目　　录

1 计算机基础知识

1.1 计算机概述

一、发展史

1、计算机发展代别划分：

代别	年代	逻辑（电子）元件
第一代	1946-1958	电子管
第二代	1959-1964	晶体管
第三代	1965-1970	集成电路
第四代	1971-	大规模、超大规模集成电路

2、1946 年 2 月，在美国宾夕法尼亚大学诞生了世界上第一台电子计算机 ENIAC（Electronic Numerical Integrator And Computer），这台计算机占地 170 平方米，重 30 吨，用了 18000 多个电子管，每秒能进行 5000 次加法运算。

3、冯·诺依曼理论

1944 年，美籍匈牙利数学家冯·诺依曼提出计算机基本结构和工作方式的设想，为计算机的诞生和发展提供了理论基础。时至今日，尽管计算机软硬件技术飞速发展，但计算机本身的体系结构并没有明显的突破，当今的计算机仍属于冯·诺依曼架构。

其理论要点如下：

计算机硬件设备由存储器、运算器、控制器、输入设备和输出设备 5 部分组成。

存储程序思想——把计算过程描述为由许多命令按一定顺序组成的程序，然后把程序和数据一起输入计算机，计算机对已存入的程序和数据处理后，输出结果。

二、计算机的分类

根据计算机的性能指标，如机器规模的大小、运算速度的高低、主存储容量的大小、指令系统性能的强弱以及机器的价格等，可将计算机分为巨型机、大型机、中型机、小型机、微型机和工作站。

1.巨型机：具有很强的计算和处理数据的能力，主要特点表现为高速度和大容量，配有多种外部和外围设备及丰富的、高功能的软件系统。主要用来承担重大的科学研究、国防尖端技术和国民经济领域的大型计算课题及数据处理任务。如大范围天气预报，整理卫星照片，原子核物的探索，研究洲际导弹、宇宙飞船等。“天河一号”为我国首台千万亿次超级计算机。

2. 大、中型机：大型机使用专用的处理器指令集、操作系统和应用软件，大量使用冗余等技术确保其安全性及稳定性，擅长非数值计算(数据处理)，主要用于商业领域，如银行和电信。

3.小型机是指采用精简指令集处理器，性能和价格介于 PC 服务器和大型主机之间的一种高性能 64 位计算机。

小型机与普通服务器相比具有：

①高可靠性（Reliability）：计算机能够持续运转，从来不停机。

②高可用性（Availability）：重要资源都有备份；能够检测到潜在要发生的问题，并且能够转移其上正在运行的任务到其它资源，以减少停机时间，保持生产的持续运转；具有实时在线维护和延迟性维护功能。

③高服务性（Serviceability）：能够实时在线诊断，精确定位出根本问题所在，做到准确无误的快速修复。

4. 微型机：通常作为个人计算机，由硬件系统和软件系统组成，是一种能独立运行，完成特定功能的设备。个人计算机不需要共享其他计算机的处理、磁盘和打印机等资源也可以独立工作。从台式机、笔记本电脑到上网本和平板电脑以及超级本等都属于个人计算机的范畴。

5.工作站是一种高端的通用微型计算机。它是为了单用户使用并提供比个人计算机更强大的性能，尤其是在图形处理能力，任务并行方面的能力。通常配有高分辨率的大屏、多屏显示器及容量很大的内存储器和外部存储器，并且具有极强的信息和高性能的图形、图像处理功能的计算机。另外，连接到服务器的终端机也可称为工作站。

三、计算机的应用

计算机的快速性、通用性、准确性和逻辑性等特点，使它不仅具有高速运算能力，而且还具有逻辑分析和逻辑判断能力。如今计算机已渗透到人们生活和工作的各个层面中，主要体现在以下几个方面的运用。

1. 科学计算

科学计算(或数值计算)是指利用计算机来完成科学研究和工程技术中提出的数学问题的计算。在现代科学技术工作中，科学计算问题是大量的和复杂的。利用计算机的高速计算、大存储容量和连续运算的能力，可以实现人工无法解决的各种科学计算问题。

2. 信息处理

信息处理(数据处理)是指对各种数据进行收集、存储、整理、分类、统计、加工、利用、传播等一系列活动的统称。80%以上的计算机主要用于数据处理，决定了计算机应用的主导方向。

3. 自动控制

自动控制(过程控制)是利用计算机及时采集检测数据，按最优值迅速地对控制对象进行自动调节或自动控制。采用计算机进行自动控制，不仅可以大大提高控制的自动化水平，而且可以提高控制的及时性和准确性，提高产品质量及合格率。目前，计算机过程控制已在机械、冶金、石油、化工、纺织、水电、航天等部门得到广泛的应用。

4. 计算机辅助技术

计算机辅助技术是指利用计算机帮助人们进行各种设计、处理等过程，它包括计算机辅助设计(CAD)、计算机辅助制造(CAM)、计算机辅助教学(CAI)和计算机辅助测试(CAT)等。另外，计算机辅助技术还有辅助生产、辅助绘图和辅助排版等。

5. 人工智能

人工智能(Artificial Intelligence，简称 AI)又可称为智能模拟，是计算机模拟人类的智能活动，诸如感知、判断、理解、学习、问题求解和图像识别等。人工智能的研究目标是计算机更好地模拟人的思维活动，那时的计算机将可以完成更复杂的控制任务。

6. 网络应用

随着社会信息化的发展，通信业也发展迅速，计算机在通信领域的作用越来越大，特别是促进了计算机网络的迅速发展。目前，全球最大的网络(Internet，即国际互联网)已把全球的大多数计算机联系在一起。计算机在信息高速公路、电子商务、娱乐和游戏等领域也得到了快速的发展。

习题:

1. 美籍匈牙利数学家冯·诺依曼对计算机科学发展所做出的贡献是（　　）。

A. 提出理想计算机的数学模型，成为计算机科学的理论基础。

B. 是世界上第一个编写计算机程序的人。

C. 提出存储程序工作原理，并设计出第一台具有存储程序功能的计算机 EDVAC。

D. 采用集成电路作为计算机的主要功能部件。

2. 在下面各世界顶级的奖项中，为计算机科学与技术领域做出杰出贡献的科学家设立的奖项是（　　）。

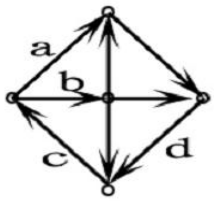
A. 沃尔夫奖　　B. 诺贝尔奖　　C. 菲尔兹奖　　D. 图灵奖

3. IT 的含义是（　　）。

A. 通信技术　　B. 信息技术　　C. 网络技术　　D. 信息学

4. 在下列关于图灵奖的说法中，不正确的是（　　）。

A. 图灵奖是美国计算机协会于 1966 年设立的，专门奖励那些对计算机事业作出重要贡献的个人



B. 图灵奖有“计算机界诺贝尔奖”之称

C. 迄今为止，还没有华裔计算机科学家获此殊荣

D. 图灵奖的名称取自计算机科学的先驱、英国科学家阿兰·图灵

5. 关于图灵机下面的说法哪个是正确的：（　　）

A. 图灵机是世界上最早的电子计算机。

B. 由于大量使用磁带操作，图灵机运行速度很慢。

C. 图灵机是英国人图灵发明的，在二战中为破译德军的密码发挥了重要作用。

D. 图灵机只是一个理论上的计算模型。

6. 提出“存储程序”的计算机工作原理的是（　　）。

A. 克劳德·香农　　B. 戈登·摩尔　　C. 查尔斯·巴比奇　　D. 冯·诺依曼

7. 摩尔定律（Moore's law)是由英特尔创始人之一戈登·摩尔(Gordon Moor)提出来的。根据摩尔定律，在过去几十年一级在可预测的未来纪念，单块集成电驴的集成度大约每（　　）个月翻一番。

A. 1　　B. 6　　C. 18　　D. 36

8. 1956 年（　　）授予肖克利、巴丁和布拉顿，以表彰他们对半导体的研究和晶体管效应的发现。

A. 诺贝尔物理学奖　　B. 约翰·冯·诺依曼奖　　C. 图灵奖　　D. 高德纳奖

9. 从 ENIAC 到当前最先进的计算机，冯·诺依曼体系结构始终占有重要地位。冯诺依曼提醒结

构的核心内容是（　　）。

A．采用开关电路　　B．采用半导体器件

C．采用存储程序和程序控制原理　　D．采用键盘输入

10．　1946 年诞生于美国宾夕法尼亚大学的 ENIAC 属于（　　）计算机。

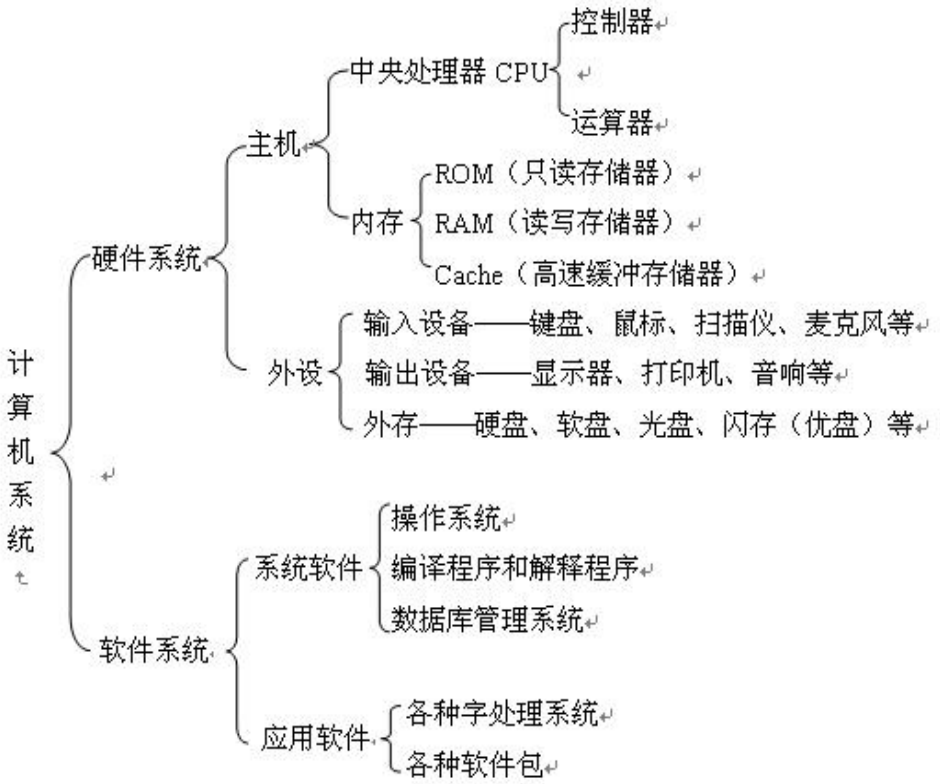
A．电子管　　B．晶体管　　C．集成电路　　D．超大规模集成电路

11．计算机界的最高奖是(　　)。

A.菲尔兹奖　　B.诺贝尔奖　　C.图灵奖　　D.普利策奖

1.2 计算机硬件

1.2.1 计算机系统的基本结构



计算机系统由硬件和软件两部分组成。硬件系统是计算机的“躯干”，是物质基础。而软件系统则是建立在这个“躯干”上的“灵魂”。

计算机硬件由五大部分组成：运算器、控制器、存储器、输入设备、输出设备。

中央处理器（CPU——Central Processing Unit）

由运算器、控制器和一些寄存器组成；

运算器进行各种算术运算和逻辑运算；

控制器是计算机的指挥系统；

CPU 的主要性能指标是主频和字长。

存储器

存储器的主要功能是用来保存各类程序的数据信息。

存储器可分为主存储器和辅助存储器两类。

①主存储器（也称为内存储器），属于主机的一部分。用于存放系统当前正在执行的数据和程序，属于临时存储器。

②辅助存储器（也称外存储器），它属于外部设备。用于存放暂不用的数据和程序，属于永久存储器。存储器与 CPU 的关系表示:

（1）内存储器

内存又称为主存，它和 CPU 一起构成了计算机的主机部分，它存储的信息可以被 CPU 直接访问。内存由半导体存储器组成，存取速度较快，但一般容量较小。内存中含有很多的存储单元，每个单元可以存放 1 个 8 位的二进制数，即 1 个字节（Byte，简称“B”）。内存中的每个字节各有一个固定的编号，这个编号称为地址。CPU 在存取存储器中的数据时是按地址进行的。所谓存储器容量即指存储器中所包含的字节数，通常用 KB、MB、GB、 TB 和 PB 作为存储器容量单位。它们之间的关系为：

1KB=1024B　1MB=1024KB　1GB=1024MB　1TB=1024GB　1PB=1024TB

内存储器通常可以分为随机存储器 RAM、只读存储器 ROM 和高速缓冲存储器 Cache 三种。

① RAM 是一种读写存储器，其内容可以随时根据需要读出，也可以随时重新写入新的信息。当电源电压去掉时，RAM 中保存的信息都将全部丢失。 random access memory，RAM

② ROM 是一种内容只能读出而不能写入和修改的存储器，其存储的信息是在制作该存储器时就被写入的。在计算机运行过程中，ROM 中的信息只能被读出，而不能写入新的内容。计算机断电后，ROM 中的信息不会丢失。它主要用于检查计算机系统的配置情况并提供最基本的输入/输出（I/O）控制程序。Read-Only Memory，ROM

③由于计算机的 CPU 速度的不断提高，RAM 的速度很难满足高速 CPU 的要求，所以在读/写系统内存都要加入等待的时间，这对高速 CPU 来说是一种极大的浪费。Cache 是指在 CPU 与内存之间设置的一级或两级高速小容量存储器，称之为高速缓冲存储器，固化在主板上。在计算机工作时，系统先将数据由外存读入 RAM 中，再由 RAM 读入 Cache 中，然后 CPU 直接从 Cache 中取数据进行操作。

（2）外存储器

外存储器又称为辅助存储器，它的容量一般都比较大，而且大部分可以移动，便于在不同计算机之间进行信息交流。在微型计算机中，常用的外存有软盘、硬盘、闪存和光盘 4 种。

①软盘存储器

软盘存储器由软盘、软盘驱动器和软盘适配器三部分组成。软盘是活动的存储介质，软盘驱动器是读写装置，软盘适配器是软盘驱动器与主机连接的接口。软盘驱动器安装在主机箱内，软盘驱动器插槽暴露在主机箱的前面板上，可方便地插入或取出软盘。

②硬盘存储器

硬盘存储器是由电机和硬盘组成的，一般置于主机箱内。硬盘是涂有磁性材料的磁盘组件，用于存放数据。硬盘的机械转轴上串有若干个盘片，每个盘片的上下两面各有一个读/写磁头，与软盘磁头不同，硬盘的磁头不与磁盘表面接触，它们“飞”在离盘片面百万分之一英寸

的气垫上。硬盘是一个非常精密的机械装置，磁道间只有百万分之几英寸的间隙，磁头传动装置必须把磁头快速而准确地移到指定的磁道上。

③闪存

闪存又名优盘，是在存储速度与容量上介于软盘与硬盘之间的一种外部存储器。

④光盘

光盘的存储介质不同于磁盘，它属于另一类存储器。由于光盘的容量大、存取速度较快、不易受干扰等特点，其应用越来越广泛。光盘根据其制造材料和记录信息方式的不同一般分为三类：只读光盘、一次写入型光盘和可擦写光盘。

（3）输入设备

输入设备是外界向计算机传送信息的装置。在微型计算机系统中，最常用的输入设备是键盘和鼠标，此外还有光电笔、数字化仪、图像扫描仪、触摸屏、麦克风、视频输入设备、条形码扫描器等。也可以用磁盘和磁带进行输入。

（4）输出设备

输出设备的作用是将计算机中的数据信息传送到外部媒介，并转化成某种为人们所认识的表示形式。在微型计算机中，最常用的输出设备有显示器和打印机。此外，还有绘图仪等，也可以通过磁盘和磁带输出。

二、总线结构

按照总线上传输信息的不同，总线可以分为数据总线（DB），地址总线（AB）和控制总线（CB）三种。

①数据总线：用来传送数据信息，它主要连接了 CPU 与各个部件，是它们之间交换信息的通路。数据总线是双向的，而具体的传送方向由 CPU 控制。

②地址总线：用来传送地址信息。CPU 通过地址总线中传送的地址信息访问存储器。通常地址总线是单向的。同时，地址总线的宽度决定可以访问的存储器容量大小，如 20 条地址总线可以控制 1MB 的存储空间。

③控制总线：用来传送控制信号，以协调各部件之间的操作。控制信号包括 CPU 对内存储器和接口电路的读写控制信号、中断响应信号，也包括其他部件传送给 CPU 的信号，如中断申请信号、准备就绪信号等。

三、主要的性能指标

计算机的常用的指标有：

1. 字长

字长是指一台计算机所能处理的二进制代码的位数。计算机的字长直接影响到它的精度、功能和速度。字长愈长，能表示的数值范围就越大，计算出的结果的有效位数也就越多；字长愈长，能表示的信息就越多，机器的功能就更强。目前常用的是 16 位、32 位、64 位字长。

2. 运算速度

运算速度是指计算机每秒钟所能执行的指令条数，一般用 MIPS（Million of Instructions Per Second，即每秒百万条指令）为单位。由于不同类型的指令执行时间长短不同，因而运算速度的计算方法也不同。

3. 主频

主频是指计算机 CPU 的时钟频率，它在很大程度上决定了计算机的运算速度。一般时钟频率越高，运算速度就越快。主频的单位一般是 MHz（兆赫）或 GHz（吉赫），如微处理器 Pentium4/2.0GHz 的主频为 2×1000MHz。

4. 内存容量

内存容量是指内存储器中能够存储信息的总字节数，一般以 GB 为单位。内存容量反映内存储器存储数据的能力。目前计算机的内存容量有 2GB、4GB、8GB 等。

1.2.2 中央处理器 CPU

CPU(中央处理单元)是微机的核心部件，是决定微机性能的关键部件。20 世纪 70 年代微型机的 CPU 问世，微型计算机的核心部件微处理器从 Intel 4004，80286，80386，80486 发展到 Pentium II/Ill 和 Pentium 4，数位从 4 位、8 位、16 位、32 位发展到 64 位，主频从几 MHZ 到今天的数 GHZ 以上(1GHZ＝1000MHZ),CPU 芯片里集成的晶体管数由 2 万个跃升到 1000 万个以上。CPU 的发展和技术的进展直接推动了微型计算机的发展，也是微机各个发展阶段的主要标志。

从原理上看，CPU 的内部结构分控制单元、逻辑单元、存储单元三部分。从组成器件上看，CPU 的内部是由成千上万个晶体管组成，晶体管实质上就是一双位开关：即“开”和“关”。

CPU 的主要性能指标包括时主频、字长、高速缓存容量、指令集合和动态处理技术、制造工艺、封装方式和工作电压等。

主频是指 CPU 的工作时钟频率，是 CPU 内核电路的实际运行频率。一般说主频越高，一个时钟周期里面完成的指令数也越多，速度也越快。主频的单位为兆赫兹(MHz)和吉赫兹(GHz)。我们通常所说的 2.8GHz，3.0GHz 就是指 CPU 的主频。

字长(word size)指的是微处理器 CPU 能够同时处理的二进制位数的个数。字长的大小取决于 ALU 中寄存器的容量和连接着这些寄存器的电路性能。例如，8 位字长的微处理器有 8 位的寄存器，每次能处理 8 位的数据，因此被称为“8 位处理器”。有更大字长的处理器能够在每个处理器周期内处理更大的数据，因此字长越长计算机性能越好。目前的个人计算机通常都带有 32 位或 64 位的处理器。

高速缓存(cache)也称为“RAM 缓存”或“缓冲存储器”。它是一种具有很高速度的特殊内部存储器，与安装在主板上其他位置的内存相比，它能够使微处理器更快的获得数据。

字节和字长的区别：常用的英文字符用 8 位二进制就可以表示，所以通常就将 8 位称为一个字节，字节是一种存储容量单位。而字长是 CPU 处理能力的一种标准，字长的长度是不固定的，对于不同的 CPU、字长的长度也不一样。8 位的 CPU 一次只能处理一个字节，而 32 位的 CPU 一次就能处理 4 个字节，同理字长为 64 位的 CPU 一次可以处理 8 个字节。

1971 年，英特尔公司推出了世界上第一款微处理器 4004，字长 4 位，四位微处理器。
1978 年，英特尔公司生产的 8086 是第一个 16 位的微处理器。
1985 年，英特尔生产出 32 位字长处理器 80386。
目前市场上主流的 CPU 的字长几乎都达到了 64 位。

1.2.3 作业

1. 下列哪个不是 CPU（中央处理单元）（　　）。
   A. Intel Itanium　　B. DDR SDRAM　　C. AMD Athlon64　　D. AMD Opteron
2. 下面哪个部件对于个人桌面电脑的正常运行不是必需的（　　）。
   A. CPU　　B. 图形卡（显卡）　　C. 光驱　　D. 主板
3. 下列哪个不是计算机的存储设备（　　）。
   A. 文件管理器　　B. 内存　　C. 高速缓存　　D. 硬盘
4. 下列说法中错误的是（　　）。
   A. CPU 的基本功能就是执行指令。
   B. CPU 访问内存的速度快于访问高速缓存的速度。
   C. CPU 的主频是指 CPU 在 1 秒内完成的指令周期数。
   D. 在一台计算机内部，一个内存地址编码对应唯一的一个内存单元。
5. 彩色显示器所显示的五彩斑斓的色彩，是由红色、蓝色和（　　）。 色混合而成的。
   A. 紫　　B. 白　　C. 黑　　D. 绿
6. 用静电吸附墨粉后转移到纸张上，是哪种输出设备的工作方式（　　）
   A. 针式打印机　B. 喷墨打印机　　C. 激光打印机　　D. 笔式绘图仪
7. 下列哪个不是数据库软件的名称（　　）
   A. MySQL　　B. SQL Server　　C. Oracle　　D. 金山影霸
8. 下列哪个程序设计语言不支持面向对象程序设计方法（　　）
   A. C++　　B. Object Pascal　　C. C　　D. Java
9. 处理器 A 每秒处理的指令数是处理器 B 的 2 倍。某一特定程序 P 分别编译为处理器 A 和处理器 B 的指令，编译结果处理器 A 的指令数是处理器 B 的 4 倍。已知程序 P 在处理器 A 上执行需要 1 个小时，那么在输入相同的情况下，程序 P 在处理器 B 上执行需要（　　）小时。
   A. 4　　B. 2　　C. 1　　D. 1 / 2
10. 以下哪个不是计算机的输出设备（　　）
   A. 音箱　　B. 显示器　　C. 打印机　　D. 扫描仪
11. 以下断电之后仍能保存数据的是（　　）。
   A. 硬盘　　B. 寄存器　　C. 显存　　D. 内存
12. 以下断电之后仍能保存数据的有（　　）
   A. 寄存器　　B. ROM　　C. RAM　　D. 高速缓存
13. CPU 是（　　）的简称。
   A. 硬盘　　B. 中央处理器　　C. 高级程序语言　　D. 核心寄存器
14. 在以下各项中，（　　）不是 CPU 的组成部分。
   A. 控制器　　B. 运算器　　C. 寄存器　　D. 主板

15. 一个完整的计算机系统应包括（　　）

A．系统硬件和系统软件　　B．硬件系统和软件系统

C．主机和外部设备　　D．主机、键盘、显示器和辅助存储器

16. 微型计算机中，控制器的基本功能是（　　）。

A. 控制机器各个部件协调工作　　B. 实现算术运算和逻辑运算

C. 获取外部信息　　D. 存放程序和数据

17. 计算机在工作过程中，若突然停电，（　　）中的信息不会丢失。

A. ROM 和 RAM　　B. CPU　　C.ROM　　D. RAM

18. 关于计算机内存下面的说法哪个是正确的：（　　）

A. 随机存储器（RAM）的意思是当程序运行时，每次具体分配给程序的内存位置是随机而不确定的。

B. 1MB 内存通常是指 1024*1024 字节大小的内存。

C. 计算机内存严格说来包括主存（memory）、高速缓存（cache）和寄存器（register）三个部分。

D. 一般内存中的数据即使在断电的情况下也能保留 2 个小时以上。

19. 关于 BIOS 下面说法哪个是正确的：（　　）

A. BIOS 是计算机基本输入输出系统软件的简称。

B. BIOS 里包含了键盘、鼠标、声卡、显卡、打印机等常用输入输出设备的驱动程序。

C. BIOS 一般由操作系统厂商来开发完成。

D. BIOS 能提供各种文件拷贝、复制、删除以及目录维护等文件管理功能。

20. 关于 CPU 下面哪个说法是正确的：（　　）

A. CPU 全称为中央处理器（或中央处理单元）。

B. CPU 可以直接运行汇编语言。

C. 同样主频下，32 位的 CPU 比 16 位的 CPU 运行速度快一倍。

D. CPU 最早是由 Intel 公司发明的。

21. 主存储器的存取速度比中央处理器（CPU）的工作速度慢得多，从而使得后者的效率受到影响。而根据局部性原理，CPU 所访问的存储单元通常都趋于聚集在一个较小的连续区域中。于是，为了提高系统整体的执行效率，在 CPU 中引入了（　　）。

A. 寄存器　　B. 高速缓存　　C. 闪存　　D. 外存

22. 寄存器是（　　）的重要组成部分。

A．硬盘　　B．高速缓存　　C．内存　　D．中央处理器（CPU）

23. 计算机如果缺少（　　），将无法正常启动。

A．内存　　B．鼠标　　C． U 盘　　D． 摄像头

24. 目前计算机芯片（集成电路）制造的主要原料是（　　），它是一种可以在沙子中提炼出的物质。

A．硅　　B．铜　　C．锗　　D．铝

25. 目前个人电脑的（　　）市场占有率最靠前的厂商包括 Intel、AMD 等公司。

A．显示器　　B．CPU　　C．内存　　D．鼠标

26. 地址总线的位数决定了 CPU 可直接寻址的内存空间大小，例如地址总线为 16 位，其最大的可寻址空间为 64KB。如果地址总线是 32 位，则理论上最大可寻址的内存空间为（　　）。

A．128KB　　B．1MB　　C．1GB　　D．4GB

27. 以下哪一种设备属于输出设备(　　)。

A.扫描仪　　B.键盘　　C.鼠标　　D.打印机

28. 下列对操作系统功能的描述最为完整的是(　　)。

A.负责外设与主机之间的信息交换

B.负责诊断机器的故障

C.控制和管理计算机系统的各种硬件和软件资源的使用

D.将没有程序编译成目标程序

29. CPU、存储器、I/O 设备是通过(　　)连接起来的。

A. 接口　　B.总线　　C.控制线　　D.系统文件

30. 断电后会丢失数据的存储器是(　　)。

A.RAM　　B.ROM　　C.硬盘　　D.光盘

31. 下列说法正确的是（　　）。

A. CPU 的主要任务是执行数据运算和程序控制

B. 存储器具有记忆能力，其中信息任何时候都不会丢失

C. 两个显示器屏幕尺寸相同，则它们的分辨率必定相同

D. 个人用户只能使用 Wifi 的方式连接到 Internet

32. 所谓的“中断”是指（　　）。

A. 操作系统随意停止一个程序的运行

B. 当出现需要时，CPU 暂时停止当前程序的执行转而执行处理新情况的过程

C. 因停机而停止一个程序的运行

D. 电脑死机

33. 以下不是 CPU 生产厂商的是（　　）。

A. Intel　　B. AMD　　C. Microsoft　　D. IBM

34. 以下不是存储设备的是（　　）。

A. 光盘　　B. 磁盘　　C. 固态硬盘　　D. 鼠标

35. 位机器和 64 位机器的区别的是（　　）。

A. 显示器不同　　B. 硬盘大小不同

C. 寻址空间不同　　D. 输入法不同

36. 1TB 代表的字节数是(　　)。

A.2 的 10 次方　　B.2 的 20 次方　　C.2 的 30 次方　　D.2 的 40 次方

37. 1MB 等于（　　）。

A. 1000 字节　　B. 1024 字节　　C.1000 X 1000 字节 D. 1024 X 1024 字节

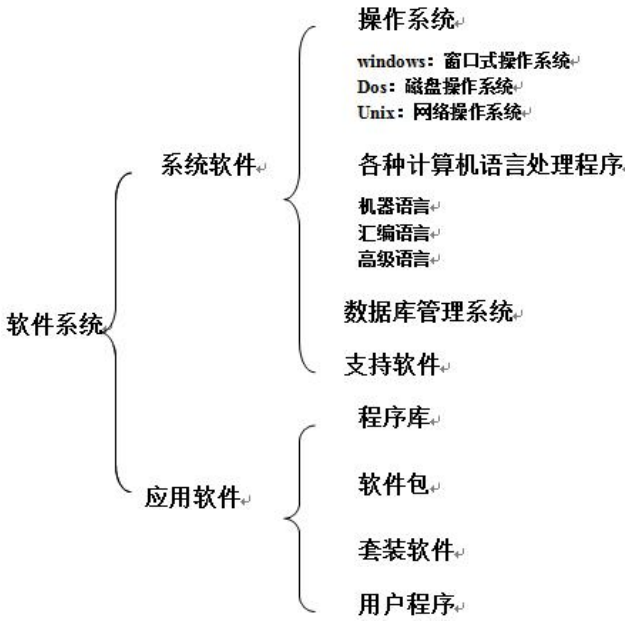
38. 以下断电后仍能保存数据的有（　　）。

A. 硬盘　　B. 高速缓存　　C. 显存　　D. RAM

1.3 计算机软件

1.3.1 软件系统

软件是计算机的灵魂。没有安装软件的计算机称为“裸机”，无法完成任何工作。硬件为软件提供运行平台。软件和硬件相互关联，两者之间可以相互转化，互为补充。计算机的软件分成系统软件和应用软件两大类。



一、系统软件

系统软件是指控制和协调计算机及外部设备,支持应用软件开发和运行的系统，是无需用户干预的各种程序的集合，主要功能是调度，监控和维护计算机系统；负责管理计算机系统中各种独立的硬件，使得它们可以协调工作。系统软件使得计算机使用者和其他软件将计算机当作一个整体而不需要顾及到底层每个硬件是如何工作的。

常用的操作系统：

1、桌面操作系统从软件上可主要分为两大类，分别为类 Unix 操作系统和 Windows 操作系统：

Unix 和类 Unix 操作系统：Mac OS X，Linux 发行版（如 Debian，Ubuntu，Linux Mint，openSUSE，Fedora，Mandrake，Red Hat，Centos 等）；

微软公司 Windows 操作系统 ：Windows 98，Windows 2000，Windows XP，Windows Vista，Windows 7，Windows 8，Windows 8.1，Windows10 等

2、服务器操作系统

服务器操作系统主要集中在三大类：

Unix 系列：SUNSolaris，IBM-AIX，HP-UX，FreeBSD，OS X Server 等；

Linux 系列：Red Hat Linux，CentOS，Debian，UbuntuServer 等；

Windows 系列：Windows NT Server，Windows Server 2003，Windows Server 2008，Windows Server 2008 R2，windows server 2012，windows server technical 等。

二、应用软件

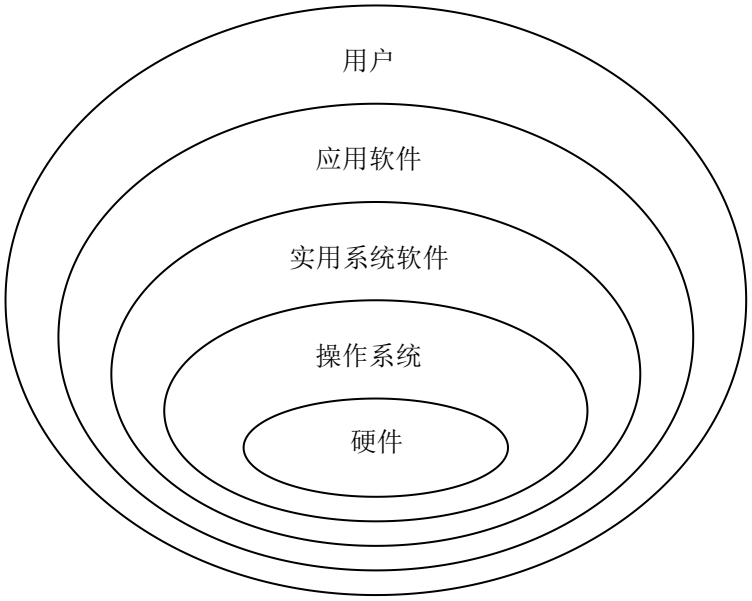
应用软件是用户为了解决各自的应用领域里的具体任务而编写的各种应用程序和有关文档资料的统称。这类软件能解决特定问题。应用软件与系统软件的关系是：系统软件为应用软件提供基础和平台，没有系统软件应软件是无源之本，反过来应用软件又为系统服务。

常用的应用软件有以下几类：

（1）字处理软件　　（2）电子制表软件

（3）计算机辅助设计软件　　（4）图形软件

（5）教育软件　　（6）电子游戏软件



三、计算机的指令

指令是一组二进制代码，它规定了由计算机执行的程序的一步操作。一条指令由操作码和操作数组成，前者规定指令要完成的操作，必不可少；后者是这个操作针对的对象，可以没有。

指令系统是一种计算机所能识别并可执行的全部指令的集合。例如，80386 的指令系统共有 123 种指令，可分为 9 类指令操作：数据传递、算术运算、逻辑运算、串操作、位操作、程序控制、高级语言指令、保护模式；处理器控制指令。

程序是计算机为了执行某种操作任务而将一条条指令按照一定的顺序排列起来的指令集。

1.3.2 计算机语言

程序就是一系列的操作步骤，计算机程序就是由人事先规定的计算机完成某项工作的操作步骤。每一步骤的具体内容由计算机能够理解的指令来描述，这些指令告诉计算机“做什么”和“怎样做”。编写计算机程序所使用的语言称为程序设计语言。

