

分区联赛初赛复习

初赛考的知识点就是计算机基本常识、基本操作和程序设计基础知识。其中选择题考查的是知识，而问题解决类型的题目更加重视能力的考查。一般说来，选择题只要多用心积累就可以了。问题解决题目的模式比较固定，大家应当做做以前的题目。写运行结果和程序填空也需要多做题目，并且培养良好的程序阅读和分析能力，就像语文的阅读理解一样。

近几年来，初赛的考查范围有了很大的变化，越来越紧跟潮流了。这就需要大家有比较广泛的知识，包括计算机硬件、软件、网络、简单的数据结构（例如栈、队列、树和图等）和简单的算法（例如排序、查找和搜索等），程序设计语言以及一些基本的数学知识和技巧（例如排列组合）。但最主要的，还是取决于你对程序设计语言的熟悉程度，再加上认真仔细的心态。

选择题

一、硬件

计算机发展可划分：

	年代	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—？	大规模集成电路

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。

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：

- 1、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。
- 2、**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。

我国的计算机发展情况

- 我国从 1956 年开始计算机的科研和教学工作；
- 1960 年我国第一台自行设计的通用电子计算机 107 机诞生；
1964 年我国研制成大型通用电子计算机 119 机；
- 1983 年每秒运行一亿次的银河巨型计算机在国防科技大学诞生；
1992 年研制成功每秒运行 10 亿次的“银河 II”巨型计算机；
1997 年又研制成功每秒运行 130 亿次的“银河 III”巨型计算机；
- 我国较有名的微型计算机品牌有：“联想”、“长城”、“方正”等；

微型机的主要技术指标

- 1、字长：知己算计能够直接处理的二进制数据的位数。单位为位（BIT）
- 2、主频：指计算机主时钟在一秒钟内发出的脉冲数，在很大程度上决定了计算机的运算速度。
- 3、内存容量：是标志计算机处理信息能力强弱的一向技术指标。单位为字节（BYTE）。
8BIT=1BYTE 1024B=1KB 1024KB=1MB
- 4、外存容量：一般指软盘、硬盘、光盘。

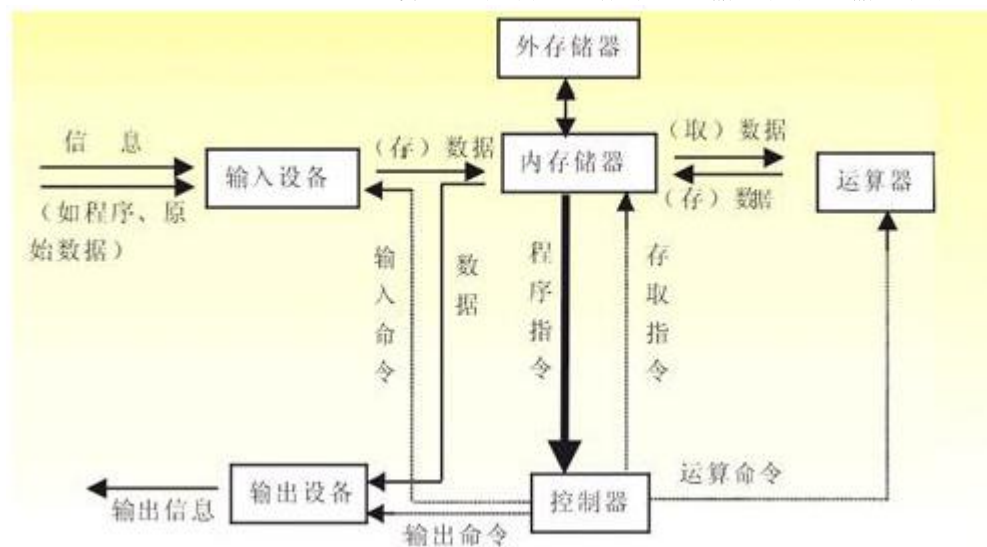
计算机的特点：

运算速度快，运算精度高，具有记忆能力，具有逻辑判断能力，具有自动控制能力；

计算机的应用：

- 1、数值计算：弹道轨迹、天气预报、高能物理等等
- 2、信息管理：企业管理、物资管理、电算化等
- 3、过程控制：工业自动化控制，卫星飞行方向控制
- 4、辅助工程：计算机辅助教学（CAI）、计算机辅助设计（CAD）、计算机辅助制造（CAM）、计算机辅助测试（CAT）、计算机集成制造（CIMS）等

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

内部存储器

中央处理器能直接访问的存储器称为内部存储器，它包括快速缓冲存储器和主存储器，中央处理器不能直接访问的存储器称为外部存储器，外部存储器中的信息必须调入内存后才能为中央处理器处理。

主存储器：内存也常泛称主存，但严格上说，只有当内存中只有主存，而没有快速缓冲存储器时，才能称为主存。

主存储器按读写功能，可分只读存储器（ROM）和随机存储器（RAM）两种。

外部存储器

外存储器：也称为辅助存储器，一般容量较大，速度比主存较慢。

硬盘（Hard disk）：目前的硬盘大多采用了**温彻斯特技术**，所以又称为“温盘”；

温氏技术的**特点**是：将盘片、读写磁头及驱动装置精密地组装在一个密封盒里；采用接触式起停，非接触式读写的方式（磁盘不工作时，磁头停在磁盘表面的起停区，一旦加电后，磁头随着盘片旋转的气流“飞”起来，悬浮在磁盘表面，进行读写）。

软盘（Floppy Disk）：目前常见的是 **3.5 英寸/1.44 MB** 的软盘。

光盘存储器（CD-ROM）：普通的 CD-ROM，只能读，不能写；CD 盘片的存储量大约是 **650 MB**。

输入设备

- 键盘（Keyboard）：目前大多使用 104 或 108 键盘
- 鼠标（Mouse）：主要有机械型鼠标和光电型鼠标两种
- 手写笔
- 触摸屏
- 麦克风
- 扫描仪（Scanner）
- 视频输入设备
- 条形码扫描器

输出设备

- 显示器（Monitor）：目前主要有 CRT（阴极射线管）显示器和 LCD 液晶显示器。
- 打印机（Printer）：主要有针式打印机、喷墨打印机、激光打印机。
- 绘图仪
- 音箱

例题

微型计算机的问世是由于（ C ）的出现。

- A) 中小规模集成电路 B) 晶体管电路 C) （超）大规模集成电路 D) 电子管电路

中央处理器(CPU)能访问的最大存储器容量取决于（ A ）。

- A) 地址总线 B) 数据总线 C) 控制总线 D) 实际内存容量

微型计算机中，（ C ）的存取速度最快。

- A) 高速缓存 B) 外存储器 C) 寄存器 D) 内存储器

在计算机硬件系统中，cache 是（ D ）存储器。

- A) 只读 B) 可编程只读 C) 可擦除可编程只读 D) 高速缓冲

若我们说一个微机的 CPU 是用的 PII300, 此处的 300 确切指的是 (A) 。

- A) CPU 的主时钟频率
- B) CPU 产品的系列号
- C) 每秒执行 300 百万条指令
- D) 此种 CPU 允许最大内存容量

计算机主机是由 CPU 与 (D) 构成的。

- A. 控制器
- B. 输入、输出设备
- C. 运算器
- D. 内存储器

计算机系统总线上传送的信号有 (B)。

- A. 地址信号与控制信号
- B. 数据信号、控制信号与地址信号
- C. 控制信号与数据信号
- D. 数据信号与地址信号

不同类型的存储器组成了多层次结构的存储器体系, 按存取速度从快到慢的排列是 (C)。

- A. 快存/辅存/主存
- B. 外存/主存/辅存
- C. 快存/主存/辅存
- D. 主存/辅存/外存

微机内存储器的地址是按 (C) 编址的。

- A. 二进制位
- B. 字长
- C. 字节
- D. 微处理器的型号

在微机中, 通用寄存器的位数是 (C)。

- A. 8 位
- B. 16 位
- C. 计算机字长
- D. 32 位

不同的计算机, 其指令系统也不同, 这主要取决于 (C)。

- A. 所用的操作系统
- B. 系统的总体结构
- C. 所用的 CPU
- D. 所用的程序设计语言

下列说法中, 哪个 (些) 是错误的 (BDE)。

- A) 程序是指令的序列, 它有三种结构: 顺序、分支和循环。
- B) 数据总线决定了中央处理器 CPU 所能访问的最大内存空间的大小。
- C) 中央处理器 CPU 内部有寄存器组, 用来储存数据。
- D) 不同厂家生产的 CPU 所能处理的指令集是相同的。
- E) 数据传输过程中可能会出错, 奇偶校验法可以检测出数据中哪一位在传输中出了差错。

CPU 访问内存的速度比访问下列哪个 (些) 存储设备要慢 (AD)。

- A) 寄存器
- B) 硬盘
- C) 软盘
- D) 高速缓存
- E) 光盘

下列哪个 (些) 不是个人计算机的硬件组成部分 (B)。

- A) 主板
- B) 虚拟内存
- C) 电源
- D) 硬盘
- E) 总线

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 (C)。

- 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。
- E. 指出计算机性能将以每两年翻一番的速度向前发展。

下列哪个不是 CPU（中央处理单元）（ B ）。

- A. Intel Itanium B. DDR SDRAM C. AMD Athlon64
- D. AMD Opteron E. IBM Power 5

下列说法中错误的是（ B ）。

- 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。
- E. 数据总线的宽度决定了一次传递数据量的大小，是影响计算机性能的因素之一。

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（ C ）。

- A. 针式打印机 B. 喷墨打印机 C. 激光打印机 D. 笔式绘图仪 E. 喷墨绘图仪

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（D）小时。

- A. 4 B. 2 C. 1 D. 1 / 2 E. 1 / 4

以下哪个不是计算机的输出设备（D）。

- A. 音箱 B. 显示器 C. 打印机 D. 扫描仪 E. 绘图仪

二、进制与编码

四种常用的数制及它们之间的相互转换：

进制	基数	基数个数	权	进数规律
十进制	0、1、2、3、4、5、6、7、8、9	10	10^i	逢十进一
二进制	0、1	2	2^i	逢二进一
八进制	0、1、2、3、4、5、6、7	8	8^i	逢八进一
十六进制	0、1、2、3、4、5、6、7、8、9、 A、B、C、D、E、F	16	16^i	逢十六进一

十进制数转换为二进制数、八进制数、十六进制数的方法：

二进制数、八进制数、十六进制数转换为十进制数的方法：按权展开求和法

1. 二进制与十进制间的相互转换：

（1）二进制转十进制

方法：“按权展开求和”

$$\begin{aligned}\text{例: } (1011.01)_2 &= (1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2})_{10} \\ &= (8 + 0 + 2 + 1 + 0 + 0.25)_{10} \\ &= (11.25)_{10}\end{aligned}$$

规律：个位上的数字的次数是 0，十位上的数字的次数是 1，……，依次递增，而十分位的数字的次数是-1，百分位上数字的次数是-2，……，依次递减。

注意：不是任何一个十进制小数都能转换成有限位的二进制数。

(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**排列”（短除反取余法）

$$\begin{array}{r|l} \text{例: } (89)_{10} = (1011001)_2 & \\ 2 & 89 \\ \hline 2 & 44 \quad \dots\dots 1 \\ \hline 2 & 22 \quad \dots\dots 0 \\ \hline 2 & 11 \quad \dots\dots 0 \\ \hline 2 & 5 \quad \dots\dots 1 \\ \hline 2 & 2 \quad \dots\dots 1 \\ \hline 2 & 1 \quad \dots\dots 0 \\ \hline & 0 \quad \dots\dots 1 \end{array}$$

- 十进制小数转二进制数：“乘以 2 取整，**顺序**排列”（乘 2 取整法）

$$\begin{array}{r|l} \text{例: } (0.625)_{10} = (0.101)_2 & \\ & 0.625 \\ \hline X & 2 \quad \quad \quad 1 \\ \hline & 1.25 \\ \hline X & 2 \quad \quad \quad 0 \\ \hline & 0.5 \\ \hline X & 2 \quad \quad \quad 1 \\ \hline & 1.0 \end{array}$$

2. 八进制与二进制的转换：

二进制数转换成八进制数：从小数点开始，整数部分向左、小数部分向右，每 3 位为一组用一位八进制数的数字表示，不足 3 位的要用“0”补足 3 位，就得到一个八进制数。

八进制数转换成二进制数：把每一个八进制数转换成 3 位的二进制数，就得到一个二进制数。

例：将八进制的 37.416 转换成二进制数：

$$\begin{array}{ccccccc} 3 & 7 & . & 4 & 1 & 6 & \\ 011 & 111 & . & 100 & 001 & 110 & \\ \text{即: } (37.416)_8 & = & (11111.10000111)_2 \end{array}$$

例：将二进制的 10110.0011 转换成八进制：

$$\begin{array}{ccccccc} 0 & 1 & 0 & 1 & 1 & 0 & . & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 2 & 6 & . & 1 & 4 & & & & & & & & \\ \text{即: } (10110.011)_2 & = & (26.14)_8 \end{array}$$

3. 十六进制与二进制的转换：

二进制数转换成十六进制数：从小数点开始，整数部分向左、小数部分向右，每 4 位为一组用一位十六进制数的数字表示，不足 4 位的要用“0”补足 4 位，就得到一个十六进制数。

十六进制数转换成二进制数：把每一个八进制数转换成 4 位的二进制数，就得到一个二进制数。

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

注意：以上所说的二进制数均是无符号的数。这些数的范围如下表：

无符号位二进制数位数	数值范围	十六进制范围表示法
8 位二进制数	0~255 (255=2 ⁸ -1)	00~0FFH
16 位二进制数	0~65535 (65535=2 ¹⁶ -1)	0000H~0FFFFH
32 位二进制数	0~2 ³² -1	00000000H~FFFFFFFFH

带符号数的机器码表示方法

1. 带符号二进制数的表示方法：

带符号二进制数用最高位的一位数来表示符号：0 表示正，1 表示负。

含符号位二进制数位数	数值范围	十六进制范围表示法
8 位二进制数	-128 ~ +127	80H~7FH
16 位二进制数	-32768 ~ +32767	8000H~7FFFH
32 位二进制数	-2147483648 ~ +2147483647	80000000H~FFFFFFFFH

2、符号位的表示：最常用的表示方法有原码、反码和补码。

(1) 原码表示法：一个机器数 x 由符号位和有效数值两部分组成，设符号位为 x₀，x 真值的绝对值 |x| = x₁x₂x₃...x_n，则 x 的机器数原码可表示为：

$[x]_{\text{原}} = x_0x_1x_2...x_n$ ，当 x ≥ 0 时，x₀=0，当 x < 0 时，x₀=1。

例如：已知：x₁ = -1011B，x₂ = +1001B，则 x₁，x₂ 有原码分别是

$[x_1]_{\text{原}} = 11011B$ ， $[x_2]_{\text{原}} = 01001B$

规律：正数的原码是它本身，负数的原码是取绝对值后，在最高位（左端）补“1”。

(2) 反码表示法：一个负数的原码符号位不变，其余各位按位取反就是机器数的反码表示法。正数的反码与原码相同。

按位取反的意思是该位上是 1 的，就变成 0，该位上是 0 的就变成 1。即 $\overline{1}=0$ ， $\overline{0}=1$

例：x₁ = -1011B，x₂ = +1001B，求 $[x_1]_{\text{反}}$ 和 $[x_2]_{\text{反}}$ 。

解： $[x_1]_{\text{反}} = 10100B$ ， $[x_2]_{\text{反}} = 01001B$

(3) 补码表示法：

首先分析两个十进制数的运算：78-38=41，79+62=141

如果使用两位数的运算器，做 79+62 时，多余的 100 因为超出了运算器两位数的范围而自动丢弃，这样在做 78-38 的减法时，用 79+62 的加法同样可以得到正确结果。

模是批一个计量系统的测量范围，其大小以计量进位制的基数为底数，位数为指数的幂。如两位十进制数的测量范围是 1——9，溢出量是 100，模就是 10²=100，上述运算称为模运算，可以写作：

$$79 + (-38) = 79 + 62 \quad (\text{mod } 100)$$

进一步写为 -38=62，此时就说 -38 的补法（对模 100 而言）是 62。计算机是一种有限字长的数字系统，因此它的运算都是有模运算，超出模的运算结果都将溢出。n 位二进制的模是 2ⁿ，

一个数的补码记作 $[x]_{\text{补}}$ ，设模是 M，x 是真值，则补码的定义如下：

$$[x]_{\text{补}} = \begin{cases} [x]_{\text{原}} & (x \geq 0) \\ M + x & (x < 0) \end{cases}$$

