



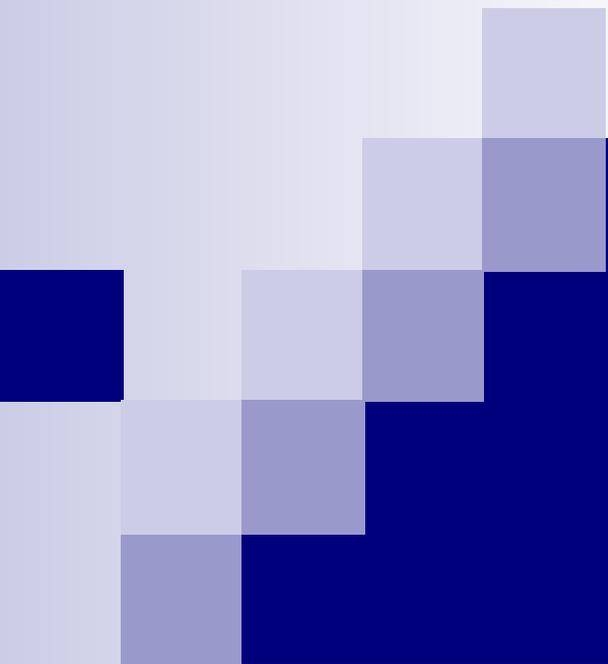
Примеры разработки программ

Алтайский государственный университет
Факультет математики и ИТ
Кафедра информатики
Барнаул 2014

Лекция 7

■ План

- Пара заданий для самопроверки
- Задача о разложении функции
- Задача о площади пересечения прямоугольников



Пара заданий для самопроверки

Задание «Циклы»

- Что выведет программа?

```
#include <stdio.h>

void main() {
    int i=0;
    for(;i<=2;)
        printf(" %d", ++i);
}
```

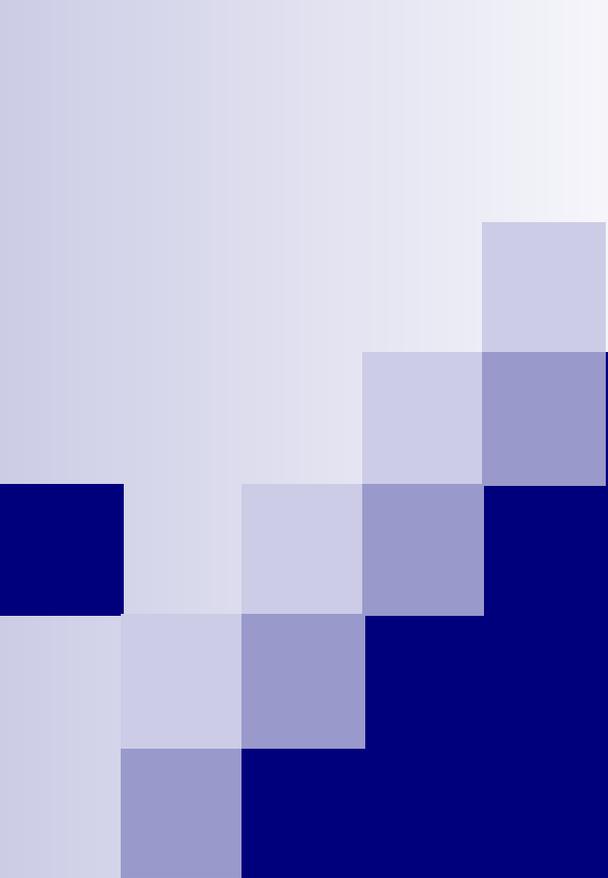
```
1 2 3
```

Задание «Циклы»

- Что выведет программа?

```
#include <stdio.h>

void main() {
    int x;
    for (x=1; x<=5; x++) ;
    printf ("%d", x) ;
}
```



Задача о разложении функции

- Постановка задачи
- Алгоритм
- Программа 1
- Замечание об эффективности
- Программа 2

Разложение функции: постановка задачи

- Необходимо численно убедиться в справедливости равенства, то есть сравнить результаты вычисления значения функции для заданного значения аргумента, полученные двумя способами:
 - вычислением с использованием стандартных математических функций;
 - путем разложения функции с заданной точностью ε .

При вычислении разложения необходимо выяснить количество шагов разложения, потребовавшееся для достижения точности ε .

$$\frac{1}{\sqrt[3]{1+x^3}} = 1 - \left[\frac{1}{3} x^3 - \frac{1 \cdot 4}{3^2 \cdot 2!} x^6 + \dots + (-1)^{n+1} \frac{1 \cdot 4 \cdot \dots \cdot (3n-2)}{3^n \cdot n!} x^{3n} + \dots \right].$$

