

### Задача 1.

Заметим, что если степень четная, то в таком случае можно переписать возведение следующим образом:  $a^{2n} = (a^n)^2$ . Таким образом, можно посчитать сначала  $a^n$ , а после этого возвести полученное число в квадрат. Таким образом, мы уменьшаем степень каждую итерацию в 2 раза, а это значит, что нам потребуется всего лишь  $\log_2(n)$  перемножений (возведений в квадрат).

Если же степень нечетная, то логика такая же:  $a^{2n+1} = (a^n)^2 * a$ .

```
long long fast_pow(int a, int n) {
    if(n == 1) return a;
    if(n % 2 == 0) {
        auto s = fast_pow(a, n / 2);
        return s * s;
    } else {
        auto s = fast_pow(a, n-1);
        return s * a;
    }
}
```

### Задача 2.

Прием, который потребуется в данной задаче называется схемой Горнера. Подробно о нем можно почитать например на русской Википедии: [Схема Горнера](#).

Основная идея заключается в том, что мы выносим за скобки иксы, где это возможно и начинаем подсчет с конца, каждый раз домножая на x и добавляя следующий коэффициент многочлена.

```
double gorner(std::vector<double> &p, double x) {
    int b = p.back();
    for(auto it = p.rbegin() + 1; it != p.rend(); it++) {
        b = *it + b * x;
    }
    return b;
}
```