CSP-J 中的数学知识

信息学竞赛里数学有多牛？等于考试自带“解题 Buff”！

想象一下信息学竞赛现场：
一堆烧脑题劈头盖脸砸过来？别慌！数学就是你最神的“翻译官”，把那些奇奇怪怪的问题（排队、分糖、找最优路...）瞬间变成你能懂的公式、图形、方程！没它？题目秒变摩斯密码，读题都得靠猜！

你以为猛敲键盘写代码就行了？Too simple！

- **算法是武器？数学就是武器使用说明书！** “最短路径”、“最大公约数”、“动态规划”...名字听着高大上吧？看不懂背后的数学道理？那只能瞎抡王八拳，累死电脑还没分！懂了数学？直接神装附体，招招暴击！
- **想代码又准又快？数学是你防止熬夜秃头的灵丹妙药！** 它能帮你秒看穿问题本质，躲开“无脑暴力循环”这个 CPU 杀手的大坑！省下时间够喝杯奶茶+检查三遍，不爽吗？
- **想挑战高端局？数学就是你的通关秘籍！** 数论、组合、图论...名字唬人吧？但在信息学竞赛后半场，它们就藏在能甩开对手的大题里！懂点门道，别人还在画圈圈，你就能摸到钥匙，分数嗖嗖涨！

真相只有一个：信息学竞赛表面考编程，骨子里狂考“数学内功”！
数学弱？开局就是地狱难度。
数学强？恭喜你，自带“外挂式理解力”！
想拿奖冲到前面？　狂练代码的同时，必须狠狠抱紧数学大腿——它才是你程序跑得飞起的隐藏涡轮增压！记住，没有数学加持的信息学竞赛之路，就跟没带指南针闯迷宫一样惨！

如何提升信息学竞赛里的数学水平，答案只有三个字：

看这里！！

目录

一、核心四类数集

自然数 (Natural Numbers) - ℕ

- **定义**：最常见的定义是从 **1** 开始的正整数 (1, 2, 3, ...)，有时也包括 **0**。信息学中通常包括 0（如数组索引从 0 开始）。
- 信息学要点：
  - **基础计数、循环控制**： 数组索引(for i from 0 to n-1)、计数器。
  - **非负性**：代表“数量”，不能为负。
  - **数据类型**： unsigned int, unsigned long long 适用于存储范围。
  - **常见运算**： + (加), * (乘), 乘方(pow 但需注意效率) 结果封闭(在表示范围内仍是自然数)， - (减) 可能为负（溢出或负数，编程中需判断）。

整数 (Integers) - ℤ

- **定义**：包括正整数、零、负整数 (..., -3, -2, -1, 0, 1, 2, 3, ...)。
- 信息学要点：
  - **有符号、范围关键**：使用 int, long, long long 存储， **时刻关注取值范围** ！整数溢出是常见错误源。
  - **离散性**：数轴上不连续的“点”。
  - **全部算术运算**： +, -, * 结果封闭。/ (除) 在整数运算中默认“整除”，结果为整数（商），**余数丢失**(用 % 取模可得余数)。
  - **重要概念**：**模运算(%)是信息学核心**！用于哈希、循环队列、同余方程、检查整除性等。

有理数 (Rational Numbers) - ℚ

- 定义：可以表示为两个整数之比 (p/q, 其中 p, q ∈ ℤ，q ≠ 0)的数。如 1/2, -3/4, 2 (即 2/1), 0.75 (即 3/4)。包含所有整数和有限小数、无限循环小数。

- 信息学要点：
  - 存储策略：通常不用内置小数类型！优先用分数形式：两个整数 (p, q) 表示一个有理数 p/q。
  - **约分：**始终保持 p 和 q 的最大公约数(gcd)为 1（分子分母互质）和分母非负，避免数值过大和不一致。
  - **精确性：分数存储能完美避免浮点数精度误差！**尤其需要比较、判断是否相等的场景。运算基于分数规则（通分、约分）。适用于需要高精度小数的简单场景（如概率计算中的分数形式）。
  - **局限性：**开根号结果未必是有理数（如√2）。