通常分为三类：机器语言、汇编语言和高级语言。

1. 机器语言

计算机最早的语言处理程序是机器语言，它是计算机能直接识别的语言，而且速度快。机器语言是用二进制代码来编写计算机程序，因此又称二进制语言。例如用机器语言来表示“8+4”这个算式，是一串二进制码“00001000 00000100 00000100”。机器语言书写困难、记忆复杂，一般很难掌握。

2.汇编语言

由于机器语言的缺陷，人们开始用助记符编写程序，用一些符号代替机器指令所产生的语言称为汇编语言。但是用汇编语言编写的源程序不能被计算机直接识别，必须使用某种特殊的软件将用汇编语言写的源程序翻译和连接成能被计算机直接识别的二进制代码。其示意图如图所示。

汇编语言虽然采用了助记符来编写程序，比机器语言简单，但是汇编语言仍属于低级语言，它与计算机的体系结构有关，在编写程序前要花费相当多的时间和精力去熟悉机器的结构。因此工作量大、繁琐，而且程序可移植性差。

3.高级语言

计算机并不能直接地接受和执行用高级语言编写的源程序,源程序在输入计算机时,通过“翻译程序”翻译成机器语言形式的目标程序,计算机才能识别和执行。这种“翻译”通常有两种方式,即编译方式和解释方式。

编译方式是：编译方式的翻译工作由“编译程序”来完成，它是先将整个源程序都转换成二进制代码，生成目标程序，然后把目标程序连接成可执行的程序，以完成源程序要处理的运算并取得结果。

解释方式是：源程序进入计算机时,解释程序边扫描边解释，对源程序的语句解释一条，执行一条，不产生目标程序。解释方式的翻译工作由“解释程序”

编译性语言有 C/C++、Pascal/Object Pascal（Delphi）等

解释性语言有 ASP、PHP、Java、JavaScript、VBScript、Perl、Python、Ruby 等

使用编译语言程序将整个源程序编译连接可执行的文件，这种方式效率高、可靠性高、可以移植性好。不过当源程序修改后，必需重新编译。

1.3.3 作业

1. Linux 是一种(　　)。

A. 绘图软件　　B. 程序设计语言　　C. 操作系统　　D. 网络浏览器

2. 下列关于高级语言的说法错误的是（　　）。

A. Fortran 是历史上的第一个面向科学计算的高级语言

B. Pascal 和 C 都是编译执行的高级语言

C. C++是历史上的第一个支持面向对象的语言

D. 编译器将高级语言程序转变为目标代码

3. 在下列关于计算机语言的说法中，不正确的是（　　）。

A. Pascal 和 C 都是编译执行的高级语言

B. 高级语言程序比汇编语言程序更容易从一种计算机移植到另一种计算机上

C. C++是历史上的第一个支持面向对象的计算机语言

D. 与汇编语言相比，高级语言程序更容易阅读

4. 在关系数据库中，存放在数据库中的数据的逻辑结构以（　　）为主。

A．二叉树　　B．多叉树　　C．哈希表　　D．二维表

5．在下列关于计算机语言的说法中，正确的有（　　）。

A．高级语言比汇编语言更高级，是因为它的程序的运行效率更高

B．随着 Pascal、C 等高级语言的出现，机器语言和汇编语言已经退出了历史舞台

C．高级语言比汇编语言程序更容易从一种计算机上移植到另一种计算机上

D．C 是一种面向对象的高级计算机语言

6. 在以下各项中，（　　）不是操作系统软件。

A. Solaris　　B. Linux　　C. Windows Vista　　D. Sybase

7. 面向对象程序设计（Object-Oriented Programming）是一种程序设计的方法论，它将对象作为程序的基本单元，将数据和程序封装在对象中，以提高软件的重用性、灵活性和扩展性。下面关于面向对象程序设计的说法中，不正确的是（　　）。

A. 面向对象程序设计通常采用自顶向下设计方法进行设计。

B. 面向对象程序设计方法具有继承性（inheritance）、封装性（encapsulation）、多态性（polymorphism）等几大特点。

C. 支持面向对象特性的语言称为面向对象的编程语言，目前较为流行的有 C++、JAVA、C#等。

D. 面向对象的程序设计的雏形来自于 Simula 语言，后来在 SmallTalk 语言的完善和标准化的过程中得到更多的扩展和对以前思想的重新注解。至今，SmallTalk 语言仍然被视为面向对象语言的基础。

8. 下列软件中不是计算机操作系统的是：（　　）

A. Windows　　B. Linux　　C. OS/2　　D. WPS

9. 关于程序设计语言，下面哪个说法是正确的：（　　）

A. 加了注释的程序一般会比同样的没有加注释的程序运行速度慢。

B. 高级语言开发的程序不能使用在低层次的硬件系统如：自控机床或低端手机上。

C. 高级语言相对于低级语言更容易实现跨平台的移植。

D. 以上说法都不对。

10. Pascal 语言、C 语言和 C++语言都属于（　　）。

A. 面向对象语言　　B. 脚本语言　　C. 解释性语言　　D. 编译性语言

11.有人认为，在个人电脑送修前，将文件放入回收站中就是已经将其删除了。这种想法是（　　）。

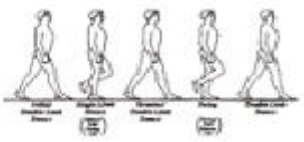
A．正确的，将文件放入回收站以为着彻底删除、无法恢复

B．不正确的，只有将回收站清空后，才意味着彻底删除、无法恢复

C．不正确的，即使回收站清空，文件只是被标记为删除，仍可能通过回复软件找回

D．不正确的，只要在硬盘上出现过的文件，永远不可能被彻底删除

12.生物特征识别，是利用人体本身的生物特征进行身份认证的一种技术。目前，指纹识别、虹膜识别、人脸识别等技术已广泛应用于政府、银行、安全防卫等领域。一下不属于生物特征识别技术及其应用的是（　　）。

A．指静脉验证　　B．步态验证　　C．ATM 机密码验证　　D．声音验证

13．关于汇编语言，下列说法错误的是（　　）

A．是一种与具体硬件相关的程序设计语言

B．在编写复杂程序时，相对于高级语言而言代码量较大，且不易调试

C．可以直接反问寄存器、内存单元、以及 I/O 端口

D．随着高级语言的诞生，如今已完全被淘汰，不再使用

14．（　　）不属于操作系统。

A．Windows　　B．DOS　　C．Photoshop　　D．NOI Linux

15．仿生学的问世开辟了独特的科学技术发展道路。人们研究生物体的结构、功能和工作原理，并将这些原理移植于新兴的工程技术中。以下关于仿生学的叙述，错误的是（　　）

A．由研究蝙蝠，发明雷达　　B．由研究蜘蛛网，发明因特网

C．由研究海豚，发明声纳　　D．由研究电鱼，发明伏特电池

16．以下哪个是面向对象的高级语言(　　)。

A.汇编语言　　B.C++　　C.Fortran

D.Basic

17.下列选项中不属于图像格式的是(　　)。

A.JPEG 格式　　B.TXT 格式　　C.GIF 格式　　D.PNG 格式

18．操作系统的作用是（　　）。

A.把源程序译成目标程序　　B.便于进行数据管理

C．　控制和管理系统资源　　D.实现硬件之间的连接

19．在计算机内部用来传送、存贮、加工处理的数据或指令都是以（　　）形式进行的。

A．二进制码　　B.八进制码　　C.十进制码　　D.智能拼音码

20．下列选项中不属于视频文件格式的是（　　）。

A．TXT　　B．AVI　　C．MOV　　D．RMVB

1.4 数制与编码

1.4.1 进制转换

一、进位计数制的基本概念

将数字符号按序排列成数位，并遵照某种由低位到高位的进位方式计数表示数值的方法，称作进位计数制。

1．十进制

十进制计数制由 0、 1、 2、 3、 4、 5、6、 7、 8、 9 共 10 个数字符号组成。相同数字符号在不同的数位上表示不同的数值，每个数位计满十就向高位进一，即“逢十进一”。

2．八进制

八进制计数制由 0、 1、 2、3、 4、 5、 6、 7 共 8 个数字符号组成。相同数字符号在不同的数位上表示不同的数值，每个数位计满八就向高位进一，即“逢八进一”。

3. 二进制

二进制计数制由 0 和 1 共 2 个数字符号组成。相同数字符号在不同的数位上表示不同的数值，每个数位计满二就向高位进一，即“逢二进一”。

4. 其他进制

在日常生活和日常工作中还会使用其他进制数。如：十二进制数、十六进制数、百进制数和千进制数等。无论哪种进制数，表示的方法都是类似的。如：十六进制数由 0、 1、 2、 3、 4、 5、 6、 7、 8、 9、 A、 B、 C、 D、 E 和 F 共十六个符号组成，“逢十六进一”。不同的是用 A、 B、 C、D、 E 和 F 分别表示 10、 11、 12、 13、 14 和 15 六个数字符号。

5. 基数与权

某进制计数制允许选用的基本数字符号的个数称为基数。一般而言，J 进制数的基数为 J，可供选用的基本数字符号有 J 个，分别为 0 到 J-1，每个数位计满 J 就向高位进一，即“逢 J 进一”。

某进制计数制中各位数字符号所表示的数值表示该数字符号值乘以一个与数字符号有关的常数，该常数称为“位权”（简称“权”）。位权的大小是以基数为底，数字符号所处的位置的序号为指数的整数次幂。

十进制数允许使用十个基本数字符号，所以基数为 10，每位数字符号代表的位数的大小是以 10 为底，数字符号所处位置的序号为指数的整数次幂。

十进制数的百位，十位，个位和十分位的权分别为：$10^2, 10^1, 10^0, 10^{-1}$。故$(555.5)_{10}$可表示成$(555.5)_{10}=5*10^2+5*10^1+5*10^0+5*10^{-1}$。J 进制数相邻两位数相差 J 倍，若小数点向左移 n 位，则整个数值就缩小 J^n。反之，小数点向右移 n 位，数值就放大 J^n。

如图所示，给出了任意进制数（ K_2 K_1 K_0 K_{-1} K_{-2}），当 J 分别为： 2， 8， 10 和 16 时各位权值对照。

数位 / 权 / 进位制 J	K_2	K_1	K_0	小数点位置	K_{-1}	K_{-2}
J=2	$2^2=4$	$2^1=2$	$2^0=1$		$2^{-1}=0.5$	$2^{-2}=0.25$
J=8	$8^2=64$	$8^1=8$	$8^0=1$		$8^{-1}=0.125$	$8^{-2}=0.015625$
J=10	$10^2=100$	$10^1=10$	$10^0=1$		$10^{-1}=0.1$	$10^{-2}=0.01$
J=16	$16^2=256$	$16^1=16$	$16^0=1$		$16^{-1}=0.0625$	$16^{-1}=0.00390625$

二、数制之间的转换：

计算机内部使用的数字符号只有“0”和“1”两个。也就是说计算机内部使用的是二进制数所有的数值数据和非数值数据，都是由“0”和“1”这两个数字符号加以组合而成的，我们称之为“二进制代码”。

计算机只用二进制的两个数码“0”和“1”来实现算术和逻辑运算，而人们仍然用十进制的形式向计算机中输入原始数据，并让计算机也用十进制形式显示和打印运算结果。所以必须有一种自动转换方法，即让数据输入计算机后，将十进制转换成对应的二进制数，并在处理完毕后，再自动将二进制结果转换为十进制数。

为了表达方便起见，常在数字后加一缩写字母后缀作为不同进制数的标识。各种进制数的后缀字母分别为：

B ：二进制数。　　O ：八进制数。

D ：十进制数。　　H ：十六进制数。

对于十进制数通常不加后缀，也即十进制数后的字母 D 可省略。

1．二进制与十进制的转换

（1）二进制转十进制

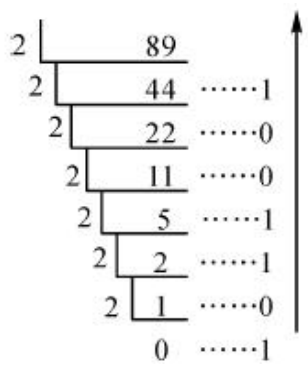
方法：“按权展开求和”

$$例：(1011.01)_2=(1*2^3+0*2^2+1*2^1+1*2^0+0*2^{-1}+1*2^{-2})_{10}$$
$$=(8+0+2+1+0+0.25)_{10}$$
$$=(11.25)_{10}$$

（2）十进制转二进制

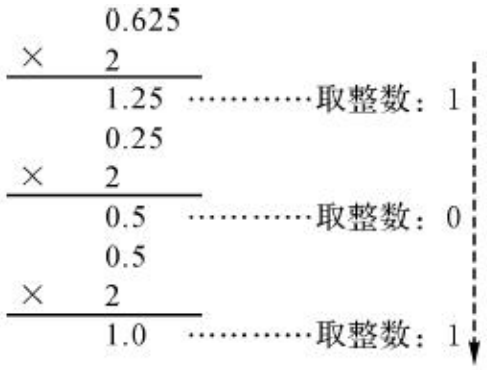
·十进制整数转二进制数：“除以 2 取余，逆序输出”

例：$(89)_{10}=(1011001)_2$



·十进制小数转二进制数：“乘以 2 取整，顺序输出”

例：　$(0.625)_{10}=(0.101)_2$



2．八进制与二进制的转换

例：将八进制的 37.416 转换成二进制数：

3　7　.　4　1　6

011　111　.　100　001　110

即：$(37.416)_8 = (11111.10000111)_2$

例：将二进制的 10110.0011 转换成八进制：

$$\frac{010}{2}\ \frac{110}{6}.\frac{001}{1}\ \frac{100}{4}$$

即：$(10110.011)_2 = (26.14)_8$

3．十六进制与二进制的转换

例：将十六进制数 5DF.9 转换成二进制：

5　D　F　.　9

0101　1101　1111　.　1001

即：$(5DF.9)_{16} = (10111011111.1001)_2$

例：将二进制数 1100001.111 转换成十六进制：

0110　0001　.　1110

6　1　.　E

即：$(1100001.111)_2 = (61.E)_{16}$

1.4.2 信息编码表示

一、基本概念

1.编码:计算机要处理的数据除了数值数据以外，还有各类符号、图形、图像和声音等非数值数据。而计算机只能识别两个数字。要使计算机能处理这些信息，首先必须将各类信息转换成“0”和“1”表示的代码，这一过程成为编码。

2.数据:能被计算机接受和处理的符号的集合都称为数据。

3.比特:比特（Bit：——二进制数位）是指 1 位二进制的数码（即 0 或 1)。比特是计算机中表示信息的数据编码中的最小单位。

4.字节:字节表示被处理的一组连续的二进制数字。通常用 8 位二进制数字表示一个字节，即一个字节由 8 个比特组成。

字节是存储器系统的最小存取单位。

二、字符的表示

字符是人与计算机交互过程中不可缺少的重要信息。要使计算机能处理、存储字符信息，首先也必须用二进制“0”和“1”代码对字符进行编码。

下面以西文字符和汉字字符为例，介绍常用的编码标准。

三、ASCII 编码

ASCII 编码是由美国国家标准委员会制定的一种包括数字、字母、通用符号和控制符号在内的字符编码集，全称叫美国国家信息交换标准代码（American StandardCode for Information Interchange）。

ASCII 码是一种 7 位二进制编码，能表示 27＝128 种国际上最通用的西文字符，是目前计算机中，特别是微型计算机中使用最普遍的字符编码集。

“0” —— 48
“A” —— 65
“a” —— 97

ASCII 编码包括 4 类最常用的字符。

①数字“0”～“9”。ASCII 编码的值分别为 0110000B ～ 0111001B ，对应十六进制数为 30H～39H。

②26 个英文字母。大写字母“A”～“Z”的 ASCII 编码值为 41H～5AH，小写字母“a”～“z”的 ASCII 编码值为 61H ～ 7AH。

③通用符号。如“＋”、“－”、“＝”、“*”和“/”等共 32 个。

④控制符号。如空格符和回车符等共 34 个。

ASCII 码是一种 7 位编码，它存时必须占全一个字节，即占用 8 位：b7 b6 b5 b4 b3 b2 b1 b0，其中 b7 恒为 0，其余几位为 ASCII 码值。

十进制	字符	十进制	字符
48	0	97	a
49	1	98	b
50	2	99	c
…	…	…	…
57	9	122	z
65	A		
66	B		
67	C		
…			
90	z		

人们可以通过键盘和显示器输入和显示不同的字符，但在计算机中，所有信息都是用二进制代码表示。n 位二进制代码能表示 2n 个不同的字符，这些字符的不同组合就可表示不同的信息。为使计算机使用的数据能共享和传递，必须对字符进行统一的编码。 ASCII 码(美国标准信息交换码)是使用最广泛的一种编码。ASCII 码由基本的 ASCII 码和扩充的 ASCII 码组成。在 ASCII 码中，把二进制位最高位为 0 的数字都称为基本的 ASCII 码，其范围是 0～127；把二进制位最高位为 1 的数字都称为扩展的 ASCII 码，其范围是 128～255。

四、内码和外码

内码：对于输入计算机的文本文件，机器是存储其相应的字符的 ASCII 码（用一个 ASCII 码存储一个字符需 8 个二进制位，即一个字节），这些可被计算机内部进行存储和运算使用的数字代码称内码。如输入字符“A”，计算机将其转成内码 65 后存于内存。

外码：计算机与人进行交换的字形符号称为外码，如字符"A"的外码是"A"。通常一个西文字符占一个字节（半角），一个中文字符占二个字节。

五、汉字信息编码

1．汉字交换码

汉字交换码是指不同的具有汉字处理功能的计算机系统之间在交换汉字信息时所使用的代码标准。国家标准 GB2312－80 公布以来，我国一直延用该标准所规定的国标码作为统一的汉字信息交换码(GB5007-85 图形字符代码)。

GB2312－80 标准包括了 6763 个汉字,按其使用频度分为一级汉字 3755 个和二级汉字 3008 个。一级汉字按拼音排序，二级汉字按部首排序。该标准还包括标点符号、数种西文字母、图形、数码等符号 682 个。

区位码的区码和位码均采用从 01 到 94 的十进制，国标码采用十六进制的 21H 到 73H（数字后加 H 表示其为十六进制数）。区位码和国标码的换算关系是：区码和位码分别加上十进制数 32。如“国”字在表中的 25 行 90 列，其区位码为 2590，国标码是 397AH。

2．字形存储码

字形存储码是指供计算机输出汉字（显示或打印）用的二进制信息，也称字模。通常，采用的是数字化点阵字模。

一般的点阵规模有 16×16，24×24 等，每一个点在存储器中用一个二进制位（bit）存储。在 16×16 的点阵中，需 8×32 bit 的存储空间，每 8 bit 为 1 字节，所以，需 32 字节的存储空间。在相同点阵中，不管其笔划繁简，每个汉字所占的字节数相等。

为了节省存储空间，普遍采用字形数据压缩技术。所谓的矢量汉字是指用矢量方法将汉字点阵字模进行压缩后得到的汉字字形的数字化信息。

五、数的原码、补码和反码表示

（1）机器数与真值

在计算机中，表示数值的数字符号只有 0 和 1 两个数码，我们规定最高位符号位，并用 0 表示正数符号，用 1 表示负数符号。这样，机器中的数值和符号全“数码化”了。为简化机器中数据的运算操作，人们采用了原码、补码、反码及移码等几种方法对数值位和符号位统一进行编码。为区别起见，我们将数在机器中的这些编码表示称为机器数（如：10000001），而将原来一般书写表示的数称为机器数的真值（如：-0000001）。

（2）原码表示法

原码表示法是一种简单的机器数表示法，即符号和数值表示法，设 x 为真值，[x]原为机器数表示。

例：设 x=1100110，则[x]原=01100110

x=-1100111，则[x]原=11100111

（3）反码表示法

正数的反码就是真值本身；负数的反码，只须对符号位以外各位按位“求反”(0 变 1，1 变 0)即可。

例：设 x=1100110，则[x]反=01100110

x=-1100111，则[x]反=10011000

（4）补码表示法

负数用补码表示时，可以把减法转化成加法。正数的补码就是真值本身；负数的补码是符号位为 1，数值各位取反（0 变为 1、1 变为 0），最低位加 1。

例：设 x=1100110，则[x]补=01100110

x=-1100111，则[x]补=10011001

从上面关于原码、反码、补码的定义可知：一个正数的原码、反码、补码的表示形式相同，符号位为 0，数值位是真值本身；一个负数的原码、反码、补码的符号位都为 1，数值位原码是真值本身，反码是各位取反，补码是各位取反，最低位加 1。真值 0 的原码和反码表示不唯一，而补码表示是唯一的，即

[+0]原=000…0，[-0]原=100…0

[+0]反=000…0，[-0]反=111…1

[+0]补=[-0]补=000…0

注：不同编码表示的整数的范围是这样这样的（以N位二进制位）:

原码：0--2n-1(无符号)　　-2n-1-1--2n-1-1(有符号)

反码：-2n-1-1--2n-1-1 (不存在无符号的情况)

补码：-2n-1--2n-1-1 (不存在无符号的情况)

大家很明显看出，补码表示的范围最大。现以八位二进制位为例说明如下：

原码：00000000-11111111　即 0-255(无符号)

11111111-01111111　即-127-+127(有符号)

反码：10000000-01111111

即-127-+127(因为 10000000 的值为-127，11111111 的值为-0)

补吗：10000000-01111111

即-128-+127(因为 10000000 的值为-128，11111111 的值为-1)

如果没说具体编码形式，则计算机中N位二进制无符号数的范围是 0--2n-1；有符号数的范围是-2n-1--2n-1-1

以 8 位有符号整数为例子:

原码:若 X 为正数,则最高位(符号位)为 0,其余按照二进制数排列

若 X 为负数,则最高位为 1,后面和正数原码一样

例: +7:　00000111　　-7:　10000111

+0:　00000000　　-0:　10000000

反码:若 X 为正数,则反码与原码相同

若 X 为负数,则将原码除符号位取反(就是这个位置 1 变成 0,0 变成 1)

例:　-7: 11111000

补码:反码 +1　　例:　-7: 1111001

二、数的定点表示和浮点表示

在计算机中小数点一般有两种表示法：一种是小数点固定在某一位置的定点表示法；另一种是小数点的位置可任意移动的浮点表示法。相应于这两种表示的计算机分别称为定点计算机和浮点计算机。

（1）定点表示法

机器中所有数的小数点位置是固定不变的，因而小数点就不必使用记号表示出来。实际上，小数点可固定在任意一个位置上。

（2）浮点表示法

在数的定点表示法中，由于数的表示范围较窄常常不能满足各种数值问题的需要。为了扩大数的表示范围，方便用户使用，有些计算机常采用浮点表示法。表示一个浮点数，要用两部分：尾数和阶码。尾数用以表示数的有效数值；阶码用以表示小数点在该数中的位置。

计算机多数情况下采作浮点数表示数值，它与科学计数法相似，把一个二进制数通过移动小数点位置表示成阶码和尾数两部分：

1.4.3 作业

1. 一位艺术史学家有 20000 幅真彩色图像，每幅图像约占 3M 空间。如果将这些图像以位图形式保存在 CD 光盘上（一张 CD 光盘的容量按 600M 计算），大约需要（　　）张 CD 光盘。

A. 1　　B. 10　　C. 100　　D. 1000

2. 设 A = true，B = false，C = false，D = true，以下逻辑运算表达式值为真的是（　　）。

A. (A∧B)∨(C∧D)　　B. ((A∧B)∨C)∧D

C. A∧((B∨C) ∧D)　　D. (A∧(B∨C))∨D

3. $(3725)_8$ + $(B)_{16}$的运算结果是（　　）。

A. $(3736)_8$　　B. $(2016)_{10}$　　C. $(1111110000)_2$　　D. $(3006)_{10}$

4. 在 C++ 中，表达式 21^2 的值是（　　）

A. 441　　B. 42　　C.23　　D.24

5. 在 C++ 中，判断 a 不等于 0 且 b 不等于 0 的正确的条件表达式是（　　）

A. !a==0 || !b==0　　B. !((a==0)&&(b==0))

C. !(a==0&&b==0)　　D. a && b

6. 与十进制数 1770 对应的八进制数是（　　）。

A.3350　　B.3351　　C.3352　　D.3540

7. 设 A=B=D=true，C=false，以下逻辑运算表达式值为真的有（　　）。

A. (¬A∧B)∨(C∧D)　　B.¬((A∨B∨D)∧C)

C. ¬A∧(B∨C∨D)　　D.(A∧B∧C)∨¬D

8. $(2010)_{16}$ + $(32)_8$的结果是（　　）。

A.$(8234)_{10}$　　B.$(202B)_{16}$　　C.$(20056)_8$　　D.$(100000000110)_2$

9. 在下列各项中，只有（　　）不是计算机存储容量的常用单位。

A. Byte　　B. KB　　C. UB　　D. TB

10. ASCII 码的含义是（　　）。

A. 二→十进制转换码　　B. 美国信息交换标准代码

C. 数字的二进制编码　　D. 计算机可处理字符的唯一编码

11. 表达式 （23 || 2 ^ 5）的值是（　　）。

A. 18　　B. 1　　C. 23　　D. 32

12. 与十进制数 1770 对应的八进制数是（　　）。

A. 3350　　B. 3351　　C. 3352　　D. 3540

13. 设 A=B=True，C=D=False，一下逻辑运算表达式值为假的有（　　）。

A. (¬A∧B)∨(C∧D∨A)　　B. ¬(((A∧B)∨C)∧D)

C. A∧(B∨C∨D)∨D　　D. (A∧(D∨C))∧B

14. $(2070)_{16}$ + $(34)_{8}$ 的结果是（　　）。

A.（8332）$_{10}$　　B.（208A）$_{16}$　　C.（100000000110）$_{2}$　　D. $(20212)_{8}$

15. 设 A=true，B=false，C=true，D=false，以下逻辑运算表达式值为真的是（　　）。

A. (A∧B)∨(C∧D∨¬A)　　B. ((¬A∧B)∨C)∧¬D

C. (B∨C∨D)∧D∧A　　D. A∧(D∨¬C)∧B

16. 与十进制数 28.5625 相等的四进制数是（　　）。

A. 123.21　　B. 131.22　　C. 130.22　　D. 130.21

17. $(2008)_{10}$ + $(5B)_{16}$的结果是（　　）。

A. $(833)_{16}$　　B. $(2089)_{10}$　　C. $(4163)_{8}$　　D. $(100001100011)_{2}$

18. 在 32*32 点阵的“字库”中，汉字“北”与“京”的字模占用字节数之和是（　　）。

A. 512　　B. 256　　C. 384　　D. 128

19. 在 C++程序中，表达式 200|10 的值是（　　）

A. 20　　B. 1　　C. 220　　D. 202

20. 关于 ASCII，下面哪个说法是正确的：（　　）

A. ASCII 码就是键盘上所有键的唯一编码。

B. 一个 ASCII 码使用一个字节的内存空间就能够存放。

C. 最新扩展的 ASCII 编码方案包含了汉字和其他欧洲语言的编码。

D. ASCII 码是英国人主持制定并推广使用的。

21. 已知大写字母A的ASCII编码为65(10进制),则大写字母J的10进制ASCII编码为:(　　)

A. 71　　B. 72　　C. 73　　D. 以上都不是

22.十进制小数 125.125 对应的 8 进制数是(　　)

A. 100.1　　B. 175.175　　C. 175.1　　D. 100.175

23. 2E+03 表示（　　）。

A. 2.03　　B. 5　　C. 8　　D. 2000

24. 一个字节（byte）由（　　）个二进制位组成。

A. 8　　B. 16　　C. 32　　D. 以上都有可能

25. 以下逻辑表达式的值恒为真的是（　　）。

A. P∨(¬P∧Q)∨(¬P∧¬Q)　　B. Q∨(¬P∧Q)∨(P∧¬Q)

C. P∨Q∨(P∧¬Q)∨(¬P∧Q)　　D. P∨¬Q∨(P∧¬Q)∨(¬P∧¬Q)

26. 一个字长为 8 位的整数的补码是 11111001，则它的原码是（　　）。

A. 00000111　　B. 01111001　　C. 11111001　　D. 10000111

27. 在二进制下，1011001 + （　　） = 1100110。

A. 1011　　B. 1101　　C. 1010　　D. 1111

28. 字符“0”的 ASCII 码为 48，则字符“9”的 ASCII 码为（　　）。

A．39　　B．57　　C．120　　D．视具体的计算机而定

29．一片容量为 8G 的 SD 卡能储存大约（　　）张大小为 2MB 的数码照片。

A．1600　　B．2000　　C．4000　　D．16000

30．一个正整数在二进制下有 100 位，则它在十六进制下有（　　）位。

A．7　　B．13　　C．25　　D．不能确定

31．十六进制数 9A 在（　　）进制下是 232。

A．四　　B．八　　C．十　　D．十二

32．矢量图（Vector Image）图形文件所占的贮存空间比较小，并且无论如何放大、缩小或旋转等都不会失真，是因为它（　　）。

A．记录了大量像素块的色彩值来表示图像

B．用点、直线或者多边形等基于数学方程的几何图元来表示图像

C．每个像素点的颜色信息均用矢量表示

D．把文件保存在互联网，采用在线浏览的方式查看图像

33．一个 32 位整型变量占用（　　）个字节。

A．4　　B．8　　C．32　　D．128

34．二进制数 11.01 在十进制下是（　　）。

A．3.25　　B．4.125　　C．6.25　　D．11.125

35．逻辑表达式（　　）的值与变量 A 的真假无关。

A．(A ∨ B) ∧¬A　　B．(A ∨ B) ∧¬B

C．(A ∧ B) ∨ (¬ A ∧ B)　　D．(A ∨ B) ∧¬A ∧ B

36．在十六进制表示法中，字母 A 相当于十进制中的（　　）。

A．9　　B．10　　C．15　　D．16

37．二进制数 00100100 和 00010101 的和是(　　)。

A.00101000　　B.001010100　　C.01000101　　D.00111001

38 下列各无符号十进制整数中，能用八位二进制表示的数中最大的是(　　)。

A.296　　B.133　　C.256　　D.199

39．二进制数 00100100 和 00010100 的和是（　　）。

A．00101000　　B．01000001　　C．01000100　　D．00111000

40．与二进制小数 0.1 相等的十六进制数是（　　）

A．0.8　　B．0.4　　C．0.2　　D．0.1

41．如果 256 种颜色用二进制编码来表示，至少需要（　　）位。

A.6　　B．7　　C.8　　D．9

42．二进制数 00101100 和 00010101 的和是（　　）。

A.00101000　　B．01000001　　C．01000100　　D．00111000

43．与二进制小数 0.1 相等的八进制数是（　　）。

A．0.8　　B．0.4　　C．0.2　　D．0.1

1.5 安全与网络

1.5.1 计算机安全

计算机安全中最重要的是存储数据的安全，其面临的主要威胁包括：计算机病毒、非法访问、计算机电磁辐射、硬件损坏等。

计算机病毒是附在计算机软件中的隐蔽的小程序，它和计算机其他工作程序一样，但会破坏正常的程序和数据文件。恶性病毒可使整个计算机软件系统崩溃，数据全毁。要防止病毒侵袭主要是加强管理，不访问不安全的数据，使用杀毒软件并及时升级更新。

由于计算机硬件本身就是向空间辐射的强大的脉冲源，如和一个小电台差不多，频率在几十千周到上百兆周。盗窃者可以接收计算机辐射出来的电磁波，进行复原，获取计算机中的数据。为此，计算机制造厂家增加了防辐射的措施，从芯片，电磁器件到线路板、电源、转盘、硬盘、显示器及连接线，都全面屏蔽起来，以防电磁波辐射。更进一步，可将机房或整个办公大楼都屏蔽起来，如没有条件建屏蔽机房，可以使用干扰器，发出干扰信号，使接收者无法正常接收有用信号。

计算机存储器硬件损坏，使计算机存储数据读不出来也是常见的事。防止这类事故的发生有几种办法，一是将有用数据定期复制出来保存，一旦机器有故障，可在修复后把有用数据复制回去。二是在计算机中使用 RAID 技术，同时将数据存在多个硬盘上；在安全性要求高的特殊场合还可以使用双主机，一台主机出问题，另外一台主机照样运行。

计算机硬件安全：

计算机在使用过程中，对外部环境有一定的要求，即计算机周围的环境应尽量保持清洁、温度和湿度应该合适、电压稳定，以保证计算机硬件可靠的运行。计算机安全的另外一项技术就是加固技术，经过加固技术生产的计算机防震、防水、防化学腐蚀，可以使计算机在野外全天候运行。

从系统安全的角度来看，计算机的芯片和硬件设备也会对系统安全构成威胁。比如 CPU，电脑 CPU 内部集成有运行系统的指令集，这些指令代码是都是保密的，我们并不知道它的安全性如何。据有关资料透漏，国外针对中国所用的 CPU 可能集成有陷阱指令、病毒指令，并设有激活办法和无线接收指令机构。他们可以利用无线代码激活 CPU 内部指令，造成计算机内部信息外泄、计算机系统灾难性崩溃。如果这是真的，那我们的计算机系统在战争时期有可能全面被攻击。

硬件泄密甚至涉及了电源。电源泄密的原理是通过市电电线，把电脑产生的电磁信号沿电线传出去，利用特殊设备可以从电源线上就可以把信号截取下来还原。

计算机里的每一个部件都是可控的，所以叫做可编程控制芯片，如果掌握了控制芯片的程序，就控制了电脑芯片。只要能控制，那么它就是不安全的。因此，我们在使用计算机时首先要注意做好电脑硬件的安全防护，把我们所能做到的全部做好！

1.5.2 计算机网络

一、网络的定义：

所谓计算机网络，就是利用通信线路和设备，把分布在不同地理位置上的多台计算机连接起来。

计算机网络是现代通信技术与计算机技术相结合的产物。

网络中计算机与计算机之间的通信依靠协议进行。协议是计算机收、发数据的规则。

TCP/IP：用于网络的一组通讯协议。包括 IP(Internet Protocol)和 TCP(Transmission Control Protocol)

二、网络的发展

计算机网络的发展过程大致可以分为三个阶段：

远程终端联机阶段：主机—终端

计算机网络阶段：计算机—计算机

Internet 阶段： Internet

三、网络的主要功能：

（1）资源共享 （2）信息传输

（3）分布处理 （4）综合信息服务

四、网络的分类

计算机网络的分类按网络的地理范围进行分类：局域网（LAN）、城域网（MAN）、广域网（WAN）

1、局域网（Local Area Network）：一般局限在 1KM 的范围内，局域网内传输速率较高，误码率低，结构简单容易实现，具体标准是美国电气工程师协会制订的 IEEE802 系列标准。

2、城域网（Metropolitan Area Network）：其范围几千米到几十千以内。

3、广域网（Wide Area Network）：其范围在几十千米到几千千米以上注：MAN 和 WAN 一般都是由多个 LAN 构成的。

五、网络拓扑结构

按网络的拓扑结构进行分类：星型、总线型、环型、树型、网状形

1、星型：通信协议简单，对外围站点要求不高，单个站点故障不会影响全网，电路利用率低，连线费用大，网络性能依赖中央结点，每个站点需要有一个专用链路。

2、总线型：结构简单，可靠性高；布线容易，连线总长度小于星型结构；总线任务重，易产生瓶颈问题；总线本身的故障对系统是毁灭性的。

3、环型：传输速率高，传输距离远；各节点的地位和作用相同；各节点传输信息的时间固定；容易实现分布式控制；站点的故障会形起整个网络的崩溃。

4、树型：分级结构，又称为分级的集中式网络。

5、网状形（不规则形）：计算机之间无规则地连接，一般广域网属于不规则形。

网络拓扑结构是指计算机网络节点和通信链路所组成的几何形状。

Internet 网是当今世界上规模最大、用户最多、影响最广泛的计算机互联网络。 Internet 网上联有大大小小成千上万个不同拓扑结构的局域网、城域网和广域网。因此，Internet 网本身的拓扑只是一种虚拟拓扑结构，无固定形式。

按所采用的交换技术进行分类：电路交换、报文交换、分组交换

五、网络的体系结构

ISO 的 OSI 七层参考模型

国际标准化组织（ISO，International Standardization Organization）提出的开放式系统互联(OSI，Open System Interconnection)参考模型。它将数据从一个站点到达另一个站点的工作按层分割成七个不同的任务。

开放性是指任何遵循 OSI 标准的系统，只要物理上连结起来，它们之间都可以相互通信。OSI 参考模型并不是网络体系结构。OSI 只是描述了每一层的功能，并没有经确定一个层的协议。而网络体系结构是网络层次结构和相关协议的集合。

六、IP 地址：

IP 地址：在 TCP/IP 体系中，IP 地址是一个最重要的概念，什么叫 IP 地址?

