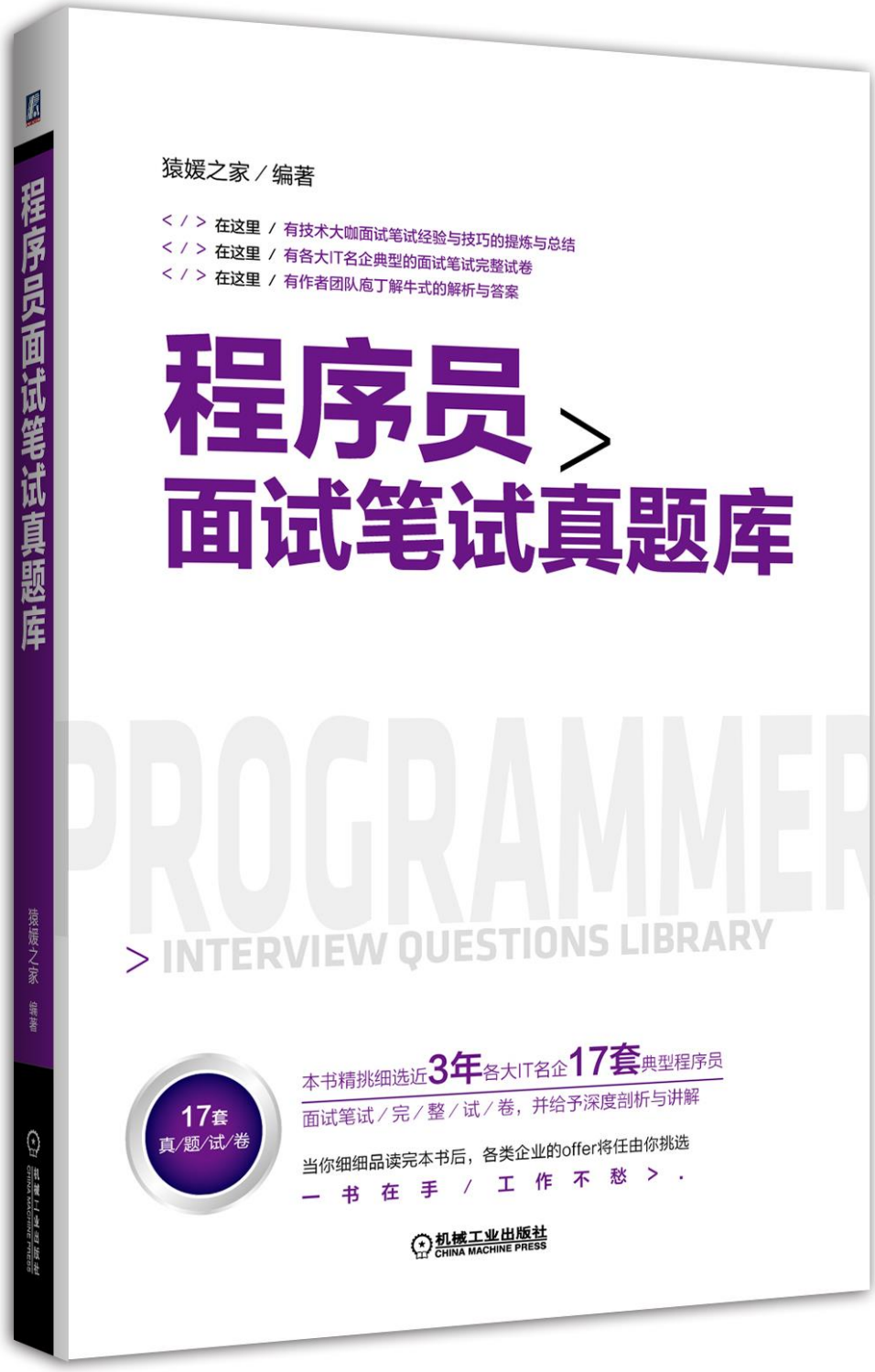


支持作者劳动成果，欢迎购买纸质图书，京东、当当、亚马逊、淘宝均有售。欢迎分享、欢迎转发。

程序员面试笔试真题库

猿媛之家　编著

程序员最可信赖的求职帮手

机械工业出版社

本书针对当前各大 IT 企业面试笔试中的特点与侧重点，精心挑选了近 3 年来 17 家著名 IT 企业的面试笔试真题。由于这些企业所涉及的业务包括系统软件、搜索引擎、电子商务、手机 App、安全关键软件等，非常具有代表性与参考性。同时，本书对这些题目进行了庖丁解牛式的分析与讲解，针对试题中涉及的部分重点、难点问题都进行了适当的扩展与延伸，力求对知识点的讲解清晰而不紊乱，全面而不啰嗦，使读者不仅能够获取到求职的知识，还能更有针对性地进行求职准备，最终得到一份满意的工作。

本书是一本计算机相关专业毕业生面试、笔试的求职用书，同时也适合那些期望在计算机软、硬件行业大显身手的计算机爱好者阅读。

图书在版编目（CIP）数据

程序员面试笔试真题库 / 猿媛之家编著. —北京：机械工业出版社，2016.10
ISBN 978-7-111-55104-1

Ⅰ. ①程… Ⅱ. ①猿… Ⅲ. ①程序设计－资格考试－习题集 Ⅳ. ①TP311.1-44

中国版本图书馆 CIP 数据核字（2016）第 244941 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
策划编辑：时 静 责任编辑：时 静 吴晋瑜
责任校对：张艳霞 责任印制：

印刷（ 装订）

2016 年 11 月第 1 版·第 1 次印刷
184mm×260mm·21.5 印张·518 千字
0001－ 册
标准书号：ISBN 978-7-111-55104-1
定价： 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务 网络服务
服务咨询热线：010-88361066 机 工 官 网：www.cmpbook.com
读者购书热线：010-68326294 机 工 官 博：weibo.com/cmp1952
010-88379203 金 书 网：www.golden-book.com
封面无防伪标均为盗版 教育服务网：www.cmpedu.com

前　言

程序员求职始终是当前社会的一个热点，而市面上有很多关于程序员求职的书籍，例如《程序员代码面试指南》《剑指 offer》《程序员面试笔试宝典》《Java 程序员面试笔试宝典》《编程之美》《编程珠玑》等，它们都是针对基础知识的讲解，各有侧重点，而且得到了广大读者的认可，但是，我们发现，当前市面上没有一本专门针对 C/C++程序员、Java 程序员的面试、笔试真题的分析与讲解。很多读者朋友们反映，他们经过精心准备之后，不知道自己是否真的能够在程序员面试笔试中得心应手。而网上一些 IT 企业的面试、笔试真题大都七零八凑，且绝大多数都是一些博主自己做的，答案简略，准确性不高，这就导致读者做完了这些真题，也不知道自己做得是否正确。

针对这种情况，编写团队经过精心准备，细致挑选了 17 套当前著名 IT 企业的笔试真题，将其汇集成册，形成了这样一本《程序员面试笔试真题库》。编者从数百份真题中挑选的这 17 套真题非常有代表性，这主要体现在以下三点：

第一，考查率高。本书所选真题均为程序员面试笔试常考点，例如语言基础、链表、算法、海量数据处理等。

第二，行业代表性强。本书所选真题均来自知名 IT 企业，其中绝大多数真题因为题目难易适中且具有非常好的区分度，通常会被众多企业借鉴。

第三，答案详尽，授之以渔。本书对每一道题目都给出了非常详细的解答，不仅给出了正确答案，还提供了详细的解答过程。读者在学完基础知识以后，可以通过做本书中的习题来找出自己的知识盲区，查漏补缺，从而真正掌握这些知识点。

由于篇幅所限，本书无法将所有程序员面试、笔试真题囊括其中，鉴于此，编写团队（猿媛之家）在官方网站（www.yuanyuanba.com）上提供了一个读者交流平台，供读者上传各类面试笔试真题，查找自己所需要的知识，同时也可供读者向本平台提供当前最新、最热门的程序员面试笔试题、面试技巧等相关材料。除此以外，我们还建立了公众号“**猿媛之家**”，作为对外消息发布平台，以期最大限度地满足读者需要。

本书主要针对 C/C++用户，后续还有专门针对 Java 用户的类似图书。有需要的读者可以在各大电商网站或实体书店进行购买。

感谢给予我们帮助的亲人、同事、朋友以及同学，无论我们遇到了多大的挫折与困难，他们都能一如既往地支持和帮助我们，在此对他们致以最衷心的感谢。

所有的成长和伟大，如同中药和老火汤，都是一个时辰一个时辰地熬出来的；所有的好书，都是斟字酌句地打磨出来的。在技术的海洋里，我们不是创造者，但我们更愿意去当好一名传播者，以期让更多的求职者能够通过本书的系统学习，找到一份自己满意的工作，实现自己的人生理想与抱负。

人生如戏，我们每个人都应成为自己这场戏剧的主角，所以，求职者在求职的道路上，无论遇到了多大的困难、遭遇了多大的挫折，都不要轻言放弃，只要你认真努力，就能在属于你的舞台上绽放自己的精彩。请记住：“在这个世界上，没有人可以让你仰视，除非你自己跪着”。

由于编者水平有限，书中不足之处在所难免，还望读者见谅。读者如果发现问题或是有此方面的困惑，可以通过邮箱 yuancoder@foxmail.com 联系我们。

目　　录

真题详解篇

面试笔试经验技巧篇

想找到一份程序员的工作,没有技术显然是不行的,但只有技术也是不够的。面试笔试经验技巧篇主要针对程序员面试笔试中遇到的 13 个常见问题进行深度解析，并且结合实际情景，给出了一个较为合理的参考答案供读者学习与应用。掌握这 13 个问题的应答技巧，对于求职者大有裨益。

面试笔试经验技巧 1　如何巧妙地回答面试官的问题

在程序员面试中，求职者不可避免地需要回答由面试官提出的各种刁钻、犀利的问题，这时不能简单地回答“是”或者“不是”，而应该具体分析“是”或者“不是”的理由。

那么，面对面试官提出的各类问题，如何才能条理清晰地回答呢？如何才能让自己的回答令面试官满意呢？

谈话是一种艺术，回答问题也是一种艺术。同样的问题，不同的回答方式，往往会产生不同的效果，甚至是截然不同的效果。在此，编者提出以下几点建议，供读者参考。

首先，回答问题务必谦虚谨慎。既不能让面试官觉得自己很自卑，唯唯诺诺，也不能让面试官觉得自己清高自负，而应该通过回答问题表现出自己自信从容、不卑不亢的一面。例如，当面试官提出“你在项目中起到了什么作用”的问题时，如果回答“我完成了团队中最难的工作”，可能会给面试官一种居功自傲的感觉；而如果回答“我完成了文件系统的构建工作，这个工作被认为是整个项目中最具挑战性的一部分内容，因为它几乎无法重用以前的框架，需要重新设计”，这种回答有理有据，更能打动面试官。

其次，回答面试官的问题时，要适当地留有悬念。人一般都有猎奇的心理，面试官自然也不例外，而且人们往往对好奇的事情更有兴趣，也更加记忆深刻。所以，在回答面试官问题时，应说关键点而非细节，说重点而非和盘托出，通过关键点，吸引面试官的注意力，等待他们继续“刨根问底”。例如，当面试官希望对简历中的一个算法问题有进一步了解时，求职者可以这样回答“我设计的这种查找算法，对于80%以上的情况，都可以将时间复杂度从O(n)降低到O(logn)，如果您有兴趣，我可以详细给您分析具体的细节。”

最后，回答问题要条理清晰、简单明了，最好使用“三段式”方式。所谓“三段式”，有点类似于中学作文中的写作风格，包括“场景/任务”“行动”“结果”三部分内容。以面试官提的问题“你在团队建设中，遇到的最大挑战是什么”为例，第一步，分析“场景/任务”：在我参与的一个企业资源计划（Enterprise Resource Planning，ERP）项目中，我们团队一共四个人，除了我以外的其他三个人中，有两个人能力较好，人也比较好相处，但有一个人却不太好相处，每次小组讨论问题时，他都不太爱说话，也很少发言，分配给他的任务也很难完成。第二步，分析“行动”：为了提高团队的综合实力，我决定找个时间和他单独谈谈。于是我利用周末时间，约他一起吃饭，顺便讨论项目问题，并询问了一些项目中他遇到的问题，通过他的回答，我发现他并不懒，只是对项目不太了解，缺乏经验，缺乏自信而已，所以越来越孤立，越来越不愿意讨论问题。为了解决这个问题，我尝试着把问题细化到他可以完成的程度，从而帮助他建立自信心。第三步，分析“结果”：慢慢地，他的技术水平有了大幅提升，不仅能够按时完成安排给他的工作，人也越来越自信了，也越来越喜欢参与小组的讨论，并能良好地表达自己的想法了。由此，团队整体协作能力也得到了提升。“三段式”回答一个最明显的好处就是条理清晰，既有描述，也有结果，有理有据，让面试官一目了然。

回答问题是一门大学问。求职者可以在平时的生活中加以练习，提高自己与人沟通的技能，等到面试时，自然也得心应手了。

面试笔试经验技巧 2　如何回答技术性问题

在面试中，面试官会经常询问一些技术性的问题，有的问题可能是历年的笔试面试真题，求职者在平时的复习中会经常遇到，应对自然不在话下，但有的题目可能比较难，

来源于 Google、Microsoft 等大企业的题库或是企业为了招聘需要自己设计的题库，求职者可能从来没见过或从来都不能完整地、独立地想到解决方案，而这些题目往往又是企业比较关注的。

如何能够回答好这些技术性的问题呢？编者建议，会做的一定要拿满分，不会做的一定要拿部分分。也就是说，对于简单的题目，求职者要努力做到完全正确；对于难度比较大的题目，不要有畏难心理，即使无法完全做出来，也要努力思考问题，至少要把自己的思路表达清楚，而不是完全回答“不会”或放弃，因为面试官除了关注你独立思考问题的能力以外，还会关注你技术能力的可塑性，观察求职者是否能够在别人的引导下正确地解决问题，所以，对于你不会的问题，他们很有可能会循序渐进地启发你去思考，以期通过这个过程进一步了解你。

一般而言，在回答技术性问题时，求职者大可不必胆战心惊，除非是没学过的新知识，否则，一般都可以采用以下六个步骤来分析解决。

（1）勇于提问　面试官提出的问题有时可能过于抽象，让求职者不知所措或无从下手，所以，对于面试中的疑惑，求职者要勇敢地提出来，多向面试官提问，把不明确或是有二义性的情况都问清楚。这样做不仅不会让面试官烦恼，影响你的面试成绩，相反还会对面试结果产生积极影响：一方面，提问可以让面试官知道你在思考，也可以给面试官留下一个心思缜密的好印象；另一方面，提问有利于后续自己对问题的解答。

例如，面试官提出一个问题——设计一个高效的排序算法。求职者可能摸不到头脑，排序对象是链表还是数组？数据类型是整型、浮点型、字符型，还是结构体类型？数据基本有序还是杂乱无序？数据量有多大？1000 以内，还是百万以上个数？此时，求职者大可以将自己的疑问提出来，问题清楚了，就可以有针对性地设计解决方案了。

（2）高效设计　对于技术性问题，如何才能打动面试官？完成基本功能肯定是必需的，仅此而已吗？显然不是，完成基本功能只能算是及格，要想达到优秀，至少还应该考虑更多的内容。以排序算法为例，时间是否高效？空间是否高效？数据量不大时也许没有问题，如果是海量数据呢？是否考虑了相关环节，例如数据的“增删改查”？是否考虑了代码的可扩展性、安全性、完整性以及鲁棒性？如果是网站设计，是否考虑了大规模数据访问的情况？是否需要考虑分布式系统架构？是否考虑了开源框架的使用？

（3）伪代码先行　有时实际代码会比较复杂，上手就写很有可能漏洞百出、条理混乱，所以，求职者可以先征求面试官的同意，在编写实际代码前，写一个伪代码或画好流程图，这样做往往会让思路更加清晰明了。

切记在写伪代码前要告诉面试官，否则他们很有可能对你产生误解，认为你只会纸上谈兵，缺乏实际编码能力。

（4）控制节奏　如果是算法设计题，面试官都会给求职者一个用以完成设计的时间限制，一般为 20min 左右。完成得太慢，会给面试官留下能力不够的印象；但完成得太快，如果不能保证 100%正确，也会给面试官留下毛手毛脚的印象。速度快当然是好事情，但只有速度没有质量，就不会起到加分的作用。所以，编者建议控制好答题节奏，如果完成得比较快，也不要急于提交给面试官，最好能够利用剩余的时间，认真仔细地检查一些边界情况、异常情况、极性情况等是否也能满足要求。

（5）规范编码　回答技术性问题时，多数都是在纸上写代码，离开了编译器的帮助，求职者要想让面试官对自己的代码一看即懂，除了字迹工整以外，最好能够严格遵循编码规范，同时，代码设计应该具有完整性，保证代码能够完成基本功能、输入边界值能够得到正确的输出、对各种不合规范的非法输入能够做出合理的错误处理，否则，写出的代码即使很高效，面试官也不一定看得懂或看起来非常费劲，这对面试成功都是非常不利的。

（6）精心测试　众所周知，任何软件都有缺陷（Bug）。但不能因为如此，就让自己的代码

漏洞百出。尤其是在面试过程中，实现功能也许并不那么困难，难的是在有限的时间内设计出的算法是否使各种异常都得到了有效的处理，是否使各种边界值都在算法设计的范围内，等等。

测试代码是让代码变得更加完美的高效方式之一，也是一名优秀程序员必备的素质之一。所以，在编写代码前，求职者最好能够了解一些基本的测试知识，做一些基本的单元测试、功能测试、边界测试以及异常测试。

在回答技术性问题时，即便是在思考问题，也不要“惜字如金”。面试官面试的时间是有限的，他们希望在有限的时间内尽可能地去了解求职者，求职者的“惜字如金”很有可能会让面试官觉得思考问题能力以及沟通能力存在问题。

其实，在面试时，求职者往往会存在一种思想误区——把技术性面试的结果看得过于重要。对于面试过程中的技术性问题，结果固然重要，但也并非最重要的内容，因为面试官看重的不仅仅是最终的结果，还包括求职者在解决问题的过程中体现出来的逻辑思维能力以及分析问题能力。所以，求职者在面试过程中，要适当地提问，通过提问获取面试官的反馈信息，并抓住这些有用信息进行辅助思考，从而提高面试的成功率。

面试笔试经验技巧 3　如何回答非技术性问题

评价一个人的能力，除了专业能力，还有一些非专业能力，如智力、沟通能力、反应能力等，所以在 IT 企业招聘过程的笔试面试环节中，并非所有的笔试内容都是 C/C++、数据结构与算法、操作系统等专业知识，也包括其他一些非技术类的知识，如智力题、推理题、作文题等。技术水平类测试可以考查一个求职者的专业素养，而非技术类测试则更侧重于考查求职者的综合素质，包括数学分析能力、反应能力、临场应变能力、思维灵活性、文字表达能力、性格特征等。考查的形式多种多样，但与公务员考查相似，主要包括行测（占大多数）、性格测试（大部分都有）、应用文、开放问题等内容。

每个人都有自己的答题技巧，答题方式也各不相同，以下是一些相对比较好的答题技巧（以行测为例）。

1）合理有效的时间管理。由于题目的难易不同，因此不要对所有题目都“绝对公平”，要有轻重缓急，最好的做法是不按顺序回答。行测有各种题型，如数量关系、图形推理、应用题、资料分析、文字逻辑等，而不同的人擅长的题型是不一样的，因此应该先回答自己最擅长的问题。例如，如果对数字比较敏感，那么就先答数量关系。

2）注意时间的把握。由于题量一般都比较大，因此可以先按照总时间/题数来计算每道题的平均答题时间，如 10s，如果看到某一道题 5s 后还没思路，则应马上放弃。在做行测题目时，以在最短的时间内拿到最多分为目标。

3）平时多关注图表类题目，培养迅速抓住图表中的各个数字要素间相互逻辑关系的能力。

4）做题要集中精力，只有集中精力、全神贯注，才能将自己的水平最大限度地发挥出来。

5）学会关键字查找，能够提高做题效率。

6）提高估算能力，很多时候，估算能够极大地提高做题速度，同时保证正确率。

除了行测以外，一些企业非常相信个人性格对职位的影响，所以都会引入相关的性格测试题用于测试求职者的性格，看其是否适合所申请的职位。大多数情况下，只要按照自己的真实想法选择就行了，不要弄巧成拙，因为测试是为了得出正确的结果，所以大多数测试题前后都有相互验证的题目。如果求职者自作聪明，选择该职位可能要求的性格选项，则很可能导致测试前后不符，这样很容易让企业认为你是个不诚实的人，从而不予考虑。

面试笔试经验技巧 4　如何回答快速估算类问题

有些企业的面试官喜欢出一些快速估算类问题。对他们而言，这些问题只是手段，不是目的，能够得到一个满意的结果固然是他们所需要的，但他们更希望通过这些题目去考查求职者的快速反应能力以及逻辑思维能力。由于求职者平时准备的时候可能对此类问题有所遗漏，因此一时很难想到解决的方案。这些题目看似毫无头绪，但实际上比较灵活，属于开放性试题，一般没有标准答案，只要找准回答要点，分析合理到位，使答案具有说服力，就可以了。