实数 (Real Numbers) - ℝ ——每年必考！

- **定义：**包含所有有理数**和**所有无理数（如π, e, √2, √3, 无限不循环小数）。
- 信息学要点：
  - **近似存储：**计算机无法精确表示所有实数（无理数和大部分有理数）。通常使用**浮点数格式(IEEE 754 标准)：**float (单精度), double (双精度)，本质是二进制科学计数法(± significand * 2^exponent)。
  - 浮点陷阱：
    - **精度误差：**尾数(significand)位数有限，二进制下很多看似简单的有限小数（如 0.1）无法精确表示，导致累积误差。.1 + .2 可能不等于.3！
    - **比较：严禁用 == 直接比较浮点数！**使用容差法： abs(a - b) < EPSILON (选择合适的 EPSILON，常为 1e-9 或更小)。
  - **运算结果：**+, -, *, / 原则上封闭（但存储有精度损失），乘方（开方）也常用。
  - **应用场景：**几何计算、物理模拟、数值分析。在精度要求极高的数值问题（如高精度解方程），可能需要高精度实数（库或字符串模拟）。

二、二进制与十进制的转化——每年必考！

转换本质：数的多项式表示

核心概念　：任何进制数本质是　**系数×基数的幂次和**

十进制（基数为 10）：

1310=**1**×**101**+**3**×**100**

（系数是 1 和 3，幂次由右向左递增）

二进制（基数为 2）：

11012　=**1**×**23**+**1**×**22**+**0**×**21**+**1**×**20**

☑ **关键规律**　：

从右向左，第 n 位的权重 = 2*n*−1（n 从 1 开始计数）

十进制→二进制（除基取余法）

数学原理：反复对 2 取余数，得到**从低到高**的系数

步骤拆解（以 13 为例）：

步骤	**计算**	**意义**	**当前位**
1	13÷2=6　**余 1**	最低位系数 = 1	*b*0
2	6÷2=3　**余 0**	次低位系数 = 0	*b*1
3	3÷2=1　**余 1**	中位系数 = 1	*b*2
4	1÷2=0　**余 1**	最高位系数 = 1	*b*3

☑ **结果：**按余数**倒序**排列：11012

（因为余数反映的是从低到高的系数）

二进制→十进制（位权展开法）

数学原理：直接计算多项式值

快速计算技巧（以 11012 为例）：

二进制位	权重计算	结果	数学原理
右 1 位	1×20=1	**1**	20=1
右 2 位	0×21=0	**0**	21=2
右 3 位	1×22=4	**4**	22=4
右 4 位	1×23=8	**8**	23=8

☑ **求和**：8 + 4 + 0 + 1 = 13

速算口诀：

从右向左，权重按 20,21,22...递增

三、初等数论

整除（数学的“整除规则”）

定义：整数 a 除以整数 b（b ≠ 0），如果结果正好是 **整数且没有余数**，就说 b 整除 a。

符号：b | a （读作 "b 整除 a"）

核心：除法结果是整数

例子：

12 ÷ 3 = 4（整数）→ 3 | 12

15 ÷ 4 = 3.75（非整数）→ 4 不能整除 15

☑ **一句话秒懂**：蛋糕必须分完！

把 12 块蛋糕分给 3 人，每人 4 块正好分完 → 3 整除 12。

因数与倍数（密不可分的“孪生兄弟”）

概念	定义	比喻	例子
因 数	整除一个数的整数	拆积木的小块零件	12 的因数：1,2,3,4,6,12
倍 数	能被一个数整除的数	积木拼出的大结构	3 的倍数：3,6,9,12...

关系 ：

如果 a | b，那么：

a 是 b 的因数 （如 3 是 12 的因数）

b 是 a 的倍数 （如 12 是 3 的倍数）

口诀：因数是零件，倍数是成品！

质数 vs 合数（数字世界的“原子”与“分子”）

类别	定义	特性	例子
质数	大于 1，且只有 1 和它本身两个因数	不可拆分（数学原子）	2,3,5,7,11,13...
合数	大于 1，有至少三个因数	可拆分（如 6=2×3）	4,6,8,9,10,12...