例：设字长 $n=8$ 位， $x=-1011011\text{B}$ ，求 $[x]_{\text{补}}$ 。

解：因为 $n=8$ ，所以模 $M=2^8=100000000\text{B}$ ， $x<0$ ，所以

$$[x]_{\text{补}} = M + x = 100000000\text{B} - 1011011\text{B} = 10100101\text{B}$$

注意：这个 x 的补码的最高位是“1”，表明它是一个负数。对于二进制数还有一种更加简单的方法由原码求出补码：

(1) 正数的补码表示与原码相同；

(2) 负数的补码是将原码符号位保持“1”之后，其余各位按位取反，末位再加 1 便得到补码，即取其原码的反码再加“1”： $[x]_{\text{补}} = [x]_{\text{反}} + 1$ 。

下表列出 $\pm 0, \pm 39, \pm 127$ 及 -128 的 8 位二进制原码，反码和补码并将补码用十六进制表示。

真值	原码 (B)	反码 (B)	补码 (B)	补码 (H)
+127	0 111 1111	0 111 1111	0 111 1111	7F
+39	0 010 0111	0 010 0111	0 010 0111	27
+0	0 000 0000	0 000 0000	0 000 0000	00
-0	1 000 0000	1 111 1111	0 000 0000	00
-39	1 010 0111	1 101 1000	1 101 1001	D9
-127	1 111 1111	1 000 0000	1 000 0001	81
-128	无法表示	无法表示	1 000 0000	80

从上可看出，真值+0和-0的补码表示是一致的，但在原码和反码表示中具有不同形式。8 位补码机器数可以表示-128，但不存在+128的补码与之对应，由此可知，8 位二进制补码能表示数的范围是-128——+127。还要注意，不存在-128的 8 位原码和反码形式。

定点数和浮点数

(一) 定点数 (Fixed-Point Number)

计算机处理的数据不仅有符号，而且大量的数据带有小数，小数点不占有二进制一位而是隐含在机器数里某个固定位置上。通常采取两种简单的约定：一种是约定所有机器数的小数的小数点位置隐含在机器数的最低位之后，叫定点纯整机器数，简称定点整数。另一种约定所有机器数的小数点隐含在符号位之后、有效部分最高位之前，叫定点纯小数机器数，简称定点小数。无论是定点整数，还是定点小数，都可以有原码、反码和补码三种形式。

(二) 浮点数 (Floating-Point Number)

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：

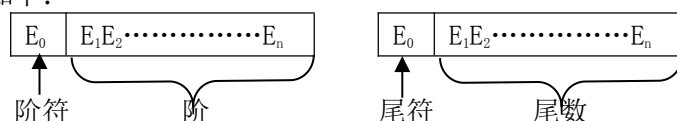
$$N = 2^E \times S$$

其中：E——N 的阶码 (Exponent)，是有符号的整数

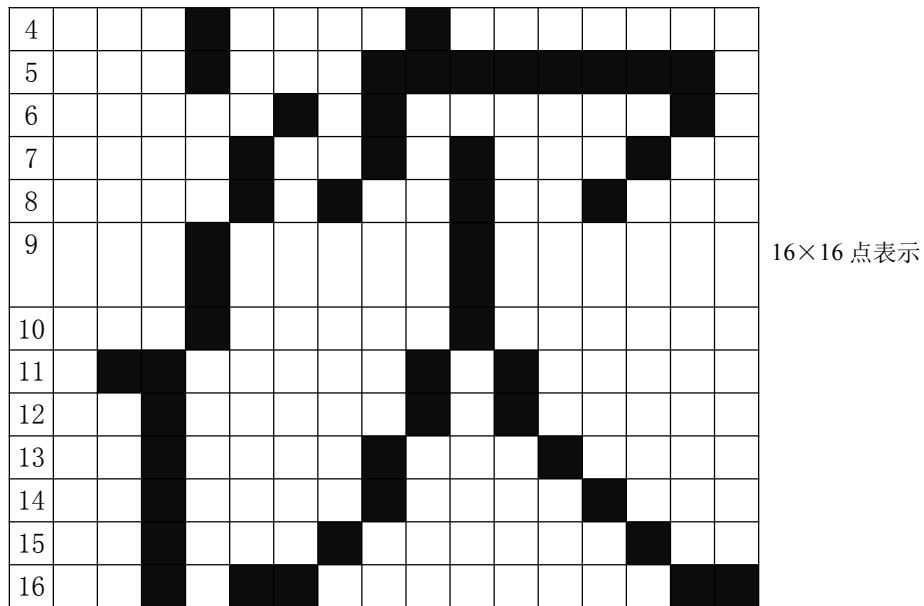
S——N 的尾数 (Mantissa)，是数值的有效数字部分，一般规定取二进制定点纯小数形式。

例： $1011101\text{B} = 2^{+7} * 0.1011101$ ， $101.1101\text{B} = 2^{+3} * 0.1011101$ ， $0.01011101\text{B} = 2^{-1} * 0.1011101$

浮点数的格式如下：



浮点数由阶码和尾数两部分组成，底数 2 不出现，是隐含的。阶码的正负符号 E_0 ，在最前位，阶反映了数 N 小数点的位置，常用补码表示。二进制数 N 小数点每左移一位，阶增加 1。尾数是这点小数，常取补码或原码，码制不一定与阶码相同，数 N 的小数点右移一位，在浮点数中表现为尾数左移一位。尾数的长度决定了数 N 的精度。尾数符号叫尾符，是数 N 的符号，也占一位。



一般的点阵规模有 16×16 , 24×24 , 32×32 , 64×64 等, 每一个点在存储器中用一个二进制位 (bit) 存储。例如, 在 16×16 的点阵中, 需 $16 \times 16 \text{ bit} = 32 \text{ byte}$ 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, 普遍采用了字形数据压缩技术。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。

例题

十进制数 $11/128$ 可用二进制数码序列表示为 (D)。

- A) $1011/1000000$ B) $1011/100000000$ C) 0.001011 D) 0.0001011

算式 $(2047)_{10} - (3FF)_{16} + (2000)_8$ 的结果是 (A)。

- A) $(2048)_{10}$ B) $(2049)_{10}$ C) $(3746)_8$ D) $(1AF7)_{16}$

已知 $x = (0.1011010)_2$, 则 $[x/2] = (C)_2$ 。

- A) 0.1011101 . B) 11110110 C) 0.0101101 D) 0.100110

已知 $A = 35H$, 则 $A \wedge 05H \vee A \wedge 30H$ 的结果是: (C)。

- A) $30H$ B) $05H$ C) $35H$ D) $53H$

$[x]$ 补码 = 10011000 , 其原码为 (B)

- A) 011001111 B) 11101000 C) 11100110 D) 01100101

下列无符号数中, 最小的数是 (C)

- A. $(11011001)_2$ B. $(75)_{10}$ C. $(37)_8$ D. $(2A)_{16}$

计算机的运算速度取决于给定的时间内, 它的处理器所能处理的数据量。处理器一次能处理的数据量叫字长。已知 64 位的奔腾处理器一次能处理 64 个信息位, 相当于 (A) 字节。

- A. 8 个 B. 1 个 C. 16 个 D. 2 个

在 24*24 点阵的“字库”中，汉字“一”与“编”的字模占用字节数分别是（C）

A. 32, 32 B. 32, 72 C. 72, 72 D. 72, 32

计算机中的数有浮点数与定点数两种，其中用浮点数表示的数，通常由（C）这两部分组成。

A. 指数与基数 B. 尾数与小数 C. 阶码与尾数 D. 整数与小数

十进制算术表达式： $3*512+7*64+4*8+5$ 的运算结果，用二进制表示为（B）。

A. 10111100101 B. 11111100101
C. 111110100101 D. 11111101101

组成‘教授’（jiao shou）’副教授’（fu jiao shou）与‘讲师’（jiang shi）这三个词的汉字，在 GB2312—80 字符集中都是一级汉字。对这三个词排序的结果是（D）。

A. 教授，副教授，讲师 B. 副教授，教授，讲师
C. 讲师，副教授，教授 D. 副教授，讲师，教授

GB2312-80 规定了一级汉字 3755 个，二级汉字 3008 个，其中二级汉字字库中的汉字是以（ B ）为序排列的。

A. 以笔划多少 B. 以部首 C. 以 ASCII 码 D. 以机内码

十进制数 2004 等值于八进制数（ B ）。

A. 3077 B. 3724 C. 2766 D. 4002 E. 3755

$(2004)_{10} + (32)_{16}$ 的结果是（ D ）。

A. $(2036)_{10}$ B. $(2054)_{16}$ C. $(4006)_{10}$ D. $(100000000110)_2$ E. $(2036)_{16}$

十进制数 100.625 等值于二进制数（ B ）。

A. 1001100.101 B. 1100100.101 C. 1100100.011 D. 1001100.11 E. 1001100.01

以下二进制数的值与十进制数 23.456 的值最接近的是（D）。

A. 10111.0101 B. 11011.1111 C. 11011.0111 D. 10111.0111 E. 10111.1111

三、软件与操作系统

计算机软件可分为**系统软件**和**应用软件**两大类。

- 系统软件：用来支持应用软件的开发和运行的，主要是操作系统软件，如：
DOS、Windows95/98/2000、Unix、Linux、WindowsNT；
- 应用软件：为了某个应用目的而编写的软件，主要有文字处理软件、电子表格软件、数据库管理软件等。

操作系统（OS——Operating System）

操作系统是控制与管理计算机系统资源的软件，是硬件的第一层扩充，任何应用软件的运行都必须依靠操作系统的支持。

Windows 系列操作系统

Windows 是 Microsoft 公司开发的图形化界面的操作系统。

- 基本概念：
 - 图标、任务栏、标题栏、菜单栏、滚动条、工具栏、对话框、开始菜单……
- 基本操作：
 - (1) 鼠标单击、双击、拖动，左键、右键功能；
 - (2) 窗口操作：最大（小）化、大小调整、拖动、关闭、排列、切换；
 - (3) 菜单操作：
 - 激活、选择；
 - ★ 命令项的约定——
 - 正常显示和灰色显示；
 - 命令后带“…”：执行命令则弹出对话框；
 - 带快捷键：某些菜单命令的后面标有对应的键盘命令，称为该命令的快捷键或热键；
 - 选中标志：某些命令选项的左侧有用打勾表示的选中标志，说明此命令功能正在起作用；
 - 命令后带“▶”：级联：此命令后会有下一级的子命令菜单弹出供用户作进一步选择；
 - ★ 快捷菜单——当鼠标位于某个对象上，单击鼠标右键，可打开有关对象的快捷菜单；
 - (4) 剪贴板：复制（Ctrl+C）、粘贴（Ctrl+V）、剪切（Ctrl+X）
 - 复制屏幕图像：可将当前屏幕图形以 BMP 格式传送到剪贴板……
 - (5) 其它：查找、运行、切换 Windows、进入 DOS 环境、文件夹选项
 - 输入法切换，中、英文切换，半角/全角切换
 - 软键盘：是在屏幕上显示的一个键盘图形，用户可用鼠标点击其中某个键以替代实际的按键；
- 各种文件的后缀名：
 - bat、com、exe、sys、tmp、zip、……
 - doc、xls、txt、htm、……
 - bmp、gif、jpg、psd、……
 - wav、avi、mp3、swf……

DOS (Disk Operating System) 操作系统

由美国 Microsoft 公司发行的 DOS 称为 MS-DOS，主要由 IO.sys、MSDOS.sys、COMMAND.COM 三个基本文件和几十个内、外部命令文件组成。

* 主要命令：

- | | | |
|---|---|------|
| <ul style="list-style-type: none">• DIR——显示磁盘文件目录• CD——改变当前目录• MD——建立目录• RD——删除目录• DATE——显示和设置系统日期• TIME——显示和设置系统时间• COPY——复制文件• DEL——删除文件• REN——文件重命名• TYPE——显示文本文件内容 | } | 内部命令 |
| <ul style="list-style-type: none">• FORMAT——磁盘格式化• DISKCOPY——全盘复制 | } | 外部命令 |

- BACKUP——文件备份
- CHKDSK——检查磁盘
-

例题

在磁盘上建立子目录有许多优点,下列描述中不属于建立子目录优点的是(D)。

- A) 便于文件管理 B) 解决根目录中目录项个数有限问题
C) 加快文件查找速度 D) 节省磁盘使用空间

资源管理器的目录前图标中增加“+”号,这个符号的意思是(B)。

- A) 该目录下的子目录已经展开 B) 该目录下还有子目录未展开
C) 该目录下没有子目录 D) 该目录为空目录

在树型目录结构中,不允许两个文件名相同主要指的是(D)

- A) 同一个磁盘的不同目录下 B) 不同磁盘的同一个目录下
C) 不同磁盘的不同目录下 C) 同一个磁盘的同一个目录下

以下对 Windows 的叙述中,正确的是(A)

- A) 从软盘上删除的文件和文件夹,不送到回收站
B) 在同一个文件夹中,可以创建两个同类、同名的文件
C) 删除了某个应用程序的快捷方式,将删除该应用程序对应的文件
D) 不能打开两个写字板应用程序

WINDOWS 9X 是一种(D)操作系统

- A. 单任务字符方式 B. 单任务图形方式 C. 多任务字符方式 D. 多任务图形方式

在 config.sys 文件中,装入特定的可安装设备驱动程序的命令是 (D).

- A. buffer B. files C. xcopy D. device

下列文件名中,属于 DOS 中的保留设备名的为(A)

- A. aux B. com C. conl D. pr nl

启动计算机引导 DOS 是将操作系统(D)

- A. 从磁盘调入中央处理器 B. 从内存存储器调入高速缓冲存储器
C. 从软盘调入硬盘 D. 从系统盘调入内存存储器

DOS 暂驻区中的程序主要是用于(A)

- A) 执行 DOS 内部命令 B) 执行 DOS 外部命令
C) 执行 DOS 所有命令 D) 基本输入输出

下列哪个软件属于操作系统软件(E)。

- A. Microsoft Word B. 金山词霸 C. Foxmail D. WinRAR E. Red Hat Linux

下列哪个不是数据库软件的名称（ D ）。

A. MySQL B. SQL Server C. Oracle D. 金山影霸 E. Foxpro

以下哪个软件不是即时通信软件（D）。

A. 网易泡泡 B. MSN Messenger C. Google Talk D. 3DS Max E. QQ

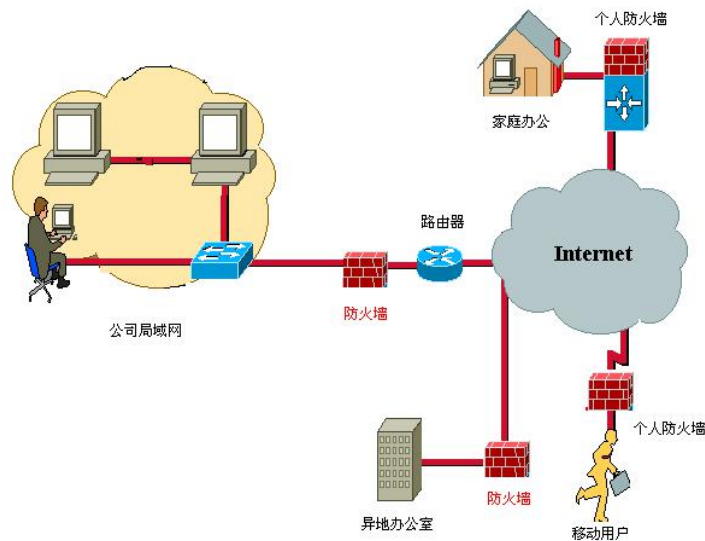
四、信息安全

计算机安全(computer security)是指防范与保护计算机系统及其信息资源在生存过程中免受蓄意攻击、人为失误和自然灾害等引起的损失和破坏。

计算机病毒是人类自己想像和发明出来的，它是一种特殊的程序，有着与生物病毒极为相似的特点。一是寄生性，它们大多依附在别的程序上面。二是隐蔽性，它们是悄然进入系统的，人们很难察觉。三是潜伏性，它们通常是潜伏在计算机程序中，只在一定条件下才发作的。四是传染性，它们能够自我复制繁殖，通过传输媒介蔓延。五是破坏性，轻则占用一定数量的系统资源，重则破坏整个系统。

对于计算机病毒，我们不必谈虎变色，而应采取积极的防治态度。首先，要防止“病从口入”，因为病毒不是自生的，而是外来的。另外，要用优秀的防杀病毒软件，对外来的软件 and 资料要进行严格的检查和杀毒。注意，防杀病毒软件需要及时更新(主要是其中的数据文件)，一般每周一次，不更新基本上等于没有防杀毒功能。

