暑期集训

信奥普及组 CSP-J

扣哒世界花儿实验班教研团队

第一章 基础知识与编程环境

【大纲内容】

【1】计算机的基本构成(CPU、内存、I/0 设备等)

【1】Windows、Linux 等操作系统的基本概念及其常见操作

【1】计算机网络和 Internet 的基本概念

【1】计算机的历史和常见用途

【1】NOI 以及相关活动的历史

【1】NOI 以及相关活动的规则

【1】位、字节与字

【1】程序设计语言以及程序编译和运行的基本概念

【1】使用图形界面新建、复制、删除、移动文件或目录

【1】使用 Windows 系统下的集成开发环境(例如 Dev-C++等)

【1】使用 Linux 系统下的集成开发环境(例如 Code::Blocks 等)

【1】常用编译命令 g++的基本使用

一、计算机的基本构成(CPU、内存、I/0 设备等)

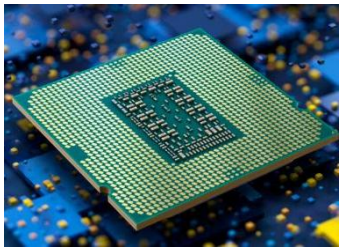
1. CPU——计算机的“超级大脑” (CPU, Central Processing Unit)

想象 CPU 是学校的数学天才！你给他题目（程序指令），他立刻心算解答（每秒处理数十亿次计算）。

重要功能:

- **做计算**：加减乘除、比较大小（if(a>b) 就是 CPU 在判断）
- **下命令**：**指挥**内存、硬盘等其他部件工作
- **跑程序**：你写的代码最终变成 CPU 能懂的指令

编程比赛考点 ：



① **算法效率**：CPU 计算越快，程序跑得越快！

- 笨方法：用循环算 1+2+...+100 → CPU 要算 100 次 ✗
- 聪明方法：用公式 100×101/2=5050 → CPU 算 1 次 ☑

② **避免死循环**：

```
while(1) { } // CPU 被无限占用 → 程序卡死！ （比赛 0 分）
```

2. 内存——程序的“临时工作室” （Memory / RAM）

像你桌上的草稿纸！写作业时放课本和铅笔（**存数据**），写完就收走（**关机后清空**）。

关键特点：

- **闪电速度**：比硬盘快 100 倍（CPU 直接从内存拿数据）
- **容量有限**：比赛电脑通常 4~16GB（1GB≈10 亿字节）
- **临时存储**：只存运行中的程序和数据

编程比赛必考陷阱 ：

① **数组开太大 → 内存爆炸！**

```
int a[10000000]; // 占 40MB → 超过内存限制 → MLE 错误（0 分！）
```

救命技巧：全局数组开在函数外面！

```
int safe[1000000]; // 全局数组 → 内存大空间 ☑
int main() {
  int danger[1000000]; // 局部数组 → 可能爆栈 ✗
}
```

② **递归太深 → 栈溢出！**

```
void f() { f(); } // 无限递归 → 栈区塞满 → "Stack Overflow"
```

3. I/O 设备——数据的“搬运工” （I/O Devices）

输入设备（送数据给计算机）

- **键盘**：比赛时手动输入测试数据（小心输错格式！）
- **文件**：省赛常用！从 data.in 读大数据（代码要写对路径）

```
freopen("data.in", "r", stdin); // 文件重定向 ★
```

输出设备（展示结果）

- **显示器**：答案必须严格按格式输出！（多空格/少换行都扣分）

```
cout << "答案：" << ans << endl; //    注意题目要求"Case 1: ans"
```

- **文件**：结果写入 result.out（忘关文件可能丢数据！）

```
freopen("result.out", "w", stdout);
```