关键规则 ：

1 既不是质数也不是合数 （只有 1 个因数）

2 是最小的质数，也是唯一的偶质数

判断方法 ：试除法（用小于自己的质数挨个除）

比如判断 13 是不是质数？用 2,3,5,7（√13≈3.6，试到 3 即可）除都不整除 → 质数！

取整：数学的“严格舍入”

- **作用**：把小数变成最接近的整数（直接砍掉尾巴或进位）
- **两种方式**：
  - **向下取整（地板函数）**：⌊ x⌋
    规则：直接砍掉小数部分
    例子：
    ⌊ 3.8⌋ = 3（比 3.8 小的最大整数）
    ⌊ -2.3⌋ = -3（比-2.3 小的最大整数是-3）
    生活场景：奶茶店只能整杯卖，3.8 杯算 3 杯
  - **向上取整（天花板函数）**：⌈ x⌉
    规则：只要小数>0，整数部分就+1
    例子：
    ⌈ 4.1⌉ = 5（比 4.1 大的最小整数）
    ⌈ -1.7⌉ = -1（比-1.7 大的最小整数是-1）
    生活场景：打车软件计费，4.1 公里按 5 公里收费

☑ **口诀**：
地板函数砍小数，天花板见小就进一

取余：除法中的“剩菜打包”——每年必考

- **定义**：整数除法中 **除不尽剩下的部分**
- **符号**：a % b（读作 "a mod b"）
- **核心公式**：被除数 = 除数 × 商 + 余数
- **规则**：**余数必须 ≥0 且 < |b|**（重点！）

例子 1（正数）：

- 13 ÷ 5 = 2 余 3 → 13 % 5 = 3（因为 13=5×2 + **3**）
- 8 ÷ 3 = 2 余 2 → 8 % 3 = 2

例子 2（负数）：

- -13 ÷ 5 = -3 余 2 → -13 % 5 = 2（因为 -13=5×(-3) +**2**）

注意：C++中负数取余结果符号取决于编译器，但竞赛要求我们手动调整保证非负

生活比喻：
妈妈烤了 13 个饼干（被除数），每人分 5 个（除数）
分给 2 人（商）后，还剩 3 个饼干（余数）→ **余数就是分不完的“剩菜”**

模运算：循环世界的“数学钟表”——每年必考

- **本质**：**周期性取余**，结果在 0 到 b-1 之间循环
- **符号**：同取余 (a mod b)
- **核心思想**：把数映射到一个圆环上（想象钟表）

神奇的特性：

1. **周期循环性**：
a mod b 的结果一定是 0,1,2,...,b-1 中的一个
例子：
7 mod 3 = 1,
10 mod 3 = 1,
13 mod 3 = 1（每隔 3 个数循环一次）

2. **时间计算**：
现在是 11 点，过 35 小时后是几点?
(11 + 35) % 12 = 46 % 12 = 10（因为 46 ÷ 12 = 3 余 **10**）

3. **星期几计算**：
今天是周三，100 天后是周几?
100 % 7 = 2 → 周三 + 2 天 = **周五**

终极理解：
模运算像转盘：数值超过最大值就回零重来！

取模，取余运算在编程中的实战技巧（C++版）——每年必考！

操作	数学	C++代码示例	用途场景

	表示		

向下取整	$\lfloor x \rfloor$	int a = floor(3.8); // a=3
		#include <cmath>
			分页计算（每页 10 条，需多少页？）

向上取整	$\lceil x \rceil$	int b = ceil(2.1); // b=3
		#include <cmath>
			资源分配（每辆车载 5 人，需几辆车？）