20 世纪 50、60 年代，黑客(hacker)曾是编程高手的代名词。后来，黑客成为一个独特的群体，他们通过各种渠道交流技艺，不少人以攻击计算机及其网络系统为乐趣。黑客们的胆大妄为已经给社会造成了很大的影响，一些黑客已经蜕变为威胁社会安全的罪犯。要防止“黑客”攻击，主要方法是加强安全措施，例如设置防火墙（见图 3.1.1）。防火墙是一种计算机设备，它设置在内部网络与外部网络之间，起一个隔离的作用，既可以阻止外部信息非法进入内部系统，也可以阻止内部人员非法访问外部系统。



例题

计算机病毒传染的必要条件是(B)。

- A) 在内存中运行病毒程序 B) 对磁盘进行读写操作
C) 在内存中运行含有病毒的程序 D) 复制文件

计算机病毒是(B)

- 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

计算机病毒的特点是(C)

- A. 传播性、潜伏性、易读性与隐蔽性 B. 破坏性、传播性、潜伏性与安全性
C. 传播性、潜伏性、破坏性与隐蔽性 D. 传播性、潜伏性、破坏性与易读性

一台计算机如果要利用电话线上网,就必须配置能够对数字信号和模拟信号进行相互转换的设备,这种设备是(A)。

- A. 调制解调器 B. 路由器 C. 网卡 D. 网关 E. 网桥

五、网络

1. 关于网络的一些定义:

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。

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

TCP/IP是一组协议, 包括上百个各种功能的协议, 其中TCP 和IP是最核心的两个协议。TCP/IP 协议把Internet网络系统描述成具有四个层次功能的网络模型。

1. 链路层: 这是TCP/IP 结构的第一层, 也叫网络接口层, 其功能是提供网络相邻节点间的信息传输以及网络硬件和设备驱动。

2. 网络层: (IP协议层) 其功能是提供源节点和目的节点之间的信息传输服务, 包括寻址和路由器选择等功能。

3. 传输层: (TCP 协议) 其功能是提供网络上的各应用程序之间的通信服务。

4. 应用层: 这是TCP/IP最高层, 其功能是为用户提供访问网络环境的手段, 主要提供FTP、TELNET、GOPHER等功能软件。

IP协议适用于所有类型网络。TCP 协议则处理IP协议所遗留的通信问题, 为应用程序提供可靠的通信连接, 并能自动适应网络的变化。TCP/IP 目前成为最为成功的网络体系结构和协议规范。

2、Netbeui: 一种非常简单的协议, MICROSOFT 开发。

3、IPX: 用于 NOVELL 网络。

2. 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

3. 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

4. 网络的分类

计算机网络的分类方式有很多种, 可以按地理范围、拓扑结构、传输速率和传输介质等分类。

(1)按地理范围分类

①局域网LAN(Local Area Network)

局域网地理范围一般几百米到10km 之内, 属于小范围内的连网。如一个建筑物内、一个学校内、一个工厂的厂区内等。局域网的组建简单、灵活, 使用方便。

②城域网MAN(Metropolitan Area Network)

城域网地理范围可从几十公里到上百公里, 可覆盖一个城市或地区, 是一种中等形式的网络。

③广域网WAN(Wide Area Network)

广域网地理范围一般在几千公里左右, 属于大范围连网。如几个城市, 一个或几个国家, 是网络系统中的最大型的网络, 能实现大范围的资源共享, 如国际性的Internet 网络。

(2)按传输速率分类

网络的传输速率有快有慢, 传输速率快的称高速网, 传输速率慢的称低速网。传输速率的单位是b/s(每秒比特数, 英文缩写为bps)。一般将传输速率在Kb/s—Mb/s范围的网称低速网, 在Mb/s—Gb/s 范围的网称高速网。也可以将Kb/s 网称低速网, 将Mb/s网称中速网, 将Gb/s网称高速网。

网络的传输速率与网络的带宽有直接关系。带宽是指传输信道的宽度, 带宽的单位是

Hz(赫兹)。按照传输信道的宽度可分为窄带网和宽带网。一般将KHz—MHz带宽的网称为窄带网,将MHz—GHz 的网称为宽带网,也可以将kHz 带宽的网称窄带网,将MHz 带宽的网称中带网,将GHz 带宽的网称宽带网。通常情况下,高速网就是宽带网,低速网就是窄带网。

(3)按传输介质分类

传输介质是指数据传输系统中发送装置和接受装置间的物理媒体,按其物理形态可以划分为有线和无线两大类。

①有线网

传输介质采用有线介质连接的网络称为有线网,常用的有线传输介质有双绞线、同轴电缆和光导纤维。

●双绞线是由两根绝缘金属线互相缠绕而成,这样的一对线作为一条通信线路,由四对双绞线构成双绞线电缆。双绞线点到点的通信距离一般不能超过100m。目前,计算机网络上使用的双绞线按其传输速率分为三类线、五类线、六类线、七类线,传输速率在10Mbps到600Mbps之间,双绞线电缆的连接器一般为RJ-45。

●同轴电缆由内、外两个导体组成,内导体可以由单股或多股线组成,外导体一般由金属编织网组成。内、外导体之间有绝缘材料,其阻抗为50Ω。同轴电缆分为粗缆和细缆,粗缆用DB-15连接器,细缆用BNC和T 连接器。

●光缆由两层折射率不同的材料组成。内层是具有高折射率的玻璃单根纤维体组成,外层包一层折射率较低的材料。光缆的传输形式分为单模传输和多模传输,单模传输性能优于多模传输。所以,光缆分为单模光缆和多模光缆,单模光缆传送距离为几十公里,多模光缆为几公里。光缆的传输速率可达到每秒几百兆位。光缆用ST 或SC 连接器。光缆的优点是不会受到电磁的干扰,传输的距离也比电缆远,传输速率高。光缆的安装和维护比较困难,需要专用的设备。

②无线网

采用无线介质连接的网络称为无线网。目前无线网主要采用三种技术:微波通信,红外线通信和激光通信。这三种技术都是以大气为介质的。其中微波通信用途最广,目前的卫星网就是一种特殊形式的微波通信,它利用地球同步卫星作中继站来转发微波信号,一个同步卫星可以覆盖地球的三分之一以上表面,三个同步卫星就可以覆盖地球上全部通信区域。

(4)按拓扑结构分类

计算机网络的物理连接形式叫做网络的物理拓扑结构。连接在网络上的计算机、大容量的外存、高速打印机等设备均可看作是网络上的一个节点,也称为工作站。计算机网络中常用的拓扑结构有总线型、星型、环型等。

①总线拓扑结构

总线拓扑结构是一种共享通路的物理结构。这种结构中总线具有信息的双向传输功能,普遍用于局域网的连接,总线一般采用同轴电缆或双绞线。

总线拓扑结构的优点是:安装容易,扩充或删除一个节点很容易,不需停止网络的正常工作,节点的故障不会殃及系统。由于各个节点共用一个总线作为数据通路,信道的利用率高。但总线结构也有其缺点:由于信道共享,连接的节点不宜过多,并且总线自身的故障可以导致系统的崩溃。

②星型拓扑结构

星型拓扑结构是一种以中央节点为中心,把若干外围节点连接起来的辐射式互联结构。这种结构适用于局域网,特别是近年来连接的局域网大都采用这种连接方式。这种连接方式以双绞线或同轴电缆作连接线路。

星型拓扑结构的特点是:安装容易,结构简单,费用低,通常以集线器(Hub)作为中央节点,便于维护和管理。中央节点的正常运行对网络系统来说是至关重要的。

③环型拓扑结构

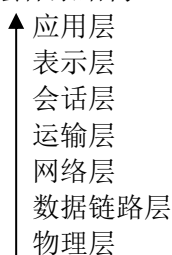
环型拓扑结构是将网络节点连接成闭合结构。信号顺着一个方向从一台设备传到另一台设备，每一台设备都配有一个收发器，信息在每台设备上的延时时间是固定的。

这种结构特别适用于实时控制的局域网系统。

环型拓扑结构的特点是：安装容易，费用较低，电缆故障容易查找和排除。有些网络系统为了提高通信效率和可靠性，采用了双环结构，即在原有的单环上再套一个环，使每个节点都具有两个接收通道。环型网络的弱点是，当节点发生故障时，整个网络就不能正常工作。

5. 网络的体系结构

OSI 的七层体系结构：



6. 局域网的工作方式

通常有两种：

- 客户机/服务器(Client/Server)：
提供资源并管理资源的计算机称为服务器；使用共享资源的计算机称客户机；
- 对等(Peer-to-Peer)：
不使用服务器来管理网络共享资源，所以的计算机处于平等的地位。

7. Internet 的形成与发展

又称国际互联网，规范的译名是“因特网”，指当前各国、各地区众多开发的网络连接在一起而形成的全球性网络。

- 我国 Internet 的发展情况：

八十年代末，九十年代初才起步。

1989 年我国第一个公用分组交换网 CNPAC 建成运行。

- 我国已陆续建成与 Internet 互联的四个全国范围的公用网络：

中国公用计算机互联网 (CHINANET)、中国金桥信息网 (CHINAGBN)

中国教育和科研计算机网 (CERNET)、中国科学技术网 (CSTNET)

8. IP 地址：

我们把整个 Internet 看作一个单一的、抽象的网络，所谓 IP 地址，就是为 Internet 中的每一台主机分配一个在全球范围唯一地址。IP v4 地址是由 32 位二进制数码表示的，为方便记忆，把这 32 位二进制数每 8 个一段用“.” 隔开，再把每一段的二进制数化成十进制数，也就得到我们现在所看到的 IP 地址形式。

IP 地址是用“.” 隔开地四个十进制整数，每个数字取值为 0—255。

IP 地址分 A、B、C、D、E 五类，目前大量使用的是 A、B、C 三类，D 类为 Internet 体系结构委员会 IAB 专用，E 类保留在今后使用。

最高位 1..126 为 A 类，128..191 是 B 类，192..223 是 C 类。

9. 域名：域名地址采用层次结构，一个域名一般有 3—5 个子段，中间用“.” 隔开。IP 地址作为 Internet 上主机的数字标识，对计算机网络来说是非常有效的。但对于使用者来说，很难记忆这些由数字组成的 IP 地址了。为此，人们研究出一种字符型标识，在 Internet 上采用“名称”寻址方案，为每台计算机主机都分配一个独有的“标准名称”，这个用字符表示

的“标准名称”就是我们现在所广泛使用的域名(DN, domain name)。因此主机的域名和IP地址一样,也采用分段表示的方法。其结构一般是如下样式:计算机名.组织结构名.网络名.最高层域名。

顶级域名有三类:

- 国家顶级域名,如 cn (中国)、us (美国)、uk (英国);
- 国际顶级域名—— int , 国际性组织可在 int 下注册;
- 通用顶级域名,如: com、net、edu、gov、org、……

有了域名标识,对于计算机用户来说,在使用上的确方便了很多。但计算机本身并不能自动识别这些域名标识,于是域名管理服务器DNS(domain name system)就应运而生了。所谓的域名管理系统DNS(domain name system)就是以主机的域名来代替其在Internet上实际的IP地址的系统,它负责将Internet上主机的域名转化为计算机能识别的IP地址。从DNS的组织结构来看,它是一个按照层次组织的分布式服务系统;从它的运行机制来看,DNS更像一个庞大的数据库,只不过这个数据库并不存储在任一计算机上,而是分散在遍布于整个Internet上数以千计的域名服务器中而已。

通过上面的IP地址、域名DN和域名管理系统DNS,就把Internet上面的每一台主机给予了唯一的定位。三者之间的具体联系过程如下:当连接网络并输入想访问主机的域名后,由本地机向域名服务器发出查询指令,域名服务器通过连接在整个域名管理系统查询对应的IP地址,如找到则返回相应的IP地址,反之则返回错误信息。说到这里,想必大家都明白了为什么当我们在浏览时,浏览器左下角的状态条上会有这样的信息:“正在查找xxxxxx”、“xxxxxx已经发现,正在连接xxxxxx”,其实这也就是域名通过DNS转化为IP地址的过程。

当然域名通过DNS转化为IP地址需要等待一段时间,因为如果你所使用的域名服务器上如果没有你所需要域名的对应IP地址,它就会向上级域名服务器查询,如此类推,直至查到结果,或返回无效信息。一般而言,这个查询过程都非常短,你很难察觉到。

10. Internet(译为因特网或国际互联网)的服务与工具

Internet的服务有:电子邮件、远程登陆、文件传输、信息服务等;

- 电子邮件(E-Mail):电子邮件地址格式为:

收信人邮箱名@邮箱所在主机的域名。例:winner01@21cn.com, qfit168@yahoo.com.cn

- 远程登陆(Telnet):指通过Internet与其它主机连接。

登陆上另一主机,你就可以使用该主机对外开放的各种资源,如联机检索、数据查询。

- 文件传输(FTP):用于在计算机间传输文件。如下载软件等。

11. 全球信息网(WWW—World Wide Web):

又称万维网,是一个全球规模的信息服务系统,由遍布于全世界的数以万计的Web站点组成。

例题

在使用E-mail前,需要对OUTLOOK进行设置,其中接收电子邮件的服务器称为(A)服务器。

- A)POP3 B)SMTP C)DNS D)FTP

Ip v4地址是由(B)位二进制数码表示的。

- A)16 B)32 C)24f D)8

Email 邮件本质上是一个 (A)

- A) 文件 B) 电报 C) 电话 D) 传真

TCP/IP 协议共有 (B) 层协议

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

Internet 的规范译名应为 (B)

- A. 英特尔网 B. 因特网 C. 万维网 D. 以太网

计算机网络是一个 (D)

- A. 管理信息系统 B. 管理数据系统 C. 编译系统 D. 在协议控制下的多机互连系统

下面哪些计算机网络不是按覆盖地域划分的 (D)

- A. 局域网 B. 都市网 C. 广域网 D. 星型网

下列网络上常用的名字缩写对应的中文解释错误的是 (D)。

- 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。
E. TCP (Transfer Control Protocol): 传输控制协议。

常见的邮件传输服务器使用 (B) 协议发送邮件。

- A. HTTP B. SMTP C. TCP D. FTP E. POP3

不能在Linux 上使用的网页浏览器是 (A) 。

- A. Internet Explore B. Netscape C. Opera D. Firefox E. Mozilla

六、数据结构与算法

例题

一个高度为 h 的二叉树最小元素数目是 (B)。

- A) $2h+1$ B) h C) $2h-1$ D) $2h$
E) $2h-1$

一个向量第一个元素的存储地址是 100, 每个元素的长度是 2, 则第 5 个元素的地址是 (B)。

A) 110 B) 108 C) 100 D) 109

设有一个含有 13 个元素的 Hash 表 ($0 \sim 12$), Hash 函数是: $H(\text{key}) = \text{key} \% 13$, 其中 $\%$ 是求余数运算。用线性探查法解决冲突, 则对于序列 (2、8、31、20、19、18、53、27), 18 应放在第几号格中(B)。

A) 5 B) 9 C) 4 D) 0

按照二叉树的定义, 具有 3 个结点的二叉树有(C) 种。

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

在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的(B) 倍。

A) $1/2$ B) 1 C) 2 D) 4

要使 1...8 号格子的访问顺序为: 8、2、6、5、7、3、1、4, 则下图中的空格中应填入(C)。

1	2	3	4	5	6	7	8
4	6	1	-1	7		3	2

A) 6 B) 0 C) 5 D) 3

设栈 S 和队列 Q 的初始状态为空, 元素 $e_1, e_2, e_3, e_4, e_5, e_6$ 依次通过栈 S, 一个元素出栈后即进入队列 Q, 若出队的顺序为 $e_2, e_4, e_3, e_6, e_5, e_1$, 则栈 S 的容量至少应该为(B)。

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

设有一棵 k 叉树, 其中只有度为 0 和 k 两种结点, 设 n_0, n_k 分别表示度为 0 和度为 k 的结点个数, 试求出 n_0, n_k 之间的关系 ($n_0 =$ 数学表达式, 数学表达式仅含 n_k, k 和数字)

$$N_0 = (K-1) N_k + 1$$

若已知一个栈的入栈顺序是 1, 2, 3, ..., n, 其输出序列为 $P_1, P_2, P_3, \dots, P_n$, 若 P_1 是 n, 则 P_i 是(C)

A) i B) $n-1$ C) $n-i+1$ D) 不确定

以下哪一个不是栈的基本运算(B)

