

2025-3-25 日 递归第二课

最大公约数：

如果数 a 能被数 b 整除， a 就叫做 b 的倍数， b 就叫做 a 的约数。约数和倍数都表示一个整数与另一个整数的关系，不能单独存在。

最大公约数是指两个或多个整数共有约数中最大的一个。例如：24，18 共同的约数为 1,2,3,6，其中 6 是最大的。所以 $\text{Gcd}(18,24)=6$ 。

我们在小学学过，求最大公约数可以使用辗转相除法

辗转相除法：

又名欧几里德算法。(Euclidean algorithm)，是求最大公约数的一种方法。它的具体做法是：用较大数除以较小数，再用出现的余数（第一余数）去除除数，再用出现的余数（第二余数）去除第一余数，如此反复，直到最后余数是 0 为止。如果是求两个数的最大公约数，那么最后的除数就是这两个数的最大公约数。

最小公倍数：

两个或多个整数公有的倍数叫做它们的公倍数，其中除 0 以外最小的一个公倍数就叫做这几个整数的最小公倍数。

最大公因数和最小公倍数之间的性质：两个自然数的乘积等于这两个自然数的最大公约数和最小公倍数的乘积。

最小公倍数特点：倍数的只有最小的没有最大，因为两个数的倍数可以无穷大。

$$\text{LCM}(a,b)=a \times b / \text{GCD}(a,b)$$

练习一：最大公约数 GCD

输入 2 个正整数 A ， B ，求 A 与 B 的最大公约数。

输入格式

2 个数 A,B ，中间用空格隔开。(1 ≤ A,B ≤ 10^9)

输出格式

输出 A 与 B 的最大公约数。

参考程序：

```
#include <bits/stdc++.h>
using namespace std;
int gcd(int a, int b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}
int main()
{
    int a, b;
    cin >> a >> b;
    cout << gcd(a, b) << endl;
    return 0;
}
```

练习二：最小公倍数 LCM

输入 2 个正整数 A, B, 求 A 与 B 的最小公倍数

输入格式

2 个数 A,B, 中间用空格隔开。 ($1 \leq A, B \leq 10^9$)

输出格式

输出 A 与 B 的最小公倍数。

参考程序：

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

int gcd(ll a, ll b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}

int main()
{
    ll a, b;
    cin >> a >> b;
    cout << a / gcd(a, b) * b << endl;
    return 0;
}
```

练习三：分数约分

小 Z 是个小学生，最近刚刚学习到了分数的知识，他苦恼于将分数化简，现在需要写一个程序来帮助他

输入格式

两个整数 a b (分别表示分子，分母。 $0 \leq a < 10000$ $0 < b < 10000$)

输出格式

两个整数 x y (分别表示值为 a/b 的最简分数的分子与分母)

输入样例

3

输出样例

001

010

011
100
101
110

参考程序：

```
#include <bits/stdc++.h>
using namespace std;
int gcd(int a, int b) {
    return b == 0 ? a : gcd(b, a % b);
}
int main()
{
    int a, b;
    cin >> a >> b;
    int d = gcd(a, b);
    cout << a / d << " " << b / d << endl;
    return 0;
}
```

练习四：选数字

从 $1 \rightarrow n$ 中选出 m 个数，要求同样的数字不能重复选择，按照字典序正序输出所有方案。例如：从 1 到 4 中选出 2 个数，共有 6 种方法，按照字典序输出，依次为：

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

输入格式

输入共 2 个数 n, m ，中间用空格分隔。（ $1 \leq m \leq n \leq 20$ ）

输出格式

按照字典序正序输出所有方案。

参考程序 1：

```
#include <bits/stdc++.h>
using namespace std;
int nums[21], n, m;
```

```

void F(int min, int sel) {
    if (sel == 0) {
        for (int i = 0; i < m; i++)
            cout << nums[i] << " ";

        cout << endl;
        return;
    }

    for (int i = min; i <= n; i++) {
        nums[m - sel] = i;
        F(i + 1, sel - 1);
    }
}

int main() {
    cin >> n >> m;
    F(1, m);
    return 0;
}

```

参考程序 2:

```

#include <bits/stdc++.h>
using namespace std;
int nums[21], m, n;

void F(int cur, int sel) {
    if(sel == 0) {
        for (int i = 0; i < m; i++)
            cout << nums[i] << " ";
        cout << endl;
        return;
    }
    if(cur > n) return;
    nums[m - sel] = cur;
    F(cur + 1, sel - 1);
    F(cur + 1, sel);
}

int main() {
    cin >> n >> m;
    F(1, m);
    return 0;
}

```

```
}
```

练习五：N!

用递推法实现,输入 n, 输出 $n! = n*(n-1)*(n-2)*(n-3)*\cdots*3*2*1$

输入格式

输入共 1 个整数 n。 ($1 \leq n \leq 20$)

输出格式

输出一个整数 表示 n 的阶乘的值。

参考程序：

```
#include<bits/stdc++.h>
using namespace std;
int cal(int n){
    if(n==1) return 1;
    else return n*cal(n-1);
}
int main(){
    int n;
    cin>>n;
    cout<<cal(n);
    return 0;
}
```