取余/模运算	a % b	int c = 17 % 5; // c=2	判断奇偶性（if(n % 2 == 0)）

		int d = (-17 % 5 + 5) % 5; // d=3
		(竞赛处理负数的技巧)
			哈希表定位数据位置

		int hour = (current_hour + add) % 24;	24 小时制时间转换

重要说明 ：

C++中 floor()和 ceil()需要#include <cmath>

负数取余处理：先用(负数 % 正数 + 正数) % 正数转为非负

辗转相除法（求两数最大公约数 GCD）

输入： 两个正整数 a 和 b（设 a > b）
输出： a 和 b 的最大公约数

步骤：

用大数除以小数 ：

计算 a ÷ b，得到 **商**（忽略）和 **余数 r**（r = a % b）。

例：若 a = 48, b = 18 → 48 ÷ 18 = 2...12 → r = 12

替换更新 ：

将原来的 **除数 b 作为新被除数**（a 新 = b）；

将 **余数 r 作为新除数**（b 新 = r）。

例：a 新 = 18, b 新 = 12

循环直到余数为 0：

重复步骤 1–2，计算 **新的** a ÷ b，更新余数 r。

例：18 ÷ 12 = 1...6 → a 新 = 12, b 新 = 6
再重复：12 ÷ 6 = 2...0 → **余数 r = 0**

余数为 0 时终止：

当某次余数 **r = 0 **时，当前的** 除数 b 即为最大公约数**。

例：余数为 0 → **GCD = 6**

口诀速记：

大除小，取余数；除数变被除，余数变除数；余零即终止，除数即答案！

示例验证 （求 GCD(81, 57)）：

步骤	**操作**	**余数 r**	**更新后**：(a, b)
1	81 ÷ 57 = 1	**24**	(57, 24)
2	57 ÷ 24 = 2	**9**	(24, 9)
3	24 ÷ 9 = 2	**6**	(9, 6)
4	9 ÷ 6 = 1	**3**	(6, 3)
5	6 ÷ 3 = 2	**0**	→ **GCD = 3** ☑

埃拉托斯特尼筛法（埃氏筛法）——“全班点名排除法”

操作步骤（以 n=20 **为例）**：

1. **列出全班**：写下 2 到 20 所有数
[2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]

2. **第一个质数起立**：2 是质数 → 保留
   让 2 的倍数坐下（排除）：划掉 4,6,8,10,12,14,16,18,20

☑ **原理：**大于 2 的偶数全是合数！

3. **下一个站着的数**：3 是质数 → 保留
   让 3 的倍数坐下：划掉 6,9,12,15,18（已划的可跳过）

注意：6 和 12 已被划，跳过重复操作

4. **继续点名**：下一个站着的是 5 → 质数
   让 5 的倍数坐下 ：划掉 10,15,20（已划）
5. **跳过坐下的人** ：7 站着 → 质数
   让 7 的倍数坐下 ：14（已划）
6. **直到√n 停止** ：√20≈4.47 → 检查到 5 即可
   最终站着的 ：2,3,5,7,11,13,17,19（所有质数）

线性筛法（欧拉筛）——"精准追踪法"

目标：让每个合数只被标记一次 → 速度更快！

核心原理：

每个合数 = 最小质因数 × 最大因数
关键：用最小质因数筛掉合数

操作步骤（n=20）：

1. **准备名单**：空质数表 primes，标记数组 is_prime
2. **从 2 开始遍历：**
   - 2 未被标记 → 加入质数表 [2]
   - **用 2 筛**：2×2=4 → 标记 4（最小质因是 2）
3. **检查 3：**
   - 未被标记 → 加入质数表 [2,3]
   - **用 3 筛：**3×2=6, 3×3=9 → 标记 6 和 9
4. **检查 4：**

- **已被标记为合数** → 跳过加入质数表
- **用 4 筛？NO！** → 用最小质因数 2 筛：4×2=8 → 标记 8

✅ **关键点**：4 的最小质因是 2 → 只筛 4×2

5. **检查 5**：
    - 未被标记 → 加入质数表 [2,3,5]
    - **用 5 筛**：5×2=10, 5×3=15, 5×5=25>20 停止
6. **后续同理**：
    - 6：标记 6×2=12
    - 7：加入质数表并筛 7×2=14
    - 8：标记 8×2=16
    - 9：标记 9×2=18（**不筛** 9×3=27！）
    - 10：标记 10×2=20