存储设备（计算机的“仓库”）

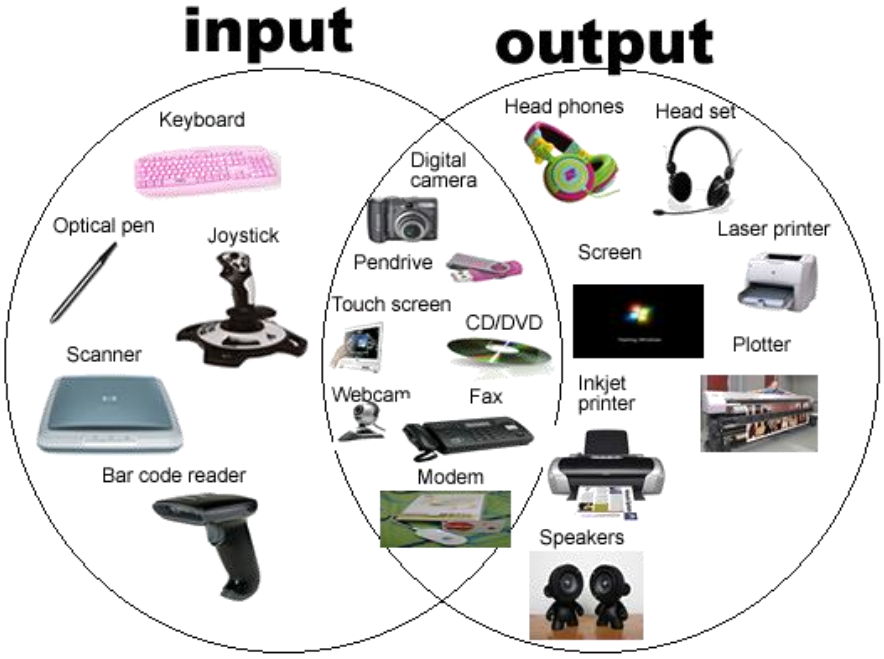
- **硬盘**：长期存数据（程序、题目）
- **机械硬盘**：容量大（1TB=1024GB），速度慢
- **固态硬盘**：速度快 5 倍！装系统的电脑开机秒启
- **U 盘**：拷贝常用文件（建议备份 2 份！）

数据旅行故事（比赛结束后你的程序做了哪些事）

- **出发**：评测机从硬盘加载 test.in 测试数据
- **中转**：数据搬进内存 → CPU 随时取用
- **计算**：CPU 狂算你的代码（此时内存像草稿纸堆满中间结果）
- **抵达**：答案从内存搬到 result.out → 评测机对比正确答案

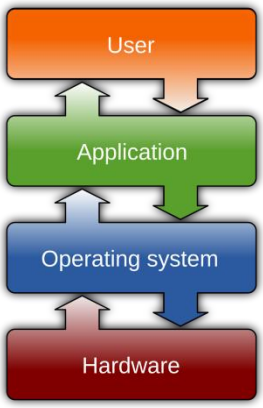
实战建议：

- 数组大小用全局变量！
- 文件读写练到闭眼能写
- 输出格式复制题目要求
- 死循环检查



二、Windows、Linux 等操作系统的基本概念及其常见操作

操作系统是什么？——计算机的“超级大管家”



你的电脑像一座游乐园，操作系统（OS）就是总指挥！它负责：

- **开机启动**：开机时启动所有设备（CPU、内存、键盘亮起来！）
- **分配任务**：给每个程序安排“玩耍时间”（CPU 时间片）
- **管理资源**：像发门票一样分配内存空间
- **维持秩序**：防止程序打架（比如游戏和浏览器抢内存）
- **考点警报** ：比赛时程序崩溃，可能是 OS 终止了超时/超内存的代码！

Windows 系统——图形世界的“游乐场向导”

特点：

- 鼠标点点点：图标、窗口、开始菜单（适合初学者）
- 常见操作（比赛前必会！）：
  - 创建代码文件：右键 → 新建 → 文本文档 → 改名 hello.
  - 打开命令行：Win + R 输入 cmd → 进入代码文件夹：

```
cd Desktop\mycode   # 进入"桌面/mycode"文件夹
```

- 运行程序：双击编译好的 hello.exe

比赛考点 ：

路径分隔符用 反斜杠\（Linux 用正斜杠/！）

```
freopen("data\\test.in", "r", stdin); // Windows 路径写法 ☑
```

Linux 系统——编程比赛的“终极擂台”

为什么比赛都用它?

