

三、初等数论 (Elementary Number Theory)

【大纲内容】

- 【3】整除、因数、倍数、指数、质(素)数、合数
- 【3】取整
- 【3】模运算与取余
- 【3】整数唯一分解定理
- 【3】辗转相除法(欧几里得算法)
- 【4】素数筛法：埃氏筛法与线性筛法

(七) 取整 (Integer Division)

取整是指两个整数相除后，**只保留整数部分**，舍去小数部分的操作。

例如 $7 \div 3 = 2.333\dots$ ，取整结果是 2。

就像你有 7 颗糖，要平均分给 3 个朋友，每人能分几颗糖？答案是 2 颗，剩下的糖不算。

举例：

- $10 \div 4 = 2.5$ ，取整结果是 2。
- $15 \div 5 = 3$ ，取整结果是 3。

(八) 模运算与取余 (Modulo Operation and Remainder)

模运算是**求两个整数相除后余下的部分（余数）**。

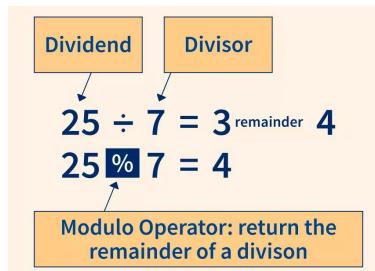
符号是 **%**，例如 $7 \% 3 = 1$ ，表示 7 除以 3 余 1。

你有 7 颗糖，分给 3 个朋友，每人分 2 颗，剩下 1 颗没分，就是余数。

举例：

- $14 \% 4 = 2$
- $20 \% 6 = 2$

C++示例代码 (取整和取余)



```

#include <iostream>

using namespace std;

int main() {

    int a = 7, b = 3;

    int quotient = a / b; // 取整

    int remainder = a % b; // 取余

    cout << a << " 除以 " << b << " 的商是: " << quotient << endl;

    cout << a << " 除以 " << b << " 的余数是: " << remainder <<
    endl;

    return 0;
}

```

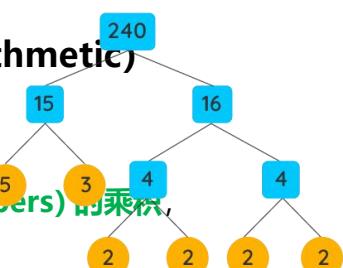
考试注意事项 & 易错点

易错点	正确说明
误以为除法结果自动取整	只有整数除法才会自动取整，浮点数除法不会
余数不能大于除数	余数总是小于除数
忘记余数的符号	余数和被除数符号有关（一般非负）
除数为零未考虑	除数不能为零，考试时需注意

(九) 整数唯一分解定理 (Fundamental Theorem of Arithmetic)

整数唯一分解定理指出：

任何大于 1 的整数，都可以且只能分解成若干个质数 (Prime Numbers) 的乘积，而且这个分解是唯一的 (顺序不同不算不同)。



举例说明

- 每个数字都是由一些特别的“积木块”搭成的，这些积木块叫做**质数**。
- 不管你怎么拆，最后的积木块种类和数量都是唯一的。

举例：

- 12 可以分解成 $2 \times 2 \times 3$ (质数)
- 30 可以分解成 $2 \times 3 \times 5$
- 不可能用其他质数组合分出不同结果。

简单 C++示例：质因数分解

```
#include <iostream>

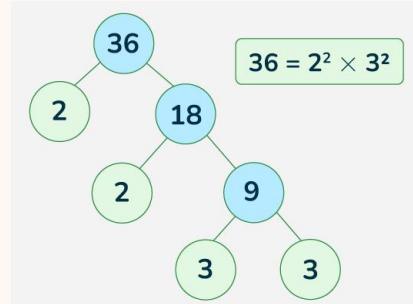
using namespace std;

void primeFactorization(int n) {
    cout << n << " 的质因数分解是: ";

    for (int i = 2; i * i <= n; i++) {
        while (n % i == 0) {
            cout << i << " ";
            n /= i;
        }
    }

    if (n > 1) cout << n; // 剩下的数一定是质数
    cout << endl;
}

int main() {
    int num = 84;
    primeFactorization(num);
    return 0;
}
```



输出：84 的质因数分解是：2 2 3 7

考试注意事项 & 易错点

易错点

正确说明

忽略质数定义

质数是只能被 1 和自身整除的数

分解不完全

要分解到所有因子都是质数

错误顺序认为是不同分解

顺序不同不算不同

质数判断时未检测到所有因子

检查到 $n\sqrt{n}$ 即可，不用更大