所谓 IP 地址，是用于标识 IP Internet 网络上节点的 32 位地址(以后可能使用的 V6 版本是 128 位的，分 8 组，每组 16 位)。对于 IP Internet 网络上的每个节点都必须指派一个唯一的地址，它由网络 ID 和唯一的主机 ID 组成。该地址通常用由句点分隔的八位字节的十进制数表示(例:192.168.7.27)。

IP 地址的分类：Internet 的 IP 地址分成 A、B、C、D、E 五类，其中 A、B、C 为常用类，都由网络 ID 和主机 ID 两个部分组成，网络 ID 也叫做网络地址，标识大规模 TCP/IP 网际网络(由网络组成的网络)内的单个网段,连接到并共享访问同一网络的所有系统在其完整的 IP 地址内都有一个公用的网络 ID，这个 ID 也用于唯一地识别大规模的网际网络内部的每个网络；主机 ID，也叫做主机地址，识别每个网络内部的 TCP/IP 节点(工作站、服务器、路由器或其他 TCP/IP 设备)，每个设备的主机 ID 唯一地识别所在网络内的单个系统。

1.5.3 作业

1. 下列网络上常用的名字缩写对应的中文解释错误的是（　　）。

A. WWW（World Wide Web）：万维网。

B. URL（Uniform Resource Locator）：统一资源定位器。

C. HTTP（Hypertext Transfer Protocol）：超文本传输协议。

D. FTP（File Transfer Protocol）：快速传输协议。

2. 常见的邮件传输服务器使用（　　）协议接收邮件。

A. HTTP　　B. SMTP　　C. TCP　　D. POP3

3. LAN 的含义是（　　）。

A. 因特网　　B. 局域网　　C. 广域网　　D. 城域网

4. 关于互联网，下面的说法哪一个是正确的：（　　）

A. 新一代互联网使用的 IPv6 标准是 IPv5 标准的升级与补充。

B. 互联网的入网主机如果有了域名就不再需要 IP 地址。

C. 互联网的基础协议为 TCP/IP 协议。

D. 互联网上所有可下载的软件及数据资源都是可以合法免费使用的。

5、关于 HTML 下面哪种说法是正确的：（　　）

A. HTML 实现了文本、图形、声音乃至视频信息的统一编码。

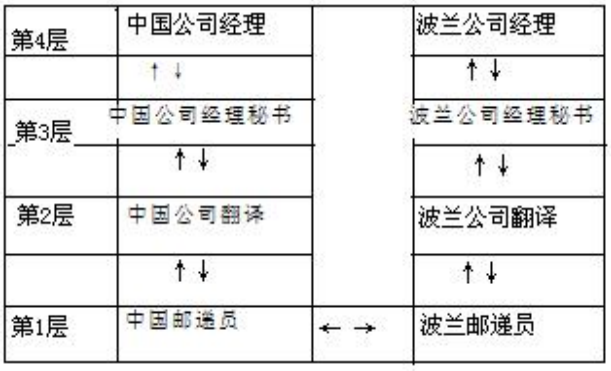
B. HTML 全称为超文本标记语言。

C. 网上广泛使用的 Flash 动画都是由 HTML 编写的。

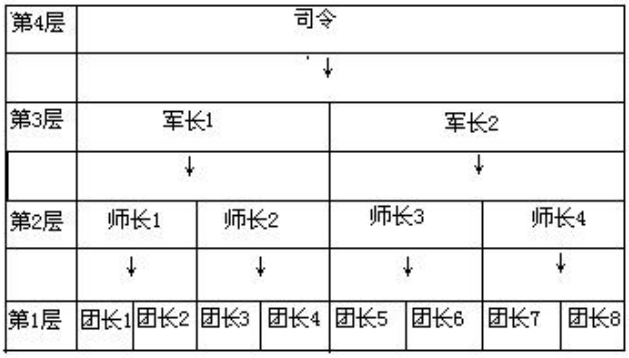
D. HTML 也是一种高级程序设计语言。

6. 无论是 TCP/IP 模型还是 OSI 模型，都可以视为网络的分层模型，每个网络协议都会被归入某一层中。如果用现实生活中的例子来比喻这些“层”，以下最恰当的是（　　）。

A. 中国公司的经理与波兰公司的经理交互商业文件



B. 军队发布命令



C. 国际会议中，每个人都与他国地位对等的人直接进行会谈

D. 体育比赛中，每一级比赛的优胜者晋级上一级比赛

7.（　　）是目前互联网上常用的 E-mail 服务协议。

A. HTTP　　B. FTP　　C. POP3　　D. Telnet

8. 蓝牙和 Wi-Fi 都是（　　）设备。

A. 无线广域网　　B. 无线城域网　　C. 无线局域网　　D. 无线路由器

9. IPv4 协议使用 32 位地址，随着其不断被分配，地址资源日趋枯竭。因此，它正逐渐被使用（　　）位地址的 IPv6 协议所取代。

A. 40　　B. 48　　C. 64　　D. 128

10. 中国的国家顶级域名是（　　）。

A. .cn　　B. .ch　　C. .chn　　D. .china

11. 以下哪一种是属于电子邮件收发的协议(　　)。

A.SMTP　　B.UDP　　C.P2P　　D.FTP

12. 下列几个 32 位 IP 地址中，书写错误的是(　　)。

A.162.105.135.27　　B.192.168.0.1

C.256.256.129.1　　D.10.0.0.1

13. 计算机病毒是（　　）。

A. 通过计算机传播的危害人体健康的一种病毒

B. 人为制造的能够侵入计算机系统并给计算机带来故障的程序或指令集合

C. 一种由于计算机元器件老化而产生的对生态环境有害的物质

D. 利用计算机的海量高速运算能力而研制出来的用于疾病预防的新型病毒

14. FTP 可以用于（　　）　。

A.远程传输文件　　B.发送电子邮件　　C.浏览网页　　D.网上聊天

15.以下不属于无线通信技术的是（　　）。

A.蓝牙　　B.WiFi　　C.GPRS　　D.以太网

16. 在计算机中，防火墙的作用是（　　）。

A. 防止火灾蔓延　　B.防止网络攻击

C. 防止计算机死机　　D. 防止使用者误删除数据

1.6 程序设计

1. 二分查找

```
int Binary_Search(int a[],int n,int key)
{
    int low,high,mid;
    low=1;        /*定义最底下标为记录首位*/
    high=n;       /*定义最高下标为记录末位*/
    while(low<=high)
    {
        mid=(low+high)/2;       /*折半*/
        if(key<a[mid])
            high=mid-1;
        if(key>a[mid])
            low=mid+1;
        else
            return mid;
    }
    return 0;
}
```

```
int l=1,r=n;
while(l<=r){
    int mid=(l+r)/2;
    if(check(mid)){
        ans=mid;
        r=mid-1;
    }
    else
        l=mid+1;
}
```

2. 插入排序

```
#include<bits/stdc++.h>
using namespace std;
int main(){
  int a[10],b[10],n=10;
  int i,j,k,pos;
  for(i=0;i<n;i++) cin>>a[i];
  for(i=0;i<n;i++){
    for(j=0;j<=i-1;j++)
      if(b[j]>a[i])break;
    pos=j;
    for(k=n-1;k>=pos;k--)
      b[k+1]=b[k];
    b[pos]=a[i];
  }
  for(i=0;i<n;i++) cout<<b[i]<<" ";
  return 0;
```

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int t,a[100],b[100],i,j,k;
    int n=10,pos;
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n;i++){
        for(j=0;j<i;j++)
            if(b[j]>a[i])break;
        pos=j;
        for(k=i;k>=pos;k--)
            b[k+1]=b[k];
        b[pos]=a[i];
    }
```

```
}
```

```
    for(i=0;i<n;i++)
        cout<<b[i]<<" ";
return 0;
}
```

3. 冒泡排序

```
#include<iostream>
using namespace std;
int main()
{
   int n=10,i,j,k,a[10];
   for(i=0;i<n;i++)    cin>>a[i];
   for(i=0;i<n-1;i++){
      for (j=0;j<n-i-1;j++)
         if(a[j]>a[j+1])    swap(a[j],a[j+1]);
   }
   for(i=0;i<n;i++)cout<<a[i]<<" ";
}
```

4. 选择排序

```
#include<bits/stdc++.h>
using namespace std;
int a[10];
void SelectSort(int a[],int n,int i)
{     int j,k;
      if (i==n-1) return;
      else
      {
            k=i;
            for (j=i+1;j<n;j++)
                  if (a[j]>a[k])k=j;
            if (k!=i)
                  swap(a[i],a[k]);
            SelectSort(a,n,i+1);
      }
}

int main(){
      int n;
      cin>>n;
      for(int i=0;i<n;i++)cin>>a[i];
```

```
#include<iostream>
using namespace std;
int main(){
      int a[11] ,T;
      int i,k,j,n;
      cin>>n;
      for(i=0;i<n;i++)
            cin>>a[i];
      for(i=0;i<n-1;i++){
            k=i;
            for(j=i+1;j<n;j++){
                  if(a[j]>a[k])
                        k=j;
            }
            swap(a[i],a[k]);
      }
      for(i=0;i<n;i++)
            cout<<a[i]<<"    ";
      cout<<endl;
      return 0;
}
```

```
    SelectSort(a,n,0);
    for(int i=0;i<n;i++)cout<<a[i]<<"    ";
    return 0;
}
```

5. 快排

```
void QuickSort(int l, int r){
    int i=l,j=r,key=a[i];
    while(i<j){
        while(i<j&&a[j]>=key)j--;
        a[i]=a[j];
        while(i<j&&a[i]<=key)i++;
        a[j]=a[i];
    }
    a[i]=key;
    if(l<r){
        QuickSort(l,i-1);
        QuickSort(i+1,r);
    }
}
```

```
void quicksort(int l,int r){
    int i=l,j=r,mid=a[(l+r)/2];
    while(i<=j){
        while(a[i]<mid) i++;
        while(a[j]>mid) j--;
        if(i<=j){
            swap(a[i],a[j]);
            i++; j--;
        }
    }
    if(l<j) quicksort(l,j);
    if(i<r) quicksort(i,r);
}
```

6. 归并排序

```
void Merge(int A[], int low, int mid, int high)
{
    int i = low, j = mid+1, k = 0;
    while(i <= mid && j <= high)
    {//按从小到大存放到辅助数组 B[]中
        if(A[i] <= A[j])
            B[k++] = A[i++];
        else
            B[k++] = A[j++];
    }
    while(i <= mid) B[k++] = A[i++];
    while(j <= high) B[k++] = A[j++];
    for(i = low, k = 0; i <= high; i ++)
      A[i] = B[k++];
}
void MergeSort(int A[], int low, int high)
{
```

```
        if(low < high)
        {
            int mid = (low+high)/2;
            MergeSort(A, low, mid);
            MergeSort(A, mid+1, high);
            Merge(A, low, mid, high);
        }
}
```

排序方法	平均时间复杂度	最坏情况下时间复杂度	额外空间复杂度	稳定性
简单选择排序	$O(N^2)$	$O(N^2)$	$O(1)$	不稳定
直接插入排序	$O(N^2)$	$O(N^2)$	$O(1)$	稳定
冒泡排序	$O(N^2)$	$O(N^2)$	$O(1)$	稳定
希尔排序	$O(N^d)$ $(1<d<1.5)$	$O(N^2)$	$O(1)$	不稳定
堆排序	$O(N\log_2 N)$	$O(N\log_2 N)$	$O(1)$	不稳定
快速排序	$O(N\log_2 N)$	$O(N^2)$	$O(\log_2 N)$	不稳定
归并排序	$O(N\log_2 N)$	$O(N\log_2 N)$	$O(N)$	稳定
基数排序	$O(D(N+R))$	$O(D(N+R))$	$O(N+R)$	稳定

1.6 作业

1. 在下列关于计算机算法的说法中，不正确的是（　　）。

A. 一个正确的算法至少要有一个输入

B. 算法的改进，在很大程度上推动了计算机科学与技术的进步

C. 判断一个算法的好坏的主要标准是算法的时间复杂性与空间复杂性

D. 目前仍然存在许多涉及到国计民生的重大课题，还没有找到能够在计算机上实施的有效算法

2. 在下列各种排序算法中，不是以“比较”作为主要操作的算法是（　　）。

A. 选择排序　　B. 冒泡排序　　C. 插入排序　　D. 基数排序

3. 在编程时（使用任一种高级语言，不一定是 C++），如果需要从磁盘文件中输入一个很大的二维数组（例如 1000*1000 的 double 型数组），按行读（即外层循环是关于行的）与按列读（即外层循环是关于列的）相比，在输入效率上（　　）。

A. 没有区别　　B. 按行读的方式要高一些

C. 按列读的方式要高一些　　D. 取决于数组的存储方式。

4．将 5 个数的序列排序，不论原先的顺序如何，最少都可以通过（　　）次比较，完成从小到大的排序。

A．6　　B．7　　C．8　　D．9

5．近 20 年来，许多计算机专家都大力推崇递归算法，认为它是解决较复杂问题的强有力的工具。在下列关于递归算法的说法中，正确的是（　　）。

A．在 1977 年前后形成标准的计算机高级语言“FORTRAN77”禁止在程序使用递归，原因之一是该方法可能会占用更多的内存空间

B．和非递归算法相比，解决同一个问题，递归算法一般运行得更快一些

C．对于较复杂的问题，用递归方式编程一般比非递归方式更难一些

D．对于已经定义好的标准数学函数 sin(x)，应用程序中的语句“y=sin(sin(x));”就是一种递归调用

6．一个无法靠自身的控制终止的循环成为“死循环”，例如，在 C 语言程序中，语句“while(1) printf(“*”);”就是一个死循环，运行时它将无休止地打印*号。下面关于死循环的说法中，只有（　　）是正确的。

A．不存在一种算法，对任何一个程序及相应的输入数据，都可以判断是否会出现死循环，因而，任何编译系统都不做死循环检查

B．有些编译系统可以检测出死循环

C．死循环属于语法错误，既然编译系统能检查各种语法错误，当然也应该能检查出死循环

D．死循环与多进程中出现的“死锁”差不多，而死锁是可以检测的，因而，死循环也可以检测的

7．地面上有标号为 A、B、C 的三根柱，在 A 柱上放有 10 个直径相同中间有孔的圆盘，从上到下依次编号为 1，2，3……，将 A 柱上的部分盘子经过 B 柱移入 C 柱，也可以在 B 柱上暂存。如果 B 柱上的操作记录为“进、进、出、进、进、出、出、进、进、出、进、出、出”。那么，在 C 柱上，从下到上的编号为（　　）。

A．2 4 3 6 5 7　　B．2 4 1 2 5 7

C．2 4 3 1 7 6　　D．2 4 3 6 7 5

8. 设字符串 S=”Olympic”，S 的非空子串的数目是（　　）。

A. 28　　B. 29　　C. 16　　D. 17

9．将数组{8, 23, 4, 16, 77, -5, 53, 100}中的元素按从大到小的顺序排列，每次可以交换任意两个元素，最少需要交换（　　）次。

A. 4　　B. 5　　C. 6　　D. 7

10.　对有序数组{5, 13, 19, 21, 37, 56, 64, 75, 88，92，100}进行二分查找，成功查找元素 19 的查找长度（比较次数）是（　　）。

A. 1　　B. 2　　C. 3　　D. 4

11、快速排序最坏情况下的算法时间复杂度为：（　　）

A.　O(log2n)　　B.　O(n)　　C.　O(nlog2n)　　D.　O(n2)

12. 有一个由 4000 个整数构成的顺序表，假定表中的元素已经按升序排列，采用二分查找定位一个元素。则最多需要几次比较就能确定是否存在所查找的元素：（　　）

A.　11 次　　B.　12 次　　C.　13 次　　D.　14 次

13、排序算法是稳定的意思是关键码相同的记录排序前后相对位置不发生改变，下列哪种排序算法是不稳定的：（　　）

A. 冒泡排序　　B. 插入排序　　C. 归并排序　　D. 快速排序

14. 设 X、Y、Z 分别代表三进制下的一位数字，若等式 XY + ZX = XYX 在三进制下成立，那么同样在三进制下，等式 XY * ZX = （　　）也成立。

A. YXZ　　B. ZXY　　C. XYZ　　D. XZY

15. 基于比较的排序时间复杂度的下限是（　　），其中 n 表示待排序的元素个数。

A. Θ(n)　　B. Θ(n log n)　　C. Θ(log n)　　D. Θ(n2)

16. 一个自然数在十进制下有 n 位，则它在二进制下的位数与（　　）最接近。

A. 5n　　B. $n*\log_2 10$　　C. $10*\log_2 n$　　D. $10n\log_2 n$

17. 完全二叉树的顺序存储方案，是指将完全二叉树的结点从上至下、从左至右依次存放到一个顺序结构的数组中。假定根结点存放在数组的 1 号位置，则第 k 号结点的父结点如果存在的话，应当存放在数组的（　　）号位置。

A. 2k　　B. 2k+1　　C. k/2 下取整　　D. (k+1)/2 下取整

18. 体育课的铃声响了，同学们都陆续地奔向操场，按老师的要求从高到矮站成一排。每个同学按顺序来到操场时，都从排尾走到排头，找到第一个比自己高的同学，并站在他的后面。这种站队的方法类似于（　　）算法。

A. 快速排序　　B. 插入排序　　C. 冒泡排序　　D. 归并排序

19. 使用冒泡排序对序列进行升序排列，每执行一次交换操作系统将会减少 1 个逆序对，因此序列 5，4，3，2，1 需要执行（　　）次操作，才能完成冒泡排序。

A. 0　　B. 5　　C. 10　　D. 15

20.（　　）就是把一个复杂的问题分成两个或更多的相同类似的子问题，再把子问题分解成更小的子问题……直到最后的子问题可以简单地直接求解。而原问题的解就是子问题解的并。

A. 动态规划　　B. 贪心　　C. 分治　　D. 搜索

21. 在程序运行过程中，如果递归调用的层数过多，会因为（　　）引发错误。

A. 系统分配的栈空间溢出　　B. 系统分配的堆空间溢出

C. 系统分配的队列空间溢出　　D. 系统分配的链表空间溢出

22. 原字符串中任意一段连续的字符所组成的新字符串称为子串。则字符“AAABBBCCC”共有（　　）个不同的非空子串。

A. 3　　B. 12　　C. 36　　D. 45

23. 下面的故事与（　　）算法有着异曲同工之妙。

从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：‘从前有座山，山里有座庙，庙里有个老和尚给小和尚讲故事....’

A. 枚举　　B. 递归　　C. 贪心　　D. 分治

24. 将（ 2, 6, 10, 17）分别存储到某个地址区间为 0~10 的哈希表中，如果哈希函数 h(x) =（　　），将不会产生冲突，其中 a mod b 表示 a 除以 b 的余数。

A. x mod 11　　B. x2 mod 11

C. 2x mod 11　　D. |√2| mod 11 ，其中 √X 表示 √X 下取整

25 （　　）的 平均时间复杂度为 O(n log n)，其中 n 是待排序的元素个数。

A. 快速排序　　B. 插入排序　　C. 冒泡排序　　D. 基数排序

26. 下面是根据欧几里得算法编写的函数，它所计算的是 a 和 b 的（　　）。

```
int euclid(int a, int b)
{
  if (b == 0)
  return a;
  else
  return euclid(b, a % b);
}
```

A. 最大公共质因子　B. 最小公共质因子

C. 最大公约数　　D. 最小公倍数

27. 通常在搜索引擎中，对某个关键词加上双引号表示（　　）。

A. 排除关键词，不显示任何包含该关键词的结果

B. 将关键词分解，在搜索结果中必须包含其中的一部分

C. 精确搜索，只显示包含整个关键词的结果

D. 站内搜索，只显示关键词所指向网站的内容

28. 把 64 位非零浮点数强制转换成 32 位浮点数后，不可能（　　）。

A. 大于原数　　B. 小于原数

C. 等于原数　　D. 与原数符号相反

29. 下列程序中，正确计算 1，2，…，100 这 100 个自然数之和 sum（初始值为 0）的是（　　）。

A.
```
i = 1;
do {
    sum += i;
    i++;
} while (i <= 100);
```

B.
```
i = 1;
do {
    sum += i;
    i++;
} while (i > 100);
```

C.
```
i = 1;
while (i < 100) {
    sum += i;
    i++;
}
```

D.
```
i = 1;
while (i >= 100) {
    sum += i;
    i++;
}
```

30.要求以下程序的功能是计算：s=1+1/2+1/3+...+1/10。

```
#include <iostream>
using namespace std;
int main() {
    int n;
    float s;
    s = 1.0;
```

```
    for(n = 10; n > 1; n--)
        s = s + 1 / n;
    cout << s << endl;
    return 0;
}
```

程序运行后输出结果错误，导致错误结果的程序行是(　　)。

A.s = 1.0;　　B.for(n = 10; n > 1; n--)

C.s = s + 1 / n;　　D.cout << s << endl;

31. 设变量 x 为 float 型且已赋值，则以下语句中能将 x 中的数值保留到小数点后两位，并将第三位四舍五入的是(　　)。

A.x = (x * 100) + 0.5 / 100.0;　　B.x = (x * 100 + 0.5) / 100.0;

C.x = (int)(x * 100 + 0.5)/100.0;　　D.x = (x / 100 + 0.5) * 100.0;

32. 有以下程序

```
#include <iostream>
using namespace std;
int main() {
    int s, a, n;
    s = 0;
    a = 1;
    cin >> n;
    do {
        s += 1;
        a -= 2;
    } while(a != n);
    cout << s << endl;
    return 0;
}
```

若要使程序的输出值为 2，则应该从键盘给 n 输入的值是(　　)。

A.-1　　B.-3　　C.-5　　D.0

33. 设有 100 个数据元素，采用折半搜索时，最大比较次数为(　　)。

A.6　　B.7　　C.8　　D.10

34. 若有如下程序段，其中 s、a、b、c 均已定义为整型变量，且 a、c 均已赋值，c>0。 s = a;
for(b = 1; b <= c; b++)　　s += 1;
则与上述程序段功能等价的赋值语句是(　　)。

A.s=a+b　　B.s=a+c　　C.s=s+c　　D.s=b+c

35.如果开始时计算机处于小写输入状态，现在有一只小老鼠反复按照 CapsLock、 字母键 A、字母键 S 和字母键 D 的顺序循环按键，即 CapsLock、A、S、D、CapsLock、A、S、D、……，屏幕上输出的第 81 个字符是字母（　　）。

A.A　　B. S　　C.D　　D.a

36. 以下关于字符串的判定语句中正确的是（　　）。

A.字符串是一种特殊的线性表　　B.串的长度必须大于零

C.字符串不可以用数组来表示　　D.空格字符组成的串就是空串

37.若有如下程序段，其中 s、a、b、c 均已定义为整型变量，且 a、c 均已赋值 （c 大于 0）。

```
s = a;
for (b = 1; b <= c; b++) s = s + 1;
```

则与上述程序段修改 s 值的功能等价的赋值语句是（　　）。

A. s = a + b;　　B. s = a + c;　　C. s = s + c;　　D. s = b + c;

38. 有以下程序：

```
#include <iostream>
using namespace std;
int main() {
    int k = 4, n = 0;
    while (n < k) {
        n++;
        if (n % 3 != 0)
            continue;
        k--;
    }
    cout << k << "," << n << endl;
    return 0;
}
```

程序运行后的输出结果是（　　）。

A. 2,2　　B. 2,3　　C. 3,2　　D. 3,3

39. 给定含有 n 个不同的数的数组 L=<x 1，x2，...，xn >。如果 L 中存在 x（i1 < i < n）使得 x1 < x2 < ... < xi -1 < xi > xi+1 > ... > xn ， 则称 L 是单峰的，并称 xi 是 L 的“峰顶”。现在已知 L 是单峰的，请把 a-c 三行代码补全到算法中使得算法 正确找到 L 的峰顶。

```
Search(k+1, n)
Search(1, k-1)
return L[k]
Search(1, n)
1. k<-[n/2]
2. if L[k] > L[k-1] and L[k] > L[k+1]
3. then __________
4. else if L[k] > L[k-1] and L[k] < L[k+1]
5. then __________
6. else __________
```

正确的填空顺序是（　　）。

A. c, a, b　　B. c, b, a　　C. a, b, c　　D. b, a, c

2 数据结构专题

“数据结构是计算机中存储、组织数据的方式。精心选择的数据结构可以带来最优效率的算法。

[例 1] 该如何摆放书，才能让读者很方便地找到你手里这本《数据结构》？

[方法 1] 随便放——任何时候有新书进来，哪里有空就把书插到哪里。

[方法 2] 按照书名的拼音字母顺序排放。

有时插入新书很困难！

[方法 3] 把书架划分成几块区域，每块区域指定摆放某种类别的图书；在每种类别内，按照书名的拼音字母顺序排放。

可能造成空间的浪费！

数据结构是计算机存储、组织数据的方式。通常情况下，精心选择的数据结构可以带来更高的运行或者存储效率。数据结构往往同高效的检索算法和索引技术有关。

逻辑结构：数据对象的逻辑组织关系。分为“线性”、“树”和“图”。例 1.1 中按方法 1 来处理，就是把图书集看成是线性的结构；按方法 3 来处理，就是把图书集看成是树形的结构。

物理结构：数据对象信息在计算机内存中的存储组织关系。一般分为“顺序存储”和“链式存储”。

一个算法是解决某一类问题的步骤的描述。一般而言，算法应该符合以下五项要求：

(1) 输入：它接受一些输入（有些情况下不需要输入）；

(2) 输出：至少产生一个输出；

(3) 确定性：算法的每一步必须有充分明确的含义，不可以有歧义；

(4) 有限性：算法是一个有限指令集，并一定在有限步骤之后终止；

(5) 可行性：算法的每一步必须在计算机能处理的范围之内。

2.1 复杂度与链表

2.1.1 复杂度

什么是“好”的算法？

具体衡量、比较算法优劣的指标主要有两个：

空间复杂度 S(n) ——根据算法写成的程序在执行时占用存储单元的长度。这个长度往往与输入数据的规模有关。空间复杂度过高的算法可能导致使用的内存超限，造成程序非正常中断。

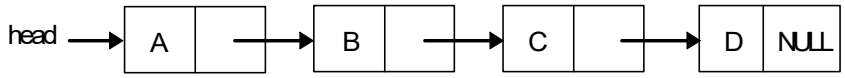
时间复杂度 T(n) ——根据算法写成的程序在执行时耗费时间的长度。这个长度往往也与输入数据的规模有关。时间复杂度过高的低效算法可能导致我们在有生之年都等不到运行结果。

问题规模 n	可用算法的时间复杂度					
	O(logn)	O(n)	O(nlogn)	$O(n^2)$	$O(2^n)$	O(n!)
n ≤ 11	√	√	√	√	√	√
n ≤ 25	√	√	√	√	√	×
n ≤ 5000	√	√	√	√	×	×
$n \le 10^6$	√	√	√	×	×	×
$n \le 10^7$	√	√	×	×	×	×
$n > 10^8$	√	×	×	×	×	×

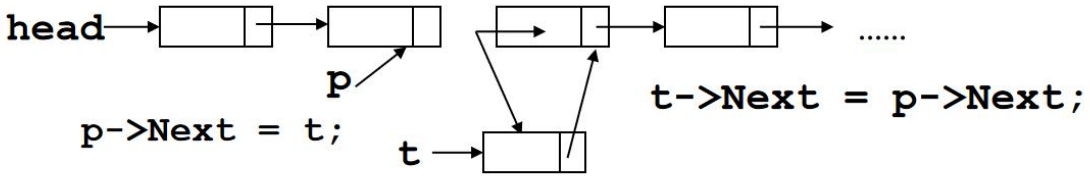
2.1.2 链表

链表是一种重要的基础数据结构，也是实现复杂数据结构的重要手段。它不按照线性的顺序存储数据，而是由若干个同一结构类型的“结点”依次串接而成的，即每一个结点里保存着下一个结点的地址（指针）。

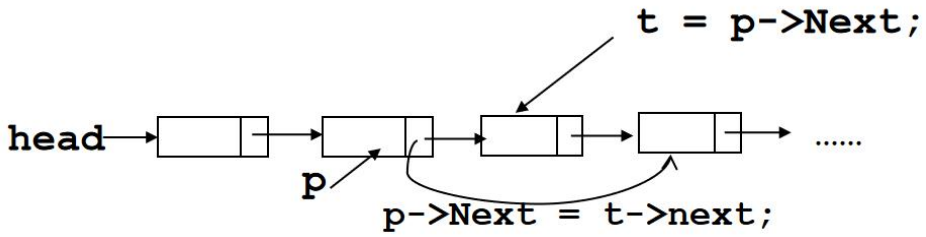
链表又分单向链表，双向链表以及循环链表等。



1） 插入结点（ p 之后插入新结点 t ）



2.2 单链表-删除结点



2.1 作业

1. 链表不具备的特点是（　　）

A.可随机访问任何一个元素　　B.插入、删除操作不需要移动元素

C.无需事先估计存储空间大小　D.所需存储空间与存储元素个数成正比

2. 在单链表中，若 p 所指的结点不是最后结点，在 p 之后插入 s 所指结点，则执行（　　）

A) s->next=p; p->next=s;　　B) s->next=p->next; p=s;

C) s->next=p->next; p->next=s;　　D) p->next=s; s->next=p;

3. 设 h 为不带头结点的单向链表。在 h 的头上插入一个新结点 t 的语句是：（　　）

A) h=t; t->next=h->next;　　B) t->next=h->next; h=t;

C) h=t; t->next=h;　　D) t->next=h; h=t;

4. 双向链表中有两个指针域 llink 和 rlink，分别指向该结点的前驱及后继。设 p 指向链表中的一个结点，它的左右结点均非空。现要求删除结点 p，则下面语句序列中错误的是（　　）。

A. p->rlink->llink = p->rlink;p->llink->rlink = p->llink; delete p;

B. p->llink->rlink = p->rlink;p->rlink->llink = p->llink; delete p;

C. p->rlink->llink = p->llink;p->rlink->llink->rlink = p->rlink; delete p;

D. p->llink->rlink = p->rlink;p->llink->rlink->llink = p->llink; delete p;

5. 在使用高级语言编写程序时，一般提到的“空间复杂度”中的“空间”是指（　　）。

A. 程序运行时理论上所占的内存空间

B. 程序运行时理论上所占的数组空间

C. 程序运行时理论上所占的硬盘空间

D. 程序源文件理论上所占的硬盘空间

6. 在含有 n 个元素的双向链表中查询是否存在关键字为 k 的元素，最快情况下运行的时间复杂度是（　　）。

A. O(1)　　B. O(log n)　　C. O(n)　　D. O(n log n)

7. 链表不具有的特点是(　　)。

A.不必事物估计存储空间　　B.可随机访问任一元素

C.插入删除不需要移动元素　　D.所需空间与线性表长度成正比

8. 线性表若采用链表存储结构，要求内存中可用存储单元地址（　　）

A.必须连续　B.部分地址必须连续　C. 一定不连续　D.连续不连续均可

9. 设某算法的计算时间表示为递推关系式 T(n)=T(n-1)+n(n 为正整数）及 T(0) = 1，则 该算法的时 间复杂度为（　　）。

A. O(logn)　B. O(n log n)　C. O(n)　D. $O(n^2)$

2.2 栈与队列

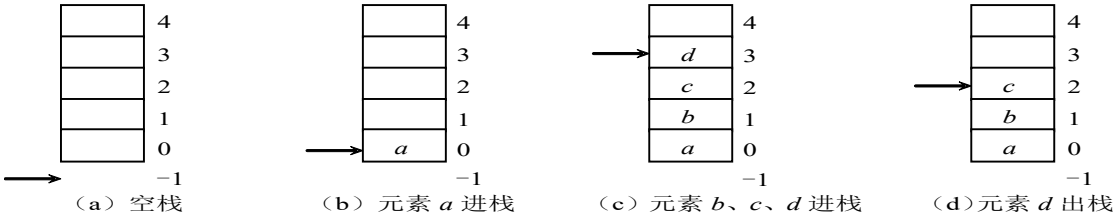
2.2.1 栈

栈是一种只能在一端进行插入或删除操作的线性表。表中允许进行插入、删除操作的一端称为栈顶。栈顶的当前位置是动态的，由一个称为栈顶指针的位置指示器来指示。表的另一端称为栈底。当栈中没有数据元素时，称为空栈。

栈的插入操作通常称为进栈或入栈，栈的删除操作通常称为退栈或出栈。

栈的主要特点是“后进先出”，即后进栈的元素先弹出。每次进栈的数据元素都放在原当前栈顶元素之前成为新的栈顶元素，每次出栈的数据元素都是原当前栈顶元素。栈也称为后进先出表。

下图是一个栈的动态示意图，图中箭头表示当前栈顶元素位置。图（a）表示一个空栈；图（b）表示数据元素 a 进栈以后的状态；图（c）表示数据元素 b、c、d 进栈以后的状态；图（d）表示出栈一个数据元素以后的状态。

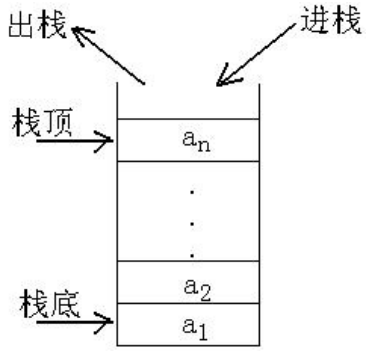


（a）空栈　（b）元素 a 进栈　（c）元素 b、c、d 进栈　（d）元素 d 出栈

[例 1] 若元素进栈顺序为 1234，能否得到 3142 的出栈顺序？

解：为了要让 3 作为第一个出栈元素，1、2 先进栈，此时要么 2 出栈，要么 4 进栈后出栈，出栈的第 2 个元素不可能是 1。所以得不到 3142 的出栈顺序。

1 栈的操作



栈可以采用顺序存储结构，分配一块连续的存储空间 data（大小为常量 MaxSize）来存放栈中元素，并用一个变量 top（栈顶指针）指向当前的栈顶以反映栈中元素的变化，采用顺序存储的栈称为顺序栈。

栈的存储方式和实现也有数组模拟和链表模拟两种，本讲义只介绍数组实现。

a 栈的定义