- 轻便高效：省内存，更多资源给程序!
- 终端为王：键盘操作快如闪电（选手必须会命令！）

必学命令（终端操作）：

```
ls                  # 查看当前文件（像打开抽屉）
cd homework         # 进入"homework"文件夹（像走进房间）
g++ code. -o code   # 编译 C++ → 生成 code
./code              # 运行程序（输出在终端）
```

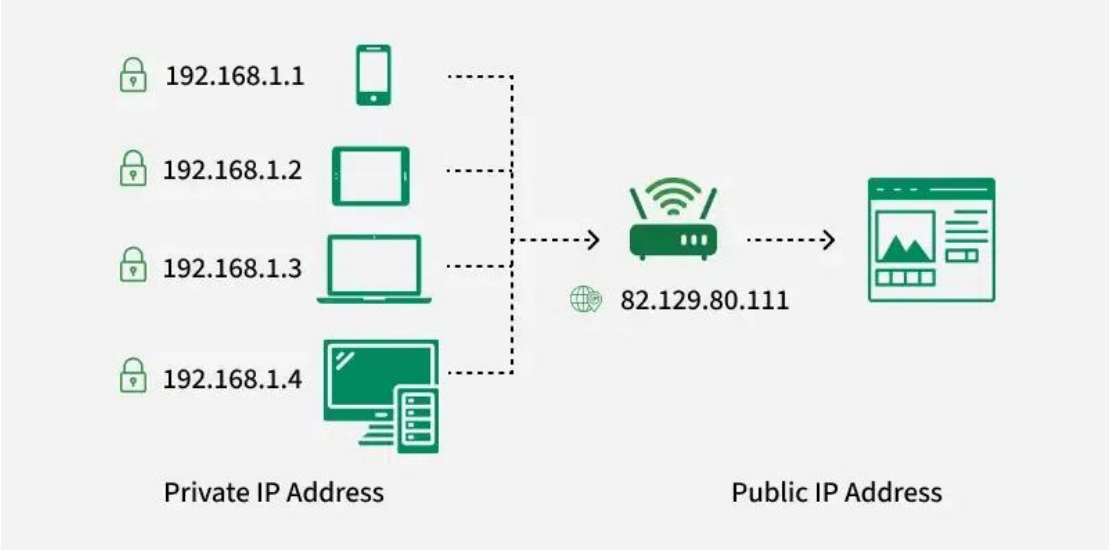
三、计算机网络和 Internet 的基本概念

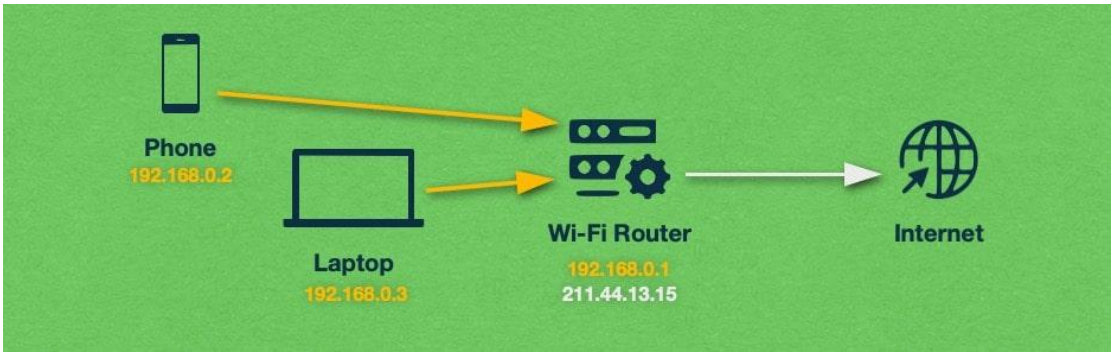
计算机网络是什么？——计算机的“快递系统”

你家电脑是超市，朋友电脑是蛋糕店。网络就是快递车队，把数据包（包裹）在计算机间运输!