四、离散与组合数学

什么是集合？—— 数学里的分组游戏

定义：把一些 **确定的对象** 放在一起形成的整体
关键特点：

- **无序性**：组内顺序不重要 → {苹果, 香蕉} = {香蕉, 苹果}
- **互异性**：成员不重复 → 不会有两个"苹果"
- **确定性**：要么在组内，要么不在（没有"可能"）

初中常见集合类型：

类型	例子	符号表示
数字集合	1~10 的整数	A = {1,2,3,...,10}
点坐标集合	直线上的点	B = {(x,y) | y=2x}

解集合	方程 $x^2=4$ 的解	C = {-2,2}

生活类比：
数学课代表把全班同学按兴趣分组：

- 篮球组 = {小明, 小刚, 小丽}
- 美术组 = {小丽, 小芳, 小红}

集合关系——小组之间的包含与分离

1. 子集（小组被大组包含）

- **定义**：若 A 组所有人都在 B 组里 → **A 是 B 的子集**
- **符号**：A ⊆ B（读作“A 包含于 B”）
- **例子**：

素数组 = {2,3,5,7}
10 以内自然数组 = {1,2,3,4,5,6,7,8,9,10}
→ 素数组 ⊆ 自然数组

2. 真子集（小组是大组的真部分）

- **定义**：A ⊆ B 且 A ≠ B（大组至少多 1 人）
- **符号**：A ⊂ B
- **例子**：

奇数组 = {1,3,5,7,9}
自然数组 = {1,2,3,4,5,6,7,8,9,10}
→ 奇数组 ⊂ 自然数组（因为缺了偶数）

3. 相等关系（两组成员完全相同）

- **定义**：A ⊆ B 且 B ⊆ A → A = B
- **例子**：

A = {大于 1 小于 4 的整数} = {2,3}
B = {方程 $x^2-5x+6=0$ 的解} = {2,3}
→ A = B

4. 交集（共同好友组）

- **定义**：同时属于 A 组和 B 组的人 → **A ∩ B**
- **例子：**
篮球组 ∩ 美术组 = {小丽}（小丽既打篮球又学美术）

5. 并集（合并两个组）

- **定义**：A 组或 B 组的成员（合并去重） → **A** ∪ **B**
- **例子：**
篮球组 ∪ 美术组 = {小明,小刚,小丽,小芳,小红}

6. 补集（全班剔除某小组）

- **定义：**在整体中剔除 A 组的成员 → **A^c**
- **例子：**
假设全班 = {小明,小刚,小丽,小芳,小红}
篮球组^c = {小芳,小红}（没参加篮球的人）

7. 空集（没人报名的小组）

- **定义：**没有元素的集合 → **∅ **
- **特点：**空集是任何集合的子集（∅ ⊆ A）

逻辑运算——小组关系的数学语言

在集合操作中隐藏的逻辑关系：

集合关系	**逻辑符号**	**含义**	**生活例子**
A ⊆ B	∀x (x∈A → x∈B)	如果 x 在 A 组，就一定在 B 组	篮球组成员一定属于运动社
**A ∩ B ≠ ∅ **	∃x (x∈A ∧ x∈B)	存在同时在 A 和 B 组的人	有人既是班长又是课代表
**A = ∅ **	¬∃x (x∈A)	没有任何元素在 A 组	没人报名的冷门社团

