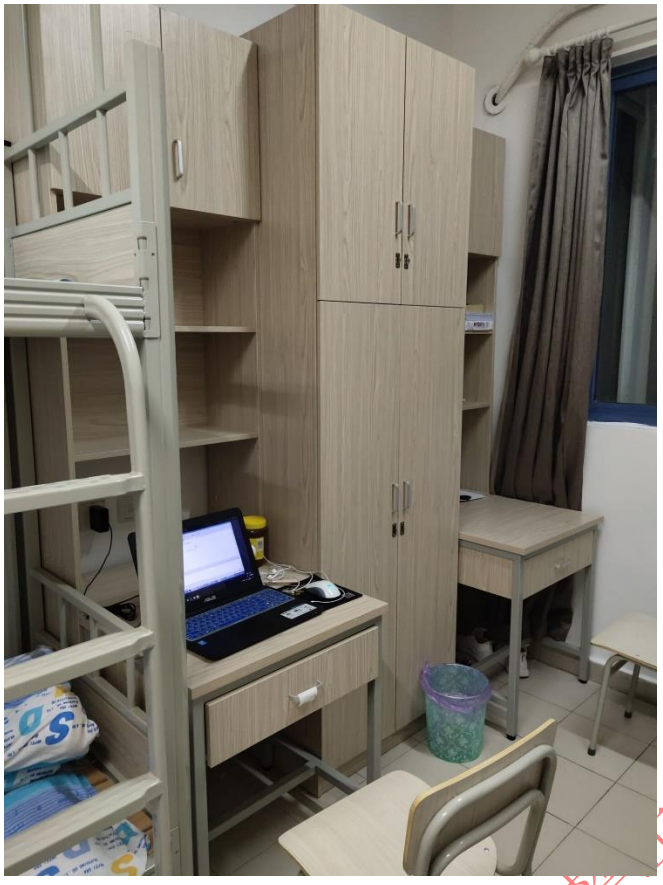
组合数学，我感觉比较难，想要学好要花很多时间，学弟学妹们 可以根据自身情况来选择。 设备驱动程序设计，是嵌入式方向的课，涉及到很多硬件知识， 根据自身情况选择，如果像我一样没课选，那这门课还是值得选的，最后的考题不是很难，容易过。

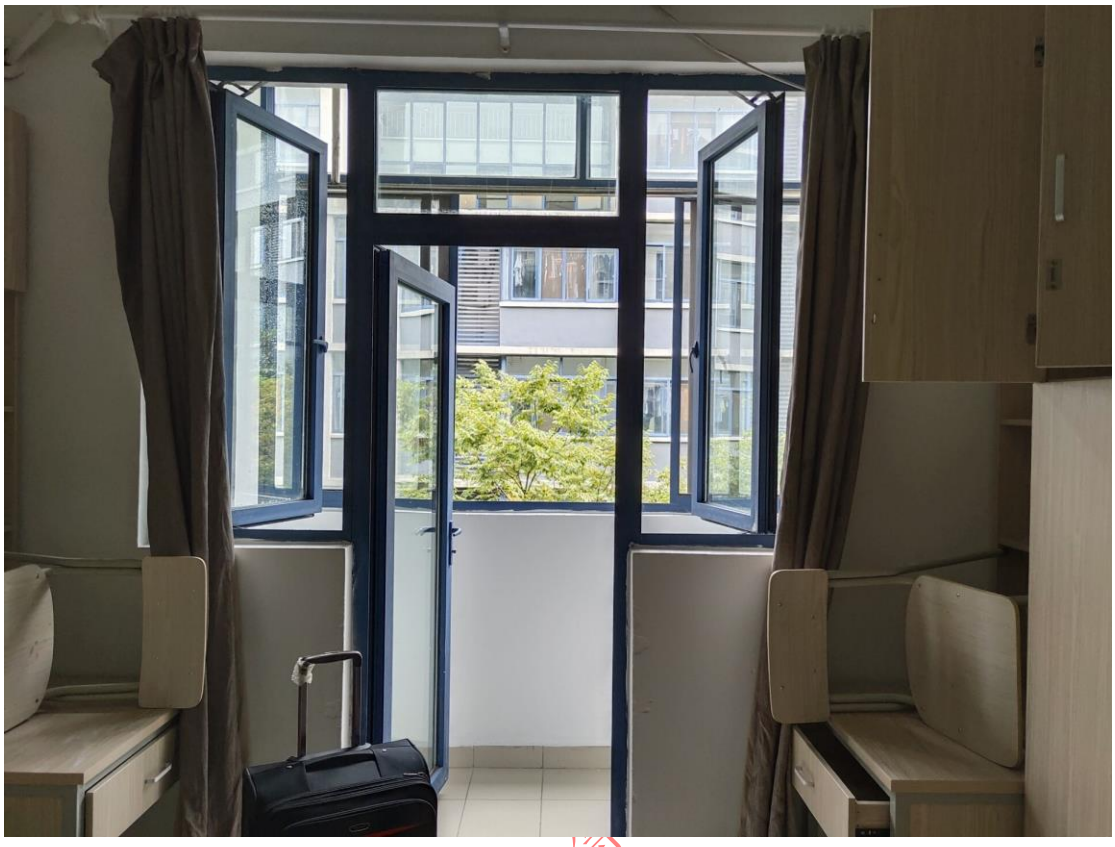
设计模式，一周两节课，其中一节是讨论课，大量的学习内容是 放在课下看视频学习，朱洪军老师在网上有录制课程，作业要组三人 队进行协作完成，作业量偏多，但是可以学到很多东西，对想在后端开发的同学有益处，可以根据自身情况选择。 以上的就是部分课程情况，下面说说软院的一些活动。 迎新晚会和相亲大会是比较大的俩个活动，迎新晚会我参加了， 节目都是由我们科大的学生自己出的，不看不知道啊，我们软院真是 藏龙卧虎，不光代码写的好，唱歌说相声跳舞也是样样都精通啊，作 为一个只会喊 666 的吃瓜群众，真心觉得晚会的质量很好。

相亲大会 是三校联谊（人大，东南，科大），但是我没有去，因为本人已经有 了女朋友，不过听我舍友讲，现场气氛活跃，人来的特别多，应该会有很大的机会，这样的相亲大会据说是科大的传统，在合肥本校有着 悠久的历史。 总结一下，科大的无论是学习氛围还是活动氛围，都是很不错的， 这样的发展模式其实我认为才是正确的，如果每天只是不知疲倦的学 习，缺乏锻炼和娱乐，这样的学习效率不会太高，放松心态身体健康 努力学习工作才是一个正确的研究生生活，也是一个可以持续发展的 生活状态。最后预祝学弟学妹们可以金榜题名，亲自感受科大的生活。

宿舍图

苏州





以下为合肥宿舍