核心成员:

- **IP 地址** → 计算机的“门牌号”（如 192.168.1.101）
  - 127.0.0.1 是本机地址（叫“localhost”，就是“我家”）
- **路由器** → 快递中转站（决定包裹走哪条路最快）
- **协议** → 运输规则（TCP 像“必须签收的快递”，UDP 像“丢门口就走”）

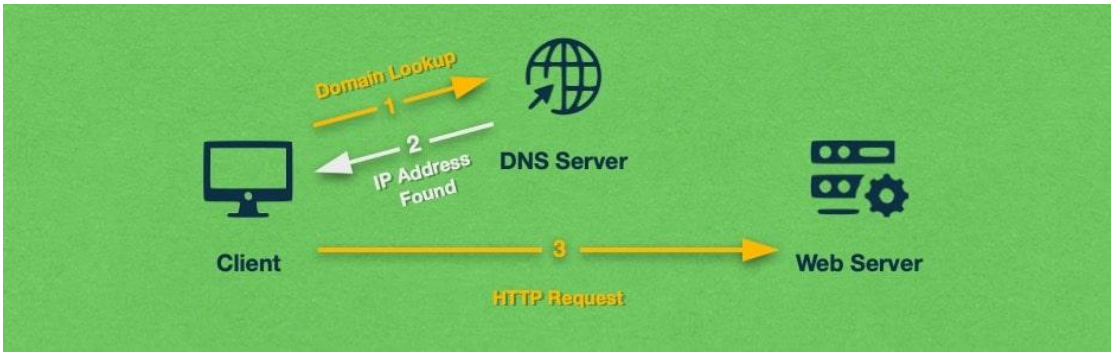




Internet——全球计算机的“超级联盟”

它怎么工作?

- 你在浏览器输入 www.koudashijie.com（域名）
- DNS 服务器（像电话簿）查出扣哒世界的 IP：220.181.38.251
- 路由器接力传输请求 → 百度服务器返回网页