A) 删除栈顶元素 B) 删除栈底的元素
C) 判断栈是否为空 D) 将栈置为空栈

下面关于算法的错误说法是(B)

A) 算法必须有输出 B) 算法必须在计算机上用某种语言实现
C) 算法不一定有输入 D) 算法必须在有限步执行后能结束

在顺序表 (2, 5, 7, 10, 14, 15, 18, 23, 35, 41, 52) 中, 用二分法查找 12, 所需的关键词比较的次数为(C)

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

一棵二叉树的高度为 h ，所有结点的度为 0，或为 2，则此树最少有 (B) 个结点

A) $2h-1$ B) $2h-1$ C) $2h+1$ D) $h+1$

无向图 $G=(V,E)$ ，其中 $V=\{a, b, c, d, e, f\}$ $E=\{(a, b), (a, e), (a, c), (b, e), (c, f), (f, d), (e, d)\}$

对该图进行深度优先遍历，得到的顶点序列正确的是 (D)

A) a, b, e, c, d, f B) a, c, f, e, b, d C) a, e, b, c, f, d D) a, b, e, d, f, c

已知一棵二叉树的结点名为大写英文字母，其中序与后序遍历的顺序分别为：CBGEAFHDIJ 与 CGEBHFJIDA 则该二叉树的先序遍历的顺序为：

ABCEGDFHIJ

在有 N 个叶子节点的哈夫曼树中，其节点总数为 (B)

A. 不确定 B. $2N-1$ C. $2N+1$ D. $2N$

某数列有 1000 个各不相同的单元，由低至高按序排列；现要对该数列进行二分法检索 (binary-search)，在最坏的情况下，需检视 (B) 个单元。

A. 1000 B. 10 C. 100 D. 500

线性表若采用链表存贮结构，要求内存中可用存贮单元地址 (D)

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

下列叙述中，正确的是 (D)

A. 线性表的线性存贮结构优于链表存贮结构 B. 队列的操作方式是先进后出
C. 栈的操作方式是先进先出 D. 二维数组是指它的每个数据元素为一个线性表的线性表

已知，按中序遍历二叉树的结果为：abc

问：有多少种不同形态的二叉树可以得到这一遍历结果，并画出这些二叉树。

5 种

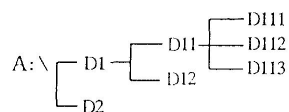
设有一个共有 n 级的楼梯，某人每步可走 1 级，也可走 2 级，也可走 3 级，用递推公式给出某人从底层开始走完全部楼梯的走法。例如：当 $n=3$ 时，共有 4 种走法，即 $1+1+1$ ， $1+2$ ， $2+1$ ， 3 。

$F(n)=f(n-1)+f(n-2)+f(n-3)$, $n \geq 4$;

$F(1)=1$; $f(2)=2$; $f(3)=4$;

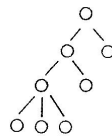
在磁盘的目录结构中，我们将与某个子目录有关联的目录数称为度。

例如下图：



该图表达了 A 盘的目录结构：D1，D11，……D2 均表示子目录的名字。在这里，根目录

的度为 2，D1 子目录的度为 3，D11 子目录的度为 4，D12，D2，D111，D112，D113 的度均为 1。又不考虑子目录的名字，则可简单的图示为如下的树结构：



若知道一个磁盘的目录结构中，度为 2 的子目录有 2 个，度为 3 的子目录有 1 个，度为 4 的子目录有 3 个。

试问：度为 1 的子目录有几个？

$$2 \times 2 + 3 \times 1 + 4 \times 3 + 1 \times x = (2 + 1 + 3 + x - 1) \times 2$$

根据 Nocomachns 定理，任何一个正整数 n 的立方一定可以表示成 n 个连续的奇数的和。

例如：

$$1^3 = 1$$

$$2^3 = 3 + 5$$

$$3^3 = 7 + 9 + 11$$

$$4^3 = 13 + 15 + 17 + 19$$

在这里，若将每一个式中的最小奇数称为 X ，那么当给出 n 之后，请写出 X 与 n 之间的关系表达式： $n^2 - n + 1$

设循环队列中数组的下标范围是 $1 \sim n$ ，其头尾指针分别为 f 和 r ，则其元素个数为（ D ）

- A. $r - f$ B. $r - f + 1$ C. $(r - f) \text{ MOD } n + 1$ D. $(r - f + n) \text{ MOD } n$

有 $2 \times n$ 的一个长方形方格，用一个 1×2 的骨牌铺满方格。例如 $n=3$ 时，为 2×3 方格。

此时用一个 1×2 的骨牌铺满方格，共有 3 种铺法：



试对给出的任意一个 $n (n > 0)$ ，求出铺法总数的递推公式。

$$F(1)=1 \quad F(2)=2 \quad F(n)=F(n-1)+F(n-2), \quad n \geq 3$$

FUNCTION ACK(M, N: INTEGER): INTEGER;

BEGIN

IF M=0 THEN ACK:=N+1

ELSE IF N=0 THEN ACK:=ACK(M-1, 1)

ELSE ACK:=ACK(M-1, ACK(M, N-1))

END;

BEGIN WRITELN(ACK(3, 4)); READLN; END.

输出

125

表达式 $(1+34) \times 5 - 56/7$ 的后缀表达式为（ C ）。

A) $1+34 \times 5 - 56/7$

B) $- * + 1 \ 34 \ 5 / 56 \ 7$

C) $1 \ 34 + 5 * 56 \ 7 / -$

D) $1 \ 34 \ 5 * + 56 \ 7 / -$

E) $1 \ 34 + 5 \ 56 \ 7 - *$

已知元素 (8, 25, 14, 87, 51, 90, 6, 19, 20), 问这些元素以怎样的顺序进入栈, 才能使出栈的顺序满足: 8 在 51 前面; 90 在 87 的后面; 20 在 14 的后面; 25 在 6 的前面; 19 在 90 的后面。 (D)。(题意是全部进栈, 再依次出栈)

- A) 20, 6, 8, 51, 90, 25, 14, 19, 87
- B) 51, 6, 19, 20, 14, 8, 87, 90, 25
- C) 19, 20, 90, 7, 6, 25, 51, 14, 87
- D) 6, 25, 51, 8, 20, 19, 90, 87, 14
- E) 25, 6, 8, 51, 87, 90, 19, 14, 20

假设我们用 $d=(a_1, a_2, \dots, a_5)$, 表示无向图 G 的 5 个顶点的度数, 下面给出的哪(些)组 d 值合理 (BE)。

- A) {5, 4, 4, 3, 1}
- B) {4, 2, 2, 1, 1}
- C) {3, 3, 3, 2, 2}
- D) {5, 4, 3, 2, 1}
- E) {2, 2, 2, 2, 2}

下列关于程序语言的叙述, 不正确的是 (D)。

- A) 编写机器代码不比编写汇编代码容易。
- B) 高级语言需要编译成目标代码或通过解释器解释后才能被 CPU 执行。
- C) 同样一段高级语言程序通过不同的编译器可能产生不同的可执行程序。
- D) 汇编代码可被 CPU 直接运行。
- E) 不同的高级语言语法略有不同。

下列哪个程序设计语言不支持面向对象程序设计方法 (C)。

- A. C++
- B. Object Pascal
- C. C
- D. Smalltalk
- E. Java

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, 5, 6, 7
- E. 1, 3, 6, 5, 7

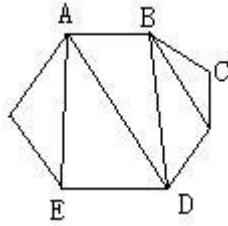
二叉树 T, 已知其前序遍历序列为 1 2 4 3 5 7 6, 中序遍历序列为 4 2 1 5 7 3 6, 则其后序遍历序列为 (B)。

- 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
- E. 4 5 2 6 3 7 1

满二叉树的叶结点个数为 N, 则它的结点总数为 (C)。

- A. N
- B. $2 * N$
- C. $2 * N - 1$
- D. $2 * N + 1$
- E. $2^N - 1$

在下图中, 从顶点 (E) 出发存在一条路径可以遍历图中的每条边一次, 而且仅遍历一次。



A. A 点 B. B 点 C. C 点 D. D 点 E. E 点

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	C ₀	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇
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			C ₀ , C ₁	C ₁ , C ₂	C ₃	C ₃ , C ₇	C ₀	C ₆

请你判断下列课程安排方案哪个是不合理的（ D ）。

- 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
 E. C₀, C₁, C₂, C₃, C₆, C₇, C₅, C₄

完全二叉树的结点个数为 $4 * N + 3$ ，则它的叶结点个数为（ E ）。

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

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（ D ）。

- A. AD B. BD C. CD D. DE E. EA

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（ C ）。

- A. 无法确定 B. B C. C D. D E. E

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（ E ）。

- 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. d, c, f, e, b, a, g E. g, e, f, d, c, b, a

将数组 {32, 74, 25, 53, 28, 43, 86, 47} 中的元素按从小到大的顺序排列，每次可以交换任意两个元素，最少需要交换 5 次。

取火柴游戏的规则如下：一堆火柴有 N 根，A、B 两人轮流取出。每人每次可以取 1 根或 2 根，最先没有火柴可取的人为败方，另一方为胜方。如果先取者有必胜策略则记为 1，先取者没有必胜策略记为 0。当 N 分别为 100, 200, 300, 400, 500 时，先取者有必胜策略的标记顺序为 11011（回答应为一个由 0 和/或 1 组成的字符串）

在所有排序方法中,关键字比较的次数与记录的初始排列次序无关的是(BD)。

A) 希尔排序 B) 起泡排序 C) 插入排序 D) 选择排序

七、排列组合

例题

在书架上放有编号为 $1, 2, \dots, n$ 的 n 本书。现将 n 本书全部取下然后再放回去,当放回去时要求每本书都不能放在原来的位置上。例如: $n=3$ 时:

原来位置为:123

放回去时只能为:312 或 231 这两种

问题:求当 $n=5$ 时满足以上条件的放法共有多少种?(不用列出每种放法)

$$c(5, 0) * 5! - c(5, 1) * 4! + c(5, 2) * 3! - c(5, 3) * 2! + c(5, 4) * 1! - c(5, 5) * 0! = 60 - 20 + 5 - 1 = 44$$

平面上有三条平行直线,每条直线上分别有 7, 5, 6 个点,且不同直线上三个点都不在同一条直线上。问用这些点为顶点,能组成多少个不同三角形?

$$C(7, 2) * (5+6) + C(5, 2) * (7+6) + C(6, 2) * (7+5) + 7 * 6 * 5 = 21 * 11 + 10 * 13 + 15 * 12 + 210 = 231 + 130 + 180 + 210 = 751$$

平面上有三条平行直线,每条直线上分别有 7, 5, 6 个点,且不同直线上三个点都不在同一条直线上。问用这些点为顶点,能组成多少个不同四边形?

$$21 * 10 + 21 * 15 + 10 * 15 + 21 * 30 + 10 * 42 + 15 * 35 = 1155 + 525 + 570 = 2250$$

由 3 个 a, 1 个 b 和 2 个 c 构成的所有字符串中,包含子串“abc”的共有 (D) 个。

A. 20 B. 8 C. 16 D. 12 E. 24

由 3 个 a, 5 个 b 和 2 个 c 构成的所有字符串中,包含子串“abc”的共有 (D) 个。

A. 40320 B. 39600 C. 840 D. 780 E. 60

$$8 * 7! / 2! / 4! - 4 * C(5, 2) - 4 * 5 = 8 * 3 * 5 * 7 - 40 - 20 = 840 - 60 = 780$$

八、综合

下面一段程序是用 (C) 语言书写的。

```
int func1(int n){
    int i, sum=0;
    for(i=1; i<=n; i++)
        sum+=i*i;
    return sum;
}
```

A) FORTRAN B) PASCAL C) C D)
PROLOG E) BASIC

多媒体计算机是指(D) 计算机。

- A) 专供家庭使用的 B) 装有 CD-ROM 的
B) 连接在网络上的高级 D) 具有处理文字、图形、声音、影像等信息的

在 WORD 文档编辑中实现图文混合排版时,关于文本框的下列叙述正确的是(C)。

- A) 文本框中的图形没有办法和文档中输入文字叠加在一起,只能在文档的不同位置
B) 文本框中的图形不可以衬于文档中输入的文字的下方。
C) 通过文本框,可以实现图形和文档中输入的文字的叠加,也可实现文字环绕。
D) 将图形放入文本框后,文档中输入的文字不能环绕图形。

计算机软件保护法是用来保护软件(D)的。

- A) 编写权 B) 复制权 C) 使用权 D) 著作权

64KB 的存储器用十六进制表示, 它的最大的地址码是(B)

- A) 10000 B) FFFF C) 1FFFF D) EFFFF

在外部设备中, 绘图仪属于(B)

- A. 输入设备 B. 输出设备 C. 辅(外)存储器 D. 主(内)存储器

某种计算机的内存容量是 640K, 这里的 640K 容量是指(C) 个字节

- A. 640 B. 640*1000 C. 640*1024 D. 640*1024*1024

已知数组中 A 中, 每个元素 A (I, J) 在存贮时要占 3 个字节, 设 I 从 1 变化到 8, J 从 1 变化到 10, 分配内

存时是从地址 SA 开始连续按行存贮分配的。

试问: A (5, 8) 的起始地址为(A)

- A. SA+141 B. SA+180 C. SA+222 D. SA+225

电线上停着两种鸟 (A, B), 可以看出两只相邻的鸟就将电线分为了一个线段。这些线段可分为两类;

一类是两端的小鸟相同; 另一类则是两端的小鸟不相同。

已知: 电线两个顶点上正好停着相同的小鸟, 试问两端为不同小鸟的线段数目一定是(B)。

- A. 奇数 B. 偶数 C. 可奇可偶 D. 数目固定

一个文本屏幕有 25 列及 80 行, 屏幕的左上角以 (1, 1) 表示, 而右下角则以 (80, 25) 表示, 屏幕上每

一个字符占用两字节 (byte), 整个屏幕则以线性方式存储在电脑的存储器内, 内屏幕左上角开始, 位移为 0, 然后逐列逐列存储。求位于屏幕 (X, Y) 的第一个字节的位移是(B)

- A. $(Y*80+X)*2-1$ B. $((Y-1)*80+X-1)*2$
 C. $(Y*80+X-1)*2$ D. $((Y-1)*80+X)*2-1$

计算机能直接执行的指令包括两部分，它们是 (B)。

- A. 源操作数与目标操作数 B. 操作码与操作数
 C. ASCII 码与汉字代码 D. 数字与字符

解释程序的功能是 (C)

- A. 将高级语言程序转换为目标程序 B. 将汇编语言程序转换为目标程序
 C. 解释执行高级语言程序 D. 解释执行汇编语言程序

1. 已知一个数列 $U_1, U_2, U_3, \dots, U_n, \dots$ 往往可以找到一个最小的 k 值和 k 个数 a_1, a_2, \dots, a_k , 使得数列从某项开始都满足:

$$U_{n+k} = a_1 U_{n+k-1} + a_2 U_{n+k-2} + \dots + a_k U_n \quad (A)$$

例如对斐波拉契数列 $1, 1, 2, 3, 5, \dots$ 可以发现: 当 $k=2$, $a_1=1$, $a_2=1$ 时, 从第 3 项起 (即 $n \geq 1$) 都满足 $U_{n+2} = U_{n+1} + U_n$ 。

试对数列 $1^2, 2^2, 3^2, \dots, n^2, \dots$ 求 k 和 a_1, a_2, \dots, a_k 使得 (A) 成立。 { 7% }

192.168.0.1 属于 (C)

- A. A 类地址 B. B 类地址 C. C 类地址 D. D 类地址

最高位 1..126 为 A 类, 128..191 是 B 类, 192..223 是 C 类。

十进制数 13 和 14, 进行“与”操作的结果是 (B)

- A. 27 B. 12 C. 15 D. 11

$1101 \text{ and } 1110 = 1100 = 12$

完全二叉树对每个节点从上往下, 从左往右编号, 第 i 层的第 j 个节点的编号是 (D)

- A. 2^i+j B. 2^i+j-1 C. $2^{i-1}+j$ D. $2^{i-1}+j-1$

以下排序方法, 那种是稳定的 (C)

- A. 希尔排序 B. 堆排序 C. 冒泡排序 D. 快速排序

排序的稳定性指的是对于原来所有的 $a[i]=a[j]$, $i < j$, 排序以后 $a[i]$ 的新位置仍然在 $a[j]$ 的前面。

关于“0”的原码、反码和补码描述正确的是 (C)