例如，面试官可能会问这样一个问题："请你估算一家商场在促销时一天的营业额"，求职者如何能够得出一个准确的数据呢？

其实本题只要能够分析出一个概数就行了，不一定要得出精确数据，而分析概数的前提就是做出各种假设。以该问题为例，可以尝试从以下思路入手：从商场规模及商铺规模入手，通过每平方米的租金估算出商场的日租金，再根据商铺的成本构成得到全商场日均交易额，然后考虑促销时的销售额与平时销售额的倍数关系，乘以倍数，即可得到促销时一天的营业额。具体而言，包括以下估计数值：

1）以一家较大规模商场为例，商场一般按 6 层计算，每层大约长 100m，宽 100m，合计 60000m^2。

2）商铺规模约占商场规模的一半，合计 30000m^2。

3）商铺租金约为 40 元/m^2，估算出年租金为 40 元/m^2×30000m^2×365 天=4.38 亿元。

4）对商户而言，租金一般占销售额的 20%左右，则年销售额为 4.38 亿元÷20%=21.9 亿元，则计算平均日销售额为 21.9 亿元/365 天=600 万。

5）促销时的日销售额一般是平时的 10 倍，所以大约为 600 万×10=6000 万。

此类题目涉及面比较广，例如，"估算北京小吃店的数量。""估算中国在过去一年方便面的市场销售额。""估算长江的水的质量。""估算一个行进在小雨中的人 5min 内身上淋到的雨的质量。"，等等。但一般都是即兴发挥，遇到此类问题，读者应一步步地抽丝剥茧，找准要点，合理分析，给出具有说服力的答案。

面试笔试经验技巧 5　如何回答算法设计问题

在面试中，很多算法设计问题都是各家企业历年来的现成题目，不管求职者以前对算法知识学习得是否扎实，理解得是否深入，学上一段时间，牢记于心，应付此类题目应该没有问题，但遗憾的是，很多世界级知名企业也深知这一点，如果纯粹是出一些毫无技术含量的题目，会让考前"突击手"占尽便宜，这样不利于人才的选拔，所以，为了把优秀的求职者与一般的求职者能够更好地区分开来，他们往往会推陈出新，越来越倾向于出一些有技术含量的新题。

在面试中，算法的地位如同托福考试在出国考试中的地位——是必需的但不是最重要的，它只是众多考核方面中的一个，不直接决定求职者的"生死"。虽然如此，但并不是说不用去准备算法知识，因为算法知识回答得好，必然会成为面试的加分项，对于求职成功，百利而无一害。那么，如何应对此类题目呢？很显然，编者不可能将此类题目都在《程序员面试笔试宝典》中一一解答，一来由于内容众多，篇幅有限；二来也没必要，今年考过了，以后一般就不会再考了，不然还是没有区分度。编者以为，靠死记硬背肯定是行不通的，解答此类算法设计问题，需要求职者具有扎实的基本功以及良好的运用能力，这里仅提供一些比较好的答题方法和解题思路，以供求职者在面试时应对此类算

法设计问题。

（1）归纳法　此方法通过写出问题的一些特例来分析总结其中的一般规律。具体而言，就是通过列举少量的特殊情况，经过分析，最后找出其中的一般关系。例如，某人有一对兔子饲养在围墙中，如果它们每个月生一对兔子，且新生的兔子在第二个月后也是每个月生一对兔子，问：一年后围墙中共有多少对兔子？

使用归纳法解答此题，首先想到的就是第一个月有多少对兔子。第一个月，最初的一对兔子生下一对兔子，此时围墙内共有两对兔子。第二个月仍是最初的一对兔子生下一对兔子，共有 3 对兔子。到第三个月，除最初的兔子新生一对兔子外，第一个月生的兔子也开始生兔子，因此共有 5 对兔子。通过举例，可以看出，从第二个月开始，每一个月兔子总数都是前两个月兔子总数之和，$U_{n+1}=U_n+U_{n-1}$，一年后，围墙中的兔子总数为 377 对。

此种方法比较抽象，也不可能对所有的情况进行列举，所以，得出的结论只是一种猜测，还需要进一步证明。

（2）相似法　如果面试官提出的问题与求职者以前用某个算法解决过的问题相似，此时就可以触类旁通，尝试改进原有算法来解决这个新问题。而通常情况下，此种方法都会比较有效。

例如，实现字符串的逆序打印，也许求职者从来就没遇到过此问题，但将字符串逆序肯定是在求职准备的过程中见过的。将字符串逆序的算法稍加处理，即可实现字符串的逆序打印。

（3）简化法　使用此方法，应先将问题简单化，例如改变一下数据类型、空间大小等，然后尝试着将简化后的问题解决，一旦有了一个算法或思路可以解决这个被简化过的问题，再将问题还原，尝试用此类方法解决原有问题。

例如，在海量日志数据中提取出某日访问×××网站次数最多的 IP。很显然，由于数据量巨大，直接进行排序不可行，但如果数据规模不大，采用直接排序不失为一种好的解决方法。那么，如何将问题规模缩小呢？于是想到了散列法，散列往往可以缩小问题规模，然后在简化过的数据里面使用常规排序算法即可找出此问题的答案。

（4）递归法　为了降低问题的复杂度，很多时候都会将问题逐层分解，最后分解为一些最简单的问题，这就是递归。使用此种方法，应先解决最基本的情况，然后以此为基础，解决其他问题。

例如，在寻求全排列时，可能会感觉无从下手，但仔细推敲，会发现后一种排列组合往往是在前一种排列组合的基础上进行重新排列，一旦知道了前一种排列组合的各类组合情况，只需要把最后一个元素插入到前面各种组合的排列里面，就解决了这一问题，即先截去字符串 s[1，…，n]中的最后一个字母，生成所有 s[1，…，n−1]的全排列，然后再将最后一个字母插入到每一个可插入的位置。

（5）分治法　任何一个可以用计算机求解问题所需的计算时间都与其规模有关。问题的规模越小，越容易直接求解，解题所需的计算时间也越少。而分治法也是如此，即将一个难以直接解决的大问题，分割成一些规模较小的相同问题，各个击破，分而治之。分治法一般包含三个步骤：①将问题的实例划分为几个较小的实例，最好具有相等的规模；②对这些较小的实例求解，而最常见的方法一般是递归；③如果有必要，则合并这些较小问题的解，以得到原始问题的解。

分治法是程序员面试常考的算法之一，一般适用于二分查找、大整数相乘、求最大子数组和、找出伪币、金块问题、矩阵乘法、残缺棋盘、归并排序、快速排序、距离最近的点对、导线与开关等。

（6）散列法　很多面试笔试题目都要求求职者给出的算法尽可能高效。那么，究竟什么样的算法是高效的？一般而言，时间复杂度越低，算法越高效。而要想达到时间复杂度的高效，

很多时候就必须在空间上有所牺牲，即“用空间来换时间”。而用空间换时间最有效的方式就是散列法、大数组、位图法。当然，此类方法并非“包治百病”，有时面试官也会对空间大小进行限制，此时，求职者只能再去思考其他的方法了。

其实，凡是涉及大规模数据处理的算法设计，散列法就是最好的方法之一。

（7）轮询法　每道面试笔试题，在设计时，往往会有一个载体，这个载体便是数据结构，例如数组、链表、二叉树、图等。当载体确定后，可用的算法自然而然地就会显露出来。可问题是，很多时候并不确定这个载体是什么。当无法确定这个载体时，求职者一般也就很难想到合适的方法了。

编者建议，此时求职者可以采用最原始的方法——轮询，在脑海中轮询各种可能的数据结构与算法，从中找出一种合适的解法。

常考的数据结构与算法知识点见下表。

数据结构	算法	概念
链表	广度（深度）优先搜索	位操作
数组	递归	设计模式
二叉树	二分查找	内存管理（堆、栈等）
树	排序（归并排序、快速排序等）	
堆（大顶堆、小顶堆）	树的插入、删除、查找、遍历等	
栈	图论	
队列	散列法	
向量	分治法	
散列表	动态规划	

此种方法看似笨拙，其实很实用，只要求职者对常见的数据结构与算法烂熟于心。

为了更好地理解这些方法，求职者可以在平时的准备过程中，应用此类方法去答题，练习得多了，自然就熟能生巧了，面试时遇到此类问题，也就能够应对自如了。

面试笔试经验技巧 6　如何回答系统设计问题

在面试时，求职者偶尔也会遇到一些系统设计问题，而这些题目往往只是测试求职者的知识面，或者测试求职者对系统架构知识的了解，一般不会涉及具体的编码工作。虽然如此，对于此类问题，很多人还是感觉难以应对。

如何应对此类题目呢？在正式介绍基础知识之前，首先罗列几个常见的与系统设计相关的面试笔试题。

1）设计一个域名系统（Domain Name System，DNS）的 Cache 结构，要求能够满足每秒 5000 次以上的查询，满足 IP 数据的快速插入，查询的速度要快（题目还给出了一系列的数据，例如，站点数总共为 5000 万，IP 地址有 1000 万，等等）。

2）有 N 台机器，M 个文件，文件可以以任意方式存放到任意机器上，文件可任意分割成若干块。假设这 N 台机器的宕机率小于 1/3，要求在宕机时可以从其他未宕机的机器中完整地导出这 M 个文件，求最好的存放与分割策略。

3）假设有 30 台服务器，每台服务器中都存有上百亿条数据（有可能重复），如何根据某关键字，从中找出重复出现次数最多的前 100 条数据？要求使用 Hadoop 来实现。

4）设计一个系统，要求写速度尽可能快，并说明设计原理。

5）设计一个高并发系统，说明架构和关键技术要点。

6）假设有一个 25T 的 log(query->queryinfo)，log 的大小还在不段地增长，设计一个方案，

给出一个 query 能快速返回 queryinfo。
以上所有问题中，凡是不涉及高并发的，基本可以采用 Google 的三个技术解决，即 GFS、MapReduce 和 Bigtable，这三个技术被称为“Google 的三驾马车”，Google 只公开了论文而未公开源代码，开源界对此非常感兴趣，于是仿效这三篇论文实现了一系列软件，如 Hadoop、HBase、HDFS、Cassandra 等。
在 Google 这些技术还未出现之前，企业界在设计大规模分布式系统时，采用的架构往往是 Database+Sharding+Cache，现在很多公司仍采用这种架构。在这种架构中，仍有很多问题值得去探讨。如采用什么数据库，是 SQL 界的 MySQL 还是 NoSQL 界的 Redis/TFS，两者有何优劣？采用什么方式进行数据分片（Sharding）？是水平分片，还是垂直分片？据网上资料显示，weibo.com 和 taobao 图片存储中曾采用的架构是 Redis/MySQL/TFS+Sharding+Cache。该架构解释如下：前端 Cache 是为了提高响应速度；后端数据库则用于数据永久存储，防止数据丢失；而 Sharding 是为了在多台机器间分摊负载。最前端由大块的 Cache 组成，要保证至少 99%（该数据在 weibo.com 架构中的占比是编者推测的，而在 taobao 图片存储模块是真实的）的访问数据落在 Cache 中，这样可以保证用户访问速度，减少后端数据库的压力。此外，为了保证前端 Cache 中的数据与后端数据库中的数据一致，需要有一个中间件异步更新（这是因为同步代价太高。请思考，异步有缺点，如何弥补？）数据。另外，为了分摊负载压力和海量数据，会将用户的微博信息经过分片（即 Sharding）后存放到不同的结点上。
这种架构的优点非常明显——简单，在数据量和用户量较小时完全可以胜任。但其扩展性和容错性差、维护成本高，尤其是数据量和用户量暴增之后，系统不能通过简单地增加机器解决该问题。
于是，新的架构便出现了，即前面讲到的“Google 的三驾马车。
1）**GFS** 是一个可扩展的分布式文件系统，用于大型的、分布式的、对大量数据进行访问的应用。它运行于廉价的普通硬件上，提供容错功能。现在开源界有 Hadoop 分布式文件系统（Hadoop Distributed File System，HDFS），该文件系统虽然弥补了数据库+Sharding 的很多缺点，但自身仍存在一些问题，例如，由于采用 master/slave 架构，因而存在单点故障问题；元数据信息全部存放在 master 端的内存中，因而不适合存储小文件，或者说如果存储的大量小文件，那么存储的总数据量不会太大。
2）**MapReduce** 是针对分布式并行计算的一套编程模型。其最大的优点是编程接口简单、自动备份（数据默认情况下会自动备三份）、自动容错和隐藏跨机器间的通信。在 Hadoop 中，MapReduce 作为分布计算框架，而 HDFS 作为底层的分布式存储系统，但 MapReduce 不是与 HDFS 耦合在一起的，用户完全可以使用自己的分布式文件系统替换 HDFS。当前 MapReduce 有很多开源实现，如 Java 实现 Hadoop MapReduce、C++实现 Sector/sphere 等，甚至有些数据库厂商将 MapReduce 集成到数据库中。
3）**BigTable** 俗称“大表”，用来存储结构化数据。编者个人认为，BigTable 在开源界最火爆，其开源实现最多，包括 HBase、Cassandra、levelDB 等，使用也非常广泛。
除了 Google 的这“三驾马车”以外，还有其他一些技术可供学习与使用。
1）**Dynamo**：亚马逊的 key-value 模式的存储平台，可用性和扩展性都很好，采用分布式散列表（Distributed Hash Table，DHT）对数据分片，解决单点故障问题。在 Cassandra 中，也借鉴了该技术，在 BT 和电驴中，也采用了类似算法。
2）**虚拟结点技术**：该技术常用于分布式数据分片中。具体应用场景是：有一大堆数据（maybe TB 级或者 PB 级），需按照某个字段（key）分片存储到几十（或者更多）台机器上，同时想尽量负载均衡且容易扩展。传统的做法是“Hash(key) mod N”，这种方法最大的缺点是不容易扩展，即增加或者减少机器均会导致数据全部重新分布，代价太大。这时便可以使用上

面提到的 DHT（现在已经被很多大型系统采用）。还有一种是对“Hash(key) mod N”的改进：假设要将数据分布到 20 台机器上，传统做法是“Hash(key) mod 20”，而改进后，N 的取值要远大于 20，如 20000000，然后额外采用一张表记录每个结点存储的 key 的模值，例如，

node1：0～1000000
node2：1000001～2000000
……

这样当添加一个新的结点时，只需将每个结点上的部分数据移动给新结点，同时修改一下这个表即可。

3）**Thrift**：它是一个跨语言的远程过程调用协议（Remote Procedure Call Protocol，RPC）框架。RPC 的使用方式与调用一个普通函数一样，但执行体发生在远程机器上。跨语言是指不同语言之间进行通信，例如 C/S 架构中，Server 端采用 C++编写，Client 端采用 PHP 编写，若要让两者之间通信，则 thrift 是一种很好的方式。

对于这一节开始提出的几道笔试题，都可以通过上面介绍的知识点来加简答，例如：

1）关于高并发系统设计，主要有以下几个关键技术点：缓存、索引、数据分片、锁粒度尽可能小。

2）问题 2 涉及现在通用的分布式文件系统的副本存放策略。一般是将大文件切分成小的块（如 64MB）后，以块为单位存放三份到不同的结点上，这三份数据的位置需根据网络拓扑结构配置。一般而言，如果不考虑跨数据中心，可以这样存放：两个副本存放在同一个机架的不同结点上，而另外一个副本存放在另一个机架上，这样从效率和可靠性上都是最优的（Google 公布的文档对此有专门的证明）。如果考虑跨数据中心，可将两份存在一个数据中心的不同机架上，另一份存到另一个数据中心。

3）问题 4 涉及 BigTable 的模型。主要思想是将随机写转化为顺序写，进而大大提高写速度。具体方法为：由于磁盘物理结构的独特设计，其并发的随机写（主要是因为磁盘寻道时间长）非常慢，考虑到这一点，在 BigTable 模型中，首先会将并发写的大批数据放到一个内存表（称为“memtable”）中，当该表大到一定程度后，会顺序写到一个磁盘表（称为“SSTable”）中，这种写是顺序写，效率极高。说到这里，可能有读者问，随机读可不可以这样优化？答案是：看情况而定。通常而言，如果读并发度不高，则不可以这么做，因为如果将多个读重新排列组合后再执行，则系统的响应时间太慢，用户可能无法接受，而如果读并发度极高，也许可以采用类似机制。

面试笔试经验技巧 7　如何解决求职中的时间冲突问题

对于求职者而言，求职季堪比个“赶场季”，一天少则几家、十几家企业入校招聘，多则几十家、上百家企业招兵买马，企业多，选择自然也多，这固然是一件好事情，但由于招聘企业实在是太多了，自然会导致另外一个问题的发生——同一天企业扎堆，且都是自己心仪或欣赏的知名企业。如果不能够提前掌握企业的宣讲时间、地点，则很容易迟到或错过。但有时候即使掌握了宣讲时间、笔试时间、面试时间，还是有可能错过，为什么呢？时间冲突，人不是神仙，不可能具有分身术，也不可能同一时间做两件不同的事情，所以，就必须有所取舍。

到底该如何取舍呢？到底该如何应对这种时间冲突的问题呢？在此，编者将自己的一些想法经验与分享出来，以供读者参考。

1）如果多家心仪企业的校园宣讲时间发生冲突（前提是只宣讲、不笔试），此时最好的解决方法是和同学或朋友商量好，各去一家，然后大家进行信息共享。

2）如果多家心仪企业的笔试时间发生冲突，此时只能选择其一，毕竟企业的笔试时间都是考虑到了成百上千人的安排，需要提前安排考场、考务人员、阅卷人员等，不可能为了一个人而轻易改变。所以，最好选择自己更有兴趣的企业参加笔试。

3）如果多家自己心仪企业的面试时间发生冲突，求职者不要轻易放弃。对于面试官而言，面试任何人都是一样的，因为面试官谁都不认识。而面试时间也是比较灵活的，一般可以通过电话协商。求职者可以与相关工作人员（一般是企业的人力资源师）进行沟通，给出不能参加面试的原因，让其调整时间。但一般要接到面试通知后的第一时间联系相关工作人员变更时间。

那如果多家企业的校园宣讲时间、笔试时间、面试时间发生冲突，到底是去宣讲会，还是去笔试，或是去面试呢？大原则是选择自己最有兴趣的企业。但也要灵活处理，如果仅仅只是宣讲而不涉及笔试，则完全可以不去，可寻求同学或是朋友的帮忙，回来后信息共享即可。如果可以协调面试时间，那就打电话给相关工作人员，换个不冲突的时间去面试，笔试一般时间比较固定，需要协调的事情很多，所以调整时间的可能性也不大，此时如果协调得当，基本能够保证不放弃任何一个机会。

正如世界上没有能够包治百病的药物一样，以上这些建议在应用时，很多情况下也做不到全盘兼顾，当必须进行多选一时，求职者就要对此进行评估了，评估的项目可以包括对企业的中意程度、获得录用通知的概率、去工作的可能性等。评估的结果往往具有很强的参考性，求职者依据评估结果做出的选择一般也会比较合理。

真 题 篇

真题篇主要列举了 17 套来自于著名 IT 企业的面试笔试真题，这些企业是行业的标杆，代表了行业的最高水准，而它们所出的面试笔试真题不但难易适中，覆盖面广（包括语言基础、链表、算法、海量数据处理等内容），而且具有非常好的区分度，代表性非常强，是历年来程序员面试笔试中的必考项或者常考项。越来越多的中小企业开始从中选取或者加以借鉴，以此作为面试笔试题。

真题 1 某知名互联网下载服务提供商软件工程师笔试题

一、单选题

1．定义有变量 int i=0; int a=i++; int b=++a; int c=a+b，那么表达式 a?b:c 的值为（　　）。

A．0　　B．1　　C．2　　D．3

2．在 32 位计算环境下，定义有语句 int *p=new int[10]，那么 sizeof(p)的值为（　　）。

A．4　　B．10　　C．40　　D．8

3．在 32 位计算环境下，定义有语句 char str[] = "abcde"，那么 sizeof(str)的值为（　　）。

A．1　　B．4　　C．5　　D．6

4．定义有函数 int func(int i)，它的实现如下：

```
int func(int i)
{
    if(i > 1)
        return i*func(i-1);
    else
        return 1;
}
```

那么调用 f(5)方法的返回值为（　　）阶乘计算。

A．5　　B．15　　C．20　　D．120

5．以下关于类的描述中，正确的是（　　）。

A．每个类都有一个无参数的构造函数　　B．每个类都有一个拷贝构造函数

C．每个类能有多个构造函数　　D．每个类能有多个析构函数

6．用关键字 class 定义的类，其成员默认的访问属性为（　　）。

A．private　　B．protected　　C．public　　D．无定义

7．类的成员有三种访问属性，分别是 public、protected 和 private，子类能够访问的成员是（　　）。

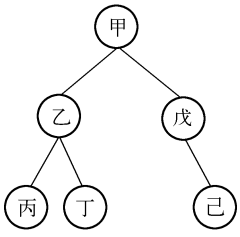
A．都能访问　　B．public 和 protected

C．public 和 private　　D．protected 和 private

8．对一个已经排好序的数组进行查找，时间复杂度为（　　）。

A．O(n)　　B．O(logn)　　C．O(nlogn)　　D．O(1)