四、计算机的历史和常见用途

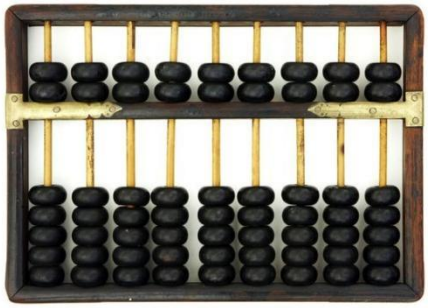
（一）计算机的“进化史”——从房间大小到口袋精灵

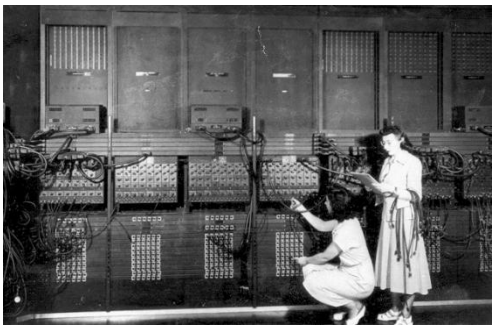
1. 史前时代（1946 年以前）

算盘 → 机械计算器：

古人用算盘珠子计算（像你的算术本），后来发明了带齿轮的机器，摇手柄做加减法。

考点：世界第一台通用计算机叫什么？（答案：**ENIAC**）





2. 初代巨无霸（1946 年）

ENIAC 诞生：

- 占地 170 平方米（比教室还大！）
- 每秒算 5000 次加法（比人类快，但不如现在手机）
- 耗电 15 万瓦（够照亮 300 个灯泡）

考点：ENIAC 用于计算什么？（答案：炮弹弹道轨迹）

3. 瘦身革命（1949-1970 年）

晶体管替代电子管：

电子管像灯泡易烧坏，晶体管像小开关更耐用！

集成电路出现：

把上千晶体管压进指甲盖大的芯片 → 计算机变小变便宜！

考点：集成电路的发明让计算机进入第__代？（答案：第三代）

4. 个人电脑时代（1980 年代）

苹果 Apple II：

第一台畅销家用电脑（键盘+显示器），小朋友用它学编程。

IBM PC：

企业办公室标配，“电脑”一词从此流行。

考点：比尔·盖茨为 IBM PC 开发了什么系统？（答案：MS-DOS）

5. 智能时代（现在）

手机=超级计算机：

你的口袋里有比登月飞船强 100 万倍的计算力！

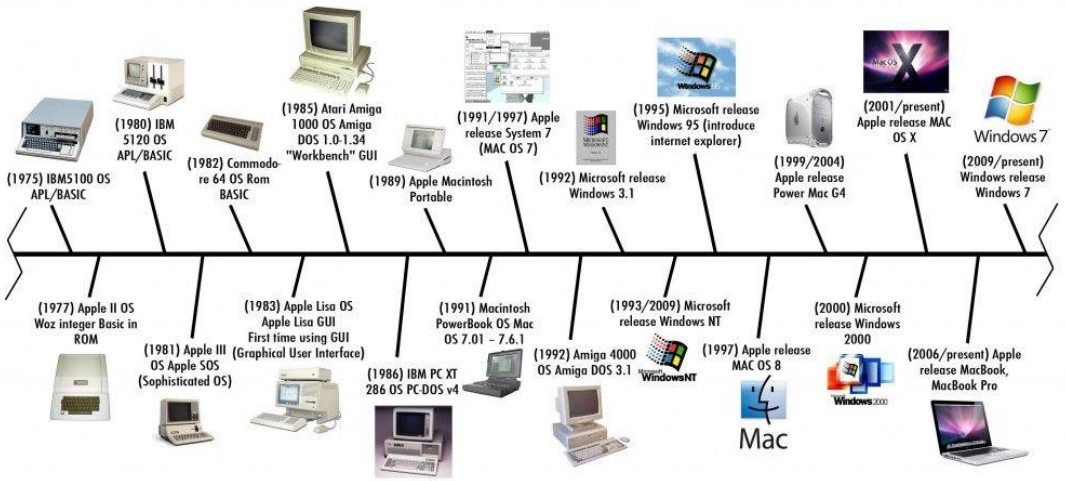
量子计算机：

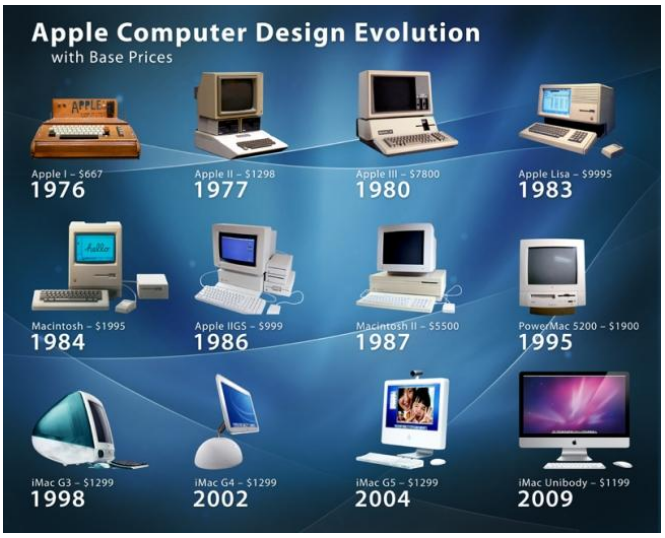
未来黑科技，1 秒破解现在 100 年才能解的密码

竞赛文化题高频考点：

冯·诺依曼提出“**存储程序**”原理（程序像菜谱存进计算机）

图灵奖 = 计算机界的诺贝尔奖





（二）计算机的用途——不只是打游戏！

1. 科学计算的“超人”

天气预报：

分析卫星云图数据 → 预测台风路径（用超级计算机算几小时）

航天工程：

控制火箭轨道（错 0.001%就会撞火星！）

竞赛直通：很多赛题原型来自科学计算！

例：计算圆周率π（用公式 π/4 = 1 - 1/3 + 1/5 - 1/7+...）

2. 数据处理的“闪电手”

学生成绩统计：

1 秒排序全校成绩单（冒泡排序 vs 快速排序）

疫情地图：

实时更新病例数 → 生成颜色预警地图

3. 自动控制的“隐形管家”