- A. “0”的原码只有一种表示方法
 B. “0”的反码只有一种表示方法
 C. “0”的补码只有一种表示方法
 D. “0”的原码、反码和补码均有两种表示方法

要使用 1280×1024 , 16 位真彩显示, 显存至少应为 (C) MB

- A. 1 B. 2 C. 4 D. 8

$1280 \times 1024 \times 2 \text{Byte} = 2.5 \text{MB}$

计算机能够自动工作，主要是因为采用了 (C)

- A. 二进制数制
- B. 高速电子元件
- C. 存储程序控制
- D. 程序设计语言

当计算机的主存储器的容量达到 1GB 的时候，其地址的表示至少需要 (C) 位

- A. 10
- B. 20
- C. 30
- D. 40

$1024 \times 1024 \times 1024 \text{Byte} = 2^{30} \text{Byte}$ ，每个字节的地址用一个数表示，所以需要 30 个位。

TCP/IP 协议中，不属于应用层的是 (D)

- A. WWW
- B. FTP
- C. SMTP
- D. TCP

一棵有 n 个节点的完全二叉树的高度是 (D)

- A. $n/2$
- B. $\log_2 n$
- C. $(\log_2 n)/2$
- D. $(\log_2 n) + 1$

借助一个栈，输入顺序是 123456，以下输出顺序不可能的是 (A)

- A. 142356
- B. 123654
- C. 231456
- D. 213546

对整数 $N=8934632178$ ，每次删除一个位置上的数字，使得新的数尽可能小，那么第四次删掉的数字是 (D)

- A. 6
- B. 8
- C. 7
- D. 4

二叉树 T ，设 n_0 ， n_1 和 n_2 分别表示度为 0，1 和 2 的顶点个数，则它们的关系是 (A)

- A. $n_0 = n_2 + 1$
- B. $n_1 = n_0 + 1$
- C. $n_2 = n_0 + 1$
- D. $n_2 = n_1 + 1$

中缀表达式 $A - (B + C/D) * E$ 的后缀表达式形式是 (D)

- A. $AB - C + D/E *$
- B. $ABC + D/-E *$
- C. $ABCD/E * + -$
- D. $ABCD/+E * -$

G 是一个非连通的无向图，共有 28 条边，则它至少有 (C) 个顶点

- A. 6
- B. 8
- C. 9
- D. 10

对 n 个元素从小到大排序，已将它们分成了 n/k 组，每组 k 个数。而且每组中的所有数都大于前一组的所有数。那么采用基于比较的排序，时间下界是 (B)

- A. $O(n \log n)$
- B. $O(n \log k)$
- C. $O(k \log n)$
- D. $O(k \log k)$

计算机是由 (D)、控制器、存储器、输入设备和输出设备构成的

A. ROM B. I/O C. CPU D. ALU

ALU 算术逻辑单元，即通常所说的运算器。

圆周上有 n 个点，任意两点间连一条弦，而且没有 3 条弦交于一点的情况，问在圆内一共有多少三角形。

$C(n, 3) + 4 * C(n, 4) + 5 * C(n, 5) + C(n, 6)$

ASCII 码的主要作用是 (A)

A. 方便信息交换 B. 方便信息存储 C. 便于管理 D. 便于输出

现在的计算机通常是将处理程序放在连续的内存地址中。CPU 在执行这个处理程序时，是使

用一个叫做 (D) 的寄存器来指示程序的执行顺序。

A. 累加寄存器 B. 指令寄存器 C. 内存地址寄存器 D. 指令地址寄存器

结构化程序设计的一种基本方法是 (B)

A. 归纳法 B. 逐步求精法 C. 递归法 D. 筛选法

二叉树后序遍历是 dabec，中序遍历是 debac，则后序遍历是 (D)

A. acbed B. decab C. deabc D. cedba

OSI 七层协议中，最底层是 ()。

(A) 会话层 (B) 数据链路层 (C) 物理层 (D) 网络层

<答案: C. 分别是物理层、数据链路层、网络层、传输层、会话层、表示层、应用层>

设 x 是值大于零的实型变量，计算 PASCAL 中 x^8 的表达式为 ()。

(A) $\ln(8 * \exp(x))$ (B) $\exp(8 * \ln(x))$ (C) x^8 (D) $\text{sqr}(\text{sqr}(\text{sqr}(x))) * x$

<答案: B. >

在微型计算机中，常用 () 码实现十进制数与二进制数之间的自动转换。

(A) BCD 码 (B) ASCII 码 (C) 海明码 (D) 机内码

<答案: A. >

已知 $A=11001010B$, $B=00001111B$, $C=01011100B$, $A \vee B \wedge C = () B$ 。

(A) 11001110 (B) 01110110 (C) 11101110 (D) 01001100

<答案: A. \vee 表示“或”， \wedge 表示“与”>

二叉树是重要的数据结构，5 个点的不同的二叉树有 () 个。

(A) 22 (B) 30 (C) 40 (D) 42

<答案: D. >

逻辑代数式子 $f = AB + ABC + AB(C + D)$ ，则 f 的简化式子为 ()。

(A) AB (B) $A + B$ (C) ABC (D) $ABCD$

<答案: A. >

插入排序是一种简单实用的工具,在对数组排序时,我们可能用二分查找,对要插入的元素快速找到在已经排好元素序列中的位置。下面的描述中正确的是()。(A) 二分查找的时间复杂度为 $O(\lg N)$, 因此排序的时间复杂度为 $O(N \cdot \lg N)$

(B) 二分查找的时间复杂度为 $O(N)$, 因此排序的时间复杂度为 $O(N \cdot \lg N)$

(C) 二分查找的时间复杂度为 $O(\lg N)$, 因此排序的时间复杂度为 $O(N \cdot N)$

(D) 二分查找的时间复杂度为 $O(N)$, 因此排序的时间复杂度为 $O(N \cdot N)$

<答案: C. >

有 5 本不同的数学书分给 5 个男同学,有 4 本不同的英语书分给 4 个女同学,将全部书收回后再重新发给他们,与原方案都不相同的方案有___种。

<答案: 1140480. >

十进制数 11/128 可用二进制数码序列表示为(D)。

A) 1011/1000000 B) 1011/100000000 C) 0.001011 D) 0.0001011

[x]补码=10011000, 其原码为(B)

A) 011001111 B) 11101000 C) 11100110 D) 01100101

下面哪些计算机网络不是按覆盖地域划分的(D)

A. 局域网 B. 都市网 C. 广域网 D. 星型网

设栈 S 和队列 Q 的初始状态为空,元素 e1, e2, e3, e4, e5, e6 依次通过栈 S, 一个元素出栈后即进入队列 Q, 若出队的顺序为 e2, e4, e3, e6, e5, e1, 则栈 S 的容量至少应该为(B)。

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

以下哪一个不是栈的基本运算(B)

A) 删除栈顶元素 B) 删除栈底的元素

C) 判断栈是否为空 D) 将栈置为空栈

在顺序表 (2, 5, 7, 10, 14, 15, 18, 23, 35, 41, 52) 中, 用二分查找 12, 所需的关键码比较的次数为(C)

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

某数列有 1000 个各不相同的单元, 由低至高按序排列; 现要对该数列进行二分查找 (binary-search), 在最坏的情况下, 需检视(B)个单元。

A. 1000 B. 10 C. 100 D. 500

线性表若采用链表存贮结构, 要求内存中可用存贮单元地址(D)

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

下列叙述中, 正确的是(D)

- A. 线性表的线性存储结构优于链表存储结构 B. 队列的操作方式是先进后出
C. 栈的操作方式是先进先出 D. 二维数组是指它的每个数据元素为一个线性表的线性表

设有一个共有 n 级的楼梯，某人每步可走 1 级，也可走 2 级，也可走 3 级，用递推公式给出某人从底层开始走全部楼梯的走法。例如：当 $n=3$ 时，共有 4 种走法，即 1+1+1, 1+2, 2+1, 3。

$$F(n) = f(n-1) + f(n-2) + f(n-3), n \geq 4;$$

$$F(1)=1; f(2)=2; f(3)=4;$$

有 $2 \times n$ 的一个长方形方格，用一个 1×2 的骨牌铺满方格。例如 $n=3$ 时，为 2×3 方格。此时用一个 1×2 的骨牌铺满方格，共有 3 种铺法：



试对给出的任意一个 $n(n \geq 0)$ ，求出铺法总数的递推公式。

$$F(1)=1 \quad F(2)=2 \quad F(n)=F(n-1)+F(n-2), n \geq 3$$

```
FUNCTION ACK(M, N: INTEGER): INTEGER;
BEGIN
  IF M=0 THEN ACK:=N+1
    ELSE IF N=0 THEN ACK:=ACK(M-1, 1)
      ELSE ACK:=ACK(M-1, ACK(M, N-1))
    END;
  BEGIN  WRITELN(ACK(3, 4));  READLN;  END.
```

输出
125

平面上有三条平行直线，每条直线上分别有 7, 5, 6 个点，且不同直线上三个点都不在同一条直线上。问用这些点为顶点，能组成多少个不同三角形？

$$C(7, 2) * (5+6) + C(5, 2) * (7+6) + C(6, 2) * (7+5) + 7*6*5 = 21*11 + 10*13 + 15*12 + 210 = 231 + 130 + 180 + 210 = 751$$

电线上停着两种鸟 (A, B)，可以看出两只相邻的鸟就将电线分为了一个线段。这些线段可分为两类；

一类是两端的小鸟相同；另一类则是两端的小鸟不相同。

已知：电线两个顶点上正好停着相同的小鸟，试问两端为不同小鸟的线段数目一定是 (B)。

- A. 奇数 B. 偶数 C. 可奇可偶 D. 数目固定

192.168.0.1 属于 (C)

- A. A 类地址 B. B 类地址 C. C 类地址 D. D 类地址

最高位 1..126 为 A 类，128..191 是 B 类，192..223 是 C 类。

关于“0”的原码、反码和补码描述正确的是 (C)

- A. “0”的原码只有一种表示方法
- B. “0”的反码只有一种表示方法
- C. “0”的补码只有一种表示方法
- D. “0”的原码、反码和补码均有两种表示方法

借助一个栈，输入顺序是 123456，以下输出顺序不可能的是 (A)

- A. 142356
- B. 123654
- C. 231456
- D. 213546

对整数 $N=8934632178$ ，每次删除一个位置上的数字，使得新的数尽可能小，那么第四次删掉的数字是 (D)

- A. 6
- B. 8
- C. 7
- D. 4

中缀表达式 $A-(B+C/D)*E$ 的后缀表达式形式是 (D)

- E. $AB-C+D/E*$
- F. $ABC+D/-E*$
- G. $ABCD/E*+-$
- H. $ABCD/+E*-$

已知 $A=11001010B$, $B=00001111B$, $C=01011100B$, $A \vee B \wedge C = () B$ 。

- (A) 11001110
- (B) 01110110
- (C) 11101110
- (D) 01001100

<答案: A. \vee 表示“或”， \wedge 表示“与”>

2. 128KB 的存储器用十六进制表示，它的最大的地址码是 (C)

- A) 10000
- B) EFFF
- C) 1FFFF
- D) FFFFF
- E) FFFF

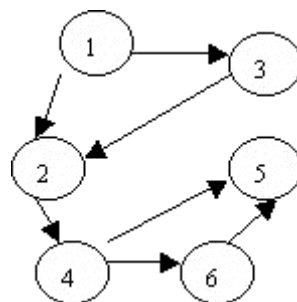
3. 能将高级语言程序转换为目标程序的是 (D)

- A) 调试程序
- B) 解释程序
- C) 编辑程序
- D) 编译程序
- E) 连接程序

9. 一棵 n 个结点的完全二叉树, 则二叉树的高度 h 为 (D).

- A) $n/2$
- B) $\log_2 n$
- C) $(\log_2 n)/2$
- D) $\lceil \log_2 n \rceil + 1$
- E) $2n-1$

10. 下图对该图进行广度优先拓扑排序得到的顶点序列正确的是 (C).



- A) 1, 2, 3, 4, 5, 6
- B) 1, 3, 2, 4, 5, 6

- C) 1, 3, 2, 4, 6, 5
- D) 1, 2, 3, 4, 6, 5,
- E) 1, 3, 2, 4, 5, 6

11. 下列属于冯·诺依曼计算机模型的核心思想是(ABC)。

- A) 采用二进制表示数据和指令;
- B) 采用”存储程序”工作方式
- C) 计算机硬件有五大部件(运算器、控制器、存储器、输入和输出设备)
- D) 结构化程序设计方法
- E) 计算机软件只有系统软件

14. 下面关于算法的正确说法是(ACDE)

- A) 算法必须有输出
- B) 算法必须在计算机上用某种语言实现
- C) 算法不一定有输入
- D) 算法必须在有限步执行后能结束
- E) 算法的每一步骤必须有确切的定义

15. 下列关于十进制数 100 的正确说法是(ABD)。

- A) 原码为 01100100B
- B) 反码为 64H
- C) 反码为 9BH
- D) 补码为 64H
- E) 补码为 9BH

19. 对于一个大小为 3 的栈，若输入顺序为 123456，则下列输出顺序有可能的是(AE)。

- A) 123456 B) 654321 C) 432165 D) 431256 E) 321654

20. 设有一个含有 13 个元素的 Hash 表 ($0 \sim 12$), Hash 函数是: $H(\text{key}) = \text{key} \% 13$, 其中 % 是求余数运算。用二次探查法解决冲突, 则对于序列 (8、31、20、33、18、53、27), 则下列说法正确的是(BCDE)。

- A) 27 在 1 号格子中
- B) 33 在 6 号格子中
- C) 31 在 5 号格子中
- D) 20 在 7 号格子中
- E) 18 在 4 号格子中

图灵 (Alan Turing) 是 (B)。

- A) 美国人 B) 英国人 C) 德国人 D) 匈牙利人
- E) 法国人

第一个给计算机写程序的人是 (B)。

- A) Alan Mathison Turing B) Ada Lovelace C) John von Neumann

D) John Mc-Carthy

E) Edsger Wybe Dijkstra

无向图G有16条边,有3个4度顶点、4个3度顶点,其余顶点的度均小于3,则G至少_____个顶点。

11

某年级学生共选修6门课程,期末考试前,必须提前将这6门课程考完,每人每天只在下午至多考一门课程,设6门课程为 $C_1, C_2, C_3, C_4, C_5, C_6$, $S(C_i)$ 为学习 C_i 的学生集合。已知 $S(C_i) \cap S(C_6) \neq \emptyset, i=1, 2, \dots, 5, S(C_i) \cap S(C_{i+1}) \neq \emptyset, i=1, 2, 3, 4, S(C_5) \cap S(C_1) \neq \emptyset$, 问至少安排_____天才能考完这6门课程。

4

一个家具公司生产桌子和椅子。现在有113个单位的木材。每张桌子要使用20个单位的木材,售价是30元;每张椅子要使用16个单位的木材,售价是20元。使用已有的木材生产桌椅(不一定要把木材用光),最多可以卖160元钱。

75名儿童到游乐场去玩。他们可以骑旋转木马,坐滑行铁道,乘宇宙飞船。已知其中20人这三种东西都玩过,55人至少玩过其中的两种。若每样乘坐一次的费用是5元,游乐场总共收入700,可知有10名儿童没有玩过其中任何一种。

已知a, b, c, d, e, f, g七个人中, a会讲英语; b会讲英语和汉语; c会讲英语、意大利语和俄语; d会讲汉语和日语; e会讲意大利语和德语; f会讲俄语、日语和法语; g会讲德语和法语。能否将他们的座位安排在圆桌旁,使得每个人都能与他身边的人交谈? 如果可以,请以“a b”开头写出你的安排方案: _____。

下列关于高级语言的说法错误的是(C)。

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

E. 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

设 $A = \text{true}$, $B = \text{false}$, $C = \text{false}$, $D = \text{true}$, 以下逻辑运算表达式值为真的是(D)。

A. $(A \wedge B) \vee (C \wedge D)$ B. $((A \wedge B) \vee C) \wedge D$ C. $A \wedge ((B \vee C) \wedge D)$

D. $(A \wedge (B \vee C)) \vee D$ E. $(A \vee B) \wedge (C \wedge D)$

其他问题类型

写运行结果

写运行结果的题，大家一定不要错过这个得分点。对于简单的问题（没有循环或者循环次数很少），机械的模拟是可行的，只要仔细即可。

```
var
  u: array [0..3] of integer;
  a, b, c, x, y, z: integer;
begin
  read(u[0], u[1], u[2], u[3]);
  a := u[0] + u[1] + u[2] + u[3] - 5;
  b := u[0] * (u[1] - u[2] div u[3] + 8);
  c := u[0] * u[1] div u[2] * u[3];
  x := (a + b + 2) * 3 - u[(c + 3) mod 4];
  y := (c * 100 - 13) div a div (u[b mod 3] * 5);
  if ((x+y) mod 2 = 0) then z := (a + b + c + x + y) div 2;
  z := (a + b + c - x - y) * 2;
  writeln(x + y - z);
end
```