核心逻辑符号：

- **∀**：所有（For all）

- **∃**：存在（Exists）
- **¬**：否定（Not）
- **→**：如果…那么…（Implies）
- **∧**：并且（And）

加法原理与乘法原理

加法原理：独立选择，方案相加

核心思想

当一件事有多类不重叠方案，每类都能独立完成任务时，总方法数 = 各方案类方法数相加。
❗ 关键：分类之间互斥（选 A 就不能选 B）

生活场景：午餐点主食

食堂提供：

包子类：鲜肉包、豆沙包、奶黄包（3 种）

面条类：炸酱面、牛肉面（2 种）

米饭类：盖饭、炒饭（2 种）

规则：每人只能选 1 份主食（不同类不共存）
✅ 总选择数= 3（包子）+ 2（面条）+ 2（米饭）= 7 种

✨ 加法原理公式：
总方法数 = 方案 A 数量 + 方案 B 数量 + … + 方案 N 数量

乘法原理：分步选择，方案相乘

核心思想

当一件事要分步骤完成，且每步选择独立不影响时，总方法数 = 各步骤方法数相乘。
❗ 关键：步骤之间关联（先选 A 再选 B）

生活场景：搭配校服

学校要求穿：

  上衣：白色、蓝色（2 件）

  裤子：黑色、灰色（2 条）

鞋子：运动鞋、帆布鞋（2 双）

规则：需同时穿上衣、裤子、鞋子（分步选择不冲突）

✅ 总搭配数= 2（上衣）× 2（裤子）× 2（鞋子）= 8 种

✨ 乘法原理公式：

总方法数 = 步骤①选择数 × 步骤②选择数 × … × 步骤 N 选择数

加法 vs 乘法：对比辨析（必考重点！）

场景	核心区别	加法原理	乘法原理
午餐点 1 份主食	多类方案选其一	✅包子/面条/米饭选 1 类	❌无关
午餐点套餐	主食+饮料全都要	❌无关	✅主食 3 种×饮料 4 种=12 套
上学路径	多条独立路线选 1 条走	✅东/南/西路选 1 条	❌无关
穿校服	上衣+裤子+鞋全都要穿	❌无关	✅2×3×2=12 种穿搭

简易判断法：

问自己 → "是选一类？还是全都要？"

选一类 → 加法（分类计数）

全都要 → 乘法（分步搭配）

排列组合——每年必考！

一、组合（Combination）：「选人组队」不讲顺序

核心特征：选出一组人就行，谁先谁后无所谓

生活场景：

班内选 3 名代表参加校运会

小 A + 小 B + 小 C

小 B + 小 A + 小 C
→ 这两种算同 1 种组队！（因为成员相同）

组合公式：
$C(n,k)=k!(n-k)!n!$（n 人中选 k 人）

计算步骤：

分子：$n\times(n-1)\times\cdots\times(n-k+1)$（选 k 人的排列数）

分母：除以 $k!$（消除重复顺序）

例题：10 个同学中选 3 人组队 →
C(10,3)=3×2×110×9×8=120 种组合

✅ 口诀：组合就是选人，顺序不重要！

二、排列（Permutation）：「排队站位」顺序是关键

核心特征，排出队伍，谁站第几位很重要

生活场景：
给 3 名运动员（A/B/C）排冠亚季军领奖台

金牌 A + 银牌 B + 铜牌 C

金牌 B + 银牌 A + 铜牌 C
→ 这两种算不同结果！（顺序不同）

排列公式：
$P(n,k)=n\times(n-1)\times\cdots\times(n-k+1)$（n 人中排 k 个位）

计算步骤：

第 1 位：n 种选择

第 2 位：n-1 种选择

...直到第 k 位：n-k+1 种

例题：5 人排 3 个领奖台 →
P(5,3)=5×4×3=60 种排法

✅ 口诀：排队讲究顺序，换位就不同！

三、对比表：一秒分清排列 vs 组合

情景	问题本质	属于	计算答案
选 5 人成立班委会	只要成员	组合	C(30,5) = 142506
给 5 人安排班委职位	职位顺序不同	排列	P(30,5) = 17100720
抽 3 人领奖（不排序）	只选人，不排位	组合	C(40,3) = 9880
抽 3 人按 1/2/3 名领奖	抽人+排序	排列	P(40,3) = 59280

四、场景强化：避开四大经典陷阱

陷阱 1：排队时特殊位置怎么办？

例题：6 人排一排，甲必须站左端 →

解法：固定甲的 1 个位置 → 剩余 5 人排列：1 × 5! = 120 种

陷阱 2：相邻捆绑法

例题：4 对情侣排节目，必须成对站一起 →

解法：每对当「1 个模块」→ 4 模块排列：4! × (每对可互换 2 种) = 24 × 16 =　384 种

陷阱 3：不相容插空法

例题：3 个老师不能站一起，站在 5 个学生之间 →

解法：①学生排好：5! ②6 个空位选 3 个塞老师：P(6,3) →　14,400 种

陷阱 4：组合分组后除序

例题：10 人分 3 组（每组人不同），有几种分法？

解法：先选组 1：C(10,3)，再组 2：C(7,3)，组 3：C(4,4)
→ 需除以组间顺序：3!C(10,3)×C(7,3) = 2100 种

五、用乘法原理推导排列组合公式

组合公式推导：
C(*n*,*k*) = k 人全排列选 k 人排列数 = *k*!P(*n*,*k*)
原因：选人后再给 k 人排序 → 每个组合对应 k!种排列