智能家居：

空调自动调温（温度传感器 → 计算机 → 控制开关）

无人驾驶：

用摄像头识别红绿灯 → 计算机决策刹车/加速

4. 人工智能的“最强大脑”

人脸识别：

从百万照片中找出你（用卷积神经网络）

围棋 AI：

AlphaGo 击败世界冠军（每秒算 2 亿步棋！）

五、位、字节与字

1. 位（Bit）——计算机的“最小细胞”

- 像一粒只有两种状态的“电子小豆”：**0（关） 或 1（开）**
- 计算机所有数据（文字/图片/程序）都由无数 0 和 1 组成

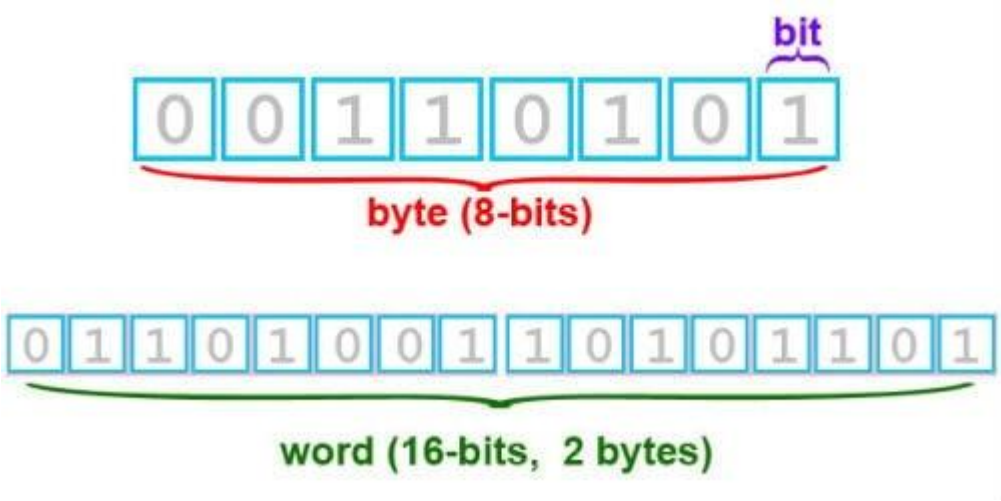
为什么重要?

- **硬件基础**：CPU 的晶体管通过开关表示 0/1
- **逻辑运算**：与(AND)、或(OR)、非(NOT) 就像开关组合电路

```
if (a > 0 && b < 10) // && 就是"与"（两个开关都开才通电）
```

位运算加速技巧（比加减乘除快 10 倍！）

- **乘法**：a * 8 → a << 3 // 左移 1 位=×2，左移 3 位=×8
- **奇偶判断**：if (n & 1) // 末位是 1→奇数，0→偶数



2. 字节（Byte）——内存的“标准小格子”

- **8 个位 = 1 个字节**（像 8 粒小豆装进 1 个盒子）
- 能表示 256 种状态（2^8=256，从 00000000 到 11111111）

为什么是 8 位?

- 历史选择!早期 IBM 系统用 8 位表示一个字符(如字母'A'的 ASCII 码是 01000001)

3. 字（Word）——CPU 的“专属运输箱”

- CPU 一次能处理的二进制位数（像卡车一次能拉多少货）
- 现代计算机通常是 **32 位（4 字节）** 或 **64 位（8 字节）**

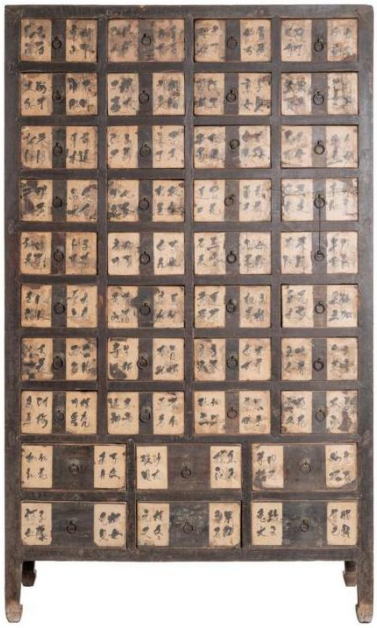
为什么影响编程?