9．有以下二叉树：



对其进行后序遍历的结果是（　　）。

A．丙乙丁甲戊己　　B．甲乙丙丁戊己

C．丙丁乙己戊甲　　D．丙丁己乙戊甲

10．有以下代码：

```
A *pa = new A[10];
delete pa;
```
则类 A 的构造函数和析构函数分别执行了（　　）次。

A．1，1　　B．10，10　　C．1，10　　D．10，1

11．有以下代码：

```
class A { public: ～A(); }; A::～A() { printf("delete A "); }
class B :public A { public: ～B(); }; B::～B() { printf("delete B "); }
```
那么执行以下代码：

```
A *pa = new B();
delete pa;
```
程序的输出结果是（　　）。

A．delete A　　B．delete B　　C．delete B delete A　　D．delete A delete B

12．文件长度是一个大于 0 的整数，用变量 unsigned file_length 来表示，把文件分成块，每块的长度也是一个大于 0 的整数，用变量 unsigned block_length 来表示，则文件被分成的块数为（　　）。

A．file_length/block_length

B．file_length/block_length+1

C．(file_length+block_length-1)/block_length

D．((file_length-1)/block_length+1

13．定义有整数 int i = 0xFE78DA45，int k = 0xAC3189B2，则 i^k 的值为（　　）。

A．0x524953f7　　B．0xAC308800

C．0xFE79DBF7　　D．0X0000001

14．有以下代码：

```
class parent
{
    public:
        virtual void output();
};
void parent::output()
{
    printf("parent");
}
class son : public parent
{
    public:
        virtual void output();
};
void son::output()
{
    printf("son");
}
```
则以下程序段：

```
son s;
::memset(&s , 0 , sizeof(s));
```

parent& p = s;
p.output();
其执行结果是（　　）。

A．parent　　B．son
C．sonparent　　D．没有输出结果，程序运行出错

15．函数的局部变量所需的存储空间是在（　　）分配的。

A．进程的数据段　　B．进程的栈上
C．进程的堆上　　D．以上都可以

16．以下 STL 的容器存放的数据中，肯定是排好序的是（　　）。

A．vector　　B．deque　　C．list　　D．map

17．有定义语句：int a[][3]={{1},{3,2},{6,7,8},{9}}，那么 a[2][1]的值是（　　）。

A．3　　B．6　　C．2　　D．7

18．以下关于头文件的描述中，正确的是（　　）。

A．#include<filename.h>，编译器寻找头文件时，会从当前编译的源文件所在的目录中找

B．#include"filename.h"，编译器寻找头文件时，会从通过编译选项指定的目录中找

C．多个源文件同时用到的全局整数变量，它的声明和定义都放在头文件中，是好的编程习惯

D．在大型项目开发中，把所有自定义的数据类型、全局变量、函数声明都放在一个头文件中，各个源文件都只需要包含这个头文件即可，省去了要写很多#include 语句的麻烦，是好的编程习惯

19．某棵完全二叉树上有 699 个结点，则该二叉树的叶子结点数为（　　）。

A．349　　B．350　　C．188　　D．187

20．一个指向字符串的指针 char *p_str，要把字符串中第四个字符的值改为'a'，正确的做法是（　　）。

A．p_str[3]='a'　　B．*(ptr+3)= 'a'　　C．p_str[4]= 'a'　　D．*(ptr+4)= 'a'

21．下列关于内联函数的描述中，正确的是（　　）。

A．类的私有成员函数不能作为内联函数

B．在所有类说明中，内部定义的成员函数都是内联函数

C．类的保护成员函数不能作为内联函数

D．使用内联函数的地方会在运行阶段用内联函数体进行替换

二、多选题

1．已知一段文本有 1382 个字符，使用了 1382 个字节进行存储，这段文本全部是由 a、b、c、d、e 这 5 个字符组成，其中，字符 a 出现了 354 次，字符 b 出现了 483 次，字符 c 出现了 227 次，字符 d 出现了 96 次，字符 e 出现了 232 次，如果对这 5 个字符使用赫夫曼（Huffman）算法进行编码，则以下说法中，正确的是（　　）。

A．使用赫夫曼算法编码后，用编码值来存储这段文本将花费最少的存储空间

B．使用赫夫曼算法进行编码，a、b、c、d、e 这 5 个字符对应的编码值是唯一确定的

C．使用赫夫曼算法进行编码，a、b、c、d、e 这 5 个字符对应的编码值可以有多套，但每个字符编码的位（bit）数是确定的

D．字符 b 的赫夫曼编码值位数应该最短，字符 d 的赫夫曼编码值位数应该最长

2．已知 double d = 3.2，int n = 3，那么下列表达式中，不合法的是（　　）。

A．d≪2　　B．d/n　　C．!d && (n-3)　　D．(d-0.2)|n

3．下列对于循环知识的描述中，正确的是（　　）。

A．while 循环语句的循环体至少执行一次

B．do-while 循环可以写成 while 循环的格式

C．continue 语句可以出现在各种循环体中

D．break 语句不可以出现在循环体内

4．下列模板声明中，非法的是（　　）。

A．template<class Type>class C1;

B．template<class T,U , class V>class C2;

C．template<class C1 , typename C2>class C3{};

D．template<typenamemyT , class myT>class C4{};

5．在使用浏览器打开一个网页的过程中，浏览器会使用的网络协议包括（　　）。

A．DNS　　B．TCP　　C．HTTP　　D．Telnet

6．下列选项中，属于构造散列函数的方法是（　　）。

A．直接定址法　　B．数字分析法　　C．除留余数法　　D．平方取中法

7．拷贝构造函数的特点是（　　）。

A．该函数名同类名，也是一种构造函数，该函数返回自身引用

B．该函数只有一个参数，必须是对某个对象的引用

C．每个类都必须有一个拷贝初始化构造函数，如果类中没有说明拷贝构造函数，那么编译器系统会自动生成一个默认的拷贝构造函数，作为该类的保护成员

D．拷贝初始化构造函数的作用是将一个已知对象的数据成员值复制给正在创建的另一个同类的对象

8．下列关于虚函数的描述中，正确的是（　　）。

A．在构造函数中调用类自己的虚函数，虚函数的动态绑定机制还会生效

B．在析构函数中调用类自己的虚函数，虚函数的动态绑定机制还会生效

C．静态函数不可以是虚函数

D．虚函数可以声明为 inline

9．下列对函数 double add(int a , int b)进行重载的描述中，正确的是（　　）。

A．int add(int a ,int b ,int c)　　B．int add(double a , double b)

C．double add(double a , double b)　　D．int add(int a , int b)

三、填空题

1．以下代码用于计算 100 以内的素数的个数，请把相应的空填上（空 1～空 4）。

```
struct prime_number_node
{
    int prime_number;
    prime_number_node* next;
};
int calc_prime_number()
{
    prime_number_node* list_head = new prime_number_node();
    list_head->next = NULL;
    list_head->prime_number = 2;
    prime_number_node* list_tail = list_head;
    for (int number = 3; number < 100; number++)
    {
        int remainder;
```

```
            prime_number_node* cur_node_ptr = list_head;
            while (cur_node_ptr != NULL)
            {
                remainder = number%cur_node_ptr->prime_number;
                if (remainder == 0)
                {
                    //1
                }
                else
                {
                    //2
                }
            }
            if (remainder != 0)
            {
                prime_number_node* new_node_ptr = new prime_number_node();
                new_node_ptr->prime_number = number;
                new_node_ptr->next = NULL;
                list_tail->next = new_node_ptr;
                //3
            }
        }
        int result = 0;
        while (list_head != NULL)
        {
            result++;
            prime_number_node* temp_ptr = list_head;
            list_head = list_head->next;
            //4
        }
        return result;
}
```

2．已知集合 A 和集合 B 的元素分别用不含头结点的单链表存储，函数 difference()用于求解集合 A 与集合 B 的差集，运算结果保存在集合 A 的单链表中。例如，若集合 A={5,10,20,15,25,30}，集合 B={5,15,35,25}，完成计算后 A={10,20,30}。

链表结点的结构类型定义如下：

```
struct node
{
    int elem;
    node* next;
};
void difference(node** LA, node* LB)
{
    node *pa, *pb, *pre, *q;
```

```
    pre = NULL;
    ;          //1
    while (pa)
    {
        pb = LB;
        while ()                          //2
            pb = pb->next;
        if ()                             //3
        {
            if (!pre)
                *LA = ;         //4
            else
                = pa->next;         //5
            q = pa;
            pa = pa->next;
            free(q);
        }
        else
        {
            ;                   //6
            pa = pa->next;
        }
    }
}
```

1、2、3、4、5、6 这六行代码依次为（　　）。

真题 2　某知名监控产品供应商和解决方案服务商

软件工程师笔试题

一、单选题

1．静态局部变量存储在进程的（　　）。

A．栈区　　B．寄存器区　　C．代码区　　D．全局区

2.在 C 语言中，有数组定义如下：char array[]="China"，则数组 array[]所占用的空间为（　　）。

A．4 个字节　　B．5 个字节　　C．6 个字节　　D．7 个字节

3．执行 C 语言代码“int a=1; int b=0; int c=0; int d=(++a)*(c=1);”后，变量 a、b、c、d 的值分别为（　　）。

A．2，0，1，2　　B．1，0，1，1　　C．2，0，1，1　　D．2，0，0，2

4．有一个变量 int a=0，两个线程同时对其进行+1 操作，每个线程加 100 次，不加锁，最后变量 a 的值是（　　）。

A．200　　B．≤200　　C．≥200　　D．都有可能

5．HTTPS 采用（　　）实现安全网站访问。

A．SSL　　B．IPsec　　C．PGP　　D．SET

6．某主机的 IP 地址为 202.117.131.12/20，其子网掩码是（　　）。

A．255.255.248.0　　B．255.255.240.0

C．255.255.252.0　　D．255.255.255.4

7．下列选项中，不属于网络安全控制技术的是（　　）。

A．防火墙技术　　B．访问控制技术
C．入侵检测技术　　D．差错控制技术

8.多媒体技术是指以计算机为平台综合处理多种媒体信息，其中“媒体”主要指的是（　　）。
A．文字和图像　　B．各种信息的编码
C．音频和视频　　D．承载信息的载体

9．一幅彩色图像（RGB），分辨率为 256×512，每一种颜色用 8 bit 表示，则该彩色图像为（　　）bit。
A．256×512×8　　B．256×512×3×8　　C．256×512×3/8　　D．256×512×3

10． 计算机通过传声器接口（MIC）收到的信号是（　　）。
A．音频数字信号　　B．音频模拟信号
C．量化信号　　D．采样信号

11．当在 CPU 内存之间进行地址转换时，（　　）将地址从虚拟（逻辑）地址空间映射到物理地址空间。
A．TCB　　B．MMU　　C．CACHE　　D．DMA

12．RS485 最少有（　　）根数据信息号。
A．2　　B．6　　C．8　　D．16

13．程序开发中，编译器的主要作用是（　　）。
A．代码编辑功能　　B．检查代码规范性
C．分析代码中的问题　　D．完成源语言与目标语言的转换

14．下列关于实时操作系统（RTOS）的任务调度器描述中，正确的是（　　）。
A．任务之间的公平性是最重要的调度目标
B．大多数 RTOS 调度算法都是可抢占式（可剥夺式）的
C．RTOS 调度器都采用了基于时间片轮转的调度算法
D．RTOS 调度算法只是一种静态优先级调度算法

二、填空题

1．当 C++语言调用 C 语言库时，需要添加（　　）。

2．sprintf 函数和 snprintf 函数的区别是（　　）。

3．定义有 a=1, b=2, c=3, d=4，那么表达式 a<b ? a:c < d ? a:d 的值是（　　）。

4．有如下代码：

```
#include<stdio.h>
int main()
{
    unsigned int a = 6;
    int b = -20;
    (a + b > 6) ? printf(">6") : printf("<=6");
    return 0;
}
```

则这段代码的输出结果是（　　）。

真题 3　某知名安全软件服务提供商软件工程师笔试题

一、不定项选择题

1．实验高中的小明暗恋某女生已经 3 年了，高考结束后，小明决定向她表白。这天，小明来到女同学家楼下等她出现，时间一分一秒地流逝，两个多小时过去了，他心仪的女生还没有出现，小明看了下表，时针和分针的位置正好跟开始等的时候互换，请问小明一共等了（　　）分钟。

A．165　　B．150　　C．172　　D．166

2．有 A、B、C 三个学生，他们一个出生在西安，一个出生在武汉，一个出生在深圳；一个学化学，一个学英语，一个学计算机。其中，①学生 A 不是学化学的，学生 B 不是学计算机的；②学化学的不出生在武汉；③学计算机的出生在西安；④学生 B 不出生在深圳。根据上述条件可知，学生 A 的专业是（　　）。

A．计算机　　B．英语　　C．化学　　D．三种专业都可能

3．在一个不透明的箱子里，一共有红、黄、蓝、绿、白五种颜色的小球，每种颜色的小球大小相同、质量相等且数量充足。每个人从篮子里抽出两个小球，那么要保证有两个人抽到的小球颜色相同，至少需要抽球的人数为（　　）。

A．11 人　　B．8 人　　C．16 人　　D．13 人

4．平面内一共有 11 个点，由它们连成 48 条不同的直线，由这些点可连成的三角形个数为（　　）。

A．162　　B．158　　C．160　　D．165

5．存在这样一个数列：“8，8，12，24，60，（　　）”，则括号内应填的内容是（　　）。

A．90　　B．180　　C．120　　D．240

6．有如下代码：

```
int func(int x)
{
    int countx = 0;
    while (x)
    {
        countx++;
        x = x&(x-1);
    }
    return countx;
}
```

假设 x 的值为 65530，那么 func(x)的返回值是（　　）。

A．20　　B．16　　C．100　　D．14

7．用某种排序方法对关键字序列（25，84，21，47，15，27，68，35，20）进行排序，序列的变化情况如下所示：

1）20，15，21，25，47，27，68，35，84

2）15，20，21，25，35，27，47，68，84

3）15，20，21，25，27，35，47，68，84

则采用的排序方法是（　　）。

A．选择排序　　B．快速排序　　C．希尔排序　　D．归并排序

8．设某棵二叉树中有 360 个结点，则该二叉树的最小高度是（　　）。

A．7　　B．9　　C．10　　D．8

9．下列排序算法中，对一个 list 排序的最快方法是（　　）。

A．快速排序　　B．冒泡排序　　C．二分插入排序　　D．线性排序

10．应用程序 ping 发出的是（　　）报文。

A．ICMP 应答B．TCP 请求　　C．TCP 应答　　D．ICMP 请求

11．有如下规约：

digit->0|1|…|9

digits->digit digit*

optionalFraction ->.digits|ε
optionalExponent ->(E(+|-|ε)digits)|ε
number -> digits optionalFraction optionlExponent
对于上面给出的正则规约的描述，下列无符号数中，不符合规约要求的是（　　）。

A．5280　　B．1　　C．2.0　　D．336E

12．语法分析器可以用于（　　）。

A．识别语法错误　　B．识别语法和语义错误

C．识别语义错误　　D．识别并修正语法和语义错误

13．IPv6 地址包含（　　）位。

A．64　　B．16　　C．32　　D．128

14．如果在一个建立了 TCP 连接的 socket 上调用 recv 函数，返回值为 0，则表示（　　）。

A．还没有收到对端数据　　B．连接发生错误

C．对端关闭了连接　　D．对端发送了一段长度为 0 的数据

15．下列选项中，不是内核对象的是（　　）。

A．进程　　B．线程　　C．互斥器　　D．临界区

16．同一进程下的多个线程可以共享的资源是（　　）。

A．栈　　B．数据区　　C．寄存器　　D．线程 ID

17．在虚拟存储系统中，若进程在内存中占 3 块（开始时为空），采用先进先出页面淘汰算法，当执行访问页号序列为 1、2、3、4、1、2、5、1、2、3、4、5、6 时，将产生缺页中断的次数是（　　）。

A．10　　B．9　　C．8　　D．7

18．下述情况中，会提出中断请求的是（　　）。

A．在键盘输入过程中，每按一次键盘上的键

B．计算结果溢出

C．一条系统汇编指令执行完成

D．两数相加的结果为零

19．单任务系统中有两个程序 A 和 B，其中，
A 程序：CPU:10s→设备 1：5s→CPU:5s→设备 2：10s→CPU:10s；
B 程序：设备 1:10s→CPU:10s→设备 2：5s→CPU：5s→设备 2：10s；
执行顺序为 A→B，那么 CPU 的利用率是（　　）。

A．40%　　B．50%　　C．60%　　D．70%

20．以下程序会打印出（　　）个“-”。

```
for (int i = 0; i<2; i++)
{
    fork();
    printf(" - \n");
}
```

A．2　　B．4　　C．6　　D．8

21．下列关于计算机的描述中，不正确的是（　　）。

A．进程调度中“可抢占”和“非抢占”两种方式，后者引起系统的开销更大

B．每个进程都有自己的文件描述符表，所有进程共享同一打开文件表和 v-node 表

C．基本的存储技术包括 RAM、ROM、磁盘以及 SSD，其中访问速度最慢的是磁盘， CPU 的高速缓存一般是由 RAM 组成的

D．多个进程竞争资源出现了循环等待可能造成系统死锁

22．下列关于 Linux 操作系统的描述中，正确的是（　　）。

A．线性访问内存非法时，当前线程会进入信号处理函数

B．用 mv 命令移动文件时，文件的修改时间会发生变化

C．ulimit –c 设置的是函数调用栈的大小

D．malloc 函数是应用程序向操作系统申请内存的接口

23. X86 体系结构在保护模式下有三种地址，下列对于这三种地址的描述中，正确的是（　　）。

A．虚拟地址先经过分段机制映射到线性地址，然后线性地址通过分页机制映射到物理地址

B．线性地址先经过分段机制映射到虚拟地址，然后虚拟地址通过分页机制映射到物理地址

C．虚拟地址先经过分页机制映射到线性地址，然后线性地址通过分段机制映射到物理地址

D．线性地址先经过分页机制映射到虚拟地址，然后线性地址通过分段机制映射到物理地址

24. 当需要对文件进行随机存取时，下列文件中，物理结构不适用于上述应用场景的是（　　）。

A．顺序文件　　B．索引文件　　C．链接文件　　D．Hash 文件

25．有如下程序：

```
#include<iostream>
using namespace std;
class MyClass
{
public:
    MyClass(int i = 0)
    {
        cout << 1;
    }
    MyClass(const MyClass& x)
    {
        cout << 2;
    }
    MyClass& operator=(const MyClass& x)
    {
        cout << 3;
        return *this;
    }
    ～MyClass()
    {
        cout << 4;
    }
};
int main()
{
    MyClass obj1(1), obj2(2), obj3(obj1);
    return 0;
```

```
}
```
这个程序的输出结果是（　　）。

A．112444　　B．11114444　　C．121444　　D．11314444

26．在一个 64 位的操作系统中，定义如下结构体：

```
struct st_task
{
    uint16_t id;
    uint32_t value;
    uint64_t timestamp;
}
```
同时定义 fool 函数如下：
```
void fool()
{
    st_task task = {};
    uint64_t a = 0x00010001;
    memcpy(&task, &a, sizeof(uint64_t));
    printf(" % 11u, % 11u, % 11u", task.id, task.value, task.timestamp);
}
```
上述 fool()函数的执行结果为（　　）。

A．1，0，0　　B．1，1，0　　C．0，1，1　　D．0，0，1

27．有如下代码：
```
int main(int argc, char **argv)
{
    int a[4] = { 1, 2, 3, 4 };
    int *ptr = (int *)(&a + 1);
    printf("%d", *(ptr-1));
}
```
程序的输出结果是（　　）。

A．1　　B．2　　C．3　　D．4

28．有如下代码：
```
int fun(int a)
{
    a = (1 << 5)-1;
    return a;
}
```
fun(21)的结果是（　　）。

A．10　　B．8　　C．5　　D．31

29．下列关于 sort 的 template 的写法中，正确的是（　　）。

A．void sort(class A first, class A last, class B pred)

B．void template(class A, class B) sort(A first, A last, B pred)

C．template<class A><class B> void sort(A first, A last, B pred)

D．template<class A, class B> void sort(A first, A last, B pred)

30．在 C++语言中，有如下代码：
```
const int i = 0;
```

```
int *j = (int *)&i;
*j = 1;
printf("%d, %d", i, *j);
```
程序的输出结果是（　　）。