```
#define MaxSize 100005
 struct Stack{
   int Data[MaxSize];
   int Top;
};
```

b 栈的初始化

```
void init(Stack &x){
   X->Top=-1;//也可以写成 x->top=-1;
}
```

c 判空

```
bool empty(Stack &x){
        if(x->Top==-1)return 1;
        return 0;
    }
```

D 取栈顶元素

```
int top(Stack &x){
    return x->Data[x->Top];
}
```

e 出栈

①若 TOP≤0，则给出下溢信息，作出错处理(退栈前先检查是否已为空栈， 空则下溢；不空则作②)；

②X=S[TOP]，（退栈后的元素赋给 X）；

③TOP--，结束（栈指针减 1，指向栈顶）。

```
void pop(Stack &x){
        x->Top--;
    }
```

f 进栈（压栈）

①若 TOP≥n 时，则给出溢出信息，作出错处理（进栈前首先检查栈是否已满，满则溢出；不满则作②）；

②TOP++（栈指针加 1，指向进栈地址）；

③S[TOP]=X，结束（X 为新进栈的元素）；

```
void push(int a,Stack &x){
    x->Top++;
    x->Data[Top]=a;
}
```

2 STL 中的栈

STL 是“Standard Template Library”的缩写，中文译为“标准模板库”。STL 是 C++ 标准库的一部分，不用单独安装。C++ 对模板（Template）支持得很好，STL 就是借助模板把常用的数据结构及其算法都实现了一遍，并且做到了数据结构和算法的分离。例如，vector 的底层为顺序表（数组），list 的底层为双向链表，deque 的底层为循环队列，set 的底层为红黑树，hash_set 的底层为哈希表。

包含头文件：#include <stack>
使用命名空间：using namespace std;
定义栈的方法：

```
stack<char> S1; //栈中的结点为字符
stack<int> S2;  //栈中的结点为整型数据
stack<pos> S3; //栈中的结点为自定义结构体 pos 变量
```

stack 的成员函数：
push：压栈，参数为需要压入栈的结点；
pop：出栈，返回值为出栈的结点；
top：取得栈顶结点，返回值为栈顶结点，该操作并不会弹出栈顶结点；
empty：判断栈是否为空，返回值为 bool 型。
size：计算栈中结点的个数。

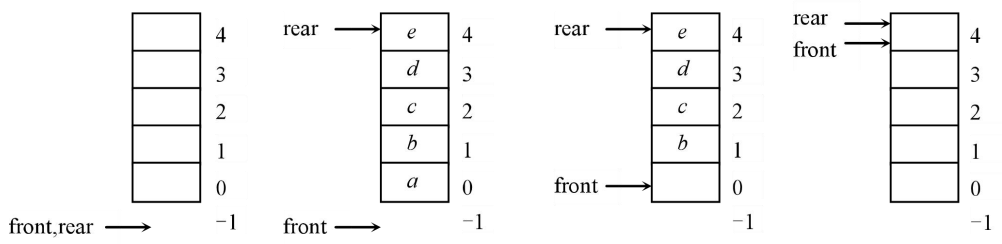
```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    stack<int> s1;//1 默认创建： stack<int> s1;
    stack<int> s2(s1);//拷贝创建： stack<int> s2(ano);
    for(int i = 0; i < 10; i++)
        s1.push(i);// 2 入栈： void push(DT data);
    cout << s1.top() << endl;//3 访问但不弹出栈顶元素：DT top();
    s1.pop();//4 弹出但不访问栈顶元素：void pop();
    cout << s1.top() << endl;
    cout << s1.empty() << " " << s2.empty() << endl;
         // 5 判断栈是否为空：bool empty();
    cout << s1.size() << endl;// 6、获得栈中元素的个数：int size();
    return 0;
}
```

2.2.2 队列

队列（简称为队）是一种操作受限的线性表，其限制为仅允许在表的一端进行插入，而在表的另一端进行删除。把进行插入的一端称做队尾（rear），进行删除的一端称做队头或队首（front）。向队列中插入新元素称为进队或入队，新元素进队后就成为新的队尾元素；从队列中删除元素称为出队或离队，元素出队后，其直接后继元素就成为队首元素。队列的插入和删除操作分别是在表的各自的一端进行的，元素进队只能从队尾进，不能从队头或中间位置进队，如下图所示。每个元素必然按照进入的次序出队，所以又把队列称为先进先出表。

下图是所示是一个队列的动态示意图，图中 front 指针指向队首位置（实际上是队首元素的前一个位置），rear 指针指向队尾位置（正好是队尾元素的位置）。图（a）表示一个空队；图（b）表示插入 5 个数据元素后的状态；图（c）表示出队一次后的状态；图（d）表示再出队 4 次后的状态。

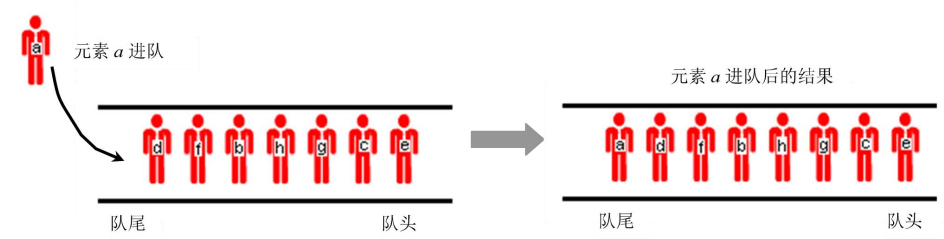


（a)空队　（b）*a,b,c,d,e* 元素进队　（c)出队一次　（d）出队 4

[例 3.6]

若元素进例 1 队顺序为 1234，能否得到 3142 的出队顺序？

解：进队顺序为 1234，则出队的顺序也为 1234（先进先出），所以不能得到 3142 的出队顺序。



1 队列的操作

队列可以采用顺序存储结构，分配一块连续的存储空间 data（大小为常量 MaxSize）来存放队列中元素，并用变量 front（队头指针）指向队头元素，用变量 rear（队尾指针）指向队头元素。队列的存储方式和实现和实现也有数组模拟和链表模拟两种，本讲义只介绍数组实现。

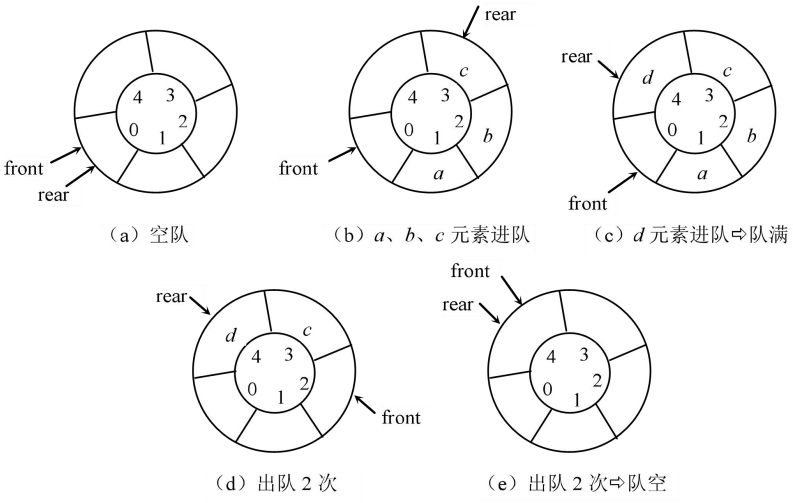
1 队列的定义
`#define MaxSize 100005`
`struct Stack{`
`int Data[MaxSize];`
`int front,rear;`
`};`
 	
2 队列的初始化
初始时置 front=rear=-1；
	3 判空
	若队列满足 front==rear 条件，则返回 true；否则返回 false。对应的算法如下：
	`bool QueueEmpty()`
	`{`
	`return(front==rear);`
	`}`

3. 出队操作:元素出队只能从队头出，不能从队头或中间位置出队
`bool deQueue( string e)`
`{`
`if (front==rear)//队空下溢出`
`return false;`
	4 入队操作
	当队列不满的条件下,先将队尾指针 rear 增 1，然后元素 e 放到该位置处。对应的算法如下：
	`bool enQueue(string e)`
	`{ if (rear==MaxSize-1)//队满上溢出`
	`return false;`

front++;
e=data[front];
return true;
}
	rear++;
	data[rear]=e;
	return true;
	}

2.3.2 循环队列

在非循环队列中，元素进队时队尾指针 rear 增 1，元素出队时队头指针 front 增 1，当进队 MaxSize 个元素后，满足队满的条件即 rear==MaxSize-1 成立，此时即使出队若干元素，队满条件仍成立（实际上队列中有空位置），这是一种假溢出。为了能够充分地使用数组中的存储空间，把数组的前端和后端连接起来，形成一个循环的顺序表，即把存储队列元素的表从逻辑上看成一个环，称为循环队列。



循环队列首尾相连，当队首指针 front=MaxSize-1 后，再前进一个位置就自动到 0，这可以利用求余的运算(%)来实现：

队首指针进 1：front=(front+1)%MaxSize

队尾指针进 1：rear=(rear+1)%MaxSize

循环队列的队头指针和队尾指针初始化时都置 0：front=rear=0。在进队元素和出队元素时，队头和队尾指针都循环前进一个位置。

那么，循环队列的队满和队空的判断条件是什么呢?显然循环队列为空条件是 rear==front。如果进队元素的速度快于出队元素的速度，队尾指针很快就赶上了队首指针，此时可以看出循环队列的队满条件也为 rear==front。

怎样区分这两者之间的差别呢?通常约定在进队时少用一个数据元素空间，以队尾指针加 1 等于队首指针作为队满的条件，即队满条件为：(rear+1) % MaxSize==front。队空条件仍为 rear==front。

下图所示说明了循环队列的几种状态，这里假设 MaxSize 等于 5。图（a）为空队，此时 front=rear=0；图（b）中有 3 个元素，当进队元素 d 后，队中有 4 个元素，此时满足队满的条件。

2.3.3 STL 中的队列

包含头文件：#include <queue>
使用命名空间：using namespace std;
定义栈的方法：
queue<char> S1; //队列中的结点为字符
queue<int> S2; //队列中的结点为整型数据
queue<pos> S3; //队列中的结点为自定义结构体 pos 变量
queue 的成员函数：
back()返回最后一个元素
empty()如果队列空则返回真
front()返回第一个元素
pop()删除第一个元素
push()在末尾加入一个元素
size()返回队列中元素的个数

3.queue 的基本操作举例如下：
queue 入队，如例：q.push(x); 将 x 接到队列的末端。
queue 出队，如例：q.pop(); 弹出队列的第一个元素，注意，并不会返回被弹出元素的值。
访问 queue 队首元素，如例：q.front()，即最早被压入队列的元素。
访问 queue 队尾元素，如例：q.back()，即最后被压入队列的元素。
判断 queue 队列空，如例：q.empty()，当队列空时，返回 true。
访问队列中的元素个数，如例：q.size()

作业

1. 某个车站呈狭长形，宽度只能容下一台车，并且只有一个出入口。已知某时刻该车站状态为空，从这一时刻开始的出入记录为：“进，出，进，进，出，进，进，进，出，出，进，出”。假设车辆入站的顺序为 1，2，3，……，则车辆出站的顺序为（ ）。

A. 1, 2, 3, 4, 5　　B. 1, 2, 4, 5, 7
C. 1, 3, 5, 4, 6　　D. 1, 3, 6, 5, 7

2. 设栈 S 的初始状态为空，元素 a, b, c, d, e, f, g 依次入栈，以下出栈序列不可能出现的是（ ）。

A. a, b, c, e, d, f, g　　B. b, c, a, f, e, g, d
C. a, e, d, c, b, f, g　　D. g, e, f, d, c, b, a

3. 某个车站呈狭长形，宽度只能容下一台车，并且只有一个出入口。已知某时刻该车站状态为空，从这一时刻开始的出入记录为：“进，出，进，进，进，出，出，进，进，进，出,出”。假设车辆入站的顺序为 1，2，3，……，则车辆出站的顺序为（ ）。

A. 1, 2, 3, 4, 5　　B. 1, 2, 4, 5, 7
C. 1, 4, 3, 7, 6　　D. 1, 4, 3, 7, 2

4. 设栈 S 的初始状态为空，元素 a, b, c, d, e 依次入栈，以下出栈序列不可能出现的有（　　）。

A. a, b, c, e, d　　B. b, c, a, e, d

C. a, e, c, b, d　　D. d, c, e, b, a

5. 设栈 S 的初始状态为空，元素 a，b，c，d，e，f 依次入栈 S，出栈的序列为 b，d，f，e，c，a，则栈 S 的容量至少应该是（　　）。

A. 6　　B. 5　　C. 4　　D. 3

6. 递归过程或函数调用时，处理参数和返回地址，通常使用一种称为（　　）的数据结构。

A. 队列　　B. 多维数组　　C. 线性表　　D. 栈

7.有六个元素 FEDCBA 从左至右依次顺序进栈，在进栈过程中会有元素被弹出栈。问下列哪一个不可能是合法的出栈序列？（　　）

A. EDCFAB　　B. DECABF　　C. CDFEBA　　D. BCDAEF

8. 前缀表达式“+ 3 * 2 + 5　12”的值是（　　）。

A. 23　　B. 25　　C. 37　　D. 65

9. 元素 R1、R2、R3、R4、R5 入栈的顺序为 R1、R2、R3、R4、R5。如果第 1 个出栈的是 R3，那么第 5 个出栈的不可能是（　　）。

A. R1　　B. R2　　C. R4　　D. R5

10. 广度优先搜索时，需要用到的数据结构是（　　）。

A. 链表　　B. 队列　　C. 栈　　D. 散列表

11.（　　）是一种选优搜索法，按选优条件向前搜索，以达到目标。当搜索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择。:

A. 回溯法　　B. 枚举法　　C. 动态规划　　D. 贪心

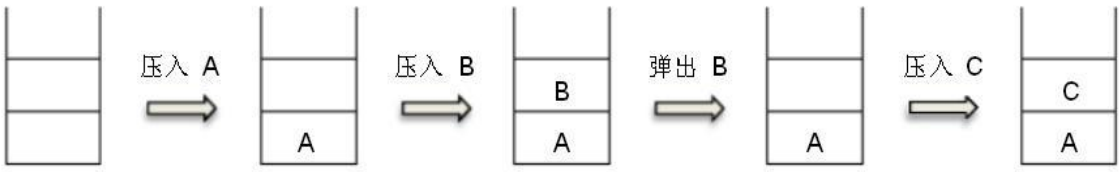
12.（　　）是一种先进先出的线性表。

A. 栈　　B. 队列　　C. 哈希表（散列表）　　D. 二叉树

13. 如果一个栈初始时为空，且当前栈中的元素从栈顶到栈底依次为 a，b，c，另有元素 d 已经出栈，则可能的入栈顺序是（　　）。

A. a, d, c, b　　B. b, a, c, d　　C. a, c, b, d　　D. d, a, b, c

14. 下图中所使用的数据结构是（　　）。



A. 哈希表　　B. 栈　　C. 队列　　D. 二叉树

15. 今有一空栈 S，对下列待进栈的数据元素序列 a,b,c,d,e,f 依次进行进栈，进栈，出栈，进栈，进 栈，出栈的操作，则此操作完成后，栈 S 的栈顶元素为（　　）。

A. f　　B. c　　C. a　　D. b

2.3 二叉树的概述

树是由 n（n≥0）个结点组成的有限集合（记为 T）。如果 n=0，它是一棵空树，这是树的特例；如果 n>0，这 n 个结点中存在（有仅存在）一个结点作为树的根结点（root），其余结点可分为 m（m≥0）个互不相交的有限集 T1、T2、…、Tm，其中每个子集本身又是一棵符合本定义的树，称为根结点的子树。

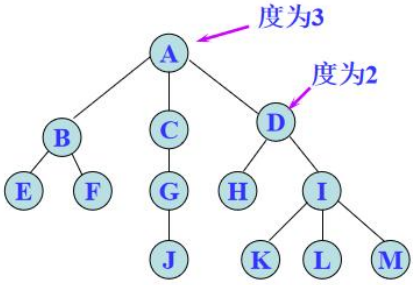
树是一种非线性数据结构，具有以下特点：

它的每一结点可以有零个或多个后继结点，但有且只有一个前趋结点（根结点除外）；这些数据结点按分支关系组织起来，清晰地反映了数据元素之间的层次关系。可以看出，数据元素之间存在的关系是一对多的关系。

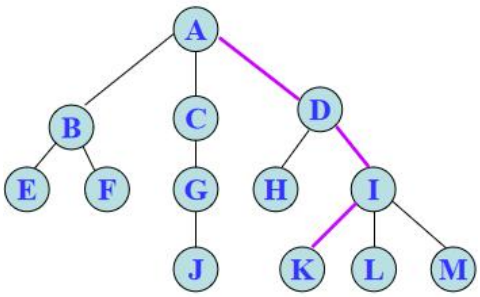
2.3.1 树的基本术语

1. 结点的度与树的度：树中某个结点的子树的个数称为该结点的度。树中各结点的度的最大值称为树的度，通常将度为 m 的树称为 m 次树。

2. 分支结点与叶结点：度不为零的结点称为非终端结点,又叫分支结点。度为零的结点称为终端结点或叶结点。在分支结点中，每个结点的分支数就是该结点的度。如对于度为 1 的结点,其分支数为 1，被称为单分支结点；对于度为 2 的结点，其分支数为 2，被称为双分支结点，其余类推。



3. 路径与路径长度：对于任意两个结点 di 和 dj，若树中存在一个结点序列 di,di1,di2,…,din,dj，使得序列中除 di 外的任一结点都是其在序列中的前一个结点的后继，则称该结点序列为由 di 到 dj 的一条路径，用路径所通过的结点序列(di,di1,di2,…,dj)表示这条路径。



A到K的路径为A,D,I,K，

其长度为3

路径长度等于路径所通过的结点数目减 1（即路径上分支数目）。

4. 孩子结点、双亲结点和兄弟结点：在一棵树中，每个结点的后继，被称作该结点的孩子结点（或子女结点）。相应地，该结点被称作孩子结点的双亲结点（或父母结点）。

具有同一双亲的孩子结点互为兄弟结点。进一步推广这些关系，可以把每个结点的所有子树中的结点称为该结点的子孙结点。

从树根结点到达该结点的路径上经过的所有结点被称作该结点的祖先结点。

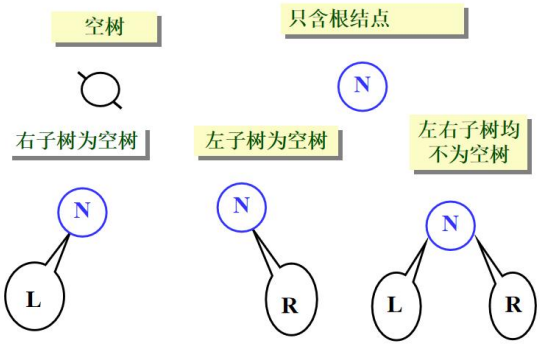
5. 结点的层次和树的高度：树中的每个结点都处在一定的层次上。结点的层次从树根开始定义，根结点为第 1 层，它的孩子结点为第 2 层,以此类推,一个结点所在的层次为其双亲结点所在的层次加 1。树中结点的最大层次称为树的高度（或树的深度）。

6. 有序树和无序树：若树中各结点的子树是按照一定的次序从左向右安排的，且相对次序是不能随意变换的，则称为有序树，否则称为无序树。

7. 森林：n（n>0）个互不相交的树的集合称为森林。森林的概念与树的概念十分相近，因为只要把树的根结点删去就成了森林。反之，只要给 n 棵独立的树加上一个结点，并把这 n 棵树作为该结点的子树，则森林就变成了树。

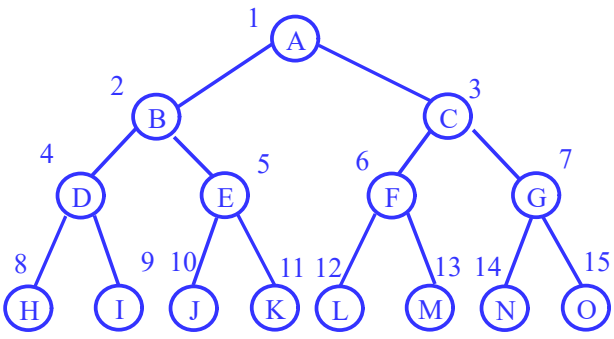
2.3.2 二叉树的概念

二叉树是有限的结点集合。这个集合或者是空。 或者由一个根结点和两棵互不相交的称为左子树和右子树的二叉树组成。二叉树有五种基本形态：

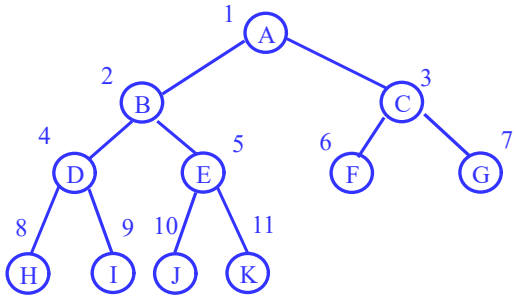


在一棵二叉树中，如果所有分支结点都有左孩子结点和右孩子结点，并且叶结点都集中在二叉树的最下一层，这样的二叉树称为满二叉树。下图所示就是一棵满二叉树。可以对满二叉树的结点进行连续编号，约定编号从树根为 1 开始，按照层数从小到大、同一层从左到右的次序进行。图中每个结点外边的数字为对该结点的编号。

若二叉树中最多只有最下面两层的结点的度数可以小于 2，并且最下面一层的叶结点都依次排列在该层最左边的位置上，则这样的二叉树称为完全二叉树。如下图所示为一棵完全二叉树。同样可以对完全二叉树中每个结点进行连续编号，编号的方法同满二叉树相同。图中每个结点外边的数字为对该结点的编号。
若二叉树中最多只有最下面两层的结点的度数可以小于 2，并且最下面一层的叶结点都依次排列在该层最左边的位置上，则这样的二叉树称为完全二叉树。



满二叉树



完全二叉树

如下图所示为一棵完全二叉树。同样可以对完全二叉树中每个结点进行连续编号，编号的方法同满二叉树相同。图中每个结点外边的数字为对该结点的编号。若二叉树中最多只有最下面两层的结点的度数可以小于 2，并且最下面一层的叶结点都依次排列在该层最左边的位置上，则这样的二叉树称为完全二叉树。

2.3.3 性质

性质 1 非空二叉树上叶结点数等于双分支结点数加 1。

证明：设二叉树上叶结点数为 n0，单分支结点数为 n1，双分支结点数为 n2，则总结点数 n=n0+n1+n2。在一棵二叉树中，所有结点的分支数（即度数）应等于单分支结点数加上双分支结点数的 2 倍，即总的分支数=n1+2n2。

由于二叉树中除根结点以外，每个结点都有唯一的一个分支指向它，因此二叉树中有：总的分支数=总结点数-1。由上述三个等式可得：n1+2n2=n0+n1+n2-1

即：n0=n2+1

性质 2 非空二叉树上第 i 层上至多有 2i-1 个结点，这里应有 i≥1。

性质 3 高度为 h 的二叉树至多有 2h-1 个结点（h≥1）。

性质 4 对完全二叉树中编号为 i 的结点（1≤i≤n，n≥1，n 为结点数）有：

（1）若 i≤ n/2 ，即 2i≤n，则编号为 i 的结点为分支结点，否则为叶子结点。

（2）若 n 为奇数，则每个分支结点都既有左孩子结点，也有右孩子结点；若 n 为偶数，则编号最大的分支结点只有左孩子结点，没有右孩子结点，其余分支结点都有左、右孩子结点。

（3）若编号为 i 的结点有左孩子结点，则左孩子结点的编号为 2i；若编号为 i 的结点有右孩子结点，则右孩子结点的编号为(2i+1)。

（4）除树根结点外,若一个结点的编号为 i，则它的双亲结点的编号为 i/2 ，也就是说，当 i 为偶数时，其双亲结点的编号为 i/2,它是双亲结点的左孩子结点，当 i 为奇数时，其双亲结点的编号为(i-1)/2，它是双亲结点的右孩子结点。

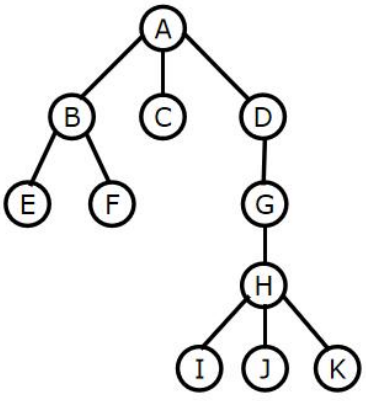
2.4.4 树的基本操作

树的运算主要分为三大类：

第一类，寻找满足某种特定关系的结点，如寻找当前结点的双亲结点等；

第二类，插入或删除某个结点，如在树的当前结点上插入一个新结点或删除当前结点的第 i 个孩子结点等；

第三类，遍历树中每个结点。



1 树的遍历

树的遍历运算是指按某种方式访问树中的每一个结点且每一个结点只被访问一次。

有以下三种遍历方法：

先根遍历：若树不空，则先访问根结点，然后依次先根遍历各棵子树。

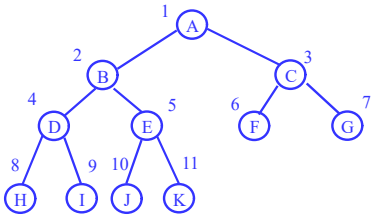
后根遍历： 若树不空，则先依次后根遍历各棵子树，然后访问根结点。

层次遍历：若树不空，则自上而下自左至右访问树中每个结点。

如图所示，遍历的顺序为：

先根遍历的顶点访问次序：A B E F C D G H I J K

后根遍历的顶点访问次序：E F B C I J K H G D A
层次遍历的顶点访问次序：A B C D E F G H I J K



完全二叉树

2 树的建立

1. 二叉树的顺序存储结构

二叉树的顺序存储结构中结点的存放次序是：对该树中每个结点进行编号，其编号从小到大的顺序就是结点存放在连续存储单元的先后次序。

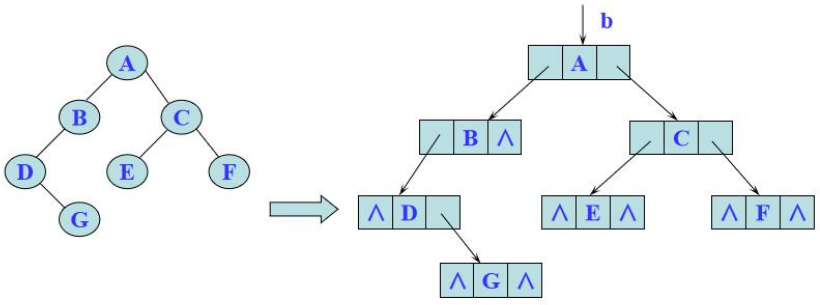
若把二叉树存储到一维数组中,则该编号就是下标值加 1（注意 C/C++语言中数组的起始下标为 0）。树中各结点的编号与等高度的完全二叉树中对应位置上结点的编号相同。

2. 二叉树的链式存储结构

在二叉树的链接存储中，结点的结构如下:

```
struct BTNode        //二叉链中结点类型
{   int data;            //数据元素
    BTNode *lchild;      //指向左孩子结点
    BTNode *rchild;      //指向右孩子结点
};
```

其中,data 表示值域,用于存储对应的数据元素，lchild 和 rchild 分别表示左指针域和右指针域，用于分别存储左孩子结点和右孩子结点（即左、右子树的根结点）的存储位置。



二叉树及其链式存储结构：二叉链

作业

1. 二叉树 T，已知其前序遍历序列为 1 2 4 3 5 7 6，中序遍历序列为 4 2 1 5 7 3 6，则其后序遍历序列为（　　）。

A. 4 2 5 7 6 3 1　　B. 4 2 7 5 6 3 1

C. 4 2 7 5 3 6 1　　D. 4 7 2 3 5 6 1

2. 满二叉树的叶结点个数为 N，则它的结点总数为（　　）。

A. N　　B. 2 * N　　C. 2 * N - 1　　D. 2 * N + 1

3. 完全二叉树的结点个数为11，则它的叶结点个数为（　　）。

A. 4　　B.3　　C.5　　D.　6

4. 二叉树 T的宽度优先遍历序列为A B C D E F G H I，已知A是C的父结点，D 是G 的父结点，F 是I 的父结点，树中所有结点的最大深度为3（根结点深度设为0），可知F的父结点是（　　）。

A. 无法确定　　B. B　　C. C　　D. D

5. 高度为n的均衡的二叉树是指：如果去掉叶结点及相应的树枝，它应该是高度为n-1的满二叉树。在这里，树高等于叶结点的最大深度，根结点的深度为0，如果某个均衡的二叉树共有2381个结点，则该树的树高为（　　）。

A.10　　B.11　　C.12　　D.13

6. 已知6个结点的二叉树的先根遍历是1 2 3 4 5 6（数字为结点的编号，以下同），后根遍历是3 2 5 6 4 1，则该二叉树的可能的中根遍历是（　　）

A. 3 2 1 4 6 5　　B. 3 2 1 5 4 6

C. 2 1 3 5 4 6　　D. 2 3 1 4 6 5

7. 已知7个节点的二叉树的先根遍历是1 2 4 5 6 3 7（数字为节点的编号，以下同），中根遍历是4 2 6 5 1 7 3，则该二叉树的后根遍历是（　　）。

A. 4 6 5 2 7 3 1　　B. 4 6 5 2 1 3 7

C. 4 2 3 1 5 4 7　　D. 4 6 5 3 1 7 2

8. 完全二叉树共有2*N-1个结点，则它的叶节点数是（　　）。

A. N-1　　B. N　　C. 2*N　　D. 2N-1

9. 二叉树T，已知其先根遍历是1 2 4 3 5 7 6（数字为结点的编号，以下同），中根遍历是2 4 1 5 7 3 6，则该二叉树的后根遍历是（　　）。

A. 4 2 5 7 6 3 1　　B. 4 2 7 5 6 3 1

C. 7 4 2 5 6 3 1　　D. 4 2 7 6 5 3 1

10. 设T是一棵有n个顶点的树，下列说法不正确的是（　　）。

A. T有n条边　　B. T是连通的

C. T是无环的　　D. T有n-1条边

11. 表达式a*(b+c)-d的后缀表达式是：（　　）

A.　abcd*+-　　B.　abc+*d-　　C.　abc*+d-　　D.　-+*abcd

12.一个包含n个分支结点（非叶结点）的非空二叉树，它的叶结点数目最多为：（　　）

A.　2n + 1　　B.　2n-1　　C.　n-1　　D.　n+1

13. 如果树根算第1层，那么一棵n层的二叉树最多有（　　）个结点。

A. 2n-1　　B. 2n　　C. 2n+1　　D. 2n+1

14. 一棵二叉树的前序遍历序列是ABCDEFG，后序遍历序列是CBFEGDA，则根结点的左子树的结点个数可能是（　　）。

A. 2　　B. 3　　C. 4　　D. 5

15. 如果根结点的深度记为1，则一棵恰有2011个叶结点的二叉树的深度最少是（　　）。

A. 10　　B. 11　　C. 12　　D. 13

16. 现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“呼”、“者”、“也”组成，它们出现的次数分别为700、600、300、200。那么，

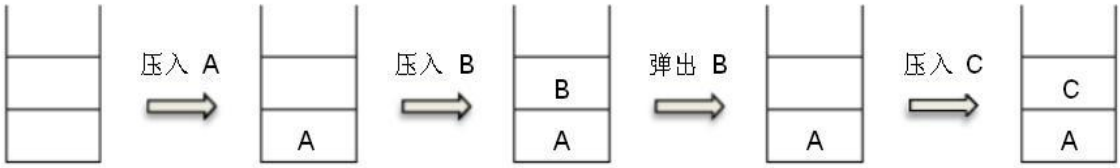
“也”字的编码长度是（　　）。

A．1　　B．2　　C．3　　D．4

17．如果一棵二叉树的中序遍历是 BAC，那么它的先序遍历不可能是（　　）。

A．ABC　　B．CBA　　C．ACB　　D．BAC

18．下图中所使用的数据结构是（　　）。

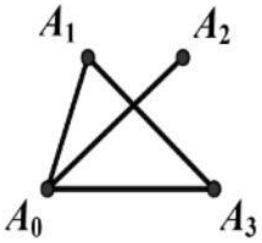


A．　哈希表　　B．　栈　　C．队列　　D．　二叉树

19．二叉树的（　　）第一个访问的节点是根节点。

A．　先序遍历　　B．　中序遍历　　C．　后序遍历　　D．　以上都是

20．以 A0 作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（　　）。



A．　A0，A1 ，A2，A3　　B．　A0，A1，A3，A2

B．　C．　A0，A2，A1，A3　　D．　A0，A3，A1，A2

21.一棵具有 5 层的满二叉树中结点数为(　　)。

A.31　　B.32　　C.33　　D.16

22．前序遍历序列与中序遍历序列相同的二叉树为（　　）。

A．　根结点无左子树的二叉树

B．　根结点无右子树的二叉树

C．　只有根结点的二叉树或非叶子结点只有左子树的二叉树

D．　只有根结点的二叉树或非叶子结点只有右子树的二叉树

23．如果根的高度为 1，具有 61 个结点的完全二叉树的高度为（　　）。

A．5　　B．6　　C．7　　D．8

24.一棵二叉树如右图所示，若采用顺序存储结构，即用一 维数组元素存储该二叉树中的结点（根结点的下标为 1， 若某结点的下标为 i ，则其左孩子位于下标 2i 处、右孩 子位于下标(2i+1)处），则图中所有结点的最大下标为（　　）。

A．　6　　B．　10　　C．　12　　D．　15

2.4 图论基础

图 G（Graph）由两个集合 V（Vertex）和 E（Edge）组成，记为 G=(V,E)，其中 V 是顶点的有限集合，记为 V(G)，E 是连接 V 中两个不同顶点（顶点对）的边的有限集合，记为 E(G)。

2.4.1 基本术语

1. 端点和邻接点

在一个无向图中，若存在一条边(i，j)，则称顶点 i 和顶点 j 为此边的两个端点，并称它们互为邻接点。

在一个有向图中，若存在一条边<i，j>，则称此边是顶点 i 的一条出边，同时也是顶点 j 的一条入边；称顶点 i 和顶点 j 分别为此边的起始端点（简称为起点）和终止端点（简称终点）；称顶点 i 和顶点 j 互为邻接点。

2. 顶点的度、入度和出度

在无向图中，顶点所具有的边的数目称为该顶点的度。在有向图中，以顶点 i 为终点的入边的数目，称为该顶点的入度。以顶点 i 为始点的出边的数目，称为该顶点的出度。一个顶点的入度与出度的和为该顶点的度。

若一个图中有 n 个顶点和 e 条边，每个顶点的度为 di（1≤i≤n），则有：

$$e=\frac{1}{2}\sum_{i=1}^{n} d_i$$

例 1 一个无向连通图中有 16 条边，所有顶点的度均小于 5，度为 4 的顶点有 3 个，度为 3 的顶点有 4 个，度为 2 的顶点有 2 个，则该图有_____个顶点。

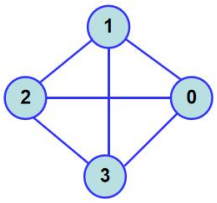
A.10　　B.11　　C.12　　D.13

解：设该图有 n 个顶点，图中度为 i 的顶点数为 ni（0≤i≤4），显然 n0=0，n=3+4+2+n1+n0=9+n1,而度之和=4×3+3×4+2×2+n1=28+n1,而度之和=2e=32,所以有 28+n1=32，得 n1=4，n=9+n1=13。本题答案为 D。

3. 完全图

若无向图中的每两个顶点之间都存在着一条边，有向图中的每两个顶点之间都存在着方向相反的两条边，则称此图为完全图。

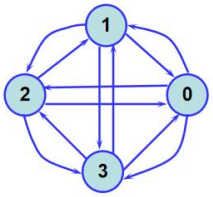
显然，完全无向图包含有条边，完全有向图包含有 n(n-1)条边。例如，图(a)所示的图是一个具有 4 个顶点的完全无向图，共有 6 条边。图(b)所示的图是一个具有 4 个顶点的完全有向图，共有 12 条边。



(a)

4. 稠密图、稀疏图

当一个图接近完全图时，则称为稠密图。相反，当一个图含有较少的边数（即当 e<<n(n-1)）时，则称为稀疏图。



(b)

5. 子图

设有两个图 G=(V，E)和 G’=(V’，E’)，若 V’是 V 的子集，即 V’　V，且 E’是 E 的子集，即 E’　E，则称 G’是 G 的子图。例如图(b)是图(a)的子图，而图(c)不是图(a)的子图。

6. 路径和路径长度

在一个图 G=(V，E)中，从顶点 i 到顶点 j 的一条路径是一个顶点序列(i,i1,i2,…,im,j)，若此图 G 是无向图，则边(i,i1)，(i1,i2)，…，(im-1,im)，(im,j)属于 E(G)；若此图是有向图，则<i,i1>，<i1,i2>，…，<im-1,im>，<im,j>属于 E(G)。

路径长度是指一条路径上经过的边的数目。若一条路径上除开始点和结束点可以相同外，其余顶点均不相同，则称此路径为简单路径。例如，有图中，(0，2，1)就是一条简单路径，其长度为 2。

7. 回路或环

若一条路径上的开始点与结束点为同一个顶点，则此路径被称为回路或环。开始点与结束点相同的简单路径被称为简单回路或简单环。

例如，右图中，(0，2，1，0)就是一条简单回路，其长度为 3。

8. 连通、连通图和连通分量

在无向图 G 中，若从顶点 i 到顶点 j 有路径，则称顶点 i 和 j 是连通的。

若图 G 中任意两个顶点都连通，则称 G 为连通图，否则称为非连通图。

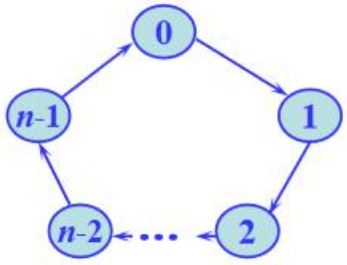
无向图 G 中的极大连通子图称为 G 的连通分量。显然，任何连通图的连通分量只有一个，即本身，而非连通图有多个连通分量。

9. 强连通图和强连通分量

在有向图 G 中，若从顶点 i 到顶点 j 有路径，则称从顶点 i 到 j 是连通的。

若图 G 中的任意两个顶点 i 和 j 都连通，即从顶点 i 到 j 和从顶点 j 到 i 都存在路径，则称图 G 是强连通图。

有向图 G 中的极大强连通子图称为 G 的强连通分量。显然，强连通图只有一个强连通分量，即本身，非强连通图有多个强连通分量。



10. 权和网

图中每一条边都可以附有一个对应的数值，这种与边相关的数值称为权。权可以表示从一个顶点到另一个顶点的距离或花费的代价。边上带有权的图称为带权图，也称作网。

例 2　有 n 个顶点的有向强连通图最多需要多少条边？最少需要多少条边？

解：有 n 个顶点的有向强连通图最多有 n(n-1)条边（构成一个有向完全图的情况）；最少有 n 条边（n 个顶点依次首尾相接构成一个环的情况）。

2.4.2 图的存储结构

1　邻接矩阵存储方法

邻接矩阵是表示顶点之间相邻关系的矩阵。设 G=(V，E)是具有 n（n＞0）个顶点的图，顶点的顺序依次为 0～n-1，则 G 的邻接矩阵 A 是 n 阶方阵，其定义如下：

（1）如果 G 是无向图，则：

A[i][j]=1：若(i,j)∈E(G)　0:其他

（2）如果 G 是有向图，则：

A[i][j]=1：若<i,j>∈E(G)　0:其他

（3）如果 G 是带权无向图，则：

A[i][j]= wij ：若 i≠j 且(i,j)∈E(G)　0：i=j　∞：其他

（4）如果 G 是带权有向图，则：

A[i][j]=　wij ：若 i≠j 且<i,j>∈E(G)　0：i=j　∞：其他

主要特点：一个图的邻接矩阵表示是唯一的。

2 邻接表存储方法

图的邻接表存储方法是一种顺序分配与链式分配相结合的存储方法。在表示含 n 个顶点的图的邻接表中，每个顶点建立一个单链表，第 i（0≤i≤n-1）个单链表中的结点表示依附于顶点 i 的边（对有向图是以顶点 i 为尾的边）。每个单链表上附设一个表头结点，将所有表头结点构成一个表头结点数组。边结点（或表结点）和表头结点的结构如下：

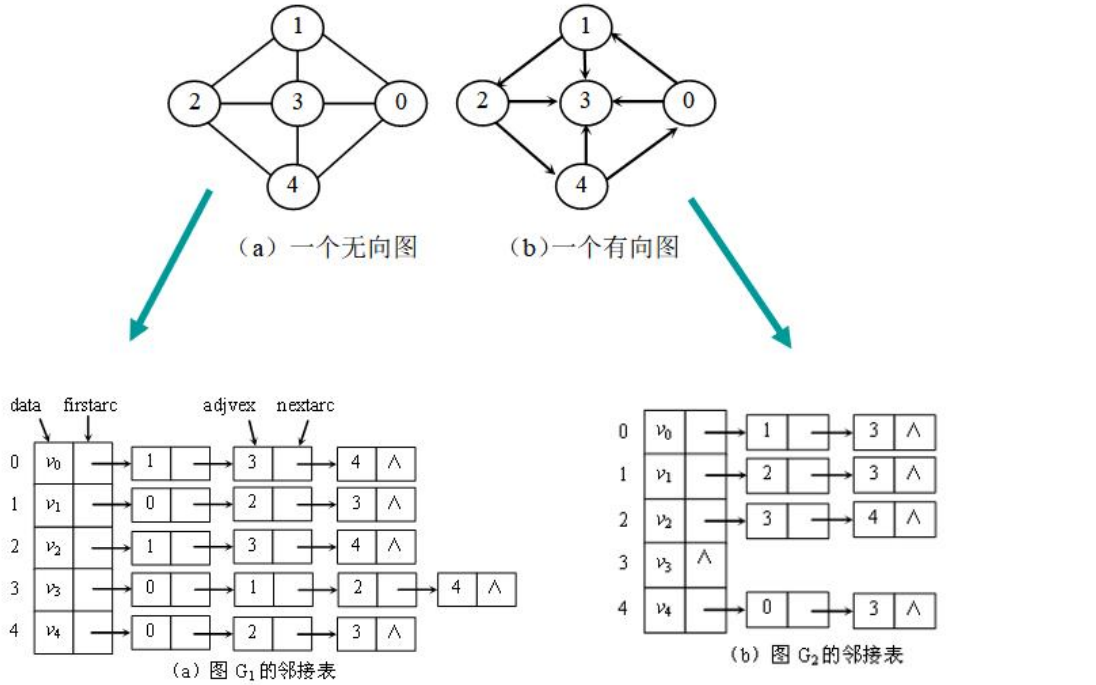
边结点（或表结点）

adjvex	nextarc	weight

表头结点

data	firstarc

其中，边结点由三个域组成，adjvex 指示与顶点 i 邻接的顶点的编号，nextarc 指示下一条边的结点，weight 存储与边相关的信息，如权值等。表头结点由两个域组成，data 存储顶点 i 的名称或其他信息，firstarc 指向顶点 i 的链表中第一个边结点。



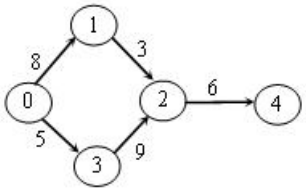
邻接表主要特点：一个图的邻接表表示是不唯一的。这是因为在每个顶点对应的单链表中，各边结点的链接次序可以是任意的，取决于建立邻接表的算法以及边的输入次序。

2.3.3 图的搜索

深度优先搜索遍历的过程是：

（1）从图中某个初始顶点 v 出发，首先访问初始顶点 v。

（2）选择一个与顶点 v 相邻且没被访问过的顶点 w 为初始顶点，再从 w 出发进行深度优先搜索，直到图中与当前顶点 v 邻接的所有顶点都被访问过为止。



例如，对于以下有向图：从顶点 0 开始进行深度优先遍历，可以得到如下访问序列：0 1 2 4 3，0 1 3 2 4。而 0 1 3 2 4 不是深度优先遍历序列。

作业

1. 平面上有五个点 A(5, 3)，B(3, 5)，C(2, 1)，D(3, 3)，E(5, 1)。以这五点作为完全图 G 的顶点，每两点之间的直线距离是图 G 中对应边的权值。以下哪条边不是图 G 的最小生成树中的边（　　）。

A. AD　　B. BD　　C. CD　　D. DE

2、已知 n 个顶点的有向图，若该图是强连通的（从所有顶点都存在路径到达其他顶点），则该

图中最少有多少条有向边？（　　）

A. n　　B. n+1　　C. n-1　　D. n*(n-1)

3．关于拓扑排序，下面说法正确的是（　　）。

A. 所有连通的有向图都可以实现拓扑排序

B. 对同一个图而言，拓扑排序的结果是唯一的

C. 拓扑排序中入度为 0 的结点总会排在入度大于 0 的结点的前面

D. 拓扑排序结果序列中的第一个结点一定是入度为 0 的点

4．无向完全图是图中每对顶点之间都恰好有一条边的简单图。已知无向完全图 G 有 7 个顶点，则它共有（　　）条边。

A．7　　B．21　　C．42　　D．49

5. 对一个有向图而言，如果每个节点都存在到达其他任何节点的路径，那么就称它是强连通的。例如，有图就是一个强连通图。事实上，在删掉边（　　）后，它依然是强连通的。

A. a　　B．b　　C．c　　D．d

6. 在一个无向图中，如果任意两点之间都存在路径相连，则称其为连通图。下图是一个有 4 个顶点、 6 条边的连通图。若要使它不再是连通图，至少要删去其中的（　　）条边。

A. 1　　B. 2　　C. 3　　D. 4

7. 有向图中每个顶点的度等于该顶点的(　　)。

A.入度　　B.出度　　C.入度和出度之和　　D.入度和出度之差

8. 6 个顶点的连通图的最小生成树，其边数为（　　）。

A. 6　　B. 5　　C. 7　　D. 4

9. 设简单无向图 G 有 16 条边且每个顶点的度数都是 2，则图 G 有（　　）个顶点。

A.10　　B.12　　C.8　　D.16

10. 某大学计算机专业的必修课及其先修课程如下表所示：

课程代号	C0	C1	C2	C3	C4	C5	C6	C7
课程名称	高等数学	程序设计语言	离散数学	数据结构	编译技术	操作系统	普通物理	计算机原理
先修课程			C0, C1	C1, C2	C3	C3, C7	C0	C6

请你判断下列课程安排方案哪个是不合理的（　　）。

A. C0, C6, C7, C1, C2, C3, C4, C5　　B. C0, C1, C2, C3, C4, C6, C7, C5

C. C0, C1, C6, C7, C2, C3, C4, C5　　D. C0, C1, C6, C7, C5, C2, C3, C4

3 排列组合专题

一、掌握优先处理元素（位置）法
二、掌握捆绑法
三、掌握插空法
四、隔板法
五、分组分配问题：
　　1、是否均匀；
　　2、是否有组别。

1、什么叫做从 n 个不同元素中取出 m 个元素的一个排列？
从 n 个不同元素中取出 m（m≤n）个元素，按照一定的顺序排成一列，叫做从 n 个不同元素中取出 m 个元素的一个排列.
2、什么叫做从 n 个不同元素中取出 m 个元素的排列数？
从 n 个不同的元素中取出 m（m≤n)个元素的所有排列的个数，叫做从 n 个不同元素中取出 m 个元素的排列数.　　用符号　　表示
3、排列数的两个公式是什么？

$$A_n^m = n(n-1)(n-2)\cdots(n-m+1)$$

$$A_n^m = \frac{n!}{(n-m)!}$$

组合定义：一般地说，从 n 个不同元素中，任取 m（m≤n）个元素并成一组，叫做从 n 个不同元素中取出 m 个元素的一个组合。

组合数公式：

$$C_n^m = \frac{n!}{m!(n-m)!} = \frac{n(n-1)\cdots(n-m+1)}{m!}$$

例 1：
（1）7 位同学站成一排，共有多少种 不同的排法？

（2）7 位同学站成两排(前 3 后 4)，共有多少种不同的排法？

（3）7 位同学站成一排，其中甲站在中间的位置,共有多少种不同的排法？

（4）7 位同学站成一排，甲、乙只能站在两端的排法共有多少种？

（5）7 位同学站成一排，甲、乙不能站在排头和排尾的排法共有多少种？

优限法:
对于“在”与“不在”等类似有限制条件的排列问题,常常使用“直接法”(主要为“特殊位置法”和“特殊元素法”)或者“排除法”,即优先考虑限制条件.这种方法就是优限法.

一般地，对于有限制条件的排列问题，有以下两种方法：
⑴直接计算法
排列的限制条件一般是：某些特殊位置和特殊元素. 解决的办法是“特事特办”，对于这些特殊位置和元素，实行优先考虑，即特殊元素预置法、特殊位置预置法.
⑵间接计算法
先抛开限制条件，计算出所有可能的排列数，再从中减去不合题意的排列数，特别要注意：不能遗漏，也不能重复. 即排除法.

捆绑法:
对于相邻问题,常常先将要相邻的元素捆绑在一起,视作为一个元素,与其余元素全排列,再松绑后它们之间进行全排列.这种方法就是捆绑法.

插空法:
对于不相邻问题,先将其余元素全排列,再将这些不相邻的元素插入空挡中,这种方法就是插空法.

有组别问题
若分成的 m 组是有组别的，只需在原来的分组基础上再
分组分配问题主要有分组后有分配对象(即组本身有序)的均分与不均分问题及分组后无分配对象(即组本身无序)的均分与不均分问题四种类型。

例 2：七个家庭一起外出旅游，若其中四家是一个男孩，三家是一个女孩，现将这七个小孩站成一排照相留念。
（1）若三个女孩要站在一起，有多少种不同的排法？
（2）若三个女孩要站在一起，四个男孩也要站在一起，有多少种不同的排法？
（3）若三个女孩互不相邻，有多少种不同的排法？
男生、女生相间排列，有多少种不同的排法？
甲、乙两人的两边必须有其他人，有多少种不 同的排法？

例 3.1、将四个不同的小球分成两组，每组两个，有多少分法？
1、将四个不同的小球分给两人，每人两个，有多少分法？
2、将四个不同的小球分成两组，一组三个，一组一个，有多少分法？
3、将四个小球分给两人，一人三个，一人一个，有多少分法？
例 3:有 6 本不同的书，分成 3 堆.
（1）如果每堆 2 本，有多少种分法？

（2）如果分成一堆 1 本，一堆 2 本，一堆 3 本，有多少种分法？

例 4:有 6 本不同的书，分成 4 堆.
（3）如果一堆 3 本，其余各堆各 1 本，有多少种分法？
（4）如果每堆至多 2 本，至少 1 本，有多少种分法？
（5）例 5:从 6 个学校中选出 30 名学生参加数学竞赛,每校至少有 1 人,这样有几种选法?
变式 1:将 7 只相同的小球全部放入 4 个不同盒子，每盒至少 1 球的放法有多少种？
变式 2:将 7 只相同的小球全部放入 4 个不同盒子，每盒可空，不同的放法有多少种？

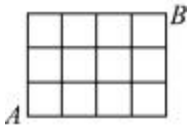
例 5：李先生与其太太有一天邀请邻家四对夫妇共 10 人围坐一圆桌聊天，试求下列各情形之排列数：

（1）男女间隔而坐。
（2）主人夫妇相对而坐。
（3）每对夫妇相对而坐。
（4）男女间隔且夫妇相邻。
（5）夫妇相邻。
（6）男的坐在一起，女的坐在一起。

一、填空题

1. 四对夫妇围一圆桌而坐，夫妇相对而坐的方法有_____种．
2. $\{1, 2\} \subset A \subset \{1, 2, 3, 4, 5\}$，且 A 有 4 个元素，则这种集合 A 有_____个．
3. 从 2000 到 3000 的所有自然数中，为 3 的倍数或 5 的倍数者共有_____个．
4. 从 1 至 10 的十个正整数中任取 3 个相异数，其中均不相邻的整数取法有_____种．
5. 从 1-10 这十个数中取出两个互异的数，使得它们的乘积是偶数，问共有多少种不同的取数方法
6. 某女生有上衣 5 件、裙子 4 件、外套 2 件，请问她外出时共有_____种上衣、裙子、外套的搭配法．（注意：外套可穿也可不穿．）
7. 李先生与其太太有一天邀请邻家四对夫妇围坐一圆桌聊天，试求下列各情形之排列数：
   (1)男女间隔而坐且夫妇相邻_____．
   (2)每对夫妇相对而坐_____．
8. 体育课后，阿珍将 4 个相同排球，5 个相同篮球装入三个不同的箱子，每箱至少有 1 颗球，则方法有_____种．
9. 0、1、1、2、2、2、2 七个数字全取排成七位数，有_____种方法．
10. 把 1～4 四个自然数排成一行，若要求除最左边的位置外，每个位置的数字比其左边的所有数字都大或都小，则共有_____种排法．（例如：2314 及 3421 均为符合要求的排列）

11. 如图，从 A 走到 B 走快捷方式，可以有_____种走法．



12. 将 100 元钞票换成 50 元、10 元、5 元、1 元的硬币，则
(1)50 元硬币至少要 1 个的换法有_____种．
(2)不含 1 元硬币的换法有_____种．

13. 有 7 个一模一样的苹果，放到 3 个一样的盘子中，一共有（　）种放法。
A. 7　　B. 8　　C. 21　　D. 37

14. 周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责 切菜、妈妈负责炒菜。假设做每道菜的顺序都是：先洗菜 10 分钟，然后切 菜 10 分钟，最后炒菜 10 分钟。那么做一道菜需要 30 分钟。注意：两道不 同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗， 也不能同时切。那么做完三道菜的最短时间需要（　）分钟。
A. 90　　B. 60　　C. 50　　D. 40

4 绍兴市分类真题

4.1 计算机基础

1. 世界上第一台电子计算机 ENIAC 于（　　）年诞生于美国。

A.1988　　B.1981　　C.1946　　D. 1979

2. 我们一般会根据计算机主要的元器件组成来划分它所属的阶段.那么 1946 年制造的 ENIAC 属于（　　）计算机。

A.第二代　　B.第三代　　C.第四代　　D. 第一代

3. 下列著名人物中，没有在计算机相关技术和理论领域作出过杰出贡献的人是（　　）。

A.王选　　B.图灵　　C,冯·诺依曼　　D.陈景润

4.“国际信息学奥林匹克竞赛”的英文缩写是（　　）。

A.GXA　　B.NOI　　C.IOI　　D.NOIP

5.为了表彰为计算机科学作出突出贡献的科学家，计算机界设立了一项素有“计算机界届的诺贝尔奖”美誉的著名奖项，该奖项名称是为了纪念一位英国计算机科学家而以他的名字命名的，这位科学家是（　　）

A.冯·诺依曼　　B.比尔·盖茨　　C.帕斯卡　　D.图灵

6.图灵奖”是计算机科学界的诺贝尔奖。迄今为止，我国只有一位科学家获得该项殊荣。这位科学家是（　　）

A.盖茨　　B.姚期智　　C.李开复　　D.王江明

7.“全国青少年信息学奥林匹克竞赛”的英文缩写是（　　）

A.IOI　　B.NOIP　　C.NOI　　D.ACM/ICPC

下列哪一位被称为“计算机科学之父”（　　）

A.冯·诺依曼　　B.比尔·盖茨　　C.帕斯卡　　D.图灵

9.人工智能英文缩写为（　　）。是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。他是计算机科学的一个分支，它企图了解智能的实质，并衍生出一种新的能以人类智能相似的方式做出反应的智能机器，该领域的研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。

A. AT　　B. ALBB　　C. AM　　D. AI

10.机器学习(Machine Learning, ML)专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能。它是人工智能的核心，是使计算机具有智能的根本途径，其应用遍及人工智能的各个领域。它主要使用归纳、综合而不是演绎。以下哪个应用是它目前还无法实现的（　　）。

A. 证券市场预测　　B. 数据挖掘

C. DNA 序列测序　　D. 人脸识别

4.2 计算机硬件软件

1. 世界上第一台电子计算机 ENIAC 于（　　）年诞生于美国。

A.1988　　B. 1981　　C. 1946　　D. 1979

2. 下列计算机设备中，是输出设备的是（　　）

A. RAM　　B. 鼠标　　C. 键盘　　D. 打印机

3. 下列计算机设备中，断电后其中的信息全部消失的是（　　）

A. RAM　　B. CPU　　C. ROM　　D. 硬盘

4. 下列操作系统中，不是微软公司产品的是（　　）

A. LINUX　　B.WINDOWS 98　　C. WINDOWS 2000　　D. WINDOWS XP

5. 一般计算机开机后总会自动启动 Windows 操作系统，那么这个操作系统软件安装在计算机的哪个硬件 中（　　）？

A.软盘　　B. 主板芯片　　C.硬盘　　D.ROM

6.下列对于计算机病毒的认识中，错误的是（　　）。

A.只要不从因特网上下载文件，而只在网上浏览网页、收发电子邮件是不会感染病毒的。

B.要定期升级杀毒软件，并利用杀毒软件对计算机进行查、杀毒处理。

C.一台接入网络的计算机，即使不进行网络的相关操作（浏览网页、收发邮件、下载文件等），也有可能被病毒感染。

D.安装病毒防火墙可以从一定程度上有效地预防病毒的感染。

7.下列计算机设备中是存储设备的是（　　）

A.键盘　　B.RAM　　C.显示器　　D. CPU

8.我们一般把能播放 VCD、CD 光盘来看电影、听音乐的电脑称为“多媒体电脑”，下列硬件设备中，是多媒体电脑必须具备的是（　　）。

A.CD-ROM 光驱　　B.网卡　　C.扫描仪　　D.打印机

9.计算机有计算功能，那么这个“计算”是在下列哪个硬件设备中完成的（　　）

A.CPU　　B. ROM　　C.内存　　D.硬盘

10.计算机的存储系统中，能被 CPU 直接存取的是（　　）。

A.内存储器　　B.磁盘存储器　　C.CD-ROM　　D.外存储器

11.下列可选项中，都是硬件的是（　　）。

A.Windows、ROM 和 CPU　　B.WPS、RAM 和显示器

C.ROM、RAM 和 C++　　D.硬盘、光盘和软盘

12.应用软件是专业人员为各种应用目的而编制的程序，以下（　　）是应用软件。

A.操作系统　　B.文字处理软件

C.数据库管理系统　　D.语言处理系统

13.味子利用 WORD 软件在写作文，当她一开始启动 WORD，然后输入文字“信息学奥林匹克竞赛”，在系统没有自动存盘和手工保存这个 WORD 文件之前，“信息学奥林匹克竞赛”这几个文字存在于该计算机系统的（　　）中。

A.ROM　　B.RAM　　C.光盘　　D.硬盘

14.下列选项中，对计算机运行速度的快慢没有影响的是（　　）。

A.CPU　　B.内存

C.硬盘的缓存　　D.显示器的屏幕尺寸

15.如下图（图 1）所示的硬件设备中，主要用物输出声音信息的是（　　）。

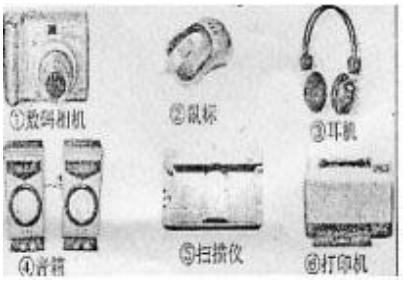
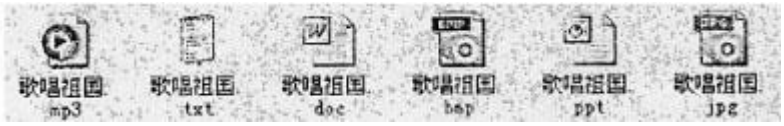


图 1

A.④⑥　　B.③④　　C.①⑤　　D.②③

16.学校组织合唱比赛，老师让味味收集一些有关歌曲《歌唱祖国》的资料，味味就通过因特网收集相关资料并保存在计算机中，今天她想通过电子邮件将其中的音乐文件发送给老师，于是打开了包含如下图所示文件信息的文件夹，其中最有可能是《歌唱祖国》音乐文件的是（　　）。



A.歌唱祖国.doc　　B.歌唱祖国.txt　　C.歌唱祖国.jpg　　D.歌唱祖国.mp3

17.当我们在 C++编程系统中刚编写完成一个程序（期间没有进行过存盘操作，此时，我们从键盘输入的 C++程序被保存在计算机的（　　）中。

A.硬盘　　B.内存　　C.光盘　　D.显示器

18.有两台计算机，1 号机和 2 号机屏幕设置分别如第 5 题图 1 和第 5 题图 2 所示，则下列说法正确的是（　　）

第 5 题　图 1

第 5 题　图 2

A.相同模式下 1 号机桌面显示的图标比 2 号机大

B.相同模式下 1 号机的颜色数比 2 号机多

C.相同模式下 1 号机的颜色数比 2 号机少

D.相同模式下 1 号机桌面显示的图标比 2 号机小

19.在下列计算机硬件设备中，承担“计算”任务的设备是（　　）

A. CPU　　B.内存　　C.硬盘　　D.光驱

20.上阅卷，是通过网络对考生的电子图像进行评阅的阅卷方式。下列设备中，最适合采集答卷图案的是（　　）。

A.显示器　　B.扫描仪　　C.打印机　　D.键盘

21.下列软件不能用于浏览网页的是（　　）

A. Google　　B. Internet Explorer

C.Access　　D. QQ 浏览器

22.路人甲要拍摄一些学校风景照片,并对拍摄的照片进行处理,下列采集工具、加工软件可实现这一功能的是（　　）

A.数码相机、photoshop　　B.扫描仪、goldwave

C.数码摄像机、goldwave　　D.手机、access

23.某计算机的部分参数如下表所示，其中不能体现“运行速度快”特征的参数是（　　）

①	硬盘容量	800G
②	网卡	千兆网卡
③	显示器分辨率	1280 × 1024
④	处理器	Inter(r) Core? i5-2450M 2.50GHz

A.①②　　B.③④　　C.①③④　　D.①②③

24.下图是乐乐在手机上的操作，该操作主要应用了人工智能中的（　　）

A.语音识别　　B.虚拟现实　　C.智能翻译　　D.人机博弈

25. 下面哪个是即时通讯工具（　　）

A.微信　　B.推特 Twitter　　C.微博　　D.脸谱 facebook

26. 一个完整的计算机系统应包括（　　）

A.系统软件和应用软件

B.硬件系统和软件系统

C.主机和外部设备

D.主机、键盘、显示器和辅助存储器