- **64 位系统优势：**
  - 能直接用超大内存（32 位最多 4GB，64 位支持 16EB）
  - 更适合科学计算（如处理 long long 类型）
- **竞赛环境：**
  - 评测机通常是 64 位系统 → 可用 long long x = 1LL<<60;（2^{60}）

易错点：

- 移位越界 → 1 << 40 在 32 位系统会溢出！（因为 int 只有 32 位）
- 正确写法：1LL << 40（LL 表示 long long）

4、位、字节、字的关系



内存就像中药柜：

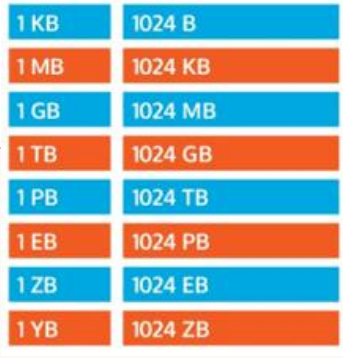
- **位** = 最小药格（放 0/1）
- **字节** = 一个抽屉（8 格 → 放 1 个 char）
- **字** = 整个药箱（多个抽屉 → CPU 一次取走 4 或 8 字节）

换算口诀：

- 1 字节(Byte) = 8 位(Bit)
- 1 千字节(KB) = 1024 字节　// 注意不是 1000！
- 1 兆字节(MB) = 1024KB

编程比赛实战指南

考点 1：用位运算极致优化




```
// 传统方法（慢 ✗）
if (n % 2 == 0) ...
// 位运算（快 ☑）
```

```
if ((n & 1) == 0) ...   // 判断偶数
```

考点 2：避免整数溢出

```
// 错误 ❌：1<<31 超过 int 范围（最大 2^31-1）
int max = 1 << 31;
// 正确 ☑：用 unsigned 或 long long
unsigned int max = 1U << 31;
long long big = 1LL << 60;
```

考点 3：精确计算内存

定义 double matrix[100][100]，占多少内存?

解：

1 元素 = 8 字节

总大小 = 100×100×8 = 80,000 字节 ≈ 78.125KB

五、其他

【大纲内容】

【2】ASCI 码

（一）什么是 ASCII 码？（What is ASCII Code?）

ASCII 码是计算机用来表示**字符（Character）和数字（Number）**的标准编码系统。

- 每个字符（字母、数字、符号）对应一个数字编号（0~127）。
- 计算机通过这些数字来识别和存储文本信息。

举例说明

就像字母和符号都有自己的“身份证号码”，电脑通过这些号码知道你输入的是哪个字符。

（二）ASCII 码的组成（Composition）

1. 控制字符（Control Characters）

编号 0~31，用来控制设备，比如换行（newline）。

2. 可打印字符（Printable Characters）

编号 32~126，是我们常用的字母、数字、标点符号。

3. 特殊字符

编号 127，表示“删除”（delete）功能。

ASCII 码部分示例：

为什么 ASCII 码重要？ (Why ASCII Code is Important?)

- 电脑只能理解**数字**，ASCII 让电脑知道数字对应什么字符。
- 是文字和数字之间的桥梁。

C++示例：打印字符和对应 ASCII 码

```
#include <iostream>
using namespace std;
int main() {
    char ch = 'A';
    int asciiCode = (int) ch;
    cout << "字符 '" << ch << "' 的 ASCII 码是: " << asciiCode <<
endl;
    // 从数字打印对应字符
```

```
        int num = 97;
        cout << "ASCII 码 " << num << " 对应的字符是：" << (char) num << endl;
        return 0;
    }
```

考试注意事项 & 易错点

易错点	正确说明
字符和 ASCII 码混淆	字符是字母或符号，ASCII 码是数字编码
类型转换写错	char 转 int 要强制类型转换（int)ch
控制字符忘记	换行和空格等特殊字符也有对应 ASCII 码
ASCII 范围不清楚	ASCII 码范围是 0~127，不是所有编码范围