A．0,1　　B．1,1　　C．1,0　　D．0,0

31．有如下代码：

```
#include<stdio.h>
char *myString()
{
    char buffer[6] = { 0 };
    char *s = "Hello World!";
    for (int i = 0; i <sizeof(buffer)-1; i++)
    {
        buffer[i] = *(s + i);
    }
    return buffer;
}
int main(int argc, char **argv)
{
    printf("%s\n", myString());
    return 0;
}
```
程序的输出结果是（　　）。

A．Hello　　B．Hello World!　　C．Well　　D．以上全部不正确

32．不能把字符串"HELLO!"赋给数组 b 的语句是（　　）。

A．char b[10]={'H'，'E'，'L'，'L'，'O'，'!'，'\0'};

B．char b[10];b="HELLO!";

C．char b[10]; strcpy(b，"HELLO!");

D．char b[10]="HELLO!";

33．下述代码定义了一个结构体：

```
struct Date
{
    char a;
    int b;
    int64_t c;
    char d;
};
Date data[2][10];
```
如果 Data 的地址是 x，那么 data[1][5].c 的地址是（　　）。

A．x+195　　B．x+365　　C．x+368　　D．x+215

34.定义一个 int 类型的指针数组，数组元素个数为 10 个。下列定义方式中，正确的是（　　）。

A．int a[10];　　B．int (*a)[10];　　C．int *a[10];　　D．int (*a[10])(int);

35．将一棵有 100 个结点的完全二叉树从根这一层开始，进行广度遍历编号，那么编号最小的叶子结点的编号是（　　）。

A．49　　B．50　　C．51　　D．52

36．当分析 XML 时，需要校验结点是否闭合，用（　　）数据结构实现比较好。

A．链表　　B．树　　C．队列　　D．栈

37．快速排序算法在序列已经有序的情况下的时间复杂度为（　　）。

A．O(nlogn)　　B．O(n2)　　C．O(n)　　D．O(n2)

38．无向图 G=（V，E），其中 V={a,b,c,d,e,f}，E={<a,b>,<a,e>,<a,c>,<b,e>,<e,f>,<f,d>, <e,d>}，对该图进行深度优先排序，得到的顶点序列正确的是（　　）。

A．a，b，e，c，d，f　　B．a，c，f，e，b，d

C．a，e，b，c，f，d　　D．a，e，d，f，c，b

二、编程题

编写一个函数，根据两文件的绝对路径算出相对路径，例如：

a="/qihoo/app/a/b/c/d/new.c"，b="/qihoo/app/1/2/test.c"，那么 b 相对于 a 的相对路径是"../../../../1/2/test.c"。

真题 4　某知名搜索引擎提供商软件工程师笔试题

一、简答题

1．进程和线程的区别是什么？

2．存储过程是什么？它有哪些优点？

3．static 关键字的作用是什么？static 全局变量与普通全局变量的区别是什么？static 局部变量与普通变量的区别是什么？static 函数与普通函数的区别是什么？

二、算法与程序设计题

1．实现内存拷贝函数 memcpy。

2．在一个二维数组中，每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请实现一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。

例如，下面的二维数组就是符合这种约束条件的。如果在这个数组中查找数字 7，则返回 true；如果查找数字 5，由于数组中不含有该数字，则返回 false。

1	2	8	9
2	4	9	12
4	7	10	13
6	8	11	15

3．定义栈的数据结构，请在该类型中实现一个能够得到栈的最小元素的 min 函数。在该栈中，调用 min、push 及 pop 的时间复杂度都是 O(1)。

三、系统设计题

微博中的 url 往往很长，发送前要转化为 tinyurl。

1）url 如何转为 tinyurl 编码？

2）如果用户输入一个已经转换过的 url，如何快速定位到已经生成了的 tinyurl？

3）如果数据为 10 亿条，需要 10 个 tinyurl 服务器，那么如何设计？

真题 5　某知名社交软件公司软件工程师笔试题

一、单选题

1．如果等式 12×25=311 成立，那么使用的是（　　）进制运算。

A．七　　B．八　　C．九　　D．十一

2．在某 32 位系统下，C++程序如下所示：

char str[] = "http://www.tianya.com"（长度为 21）

```
char *p = str;
sizeof (str) = ？①
sizeof (p) = ？②
void Foo(char str[100])
{
    sizeof(str) = ？③
}
void *p = malloc(100);
sizeof (p) = ？④
```

①、②、③、④处分别填写的值为（　　）。

A．22，22，100，100　　B．4，4，4，4

C．22，4，4，4　　D．22，4，100，4

3．有字符序列（Q,H,C,Y,P,A,M,S,R,D,F,X），那么新序列（F,H,C,D,P,A,M,Q,R,S,Y,X）是（　　）算法一趟扫描的结果。

A．堆排序　　B．快速排序　　C．希尔排序　　D．冒泡排序

4．下列关于排序算法的描述中，正确的是（　　）。

A．快速排序的平均时间复杂度为 O(nlogn)，最坏时间复杂度为 O(nlogn)

B．堆排序的平均时间复杂度为 O(nlogn)，最坏时间复杂度为 O(n2)

C．冒泡排序的平均时间复杂度为 O(n2)，最坏时间复杂度为 O(n2)

D．归并排序的平均时间复杂度为 O(nlogn)，最坏时间复杂度为 O(n2)

5．假设要存储一个数据集，数据维持有序，对其只有插入、删除和顺序遍历操作，综合存储效率和运行速度考虑，下列数据结构中最适合的是（　　）。

A．数组　　B．链表　　C．散列表　　D．队列

6．假设有 n 个关键字具有相同的散列函数值，则用线性探测法把这 n 个关键字映射到散列表中需要执行的线性探测次数为（　　）。

A．n2　　B．n×(n+1)　　C．n×(n+1)/2　　D．n×(n-1)/2

7．下列不属于数据库事务正确执行的四个基本要素的是（　　）。

A．隔离性　　B．持久性　　C．强制性　　D．一致性

8．下列进程状态变化中，不可能发生的是（　　）。

A．运行→就绪B．运行→等待　C．等待→运行　　D．等待→就绪

9．下列选项中，不可以查看某 IP 是否可达的方式/命令是（　　）。

A．telnet　　B．ping　　C．tracert　　D．top

10．当用一台机器作为网络客户端时，该机器最多可以保持（　　）个到服务端的连接。

A．1　　B．少于 1024　　C．少于 65535　　D．无限制

二、填空题

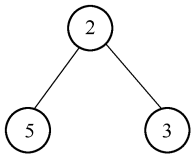
1．根据访问根结点的次序，二叉树的遍历可以分为三种：前序遍历、（　　）遍历和后序遍历。

2．由权值分别为 3、8、6、2、5 的叶子结点生成一棵赫夫曼树，它的带权路径长度为（　　）。

三、问答题

1．给定一棵二叉树，求各个路径的最大和，路径可以以任意结点作为起点和终点。

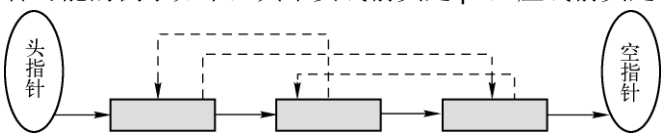
例如，给定以下二叉树：



返回 10，代码如下：

```
/**
*二叉树的定义
* struct TreeNode {
*     int val;
*     TreeNode *left;
*     TreeNode *right;
*     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
* };
*/
int maxPathSum(TreeNode *root)
```

2．有一个链表，其中每个对象包含两个指针 p1、 p2，其中指针 p1 指向下一个对象，指针 p2 也指向一个对象，沿 p1 可以像普通链表一样完成顺序遍历，沿 p2 则可能会有重复。 一种可能的例子如下，其中实线箭头是 p1，虚线箭头是 p2。



请设计函数，翻转这个链表，并返回头指针。链表结点的数据结构如下：

```
struct Node{
    Node * p1;
    Node * p2;
    int data;
};
```

函数定义如下：

```
Node * revert(Node* head);
```

3．编辑距离又称为 Levenshtein 距离，是指两个子串之间由一个转成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符、插入一个字符和删除一个字符。请实现一个算法来计算两个字符串的编辑距离，并计算其复杂度。在某些应用场景下，替换操作的代价比较高，假设替换操作的代价是插入和删除的两倍，该如何调整算法？

真题 6　某知名游戏软件开发公司软件工程师笔试题

一、单选题

1．下列命令中，可以用来查看当前系统启动时间的是（　　）。

A．w　　B．top　　C．ps　　D．uptime

2．Linux 系统下的进程有以下（　　）三种状态。

A．精确态、模糊态和随机态　　B．运行态、就绪态和等待态

C．准备态、执行态和退出态　　D．手动态、自动态和自由态

3．如果系统的 umask 设置为 244，那么创建一个新文件后，它的权限是（　　）。

A．--w-r--r--　　B．-r-xr--r--　　C．-r---w--w-　　D．-r-x-wx-wx

4．下列关于地址转换的描述中，错误的是（　　）。

A．地址转换解决了因特网地址短缺所面临的问题

B．地址转换实现了对用户透明的网络外部地址的分配

C．使用地址转换后，对 IP 包加长，快速转发不会造成太大影响

D．地址转换内部主机提供一定的“隐私”

5．下列给定地址中，与 192.168.1.110/27 不属于同一个子网的主机地址是（　　）。

A．192.168.1.94　　B．192.168.1.96　　C．192.168.1.124　　D．192.168.1.126

6．ping 命令使用 ICMP 的以下（　　）代码类型。

A．重定向　　B．echo 响应　　C．源抑制　　D．目标不可达

7．下列关于传输层协议 UDP 的描述中，正确的是（　　）。

A．比较合适传输小的数据文件　　B．提高了传输的可靠性

C．提供了高的传输效率　　D．使用窗口机制来实现流量控制

8．下列功能中，能使 TCP 准确、可靠地从源设备到目地设备传输数据的是（　　）。

A．封装　　B．流量控制　　C．无连接服务　　D．编号和定序

9．在 bash 中，下列说法正确的是（　　）。

A．$#表示参数的数量　　B．$$表示当前进程的名字

C．$@表示当前进程的 pid　　D．$?表示前一个命令的返回值

10．在 bash 中，需要将脚本 demo.sh 的标准输出和标准错误输出重定向至文件 demo.log，则下列用法中，正确的是（　　）。

A．bash demo.sh &>demo.log　　B．bash demo.sh>&demo.log

C．bash demo.sh >demo.log 2>&1　　D．bash demo.sh 2>demo.log 1>demo.log

真题详解篇

真题详解篇主要针对 17 套真题进行深度剖析，以庖丁解丁式的写法，针对每一道题目给出非常详细的解答，力求在授之以鱼的同时授之以渔——不仅告诉答案，还告诉读者同类型题目以后再遇到了该如何解答。读者在学完了基础知识以后，可以通过完成本书中的习题，找出自己的知识盲区，进而查漏补缺，达到熟练掌握这些知识的目的。

真题详解 1 某知名互联网下载服务提供商软件工程师笔试题

一、单选题

1．答案：B。

分析：本题考查的是前置++操作与后置++操作的区别。

对于变量 a，若使用前置运算符修饰（++a），则表示先取 a 的地址，增加它的内容，然后再把值放在寄存器中；若使用后置运算符（a++）修饰，则表示先取 a 的地址，再把它的值装入寄存器，然后增加内存中 a 的值。一般而言，当涉及表达式计算时，对这两种情况的计算过程区分如下：**后置的++运算符是先将其值返回，然后其值增 1；而前置的++运算符，则是先将值增 1，再返回其值**。

对于本题而言，语句 int a = i++的执行过程如下：首先，取 i 的值返回并赋值给 a，此时 a 的值变为 0，然后，i 的值变为 1。语句 int b=++a 的执行过程如下：首先，a 执行自增操作，a 的值变为 1，然后再把自增后的值赋值给 b，因此，变量 b 的值为 1。接着执行语句 c=a+b=0+1=1，对于表达式 a?b:c 的值而言，因为变量 a 的值为 0，等价于 false，所以，a=c=1。因此，选项 B 正确。

所以，本题的答案为 B。

2．答案：A。

分析：本题考查的是关键字 sizeof 的知识。

sizeof 是 C/C++语言的关键字，它以字节的形式给出其操作数的存储大小。对于本题而言，p 是一个指针，在 32 位环境下，指针只占用 4 个字节。所以，选项 A 正确。

如果题目改成 int p[10]，那么 sizeof(p)的值则是 40，因为此时 p 是一个数组类型，即数组中存放了 10 个 int 类型的元素，在 32 位环境下，每个 int 类型的变量占用 4 个字节，所以这个数组将会占用 40 个字节。

所以，本题的答案为 A。

引申：在程序员笔试中经常会考察结构体变量 sizeof 的值，而这类题的出错率往往较高。下面重点介绍 struct 的 sizeof 值的求解方法。

结构体（struct）是一种复合数据类型，其构成元素既可以是基本数据类型（例如 int、double、float、short、char 等），也可以是复合数据类型（例如数组、struct、union 等数据单元）。

一般而言，struct 的 sizeof 是将所有成员对齐后长度相加，而联合体（union）的 sizeof 取的是最大的成员变量长度。

在结构体中，编译器为结构体的每个成员按其自然边界（alignment）分配空间。各个成员按照它们被声明的顺序在内存中顺序存储，第一个成员的地址和整个结构体的地址相同。

字节对齐也称为字节填充，它是 C/C++编译器的一种技术手段，主要是为了在空间与复杂度上达到平衡。简单地讲，是为了在可接受的空间浪费的前提下，尽可能地提高对相同运算过程的最少（快）处理。字节对齐的作用不仅便于 CPU 的快速访问，使 CPU 的性能达到最佳，还通过合理地利用字节对齐来有效地节省存储空间。例如，32 位计算机的数据传输值是 4 字节，64 位计算机的数据传输值是 8 字节，这样，在默认的情况下，编译器会对 struct 进行（32 位机）4 的倍数或（64 位机）8 的倍数的数据对齐。对于 32 位机来说，4 字节对齐能够使 CPU 访问速度提高，例如一个 long 类型的变量，如果跨越了 4 字节边界存储，那么 CPU 要读取两次，这样效率就低了，但需要注意的是，如果在 32 位计算机中使用 1 字节

或者 2 字节对齐，不仅不会提高效率，反而会使变量访问速度降低。

在默认情况下，编译器为每一个变量或数据单元按其自然对齐条件分配空间。一般地，可以通过下面的方法来改变默认的对齐条件。

1）使用伪指令#pragma pack (n)，C 语言编译器将按照 n 个字节对齐。

2）使用伪指令#pragma pack ()，取消自定义字节对齐方式。

3）另外，还有一种方式：_attribute((aligned (n)))，让所作用的结构体成员对齐在 n 字节的自然边界上。如果结构体中有成员的长度大于 n，则按照最大成员的长度来对齐。_attribute_((packed))用于取消结构在编译过程中的优化对齐，按照实际占用字节数进行对齐。

例如下面的数据结构：

```
struct test
{
    char x1;
    short x2;
    float x3;
    char x4;
};
```

由于编译器默认情况下会对 struct 进行边界对齐，结构的第一个成员 x1，其偏移地址为 0，占据了第 1 个字节，第二个成员 x2 为 short 类型，其起始地址必须 2 字节对齐，因此，编译器在 x2 和 x1 之间填充了一个空字节。struct 的第三个成员 x3 和第四个成员 x4 恰好落在其自然边界地址上，在它们前面不需要额外的填充字节。在 test 结构体中，成员 x3 要求 4 字节对齐，是该结构所有成员中要求的最大边界单元，因而 test 结构的自然对界条件为 4 字节，所以编译器在成员 x4 后面填充了 3 个空字节。整个结构所占据空间为 12 字节。

再如，有如下数据结构：

```
struct s1
{
    short d;
    int a;
    short b;
}a1;
```

在 32 位计算环境下，short 型占 2 个字节，int 型占 4 个字节，所以，为了满足字节对齐的要求，变量 d 除了自身所占用的 2 个字节外，还需要再填充 2 个字节，变量 a 占用 4 个字节，变量 b 除了自身占用的 2 个字节，还需要 2 个填充字节，所以，最终 s1 的 sizeof 值为 12。

字节对齐的细节和编译器实现相关，但一般而言，满足以下三个准则。

1）结构体变量的首地址能够被其最宽基本类型成员的大小所整除。

2）结构体每个成员相对于结构体首地址的偏移量（offset）都是成员大小的整数倍，如有需要，编译器会在成员之间加上填充字节。

3）结构体的总大小为结构体最宽基本类型成员大小的整数倍，如有需要，编译器会在最末一个成员之后加上填充字节。

需要注意的是，基本类型是指前面提到的像 char、short、int、float、double 这样的内置数据类型，这里所说的“数据宽度”就是指其 sizeof 的大小，在 32 位的计算环境下，这些基本数据类型的 sizeof 大小分别为 1、2、4、4、8。由于结构体的成员可以是复合类型的，因此在寻找最宽基本类型成员时，应当包括复合类型成员的子成员，而不是把复合成员看作一个整体，例如，一个结构体中包含另外一个结构体成员，那么此时最宽基本类型成员不是该结构体成员，而是取基本类型的最宽值。但在确定复合类型成员的偏移位置时，则是将复合类型作为整体看待，即复杂类型（如结构体）的默认对齐方式是它最长的成员的对齐方式，

这样在成员是复杂类型时，可以最小化长度，达到程序优化的目的。

3．答案：D。

分析：本题考查的是 sizeof 以及字符串的存储方式。

sizeof 是 C/C++语言中的关键字，它以字节的形式给出了其操作数的存储大小。

本题中，语句 char str[] = "abcde"在内存中实际存储为一个字符数组：{'a', 'b', 'c', 'd', 'e', ' \0'}，最容易忽视的就是字符串末尾的结束符'\0'。同时，需要将 strlen 与 sizeof 进行区别，strlen 执行的是一个计数器的工作，它从内存的某个位置（可以是字符串开头、中间某个位置，甚至是某个不确定的内存区域）开始扫描，直到碰到第一个字符串结束符'\0'为止，然后返回计数器值。很显然，本题中，字符串 str 的 strlen 的值为 5，而字符串 str 的 sizeof 的值为 6，因为除了实际能看到的字符外，数组中还存储了字符串的结束符'\0'，这个结束符也需要占用一个存储空间。因此，选项 D 正确。

所以，本题的答案为 D。

4．答案：D。

分析：本题考查的是递归知识。

在分析递归的问题时，最重要的是找出递归表达式与递归结束的条件。本题中，递归调用的过程为 func(5)=5×func(4)=5*4*func(3)=5*4*3*func(2)=5*4*3*2*func(1)=5*4*3*2*1=120。这个函数实际上是用来求 i 的阶乘的。所以，选项 D 正确。

所以，本题的答案为 D。

5．答案：C。

分析：本题考查的是构造函数与析构函数知识。

构造函数是一种特殊的函数，用来在对象实例化时初始化对象的成员变量。通常构造函数具有以下特点。

1）构造函数必须与类的名字相同，并且不能有返回值（返回值也不能为 void）。

2）每个类可以有多个构造函数。当开发人员没有提供构造函数时，编译器会提供一个无参数的构造函数，但该构造函数不会执行任何代码。如果开发人员提供了构造函数，那么编译器就不会再提供默认的构造函数了。

3）构造函数是可以重载的，也就是说，程序中可以定义多个有不同参数的构造函数。

与构造函数相反的是析构函数，析构函数也是一个特殊的成员函数，它的名字是类名的前面加一个“～”符号，即“～类名()”，当对象的生命期结束时，会自动执行析构函数。从析构函数的定义可以看出，一个类只能有一个析构函数。

对于本题而言，对于选项 A，当用户自定义一个有参数的构造函数时，编译器就不会再提供默认的构造函数了，因此这个类就只有这一个有参数的构造函数。所以，选项 A 错误。

对于选项 B，一个类可以有多个拷贝构造函数，例如，有如下示例代码：

```
class C
{
public:
    C(const C&);     //const 拷贝
    C(C&);                  //非 const 的拷贝
}
```

所以，选项 B 错误。

对于选项 C，由于构造函数可以重载，因此，可以定义多个构造函数。所以，选项 C 正确。

对于选项 D，从析构函数的定义就可以看出，一个类只能定义一个析构函数（析构函数名是固定的，并且析构函数不能有参数）。所以，选项 D 错误。

所以，本题的答案为 C。

6．答案：A。

分析：本题考查的是类的默认访问权限。

在 C++语言规范中，规定成员默认访问属性为 private。所以，选项 A 正确。

所以，本题的答案为 A。

7．答案：B。

分析：本题考查的是类成员变量在继承中的访问权限。

public（公有）继承、protected（保护）继承和 private（私有）继承是三种常见的继承方式，三者的具体使用方式如下。

1）public（公有）继承。基类的 public 与 protected 成员对子类都可见，但是只有 public 成员对子类的对象是可见的。也就是说，子类可以访问父类的 public 与 protected 成员，但是子类的对象无法访问基类的 protected 成员。

2）protected（保护）继承。基类的 public 与 protected 成员对子类都可见，但是基类的成员对子类的对象都不可见。因为基类的 public 和 protected 成员都变成了子类的 protected 成员，显然一个对象无法直接访问 protected 成员。