27. 微型计算机内存储器地址是按（　　）编址的。

A.二进制位　　B.字长　　C.字节　　D.微处理器的型号

28. office 中"剪贴板"是（　　）。

A.硬盘中的一块区域　　B.内存中的一块区域

C. cache 中的一块区域　　D. cpu 中的一块区域

29. 下列存储器按存取速度由快至慢排列，正确的是（　　）。

A. 硬盘>RAM>高速缓存>U 盘

B. 高速缓存>RAM>硬盘>U 盘

C. 高速缓存>硬盘>RAM>U 盘

D. U 盘>硬盘>RAM>高速缓存

30. 下列属于输入设备的是（　　）。

A.显示器　　B.触摸屏　　C.音响　　D.打印机

31. 不同的计算机之间，指令系统也不相同，这一差异主要取决于（ B　　）。

A.系统的总体结构　　B.所使用的中央处理器

C. 所使用的操作系统　　D. 所使用的程序设计语言

4.3 进制与编码

1. ASCII 码最多能表示的符号数目是（　　）

A.256　　B.128　　C.1024　　D.64

2. 二进制数(1011)2 对应的十进制数是（　　）。

A.1011　　B.15　　C.10　　D.11

3. 十进制数 11 对应的二进制数是（　　）

A.1011　　B. 1100　　C.0011　　D. 1010

4. C++程序设计中，用 short 类型来保存整数，下列整数中用 short 类型变量正确保存的是（　　）

A.32650　　B.40000　　C.60000　　D.50000

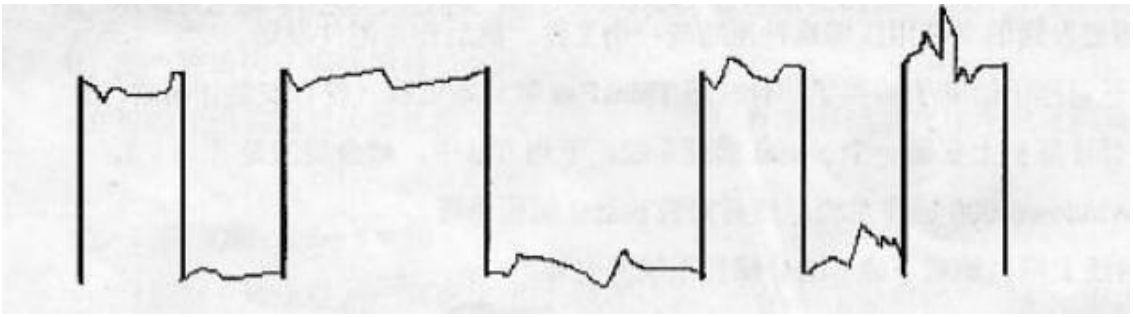
5. 与十六进制 3D 相等的数是（　　）。

A.(60)10　　B.（00111110）2　　C.(76)8　　D.（00111101）2

6. 在计算机内部，本质上只存在高电压和低电压，一般高电压用 1 表示，低电压用 0 表示（注意，某个 1 或者 0 表示的区间长度必须是相同的），下面是味子用示波器测得的 某次电压波动曲线，如果用一般二进制数表示，则应该是（　　）。

A.010011010　　B.0101010　　C.1010101　　D.101100101

7. 在计算机内部，所有的计算都是以二进制方式进行的。比如，我们要计算机 12+7=？，那么计算机首先会将 12 和 7 转化成二进制数，然后进行二进制加法运算，那么 12 加 7 的计算结果用二进制表示是（　　）。



A.1100　　B.1111　　C.10011　　D.11001

8. 下图（图 2）所示是一个 8*8 像素的黑白二色位图，假如使用 0 表示白色、1 表示黑色，那么这幅黑白二色位图如果要在计算机内部完整地保存，在不进行压缩的前提下，最少需要的存储空间是（　　）。

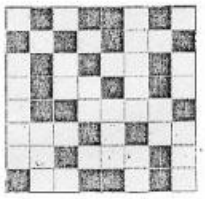


图 2

A. 8 Byte　　B. 8 bit　　C.64 KB　　D.8 KB

9. 按照前面第 3 题的约定，则第 3 题图中第 7 行的图像信息用二进制编制码为（　　）。

A. 11011101　　B. 01010101　　C. 10011010　　D. 00100010

10. 下列二进制和十进制数中，最大的是（　　）。

A.（48）10　　B.（19）10　　C.（110011）2　　D.（11001）2

11. 下列常量或表达式中，数值最大的是（　　）。

A.（257）10　　B.（1000）2+（10）2

C.（110011）2　　D.（250）10+（11）2

12.国庆长假期间，凯月跟着爸爸去了九寨沟，听爸爸说九寨沟风景优美，凯月就打算多拍一些照片回来。凯月的数码相机一般每张照片需要占用存储卡 980KB 的空间，现在凯月带了一张容量为 256MB 的存储卡。则从理论上讲，凯月最多能拍的照片张数为（　　）。

A.261　　B.267　　C.130　　D.131

13.下列表达式逻辑值为“true”的是（　　）

A. (256)10<(10000001)2　　B. (256)10>=(512)10

C. (1111111)2>(512)10　　D. (10000001)2<(131)10

14.在 C++ 中，“>>x”命令的作用是将一个数对应的二进制数各位置上的数字右移 x 个位置，前面（高位）新位置用零补充。如对二进制数(01100011)2 执行“>>2”命令后的结果就是(00011000)2。则对十进制数 23 执行命令“>>2”后结果为（　　）

A.(92)10　　B. (21)10　　C. (00000101)2　　D. (101)10

15. 二进制加法表达式(1001)2 + (1101)2 的计算结果为（　　）

A.(22)2　　B. (11)10　　C. (0110)2　　D. (22)10

16. 单字节 5 和-5 在计算机内部分别表示为（　　）

A.00000101　　10000101　　B. 00000101　　11111010

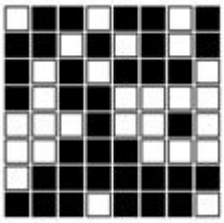
C.11111011　　11111011　　D. 00000101　　11111011

17. 十进制数 71 转换成二进制数是（　　）。

A.(1000111)2　　B. (1110001)2　　C. (1001100)2　　D. (1000011)2

18. 有一幅 8*8 像素的黑白图像，如右图所示。如果该图像的每一行按照从左到右编码，且第一行编码为 10010010，那么第三行的编码是（　　）。

A. 11010101　　B.01010001　　C. 10101110　　D. 00101010



19. 每个不同的二进制数可以表示一种颜色，如果一幅图像有 256 种颜色，最少需要几位二进制数来表示？（　　）

A.8　　B.16　　C. 128　　D. 256

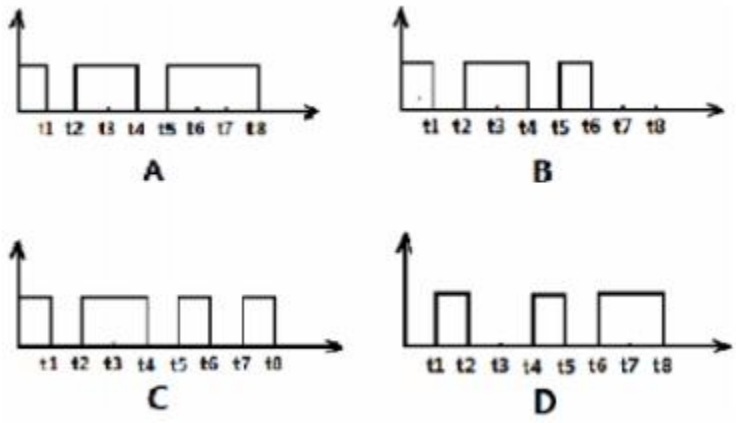
20. ASCII 码表中的大写字母 Z 后有 6 个其他字符，接着便是小写字母。现在已知：字母 Y 的 ASCII 码为(1011001)2，则字母 a 的 ASCII 码用十六进制表示是（　　）

A. 61H　　B. 62H　　C. 63H　　D. 64H

21. 字符“T”的 ASCII 码对应的二进制数为 1010100，则大写字符“P”的 ASCII 码对应的二进制是（　　）

A. 1011001　　B. 1010000　　C. 1011101　　D. 1000111

22. 若用 O 和 1 表示波形电平的两种状态，则与 10110101 相符的波形是（　　）



23. 十进制算术表达式 3*4+5*6+7*2+9 的运算结果，用二进制表示为（　　）

A. 1000001　　B. 1000010　　C.1000011　　D.1000100

24. 4KB 的内存能存储（　　）个汉字的机内码。

A.1024　　B. 516　　C. 2048　　D. 218

25. 读卡器根据透光检测判断哪些孔位已打孔，哪些未打孔，从而识别出卡的编码。如果要设计一种供 300 人使用的身份卡，则卡上的预定孔位至少需要（　　）。

A. 5 个　　B. 7 个　　C. 9 个　　D. 10 个

26. 字母 "c" 的 ASCII 码值为 99，则字母 "f" 的十六进制 ASCII 码是（　　）。

A. 66H　　B. 9CH　　C. 67H　　D. 9DH

27. 小写字母“a”的 ASCII 码为 97,小写字母 i 的 ASCII 码的值是（　　）。

A. 72　　B. 73　　C. 105　　D. 106

28. RGB 色彩模式是一种通过对红(R)、绿(G)、 蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色标准。通常情况下，RGB 三个通道各有 256 级亮度，用数字表示为

29. 从 0、1、....一直到 255。我们用二进制来表示各个通道的亮度数值，则 256 级的 RGB 色彩通常也被称为（　　）。

A. 3 位色　　B. 8 位色

C. 24 位色　　D. 256 位色

29. [x]补码=01100110，其原码是（　　）。

A.00110011　　B. 00010110　　C. 00011011　　D. 01100110

30. 已知变量 a=true, b=false。 如下关系表达式中，运算结果为 true 的是（　　）。

A.a^b　　B. (a^b)V(b^a)

C. (-a^b)V(-b^a)　　D. (-aVb)^(-bVa)

31. 已知十进制数 120，则其十六进制表示是（　　）。

A. 76H　　B. 77H　　C. 78H　　D. 79H

4.4 安全

1. 为了有效地预防计算机感染计算机病毒，下列措施中，错误的是（　　）

A. 安装病毒防火墙

B. 定期用杀毒软件对计算机进行查、杀毒处理

C. 定期用酒精或消毒药水对计算机各个部件进行擦拭消毒处理

D. 不使用非法盗版软件

2. 下列不是计算机病毒特征的是（　　）

A. 破坏性　　B. 传染性　　C. 可见性　　D. 隐蔽性

3. 下列对于计算机病毒的认识中，错误的是（　　）。

A.只要不从因特网上下载文件，而只在网上浏览网页、收发电子邮件是不会感染病毒的。

B.要定期升级杀毒软件，并利用杀毒软件对计算机进行查、杀毒处理。

C.一台接入网络的计算机，即使不进行网络的相关操作（浏览网页、收发邮件、下载文件等），也有可能被病毒感染。

D.安装病毒防火墙可以从一定程度上有效地预防病毒的感染。

4. 下列关于计算机病毒的描述中，正确的是（　　）

A. 如果一个人有感冒病毒，那么他使用的计算机就有可能感染这个人身上的病毒，并最终发展为计算机病毒。

B. 如果一台计算机在生产厂家组装时周围环境不好(有灰尘等)，这台计算机就会有计算机病毒。

C. 计算机病毒实质上是一段计算机程序。

D. 计算机病毒只能通过计算机网络传播。

5. 以下是关于计算机病毒的说法，不正确的是（　　）

A. 病毒属于计算机软件

B. 病毒属于硬件

C. 病毒具有破坏性、传播性、可激发性、潜伏性、隐蔽性等特点

D. 若软盘上染上病毒，格式化软盘可以清除病毒

6. 信息时代的我们有很多的网上密码，如 QQ、电子邮箱、电子公告板等应用都需要用到密码。

从网络安全的角度来选择，下列密码中相对最安全的密码的是（　　）。

A.123456　　B. 20090901

C. zjsxZHNG&8002　　D. wangjiegang

7. 计算机系统中的“防火墙”主要作用是（　　）。

A. 从一定程度上抵御来自网络的非法入侵和攻击；

B. 检测网络中的所有电脑是否感染病毒；

C. 通过在线升级可以达到杀除所有的计算机网络病毒；

D. 网络中心用来防御火灾的隔离设备的总称。

8.下列操作习惯中，最符合信息安全要求的是（　　）

A.邮箱中有看到来历不明且带有附件的邮件，直接下载并打开附件。

B.在把别人优盘中文件复制到自己计算机前，先用金山毒霸软件对 U 盘进行查杀毒处理。

C.把一台联接到因特网的计算机中的防火墙程序关闭。

D.把自己的数字化资料全部保存在安装操作系统的 C 盘上。

9.下列做法中，能增强计算机系统安全性的有（　　）。

①安装正版杀毒软件，并定期对系统进行病毒扫描

②将复杂的管理员密码修改为相对简单的“123456”

③安装防火墙软件，防御外部攻击

④及时进行软件更新，修复系统高位漏洞

A. ①②③　　B. ①②④　　C. ①③④　　D.②③④

10..下列描述计算机病毒的特性中，（　　）不是正确的。

A.潜伏性　　B.传染性　　C.智能性　　D.危害性

11. 黑客通常是利用种植在电脑上的木马程序获取你使用的账号和密码等信息。乐乐在使用电脑过程中以下行为存在风险的是（　　）

定时为操作系统升级打补丁程序

在网络上下载的文件先杀毒后再打开

在没有打开防火墙软件情况下随意浏览网页

对 QQ 上传的软件先杀毒再打开

12.下列描述计算机病毒的特性中，（　　）不是正确的。

潜伏性　　传染性　　智能性　　危害性

4.5 网络

1. 小明为多个账户设置密码，下列方式相对安全的是（　　）

A.不同账户设置相同的密码，密码均设置为自己的生日

B.不同账户设置不同的密码，密码采用 8 位数字形式

C.不同账户设置相同的密码，密码均设置为某个英语单词

D.不同账户设置不同的密码,密码采用足够长度的字母和数字混合形式

2.下列做法符合信息安全的是（　　）。

①智能手机随意扫描商家发来的二维码以便获取优惠

②网上下载共享软件后先查杀病毒再安装使用

③不随意打开陌生邮件中的 exe 格式的附件

④为方便记忆用自己的出生年月作为网上银行登录密码
⑤定期安装操作系统补丁并升级杀毒软件
A.①②③　　B.②③④　　C.②③⑤　　D.①④⑤
3. 下列对于因特网的描述正确的是（　　）
A.是一个局域网　　B.是一个城域网
C.因特网唯一的功能就是网页浏览　　D.是一个广域网
4. 为了方便我们在因特网上查找信息，人们推出了“搜索引擎”（网站），下列网站中是“搜索引擎”的是（　　）
A. WWW.CCTV.COM　　B. WWW.GOOGLE.COM
C. WWW.SXSEDU.NET　　D. WWW.ZJEDU.ORG
5. 通过因特网.我们可以利用（　　）和远方的朋友实时聊天。
A. E-MAIL　　B. 腾讯 QQ　　C. BBS　　D. FTP
6. 我们可以按照网络覆盖的区域大小来对网络分类，下列网络类别不是按照网络覆盖区域大小分类的是（　　）
A. 广域网　　B. 城域网　　C. 以太网　　D. 局域网
7. 计算机网络的最大优点是（　　）
A. 资源共享　　B. 运算速度加快
C. 计算机精度提高　　D. 内存容量增大
8. 计算机网络中，互连的各种数据终端，是按（　　）相互通信。
A. 网络协议　　B. 连线　　C. 以太网　　D. 数据格式
9. 电子邮件的邮箱（　　）。
A. 在 ISP 的服务器上　　B. 在你申请的网站的服务器上
C. 在 Outlook Express 里　　D. 在 Outlook Express 里的电脑里
10. 为了有效地预防计算机感染计算机病毒，下列措施中，正确的是（　　）。
A. 安装病毒防火墙并开启所有实时监视功能，同时注意及时升级
B. 保护计算机所在房间的干净整洁即可。
C. 只要安装了杀毒软件后就万事大吉了。
D. 定期让专业人员打开主机机箱，然后进行除尘处理即可。
11. 味味今天发现计算机工作有点不正常，她马上用杀毒软件进行查杀，杀毒软件报告说：E 盘上发现病毒但无法清除病毒。下列措施中，肯定不能有效清楚病毒的是（　　）。
A. 关闭计算机　　B.马上升级杀毒软件，然后重新杀毒
C.对 E 盘进行格式化操作　　D. 删除所有感染病毒的文件
12. 人们在为自己的网站设置域名时总是让域名与网站的功能、内容特点相吻合。下列网络域名中，最有可能是网上电子公告板的域名是（　　）。
A. www.cctv.com　　B. club@sohu.com
C. bbs.sina.com.cn　　D. zggb.163.com
13. 下列信息中，最有可能是电子邮箱地址的是（　　）
A. hzjqjc@163.com　　B. Jsj&cctv.com
C. ftp.tinghua.cn　　D. 10.200.0.161
14.小薇要将自己计算机硬盘中一张电子贺卡（图像文件）以电子邮件的附件形式发送给朋友（操作界面如图-2 所示）。为了将该图像文件作为附件添加到邮件中，应选择的操作是（　　）

A.1　　B.2　　C.3　　D.4

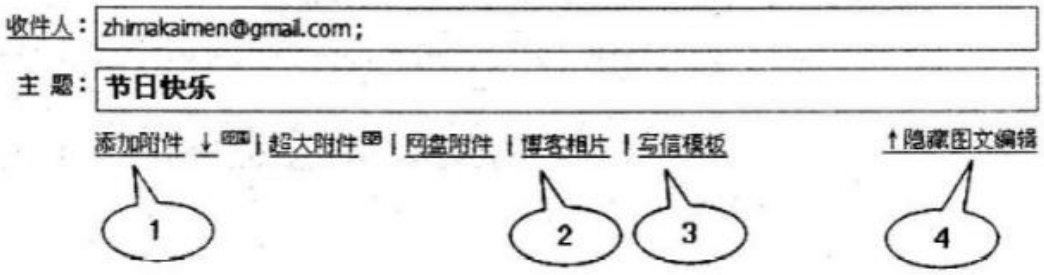


图-2

15.. 为了将浏览网页时看到的图片保存到自己的计算机上，可以将鼠标移到该图片上，单击鼠标右键，然后在如图-3 所示的快捷菜单中选择菜单命令（　　）

A. 打开链接　　B. 电子邮件图片

C. 图片另存为　　D. 在新窗口中打开链接

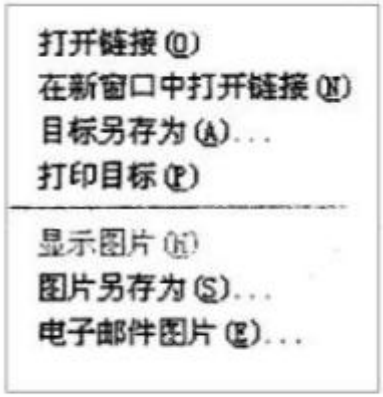


图-3

16.. 为了给自己的 QQ 设置密码，下列密码中安全性最好的是（　　）

A. Fhjts345QWBK　　B. qq2011　　C. 123456　　D. wangqiang

17. 某用户电子邮箱收件夹中的内容如图所示:

从图中可以看出该收件夹内（　　）

A.全部邮件有 6 封。　　B.未读的邮件有 4 封。

C.带附件的邮件有 5 封。　　D. 当天收到的邮件有 3 封。

18. 笑笑的爷爷最近在学电脑，现在已经会上网了，但总是记不住网址，每次上网时都要问笑笑该输什么网址，后来笑笑略施小“技”，帮爷爷解决了这个难题。之后爷爷再上网时，不仅不用记住网址就能快速地访问自己常去的一些网站，而且每次一启动 IE 就可以直接打开新华网浏览新闻，笑笑采用的方法是（　　）

①将爷爷常用的网址都添加到 IE 的收藏夹中

②将爷爷常用的网址都添加到 IE 的临时文件夹中

③将新华网的网址设置为 IE 的主页

④将爷爷常用的网址都添加到 IE 的历史记录中

A. ①②　　B. ①③　　C. ②③　　D. ②④

19. 以下不属于无线通信技术的是（　　）。

A. 蓝牙　　B. WiFi　　C. GPRS　　D. 以太网

20. IP 地址是每个上网的电脑必须的，下列 IP 地址中合法的是（　　）。

A. 225.225. 225.225　　B.. 200.256.192. 8

C. 192.168.1.1. 2　　D. 0.0.0

21. 下面哪种计算机网络不是按照拓扑结构划分的(　　)。

A.星形网　　B. 总线网　　C. 环形网　　D. 都市网

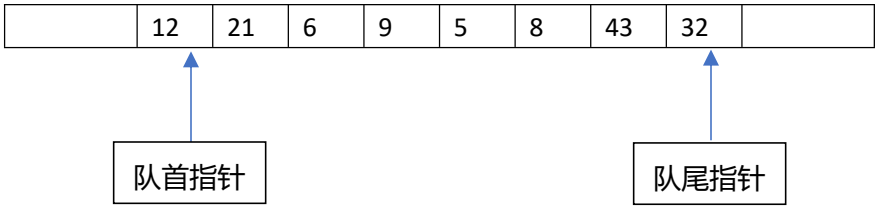
4.6 数据结构

1. 已知一个堆栈中包含了 4 个元素， 而且知道他们在堆栈中的位置依次是 a,b,c,d(从栈底开始往上数)，现在让这个堆栈进行连续出栈操作，直到堆栈空，则这些元素的出栈顺序是（　　）

A. dcba　　B. abcd　　C. abdc　　D. cdab

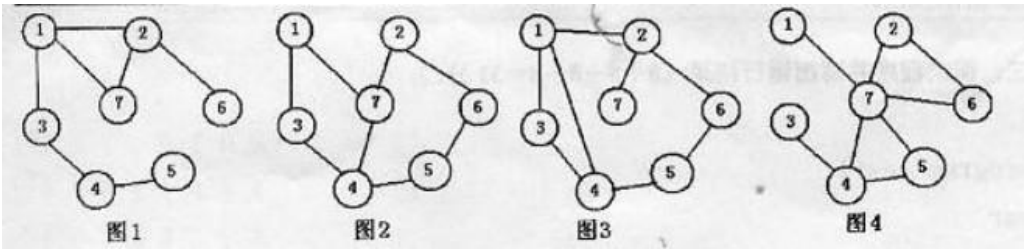
2. 下列关于树这种数据结构的说法中，正确的是（　　）。

A. 任何结点都有子结点和父结点
B. 任何结点都必须有子结点
C. 任何结点都必须有父结点
D. 在二叉树中，每个结点可能没有子结点，有的话最多只能有二个子结点
3. 下列关于二叉树的描述中， 正确的是（ ）
A.某个结点可以没有子结点、或有 1 个子结点、或有 2 个子结点
B.根结点可以有 3 个子结点
C.根结点可以有 4 个子结点
D.任何结点最多都可以有 4 个子结点
4. 已知一个队列中有若干个数（如下图所示），则最后出队的元素是（ ）



A. 12 B. 5 C. 9 D. 32
5. 给定队列的入队顺序 1，2，3，共有几种可能的出队序列（ ）。
A. 3 B. 2 C. 1 D. 4
6. 如果一棵满二叉树有 n 个叶结点，则这棵树的结点总数为（ ）。
A. 2n B. 2n-1 C. 2n+1 D. n2-1

7. 下列图中，能用“一笔画”画出（经过每条边一次且只经过一次）的图是（ ）。

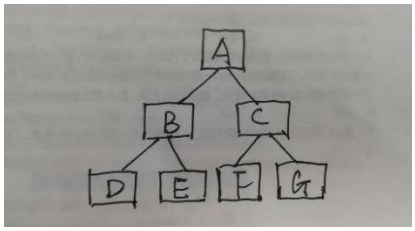


A. 图 1 B. 图 4 C. 图 2 D. 图 3
8. 如果一棵二叉树的前序遍历序列和后序遍历序列正好相反，那么该二叉树具有下列选项中哪个特征能满足前面这个条件（ ）。
A. 所有结点都只有儿子 B. 只有一个叶子结点
C. 任意一棵二叉树 D. 所有结点都只有右儿子
9. 已知队列{13，2，11，34，41，77，5，7，18，26，15}，第一个进入队形的元素是 13，后面的元素按照上述顺序依次入队然后依次出队，那么第 5 个出队的元素是（ ）。
A. 13 B. 5 C. 77 D. 41
10. 下列关于堆栈的操作中，不属于堆栈基本操作的描述是（ ）。
A. 将堆栈置空 B. 删除栈顶元素
C. 删除栈底元素 D. 判断堆栈是否为空
11.已知原始数据序列的排列是 8、7、6、5、4、3、2、1 现将通过纯粹的冒泡排序对该数列进

行从小到大的排序处理，则数据进行两两交换的总次数为（　　）。

A.28　　B.64　　C.32　　D.8

12. 所谓满二叉树指的是这样一种特殊二叉树“除了最底下一层的节点没有任何子节点，上面所有节点都有两个儿子”如下图所示就是一棵 3 层的满二叉树。那么，一棵有 5 层的满二叉树，一共包含的节点总数是（　　）。



A.32　　B.15　　C.31　　D.16

13. 农博会即将举行，主办单位收到了很多参展闪的参展申请，为了体现公平，主办单位按照参展商申请时间的先后依次给与编号从小到大排列，并且连续的展位（每个参展商一般都会申请 2 个以上的展位），现在已经有 6 家参展商提出了申请，他们各自需要的展位数量分别是：3，4，2，7，6，5，而且我们知道所有展位中第一个展位的编号是 1000（编号全部是 1000 之后连续的偶数。如，1000、1002、1004）。现在味味也来申请展位，那么她申请到的展位起始编号至少是（　　）。

A.1027　　B.27　　C.1054　　D.1108

14. 在 Word 中依次进行下列操作：

（1）输入“第六届绍兴市少儿信息学奥赛”，按回车键；

（2）进行存盘操作；

（3）继续输入“初赛试题”，然后进行存盘操作。（操作结果如下图所示）。

现在在“编辑”菜单中选择“撤销（U）键入”选项，我们发现第二行的文字“初赛试题”被删除了，继续选择“撤销（U）键入”，我们发现第一行的文字“第六届绍兴市少儿信息学竞赛”也被删除了。这种“撤销（U）键入”操作的特点，说明 word 对输入文字的保存，采用的数据结构是（　　）。

A.线性队列　　B.堆栈　　C.平衡树　　D.循环队列

15. 设有一棵 5 叉树，其中只有度为 0 或 5 两种结点。设度为 0 的结点个数有 2001 个，则度为 5 的结点个数有（　　）。

A. 400 个　　B. 401 个　　C. 500 个　　D. 501 个

16. 若对一棵完全二叉树按从上到下，从左到右进行编号，设根节点所在层为第 1 层，且根节点编号为 1，则该树的第 i 层第 j 个节点的编号为（　　）。

A. 2i-1+j　　B. 2i+j　　C. 2i+j-1　　D. 2i-1 +j-1

17. 设栈 S 的初始状态为空，现有 5 个元素组成的队列{1，2，3，4，5}，对该序列在 S 栈上依次进行如下操作（从队首元素 1 开始进行操作，出栈后不再进栈）：进栈，进栈，进栈，出栈，进栈，出栈，进栈，则先后出栈的元素序列是（　　）。

A. 4，3　　B. 2，1

C. 5，4，3，2，1　　D. 3，4

18. 为了用计算机程序对世博会某检票入口处的检票过程进行处理，则下列数据结构中，最合

适进行该处理的是（　　）

A.二叉树　　B. 队列　　C. 图　　D. 堆栈

19. 在 Word 软件中，采用堆栈来保存我们对于文档的操作行为。在 Word 操作中，我们可以按照堆栈“后进先出”的原则，通过快捷键“Ctrl+Z”逐个撤销操作效果(按一次撤销一次)。现在在一个新建 Word 文件中的同一行中依次进行下列操作（数字只表示顺序，不列入操作内容），则最后该行保留的文字信息为（　　）

①逐个输入文字(不使用词组输入)“绍兴市少儿信息学学”

②按快捷键“Ctrl+Z”一次

③逐个输入文字“奥奥赛”

④按快捷键“Ctrl+Z”两次

A. 绍兴市少儿信息学奥　　B. 绍兴市少儿信息学奥赛

C. 绍兴市少儿信息学　　D. 绍兴市少儿信息学学

20. 如图-4 所示，有数字队列 1, 2, 3, 4,...,9,按照堆栈操作序列“进、进、进、出、进、出、进、出”对右边队列中的数字逐个进行操作。上述操作结束后，堆栈左边队列中存在数字依次为（自左往右）（　　）

A. 321　　B. 125　　C. 345　　D. 123

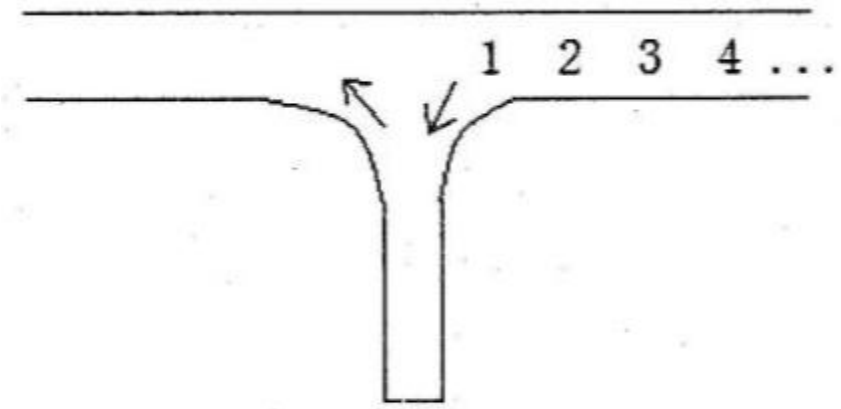


图-4

21. 在一个果园里，笑笑将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。笑笑决定把所有的果子合成一堆。每一次合并，笑笑可以把任意两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 n-1 次合并之后，就只剩下一堆了。笑笑在合并果子时总共消耗的体力等于每次合并所耗体力之和。假定有 5 堆果子，每堆果子的数量为 12，4，20，15，10，每个果子重量都为 2，笑笑想知道最小的体力耗费值是多少。（　　）

A. 272　　B. 284　　C. 164　　D. 136

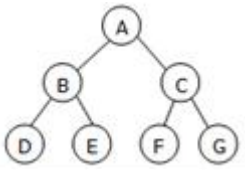
22. 有两个栈 s1,s2，有一数字序列 2 3 4 1，依次进入其中任意一个栈，任一个数字入栈后也可随时出栈，则不可能的出栈序列是：（　　）

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

23. 满二叉树一种特殊的二叉树，他除了最底下一层的结点没有任何子结点外其他所有结点都

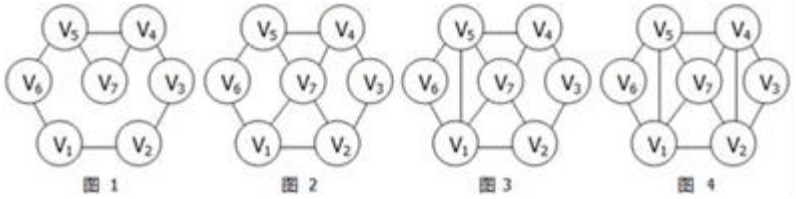
有两个子结点，如右图是一颗三层的满二叉树，那么一颗有 10 层满二叉树，一共有（　　）结点。

A. 1023　　B. 1024　　C. 2047　　D. 2048



24. 右边图形中，不能用“一笔画”（经过每条边一次仅一次）画出的图是（　　）。

图（一）　　图（二）　　图（三）　　图（四）



25. 在解决计算机主机与打印机之间速度不匹配时通常设置一个打印数据缓冲区，主要将要输出打印的数据依次写入该缓冲区，而打印机从该缓冲区中取出数据打印。该缓冲区应该是一个（　　）结构。

A. 堆栈　　B. 数组　　C. 线性表　　D. 队列

26. 地面上有标号为 A、B、C 的三根柱，在 A 柱上放有 10 个直径相同中间有孔的圆盘，从上到下依次编号为 1，2，3…，将 A 柱上的部分盘子经过 B 柱移入 C 柱，也可以在 B 柱上暂存。如果 B 柱上的操作记录为“进、进、出、进、进、出、出、进、进、出、进、出、出”。那么，在 C 柱上，从下到上的编号为（　　）。

A. 243657　　B. 241257　　C. 243176　　D. 243675

27. 一棵二叉树如图 1 所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点，根结点的下标为 1，若某结点的下标为 i,则其左孩子位于下标 2i 处、右孩子位于下标（2i+1）处，则图中所有结点的最大下标为（　　）。

A. 27　　B. 6　　C. 24　　D. 26

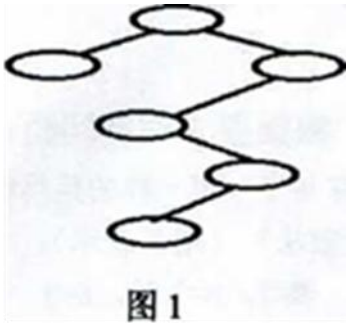


图 1

28. 今有一空栈 S,对下列待进栈的数据元素序列 a,b,c,d,e,f,g 依次进行进栈，进栈，出栈，进栈，进栈，出栈的操作，则此操作完成后，栈 S 的栈顶元素为（　　）。

A. f　　B. c　　C. a　　D. b

29. 在有 2016 个结点的连通图中，其边数最少需要（　　）。

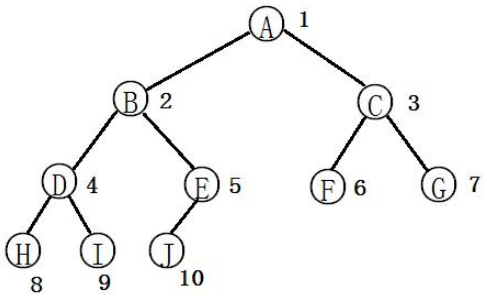
A. 2017 条　　B. 2016 条　　C. 2015 条　　D. 2014 条

30. 已知一个栈的入栈顺序是 1, 2, 3,……，n,其输出序列为 P1，P2， P3，……，pn，如果 P1 是 n,则 Pi 是（　　）

A. 不确定　　B. n-i+1　　C. n-1　　D. i

31. 若设二叉树的深度为 h,除第 h 层外，其它各层（1∽h-1)的结点数都达到最大个数，第 h 层所有的结点都连续集中在最左边，这就是完全二叉树。如图 1 所示，共有 10 个结点，5 个叶子结点，深度为 4,1~3 层的结点数都达到了最大个数。那么如果完全二叉树共计 39 个点，那么他的叶子结点的数量是（ A ）。

A. 20　　B. 21　　C. 19　　D. 23



32. 在 TCP/IP 协议栈中，TCP 是主机对主机层的传输控制协议，采用客户端与服务端相互提出与确认请求建立一个连接。通常情况下，操作系统会使用一块限定内存使用量的 TCP 缓存来处理客户端到服务端的 TCP 连接请求。如果该缓存中项目被填满，其他所有新的 TCP 连接请求会被丢弃。TCP 会依次处理 TCP 缓存中的请求，直至缓存中请求为空。TCP 缓存应该是一个（　　）结构。

A. 队列　　B. 线性表　　C. 数组　　D. 堆栈

33. 以下哪个出栈序列不能由入栈序列(1，2，3，4，5)得到（　　）

A. (1,2,3,4,5)　　B. (5,4,3,2,1)

C. (2,4,1,3,5)　　D. (2,5,4,3,1)

4.7 问题求解

1. 2 名驾驶员和 6 名空中小姐分别上二架不同型号的旅游直升飞机，每机 1 名驾驶员及 3 名空中小姐，则上机方法共有多少种？（　　）

A. 60　　B. 80　　C. 40　　D. 20

2. 4 个班长依次来图书室分书，方法都是：将自己看到的书平分成 4 份，多余 1 本送还书库，拿走 1 份。问：最后那个班长至少看到了（　　）本书。

A. 78　　B. 104　　C. 105　　D. 5

3. 如下图 2 所示，小明从家里（图中用①表示）出发到学校（图中用⑤表示）中间可能经过的路口有 3 个（图中分别用②、③、④表示），图中带箭头的线条表示从某个地点到达另外一个地点的可行线路，线条旁边的数字表示该路线的长度。由于可以选择走的路线不止一条，所以请你帮助小明选择一条从家里到学校长度最短的行走线路，并计算这个最短长度是（　　）

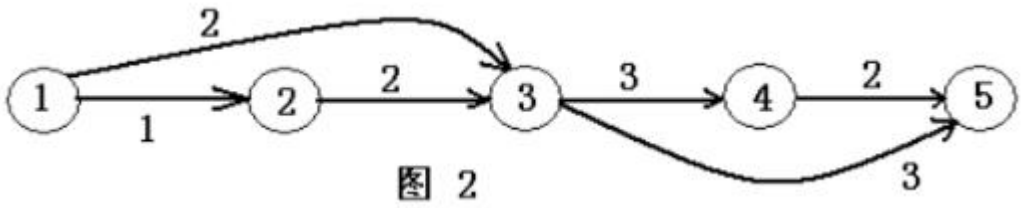


图 2

A.4　　B.5　　C.6　　D.3

4. 我们可以将一个正整数 N 拆分成 K 个正整数的和，并且任意二种拆分方法产生的正整数不能全部相同。例如，当 N=5,K=2 时，“1＋4”和“4＋1”我们认为是同一种方法。现在，N=7,K=3，请你计算一共有多少种拆分方法(　　)

A.4　　B.5　　C.6　　D.3

5. 下列程序段用来将 10 个整数（从键盘输入）进行从大到小的排序，然后按这个顺序输出排序后的 10 个整数。问：在最坏情况下，二个整数两两交换的次数是多少(　　)
例如，当将 3 个整数 1、2、3（最坏情况）从大到小排序时，两两交换的次数是 3 次。

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int i, j, a[11], t;
    for (i = 1; i <= 10; i++) {
        cin >> a[i];
    }
    for (i = 1; i <= 9; i++) {
        for (j = i + 1; j <= 10; j++) {
            if (a[i] < a[j]) {
                t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
        }
    }
    for (i = 1; i <= 10; i++) {
        cout << a[i] << endl;
    }
```

```
    return 0;
}
```