输入: 2 5 7 4

输出: 263。

```
var
  i, number, ndata, sum: integer;
  data: array[1..100] of integer;
procedure solve(s, sign, n: integer);
var i: integer;
begin
  for i := s to ndata do begin
    inc(sum, sign * (number div (n * data[i])));
    solve(i + 1, -sign, n * data[i]);
  end;
end;
begin
  read(number, ndata);
  sum := 0;
  for i := 1 to ndata do read(data[i]);
  solve(1, 1, 1);
  writeln(sum);
end.
```

输入: 1000 3 5 13 11

输出： 328 。

几大方法

- a. 直接模拟
- b. 先模拟几次循环后找规律
- c. 直接看程序了解算法功能
- d. 了解程序本质后换一个方法解决
- e. 有时不知道算法可以通过观察猜出来
- f. 极少数的格子可以放弃

一般做这类题目的核心是找程序目的，即这个程序想干什么。很少有复杂的程序是“乱写”的，总有一点“写作目的”。抓住了它，不仅得出答案变得很容易了，而且对自己的结果也会比较有信心。机械模仿计算机硬算出结果的同学往往做的慢的多，而且容易失误。

(99 年分区联赛)

```
Program excpl;
var
  x, y, y1, jk, j1, g, e: Integer;
  a: array[1..20] of 0..9;
begin
  x:=3465; y:=264; jk:=20;
  for j1:=1 to 20 do a[j1]:=0;
  while y<>0 do
  begin
    y1:=y mod 10;
    y:=y div 10;
    while y1<>0 do
    begin
      g:=x;
      for e:=jk downto 1 do
      begin
        g:=g+a[e];
        a[e]:=g mod 10;
        g:=g div 10;
      end;
      y1:=y1-1
    end;
    jk:=jk-1
  end;
  j1:=1;
  while a[j1]=0 do j1:=j1+1;
  for Jk:=j1 to 20 do write(a[jk]:4);
  writeln;
end.
```

程序不长,但是有一定的难度。但其实有经验的话,看到 g 这个变量名和 $g:=g+a[e]; a[e]:=g \bmod 10$; 这几个标志句子。就可以一下子知道程序的用意了。高精度加法??? 对。

它执行了 $y1$ 次, $y1$ 每次都是 y 的个位... 程序就是在做 $x*y$

所以答案就是 $3465*264=914760$

再看它的输出格式,输出的应该是: 9 1 4 7 6 0

```
var n, jr, jw, jb: integer;
    ch1: char;
    ch: array[1..20] of char;
begin
    readln(n);
    for i:=1 to n do read(ch[i]);
    jr:=1; jw:=n; jb:=n;
    while (jr<=jw) do
        begin
            if(ch[jw]='R')
                then begin
                    ch1:=ch[jr]; ch[jr]:=ch[jw]; ch[jw]:=ch1; jr:=jr+1;
                end
            else if ch[jw]='W'
                then jw:=jw-1
                else begin
                    ch1:=ch[jw]; ch[jw]:=ch[jb]; ch[jb]:=ch1; jw:=jw-1; jb:=jb-1;
                end
        end;
    end;
    for i:=1 to n do write(ch[i]);
    writeln;
end.
输入:10
RBRBWWRBBR
输出:
RRRRWWBBBB
```

这道题的关键在于看出交换两个变量的块,以及 jr , jw 和 jb 的含义。整个过程有点像排序。

```
var
    a : array [1..50] of integer;
    n, i, sum : integer;
procedure work(p, r: integer);
```

```

var
  i, j, temp : integer;
begin
  if p < r then begin
    i := p - 1;
    for j := p to r - 1 do
      if a[j] >= a[r] then begin
        inc(i);
        temp := a[i]; a[i] := a[j]; a[j] := temp;
      end;
    temp := a[i + 1]; a[i + 1] := a[r]; a[r] := temp;
    work(p, i);
    work(i + 2, r);
  end;
end;

begin
  read(n);
  for i := 1 to n do read(a[i]);
  work(1, n);
  for i := 1 to n - 1 do sum := sum + abs(a[i + 1] - a[i]);
  writeln(sum);
end.

```

输入: 10 23 435 12 345 3123 43 456 12 32 -100

输出: 3223

关键在于先看出 **work** 是快速排序，其次最后计算 **sum** 的时候化简。

质数

```

Var I, j, s, spl:integer;
    p:boolean;
    a :array[1..10] of integer;
begin
  spl:=1;a[1]:=2;j:=2:
  while spl<10 do
  begin
    j :=j+1;p:=true;
    for i:=2 to j-1 do
      if(j mod i=0) then p:=false;
    if p then begin
      spl:=spl+1;a[spl]:=j;
    end;
  end;
end;

```

```

j:=2; p:=true;
while p do
begin
    s:=1;
    for i:=1 to j do s:=s*a[i];
    s:=s+1;
    for i:=2 to s-1 do
        if S mod i=0 then p:=false;
    j :=j+1;
end;
writeln(s);writeln;
end.

```

输出:

30031

```

Var d1,d2,X,Min:real;
begin
    min:=10000;X:=3;
    while X<15 do
    begin
        d1:=sqrt(9+(X-3)*(X-3));d2:=sqrt(36+(15-X)*(15-X));
        if(d1+d2)<Min then Min:=d1+d2;
        X:=x+0.001;
    end;
    writeln(Min:0:2);
end.

```

输出:

15.00

```

FUNCTION ACK(M, N: INTEGER):INTEGER;
BEGIN
    IF M=0 THEN ACK:=N+1
        ELSE IF N=0 THEN ACK:=ACK(M-1, 1)
            ELSE ACK:=ACK(M-1, ACK(M, N-1))
    END;
    BEGIN    WRITELN(ACK(3, 4));  READLN;  END.

```

输出

125

模拟


```

VAR P, Q, S, T: INTEGER;
BEGIN
  READLN(P);
  FOR Q:=P+1 TO 2*P DO
  BEGIN
    T:=0; S:=(P*Q)MOD(Q-P);
    IF S=0 THEN BEGIN T:=P+Q+(P*Q)DIV(Q-P); WRITE(T: 4); END;
  END;
END.

```

输入 12 输出

181 110 87 76 66 62 61 60

```

VAR I, J, H, M, N, K: INTEGER;
    B : ARRAY[1..10] OF INTEGER;
BEGIN
  READLN(N);
  FOR I:=1 TO 10 DO
  BEGIN
    M:=N; J:=1;
    WHILE M>0 DO {高精度分解}
      BEGIN J:=J+1; B[J]:=M MOD 10; M:=M DIV 10                      END;
    FOR H:=J TO 10 DO                      N:=N+B[H]; {数位累加}
    END;
  WRITELN(N);
END.

```

输入 1234 输出

1348

```

VAR X, Y1, Y2, Y3: INTEGER;
BEGIN
  READLN(X); Y1:=0; Y2:=1; Y3:=1;
  WHILE Y2<=X DO
  BEGIN
    Y1:=Y1+1; Y3:=Y3+2; Y2:=Y2+Y3; {y3 等差数列, y2 是 y3 求和}
  END;
  WRITELN(Y1);    {循环次数}
END.

```

输入: 23420 输出

153

```

VAR
  I, J, L, N, K, S, T: INTEGER;
  B: ARRAY[1..10] OF 0..9;

```

```

BEGIN
  READLN(L,N); S:=L; K:=1; T:=L;
  IF N>L THEN BEGIN
    WHILE S<N DO
      BEGIN K:=K+1;T:=T*L;S:=S+T END;
    S:=S-T;N:=N-S-1;
    FOR I:=1 TO 10 DO B[I]:=0;
    J:=11;
    WHILE N>0 DO
      BEGIN J:=J-1; B[J]:=N MOD L; N:=N DIV L END; {进制转换}
    FOR I:=10-K+1 TO 10 DO WRITE(CHR(ORD(' A ')+B[I]));
    READLN;
    END
  ELSE WRITELN(CHR(ORD(' A ')+N-1))
END.
输入:  4   167      输出
BBAC

```

```

const
  u: array[0..2] of integer = (1, -3, 2);
  v: array[0..1] of integer = (-2, 3);
var
  i, n, sum: integer;
function g(n: integer): integer;
var i, sum: integer;
begin
  sum := 0;
  for i := 1 to n do inc(sum, u[i mod 3] * i);
  g := sum;
end;
begin
  sum := 0;
  read(n);
  for i := 1 to n do inc(sum, v[i mod 2] * g(i));
  writeln(sum);
end.
输入: 103
输出:  -400 。

```

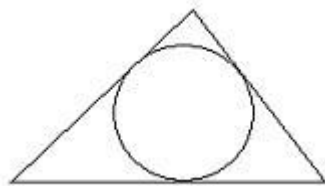
程序填空

数学

三角形内切圆的面积

题目描述：

给出三角形三边的边长，求此三角形内切圆（如下图所示，三角形的内切圆是和三角形三边都相切的圆）的面积。



输入：

三个正实数 a 、 b 、 c （满足 $a+b>c$ ， $b+c>a$ ， $c+a>b$ ），表示三角形三边的边长。

输出：

三角形内切圆的面积，结果四舍五入到小数点后面2位。

输入样例：

3 4 5

输出样例：

3.14

程序：

```
program program1;
var
    a, b, c, r, s, t: real;
begin
    read(a, b, c);
    s := (①) / 2;
    t := ②(s * (s - a) * (s - b) * (s - c));
    r := t / s;
    writeln(3.1415927 * r * ③ : 0 : ④);
end.
```

- ① $a+b+c$
- ② sqrt
- ③ r
- ④ 2

贪心

问题描述:工厂在每天的生产中,需要一定数量的零件,同时也可以知道每天生产一个零件的生产单价。在 N 天的生产中,当天生产的零件可以满足当天的需要,若当天用不完,可以放到

下一天去使用,但要收取每个零件的保管费,不同的天收取的费用也不相同。

问题求解:求得一个N天的生产计划(即N天中每天应生产零件个数),使总的费用最少。

输入:N(天数 $N \leq 29$)

每天的需求量(N个整数)

每天生产零件的单价(N个整数)

每天保管零件的单价(N个整数)

输出:每天的生产零件个数(N个整数)

例如:当N=3时,其需要量与费用如下:

	第一天	第二天	第三天
需要量	25	15	30
生产单价	20	30	32
保管单价	5	10	0

生产计划的安排可以有許多方案,如下面的三种:

第一天	第二天	第三天	总的费用
25	15	30	$25*20+15*30+30*32=1910$
40	0	30	$40*20+15*5+30*32=1835$
70	0	0	$70*20+45*5+30*10=1925$

程序说明:(应该特别注意)

b[n]:存放每天的需求量

c[n]:每天生产零件的单价

d[n]:每天保管零件的单价

e[n]:生产计划

程序:

Program exp5;

Var

i, j, n, yu, j0, j1, s: integer;

b, c, d, e: array[0..30] of integer;

begin

readln(n);

for i:=1 to n do readln(b[[i], c[i], d[i]]);

for i:=1 to n do e[i]:=0;

①:=10000; c[n+2]:=0; b[n+1]:=0; j0:=1;

while (j0<=n) do

begin

yu:=c[j0]; j1:=j0; s:=b[j0];

while ② do

begin

```

        ③ j1:=j1+1;s:=s+b[j1];
    end;
    ④ j0:=j1+1;
end;
for i:=1 to n do ⑤
    readln;
end.

```

- 1、c[n+1]
- 2、(yu+d[j1]<c[j1+1])
- 3、yu:=yu+d[j1];
- 4、e[j0]:=s;
- 5、write(e[i]:4);

二分查找

这道题是二分查找答案，并每次判断其可行性。

题目描述：

木材厂有一些原木，现在想把这些木头切割成一些长度相同的小段木头（木头有可能有剩余），需要得到的小段的数目是给定的。当然，我们希望得到的小段越长越好，你的任务是计算能够得到的小段木头的最大长度。木头长度的单位是cm。原木的长度都是正整数，我们要求切割得到的小段木头的长度也是正整数。

输入：

第一行是两个正整数N和K($1 \leq N \leq 10000$, $1 \leq K \leq 10000$)，N是原木的数目，K是需要得到的小段的数目。

接下来的N行，每行有一个1到10000之间的正整数，表示一根原木的长度。

输出：

输出能够切割得到的小段的最大长度。如果连1cm长的小段都切不出来，输出”0”。

输入样例：

```

3 7
232
124
456

```

输出样例：

```

114

```

程序：

```

var
    n, k : integer;
    len : array [1..10000] of integer;
    i, left, right, mid : integer;

```

```

function isok(t : integer) : boolean;
var
    num, i : integer;
begin
    num := 0;
    for i := 1 to n do begin
        if num >= k then break;
        num := ① ;
    end;
    if ② then isok := true
    else isok := false;
end;

begin
    readln(n, k);
    right := 0;
    for i := 1 to n do begin
        readln(len[i]);
        if right < len[i] then right := len[i];
    end;
    inc(right); ③ ;
    while ④ < right do begin
        mid := (left + right) div 2;
        if ⑤ then right := mid
        else left := mid;
    end;
    writeln(left);
end.

```

- ① num + len[i] div t
- ② num >= k
- ③ left := 0
- ④ left + 1
- ⑤ not isok(mid) （或者 isok(mid) = false）

回溯法

问题描述:有 n 种基本物质 ($n \leq 10$), 分别记为 P_1, P_2, \dots, P_n , 用 n 种基本物质构造一种物品, 物品使用在 k 个不同地区 ($k \leq 20$), 每个地区对物品提出自己的要求, 这些要求用一个 n 位的数表示: $a_1 a_2 \dots a_n$, 其中:

$a_i = 1$ 表示必须有第 i 种基本物质

$= -1$ 表示必须不能有第 i 种基本物质

$= 0$ 无所谓

问题求解:当 k 个不同地区要求给出之后,给出一种方案,指出哪些物质被使用,哪些物质不被使用。

程序说明:数组 $b[1], b[2], \dots, b[n]$ 表示某种物质是否需要

$a[1..k, 1..n]$ 记录 k 个地区对物质的要求, 其中:

$a[i, j]=1$ 表示第 i 个地区对第 j 种物质是需要的

$a[i, j]=0$ 表示第 i 个地区对第 j 种物质是无所谓的

$a[i, j]=-1$ 表示第 i 个地区对第 j 种物质是不需要的

程序:

```
program gxp2;
Var i, j, k, n :integer;
    p:boolean;
    b :array [0..20] of 0..1;
    a :array[1..20, 1..10] of integer;
begin
  readln(n, k);
  for i:=1 to k do
  begin
    for j:=1 to n do read(a[i, j]);
    readln;
  end;
  for i:=0 to n do b[i]:=0;
  p:=true;
  while  do
  begin
    j:=n;
    while b[j]=1 do j:=j-1;
    
    
    for i:=j+1 to n do b[i]:=0;
    for i:=1 to k do
      for j:=1 to n do
        if( a[i, j]=1 ) and (b[j]=0) or 
          then p:=true; {明确 p 的含义}
    end;
    if 
    then writeln('找不到! ')
    else for i:=1 to n do
```

```

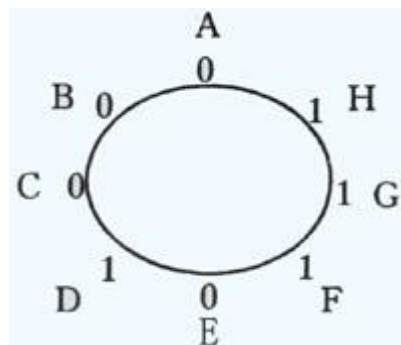
        if (b[i]=1) then writeln('物质',i,' 需要')
            else writeln('物质',i,' 不需要');
end.