3）private（私有）继承。基类的 public 与 protected 成员对子类都可见，但是基类的成员对子类的对象都不可见。因为基类的 public 和 protected 成员都变成了子类的 private 成员，显然一个对象无法直接访问 private 成员。

下表为成员访问控制列表。

基类性质	继承类型	派生类性质
public	public	public
protected	public	protected
private	public	不能访问
public	protected	protected
protected	protected	protected
private	protected	不能访问
public	private	private
protected	private	private
private	private	不能访问

从上表分析可知，选项 B 正确。

所以，本题的答案为 B。

8．答案：B。

分析：本题考查的是二分查找的知识。

通常，对一个有序数组进行查找的最好方法是二分查找法。

二分查找的过程如下（假设表中元素是按升序排列的）：先将表中间位置记录的关键字与查找关键字比较，如果两者值相等，则查找成功；否则，利用中间位置记录将表分成前、后两个子表，如果中间位置记录的关键字的值大于查找关键字的值，则进一步查找前一子表，否则，进一步查找后一子表。重复以上过程，直到找到满足条件的记录使查找成功或直到子表不存在为止，此时查找不成功。

通过以上分析可知，二分查找的时间复杂度为 O(logn)。所以，选项 B 正确。

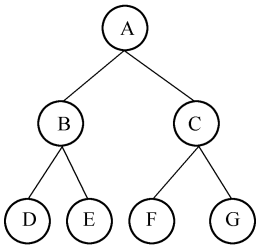
所以，本题的答案为 B。

9．答案：C。

分析：本题考查的是二叉树的遍历。

遍历（Traversal）指的是沿着某条搜索路线，依次对树或图中每个结点均做且仅做一次访问。树的遍历是树的一种重要运算，主要有先序遍历、中序遍历及后序遍历三种不同的遍历方式。

下面通过下图来介绍二叉树的三种遍历方式。



1）先序遍历。先遍历根结点，再遍历左子树，最后遍历右子树。上图的先序遍历序列为 ABDECFG。

2）中序遍历。先遍历左子树，再遍历根结点，最后遍历右子树。上图的中序遍历序列为 DBEAFCG。

3）后序遍历。先遍历左子树，再遍历右子树，最后遍历根结点。上图的后序遍历序列为 DEBFGCA。

从上述分析可以很容易得到本题后序遍历的结果为丙丁乙己戊甲。

所以，选项 C 正确。

所以，本题的答案为 C。

10．答案：D。

分析：本题考查的是 new 与 delete 的用法。

new 与 delete 是 C++语言中预定的操作符，它们一般需要配套使用。new 用于从堆中申请一块空间，一般用于动态申请内存空间，即根据程序需要，申请一定长度的空间；而 delete 则用于释放 new 申请的内存空间。

具体而言，在使用 new 创建对象时，会调用这个对象的构造函数。本题中，语句 A *pa = new A[10]表示创建了 10 个 A 类型的对象，因此，调用了 10 次构造函数，数组一旦创建出来，指针 pa 就指向了数组的首元素，因此，在调用 delete pa 语句时，只调用了 A[0]的析构函数，而对后面 9 个对象没有调用析构函数，这可能会导致内存泄漏（内存泄漏指的是分配函数动态开辟的空间在使用完毕后未释放，导致一直占据该内存单元，直到程序结束）。正确的调用方法应该是 delete[] pa，此时 10 个对象的析构函数就都会被调用。因此，在使用 new 时，必须把 new/delete 和 new[]/delete[]配对使用。所以，选项 D 正确。

所以，本题的答案为 D。

11．答案：A。

分析：本题考查的是在继承中构造函数与析构函数的知识。

通常，程序在结束时会先调用派生类的析构函数，然后再调用基类的析构函数。如果析构函数不是虚函数，而当程序执行时又要通过基类的指针去销毁派生类的动态对象，那么在用 delete 销毁对象时，只调用了基类的析构函数，未调用派生类的析构函数。这样会造成销毁对象不完全。

对于本题而言，析构函数没有被定义为虚函数，而 pa 是一个指向基类的指针，因此，在执行 delete pa 语句时，只会调用基类的析构函数。所以，选项 A 正确。

所以，本题的答案为 A。

12．答案：D。

分析：本题考查的是关键字 unsigned 的知识。

本题可以采用排除法解答：

假设 file_length=5，block_length=2，则可以分成 3 块，排除选项 A。

假设 file_length=6，block_length=2，也可以分成 3 块，排除选项 B。

对于选项 C，两个 unsigned 类型的值相加可能会溢出，所以，可以排除选项 C。

对于选项 D，上述两种情况都能得到正确的结果。因此，选项 D 正确。

所以，本题的答案为 D。

13．答案：A。

分析：本题考查的是位运算中的异或运算的知识。

按位异或就是在同一位置上，当两者相同时，结果位为 0；当两者不同时，结果位为 1。也就是说，0^0=0，1^1=0，1^0=1，0^1=1。

对于本题而言，可以先计算高位的异或值：F^A=1111^1010=0101，0101 转换为十进制后，该值为 5，符合条件的答案里面只有选项 A，从而排除了选项 B、选项 C、选项 D 的可能性。

所以，本题的答案为 A。

14．答案：D。

分析：本题考查的是虚函数的知识。

被 virtual 关键字修饰的成员函数就是虚函数，虚函数的作用是实现多态性（Polymorphism）。当类中定义了虚函数后，在实例化的对象中会有一个额外的指针 vptr 指向一个虚函数表 vtable，虚函数表中存放了实际调用函数的地址。

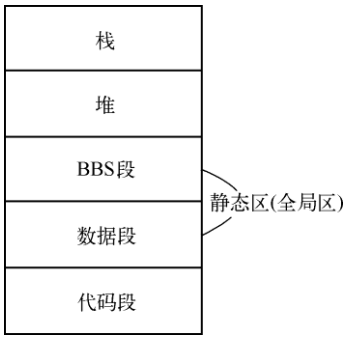
对于本题而言，由于 output 是一个虚函数，在调用时，需要先根据 vptr 找到 vtable，然后在 vtable 中查找实际要调用的 output 函数的地址。但是，调用 memset 函数会把 vptr 指针的值改为 0，因此，通过 vptr 无法找到 vtable，这会导致程序运行出错。所以，选项 D 正确。

所以，本题的答案为 D。

15．答案：B。

分析：本题考查的是 C++内存管理的知识。

一个 C/C++编译的程序所占用的系统内存一般分为 BBS 段、数据段、代码段、堆和栈，如下图所示。



1）符号起始的区块（Block Started by Symbol，BSS）段：BSS 段通常是指用来存放程序中未初始化的全局数据和静态数据的一块内存区域。BSS 段属于静态内存分配，程序结束后静态变量资源由系统自动释放。

2）数据段（Data Segment）：数据段通常是指用来存放程序中已初始化的全局变量的一块内存区域。数据段也属于静态内存分配，因此，BBS 段与数据段都属于静态区（全局区）。

3）代码段（Code Segment/Text Segment）：代码段有时也叫文本段，通常是指用来存放程序执行代码（包括类成员函数和全局函数以及其他函数代码）的一块内存区域，这部分区域的大小在程序运行前就已经确定，并且内存区域通常是只读，某些架构也允许代码段为可写，即允许修改程序。在代码段中，也有可能包含一些只读的常数变量，如字符串常量。这个段一般是可以被共享的，如在 Linux 操作系统中打开了两个 vi 来编辑文本，那么一般来说，这两个 vi 是共享一个代码段的。

4）堆（Heap）：堆是用于存放进程运行中被动态分配的内存段，其大小并不固定，可动态扩张或缩减。当进程调用 malloc 或 new 等函数分配内存时，新分配的内存就被动态添加

到堆上（堆被扩张），当利用 free 或 delete 等函数释放内存时，被释放的内存从堆中被删除（堆被缩减）。堆一般由程序员分配释放，如果程序员自己不释放，在程序结束时，该块内存空间可能会由操作系统回收。需要注意的是，它与数据结构中的堆是两回事，其分配方式类似于链表。

5）栈（Stack）：栈上存放的是用户临时创建的局部变量，一般包括函数括弧“{}”中定义的变量（但不包括 static 声明的变量，static 意味着在数据段中存放变量）。除此之外，在函数被调用时，其参数也会被压入发起调用的进程栈中，并且等到调用结束后，函数的返回值也会被存放回栈中。栈由编译器自动分配释放，存放函数的参数值、局部变量的值等。其操作方式类似于数据结构中的栈。栈内存分配运算内置于处理器的指令集中，一般使用寄存器来存取，效率很高，但分配的内存容量有限。

通过上述描述可知，选项 B 正确。

所以，本题的答案为 B。

16．答案：D。

分析：本题考查的是 C++语言中常见容器的知识。

容器是一组相同类型对象的集合，它实现了比数组更复杂的数据结构、比数组更强大的功能，C++标准模板库（Standard Template Library，STL）里提供了 10 种通用的容器类。vector（向量）中的元素是按照插入的顺序排列的；deque 是队列，队列中的元素是按照进队列的顺序排列的；list 中的元素也是无序的；为了能够具有较高的查询效率，map 内部采用了平衡二叉树进行排列，因此，它是排好序的。所以，选项 D 正确。

所以，本题的答案为 D。

17．答案：D。

分析：本题考查的是二维数组的知识。

对于二维数组的初始化，必须指定其列的个数。

本题中，申请并初始化了一个四行三列的二维数组，第一行中除了第一列的值为 1 以外，其余列的值都为 0；第二行前两列的值分别为 3 和 2，第三列的值为 0；第三行初始化的值分别为 6、7、8；第四行第一列的值为 9，其余两列的值为 0，如下所示。

1	0	0
3	2	0
6	7	8
9	0	0

因此，a[2][1]=7。所以，选项 D 正确。

所以，本题的答案为 D。

18．答案：B。

分析：本题考查的是头文件的知识。

对于选项 A，对于#include <filename.h>引用形式，编译器先从标准库路径开始搜索，然后再从本地目录搜索。所以，选项 A 错误。

对于选项 B，对于#include "filename.h"引用形式，编译器先从用户的工作目录开始搜索（用户的工作目录是通过编译器指定的），然后再去系统路径寻找。所以，选项 B 正确。

对于选项 C，在 C/C++语言中，头文件只能存放全局变量的声明，其定义只能放在扩展名为“.c”和“.cpp”的文件中。所以，选项 C 错误。

对于选项 D，一般而言，在开发大型项目时，会把不同的声明放在不同的头文件中。所以，选项 D 错误。

所以，本题的答案为 B。

19．答案：B。

分析：本题考查的是二叉树知识。

二叉树有如下性质：对于一棵非空的二叉树，度为 0 的结点（即叶子结点）总是比度为 2 的结点多一个，即如果叶子结点（度为 0 的结点）数为 n_0，度数为 2 的结点数为 n_2，则有 $n_0=n_2+1$。

对于本题而言，假设度为 i 的结点的个数为 n_i，则 $n_0=n_2+1$，所以，$n_0+n_1+n_2=n_0+n_1+n_{0-1}=699$，可以得到：$n_0=(700-n_1)/2$，显然，$n_1$ 只能是偶数。由于在完全二叉树中，度为 1 的结点只有 0 个或 1 个两种情况，因此，$n_1=0$，$n_0=350$，所以，叶子结点个数为 350。所以，选项 B 正确。

所以，本题的答案为 B。

20．答案：A。

分析：本题考查的是数组和指针的知识。

通常，字符串可以被看作字符数组，因此，本题中第四个字符也等价于数组下标为 3 的值（数组下标从 0 开始），因此，选项 C 与选项 D 错误。

对于选项 A，数组下标通常从 0 开始，p_str[3]可以用来访问字符串中的第四个字符，因此，选项 A 正确。

对于选项 B，正确的写法为*(p_str +3)='a'，因此，选项 B 错误。

所以，本题的答案为 A。

21．答案：B。

分析：本题考查的是内联函数的知识。

内联函数是指用关键字 inline 修饰的函数。在类中定义的函数被默认成内联函数。从源代码层看，内联函数具有函数的结构，而在编译后却不具备函数的性质。

内联函数不是在调用时发生控制转移，而是在编译时将函数体嵌入在每一个调用处，因此，选项 B 正确，选项 D 错误。private、public 与 protected 类型的成员函数都可以被定义为内联函数，因此，选项 A 和选项 C 错误。

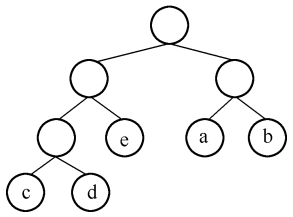
所以，本题的答案为 B。

二、多选题

1．答案：A、C、D。

分析：本题考查的是赫夫曼编码的知识。

下图给出其中的一种赫夫曼树。



从上图的二叉树可以得到字母的赫夫曼编码为“a:10，b:11，c:000，d:001，e:01”。在求解赫夫曼编码时，由于同一个结点的左右两个孩子的顺序是任意的，因此，赫夫曼编码不是固定的，如把结点 a 和结点 b 的位置交换后，a:11，b:10，但是在任何情况下编码的长度（位数）是固定的。

对于选项 A，一个字符的编码最多只占用了 3 位，而且出现次数越多，编码的长度越短。因此，用编码值来存储这段文本可以花费最少的存储空间，所以，选项 A 正确。

从上面分析可知，选项 B 错误，选项 C 正确。

对于选项 D，编码最长的位数为 3，最短的位数为 2，显然，字符 b 的编码是最短的，字符 d 的编码是最长的，所以，选项 D 正确。

所以，本题的答案为 A、C、D。

2．答案：A、D。

分析：本题考查的是 C/C++语言中的基本运算知识。

按位运算只适用于字符型和整数型变量以及它们的变体，对其他数据类型不适用。本题中，由于 d 是 double 类型的变量，因此它不适用于移位操作，所以，选项 A 与选项 D 错误。

对于选项 B，在算术运算式中，低类型能够转换为高类型，因此，在计算 d/n 时，会先把 n 隐式转换为 double 类型，然后参与计算，所以，选项 B 正确。

对于选项 C，&&是对 bool 表达式求与运算。在选项 C 中，0 代表 false，非 0 代表 true，!d && (n−3)=false&&fase=false，所以，选项 C 正确。

所以，本题的答案为 A、D。

3．答案：B、C。

分析：本题考查的是 C/C++语言中的流程控制语句知识。

本题中，对于选项 A，while 循环语句的执行过程为：判断 while 条件是否成立，如果成立，则进入循环体执行；否则，不进入循环。若第一次判断 while 条件为 false，则程序不会进入 while 循环体，此时循环体执行 0 次。所以，选项 A 错误。

对于选项 B，do-while 循环的执行过程为：先进入循环体执行一次，然后判断 while 条件是否为 true，如果为 true，则执行下一次循环；否则，退出循环。把 do-while 语句改为 while 语句的最简单方法就是把循环体的内容先在循环体外执行一次，然后把 do-while 循环改成 while 循环即可，所以，选项 B 正确。

对于选项 C，关键字 continue 用来跳过循环体中剩余的代码而进入下一次循环，因此，可以出现在各种循环体内，所以，选项 C 正确。

对于选项 D，关键字 break 一般用于跳出当前循环，因此，它可以出现在循环体内，所以，选项 D 错误。

所以，本题的答案为 B、C。

4．答案：B、D。

分析：本题考查的是 C++语言模板的知识。

类模板的格式如下：

```
template<class 形参名，class 形参名，…>  class 类名
{ ... };
```

函数模板的格式如下：

```
template <class 形参名，class 形参名，.....> 返回类型函数名(参数列表)  {函数体  }
```

其中，class 可以用 typename 关键字代替。

本题中，选项 A 和选项 C 的写法完全符合类模板的定义格式，所以，选项 A 与选项 C 正确。对于选项 B，U 没有被 class 或 typename 修饰，因此，这是一种错误的写法，所以，选项 B 错误。对于选项 D，由于 typename 与 T 中间没有空格，所以，选项 D 错误。

所以，本题的答案为 B、D。

5．答案：A、B、C。

分析：本题考查的是计算机网络与通信的知识。

一般在打开网页时，需要在浏览器中输入网址，因此，需要通过网址找到访问资源的 IP 地址，从而可以把请求发送到对应的机器上，在这个过程中需要 DNS（Domain Name System，域名系统，它是互联网上作为域名和 IP 地址相互映射的一个分布式数据库，能够使用户更方便地访问互联网，而不用去记住能够被机器直接读取的 IP 数据串。通过主机名，最终得到该主机名对应的 IP 地址的过程称为“域名解析”）协议；HTTP 是用于从 Web 服务器传输

超文本到本地浏览器的传输协议。浏览器与服务器通过 HTTP 进行交互。HTTP 是应用层协议，在传输层是通过 TCP 来传输 HTTP 请求的。Telnet 是 Internet 远程登录服务的标准协议和主要方式。它为用户提供了在本地计算机上完成远程主机工作的能力。一般使用方法为通过终端登录到远处主机，因此，在浏览器打开网页的过程中用不到。

所以，本题的答案为 A、B、C。

6．答案：A、B、C、D。

分析：本题考查的是常用的散列函数的知识。

常用的构造散列函数的方法有直接定址法、数字分析法、平方取中法、折叠法、除留余数法和随机数法。下面分别对这几种方法进行介绍。

1）直接定址法：取关键字或关键字的某个线性函数值为散列地址。例如，H(key)=a*key+b，其中，a 和 b 为常数。

2）数字分析法：假设关键字是以 r 为基数（例如以 10 为基数的十进制数）的，并且散列表中可能出现的关键字都是事先知道的，则可取关键字的若干数位组成散列地址。

3）平方取中法：取关键字平方后的中间几位作为散列地址。

4）折叠法：将关键字分割成位数相同的几部分，然后取这几部分的叠加和作为散列地址。

5）除留余数法：取关键字被某个小于或等于散列表长 m 的数 p 除后所得的余数作为散列地址（f(key) = key mod p (p≤m)，m 为散列表表长）。

6）随机数法：选择一个随机函数，取关键字的随机函数值作为它的散列地址。

所以，本题的答案为 A、B、C、D。

7．答案：B、D。

分析：本题考查的是拷贝构造函数的知识。

拷贝构造函数是一种特殊的构造函数，它由编译器调用来完成一些基于同一类的其他对象的构建及初始化。拷贝构造函数的名称必须与类名称一致，函数的形式参数是本类型的一个引用变量，并且必须是引用。当用一个已经初始化过了的自定义类型对象去初始化另一个新构造的对象时，拷贝构造函数就会被自动调用；如果没有自定义拷贝构造函数，系统将会提供一个默认的拷贝构造函数来完成这个过程。拷贝构造函数的格式为：类名（const 类名&对象名），其中，关键字 const 可有可无，由于拷贝构造函数的目的是成员复制，不应修改原对象，因此，通常建议使用 const 关键字。

通过以上分析可知，选项 B 和选项 D 正确。由于拷贝构造函数没有返回值，因此，选项 A 错误。由于默认的拷贝构造函数需要在类外使用，它一定是公有的成员，因此，选项 C 错误。

所以，本题的答案为 B、D。

8．答案：B、C。

分析：本题考查的是虚函数知识。

本题中，对于选项 A，在类的继承中，当实例化子类的对象时，会首先调用基类的构造函数，如果在构造函数中调用了虚函数，此时由于子类还没有完成初始化，因此，不会去调用子类的方法，所以，在构造函数中调用虚函数，动态绑定机制不会生效，因此，选项 A 错误。

对于选项 B，在 C++语言中，经常会把析构函数定义成虚函数，为了防止在实现多态的情况下，当用一个基类的指针删除一个派生类的对象时，派生类的析构函数会被调用。因此，在析构函数中，虚函数的动态绑定机制会生效。因此，选项 B 正确。

对于选项 C，虚函数是为了多态设计的，静态成员函数独立于对象存在，无法实现多态，而且静态成员函数没有 this 指针，因此，无法找到虚函数表，所以，不能将静态函数设计为虚函数，因此，选项 C 正确。

对于选项 D，函数的 inline 属性是在编译时确定的，是静态行为。而 virtual 的性质是在

运行时确定的，是动态行为。它们是矛盾的，二者不能同时存在，因此，选项 D 错误。

所以，本题的答案为 B、C。

9．答案：A、B、C。

分析：本题考查的是重载的知识。

重载是一种可使函数、运算符等处理不同类型数据或接收不同个数的参数的方法，通过重载，函数可以存在具有相同的名字但参数列表不相同的情形（参数的个数不同，或参数的类型不同），这样的同名但不同参数的函数或者方法，互称为“重载函数”或者“重载方法”。需要注意的是，不能用返回值来区分重载函数。

通过以上分析可以发现，选项 A 中定义的方法与题目中的方法参数个数不同；选项 B 与选项 C 的方法与题目中的方法参数类型不同。而选项 D 与题目中的方法的参数个数和类型都相同。因此，选项 A、选项 B、选项 C 正确，选项 D 错误。