A.48　　B.24　　C.45　　D.36

6. 小明拥有各种面值的硬币 n 种（假定每种面值硬币的数量都足够多），阿强手头有一张面值为 x 的大额纸币。阿强想把自己的纸币兑换成等额的硬币，又想使兑换所得的硬币个数最少，他想了个绝妙的方法去兑换，终于达到了自己的要求。
比如，n=3（硬币面值分别是 1、3、5），x=18，则阿强兑换成 3 个面值为 5 的硬币，外加一个面值为 3 的硬币，用表达式表示就是 3×5+1×3=18，这样兑换总共得到 4 个硬币。
问：现在 n=3（硬币面值分别是 1、6、8），x=20，则如何兑换才能使得所得的硬币总数最少，是多少(　　)？（答案包括二个部分，首先是兑换方法的表达式表示，其次是写出总共得到的硬币的数量）

A.3　　B.4　　C.5　　D.6

7. 二叉树的每个结点最多只有二个子结点，而且子结点有左右之分（次序不能颠倒），现在有三个结点 a,b,c 来构成一棵二叉树,现在规定根结点必须是 a,并且 b 是 a 的左儿子。问:满足上述条件的二叉树一共有多少种不同的形态？(　　)

A.3　　B.4　　C.5　　D.6

8. 一个栈的进栈序列为 1，2，3，4，请问，出栈序列一共有多少种(　　)

A.13　　B.14　　C.15　　D.16

9. 绍兴市信息学竞赛临近，某校组队参加竞赛，要从 12 名学生中选 4 名参加比赛，其中少儿组 2 名，初中组 2 名，请问共有几种组队方法？(　　)

A.2970　　B.3360　　C.1200　　D.540

10. 有一堆火柴，一共有 n 根。现在让第一个人取走总数的一半多一根火柴，让第二个人在剩余的火柴中取走一半多一根火柴，以此类推，以后的参与者都取走前一次剩余的一半多一根，到第 4 个人来取时，他刚好把剩余的火柴全部取完。问初始时火柴总的数量 n 是多少（　）

A.20　　B.28　　C.30　　D.36

11. 假设时钟到了午夜 12（注意时针和分针重叠在一起），现在让时钟继续运行，直到时钟到了凌晨 4 点整。那么在午夜 12 点到凌晨 4 点这段时间中，时针和分针共重叠多少次（包含开始的 12 点）？继续让时钟运行下去，直到再次到达中午 12 点，那么从午夜 12 到次日的中午 12 点,这个期间时针和分针一共重叠多少次？(包含一开始的 12 点和最后结束时刻的 12 点)(　　)

A.4 13　　B.5 12　　C.3 10　　D.6 12

12. 某宾馆三楼某个房间的编号是 309（第一个数字表示楼层），该宾馆 3 到 9 层全部用来开展住宿服务（住宿房间全部安排在 3 到 9 层），如果规定所有房间都只能用同于上面 3 位数的方法来表示房间号（比如 300、301、… 400、401、… 499），那么用这种编号方法最多可以标识的房间数目是多少？(　　)

A.699　　B.700　　C.698　　D.701

13. 杰克在 A 地，他要坐着马车前往 B 地（从 A 地出发到达 B 地或从 B 地出发到达 A 地都需要 10 小时）。已知 AB 两地之间每隔 1 小时都有一辆马车准点出发前往对方（24 小时不间断），现在杰克坐着马车前往 B 地，他在途中一共可以看到多少辆从 B 地出发前往 A 地的马车？（包括在出发地 A 看到的那辆马车和在目的地 B 看到的那辆马车）(　　)

A.17　　B.18　　C.20　　D.21

14. KAD 部落是个奇怪的部落，他们认为在数字“ 0、1、2、3、4、5、6、7、8、9”中，“9”是个不吉利的数字，于是在他们的数字使用中，总是会排除这个数字。现在 KAD 部落想给一批人群按照 1、2、3、4、5、… 、i-1、i、i+1、… 的顺序编号（那当然任何编号中不会出现数字“9”）。问，现在一共有 200 个人参与编号，那么最后那个人的编号是什么？(　　)

A.241　　B.242　　C.243　　D.244

5 程序理解

5.1 基础

1.[2008]
```
#include<iostream>
using namespace std;
int main()
{
    int i, a, b, c, d, f[4];
    for(i = 0; i < 4; i++) cin >> f[i];
    a = f[0] + f[1] + f[2] + f[3];
    a = a / f[0];
    b = f[0] + f[2] + f[3];
    b = b / a;
    c = (b * f[1] + a) / f[2];
    d = f[(b / c ) % 4];
    if(f[(a + b + c + d) % 4] > f[2])
        cout << a + b<< endl;
    else
        cout << c + d << endl;
    return 0;
}
```

判断题

(1)将 05 行移到 02 03 之间，程序不会出错。　　(　　)

(2)将 13 行的“>”改为“>=”,输出不会发生改变。　　(　　)

(3)将 10 行改为 b/=a;输出不会发生改变。　　(　　)

(4)输出只会有一行。 ( )

选择题

(5)当输入为“9 19 29 39″，输出为( )。

A.21 B.22 C.23 D.24

(6)时间复杂度为( )。

A.O(1) B. O(a) C. O(aloga) D. $O(a^{loga})$

2[2010].

```
#include <iostream>
using namespace std;
void swap(int & a, int & b)
{
    int t;
    t = a;
    a = b;
    b = t;
}
int main()//题意：把 x 插入 a1，a2，a3 中，并排好序
{
    int a1, a2, a3, x;
    cin>>a1>>a2>>a3;
    if (a1 > a2)
        swap(a1, a2);
    if (a2 > a3)
        swap(a2, a3);
    if (a1 > a2)
        swap(a1, a2);
    cin>>x;
    if (x < a2)
        if (x < a1)
            cout<<x<<' '<<a1<<' '<<a2<<' '<<a3<<endl;
        else
            cout<<a1<<' '<<x<<' '<<a2<<' '<<a3<<endl;
    else
        if (x < a3)
            cout<<a1<<' '<<a2<<' '<<x<<' '<<a3<<endl;
        else
            cout<<a1<<' '<<a2<<' '<<a3<<' '<<x<<endl;
    return 0;
}
```

判断题

(1)去掉第 14 行到第 19 行,不会影响程序的运行结果。 ( )

(2)将第一行的 iostream 改成 cstdio 会编译错误。 ( )

(3)本题输出结果和将四个数从小到大排序一样。 ( )

(4)如果输入 91 2 20\n77,输出为 2 20 77 91。 ( )

选择题

(5)如果输入 114514 191 810\n258,程序输出( )。

A.114 514 258 1919810 B. 114514 191 810\n258

C.114514 191 810 258 D.191 258 810 114514

(6)如果输入 1 1 1\n2,程序输出( ).

A. 1 1 1 2 B.1 1 2 1 C.1 2 1 1 D.2 1 1 1

3[2013]

```
#include <iostream>
using namespace std;
int main()
{
    int a, b;
    cin>>a>>b;
    cout<<a<<"+"<<b<<"="<<a+b<<endl;
}
```

判断题

(1)当 a,b 为负数且(a+b)在 int 范围内，输出的等式在数学上也成立。 ( )

(2)当输人为“0 0”时，输出为“0+0= 0”。 ( )

(3)将 03 行的 int 改为 signed 程序不会出错。 ( )

(4)当输出为“-1+1=0”时候,输人可以为“2 - 2”。 ( )

选择题

(5)输人为“114514 1919810”时,输出为( )。

A.114514 + 1919810= 2034324 B.114514 + 1919810= 2034314

C.114514 + 1919810= 2024324 D.114514+ 1919810= 1034324

(6)时间复杂度为( )。

A. O((a+b)In(a-b)) B.O(1)

C.O(a^b /w) D. O(√($\log^b$ a))

4[2013]

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, u, i, num;
    cin>>a>>b>>u; num = 0;//统计在 a~b 中有多少可以整除 u 的数
```

```
    for (i = a; i <= b; i++) if ((i % u) == 0)
        num++;
    cout<<num<<endl; return 0;
}
```

●判断题

(1)u 可以为 0。 ( )

(2)b-a 一定不小于 num。 ( )

(3)输入 1 9 2,输出 4。 ( )

(4)该程序时间复杂度与 u 有关。 ( )

●选择题

(5)输出结果为 10,输人数据可能为( )。

A.10 27 2 B.20 30 10 C.5 99 3 D.41 83 4

(6)输入 151 981 7,输出结果为( )。

A117 B.118 C.120 D.119

5[2014].

```
#include <iostream>
using namespace std;
int main()
{
  int a, b, c, d, ans;
    cin >> a >> b >> c;
  d = a- b;
  a = d + c;
  ans = a * b;
  cout << "Ans = " << ans << endl;          return 0;
  }
```

判断题

(1) ans 一定为 c 的倍数。 ( )

(2) 输入为“8 8 8”，输出为“64”. ( )

(3) 将 04 行移动到 02 03 行之间，程序不会发生错误。 ( )

(4) 将 09 行的 ans 改为 a*b，输出结果不会发生改变。 ( )

选择题

(5) 输入为“2 3 4”时，答案为( ).

A.Ans=8 B.Ans=9 C.Ans=10 D.9

(6) 将 06 的“—”改为“+”,(5)的输出为( ).

A.Ans=27 B.Ans=36 C.Ans=54 D.Ans=81

6 [2009].

```
#include <iostream>
using namespace std;
int main()
{
  int a[3],b[3];
  int i,j,tmp;
  for (i=0;i<3;i++)
      cin >> b[i];
  for (i=0;i<3;i++)
  {
      a[i]=0;
      for (j=0;j<=i;j++)
      {
          a[i]+=b[j];
          b[a[i]%3]+=a[j];
      }
  }//a 数组为 2 5 26   b 数组为 2 3 47
  tmp=1;
  for (i=0;i<3;i++)
  {
      a[i]%=10;
      b[i]%=10;
      tmp*=a[i]+b[i];
  }
  cout << tmp << endl;
  return 0;
}
```

判断题

(1)输入的 3 个数越大，则输出的结果也越大。 ( )

(2)执行完第 17 行后，b[i]比 a[i]相对应的值要大。 ( )

(3)将第 18 行的 tmp=1 改为 tmp=0,不论输人什么数据,输出结果都是 0。 ( )

(4)若输人数据中包含字母，如 3 3 a 则程序会出错。 ( )

选择题

(5)输人 2 3 5.输出的结果是( )。

A.414 B.415 C.416 D.417

6 输人 1 1 1.输出的结果是( )。

A.36 B.48 C.72 D.108

7[2010].

```
#include <iostream>
using namespace std;
```

```
int rSum(int j)
{
    int sum = 0;
    while (j != 0) {
        sum = sum * 10 + (j % 10);
        j = j / 10;
    }
    return sum;
}
int main()//找 n~m 之间的回文数
{
    int n, m, i;
    cin>>n>>m;
    for (i = n; i < m; i++)
        if (i == rSum(i))
            cout<<i<<' ';
    return 0;
}
```

●判断题

(1)将第 4 行的 int 改为 unsigned,答案不会错误。 ( )

(2)程序开启 O2 优化不会返回错误。 ( )

(3)如果输入-1,程序会输出-1。 ( )

(4)该问题 r(n)的值没有规律。 ( )

●选择题

(5)如果输入 7,程序会输出( )。

A.-1 B.5 C. 1 D.3

(6)如果输入 16,程序会输出( )。

A.16 B.-1 C. 4 D.1

8[2012].

```
#include <iostream>
using namespace std;
int n,i,ans;
int main()
{
    cin>>n;
    ans=0;
    for(i=1;i<=n;i++)
        if(n%i==0) ans++;//查找 1~n 中有多少 n 的因数
    cout<<ans<<endl;
    return 0;
```

```
}
```

●判断题

(1)该程序可能输出负数。 ( )

(2)可以将第 7 行的 i 初始值赋为 0。 ( )

(3)输入 18,输出结果为 6。 ( )

(4)该程序可以正常运行。 ( )

●选择题

(5)如果将第 8 行的 ans++改成 ans-- ,那么输入 18,输出结果为( )。

A.6 B.-6 C.man D.ans

(6)该算法的时间复杂度为( )。

A.O(n) B.O(1) C.$O(n^2)$ D.O(nlogn)

9[2012].

```
#include <iostream>
using namespace std;
int a,b,c,d,e,ans;
int main()
{
    cin>>a>>b>>c;
    d=a+b;
    e=b+c;
    ans=d+e;
    cout<<ans<<endl;
    return 0;
}
```

判断题

(1) 当输入为“11 45 14”时，输出为“70”. ( )

(2) 输入的数都为负数时，输出可能为正数。 ( )

(3) 输入的数都为正数时，输出可能为负数。 ( )

(4) 将 ans 的类型改为 char，输出结果不会改变。 ( )

选择题

(5) 输入为“19 19 810”时，输出为( ).

A.867 B.857 C.967 D.767

(6) 输出为“28”时，输入可以为( ).

A.5 8 7 B.1 5 7 C.1 1 25 D.1 1 4

10 [2011].

```
#include<iostream>
using namespace std;
```

```
int main()//题意：从 n~m 相加
{
    int i,n,m,ans;
    cin>>n>>m;
    i=n;
    ans=0;
    while(i<=m){
       ans+=i;
       i++;
    }
    cout<<ans<<endl;
    return 0;
}
```

判断题

(1)删去第 7 行,运行结果不变。 ( )

(2)将第 8 行的<=改为< ,输出减小 n。 ( )

(3)可以实现一个复杂度为 O(1)的代码,效果与上述代码等价。 ( )

(4)当 m<n 时,程序不会运行错误。 ( )

选择题

(5)输人 10 20,输出( )。

A.5 B.165 C.20 D.10

(6)时间复杂度为( )。

A. O(max{m- n,0}) B. O(n) C. $O(n^2)$ D. O(mlogn)

5.2 字符

1[2011].

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string map= "22233344455566677778889999";
    string tel;
    int i;
    cin>>tel;
    for(i=0;i<tel.length();i++)
       if((tel[i]>='0') && (tel[i]<='9') )//字符是数子就原样输出
           cout<<tel[i];
       else if( (tel[i]>='A') && (tel[i]<='Z'))//其他的就按照 map 对应输出
```

```
            cout<<map[tel[i]-'A'];
    cout<<endl;
    return 0;
}
```

●判断题

(1)第 7 行输入的子符串可以是任意字符,包括字母、数字、各类符号甚至中文汉字及符号。 ( )

(2)如果去掉第 13 行,程序无运行错误。 ( )

(3)输出结果可以包含大写字母。 ( )

(4)输出的字符串至多包含 8 种不同数字。 ( )

●选择题

(5)输入 CCF-NOIP-2011,输出的结果是( )。

A.22366472011 B.223-6647-2011 C. 22366472011 D.22366482011

(6)输入 WELCOME-CSP,输出的结果是( )。

A.9352663277 B.9342663277 C. 9342663278 D.9342663377

2[2014].

```
#include <iostream>
#include <string>
using namespace std;
int main()          (//题目大意：将字符串中的小写字母变为大写字母)
{
  string st;
  int i, len;
  getline(cin, st);
    len = st.size();
    for(i = 0; i < len; i++)
              if(st[i] >= 'a' && st[i] <= 'z')          //判断是否为小写字
母
         st[i] = st[i] - 'a' + 'A';              //是小写字母转化为大写字母
  cout << st << endl;
  return 0;
  }
```

判断题

(1)输入的字符串可以是任意字符,包括字母、数字、各类符号甚至中文汉字。 ( )

(2)如果去掉第 10 行,输出结果不变。 ( )

(3)输出结果可以包含小写字母。 ( )

(4 算法时间复杂度为 O(1)。 ( )

选择题

(5)输入 Hello,输出的结果是( )。

A. HELLO B. hello C. Hello D. ello

(6)输出的结果不可能是(　　)。

A.WELCOME　B.WELCOME-1　C. Welcome　D.ELCOME

3[2012].

```
#include <iostream>
#include <string>
using namespace std;
int n,i,j,ans;
string s;
char get(int i)
{
    if(i<n) return s[i];
    else return s[i-n];
}
int main()
{
    cin>>s;
    n=s.size();
    ans=0;
    for(i=1;i<=n-1;i++)
    {
        for(j=0;j<=n-1;j++)
            if(get(i+j)<get(ans+j))
            {
                ans=i;
                break;
            }
            else if(get(i+j)>get(ans+j)) break;
    }
    for(j=0;j<=n-1;j++) cout<<get(ans+j);
    cout<<endl;
    return 0;
}
```

●判断题

(1)删除第 15 行程序,运行结果不会发生变化。　(　　)

(2)将第 16 行中的 i=1 改为 i=0,运行结果不变。　(　　)

(3)在 17、18 行中间插入“if (get(i+i)==get(ans+i)) continue;”运行结果不变。(　　)

(4)将 21 行的 break 换成 continue,因为这是 for 语句的最后一句,所以运行结果不变。(　　)

●选择题

(5)当输入为 ABCDEFG 时,输出结果为(　　)。

A. ABCDEFG　B. GFEDCBA　C. ACEGFDB　D.AGBFCED

(6)当输入为 CBBADADA 时,输出结果为(　　)。

A.ABABCDAD　B. ABBCDADA　C .ACBBADAD　D. ADADACBB

4[2014].

```
#include <iostream>
using namespace std;
const int SIZE = 100;
int main()              //题目的含义：寻找 2~~n 以内的质数
{
        int p[SIZE];
        int n, tot, i, cn;
    tot = 0;
    cin >> n;
        for(i = 1; i <= n; i++)              //初始化为 1
        p[i] = 1;
        for(i = 2; i <= n; i++)
    {
                if(p[i] == 1)
       tot++;
     cn = i * 2;
     while(cn <= n)          //while 循环的作用是把当前
     {                       //是下标 i 的倍数的数变为 0;
          p[cn] = 0;    //如当 i=2 时，就把下标为 4、6、8、10 ~ ~26、28、30 的
          cn += i;      //数值变为 0
     }
  }
        cout << tot << endl;
   return 0;
}
```

●判断题

(1)将第 6 行去掉会导致运行错误。　　(　　)

(2)将第 4 行去掉会导致编译错误。　　(　　)

(3)若 a,b 都是 99 以内的整数,将第 8 行替换为“c1=i/10”没有影响。　　(　　)

(4)若 a.h 都是 99 以内的整数,则答案为区间[a,b]中 3 的倍数的数量。　　(　　)

●选择题

(5)输入 1100，则输出(　　)。

A21　B.32　C.33　D.34

(6)输入 110000，则输出(　　)。

A. 3333　B.3400　C.258　D.1950

5[2008].（字符串替换）给定一个字符串 S（S 仅包含大小写字母），下面的程序将 S 中的每个字母用规定的字母替换，并输出 S 经过替换后的结果。程序的输入是两个字符串，第一个字符串是给定的字符串 S，第二个字符串 S’由 26 个字母组成，它是 a-z 的任一排列，大小写不定，S’规定了每个字母对应的替换字母：S’中的第一个字母是字母 A 和 a 的替换字母，即 S 中的 A 用该字母的大写替换，S 中的 a 用该字母的小写替换；S’中的第二个字母是字母 B 和 b 的替换字母，即 S 中的 B 用该字母的大写替换，S 中的 b 用该字母的小写替换；…… 以此类推。

```
#include <iostream>
#include <string.h>
char change[26], str[5000];
using namespace std;
void CheckChangeRule()
{
    int i;
    for (i = 0;i < 26;i ++)
    {
        If(____①____)// change[i] >= 'A' && change[i] <= 'Z  全部转化为小写
              change[i] -= 'A' - 'a';
    }
}
void ChangeString()
{
    int i;
    for (i = 0;i <strlen(str);i ++)
    {
          if (__②___)// str[i] >= 'A' && str[i] <= 'Z' 是大写
                str[i] = change[str[i] - 'A'] -'a' + 'A';
          else //小写的变化
           ____③__ //str[i] = change[str[i] - 'a'];
    }
}
int main()
{
    int i;
    cin >> str ;
    cin >> change;
    CheckChangeRule();
    _____________④_____     // ChangeString();调用函数
    cout << str << endl;
    return 0;
```

}

●选择题

(1)①处应填(　　)。

A.change[i]>='a'&&change[i]<='z'　　B.change[i]<'A' ||change[i]>' Z'

C.change[i]>='A'&&change[i]<='Z'　　D.change[i]<' a' &&change[i]>'z'

(2)2 处应填(　　)。

A. str[i]>='a'&&str[i]<='z'　　B.str[i]<'A'&&str[i]>' Z'

C.str[i]<'a '&&str[i]>'z'　　D. str[i]>=' A'&&str[i]<=' Z'

(3)3 处应填(　　)。

A.change[str[i]-'a']=str[i];　　B.str[i]=change[str[i]- 'a'];

C.str[i]= change[str[i]-'A'];　　D.change[str[i]-'z']=str[i];

(4)4 处应填(　　)。

A. int len=strlen (str);　　B. Changestring();

C、ChangeString();　　D.changeString();

6.

```
#include <iostream>
using namespace std;
const int maxn=50;
void getnext(char str[])
{
  int l=strlen(str),i,j,k,temp;
  k=l-2;
  while(k>=0&&str[k]>str[k+1]) k--;
  i=k+1;
  while(i<l&&str[i]>str[k]) i++;
  temp=str[k];
  str[k]=str[i-1];
  str[i-1]=temp;
  for(i=l-1;i>k;i--)
      for(j=k+1;j<i;j++)
          if(str[j]>str[j+1])
          {
              temp=str[j];
              str[j]=str[j+1];
              str[j+1]=temp;
          }
  return ;
}
int main()
```

```
{
  char a[maxn];
  int n;
  cin >> a >> n;
  while(n>0)
  {
      getnext(a);
      n--;
  }
  cout << a << endl;
  return 0;
}
```

●判断题

(1)每次循环找到的 k 一定不同。 ( )

(2)如果把第 18～20 行换成 swap(str[j],str[j+1]),输出结果不变。 ( )

(3)去掉第 32 行会导致程序超时。 ( )

(4)输入的字符串可包含任意字母。 ( )

●选择题

(5)输入 NOIP3,输出的结果是( )。

A. NPOI B. NPIO C. NIPO D. PONI

(6)输入 CSP2021 2,输出的结果是( )。

A. CPS2120 B. CSP2120

C. CPS2210 D. CSP2021

7[2010].

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string s;
    char m1, m2;
    int i;//Expo 2010 Shanghai China
    getline(cin, s);//输出字符串中 ASCII 码最大的两个字符的码值
    m1 = ' ';
    m2 = ' ';
    for (i = 0; i < s.length(); i++)
        if (s[i] > m1) {
            m2 = m1;
            m1 = s[i];
        }
```

```
        else if (s[i] > m2)
            m2 = s[i];
    cout<<int(m1)<<' '<<int(m2)<<endl;
    return 0;
}
```

判断题

(1)去掉第 10、11 行,程序会运行错误。 ( )

(2)去掉第 18 行的 else,程序会运行错误。 ( )

(3)将 12 行替换为“for (i=s.length() -1;i>=0;i--)”,运行结果不变。

( )

(4)将 18 行的“s[i]>m2”替换为“s[i]>=m2”,运行结果不变。 ( )

●选择题

(5)当输入为 AABBCCDD 时,输出为( )。

A. B A B.C B

C.D D D.D A

(6)当输入的字符串为大写字符串时，输出不可能为( )。

A.Z B B.A B

C.T B D.G A

提示：

字符	空格	'0'	'A'	'a'
ASCII 码	32	48	65	97

8[2011]. (大整数开方) 输入一个正整数 n（1≤n≤10100)，试用二分法计算它的平方根的整数部分。

```
#include<iostream>
#include<string>
using namespace std;
const int SIZE=200;
struct hugeint{
    int len,num[SIZE];
};
//其中 len 表示大整数的位数；num[1]表示个位，num[2]表示十位，以此类推
hugeint times(hugeint a,hugeint b)
// 计算大整数 a 和 b 的乘积
{
    int i,j;
    hugeint ans;
    memset(ans.num,0,sizeof(ans.num));
    for(i=1;i<=a.len;i++)
       for(j=1;j<=b.len;j++)
```

```
            ①          +=a.num[i]*b.num[j]; // ans.num[i+j-1]    大整数乘法
    for(i=1;i<=a.len+b.len;i++){//进位
        ans.num[i+1]+=ans.num[i]/10;
            ②          ; // ans.num[i]%=10
    }
    if(ans.num[a.len+b.len]>0)
        ans.len=a.len+b.len;
    else
        ans.len=a.len+b.len-1;
    return ans;
}
hugeint add(hugeint a,hugeint b)
//计算大整数 a 和 b 的和
{
    int i;
    hugeint ans;
    memset(ans.num,0,sizeof(ans.num));
    if(a.len>b.len)
        ans.len=a.len;
    else
        ans.len=b.len;
    for(i=1;i<=ans.len;i++){
        ans.num[i]+=          ③          ;// a.num[i]+b.num[i] 求和
        ans.num[i+1]+= ans.num[i]/10;
        ans.num[i]%=10;
    }
    if(ans.num[ans.len+1]>0)
        ans.len++;
    return ans;
}
hugeint average(hugeint a,hugeint b)
//计算大整数 a 和 b 的平均数的整数部分
{
    int i;
    hugeint ans;
    ans=add(a,b);
    for(i=ans.len;i>=2;i--){
        ans.num[i-1]+=(      ④       )*10; // ans.num[i] % 2
        ans.num[i]/=2;
    }
    ans.num[1]/=2;
```

```
    if(ans.num[ans.len]==0)
        ans.len--;
    return ans;
}
hugeint plustwo(hugeint a)
// 计算大整数 a 加 2 之后的结果
{
    int i;
    hugeint ans;
    ans=a;
    ans.num[1]+=2;
    i=1;
    while( (i<=ans.len)&&(ans.num[i]>=10) ){
        ans.num[i+1]+=ans.num[i]/10;
        ans.num[i]%=10;
        i++;
    }
    if(ans.num[ans.len+1]>0)
    _____⑤_____ ; //ans.len++
    return ans;
}
bool over(hugeint a,hugeint b)
// 若大整数 a>b 则返回 true，否则返回 false
{
    int i;
    if(_____⑥_____)  // a.len<b.len 大整数 a>b 则返回 true，否则返回 false
        return false;
    if( a.len>b.len )
        return true;
    for(i=a.len;i>=1;i--){
        if(a.num[i]<b.num[i])
           return false;
        if(a.num[i]>b.num[i])
           return true;
    }
    return false;
}
int main()
{
    string s;
    int i;
```

```
    hugeint target,left,middle,right;
    cin>>s;
    memset(target.num,0,sizeof(target.num));
    target.len=s.length();
    for(i=1;i<=target.len;i++)
        target.num[i]=s[target.len-i]-___⑦___;//将数字字符变为数字   '0'
    memset(left.num,0,sizeof(left.num));
    left.len=1;
    left.num[1]=1;
    right=target;
    do{
        middle=average(left,right);
        if(over( _____⑧_____))// times(middle,middle),target
            right=middle;
        else
            left=middle;
    }while(!over(plustwo(left),right) );
    for(i=left.len;i>=1;i--)
       cout<<left.num[i];
    return 0;
}
```

选择题

(1)1 处应填(　　)。

A. ans.num[j-i+1]　　B. ans.num[i+j-1]

C.ans.num[i+j+1]　　D. ans.num[i-j+1]

(2)2 处应填(　　)。

A.ans.num[i]=ans.num[i-1]%10　　B.ans.num[i]=ans.num[i]/10

C. ans.num[i]=ans.num[i]%10　　D.ans.num[i]=ans.num[i-1]*10

(3)3 处应填(　　)。

A.ans.num[i]+=a.num[i]+b.num[i]

B.ans.num[i]=abs(a.num[i]-b.num[i])

C.ans.num[i]=min(a.num[i],b.num[i])

D. ans.num[i]=a.num[i]+b.num[i]

(4)4 处应填(　　)。

A. ans.num[i]|1　　B. ans.num[i]&2

C. ans.num[i- 1]&1　　D. ans.num[i]&1

(5)5 处应填(　　)。

A. ans.num[ans. len+1]%10　　B.ans.len++

C. ans.len--　　D.ans.num[ans.len++]=1

(6)6 处应填(　　)。

A.times(a,a)<times(a, b)　　B.a .len<=b.len

C.a. len<b.len　　D.add(a,a)<add(b,b)

(7)⑦处应填(　　)。

A.65　　B.97

C.34　　D.48

(8)⑧处应填(　　)。

A. times(left,right) , target　　B.times (middle+1,middle+1),target

C.times (middle,middle),target　　D. add (middle, middle) ,target

5.3 枚举

1[2009].

```
#include <iostream>
using namespace std;
const int c=2009; //题意：求 p^(n+1)/2 % c 的余数
int main()
{
  int n,p,s,i,j,t;
  cin >> n >> p;
  s=0;t=1;
  for(i=1;i<=n;i++)
  {
      t=t*p%c;
      for(j=1;j<=i;j++)
          s=(s+t)%c;
  }
  cout << s << endl;
  return 0;
}
```

判断题

(1)将 12 13 行改为 s=(s+111*t*i)% c; ,程序输出不会改变。　　(　　)

(2)将 15 行改为 printf(“%d\n",s);程序输出不会改变。　　(　　)

(3)将 08 行的 s=0;去掉，程序输出不会改变。　　(　　)

(4)将 03 行 const 去掉，程序输出不会发生变化。　　(　　)

●选择题

(5)输入为“11 2”.输出为(　　).

A.782　　B.762　　C.802　　D.114

(6)该算法的时间复杂度为(　　)。

A. O(1)　　B. O(n)　　C. $O(n^2)$　　D. O(nlogn)

2[2011].