排列公式推导：

$P(n,k) = n\times(n-1)\times\cdots\times(n-k+1)$

原因：第 1 位有 n 选择 → 第 2 位剩下 n-1 → 依此类推

五、ASCII 码——每年必考！

ASCII 是什么？—— 计算机的「文字密码本」

想象计算机的大脑只能处理数字（0 和 1），但我们输入的是字母、符号。ASCII 就是它们的**翻译字典**！

- **诞生：**1960 年代制定，统一了 128 种字符的数字编码
- **本质：**给每个键盘字符分配**专属数字 ID**（像学生学号）

例：A→65，a→97，0→48，!→33

用一张表理解 128 个位置（0-127）：

区域	**数字范围**	**包含内容**	**例子**
控制字符区	0~31	计算机控制指令（不可显示）	换行符（ID=10）
符号数字区	32~64	空格、标点、数字 0-9	空格=32，0=48，@=64
字母区	65~90（大写） 97~122（小写）	A-Z 和 a-z	A=65，a=97
扩展符号区	91~126	符号 [\ ] ^ _ { \ } ~	[=91，~=126

重点规律：

- 大写 A-Z = **65~90**（连续 26 个）
- 小写 a-z = **97~122**（连续 26 个）
- 数字 0-9 = **48~57**（0 的 ID 是 48，不是 0！）
- 大小写相差 **32**：小写 ID = 大写 ID + 32（如 A:65 → a:97）

六、初等数学

二次根式 —— 数学中的“根号魔术”

1. 定义
生活比喻：根号（√）就像“压缩包工具”，√4 表示解压后得到“平方等于 4 的数”（±2），但我们通常取正数解（算术平方根）。
数学语言：形如 √a（a≥0）的式子叫二次根式。
2. 核心性质（三大法则）

法则	公式	实例
乘方法则	$(\sqrt{a})^2 = a$	$(\sqrt{3})^2 = 3$
乘除拆合法则	$\sqrt{(a\times b)} = \sqrt{a} \times \sqrt{b}$	$\sqrt{12} = \sqrt{(4\times 3)} = 2\sqrt{3}$
分母有理化	$1/\sqrt{a} = \sqrt{a} / a$	$1/\sqrt{2} = \sqrt{2}/2$

一元二次方程 —— 寻找抛物线与 x 轴的交点

1. 标准形式

$ax^2 + bx + c = 0$（$a\neq 0$）

2. 三大解法（根据题型选工具）

方法	**适用情况**	**操作步骤**	**例题：解 $x^2-4x+3=0$**

因式分解法	能拆成两个括号相乘	$(x-p)(x-q)=0 \rightarrow x=p$ 或 q	$(x-1)(x-3)=0 \rightarrow$ **$x=1$ 或 3**
配方法	需要推导顶点或方程变形	$x^2+6x+5=0 \rightarrow (x+3)^2-9+5=0$	略
公式法	通用解法（必背）	$x = [-b\pm\sqrt{(b^2-4ac)}]/(2a)$	$x=[4\pm\sqrt{(16-12)}]/2 \rightarrow$ **$x=1$ 或 3**

3. 判别式Δ（探测根的性质）

$\Delta = b^2-4ac$

- $\Delta>0 \rightarrow$ **两个不等实根**（抛物线穿 x 轴两次）
- $\Delta=0 \rightarrow$ **两个相等实根**（抛物线顶点碰 x 轴）
- $\Delta<0 \rightarrow$ **无实根**（抛物线悬空不碰 x 轴）