所以，本题的答案为 A、B、C。

三、填空题

1．答案：1、2、3、4 四行代码依次为：

```
break;
cur_node_ptr = cur_node_ptr->next;
list_tail = list_tail->next;
delete temp_ptr;
```

分析：对于空白 1，一旦满足 remainder==0，说明这个数不是素数，因此，没有必要判断这个数能否被其他数整除，可以直接跳出循环，继续判断下一个数是否为素数，故这里的代码需要能够跳出循环，跳出循环的代码为 break。

对于空白 2，本题判断一个数 n 是否为素数的方法为：只需让 n 被小于 n 的每一个素数去除，如果都不能被整除，那么 n 就是一个素数。因此，在 2 处需要取下一个素数，然后判断能否被当前遍历到的数整除，取链表下一个结点的代码为“cur_node_ptr = cur_node_ptr->next;”。

对于空白 3，当代码执行到空白 3 时，说明当前遍历到的元素值是素数，需要把这个素数添加到链表的尾部。由于链表增加了一个新的结点，因此，需要更新指向尾结点的指针：“list_tail = list_tail->next;”或“list_tail= new_node_ptr;”。

对于空白 4，由于这个函数只需返回素数的个数，因此，在函数结束时，需要把存储素数的链表中结点占用的空间释放掉。释放结点占用的空间的代码为“delete temp_ptr;”。

2．答案：

```
pa = *LA;
pb && pa->elem != pb->elem
pb
pa->next
pre->next
pre = pa;
```

分析：本题中，外层循环遍历链表 LA，每遍历到一个值时，内层循环遍历链表 LB，判断链表 LA 与链表 LB 当前遍历的元素值是否相等，如果二者相等，那么说明这个元素在两个集合中都存在，因此，需要在链表 LA 中删除当前遍历到的结点；如果遍历完链表 LB 后仍然没找到与 LA->elem 相等的元素，说明 LA->elem 只存在链表 LA 中，因此，这个结点可以留在链表 LA 中，接着遍历链表 LA 中的下一个结点。

代码中的指针 pa 用于指向集合 A 的结点；pb 指向集合 B 的结点；临时指针 q 指向需要被删除的元素；pre 用于实现删除时结点的链接，保存 pa 的前驱结点。

对于空白 1，在遍历链表 LA 的结点前，首先要初始化 pa 为链表 LA 的头结点：pa = *LA。

对于空白 2，在内层循环遍历链表 LB 时，需要判断 pa 与 pb 指向的结点的值是否相等（pa->elem != pb->elem），同时也要判断是否遍历完链表 LB（pb 或 pb!=NULL），而且必须先判断链表是否遍历完毕，再判断元素是否相等。因此，此处应该填写的代码为 pb && pa->elem != pb->elem 或 pb!=NULL&& pa->elem != pb->elem。

对于空白 3，当 if 条件满足时，if 代码体里删除了当前遍历到的结点 pa，当前遍历到的结点在两个集合中都存在，即满足 pa->elem==pb->elem，因此，在空白 2 处的 while 循环条件中，一定因不满足 pa->elem != pb->elem 而跳出循环，此时一定满足 pb!=NULL，所以，此处应该填写的代码为 pb!=NULL 或 pb。

对于空白 4，在删除结点 pa 时，如果 pa 没有前驱结点，说明它是 LA 的头结点，即 *LA=pa，则在删除结点 pa 后，LA 的链表头变为 pa->next，因此，此处应该填写的代码为：pa->next。

对于空白 5，如果 pa 有前驱结点 pre，则通过让它的前驱结点指向 pa 的后继结点这种方法来删除结点 pa，即 pre->next= pa->next。

对于空白 6，当代码进入空白 6 所在的 else 体内时，说明 pa->elem 在链表 LB 上不存在，因此，通过语句 pa = pa->next 继续遍历链表 LA 的下一个结点，同时需要更新 pa 前驱结点 pre 的值，即“pre = pa;”。

真题详解 2 某知名监控产品供应商和解决方案服务商软件工程师笔试题

一、单选题

1．答案：D。

分析：参见真题 1 中单选题 15。

2．答案：C。

分析：本题考查的是字符数组存储的知识。

在 C/C++语言中，字符串均以字符'\0'作为结束符。因此，字符串'China'在内存中实际存储的值为{'C','h','i','n','a','\0'}，一共占用 6 个字节。所以，选项 C 正确。

所以，本题的答案为 C。

3．答案：A。

分析：本题考查的是表达式运算的知识。

在解答本题之前，首先要弄懂前置++/--运算符与后置++/--运算符的运算原理以及二者之间的区别。以变量 i 为例，i++与++i 这两种方式都会使 i 自增 1，不同之处在于其内部实现，++i 运算表示先取 i 的地址，增加它的内容，然后把值放在寄存器中；而 i++表示先取 i 的地址，把它的值装入寄存器，然后增加内存中的 i 的值。《Google C++编程风格》一书中强调，对于迭代器和其他模板对象，建议使用前缀形式（++i）的自增、自减运算符，因为前置自增（++i）通常要比后置自增（i++）效率更高。

本题中，语句 d=(++a)*(c=1)的执行过程如下：执行++a 运算后，变量 a 的值变为 2，执行 c=1 运算后，变量 c 的值变为 1，而 d=2*1=2。由于变量 b 被初始化为 0 后，一直没有被修改，因此，b 的值为 0。所以，选项 A 正确。

所以，本题的答案为 A。

4．答案：B。

分析：本题考查的是多线程的知识。

多线程（Multithreading）是指在软件或者硬件上实现多个线程的并发执行。本题中，+1 操作的执行过程如下：取出变量 a，对变量 a 执行+1 操作，把计算结果放回去。如果两个线程中+1 操作都没有被中断，所有+1 操作都生效了，那么此时相应地对 a 执行了 200 次+1 操作，在这种情况下，a 的值变为 200。由于这两个线程在对 a 执行+1 操作时，并没有加锁，因此，有可能会导致部分+1 操作丢失。过程如下所示：

1）线程 1 读取变量 a 的值（读取到寄存器中）为 0。

2）线程 2 读取变量 a 的值，此时读取到的值也为 0。

3）线程 1 对 a 执行+1 操作并放回去，此时 a 的值为 1。

4）线程 2 也对 a 执行+1 操作并放回去，由于此时线程 2 中寄存器中 a 的值为 0，执行+1 操作后变为 1 并放回去，此时 a 的值还为 1。

在这种情况下，线程 1 对 a 执行+1 的操作就会丢失。因此，执行结束后，a 的最大值为 200。所以，选项 B 正确。

所以，本题的答案为 B。

5．答案：A。

分析：本题考查的是计算机网络与通信的知识。

本题中，对于选项 A，安全超文本传输协议（Hyper Text Transfer Protocol over Secure Socket Layer，HTTPS）是以安全为目标的 HTTP 通道，是 HTTP 的安全版，通过在 HTTP 下加入安全套接层（Secure Sockets Layer，SSL）实现的。而 SSL 是一种为网络通信提供安全及数据完整性的安全协议，所以，选项 A 正确。

对于选项 B，互联网协议安全（Internet Protocol Security，IPSec）是一种开放标准的框架结构，通过使用加密的安全服务以确保在互联网上进行保密而安全的通信，所以，选项 B 错误。

对于选项 C，完美隐私（Pretty Good Privacy，PGP）是一个基于 RSA（RSA 是目前最有影响力的公钥加密算法之一，它能够抵抗到目前为止已知的绝大多数密码攻击，已被 ISO 推荐为公钥数据加密标准，其中，RSA 是创始人的名字的组合）公钥加密体系的邮件加密系统，所以，选项 C 错误。

对于选项 D，安全电子交易（Secure Electronic Transaction，SET）协议是 VISA 国际组织、万事达（MasterCard）国际组织创建，联合 IBM、Microsoft、Netscope、GTE 等公司制定的一个为了在互联网上保证交易安全性的规范，主要用于解决信用卡电子付款的安全保障性问题，所以，选项 D 错误。

所以，本题的答案为 A。

6．答案：B。

分析：本题考查的是计算机网络与通信的知识。

在计算机网络与通信中，子网掩码用来指明一个 IP 地址的哪些位标识的是主机所在的子网，它的作用就是将某个 IP 地址划分成网络地址和主机地址两部分。

子网掩码是一个 32 位地址，用于屏蔽 IP 地址的一部分，以区别网络标识和主机标识，并说明该 IP 地址是在局域网上，还是在远程网上。本题中，“/20”表示 IP 地址的前 20 位都是网络号，后 12 位是主机号。由此可以确定，子网掩码为 11111111 11111111 11110000 00000000，即 255.255.240.0，所以，选项 B 正确。

所以，本题的答案为 B。

7．答案：D。

分析：本题考查的是网络安全相关知识。

网络安全涉及计算机网络上信息的保密性、完整性、可用性、真实性以及可控性，它是一个系统工程，需要仔细考虑系统的安全需求，并将各种安全技术结合在一起才能维护计算机网络以及信息的安全。

在本题中，对于选项 A，防火墙是一种保护计算机网络安全的技术性措施，它通过在网络边界上建立相应的网络通信监控系统来隔离内部和外部网络，以阻挡来自外部的网络入侵，因此，它属于网络安全控制技术，所以，选项 A 正确。

对于选项 B，防止对任何资源进行未授权的访问，从而使计算机系统在合法的范围内使用，即通过权限控制来实现网络安全控制，因此，它属于网络安全控制技术，所以，选项 B 正确。

对于选项 C，入侵检测是指“通过对行为、安全日志、审计数据或其他网络上可以获得的信息进行收集和分析，检测网络或系统中是否有违反安全策略的行为和被攻击的迹象”，通过这种技术也能实现网络安全控制，因此，它属于网络安全控制技术，所以，选项 C 正确。

对于选项 D，差错控制用于在网络传输过程中对差错进行控制，以保证数据的准确性，因此，它不属于网络安全控制技术，所以，选项 D 错误。

所以，本题的答案为 D。

8．答案：D。

分析：本题考查的是多媒体技术的知识。

多媒体技术是指通过计算机对文字、数据、图形、图像、动画、声音等多种媒体信息进行综合处理和管理，使用户可以通过多种感官与计算机进行实时信息交互的技术，又称为“计算机多媒体技术”。其中，“媒体”是指用来传递或获取信息的工具、渠道、载体、中介物或技术手段。也可以把媒体看作实现信息从信息源传递到受信者的一切技术手段。这里的“媒体”主要有两层含义：一是承载信息的物体；二是指存储、呈现、处理、传递信息的实体，所以，选项 D 正确。

所以，本题的答案为 D。

9．答案：B。

分析：本题考查的是图像处理的知识。

RGB 色彩模式是工业界的一种颜色标准，是通过对红（Red）、绿（Green）、蓝（Blue）三种颜色通道的变化以及它们相互之间的叠加来得到各种颜色，这个标准几乎包括了人类视力所能感知的所有颜色，是目前运用最广的颜色系统之一。

具体而言，RGB 是红、绿、蓝三原色构成色彩点，由于每一种颜色用 8bit（位）表示，因此，一个像素点就需要 3×8bit 来表示。而分辨率为 256×512 说明总共有 256×512 个像素点。因此，这个图像需要 256×512×3×8bit 来表示，所以，选项 B 正确。

所以，本题的答案为 B。

10．答案：B。

分析：本题考查的是计算机设备相关知识。

传声器的作用是把声音信号转换为音频模拟信号。声卡获取到传声器的音频模拟信号后，会把其转换成计算机能识别的数字信号，所以，选项 B 正确。

所以，本题的答案为 B。

11．答案：B。

分析：本题考查的是计算机系统结构知识。

本题中，对于选项 A，TCB 是 Trusted Computing Base 的简称，指的是计算机内保护装置的总体，包括硬件、固件、软件和负责执行安全策略的组合体。它建立了一个基本的保护环境并提供一个可信计算机系统所要求的附加用户服务，所以，选项 A 错误。

对于选项 B，MMU 是 Memory Management Unit 的缩写，即内存管理单元，它用来管

理虚拟存储器、物理存储器的控制线路，同时也负责将虚拟地址映射为物理地址，以及提供硬件机制的内存访问授权，所以，选项 B 正确。

对于选项 C，CACHE 是指介于中央处理器和主存储器之间的高速小容量存储器，所以，选项 C 错误。

对于选项 D，DMA 是 Direct Memory Access 的简称，即“直接内存存取”，指的是一种高速的数据传输操作，允许在外部设备和存储器之间直接读写数据，既不通过 CPU，也不需要 CPU 干预，所以，选项 D 错误。

通过上面的分析可知，MMU 可以将地址从虚拟（逻辑）地址空间映射到物理地址空间，所以，选项 B 正确。

所以，本题的答案为 B。

12．答案：A。

分析：本题考查的是计算机系统结构的知识。

RS485 总线是一种常见的串行总线标准，采用平衡发送与差分接收的方式，具有抑制共模干扰的能力。在一些要求通信距离为几十米到上千米的场合，RS485 总线是一种应用最多的总线。而且在多结点的工作系统中，它也有着广泛的应用。

具体而言，RS485 总线采用差分信号负逻辑，−2～−6V 表示“0”，+2～+6V 表示“1”。RS485 有两线制和四线制两种接线，四线制接线方式只能实现点对点的通信，现在已经很少采用，现在多采用的是两线制接线方式，这种接线方式为总线式拓扑结构，在同一总线上最多可以挂接 32 个结点。由此可见，RS485 最少有两根数据信息号，所以，选项 A 正确。

所以，本题的答案为 A。

13．答案：D。

分析：本题考查的是编译原理的知识。

编译器的主要功能就是把源代码翻译成能在机器上执行的目标代码。

对于选项 A 和选项 B，对代码的编辑和规范检查都是由编辑器来负责的，而不是编译器。所以，选项 A 与选项 B 错误。

对于选项 C，分析代码中的问题是由调试器来完成的，所以，选项 C 错误。

对于选项 D，描述正确。

所以，本题的答案为 D。

14．答案：B。

分析：本题考查的是操作系统的知识。

实时操作系统（Real-Time Operating System，RTOS）是指当外界事件或数据产生时，能够接受并以足够快的速度予以处理，其处理的结果又能在规定的时间内来控制生产过程或对处理系统做出快速响应，并控制所有实时任务协调一致运行的操作系统。换言之，实时操作系统是保证在一定时间限制内完成特定功能的操作系统。能够提供及时响应和高可靠性是其主要特点。

对于选项 A，由于 RTOS 具有实时响应的特性，因此，它的调度目标是时间响应，而不是任务之间的公平性，所以，选项 A 不正确。

对于选项 B，为了保证响应的实时性，实时操作系统采用了抢占式的调度方式，所以，选项 B 正确。

对于选项 C 与选项 D，为了保证响应的实时性，实时操作系统采用了抢占式的调度方式，而不是采用基于时间片轮转的调度方式，也不是静态优先级调度方式，所以，选项 C 与选项 D 不正确。

所以，本题的答案为 B。

二、填空题

1．答案：extern。

分析：C++语言是一种面向对象编程语言，它支持函数重载，而 C 语言是一种面向过程的编程语言，它不支持函数重载，所以，函数被 C++编译器编译后在库中的名字与 C 语言的不同。例如，如果声明一个 C 语言函数 float f(int a,char b)，C++的编译器就会将这个名字变成像_f_int_char 之类的东西以支持函数重载。然而，C 语言编译器的库一般不执行该转换，所以，它的内部名为_f，这样链接器将无法解释 C++语言对 C 语言函数 f()的调用。

C++语言提供了 C 语言 Alternate likage specifications（替代链接说明）符号 extern“C”来解决名字匹配问题，它告诉编译器在编译时对函数名用 C 语言的规则来编译而不是采用 C++的规则。使用方法为 extern 后跟一个字符串来指定想声明的函数的链接类型，后面是函数声明。示例如下：

```
extern "C" float f(int a,char b);
```

该语句的目的是告诉编译器不需要做函数名转换，直接找函数名为 f 的函数即可。

extern 还有另外一个作用，在 C/C++语言中表明函数和全局变量作用范围（可见性）的关键字，该关键字告诉编译器，其声明的函数和变量可以在模块或其他模块中使用。

2．答案：首先给出两个函数的原型。

1）int sprintf(char *str, const char *format, …)

2）int snprintf(char *str,size_t size, const char *format, …)

其中，sprintf 函数为字符串格式化函数，主要用于把格式化的数据写入某个字符串中。而 snprintf 函数是 sprintf 函数的限制字符数量的一个表达，它最多从源串中复制 size−1 个字符到目标串中，然后，在后面加一个结束符'\0'，如果执行成功，就返回源串的长度；否则，返回负值。

通过上面的分析可知，两者都是把源串复制到目标串，但使用 snprintf 函数能够防止字符串 str 的内存越界问题，使用更安全。

3．答案：1。

分析：本题考查的是条件表达式的知识。

条件表达式的一般形式如下：

```
x=<表达式 1>?<表达式 2>:<表达式 3>
```

其意义是：先求解表达式 1，若返回值为真，则求解表达式 2，将表达式 2 的值赋给 x；若返回值为假，则求解表达式 3，将表达式 3 的值赋给 x。

对于本题而言，表达式 a<b？a:c < d？a:d 等价于 a<b？a:(c < d？a:d)。如果满足条件 a<b，那么运算结果为 a；否则，运算结果为 c < d？a:d。在本题中，由于 a<b，因此这个表达式的运算结果为 a=1。如果这道题目改为 a=5，由于 a<b 不成立，因此，表达式的运算结果实际上为 c < d？a:d，由于 c<d，因此运算结果为 a=5。

4．答案：>6。

分析：本题考查的是隐式类型转换的知识。

隐式类型转换分为三种，即算术转换、赋值转换和输出转换。对于其中的算术转换，指的是如果参与运算的数字有不同的类型，那么在运算的过程中，总会把低精度的类型自动转换为高精度的类型进行运算。对于本题而言，当执行 unsigned int a=6; int b=−20 语句后，在计算表达式 a+b 的结果时，首先会把 int 型变量 b 转换为 unsigned int（无符号 int）类型（无符号 int 的值域是[0，4294967295]）的值，转换后的值为 4294967276，因此，a+b 的结果为 4294967282，该值显然大于 6，因此，输出结果为“>6”。

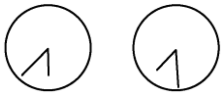
真题详解 3 某知名安全软件服务提供商软件工程师笔试题

一、不定项选择题

1．答案：D。

分析：本题考查的是数学知识。

根据题目中的描述，可以画一个表示时针与分针的图例，如下所示：



假设小明开始等待女生的那一时刻时针与分针的夹角为 θ，那么，等到时针与分针正好互换位置时，时针走过了 θ 弧度，而由于分针转动一圈表示的时间为 1h，钟表一圈是一个圆，表示的弧度值为 2π，分针因为要转若干圈才能到达时针的位置，记分钟所转圈数为 n，此时分针转过的角度则为 2πn−θ 弧度。

题目强调，“时间一分一秒地流逝，两个多小时过去了，他心仪的女生还没有出现”，通过这条信息可知，分针转了 2～3 圈，接近 3 圈，此时可知，n 值取 3，所以，时针转过的角度值为 θ，分针转过的角度值为 2π×3−θ=6π−θ。

对于时针而言，2π 代表一圈，即 12h，那么 弧度 θ 表示的时间值为 12*θ/(2π)h，对于分针而言，2π 代表一圈，即 60min，那么 6π−θ 表示的是 60*(6π−θ)/(2π)min。由于时钟走过的时间值与分钟走过的时间值所代表的时间量是一个量，故二者是相等的，由此可以构建如下等式关系：

(12θ/2π)*60 = 60*(6π−θ) / 2π

求解上述等式可知，θ=6π/13，即小明等待的时间反映在钟表上为 6π/13 弧度值，所以，小明一共等了 12*(6π/13) / 2πh，即 36/13h，合 166min。所以，选项 D 正确。

所以，本题的答案为 D。

2．答案：A。

分析：本题考查的是逻辑推理的知识。

这是一道富有挑战性的逻辑推理题，主要考察求职者的逻辑思维能力。解题的关键在于通过题中所给条件逐级推理，同时使用推理出的结果作为后续推理的条件，最终解决所有问题。

根据题目中的各类条件，分别对其进行编号：

“学生 B 不是学计算机的”①

“学计算机的出生在西安”②

“学生 B 不出生在深圳”③

“学化学的不出生在武汉”④

“学生 A 不是学化学的”⑤

“学计算机的出生在西安”⑥

根据以上六个条件可以进行如下推理。

根据①和②可以推断：学生 B 出生在武汉或深圳。a）

通过 a）和③可以推断：学生 B 出生在武汉。b）

根据①、④和 b）可以推断：学生 B 学的是英语。c）

根据 c）和⑤可以推断：学生 A 学的是计算机。d）

根据 d）和⑥可以推断：学生 A 出生在西安。e）

剩下的就是学生 C 出生在深圳，学的是化学。

所以，最后的结论为：学生 A 出生在西安，学的是计算机；学生 B 出生在武汉，学的是英语；学生 C 出生在深圳，学的是化学。可以将最后的结论带到题目中进行验证。所以，选项 A 正确。