```
#include<iostream>
#include<cstring>
using namespace std;
const int SIZE = 100;
int main()
{
    int n,i,sum,x,a[SIZE];
    cin>>n;
    memset(a,0,sizeof(a));
    for(i=1;i<=n;i++){
        cin>>x;
        a[x]++;//记录数 x 出现的次数
    }
    i=0;
    sum=0;
    while(sum<(n/2+1)){//统计在 1~i 数出现的次数大于 n/2+1
        i++;
        sum+=a[i];
    }
    cout<<i<<endl;
    return 0;
}
```

判断题

(1)当第 8 行的 sizeof(a)改为 SIZE 时,运行结果不会发生改变。 ( )

(2)当第 11 行的 a[x]++改成++a[x]时,运行结果不会发生改变。 ( )

(3)当第 14 行的 sum<(n/2+1)改为 sum<(n+1)/2 时,运行结果不会发生改变。( )

(4)数组 a 的所有数中的最大值为 1。 ( )

●选择题

(5)输入 11 4 5 6 6 4 3 3 2 3 2 1,输出的结果为( )。

A.3 B.4 C.5 D.error

(6)输入 5 1 2 3 4 5,输出的结果为( )。

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

3[2008].

```
#include <iostream>
using namespace std;//题意： 把正数和负数分开

void func(int ary[], int n )
{
    int i=0, j, x;
```

```
    j=n-1;
    while(i<j)
    {
        while (i<j&&ary[i]>0) i++;//记录从左至右第一个负数的下标 i
        while (i<j&&ary[j]<0) j--;//记录从右至左第一个负数的下标 j
        if (i<j){//如果 i<j 交换值
            x=ary[i];
            ary[i++]=ary[j];
            ary[j--]=x;
        }
    }
}
int main()
{

    int a[20], i, m;
    m=10;
    for(i=0; i<m; i++)
    {
        cin>>a[i];
    }
    func(a, m);
    for (i=0; i<m; i++)
        cout<<a[i]<<" ";
    cout<< endl;
    return 0;
}
```

●判断题

(1)将第 13 行的 ary[i++]=ary[j]改为 ary[++i]=ary[j],程序输出结果不变。

(　　)

(2)输入 5 4 -6 -11 6 -59 22 -6 1 10 输出结果为 5 4 10 16 22 -59 -11 -6 -6。

(　　)

(3)将 func 函数中的所有变量 n 全部变成 a，程序能通过编译且输出结果不变。(　　)

●选择题

(4)该代码时间复杂度为(　　)。

A.O(nlogn)　　B.$O(n^2)$　　C.O(n)　　D.O(logn)

4[2014].　(数字删除)下面程序的功能是将字符串中的数字字符删除后输出。请填空。

```
    #include  <iostream>
  using  namespace  std;
  int  delnum(char  *s)
```

```
    {
          int  i,  j;
      j  =  0;
          for(i  =  0;  s[i]  !=  '\0';  i++)
              if(s[i]  <  '0'  ___①___s[i]  >  '9')                    //  ||如果不
是数字就把这个字符存入数组中
        {
                  s[j]  =  s[i];
              _____②_____;                          //  j++  更新下标
          }
      return  ___③___;                        //  j   返回 s 数组的字符长度
  }
  const  int  SIZE  =  30;
  int  main()
  {
          char  s[SIZE];
    int  len,  i;
          cin.getline(s,  sizeof(s));
     len  =  delnum(s);
          for(i  =  0;  i  <  len;  i++)
    cout  <<___④___;                //s[i]  输出
    cout  <<  endl;
    return  0;
    }
```

(1)1 处应填(　　)。

A.&&　　B.&&s[i]>='a'&&s[i]<='z'&&

C.||　　D.&&s[i]>='A' &&s[i]<='Z'&&

(2)2 处应填(　　)。

A.i++　　B.i=j

C.++j　　Dj=i

(3)3 处应填(　　)。

A.j　　B.i

C.s[j]　　D.s[i]

(4)4 处应填(　　)。

A.i　　B.s[i]

C.(!s[i])　　D.(s[i]==1)

5[2012].（坐标统计）输入 n 个整点在平面上的坐标。对于每个点，可以控制所有位于它左下方的点（即 x、y 坐标都比它小），它可以控制的点的数目称为“战斗力”。依次输出每个点的战斗力，最后输出战斗力最高的点的编号（如果若干个点的战斗力并列最高，输出其中最大的编

号)。

```
#include <iostream>
using namespace std;
const int SIZE =100;
int x[SIZE],y[SIZE],f[SIZE];
int n,i,j,max_f,ans;
int main()
{
    cin>>n;
    for(i=1;i<=n;i++) cin>>x[i]>>y[i];
    max_f=0;
    for(i=1;i<=n;i++)
    {
        f[i]=___①___;  //初始化战斗力为0
        for(j=1;j<=n;j++)
        {
            if(x[j]<x[i] &&___②___ )//如果在左下方 y[j]<y[i]
            _____③_____;//战斗力加1  f[i]=f[i]+1
        }
        if(_____④_____ )// (i>1)&& (f[i]>f[i-1])  判断最高战斗力
        {
            max_f=f[i];
            ____⑤____; //ans=max_f 保存目前最高战斗力
        }
    }
    for(i=1;i<=n;i++) cout<<f[i]<<endl;
    cout<<ans<<endl;
    return 0;
}
```

●选择题

(1)①处应填(　　)。

A.i　　B.0

C.x[i]　　D. y[i]

(2)2 处应填(　　).

A.y[i]<y[j]　　B.y[j]<y[i]

C.y[i]<=y[j]　　D.y[j]<=y[i]

(3)3 处应填(　　)。

A. f[i]=x[i]　　B.f[i]=0

C.f[i]++　　D.f[i]=y[i]

(4)4 处应填(　　)。

A.f[i]>max _f　　B.f[i]<max_f

C.f[i]>=max_ f　　D. f[i]<=max _f

(5)5 处应填(　　)。

A.ans=i　　B.ans<max_f

C.ans=f[i]　　D.ans=0

6[2013]

```
#include <iostream>
using namespace std;
int main()//题意：查找数 f 在数组 a 中的位置
{
    const int SIZE = 100;
    int n, f, i, left, right, middle, a[SIZE];
    cin>>n>>f;
    for (i = 1; i <= n; i++)
        cin>>a[i]; left = 1;
    right = n;
    do {
        middle = (left + right) / 2;
        if (f <= a[middle])
            right = middle;
        else
            left = middle + 1;
            } while (left < right);
    cout<<left<<endl;
    return 0;
}
```

判断题

(1)将第 04 行的程序移动到 02、03 行的中间,程序能够正常运行。　　(　　)

(2)将第 11 行的“=”删除,运行结果会改变。　　(　　)

(3)将第 07 行的“i=1;i<=n;”改为“i=0;i<n;”运行结果不会改变。　　(　　)

(4)若第 08 行输入 n 个相同的数字,程序最后输出的 left 值为 1。　　(　　)

●选择题

(5)当 n=5,f=7,a={8,4,7,5,6}时,则结果为(　　)。

A.3　B.4　C.5　D.7

(6)输入仍是第(5),将第 11 行的“>=”改为“<=”,则结果为(　　)。

A.3　B.4　C. 5　D.7

7[2012].　（排列数）输入两个正整数 n，m（1<n<20,1<m<n），在 1~n 中任取 m 个数，按字典序从小到大输出所有这样的排列。例如：

输入：3 2

输出：1 2

1 3
2 1
2 3
3 1
3 2

```
#include <iostream>
#include <cstring>
using namespace std;
const int SIZE =25;
bool used[SIZE];
int data[SIZE];
int n,m,i,j,k;
bool flag;
int main()
{
    cin>>n>>m;
    memset(used,false,sizeof(used));
    for(i=1;i<=m;i++)
    {
        data[i]=i;
        used[i]=true;
    }
    flag=true;
    while(flag)
    {
        for(i=1;i<=m-1;i++) cout<<data[i]<<" ";
        cout<<data[m]<<endl;
        flag=____①____;//flag = 0
        for(i=m;i>=1;i--)
        {
            ____②____;/。/标记当前 date[i]这个数用过 used[data[i]]=0
            for(j=data[i]+1;j<=n;j++)
                if(!used[j])
                {
                    used[j]=true;//标记 j 这个数用过
                    data[i]=____③____;//下一个数填 j
                    flag=true;
                    break;
                }
            if(flag)
            {
```

```
                for(k=i+1;k<=m;k++)
                    for(j=1;j<=____④____;j++)j<=n
                    if(!used[j])
                    {
                        data[k]=j;
                        used[j]=true;
                        break;
                    }
                ____⑤____;//break
            }
        }
    }
    return 0;
}
```

(1)①处应填(　　)。

A.0　　B.1

C.!flag　　D.used[m]

(2)2 处应填(　　).

A.used[data[i]]=false　　B. used[i]=false

C. used[data[i]]=true　　D. used[i]=true

(3) 3 处应填(　　)。

A.!flag　　B. data[j]

C.j　　D.flag

(4) ④处应填(　　)。

A.n+m　　B. k

C.n　　D.m

(5)⑤处应填(　　)。

A. return 0　　B. continue

C.!flag　　D. break

8[2010].（哥德巴赫猜想）哥德巴赫猜想是指，任一大于 2 的偶数都可写成两个质数之和。迄今为止，这仍然是一个著名的世界难题，被誉为数学王冠上的明珠。试编写程序，验证任一大于 2 且不超过 n 的偶数都能写成两个质数之和。

```
#include <iostream>
using namespace std;
int main()
{
    const int SIZE = 1000;
    int n, r, p[SIZE], i, j, k, ans;
    bool tmp;
    cin>>n;
    r = 1;
```

```
    p[1] = 2;
    for (i = 3; i <= n; i++) {//求质数数组
        ____①____;//tmp = 1 ；假设 i 为质数
        for (j = 1; j <= r; j++)
            if (i %____②____ == 0) {//i%p[j]==0  不满足质数的条件
                tmp = false;
                break;
            }
        if (tmp) {//如果上面 for 循环没有改变 tem 的值，那么 i 就是质数
            r++;
            ____③____;p[r]=i;
        }
    }
    ans = 0;
    for (i = 2; i <= n / 2; i++) {
        tmp = false;
        for (j = 1; j <= r; j++)
            for (k = j; k <= r; k++)
                if (i + i ==____④____) {//i+i == p[j]+p[k]
                tmp = true;// 判断大于 2 且不超过 n 的偶数都能写成两个质数之和
                    break;
                }
        if (tmp)
            ans++;
    }
    cout<<ans<<endl;
    return 0;
}
```

若输入 n 为 2010，则输出____⑤____时表示验证成功，即大于 2 且不超过 2010 的偶数都满足哥德巴赫猜想。// 1004

●选择题

(1)①处应填(　　)。

A.k=0　　B.p[i]=i

C. tmp=true　　D.p[i]=r

(2)②处应填(　　)。

A. p[j]　　B.p[n]

C. n　　D.j

(3)3 处应填(　　)。

A. p[r]=i　　B.p[j]=i

C.tmp=true　　D.ans++

(4)④处应填(　　)。

A.j+k　　B. p[j]+p[k]

C.p[j]　　D. p[k]

(5⑤处应填(　　)。

A.2010　　B.1005

C.2009　　D.1004

9[2010].（过河问题）在一个月黑风高的夜晚，有一群人在河的右岸，想通过唯一的一根独木桥走到河的左岸。在这伸手不见五指的黑夜里，过桥时必须借助灯光来照明，很不幸的是，他们只有一盏灯。另外，独木桥上最多承受两个人同时经过，否则将会坍塌。每个人单独过桥都需要一定的时间，不同的人需要的时间可能不同。两个人一起过桥时，由于只有一盏灯，所以需要的时间是较慢的那个人单独过桥时所花的时间。现输入 n（2≤n<100）和这 n 个人单独过桥时需要的时间，请计算总共最少需要多少时间，他们才能全部到达河的左岸。

例如，有 3 个人甲、乙、丙，他们单独过桥的时间分别为 1、2、4，则总共最少需要的时间为 7。具体方法是：甲、乙一起过桥到河的左岸，甲单独回到河的右岸将灯带回，然后甲、丙再一起过桥到河的左岸，总时间为 2+1+4=7。

```
#include <iostream>
using namespace std;
const int SIZE = 100;
const int INFINITY = 10000;
const bool LEFT = true;
const bool RIGHT = false;
const bool LEFT_TO_RIGHT = true;
const bool RIGHT_TO_LEFT = false;
int n, hour[SIZE];
bool pos[SIZE];
int max(int a, int b)
{
    if (a > b)
        return a;
    else
        return b;
}
int go(bool stage)
{
    int i, j, num, tmp, ans;
    if (stage == RIGHT_TO_LEFT) {//右到左
        num = 0;
        ans = 0;
        for (i = 1; i <= n; i++) //判断在右边的人数 ，如果人数小于等于两人，记录两个人中的最大过河时间
            if (pos[i] == RIGHT) {
            if (pos[i] == RIGHT) {
                num++;
                if (hour[i] > ans)
                    ans = hour[i];
            }
```

```
        if (    ①    )//如果人数小于等于 2 返回两个之间的最大值 num<=2
            return ans;
        ans = INFINITY;
        for (i = 1; i <= n - 1; i++)
            if (pos[i] == RIGHT)
                for (j = i + 1; j <= n; j++)
                    if (pos[j] == RIGHT) {
                        pos[i] = LEFT;
                        pos[j] = LEFT;
                        tmp = max(hour[i], hour[j]) +   ②    ;// go(true) 加上
                        if (tmp < ans) //回来的时间
                           ans = tmp;
                        pos[i] = RIGHT;
                        pos[j] = RIGHT;
                    }
        return ans;
    }
    if (stage == LEFT_TO_RIGHT) {//左到右
        ans = INFINITY;
        for (i = 1; i <= n; i++)
            if (    ③    ) {//判断当前这个人是不是在左边 pos[i]==true
                pos[i] = RIGHT;
                tmp =    ④    ;//消耗的时间 hour[i]+go(false)
                if (tmp < ans)
                    ans = tmp;
                 ⑤    ;//复位（因为要遍历每一种情况）pos[i]=true;
            }
        return ans;
    }
    return 0;
}
int main()
{
    int i;
    cin>>n;
    for (i = 1; i <=n; i++) {
        cin>>hour[i];
        pos[i] = RIGHT;
    }
    cout<<go(RIGHT_TO_LEFT)<<endl;
    return 0;
```

```
}
```

●选择题

(1)1 处应填(　　)。

A.num<=0　　B.num<=1

C.num<=2　　D.num<=3

(2)②处应填(　　)。

A. go(LEFT)　　B.go(RIGHT)

C. go(LEFT_TO_RIGHT)　　D.go(RIGHT_TO_LEFT)

(3)3 处应填(　　) 。

A. pos[i]=LEFT　　B. pos[i]=RIGHT

C.pos[i]=LEFT_TO_RIGHT　　D.pos[i]=RIGHT_TO_LEFT

(4)4 处应填(　　)。

A.hour[i]+go(RIGHT_TO_ LEFT)　　B.hour[i]-go(RIGHT_ TO_ LEFT)

C.hour[i]+go(LEFT_TO_RIGHT)　　D.hour[i]-go(LEFT_TO_RIGHT)

(5)⑤处应填(　　)。

A.pos[i]=LEFT　　B. pos[i]=RIGHT

C. pos[i]=LEFT_TO_RIGHT　　D. pos[i]=RIGHT_TO_LEFT

10.[2013]　（序列重排） 全局数组变量 a 定义如下：

```
const int SIZE = 100;
int a[SIZE], n;
```

它记录着一个长度为 n 的序列 a[1], a[2], … , a[n]。

现在需要一个函数，以整数 p (1 ≤p ≤n) 为参数，实现如下功能：将序列 a 的前 p 个数与后 n －p 个数对调，且不改变这 p 个数（或 n －p 个数)之间的相对位置。例如，长度为 5 的序列 1, 2, 3, 4, 5，当 p = 2 时重排结果为 3, 4, 5, 1, 2 。

有一种朴素的算法可以实现这一需求，其时间复杂度为 O(n)、空间复杂度为 O(n)：

```
void swap1(int p)
{
    int i, j, b[SIZE];
    for (i = 1; i <= p; i++)     //前 p 个以到后面
        b[  (1)  ] = a[i];       //  n －p + i
    for (i = p + 1; i <= n; i++)     //后面 n-p 个移到前面
        b[i - p] =  (2) ;   //  a[i]
    for (i = 1; i <=    (3) ; i++)  //n  重新赋值给数组 a
        a[i] = b[i];
}
```

我们也可以用时间换空间，使用时间复杂度为 O(n2)、空间复杂度为 O(1) 的算法：

```
void swap2(int p)
{
    int i, j, temp;
    for (i = p + 1; i <= n; i++)
```

```
    {
        temp = a[i];
        for (j = i; j >=    (4) ; j--)  // i-p+1
            a[j] = a[j - 1];
        (5) = temp; // a[i-p]
 }
}
```

(1)①处应填(　　)。
A. p+i　　B.n-p+i
C.n-p+i-1　　D.i
(2)2 处应填(　　)。
A.a[i]　　B.a[i-p]
C.a[n-i]　　D.b[i]
(3)3 处应填(　　)。
A.n　　B. p
C. p+1　　D.n-1
(4)4 处应填(　　)。
A. 1　　C.n-i
B.i-p　　D.i-p+1
(5)⑤处应填(　　)。
A.a[i-p+1]　B.a[i]
C.a[i-p]　　D.a[i-1]

5.4 递归

1[2008].

```
#include<iostream>
using namespace std;
void foo(int a, int b, int c)// 3 1 2->2 3 1
{
    if(a > b)
        foo(c, a, b);
    else
        cout<<a<<','<<b<<','<<c<<endl;
}
int main()
{
    int a, b, c;
    cin >> a >> b >> c;//3  1 2
    foo(a, b, c);
```

```
    return 0;
}
```

●判断题

(1)该程序只输出一行。 ( )

(2)如果输入的三个数都相同,程序会运行错误。 ( )

(3)如果输入 312,输出 2,3,。 ( )

(4)如果输入 321,程序会超时。 ( )

●选择题

(5)输入 9 10 8,输出为( )。

A.9,10,8 B.9,8,10 C.10,8,9 D.8,9,10

(6)令 n 代表输入变量 a、b、c 的次数,则 n=3 该程序的时间复杂度为( )。

A.O(n) B.O(logn) C.O(√logn) D.O(log(n√n))

2[2014].

```
#include <iostream>
using namespace std;
int fun(int n)
{
  if(n == 1)
    return 1;
  if(n == 2)
    return 2;
  return fun(n -2) - fun(n - 1);
  }
int main()
{
  int n;
  cin >> n;
  cout << fun(n) << endl;
    return 0;
  }
```

●判断题

(1)输入 114514 时在普通计算机上程序运行时间不会超过 1s。 ( )

(2)输入 0 程序不会出现运行错误。 ( )

(3)该程序开启 O2 不会出现错误。 ( )

(4)输入 6,输出 7。 ( )

●选择题

(5)时间复杂度为( )。

A.$O(2^n)$ B .$O(n^n)$

C.$O(n^{logn})$ D.O(n)

(6)输入 7 时输出(　　)。

A.-11　B. 11

C.-10　D.10

3[2009].

```
#include <iostream>
using namespace std;//题意：求最大约数
int a,b;
int work(int a,int b){
  if (a%b)
      return work(b,a%b);
  return b;
}
int main(){
  cin >> a >> b;
  cout << work(a,b) << endl;
  return 0;
}
```

判断题

(1)在第 2 行下面添加# define int long long 程序可以正常运行。(　　)

(2)将第 3 行的 int 改成 double 结果不会改变。(　　)

(3)不能输入 0 0。(　　)

(4)输入 20 12 结果输出 4。(　　)

●选择题

(5)输入 2012 13,输出(　　)。

A.2012　B.2013　C. 13　D.1

(6)该算法的时间复杂度级别为(　　)。

A.线性时间　B.对数时间　C.平方时间　D.常数时间

4[2010].

```
#include <iostream>
using namespace std;
const int NUM = 5;
int r(int n)
{
    int i;
    if (n <= NUM)
        return n;
    for (i = 1; i <= NUM; i++)
        if (r(n - i) < 0)
```

```
            return i;
    return -1;
}
int main()
{
    int n;
    cin>>n;
    cout<<r(n)<<endl;
    return 0;
}
```

●判断题

(1)将第 4 行的 int 改为 unsigned,答案不会错误。 ( )

(2)程序开启 O2 优化不会返回错误。 ( )

(3)如果输入-1,程序会输出-1。 ( )

(4)该问题 r(n)的值没有规律。 ( )

●选择题

(5)如果输入 7,程序会输出( )。

A.-1 B.5

C. 1 D.3

(6)如果输入 16,程序会输出( )。

A.16 B.-1

C. 4 D.1

5[2011].

```
#include<iostream>
using namespace std;
int solve(int n,int m)
{
    int i,sum;
    if(m==1) return 1;
    sum=0;
    for(i=1;i<n;i++)
       sum+= solve(i,m-1);
    return sum;
}
int main()
{
    int n,m;
    cin>>n>>m;
    cout<<solve(n,m)<<endl;
```

```
    return 0;
}
```

判断题

(1)将第 7 行<改成<=程序会出现运行错误。 ( )

(2)本题使用 C++98 编译不会出现编译错误。 ( )

(3)本题输入 0 0 不会出现运行错误。 ( )

(4)本题不可能输出 0。 ( )

选择题

(5)如果输入 7 4,输出( )。

A.11 B.20 C.21 D.12

(6)如果输入 100,输出( )。

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

6[2012].

```
#include <iostream>
using namespace std;
int n,i,j,a[100][100];
int solve(int x,int y)
{
    int u,v;
    if(x==n) return a[x][y];
    u=solve(x+1,y);
    v=solve(x+1,y+1);
    if(u>v) return a[x][y]+u;
    else return a[x][y]+v;
}
int main()
{
    cin>>n;
    for(i=1;i<=n;i++)
        for(j=1;j<=i;j++) cin>>a[i][j];
    cout<<solve(1,1)<<endl;
    return 0;
}
```

●判断题

(1)该程序读入 n+n*n 个整数。 ( )

(2)该程序能正常运行。 ( )

(3)输出结果均为整数。 ( )

(4)该程序有多组测试。 ( )

●选择题

(5)若输入为:

5

2

-1 4

2 -1 -2

-1 6 4 0

3 2 -1 5 8

则结果是(　　)。

A.11　B.12　C.13　D.14

7[2008]. (找第 k 大的数) 给定一个长度为 1,000,000 的无序正整数序列, 以及另一个数 n (1<=n<=1000000), 然后以类似快速排序的方法找到序列中第 n 大的数(关于第 n 大的数：例如序列{1，2，3，4，5，6}中第 3 大的数是 4)。

```
#include <iostream>
using namespace std;
int a[1000001],n,ans = -1;
void swap(int &a,int &b)//交换函数
{
    int c;
    c = a; a = b;   b = c;
}
int FindKth(int left, int right, int n)
{
    int tmp,value,i,j;
    if (left == right) return left;  //结束边界
    tmp = rand()% (right - left) + left;// rand 函数，C 语言中用来产生一个随机数的函数。
    swap(a[tmp],a[left]);
    value =  ①    // a[left]
    i = left;
    j = right;
    while (i < j)
    {
        while (i < j &&  ②    ) j --;// a[j] < value
        if (i < j) {a[i] = a[j]; i ++;} else break;
        while (i < j &&  ③  ) i ++;// a[i]  >  value
        if (i < j) {a[j] = a[i]; j - -;} else break;
    }
    ④  // a[i]  =  value;
    if (i < n) return  FindKth( ⑤ );// FindKth( i  +  1,right,n)
    if (i > n) return ⑥ // FindKth(left,  i  -  1,  n);
```

```
    return i;
}
int main()
{
    int i;
    int m = 1000000;
    for (i = 1;i <= m;i ++)
        cin >> a[i];
    cin >> n;
    ans = FindKth(1,m,n);
    cout << a[ans];
    return 0;
}
```

●选择题

(1)1 处应填(　　)。

A. a[left]　　B.a[tmp];　　C. a[right];　　D.a[n];

(2)2 处应填(　　)。

A. a[j]<value　　B. a[j]>value　　C.a[j]<a[i]　　D.a[j]>a[i]

(3)3 处应填(　　)。

A.a[i]<value　　B.a[i]>value　　C.a[i]<a[j]　　D.a[i]>a[j]

(4)④处应填(　　)。

A. a[i]=a[left];　　B.a[i]=a[right];　　C.a[i]=a[tmp];　　D.a[i]=value;

(5)⑤处应填(　　)。

A. i,right, n　　B.i+1, m,n　　C.i+1,right, n　　D.i, m,n

(6)6 处应填(　　)。

A. FindKth(1,i-1,n);　　B.FindKth(left, i-1,n)　C. FindKth(1,i,n);　　D. FindKth(left,i,n);

5.5 递推(DP)

1[2009].（最大连续子段和）给出一个数列（元素个数不多于 100），数列元素均为负整数、正整数、0。请找出数列中的一个连续子数列，使得这个子数列中包含的所有元素之和最大，在和最大的前提下还要求该子数列包含的元素个数最多，并输出这个最大和以及该连续子数列中元素的个数。例如数列为 4，-5，3，2，4 时，输出 9 和 3；数列为 1 2 3 -5 0 7 8 时，输出 16 和 7。

```
#include <iostream>
using namespace std;
int a[101];
int n,i,ans,len,tmp,beg;

int main(){
  cin >> n;
```

```
    for (i=1;i<=n;i++)
        cin >> a[i];
    tmp=0;
    ans=0;
    len=0;
    beg=____①____;//beg = 0  初始赋值为0，因为下标从0 开始
    for (i=1;i<=n;i++){
        if (tmp+a[i]>ans){
            ans=tmp+a[i];
            len=i-beg;
        }
        else if (___②___&&i-beg>len)// tmp+a[i]==ans
            len=i-beg;
        if (tmp+a[i]___③___ ){// tmp+a[i]<0 小于0 重新寻找
            beg=____④____;// beg = i
            tmp=0;
        }
        else
            ___⑤_____  ;// tmp+=a[i] 累加
    }
    cout << ans << " " << len << endl;
    return 0;
  }
```

选择题

（1)①处应填(　　)。

A.1　　B.0

C.n　　D. 101

(2)2 处应填(　　)。

A. tmp+a[i]>ans　　B.tmp+a[i]=ans

C. tmp+a[i]>n　　D. tmp+a[i]==ans

(3)③处应填(　　)。

A.<0　　B. <ans

C.<mp　　D.<min(0,ans, tmp)

(4)4 处应填(　　)。

A.1　　B.0

C.n　　D.i

(5)⑤处应填(　　)。

A. tmp+ =a[i]　　B. tmp= i+1

C.beg= i　　D.beg= i+1

2[2014].　.(最大子矩阵和)给出 m 行 n 列的整数矩阵，求最大的子矩阵和(子矩阵不能为

空)。

输入第一行包含两个整数 m 和 n，即矩阵的行数和列数。之后 m 行，每行 n 个整数，描述整个矩阵。程序最终输出最大的子矩阵和。(最后一空 4 分，其余 3 分，共 16 分)

比如在如下这个矩阵中：　　4　4

```
0  -2  -7  0
9  2  -6  2
-4  1  -4  1
-1  8  0  -2
```

拥有最大和的子矩阵为：

```
 9  2
-4  1
-1  8
```

其和为 15

```
3   3
-2  10  20
-1  100  -2
0  -2  -3
```

最大子矩阵和为 128

```
4   4
0  -2  -9  -9
-9  11  5  7
-4  -3  -7  -6
-1   7   7   5
```

最大子矩阵和为 26

```
#include <iostream>
using namespace std;
const int SIZE = 100;
int matrix[SIZE + 1][SIZE + 1];
int rowsum[SIZE + 1][SIZE + 1]; //rowsum[i][j]记录第 i 行前 j 个数的和
int m, n, i, j, first, last, area, ans;
int main()
 {
      cin >> m >> n;
      for(i = 1; i <= m; i++)
    for(j = 1; j <= n; j++)
       cin >> matrix[i][j];
 ans = matrix___①___;
      for(i = 1; i <= m; i ++)
      ___②___
 for(i = 1; i <= m; i++)
    for(j = 1; j <= n; j++)
```

```
                    rowsum[i][j] = ____③____;
    for(first = 1; first <= n; first++)                //开始遍历
      for(last = first; last <= n; last++)
      {
            ____④____;
          for(i = 1; i <= m; i++)
          {
              area += ___⑤___;
            if(area > ans)
            ans=area;//记录当前已知的最大子矩和
          if(area < 0)    //如果 area<0，舍弃这一行，重新寻找
                area = 0;
        }
      }
      cout << ans << endl;
  return 0;
    }
```

(1)①处应填(　　)

A.[0][0]　　B.[0][n]　　C.[m][0]　　D.[1][1]

(2)②处应填(　　)。

A. rowsum[i][0]=matrixi[i][1]　　B. rowsum[i][0]=1

C. rowsum[i][0]=0　　D.rowsum[i][0]=ans

(3)3 处应填(　　)。

A. rowsum[i][j-1]　　B. rowsum[i-1][j]+matrix[i][j]

C. matrix[i][j]　　D.rowsum[i][j-1]+matrix[i[j]

(4)4 处应填(　　)。

A. area=0　　B.ans=area　　C. area=ans　　D.ans=0

(5)⑤处应填(　　)。

A. matrix[i][last]　　B. rowsum[i][last]-rowsum[i][first-1]

C. matrix[i][fist]　　D. rowsum[i][last]-rowsum[i][first]

3[2011].（子矩阵）给输入一个 n1*m1 的矩阵 a，和 n2*m2 的矩阵 b，问 a 中是否存在子矩阵和 b 相等。若存在，输出所有子矩阵左上角的坐标：若不存在输出“There isno answer”。

```
#include<iostream>
using namespace std;
const int SIZE = 50;
int n1,m1,n2,m2,a[SIZE][SIZE],b[SIZE][SIZE];
int main()
{
    int i,j,k1,k2;
    bool good ,haveAns;
```

```
    cin>>n1>>m1;
    for(i=1;i<=n1;i++)
       for(j=1;j<=m1;j++)
          cin>>a[i][j];
    cin>>n2>>m2;
    for(i=1;i<=n2;i++)
       for(j=1;j<=m2;j++)
          ____①____;  //输入 b 矩阵 cin>>b[i][j]
    haveAns=false;
    for(i=1;i<=n1-n2+1;i++)
       for(j=1;j<= ____②____;j++){//最大可能出现的列坐标 j<=m1-m2+1
            ____③____ ;//先假设相等  good=ture
           for(k1=1;k1<=n2;k1++)
               for(k2=1;k2<=____④____;k2++){//k2<m2 遍历矩阵 b
                  if(a[i+k1-1][j+k2-1]!=b[k1][k2])
                     good=false;
               }
           if(good){
              cout<<i<<' '<<j<<endl;
              ____⑤____;// haveAns=true 表示存在
           }
       }
    if(!haveAns)
       cout<<"There is no answer"<<endl;
    return 0;
}
```

●选择题

(1)1 处应填(　　)。

A. cin>>b[i][j]　　B.cin> >a[i][j]　　C.cin>>b[n1][m1]　　D.cin>>b[n2][m2]

(2)②处应填(　　)。

A.m1　　B.m1-m2+1　　C.m1-1　　D. m1+1

(3)③处应填(　　)。

A.good=0　　B. good= '1'　　C.good=false　　D.good=1

(4)4 处应填(　　)。

A.k1+1　　B. m2　　C.m2-1　　D.k1

(5)⑤处应填(　　)。

A. break　　B. return　　C.haveAns=true　　D.haveAns=false

6 答案

1.1 1-5 CDBCD 6-11 DCACAC

1.2 1-5 BCABD 6-10 CDCDD 11-15 ABBDB 16-20 ACBAA 21-25 BDAAB 26-30 DDCBA 31-35 ABCDC 36-38 DDA

1.3 1-5 CCCDC 6-10 DADCD 11-15 CCDCB 16-19 BBCAA

1.4 1-5 CDBCD 6-10 CBACB 11-15 ACDAB 16-20 DABDB 21-25 DCDAA 26-30 DBBCC 31-35 BBAAC 36-40 BDDDA 41-43 CBB

1.5 1-5 DDBCB 6-10 ACCDA 11-16 ACBADB

1.6 1-5 ADDBA 6-10 ADABB 11-15 DBDBB 16-20 BCBCC 21-25 ACBDA 26-30 CCDAC 31-35 CBBBC 36-39 ABDA

2.1 1-5 ACDAA 6-9 CBDD

2.2 1-5 DDCCC 6-10 DCCBB ABDBB

2.3 1-5 BCDCB 6-10 BABBA 11-15 BDAAB 16-20 CCBAA 21-24 ADBD

2.4 1-5 DADBA 6-10 CCBDD

4.1 1-5 CDDCD C-10BCDDA

4.2 1-5 CDAAC 6-10 ABAAC 11-15DBBDB 16-20 DBDAB 21-25 CADCA 26-30 BCBBB 31 B

4.3 1-5 BDAAD 6-10 ACADC 11-15ABDAD 16-20 DABAA 21-25 BCACC 26-30 ACCDC 31 C

4.4 1-5 CCACB 6-10 CABCC 11-15 CC

4.5 1-5 DCDBB 6-10 CAABA 11-15 ACAAC 16-20 ABBDA 21 D

4.6 1-5 ADADC 6-10 BCDDC 11-15 ACCBC 16-20 DDBAC 21-25ADABD 26-30 DDBCB 31- AAC

4.7 1-5 CCBAC 6-10 AABAC 11-14ABDB