二次函数 —— 抛物线的力量

1. 基本形式

$y = ax^2 + bx + c$

- **三大要素**　：
  - **开口方向：** $a>0$ 向上（微笑曲线）； $a<0$ 向下（哭泣曲线）
  - **顶点：** $(-b/(2a), (4ac-b^2)/(4a))$
  - **对称轴：** $x = -b/(2a)$

2. 图象性质（四类变化）

变换类型	**函数形式**	**图象变化**
上下平移	$y = x^2 + k$	$k>0$ 上移，$k<0$ 下移
左右平移	$y = (x-h)^2$	$h>0$ 右移，$h<0$ 左移
开口压缩	$y = a(x-h)^2+k$	

上下翻转	$y = -ax^2$	a 变负号即垂直翻转

3. 最值应用（实际生活）

- 抛物线顶点是最值点：
  - a>0 时顶点是**最低点** → **最小值**
  - a<0 时顶点是**最高点** → **最大值**

三角函数 —— 解三角形的神奇工具

1. 直角三角形中的定义（必背）

函数	含义	公式	形象比喻
sin	对边比斜边	sinA = 对边/斜边 = a/c	斜梯的陡峭度
cos	邻边比斜边	cosA = 邻边/斜边 = b/c	梯子底部稳固度
tan	对边比邻边	tanA = 对边/邻边 = a/b	坡面倾斜度

2. 三大核心关系

1. **互余关系** ：
$\sin(90°-A) = \cos A$, $\cos(90°-A) = \sin A$

例：$\sin 30° = \cos 60° = 0.5$

2. **同角关系** ：
$\sin^2 A + \cos^2 A = 1$（勾股定理派生）

验证：$\angle A=30° \to \sin^2 30° + \cos^2 30° = (0.5)^2 + (\sqrt{3}/2)^2 = 0.25 + 0.75 = 1$

3. **特殊角数值（必背）**

角度	**0°**	**30°**	**45°**	**60°**	**90°**
sin	0	1/2	√2/2	√3/2	1
cos	1	√3/2	√2/2	1/2	0

tan	0	$\sqrt{3}/3$	1	$\sqrt{3}$	无意义

勾股定理：直角三角形的「密码钥匙」

“勾三股四弦必五”——古代工匠的秘密法则

1. 定理本质

在 **直角三角形** 中（必须含 90°角！）：
直角边 a^2 + 直角边 b^2 = 斜边 c^2

2. 三大应用方向

用途	**实例**	**公式变形**
求斜边	梯子靠墙（知底高求长）	$c=\sqrt{a^2+b^2}$
求直角边	门框是否垂直（知三边验直角）	$a=\sqrt{c^2-b^2}$
证垂直	工地验直角（3m,4m,5m 必垂直）	若 $a^2+b^2=c^2$ → 直角

圆：宇宙最完美的几何图形

“一中同长”——墨子定义的核心特征
圆心（中心点 O）到圆周距离处处相等 → 这个距离叫半径 r

1. 三大核心部件

部件	**定义**	**生活类比**
半径 r	圆心到圆上任一点的距离	自行车轮辐条
直径 d	过圆心两端在圆上的线段	$d=2r$（直径=2 倍半径）
圆周率π	周长÷直径的固定比值 ≈3.14	任何圆都适用！

2. 圆的计算公式（永远关联π）

问题	公式	例题（r=3cm）
周长 C	C=2πr 或 C=πd	C=2×3.14×3≈18.84cm
面积 S	$S=\pi r^2$	$S=3.14\times3^2\approx28.26cm^2$
弧长 l	l=(圆心角 n°/360°)×2πr	90°弧长=¼周长

七、信息学中的实战演练

正确使用浮点数

```
...
double a = 0.1 + 0.2;
if (a == 0.3)
...
```

在以上代码中，a 等于 0.3 吗？？

正确判断质数

```
...
bool isPrime(int n) {
    for (int i=1; i*i<=n; i++)
        if (n%i==0) return false;
    return true;
    ...
```

以上代码能判断出质数吗？

正确认识 ASCII 码

...

char c = '5';

int num = c;

...

这里的 c 现在是 5 吗?

数学知识在信息学中如何应用，C++的实现与数学原理有哪些区别，尽在......