所以，本题的答案为 A。

3. 答案：C。

分析：本题考查的是排列组合的知识。

题目要求两个人抽到的小球颜色相同，而此题有两个关键点需要注意：第一，每个人取的是两个球，而不是一个球，所以，必须要求两个球的颜色一模一样，才能称为小球颜色相同。第二，每种球的数量充足，可以理解为球的数量是无限的，不存在某一种颜色的球被全部取完后面的人无法取到的情况。由于球的颜色有五种，根据排列组合原理，在这五种情况下取的球的颜色可以分为以下两类情况：

1）取的两个球的颜色相同（每个人取的球的颜色是相同的），有五种情况。

2）取的两个球的颜色不同，C(5，2)=10，有 10 种情况。

以上两种情况合计共有 15 种情况。如果前 15 个人取的球的颜色都不相同，那么当第 16 个人取球时，必然会与前面 15 个人中的某一个相同。所以，本题的答案为 16 人。

所以，本题的答案为 C。

4．答案：C。

分析：本题考查的是排列组合的知识。

由题目可知，平面内有 11 个点，如果这些点中任意三个点都没有共线的，那么一共有 $C(11,2)=55$ 种情况，但是根据题意，连接成 48 条直线，那么可知，这 11 个点中必定有三点共线以及三点以上共线的，一共七种情况（55−48=7）。

而这七种三点共线的情况又可以划分为以下多种情况：

① 假设只有三点共线，令三点共线的直线有 x 条，那么可以组成的直线在 55 的基础上应该减去这种情况的可能性，即 $C(11,2)-x\times C(3,2)+1=48$，$3\times x=8$，由于解算出来的 x 的值不是整数，因此，此种情况不满足条件。

② 假设只有四点共线，令四点共线的直线有 x 条，那么可以组成的直线在 55 的基础上应该减去这种情况的可能性，即 $C(11,2)-x\times C(4,2)+1=48$，$6\times x=8$，由于解算出来的 x 的值不是整数，因此，此种情况不满足条件。

③ 假设只有 $n(n>4)$点共线，方法同上，也无法满足条件。

④ 若有三点共线及四点共线的两种，令三点共线的直线有 x 条，四点共线的有 y 条，则有 $C(11,3)-xC(3,2)-yC(4,2)+x+y=48$ ，即 2x+5y=7，所以，x=1，y=1。这 11 个点中，必定有一组三点共线，并且还有一组四点共线。由于三点共线、四点共线都不能组成三角形，因此，这 11 个点能组成的三角形的个数为：$C(11,3)-C(3,3)-C(4,3)=165-1-4=160$（本题不考虑三角形两边之和大于第三边的要求）。

⑤ 若有三点共线、四点共线及五点共线的三种，分析方法相同。可知方程无解，超过以上情况的多点共线的情况也不符合题意。

所以，本题的答案为 160。

所以，本题的答案为 C。

5．答案：B。

分析：本题考查的是数列的知识。

本题是一个数列找规律的题目，用于考察求职者的逻辑思维能力。

虽然此题中相邻项的商并不是一个常数，但它们是按照一定规律排列的，不难发现，本题中后一项除以前一项的结果构成一个等差数列，公差为 1/2，即除第一项以外的每一项都等于其前一项的值乘以(1+0.5n)，n 的值为从 0 开始的自然数。具体为：8×1=8，8×1.5=12，12×2= 24，24×2.5=60，根据这一规律，60 后面的数的值应为 60×3=180。所以，选项 B 正确。

所以，本题的答案为 B。

6．答案：D。

分析：本题考查的是“&”运算符的知识。

解答本题的关键在于理解 x=x&(x−1)这条语句的作用。“&”是一个二进制的运算符号，表示的是二进制的与操作。二进制与操作具有如下性质：只有当参与运算的两位同时为“1”时，其运算结果才为“1”，否则，其运算结果为 0：0&0=0，0&1=0，1&0=0，1&1=1。例如，十进制数 10，其二进制表示为 1010，当它与十进制数 9（二进制表示为 1001）执行“&”运算时，其结果为 1010 & 1001 = 1000。

对于表达式 x&(x−1)而言，其结果到底是什么呢？x 会不断地与比它小 1 的数进行与运算，每执行一次 x = x&(x−1)操作，会将 x 用二进制表示时最右边的一个 1 变为 0，因为 x−1 的二进制是把 x 的二进制最低位的 1 变成 0。这段代码的目的是就是计算 x 的二进制表示中 1 的个数。65530 对应的二进制表示为 1111 1111 1111 1010，对应的二进制中有 14 个 1。所以，选项 D 正确。

所以，本题的答案为 D。

7．答案：B。

分析：本题考查的是排序算法的知识。

读者要想解答出本题，必须对各种排序算法的原理有较为深刻的认识。下面将分别对答案中的这几种排序算法进行介绍与分析。

对于选项 A，选择排序是一种简单直观的排序算法，其基本原理如下：对于给定的一组记录，经过第一轮比较后得到最小的记录，然后将该记录与第一个位置的记录进行交换；接着对不包括第一个记录以外的其他记录进行第二轮比较，得到最小的记录并与第二个记录进行位置交换；重复上述过程，直到进行比较的记录只有一个时为止。

对于选项 B，快速排序是一种非常高效的排序算法，它采用“分而治之”的思想，把大的拆分为小的，小的再拆分为更小的。其原理为：对于一组给定的记录，通过一趟排序后，将原序列分为两部分，其中前部分的所有记录均比后部分的所有记录小，再依次对前后两部分的记录进行快速排序，递归该过程，直到序列中的所有记录均有序为止。

对于选项 C，希尔排序也称为“缩小增量排序”，其原理为：首先，将待排序的元素分成多个子序列，使得每个子序列的元素个数相对较少，对各个子序列分别进行直接插入排序，待整个待排序序列“基本有序后”，再对所有元素进行一次直接插入排序。希尔排序也是形成部分有序的序列。

对于选项 D，归并排序是利用递归与分治技术将数据序列划分成越来越小的子序列（子序列指的是在原来序列中找出一部分组成的序列），再对子序列排序，最后再用递归步骤将排好序的子序列合并成越来越大的有序序列。归并排序会在第一趟结束后，形成若干部分有序的子序列，并且长度递增，直到最后的一个有序的完整序列。

在本题中，很容易发现，第一个序列前 4 个数都小于等于 25，而后 5 个数都大于 25，很显然满足快速排序的方法，而且根据以上对各种排序算法的分析可知，选项 B 正确。

所以，本题的答案为 B。

8．答案：B。

分析：本题考查的是二叉树的知识。

本题中的二叉树并没有说明到底是一棵什么类型的二叉树（完全二叉树、满二叉树、普通二叉树，或者其他二叉树），所以，其高度存在不确定性。

定义二叉树中的结点总数为 n，当每个结点只有一棵子树时，其高度值最大（为 n）。当该二叉树为完全二叉树时，其高度值最小，为$\lceil \log_2^n \rceil+1$（其中⌈ ⌉符号表示取下整），其他情况二叉树的高度都是介于这两个值之间，即$[\lceil \log_2^n \rceil+1, n]$，不大于最大值也不小于最小值。

本题中要想求二叉树的最小高度，那么此时该二叉树为完全二叉树，其对应的高度为 $\log_2 360$，向下取整再加 1 等于 9。所以，选项 B 正确。

所以，本题的答案为 B。

9．答案：A。

分析：本题考查的是 list 排序的知识。

下面首先给出常见的排序算法的性能。

排序方法	最好时间	平均时间	最坏时间	辅助存储	稳定性	备注
简单选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	O(1)	不稳定	n 小时较好
直接插入排序	O(n)	$O(n^2)$	$O(n^2)$	O(1)	稳定	大部分已有序时较好
冒泡排序	O(n)	$O(n^2)$	$O(n^2)$	O(1)	稳定	n 小时较好
希尔排序	O(n)	O(nlogn)	O(ns)(1<s<2)	O(1)	不稳定	s 是所选分组
快速排序	O(nlogn)	O(nlogn)	$O(n^2)$	O(logn)	不稳定	n 大时较好
堆排序	O(nlogn)	O(nlogn)	O(nlogn)	O(1)	不稳定	n 大时较好
归并排序	O(nlogn)	O(nlogn)	O(nlogn)	O(n)	稳定	n 大时较好

对于选项 A，需要注意的是，在 C++语言中，list 采用的是双向列表来存储的，因此，它比较适合用快速排序（快速排序不需要随机地访问元素）。此时的时间复杂度为 O(nlogn)。所以，选项 A 正确。

对于选项 B，冒泡排序也是对数据顺序遍历，不需要随机访问，因此，它也适合对 list 进行排序，但由于算法的时间复杂度为 $O(n^2)$，没有快速排序效率高。所以，选项 B 不正确。

对于选项 C，首先需要弄清楚二分插入排序的基本思想。二分插入排序的基本思想如下：假设列表[0...n]被分成两部分，其中一部分[0...i]为有序序列，另一部分[i+1...n]为无序序列，排序的过程为：从无序序列中取一个数 d，利用二分查找算法找到 d 在有序序列中的插入位置并插入。不断重复上述步骤，直到无序序列中的元素全部插入有序序列，就完成了排序。由此可以看出，二分插入排序需要对列表中的元素进行随机访问，因此，它不适合对 list 进行排序。所以，选项 C 不正确。

对于选项 D，只有当被排序的元素满足某种特定的条件时，线性排序算法才能有较好的性能。由于 list 有非常好的通用性——对任意的数据类型都能排序，因此，线性排序算法不适用对 list 进行排序。所以，选项 D 不正确。

所以，本题的答案为 A。

10．答案：A。

分析：本题考查的是计算机网络与通信的知识。

ping 命令主要是为了检查网络是否通畅，它通过向计算机发送 Internet 控制报文协议（Internet Control Message Protocol，ICMP）应答报文并且监听回应报文的返回，以校验与远程计算机或本地计算机的连接。对于每个发送报文，ping 最多等待 1s，并打印发送和接收报文的数量。比较每个接收报文和发送报文，以校验其有效性。如果能够成功校验 IP 地址，但不能成功校验计算机名，则说明名称分析存在问题。默认情况下，发送四个回应报文，每个报文包含 64 字节的数据（周期性的大写字母序列）。

为了更好地说明 ping 的原理与应用，以下是一个完整的 ping 过程。

ping ×××.×××.×××.×××（A 到 B）实际上执行了以下步骤。

1）A：构建 ICMP 数据包 data，用 ICMP 把 data 连同 A 的 IP 交给 IP 层。

2）IP 层把 B 的 IP 作为目的地址，把 A 的 IP 作为源地址，加上其他的控制信息构建 IP 数据包。

3）获取 B 的 MAC 地址，根据 B 的 IP 地址和子网掩码，检测是否属于同一子网。

① 如果属于同一子网，直接在本网络查找。查找本机的 ARP 缓存，找到 B 对应的 MAC 地址，如果缓存中找不到，则表示二者在此之前没有进行过通信，就发一个 ARP 请求广播，得到 B 的 MAC 地址。

② 如果不属于同一个子网，则直接交给路由器处理，即获取路由器的 MAC（步骤同上）。

4）交给数据链路层，构建数据帧，发送 B。

5）B 收到数据帧后，检测数据帧的目的地址，若不是发给本机的数据帧，则丢弃；若是，则接收，然后提取出 IP 数据包给 IP 层处理，然后提取数据给 ICMP 处理，处理后，构建 ICMP 应答包，发送给 A，过程同上。

通过以上的分析，选项 A 正确。

所以，本题的答案为 A。

11．答案：D。

分析：本题考查的是编译原理的知识。

正则表达式又称正规表示法、常规表示法（Regular Expression，在代码中常简写为 regex、regexp 或 RE），它是计算机科学的一个概念。正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。

下表列出了所有元字符及其相应描述。

元字符	描述
\	将下一个字符标记为一个特殊字符/一个原义字符/一个向后引用/一个八进制转义符。例如，“\\n”匹配\n。“\n”匹配换行符。序列“\\”匹配“\”，而“\(”则匹配“(”
^	匹配输入字符串的开始位置。如果设置了 RegExp 对象的 Multiline 属性，^也匹配“\n”或“\r”之后的位置
$	匹配输入字符串的结束位置。如果设置了 RegExp 对象的 Multiline 属性，$也匹配“\n”或“\r”之前的位置

（续）

元字符	描述
*	匹配前面的子表达式零次或多次(大于等于 0 次)。例如，zo*能匹配“z”、“zo”以及“zoo”。*等价于{0,}
+	匹配前面的子表达式一次或多次(大于等于 1 次)。例如，“zo+”能匹配“zo”以及“zoo”，但不能匹配“z”。+等价于{1,}
?	匹配前面的子表达式零次或一次。例如，“do(es)?”可以匹配“do”或“does”中的“do”。?等价于{0,1}
{n}	n 是一个非负整数。匹配确定的 n 次。例如，“o{2}”不能匹配“Bob”中的“o”，但是能匹配“food”中的两个 o
{n,}	n 是一个非负整数。至少匹配 n 次。例如，“o{2,}”不能匹配“Bob”中的“o”，但能匹配“foooood”中的所有 o“o{1,}”等价于“o+”。“o{0,} ”则等价于“o*”
{n,m}	m 和 n 均为非负整数，其中 n≤m。最少匹配 n 次且最多匹配 m 次。例如，“o{1,3}”将匹配“fooooood”中的前三个 o。“o{0,1}”等价于“o? ”。请注意，在逗号和两个数之间不能有空格
?	当该字符紧跟在任何一个其他限制符（*，+，?，{n}，{n,}，{n,m}）后面时，匹配模式是非贪婪的。非贪

	婪模式尽可能少地匹配所搜索的字符串，默认的贪婪模式则尽可能多地匹配所搜索的字符串。例如，对于字符串“oooo”，“o+?”将匹配单个“o”，而“o+”将匹配所有“o”
.点	匹配除“\r\n”之外的任何单个字符。要匹配包括“\r\n”在内的任何字符，请使用像“[\s\S]”的模式
(pattern)	匹配 pattern 并获取这一匹配。所获取的匹配可以从产生的 Matches 集合得到，在 VBScript 中使用 SubMatches 集合，在 JavaScript 中则使用$0…$9 属性。要匹配圆括号字符，请使用“\(”或“\)”
(?:pattern)	匹配 pattern 但不获取匹配结果，也就是说，这是一个非获取匹配，不进行存储供以后使用。这在使用或字符“(|)”来组合一个模式的各个部分是很有用的。例如，“industr(?:y|ies)”就是一个比“industry|industries”更简略的表达式
(?=pattern)	正向肯定预查，在任何匹配 pattern 的字符串开始处匹配查找字符串。这是一个非获取匹配，也就是说，该匹配不需要获取供以后使用。例如，“Windows(?=95|98|NT|2000)”能匹配“Windows2000”中的“Windows”，但不能匹配“Windows3.1”中的“Windows”。预查不消耗字符，也就是说，在一个匹配发生后，在最后一次匹配之后立即开始下一次匹配的搜索，而不是从包含预查的字符之后开始
(?!pattern)	正向否定预查，在任何不匹配 pattern 的字符串开始处匹配查找字符串。这是一个非获取匹配，也就是说，该匹配不需要获取供以后使用。例如“Windows(?!95|98|NT|2000)”能匹配“Windows3.1”中的“Windows”，但不能匹配“Windows2000”中的“Windows”
(?<=pattern)	反向肯定预查，与正向肯定预查类似，只是方向相反。例如，“(?<=95|98|NT|2000)Windows”能匹配“2000Windows”中的“Windows”，但不能匹配“3.1Windows”中的“Windows”
(?<!pattern)	反向否定预查，与正向否定预查类似，只是方向相反。例如，“(?<!95|98|NT|2000)Windows”能匹配“3.1Windows”中的“Windows”，但不能匹配“2000Windows”中的“Windows”
x|y	匹配 x 或 y。例如，“z|food”能匹配“z”或“food”。“(z|f)ood”则匹配“zood”或“food”
[xyz]	字符集合。匹配所包含的任意一个字符。例如，“[abc]”可以匹配“plain”中的“a”
[^xyz]	负值字符集合。匹配未包含的任意字符。例如，“[^abc]”可以匹配“plain”中的“plin”
[a-z]	字符范围。匹配指定范围内的任意字符。例如，“[a-z]”可以匹配“a”～“z”内的任意小写字母字符。 注意：只有连字符在字符组内部时，并且出现在两个字符之间时，才能表示字符的范围；如果出现在字符组的开头,则只能表示连字符本身
[^a-z]	负值字符范围。匹配任何不在指定范围内的任意字符。例如，“[^a-z]”可以匹配任何不在“a”～“z”内的任意字符。
\b	匹配一个单词边界，也就是指单词和空格间的位置。例如，“er\b”可以匹配“never”中的“er”，但不能匹配“verb”中的“er”
\B	匹配非单词边界。“er\B”能匹配“verb”中的“er”，但不能匹配“never”中的“er”
\cx	匹配由 x 指明的控制字符。例如，\cM 匹配一个 Control-M 或回车符。x 的值必须为 A～Z 或 a～z 之一。否则，将 c 视为一个原义的“c”字符
\d	匹配一个数字字符，等价于[0-9]
\D	匹配一个非数字字符，等价于[^0-9]
\f	匹配一个换页符，等价于\x0c 和\cL

（续）

元字符	描述
\n	匹配一个换行符，等价于\x0a 和\cJ
\r	匹配一个回车符，等价于\x0d 和\cM
\s	匹配任何空白字符，包括空格、制表符、换页符等。等价于[\f\n\r\t\v]

\S	匹配任何非空白字符，等价于[^ \f\n\r\t\v]
\t	匹配一个制表符，等价于\x09 和\cI
\v	匹配一个垂直制表符，等价于\x0b 和\cK
\w	匹配包括下画线的任何单词字符。类似但不等价于“[A-Za-z0-9_]”，这里的“单词”字符使用 Unicode 字符集
\W	匹配任何非单词字符，等价于“[^A-Za-z0-9_]”
\xn	匹配 n，其中 n 为十六进制转义值。十六进制转义值必须为确定的两个数字长。例如，“\x41”匹配“A”。“\x041”则等价于“\x04&1”。正则表达式中可以使用 ASCII 编码
\num	匹配 num，其中 num 是一个正整数。对所获取的匹配的引用。例如，“(.)\1”匹配两个连续的相同字符
\n	标识一个八进制转义值或一个向后引用。如果\n 之前至少 n 个获取的子表达式，则 n 为向后引用；否则，如果 n 为八进制数字（0～7），则 n 为一个八进制转义值
\nm	标识一个八进制转义值或一个向后引用。如果\nm 之前至少有 nm 个获得子表达式，则 nm 为向后引用。如果\nm 之前至少有 n 个获取，则 n 为一个后跟文字 m 的向后引用。如果前面的条件都不满足，若 n 和 m 均为八进制数字（0～7），则\nm 将匹配八进制转义值 nm
\nml	如果 n 为八进制数字（0～7），且 m 和 l 均为八进制数字（0～7），则匹配八进制转义值 nml
\un	匹配 n，其中 n 是一个用四个十六进制数字表示的 Unicode 字符。例如，\u00A9 匹配版权符号（©）
\< \>	匹配词（word）的开始（\<）和结束（\>）。例如，正则表达式\<the\>能够匹配字符串“for the wise”中的“the”，但是不能匹配字符串“otherwise”中的“the”。注意：这个元字符不是所有软件都支持的
\(\)	将 \(和 \) 之间的表达式定义为“组”（group），并且将匹配这个表达式的字符保存到一个临时区域（一个正则表达式中最多可以保存 9 个），它们可以用 \1～\9 的符号来引用
|	将两个匹配条件进行逻辑“或”（Or）运算。例如，正则表达式(him|her) 匹配"it belongs to him"和"it belongs to her"，但是不能匹配"it belongs to them."。注意：这个元字符不是所有软件都支持的
+	匹配 1 或多个正好在它之前的那个字符。例如，正则表达式 9+匹配 9、99、999 等。注意：这个元字符不是所有软件都支持的
?	匹配 0 或 1 个正好在它之前的那个字符。注意：这个元字符不是所有软件都支持的
{i} {i,j}	匹配指定数目的字符，这些字符是在它之前的表达式定义的。例如，正则表达式 A[0-9]{3} 能够匹配字符“A”后面跟着正好 3 个数字字符的串，如 A123、A348 等，但是不匹配 A1234。而正则表达式[0-9]{4,6} 匹配连续的任意 4 个、5 个或 6 个数字

根据以上描述可知，本题中的正则表达式表示的是无符号数集合。

对于选项 A 和选项 B，在正则表达式 number -> digits optionalFraction optionlExponent 中，只匹配 digits 即可，optionalFraction 和 optionlExponent 都匹配ε即可。所以，选项 A 与选项 B 都是正确的。