```

- 1、P AND (B[0]=0)
- 2、B[J] :=1;
- 3、P :=FALSE;
- 4、(A[I, J]=-1) AND (B[J]=1)
- 5、P

将 2^n 个 0 和 2^n 个 1，排成一个圈。从任一个位置开始，每次按逆时针的方向以长度为 $n+1$ 的单位进行数二进制数。要求给出一种排法，用上面的方法产生出来的 2^{n+1} 个二进制数都不相同。

例如，当 $n=2$ 时，即 2^2 个 0 和 2^2 个 1 排成如下一圈：



比如，从 A 位置开始，逆时针方向取三个数 000，然后再从 B 位置上开始取三个数 001，接着从 C 开始取三个数 010，…可以得到 000，001，010，101，011，111，110，100 共 8 个二进制数且都不相同。

程序说明 {重要，可以先自己设计算法}

以 $N=4$ 为例，即有 16 个 0，16 个 1，数组 A 用以记录 32 个 0，1 的排法，数组 B 统计二进制数是否已出现过。

程序清单

```

VAR
  A :ARRAY[1..36] OF 0..1;
  B :ARRAY[0..31] OF INTEGER;
  I, J, K, S, P: INTEGER;
BEGIN
  FOR I:=1 TO 36 DO A[I]:=0;

```



```

FOR I:=28 TO 32 DO A[I]:=1;
P:=1;A[6]:=1;
WHILE (P=1) DO
BEGIN
    J:=27;
    WHILE A[J]=1 DO J:=J-1;
    ( ① )
    FOR I:=J+1 TO 27 DO ( ② )
    FOR I:=0 TO 31 DO B[I]:=0;
    FOR I:=1 TO 32 DO
    BEGIN
        ( ③ )
        FOR K:=I TO I+4 DO S:=S*2+A[K];
        ( ④ )
    END;
    S:=0;
    FOR I:=0 TO 31 DO S:=S+B[I];
    IF ( ⑤ ) THEN P:=0
END;
FOR I:=1 TO 32 DO FOR J:=I TO I+4 DO WRITE(A[J]);
WRITELN
END.

```

- 1、A[J]:=1;
- 2、A[I]:=0;
- 3、S:=0;
- 4、B[S]:=1;
- 5、S=32

问题描述:将 n 个整数分成 k 组 ($k \leq n$, 要求每组不能为空), 显然这 k 个部分均可得到一个各自的和 s_1, s_2, \dots, s_k , 定义整数 P 为:

$$P = (S_1 - S_2)^2 + (S_1 - S_3)^2 + \dots + (S_1 - S_k)^2 + (S_2 - S_3)^2 + \dots + (S_{k-1} - S_k)^2$$

问题求解:求出一种分法, 使 P 为最小(若有多种方案仅记一种)

程序说明:

数组: $a[1], a[2], \dots, a[N]$ 存放原数
 $s[1], s[2], \dots, s[K]$ 存放每个部分的和
 $b[1], b[2], \dots, b[N]$ 穷举用临时空间
 $d[1], d[2], \dots, d[N]$ 存放最佳方案

程序:

```

program exp4;
Var i, j, n, k : integer;
a :array [1..100] of integer;

```

```

b,d:array [0..100] of integer;
s :array[1..30] of integer;
begin
  readln(n,k);
  for I:=1 to n do read(a[I]);
  for I:=0 to n do b[I]:=1;
  cmin:=1000000;
  while (b[0]=1) do
  begin
    for I:=1 to k do ①
    for I:=1 to n do
      ②
    sum:=0;
    for I:=1 to k-1 do
      for j:= ③
        sum:=sum+(s[I]-s[j])*(s[I]-s[j]);
    if ④ then
    begin
      cmin:=sum;
      for I:=1 to n do d[I]:=b[I];
    end;
    j:=n;
    while ⑤ do j:=j-1;
    b[j]:=b[j]+1;
    for I:=j+1 to n do ⑥
  end;
  writeln(cmin);
  for I:=1 to n do write(d[I]:40);
  writeln;
end.

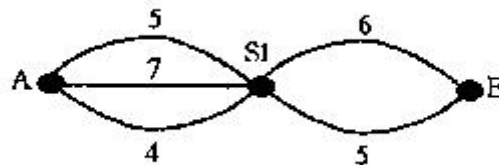
```

- 1、 s[k]:=0;
- 2、 s[b[i]]:= s[b[i]]+a[i];
- 3、 i+1 to k
- 4、 sum<cmin
- 5、 b[j]=k
- 6、 b[i]:=1;

在 A, B 两个城市之间设有 N 个路站 (如下图中的 S1, 且 $N < 100$), 城市与路站之间、路站和路站之间各有若干条路段 (各路段数 ≤ 20 , 且每条路段上的距离均为一个整数)。

A, B 的一条通路是指: 从 A 出发, 可经过任一路段到达 S1, 再从 S1 出发经过任一路段, ... 最后到达 B。通路上路段距离之和称为通路距离 (最大距离 ≤ 1000)。当所有的路段距离给出之后, 求出所有不同距离的通路个数 (相同距离仅记一次)。

例如: 下图所示是当 $N=1$ 时的情况:



从 A 到 B 的通路条数为 6, 但因其中通路 $5+5=4+6$, 所以满足条件的不同距离的通路条数为 5。

算法说明: 本题采用穷举算法。

数据结构: N: 记录 A, B 间路站的个数

数组 D[I, 0] 记录第 I-1 到第 I 路站间路段的个数

D[I, 1], D[I, 2], ... 记录每个路段距离

数组 G 记录可取到的距离

程序清单:

PROGRAM CHU7_6;

VAR I, J, N, S: INTEGER;

B: ARRAY[0..100] OF INTEGER;

D: ARRAY[0..100, 0..20] OF INTEGER;

G: ARRAY[0..1000] OF 0..1;

BEGIN

READLN(N);

FOR I:=1 TO N+1 DO

BEGIN

READLN(D[I, 0]);

FOR J:=1 TO D[I, 0] DO READLN(D[I, J]);

END;

D[0, 0]:=1;

FOR I:=1 TO N+1 DO B[I]:=1;

B[0]:=0;

FOR I:=0 TO 1000 DO G[I]:=0;

WHILE ① DO b[0]=0

BEGIN

S:=0;

FOR I:=1 TO N+1 DO

S:= ② s+d[i, b[i]];

G[S]:=1; J:=N+1;

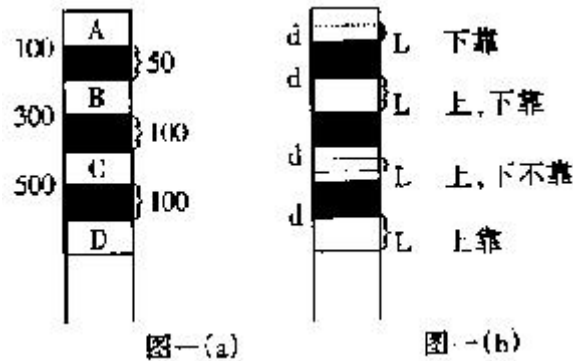
WHILE ③ DO J:=J-1; b[j]=d[j, 0]

B[J]:=B[J]+1;

```
FOR I:=J+1 TO N+1 DO          B[I]:=1;
END;
S:=0;
FOR I:=1 TO 1000 DO
  ④;      .....S:=s+g[i]
WRITELN(S); READLN;
END.
```

模拟

存储空间的回收算法。设在内存中已经存放了若干个作业 A, B, C, D。其余的空间为可用的(如图一中(a))。



此时, 可用空间可用一个二维数组 $dk[1..100, 1..2]$ 表示, (如下表一中(a)), 其中: $dk[i, 1]$ 对应第 i 个可用空间首址, $dk[i, 2]$ 对应第 i 个可用空间长度如上图, dk :

	0	0
100	50	50
300	100	100
500	100	100
10000	0	0

现某个作业释放一个区域, 其首址为 d , 长度为 L , 此时将释放区域加入到可用空间表中。要求在加入时, 若可用空间相邻时, 则必须进行合并。因此出现下面的 4 种情况:

- (1) 下靠, 即回收区域和下面可用空间相邻, 例如, $d=80, L=20$, 此时成为表二中的(a)。
- (2) 上靠, 例如, $d=600, L=50$, 此时表成为表二中的(b)。
- (3) 上、下靠, 例如, $d=150, L=150$, 此时表成为表二中的(c)。
- (4) 上、下不靠, 例如, $d=430, L=20$, 此时表成为表二中的(d)。

80	70	100	50		100	50	
300	100	300	100	100	300	100	
50	100	500	150	500	100	430	20
						500	100

表二(a)(下靠) 表二(b)(上靠) 表二(c)(上, 下靠) 表二(d)(上, 下不靠)

程序说明：对数组 dk 预置 2 个标志，即头和尾标志，成为表一中(b)，这样可使算法简单，sp 为 dk 表末地址。

程序清单：

```

VAR      I, J, SP, D, L: INTEGER;
      DK      : ARRAY[0..100, 1..2] OF INTEGER;
BEGIN
  READLN(SP);
  FOR I:=1 TO SP DO
    READLN(DK[I, 1], DK[I, 2]);
    DK[0, 1]:=0; DK[0, 2]:=0; ①;
    DK[SP, 1]:=10000; DK[SP, 2]:=0; READLN(D, L); I:=1;
  WHILE DK[I, 1]<D DO I:=I+1; ②;
  IF (DK[I, 1]+DK[I, 2]=D) THEN {与前面的一段靠在一起}
    IF (D+L=DK[I+1, 1]) THEN {和后面的一段也靠在一起，即
上下靠}
      BEGIN
        DK[I, 2]:=③;
        FOR J:=I+1 TO SP-1 DO {删除 dk[i+1]}
          DK[J]:=DK[J+1];
        SP:=SP-1;
      END
      ELSE DK[I, 2]:=DK[I, 2]+L {仅和前面靠}
    ELSE IF (D+L=DK[I+1, 1]) THEN {仅和后面靠}
      BEGIN
        DK[I+1, 1]:=④; DK[I+1, 2]:=DK[I+1, 2]+L
      END
    ELSE BEGIN {都不靠，插入}
      FOR J:=SP DOWNT0 I+1 DO DK[J+1]:=DK[J];
      ⑤:=D; DK[I+1, 2]:=L; SP:=SP+1;
    END;
  FOR I:=1 TO SP-1 DO WRITELN(DK[I, 1]:4, DK[I, 2]:4); READLN;
END.

```

1、SP:=SP+1

2、I:=I-1

3、DK[I, 2]+L+DK[I+1, 2]

4、D

5、DK[I+1, 1]

Joseph

题目描述：

原始的Joseph问题的描述如下：有n个人围坐在一个圆桌周围，把这n个人依次编号为1, ..., n。从编号是1的人开始报数，数到第m个人出列，然后从出列的下一个重新开始

报数，数到第 m 个人又出列，...，如此反复直到所有的人全部出列为止。比如当 $n=6$ ， $m=5$ 的时候，出列的顺序依次是5，4，6，2，3，1。

现在的问题是：假设有 k 个好人和 k 个坏人。好人的编号的1到 k ，坏人的编号是 $k+1$ 到 $2k$ 。我们希望求出 m 的最小值，使得最先出列的 k 个人都是坏人。

输入：

仅有的一个数字是 k ($0 < k < 14$)。

输出：

使得最先出列的 k 个人都是坏人的 m 的最小值。

输入样例：

4

输出样例：

30

程序：

```
program program2;
var
    i, k, m, start: longint;
    find: boolean;
function check(remain: integer): boolean; {remain 目前剩余人数}
var result: integer;
begin
    result := (  ① ) mod remain; {result 这次出列的人的编号}
    if (  ② ) then begin
        start := result; check := true; {符合要求，为下一次做准备}
    end
    else check := false; {不符合要求}
end;
begin
    find := false;
    read(k);
    m := k; {枚举 m}
    while (  ③ ) do begin
        find := true; start := 0; {人从 0 开始编号}
        for i := 0 to k-1 do {模拟 k 次出列}
            if ( not check(  ④ ) ) then begin
                find := false; break; {不符合要求，停止此次，直接枚举下一个 m}
            end;
        inc(m);
    end;
    writeln(  ⑤ );
end.
```

① start+m-1

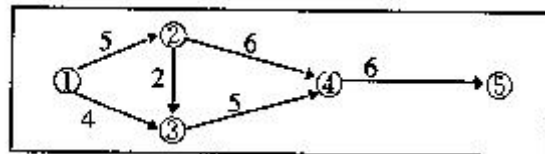
② result>=k (或者 k<=result)

- ③ not find (或者 find=false)
- ④ $2*k-i$
- ⑤ $m-1$

关键路径

设有一个工程网络如下图表示(无环路的有向图):

其中, 顶点表示活动, ①表示工程开始, ⑤表示工程结束(可变, 用 N 表示), 边上的数字表示活动延续的时间。



如上图中, 活动①开始 5 天后活动②才能开始工作, 而活动③则要等①、②完成之后才能开始, 即最早也要 7 天后才能工作。

在工程网络中, 延续时间最长的路径称为关键路径。上图中的关键路径为:

①—②—③—④—⑤共 18 天完成。

关键路径的算法如下:

1. 数据结构: {重要定义}

$R[1..N, 1..N]$ OF INTEGER; 表示活动的延续时间, 若无连线, 则用 -1 表示;
 $EET[1..N]$ 表示活动最早可以开始的时间
 $ET[1..N]$ 表示活动最迟应该开始的时间

关键路径通过点 J, 具有如下的性质: $EET[J]=ET[J]$

2. 约定:

结点的排列已经过拓扑排序, 即序号前面的结点会影响序号后面结点的活动。

程序清单:

```

VAR I, J, N, MAX, MIN, W, X, Y: INTEGER;
    R: ARRAY[1..20, 1..20] OF INTEGER;
    EET, ET: ARRAY[1..20] OF INTEGER;
BEGIN
    READLN(N)
    FOR I:=1 TO N DO
        FOR J:=1 TO N DO
            R[I, J]:=-1;
        READLN(X, Y, W); {输入从活动 X 到活动 Y 的延续时间, 以 0 为结束}
    WHILE X<>0 DO
        BEGIN
            R[X, Y]:=W; ①
        END;
    EET[1]:=0; {认为工程从 0 天开始}
    FOR I:=2 TO N DO
        BEGIN
            MAX:=0;

```

```

FOR J:=1 TO N DO
  IF R[J, I]<>-1 THEN
    IF ② THEN MAX:=R[J, I]+EET[J]; {模式}
  EET[I]:=MAX;
END;
③
FOR I:=N-1 DOWNTO 1 DO
  BEGIN
    MIN:=10000;
    FOR J:=1 TO N DO
      IF R[I, J]<>-1 THEN
        IF ④ THEN MIN:=ET[J] - R[I, J]; {模式}
      ET[I]:=MIN;
    END;
    WRITELN(EET[N]);
    FOR I:=1 TO N-1 DO
      IF ⑤ THEN WRITE(I, '→'); {关键路径定义}
    WRITE(N); READLN
  END.

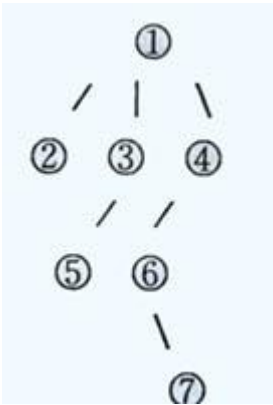
```

- 1、READLN(X, Y, W)
- 2、R[J, I]+EET[J]>MAX
- 3、ET[N]:=EET[N];
- 4、ET[J]-R[I, J]<MIN
- 5、EET[I]=ET[I]

搜索

求出一棵树的深度和宽度。例如有如下的一棵树：

其树的深度为从根结点开始到叶结点结束的最大深度，树的宽度为同一层上结点数的最大值。在上图中树的深度为 4，宽度为 3。



用邻接表来表示树，上图中的树的邻接表示如下：

1	2	3	4	0	0
2	0	0	0	0	0
3	5	0	0	0	0
4	6	0	0	0	0
5	0	0	0	0	0
6	7	0	0	0	0
7	0	0	0	0	0

程序清单

```
VAR
    I, J, SP1, SP2, L, MAX: INTEGER;
    TREE:ARRAY[1..20, 1..6] OF INTEGER;
    Q:ARRAY[1..100, 0..6] OF INTEGER;
    D:ARRAY[0..20] OF INTEGER;
BEGIN
    FOR I:=1 TO 14 DO FOR J:=1 TO 6 DO TREE[I, J]:=0;
    FOR J:=1 TO 14 DO TREE[J, 1]:=J;
```

```

TREE[1,2]:=2; TREE [1,3]:=3; TREE[1,4]:=4; TREE[2,2]:=5;

TREE[2,3]:=6; TREE [3,2]:=7; TREE[3,3]:=8; TREE[4,2]:=9;

TREE[4,3]:=10; TREE[4,4]:=11; TREE[7,2]:=12;

TREE[7,3]:=13; TREE[13,2]:=14;

SP1:=1; SP2:=1;

FOR I:=1 TO 6 DO Q[1,I]: =TREE[1, I];

Q[1,0]:=1;

WHILE ( ① ) DO

BEGIN

    L:= ( ② ) ; J:=2;

    WHILE ( ③ ) DO

    BEGIN

        SP2:=SP2+1;Q[SP2,0]:=L;Q[SP2,1]:=Q[SP1,J];

        FOR I:=2 TO 6 DO

            Q[SP2,I]:=TREE[Q[SP1,J],I];

        J:=J+1

    END;

    SP1:=SP1+1

END;

Writeln ( ④ ) ;

FOR I:=0 TO 20 DO D[I]:=0;

FOR I:=1 TO SP2 DO

D[Q[I,0]]:= ( ⑤ ) ;

MAX:=D[1];

```

```

FOR I:=2 TO 20 DO

    IF D[I]>MAX THEN MAX:=D[I];

WRITELN(MAX);

READLN;

END.

