

1) Schneeflocke

```
#include <iostream>
#include <ifmwindow>
#include <math.h>

void corner(Point p, Point q, IfmWindow& w)
{
    float f = sqrtf(3.0f)/4;

    int dx = q.x() - p.x();
    int dy = q.y() - p.y();

    if(dx*dx + dy*dy < 16)
    {
        w << Line(p.x(), p.y(), q.x(), q.y());
    } else {
        Point r = Point((p.x()*2 + q.x())/3, (p.y()*2 + q.y())/3);
        Point s = Point((p.x() + q.x())/2 - dy*f, (p.y() + q.y())/2 + dx*f);
        Point t = Point((p.x() + q.x()*2)/3, (p.y() + q.y()*2)/3);
        corner(p, r, w);
        corner(r, s, w);
        corner(s, t, w);
        corner(t, q, w);
    }
}

int main()
{
    IfmWindow w(200, 200, "Snow");

    corner(Point(30, 60), Point(100, 190), w);
    corner(Point(100, 190), Point(170, 60), w);
    corner(Point(170, 60), Point(30, 60), w);

    w.flush();
    w.wait_for_mouse_click();

    return 0;
}
```

2) Binomialkoeffizienten

```
#include <iostream>
#include <ifmwindow>

const int size = 60;

// slow version
long long binomial(int n, int k)
{
    if(k == 0 || k == n)
        return 1;
    else
        return binomial(n-1, k) + binomial(n-1, k-1);
}

// faster version
long long binomial(int n, int k, long long a[size][size])
{
    if(k == 0 || k == n)
        return 1;
    else
        return a[n-1][k] + a[n-1][k-1];
}

int main()
{
    long long a[size][size];

    IfmWindow w(size, size, "Binomialkoeffizienten");

    for(int n = 0; n < size; ++n)
        for(int k = 0; k <= n; ++k)
            a[n][k] = binomial(n, k, a);

    int s;
    std::cout << "Ganze Zahl s = ";
    std::cin >> s;

    for(int i = 0; i < size; ++i)
        for(int j = 0; j <= n; ++j)
            if(a[i][j] / s * s == a[i][j])
                w << Point(i, j);

    w.flush();
    w.wait_for_mouse_click();

    return 0;
}
```