对于选项 C，在正则表达式 number -> digits optionalFraction optionlExponent 中，只匹配 digits 和 optionalFraction，digits 匹配为 2；optionalFraction ->.digits|ε匹配.digits，这个 digits 匹配为 0。所以，选项 C 正确。

对于选项 D，字符 E 后面必须要跟一个 digits 才可以，即 E 不可能为结束字符。所以，选项 D 不正确。

所以，本题的答案为 D。

12．答案：A。

分析：本题考查的是编译原理的知识。

语法分析是根据某种给定的形式文法对由单词序列（如英语单词序列）构成的输入文本进行分析，并确定其语法结构的一种过程。语法分析器（Parser）通常作为编译器或解释器的组件出现，作用是进行语法检查并构建由输入的单词组成的数据结构（一般是语法分析树、抽象语法树等层次化的数据结构）。语法分析器通常使用一个独立的词法分析器从输入字符流中分离出一个个的“单词”，并将单词流作为其输入。实际开发中，语法分析器可以手工编写，也可以使用工具（半）自动生成。

通常，语法分析器主要可以通过以下两种方式完成。

1）自顶向下分析：根据形式语法规则，在语法分析树的自顶向下展开中搜索输入符号串可能的最左推导。单词按从左到右的顺序依次使用。

2）自底向上分析：语法分析器从现有的输入符号串开始，尝试将其根据给定的形式语法规则进行改写，最终改写为语法的起始符号。

通过以上的分析可知，语法分析器可以用于识别语法错误。所以，选项 A 正确。

语义分析是编译过程的一个逻辑阶段，语义分析的任务是对结构上正确的源程序进行上下文有关性质的审查以及进行类型审查，语义分析是审查源程序有无语义错误，为代码生成阶段收集类型信息。所以，对于语义相关的处理都是由语义分析阶段实现的，而非语法分析阶段，故选项 B、选项 C、选项 D 都是错误的。

所以，本题的答案为 A。

13．答案：D。

分析：本题考查的是计算机网络与通信的知识。

IPv6（Internet Protocol Version 6）是互联网工程任务组（Internet Engineering Task Force，IETF）设计的用于替代现行版本 IP（IPv4）的下一代 IP。它由 128 位二进制数码表示，以 16 位为一组，每组以冒号“:”隔开，可以分为 8 组，每组以 4 位十六进制方式表示，一个十六进制相当于四个二进制，即十六位二进制数表示。例如，2001:0db8:85a3:08d3:1319:8a2e:0370:7344 就是一个合法的 IPv6 地址。所以，选项 D 正确。

所以，本题的答案为 D。

14．答案：C。

分析：本题考查的是计算机网络与通信的知识。

传输控制协议（Transmission Control Protocol，TCP）是一种面向连接的、可靠的、基于字节流的传输层通信协议，由 IETF 的 RFC 793 定义。它本身是可靠的，但并不等于应用程序使用 TCP 发送数据就一定是可靠的。

在阻塞模式下，send 函数的过程是将应用程序请求发送的数据复制到发送缓存中发送，并得到确认后再返回，但由于发送缓存的存在，如果发送缓存大小比请求发送的大小要大，那么 send 函数立即返回，同时向网络中发送数据；否则，send 函数向网络发送缓存中不能容纳的那部分数据，并等待接收端确认后再返回（接收端只要将数据收到接收缓存中，就会确认，并不一定要等待应用程序调用 recv 函数）。

在非阻塞模式下，send 函数的过程仅仅是将数据复制到协议栈的缓存区而已，如果缓存区可用空间不够，则尽量复制，返回成功复制的大小；如果缓存区可用空间为 0，则返回-1，同时设置 errno 的值为 EAGAIN。如果 recv 函数在等待协议接收数据时网络中断了，那么它返回 0。

默认情况下，socket 是阻塞的。阻塞与非阻塞 recv 函数的返回值没有区分，返回值小于 0 表示出错，返回值等于 0 表示连接关闭，返回值大于 0 表示接收到数据大小。

为了更好地说明该过程，下面将对 socket 中的 send 函数和 recv 函数进行详细讲解。

（1）send 函数

send 函数的原型为 int send(SOCKET s, const char FAR *buf, int len, int flags)。函数的第

一个参数指定发送端套接字描述符，第二个参数指明一个存放应用程序要发送数据的缓冲区，第三个参数指明实际要发送的数据的字节数，第四个参数一般置 0。

无论是客户端应用程序还是服务器端应用程序，它们都使用 send 函数来向 TCP 连接的另一端发送数据。区别仅在于客户端应用程序使用 send 函数向服务器发送请求，服务器端应用程序则用 send 函数来向客户程序发送应答。

以下是同步 socket 的 send 函数的执行流程。当调用该函数时，具体步骤如下：

1）send 函数先比较待发送数据的长度 len 和套接字 s 的发送缓冲的长度， 如果 len 大于 s 的发送缓冲区的长度，则该函数返回 SOCKET_ERROR。

2）如果 len 小于或者等于 s 的发送缓冲区的长度，那么 send 函数首先检查协议是否正在发送套接字 s 的发送缓冲中的数据，如果是，就等待协议把数据发送完；如果协议还没有开始发送套接字 s 的发送缓冲中的数据或者套接字 s 的发送缓冲区中没有数据，那么 send 函数就比较套接字 s 的发送缓冲区的剩余空间和 len 的大小。

3）如果发送数据的长度 len 大于剩余空间大小，那么 send 函数就一直等待协议把套接字 s 的发送缓冲中的数据发送完。

4）如果发送数据的长度 len 小于剩余空间大小，那么 send 函数就仅仅把缓冲区 buf 中的数据复制到剩余空间里（注意：并不是 send 函数把套接字 s 的发送缓冲中的数据传到连接的另一端的，而是协议传的，send 函数仅仅是把缓冲区 buf 中的数据复制到套接字 s 的发送缓冲区 buf 的剩余空间里）。

如果 send 函数复制数据成功，那么 send 函数就返回实际复制的字节数；如果 send 函数在复制数据时出现错误，那么 send 函数就返回 SOCKET_ERROR；如果 send 函数在等待协议传送数据时网络断开，那么 send 函数也返回 SOCKET_ERROR。

需要注意的是，当 send 函数把缓冲区 buf 中的数据成功复制到套接字 s 的发送缓冲区的剩余空间里后，它就返回了，但是此时这些数据并不一定马上被传到接收端。如果协议在后续的传送过程中出现网络错误，那么下一个 socket 函数就会返回 SOCKET_ERROR（每一个除 send 函数外的 socket 函数在执行的最开始总要先等待套接字的发送缓冲区的数据被协议传送完毕才能继续，如果在等待时出现网络错误，那么该 socket 函数就返回 SOCKET_ERROR）。

注意：在 UNIX 系统下，如果 send 函数在等待协议传送数据时网络断开，调用 send 函数的进程会接收到一个 SIGPIPE 信号，进程对该信号的默认处理是进程终止。

通过测试发现，异步 socket 的 send 函数在网络刚刚断开时还能发送返回相应的字节数，同时使用 select 检测也是可写的，但是过几秒钟之后，再 send 就会出错了，返回−1，select 也不能检测出可写了。

（2）recv 函数

recv 函数的原型为 int recv(SOCKET s, char FAR *buf, int len, int flags)，该函数的第一个参数指定接收端套接字描述符，第二个参数指明一个缓冲区，该缓冲区用来存放 recv 函数接收到的数据，第三个参数指明 buf 的长度，第四个参数一般置 0。

无论是客户端应用程序还是服务器端应用程序，它们都使用 recv 函数从 TCP 连接的另一端接收数据。以下只描述同步 socket 的 recv 函数的执行流程。当应用程序调用 recv 函数时，具体步骤如下：

1）recv 函数先等待 s 的发送缓冲中的数据被协议传送完毕，如果协议在传送 s 的发送缓冲中的数据时出现网络错误，那么 recv 函数返回 SOCKET_ERROR。

2）如果 s 的发送缓冲中没有数据或者数据被协议成功发送完毕后，recv 函数先检查套接字 s 的接收缓冲区；如果 s 接收缓冲区中没有数据或者协议正在接收数据，那么 recv 函数就一直等待，直到协议把数据接收完毕。当协议把数据接收完毕时，recv 函数就把 s 的接收

缓冲中的数据复制到 buf 中（注意：协议接收到的数据可能大于 buf 的长度，所以，在这种情况下，要调用几次 recv 函数才能把 s 的接收缓冲中的数据复制完。recv 函数仅仅是复制数据，真正的接收数据是协议来完成的）。

事实上，recv 函数返回其实际复制的字节数。如果 recv 函数在复制时出错，那么它返回 SOCKET_ERROR；如果 recv 函数在等待协议接收数据时网络中断了，那么它返回 0。

注意：在 UNIX 系统下，如果 recv 函数在等待协议接收数据时网络中断了，那么调用 recv 函数的进程就会接收到一个 SIGPIPE 信号，进程对该信号的默认处理是进程终止。

所以，本题的答案为 C。

15．答案：D。

分析：本题考查的是操作系统中内核对象的知识。

一个内核对象就是在系统堆中占据一块空间的结构体。不同种类的内核对象用来管理操作系统中不同的资源，如进程、线程、文件等。所有内核对象都会保存该对象的引用计数，进程对象会保存进程 ID，文件对象会保存当前字节偏移量、共享模式、打开模式等。操作系统中所有内核对象都是保存在一块内存空间中的，系统上所有的进程共享这一块内存空间。

每个进程中访问临界资源的那段程序称为临界区（临界资源是一次仅允许一个进程使用的共享资源）。每次只允许一个进程进入临界区，进入后不允许其他进程进入。

互斥对象是一种最简单的内核对象，用它可以方便地实现对某一资源的互斥访问。临界区并不是内核对象，而是系统提供的一种数据结构，程序中可以声明一个该类型变量，之后用它来实现对资源的互斥访问。当希望访问某一临界资源时，先将该临界区加锁（如果临界区不空闲，则等待），用完该资源后，将临界区释放。

所以，本题的答案为 D。

16．答案：B。

分析：本题考查的是线程的知识。

进程是资源分配的基本单位；线程是系统调度的基本单位。

开发人员平时编写的程序都是作为进程运行的，进程可以看作一系列线程和资源的统称，一个进程至少包括一个线程（主线程，进入 main 函数时产生的），在进程中可以创建其他线程，也可以不创建。

线程共享的环境包括进程代码段、进程的公有数据（利用这些共享的数据，线程很容易实现相互之间的通信）、堆中的数据、进程打开的文件描述符、信号的处理器、进程的当前目录以及进程用户 ID 与进程组 ID。

进程拥有这许多共性的同时，还拥有自己的个性。有了这些个性，线程才能实现并发性。这些个性包括线程 ID、寄存器组的值、线程的栈、错误返回码、线程的信号屏蔽码和线程的优先级。

（1）线程 ID

每个线程都有自己的线程 ID，这个 ID 在本进程中是唯一的。进程以此来标识线程。

（2）寄存器组的值

由于线程间是并发运行的，每个线程都有自己不同的运行环境，当从一个线程切换到另一个线程上时，必须将原有线程的寄存器集合的状态予以保存，以便将来该线程重启时能得以恢复。

（3）线程的栈

栈是保证线程独立运行所必需的。线程函数可以调用函数，而被调用函数中又是可以层层嵌套的，所以，线程必须拥有自己的函数栈，使得函数调用可以正常执行，不受其他线程的影响。

（4）错误返回码

由于同一个进程中有很多个线程在同时运行，可能某个线程进行系统调用后设置了 errno 值，而在该线程还没有处理这个错误时，另外一个线程就在此时被调度器调度运行，这样错误值就有可能被修改。因此，不同的线程应该拥有自己的错误返回码变量。

（5）线程的信号屏蔽码

由于每个线程所感兴趣的信号不同，因此，线程的信号屏蔽码应该由线程自己管理。但所有线程都共享同样的信号处理器。

（6）线程的优先级

由于线程需要像进程那样能够被调度，因此就必须要有可供调度使用的参数，这个参数就是线程的优先级。

通过以上分析可知选项 A、选项 C 和选项 D 是错误的。对于数据区而言，线程通常都可以通过公共的数据区进行通信，因此，选项 B 是正确的。

所以，本题的答案为 B。

17．答案：A。

分析：本题考查的是操作系统的知识。

在本题中，首先需要弄清楚一个概念，即什么叫作“页面置换”。在地址映射过程中，如果在页面中发现所要访问的页面不在内存中，则产生缺页中断。当发生缺页中断时，操作系统必须在内存中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称为页面置换算法，也称为页面淘汰算法。

先进先出页面淘汰（First In First Out，FIFO）算法在实现时，置换出最早进入内存的页面，即在内存中驻留时间最久的页面。该算法实现简单，只需把调入内存的页面根据先后次序链接成队列，设置一个指针总指向最早进入内存的页面。

本题中，置换过程如下：

1）访问 1，缺页，调入 1，内存中为 1。

2）访问 2，缺页，调入 2，内存中为 1，2。

3）访问 3，缺页，调入 3，内存中为 1，2，3。

4）访问 4，缺页，调入 4，淘汰 1，内存中为 2，3，4。

5）访问 1，缺页，调入 1，淘汰 2，内存中为 3，4，1。

6）访问 2，缺页，调入 2，淘汰 3，内存中为 4，1，2。

7）访问 5，缺页，调入 5，淘汰 4，内存中为 1，2，5。

8）访问 1，不缺页，内存中为 1，2，5。

9）访问 2，不缺页，内存中为 1，2，5。

10）访问 3，缺页，调入 3，淘汰 1，内存中为 2，5，3。

11）访问 4，缺页，调入 4，淘汰 2，内存中为 5，3，4。

12）访问 5，不缺页，内存中为 5，3，4。

13）访问 6，缺页，调入 6，淘汰 5，内存中为 3，4，6。

所以，一共产生了 10 次缺页，因此，选项 A 正确。

所以，本题的答案为 A。

18．答案：A。

分析：本题考查的是计算机组成原理的知识。

中断是指计算机在执行期间，系统内发生任何非寻常的或非预期的急需处理事件，使得 CPU 暂时中断当前正在执行的程序而转去执行相应的事件处理程序，待处理完毕后又返回原来被中断处继续执行或调度新的进程执行。引起中断发生的事件被称为“中断源”。中断源向 CPU 发出的请求中断处理信号称为“中断请求”，而 CPU 收到中断请求后转到相应的

事件处理程序称为“中断响应”。中断是异步过程调用，简而言之，就是打断当前 CPU 正在执行的任务转而执行另一个任务。

中断必须满足以下四个基本条件：

1）一条指令执行结束。

2）CPU 处于开中断状态。

3）当前没有发生复位，保持和非屏蔽中断请求。

4）如果当前执行的指令是开中断指令和中断返回指令，则它们执行完后再执行一条指令，CPU 才能响应中断请求。

本题中，键盘每按一次键，或鼠标单击一次，都会产生一个中断，称为按键中断，执行中断响应程序，操作系统将按键消息加入消息队列，所以，选项 A 正确，而选项 B、选项 C 和选项 D 都不正确。

所以，本题的答案为 A。

19．答案：B。

分析：本题考查的是操作系统基础知识。

在计算机中，由于程序是顺序执行而不是并发执行，因此本题中，程序 A 不能在程序 B 使用设备时去使用 CPU，也就是说，只有等到程序 A 执行完毕了，程序 B 才会被开始执行。

单独来看，程序 A 单独执行需要的总时间为 10s+5s+5s+10s+10s=40s，程序 B 单独执行需要的总时间为 10s+10s+5s+5s+10s=40s，所以，二者单独运行需要的总时间为 80s。

对于程序 A 而言，CPU 时间为 10 + 5 + 10 = 25s；对于程序 B 而言，CPU 时间为 10 + 5 = 15s，CPU 时间综合为 25s+15s=40s，由此可知，CPU 的利用率为 40s/80s=50%，所以，选项 B 正确。

所以，本题的答案为 B。

20．答案：C。

分析：本题考查的是 fork 函数的使用。

要弄明白本题的输出结果，就必须清楚 fork 函数的运行机理。

fork 函数是 UNIX 操作系统下以自身进程创建子进程的系统调用，通过系统调用创建一个与原来进程几乎完全相同的进程，一个是子进程，一个是父进程，该子进程拥有与父进程相同的堆栈空间，也就是说，两个进程可以做完全相同的事，可以理解为它们俩是双胞胎兄弟，但如果初始参数或者传入的变量不同，两个进程也可以做不同的事。在 fork 函数的调用处，整个父进程空间会按原样复制到子进程中，包括指令、变量值、程序调用栈、环境变量、缓冲区等。

fork 函数的一个奇妙之处就是它仅仅被调用一次，却能够返回两次，它可能有三种不同的返回值：

1）在父进程中，fork 函数返回新创建子进程的进程 ID。

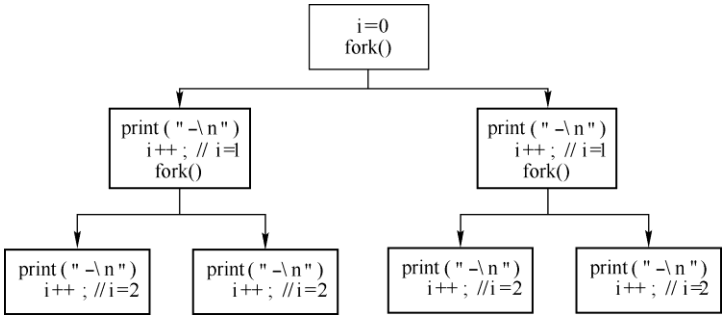
2）在子进程中，fork 函数返回 0。

3）如果出现错误，fork 函数返回一个负值。

所以，可以通过 fork 函数的返回值来判断当前进程是子进程还是父进程。

当 printf 函数遇到了换行符"\n"或是 EOF、缓冲区满、文件描述符关闭、主动 flush、程序退出等情况时，它会刷出缓冲区。对于本题而言，printf("-\n")中有换行，因此会马上输出而不会缓存，所以，此时会打印 6 个“-”，即选项 C 正确。

执行过程如下图所示：



如果将上述代码中的 printf("-\n")语句改为 printf("-")语句，结果就大相径庭了。由于printf("-")语句有缓冲区，因此，printf("-")把字符“-”放到缓存中，并没有真正地输出，在执行 fork 函数时，缓存被复制到子进程空间，所以，输出“-”的个数就变为 8 个，比 6 个多 2 个。

所以，本题的答案为 C。

如果将 printf()和 fork()这两句顺序调换，那结果会怎样？

对于 printf("-")的情况，由于“-”在缓冲区中没有实际输出，因此，printf 函数和 fork 函数的顺序调换没有影响，都是 8 个。

对于 printf("-\n")的情况，因为有实际输出调换顺序 printf()在前，所以，fork 函数在后输出为 3 个“-”。

21．答案：A。

分析：本题考查的是操作系统基础知识。

对于选项 A，可抢占式调度会导致系统的开销更大。可抢占式（Preemptive）调度保证在任何时刻具有最高优先级的进程占有处理机运行，因此，该方式增加了处理机调度的时间，同时需要为退出的进程保留现场，为获取到处理机的进程恢复现场等时间（和空间），因此，开销比较大。非抢占式（Nonpreemptive）调度是一种让进程运行直到结束或阻塞的调度方式（容易实现，适合专用系统，不适合通用系统），所以，选项 A 不正确。

对于选项 B，在内核中，对于每个进程都有一个文件描述符表，表示这个进程打开的所有文件。文件描述符表中的每一项都是一个指针，指向一个用于描述所打开文件的数据块——file 对象。file 对象中描述了文件的打开模式、读写位置等重要信息，当进程打开一个文件时，内核就会创建一个新的 file 对象。需要注意的是，file 对象不是专属于某个进程的，不同进程的文件描述符表中的指针可以指向相同的 file 对象，从而共享这个打开的文件。file 对象有引用计数，记录了引用这个对象的文件描述符个数，只有当引用计数为 0 时，内核才销毁 file 对象，因此，某个进程关闭文件，不会影响与之共享同一个 file 对象的进程，所以，选项 B 正确。

对于选项 C，只读存储器（Read Only Memory，ROM）和随机存取存储器（Random Access Memory，RAM）指的都是半导体存储器，ROM 在系统停止供电时仍然可以保持数据，而 RAM 通常都是在掉电之后就丢失数据，典型的 RAM 就是计算机的内存。磁盘是一种类似磁带的计算机的外部存储器，它将圆形的磁性盘片装在一个方的密封盒子里。固态硬盘（Solid State Drives，SSD）是用固态电子存储芯片阵列而制成的硬盘，由控制单元和存储单元（FLASH 芯片、DRAM 芯片）组成。ROM、RAM、磁盘、SSD 都是存储设备，其中，访问速度最快的是 RAM，访问速度最慢的是磁盘，CPU 的高速缓存一般是由 RAM 组成的，所以，选项 C 正确。

对于选项 D，如果系统中存在多个进程，它们中的每一个进程都占用了某种资源而又都在等待其中另一个进程所占用的资源，那么这种等待永远都不能结束，就称系统出现了“死锁”，所以，选项 D 正确。

所以，本题的答案为 A。