```

- 1、SP1<=SP2
- 2、Q[SP1, 0]+1
- 3、Q[SP1, J]<>0
- 4、Q[SP2, 0]
- 5、D[Q[I, 0]]+1

翻硬币

题目描述：

一摞硬币共有 m 枚，每一枚都是正面朝上。取下最上面的一枚硬币，将它翻面后放回原处。然后取下最上面的 2 枚硬币，将他们一起翻面后放回原处。在取 3 枚，取 4 枚……直至 m 枚。然后在从这摞硬币最上面的一枚开始，重复刚才的做法。这样一直做下去，直到这摞硬币中每一枚又是正面朝上为止。例如， m 为 1 时，翻两次即可。

输入：仅有的一个数字是这摞硬币的枚数 m ， $0 < m < 1000$ 。

输出：为了使这摞硬币中的每一枚都是朝正面朝上所必须翻的次数。

输入样例：30

输出样例：899

程序：

```

program Program1;
var    m:integer;
function solve(m: integer):integer;
    var i,t,d: integer;
        flag: Boolean;
    begin
        if (m = 1) then
            solve :=          (1)
        else begin

```


∖

这个世界的地理学家们为了方便纪录，给 OIM 所有可能的形状用正整数编好号，而且每个正整数恰好对应一种山形。他们规定，若两座山的宽度不同，则较宽的编号较大；若宽度相同，则比较从左边开始第 1 个坡度不同的地方，坡度上升的编号较大。以下三座 OIM 的编号有小到大递增：

 /\ /\ /\ /\

/ \ \ / \ \ \ / \ \ \。显然/\的编号为 1。但是地理学家在整理纪录是发觉，查找编号与山形的对应关系不是很方便。他们希望能快速地从编号得到山的形状。你自告奋勇答应他们写一个程序，输入编号，能马上输出山形。

输 入： 一个编号（编号大小不超过 600,000,000），

输 出： 输入编号所对应的山形，1 座山所占行数恰为它的高度，即山顶上不能有多余空行。

输入样例：15

输出样例： /\ /\

 / \ \ \

程 序：

```
program Program2;

const

    L:integer =19;      SZ: integer =50;

    UP: char = '/' ;    DN: char = '\';

Var

    i,nth,x,y,h,e,f:integer;

    m: array[0..1,0..38,0..19] of integer;

    pic: array[0..49,0..49] of char;

procedure init;
```

```

var k, s, a, b, c: integer;

begin

    for a:=0 to 1 do

        for b:=0 to 2*L do

            for c:=0 to L do

m[a, b, c]:=0;      m[0, 0, 0]:=1;

                for k:=0 to 2*L-1 do

                    begin

                        for s:=1 to L do

                            begin

                                m[0, k+1, s] := m[0, k, s+1] + m[1, k, s+1];

m[1, k+1, s]:=      (1)          ;

                            end;

                                m[0, k+1, 0] :=m[0, k, 1]+m[1, k, 1];

                                end;

                    end;

                end;

end;

procedure draw(k, s, nth:integer);

begin

    if (k=0) then exit;

    if ((nth-m[1, k, s])>=0) then

        begin

            nth:=nth-m[1, k, s];

            if (y>h) then          (2)          ;

```

```

                                pic[y, x]:=UP;    y:=y+1;    x:=x+1;    draw(
);
                                end
                                else begin
                                y:=y -
1;    pic[y, x]:=DN;    x:=x+1;    draw(k-1, s-1, nth);
                                end;
                                end;

begin
    init;
    read(nth);
    for e:=0 to SZ-1 do
        for f:=0 to SZ-1 do
            pic[e, f]:= ' ' ;
        x:=0;
        y:=0
        h:=0;
        i:=0;

        while ((nth-m[0, 2*i, 0])>=0) do
            begin
                nth:= nth-m[0, 2*i, 0];
                                (4)
                                ;
            end;

```

```

draw(                                     (5)                                     );

for i:=h downto x-1 do

begin

    for e:=0 to x-1 do

        write(pic[i,e]);

        writeln(' ');

    end;

end.

```

(1) $m[0, k, s-1] + m[1, k, s-1]$

(2) $h:=y$

(3) $k-1, s+1, nth$

(4) $i:=i+1$

(5) $2*i, 0, nth$

练习

下面四个不同进制的数中，最小的一个是_____。

(A) $(11011001)_2$ (B) $(75)_{10}$ (C) $(37)_8$ (D) $(A7)_{16}$

如果 $52-19=33$ 是成立的，则 52、19、33 分别是_____。

(A) 八进制、十进制、十六进制 (B) 十进制、十六进制、八进制
(C) 八进制、十六进制、十进制 (D) 十进制、八进制、十六进制

把下列二进制数分别化成八进制数、十六进制数和十进制数。

(1) 1110B (2) -101010B (3) 10.0101B (4) 101101.11B

把下列十进制数转换成二进制数（按 0 舍 1 入取 6 位二进制小数）。

(1) 75 (2) 1024 (3) 0.2 (4) 18.692

用 8 位二进制定点整数或定点小数写出下列真值的原码、补码形式，然后用 2 位十六进制数表示。

- (1) 11001B (2) -10010B (3) 100000B (4) -100000B (5) 0.1B
 (6) -0.1B (7) 0.100111B (8) -0.100111B (9) -15/128D

已知 x 的补码，写出补码的十六进制表示，再求出 x 的原码。

- (1) $[x]_{\text{补}} = 01010011\text{B}$ (2) $[x]_{\text{补}} = 10001001\text{B}$
 (3) $[x]_{\text{补}} = 11111111\text{B}$ (4) $[x]_{\text{补}} = 11000000\text{B}$

已知 $[x]_{\text{原}} = 10011011$ 是定点纯小数，写出 x 的浮点数规格化形式。设其阶码是 4 位补码，尾数是 8 位原码。

1. 数组 $A[30..100, 20..100]$ 以行优先的方式存储，每个元素占 8 个字节，且已知 $A[40, 30]$ 的地址为 2000，则 $A[60, 90]$ 的地址为：_____ 如果以列优先存储，则为：_____

考查了数据结构中数组存储方式。

~~~~~

2. 设栈 S 的初始状态为空，现有 6 个元素组成的序列 {1, 3, 5, 7, 9, 11}，对该序列在 S 栈上依次进行如下操作（从序列中的 1 开始，出栈后不在进栈）：进栈，出栈，进栈，进栈，进栈，进栈，出栈，进栈，问出栈的元素序列是：\_\_\_\_\_，栈顶指针的值为\_\_\_\_\_ 栈顶元素为：\_\_\_\_\_

考查了数据结构中的栈。

~~~~~

3. 把中缀表达式写成后缀及前缀表达式

(1) $(P+Q)*(A-B)/((C+D)/(E-F))-G$

后：_____

前：_____

(2) $A-C*D+B/E*(D/A)$

后：_____

前：_____

4. 根据后缀表达式，写出前缀及中缀表达式

$ABC/DE+GH-/*+$

前：_____

中：_____

这两题实际上考查了数据结构中的表达式树

~~~~~

5. 用一个字节来表示整数，最高位用作符号位（1 为正，0 为负），其他位表示数值，

(1) 这样的表示法称为原码表示法，表示数的范围为：\_\_\_\_\_

(2) 原码表示法，将出现\_\_\_\_\_有两种表示

(3) 实际上计算机中是用补码表示数，其表示范围为：\_\_\_\_\_

考查了数的原码，补码表示。

6. 已知  $N*N$  个数据成方阵排列：

$A_{11} A_{12} A_{13} \dots A_{1n}$

$A_{21} A_{22} A_{23} \dots A_{2n}$

...

$A_{n1} A_{n2} A_{n3} \dots A_{nn}$

已知  $A_{ij} = A_{ji}$ ,

(1)将 A11, A21, A22, A31, A32, A33... 存储到一维数组 A(1), A(2), A(3)... A(K)

给出 i, j 写出求 K 的表达式:\_\_\_\_\_

(2)将 A11, A12, ... A1n, A22, A23, ... A2n, A33... Ann 存储到一维数组 A(1), A(2), A(3)... A(K), 给出 i, j 写出求 K 的表达式:\_\_\_\_\_

7. 已知一个数列 U1, U2, U3... Un..., 往往可以找到一个最小的 K 值和 K 个数 a1, a2, ..., ak, 使得数列从某项开始都满足:  $U(n+k) = a_1 * U(n+k-1) + a_2 * U(n+k-2) + \dots + a_k * U(n)$  (式 A) 例如数列 1, 1, 2, 3, 5... 可以发现: 当 K=2, a1=1, a2=1 时, 从第 3 项起 (N>=1) 满足:  $U(n+2) = U(n+1) + U(n)$  试对数列  $1^3, 2^3, 3^3, \dots, N^3, \dots$ , 求 K 和 a1, a2, ..., ak, 使得式 A 成立.

实质是考数学。

8. 给出一棵二叉树的中序遍历: DBGEACHFI 与后序遍历: DGEBHIFCA, 画出此二叉树

9. 给出二叉树的前序遍历与后序遍历, 能确定一棵二叉树吗, 举例说明.

10. 下面是一个利用完全二叉树特性, 用顺序表来存储的一个二叉树, 结点数据为字符型 (结点层次从小到大, 同一层从左到右顺序存储, #表示空结点, @表示存储数据结束)

结点 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

数据 A B C # # D E # # # # # G F @

画出对应的二叉树:

考查了数据结构中的二叉树

~~~~~

10. 用邻接矩阵表示有向图 (图略)

考查了数据结构中的图的表示

~~~~~

11 根据 Nocomachns 定理, 任何一个正整数 n 的立方一定可以表示成 n 个连续的奇数的和。例如:

13=1

23=3+5

33=7+9+11

43=13+15+17+19

在这里, 若将每一个式中的最小奇数称为 X, 那么当给出 n 之后, 请写出 X 与 n 之间的关系表达式: \_\_\_\_\_

其实是考代数

12 某班有 50 名学生, 每位学生发一张调查卡, 上写 a, b, c 三本书的书名, 将读过的书打 “\*”, 结果统计数字如下:

只读 a 者 8 人; 只读 b 者 4 人; 只读 c 者 3 人; 全部读过的有 2 人; 读过 a, b 两本书的有 4 人; 读过 a, c 两本书的有 2 人; 读过 b, c 两本书的有 3 人。

(1) 读过 a 的人数是 ( )。

(2) 一本书也没读过的人数是 ( )。

一个商场有 m 种颜色的小球, 每种小球足够多, 在这 m 种小球中挑选 n 个小球的选法有多少种?

如 m=2, n=3 时有 4 种选法分别是: 两种小球的个数分别为 03, 12, 21, 30. 问: 当 m=4, n=4 时

选法数=\_\_\_\_\_。 35

如果一棵 m 度树中有  $n_1$  个度为 1 的结点,  $n_2$  个度为 2 的结点, ..... 有

$n_m$  个度为  $m$  的结点, 则该树中叶结点的个数=\_\_\_\_\_ .  $n_2+2n_3+\cdots+(m-1)n_m+1$

```
program t1;
var n:integer;
function count(n:integer):integer;
begin
  if n=1 then count:=0 else
    if n mod 2=0 then count:=count(n div 2)+1 else
      count:=count(n*3+1)+1;
  end;
begin
  readln(n);
  writeln(count(n));
end.
```

输入:99 输出: 25

```
2. program t2;
var hi, lo:integer;
procedure pl(m, n:integer; var hi, lo:integer);
var I:integer;
begin
  I:=n; hi:=0; lo:=0;
  Repeat
    I:=I-1; lo:=lo+m;
    If lo>=10000 then
      begin
        Lo:=lo-10000;
        Hi:=hi+1;
      end;
  Until I=0;
  Write(hi:4, ' ', lo:4);
End;
Begin
  P1(200, 343, hi, lo);
End.
```

输出: 6.8600

```
3. program t3;
Var d1, d2, X, Min : real;
begin
  Min:=10000; X:=3;
  while X < 15 do
    begin
      d1:=sqrt(9+(X-3)*(X-3));
```

```

        d2:=sqrt(4+(15-X)*(15-X));
        if (d1+d2) < Min then Min:=d1+d2;
        X:=x+0.001;
    end;
    writeln(Min:10:2);
end.
输出: 13.00

```

```

4.program t4;
var i,k,n:integer;
    x,w:array[1..500] of integer;
begin
    readln(n);
    for i:=1 to n do
    begin
        x[i]:=0;w[i]:=1;
    end;
    for i:=2 to trunc(sqrt(n))+1 do
    if x[i]=0 then
        begin
            k:=i*i;
            while K<=n do
                begin
                    x[k]:=i;
                    k:=k+i;
                end;
            end;
        for i:=n downto 1 do
        if x[i]<>0 then
            begin
                w[x[i]]:=w[x[i]]+w[i];
                w[i div x[i]]:=w[i div x[i]]+w[i];
                w[i]:=0;
            end;
        writeln(w[2],w[3]:5,w[5]:5);
    end.
输入:20 输出: 18      8      4

```

降序组合. 给定两个自然数  $n, r (n > r)$ , 输出从数 1 到  $n$  中按降序顺序取  $r$  个自然数的所有组合. 例如,  $n=5, r=3$  时, 有如下组合:

```

5 4 3
5 4 2
5 4 1
5 3 2

```

```

5 3 1
5 2 1
4 3 2
4 3 1
4 2 1
3 2 1

```

程序如下:

```

program tk1;
var n,r,i,j:integer;
    a:array[1..20] of integer;
begin
    write('n,r=');
    repeat
        readln(n,r);
    until n>r;
    i:=1;a[1]:=n;writeln('result:');
    repeat
        if i<>r then
            if a[i]>r-i then
                begin
                    ____ (1) ____; i:=i+1;
                end
            else begin
                ____ (2) ____;
                a[i]:=a[i]-1 end
            else
                begin
                    for j:=1 to r do write(a[j]:3);
                    writeln;
                    if a[r]=1 then
                        begin
                            i:=i-1; a[i]:=a[i]-1;
                        end else ____ (3) ____
                    end;
                until a[1]=r-1;
            end.
            1. a[i+1]:=a[i]-1
            2. i:=i-1;
            3. a[i]:=a[i]-1 或 a[r]:=a[r]-1;

```

2. 现在政府计划在某个区域内的的城市间架设高速公路, 以使任意两个城市间能够直接或间接到达, 怎样修路, 费用最小。

输入文件: 第一行一个整数  $n$  ( $n \leq 100$ ) 表示城市数目。

第二行至第  $n+1$  行每行两个数  $x_i, y_i$  ( $0 \leq x_i, y_i \leq 100$ ) 表示第  $i$  个城市的坐标(单位:千米);

输出最小费用(每千米一个单位价格)。

程序如下:

```
program t6;
const maxn=100;
type tcity=record
    x,y:real
end;
var c:array[1..maxn] of tcity;
    d:array[1..maxn,1..maxn] of real;
    p:array[1..maxn] of integer;
    n,i,j,k:integer;
    a,min:real;
begin
    readln(n);
    for i:=1 to n do readln(c[i].x,c[i].y);
    for i:=1 to n do
        for j:=1 to n do
            d[i,j]:=sqrt(sqr(c[i].x-c[j].x)+sqr(c[i].y-c[j].y));
        p[1]:=0;
        for i:=2 to n do ____ (4) ____
            for i:=1 to n-1 do
                begin
                    min:=1e10;
                    for j:=1 to n do
                        if ____ (5) ____ then
                            begin
                                min:=d[p[j],j];
                                ____ (6) ____
                            end;
                    a:=a+d[p[k],k];
                    p[k]:=0;
                    for j:=1 to n do
                        if ____ (7) ____ then p[j]:=k;
                    end;
                    writeln(a:0:2);
                end.
                4. p[i]:=1;
                5. (p[j]>0) and (d[p[j],j]) < min)
                6. k:=j;
```