

线性代数

线性基

```
std::vector<i64> get_linear_basis(std::vector<i64>& nums, int N = 63) {
    std::vector<i64> p(N + 1);
    auto insert = [&](i64 x) {
        for (int s = N; s >= 0; --s) if (x >> s & 1) {
            if (!p[s]) {
                p[s] = x;
                break;
            }
            x ^= p[s];
        }
    };
    for (auto& x : nums) insert(x);
    return p;
}
```

```
signed main() {
    std::ios::sync_with_stdio(false);
    std::cin.tie(0), std::cout.tie(0);
    int n; std::cin >> n;
    std::vector<i64> nums(n);
    for (auto& x : nums) std::cin >> x;
    auto p = get_linear_basis(nums, 63);
    i64 ans = 0;
    for (int s = N; s >= 0; --s)
        ans = std::max(ans, ans ^ p[s]);
    std::cout << ans;
    return 0;
}
```

三角形面积

$$S = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

```
int main(){
    float num[6];
    for(int i = 0; i < 6; i++)
        cin >> num[i];
    float sum = 0.0;
    sum = 0.5*(num[0]*num[3]+num[2]*num[5]+num[4]*num[1]-num[0]*num[5]-num[2]*num[1]-num[4]*num[3]);
    cout << "三角形的面积为: ";
    sum == 0 ? cout << "Impossible" : cout << sum;
    return 0;
}
```

$$S = \frac{1}{4} \sqrt{(a+b+c)(a+b-c)(a+c-b)(b+c-a)}$$

```
p=(a+b+c)/2;
sum=sqrt(p*(p-a)*(p-b)*(p-c));
```