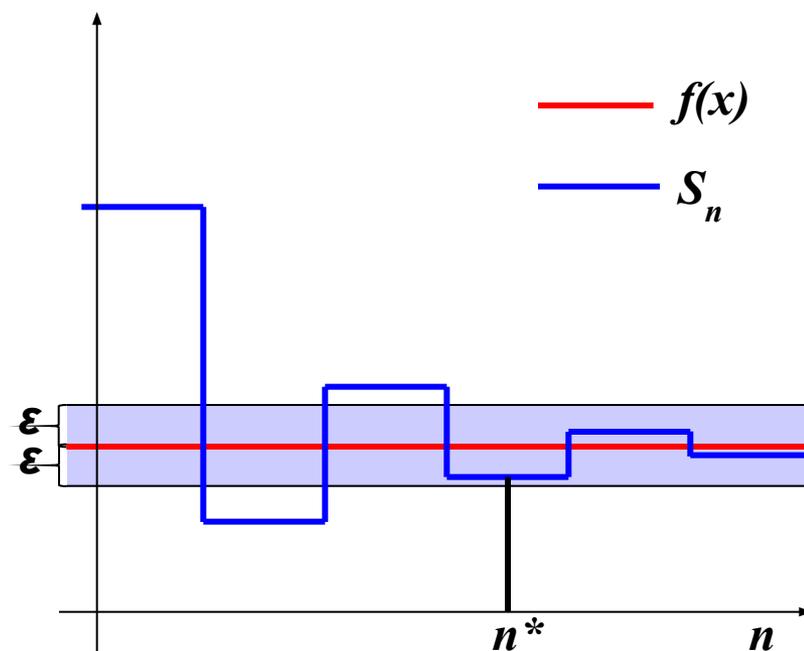
Разложение функции: суть

$$f(x) = \frac{1}{\sqrt[3]{1+x^3}}$$

$$S_n = (-1)^{n+1} \frac{1 \cdot 4 \cdot \dots \cdot (3n-2)}{3^n \cdot n!} x^{3n}$$

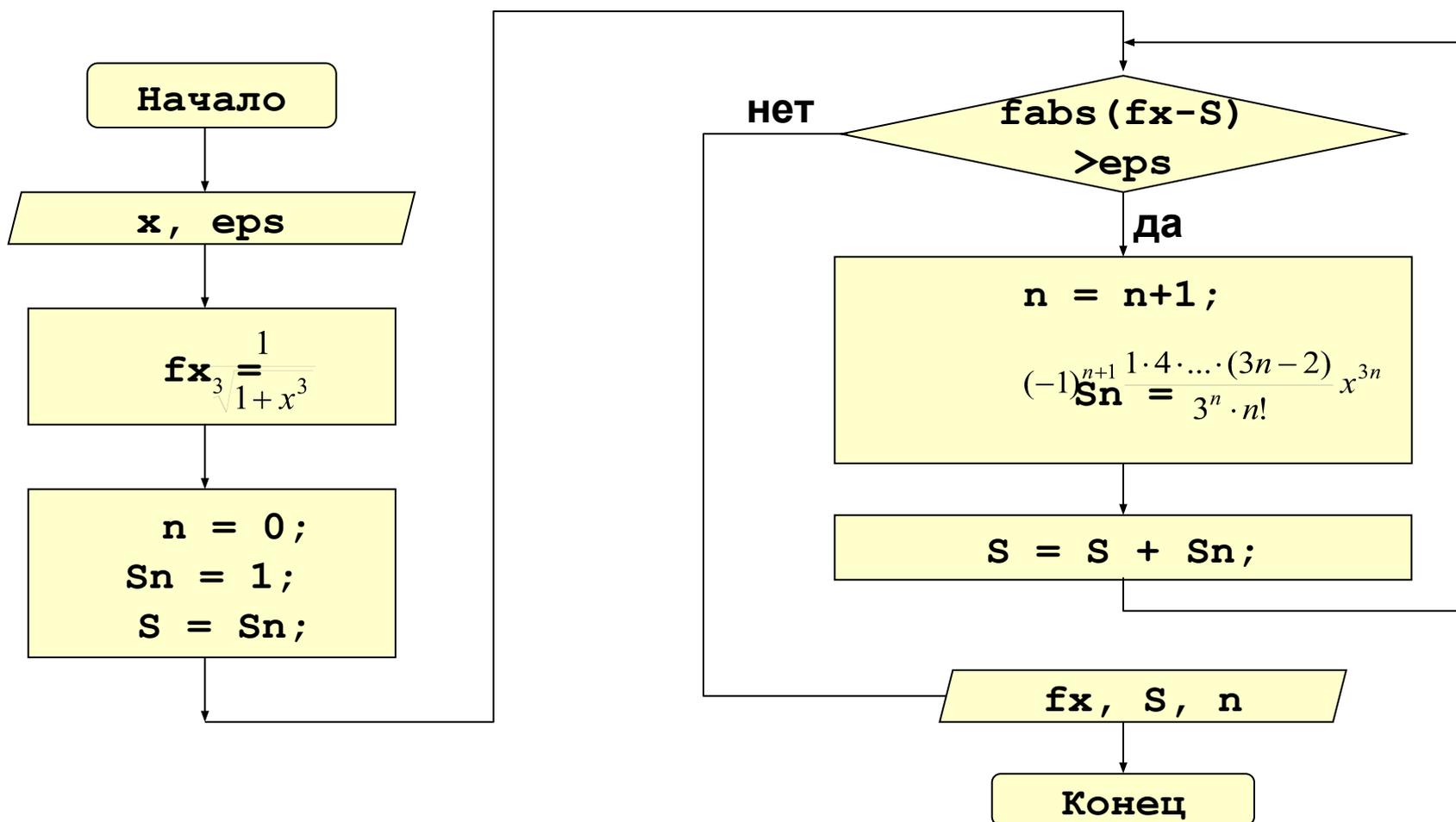
$$S = \sum_{n=0}^N S_n$$

n	$f(x)$	S_n	S
1	0.9614997135	1.0000000000	1.0000000000
2	0.9614997135	-0.0416666667	0.9583333333
3	0.9614997135	0.0034722222	0.9618055556
4	0.9614997135	-0.0003375772	0.9614679784
5	0.9614997135	0.0000351643	0.9615031427
6	0.9614997135	-0.0000038095	0.9614993332
7	0.9614997135	0.0000004233	0.9614997565
8	0.9614997135	-0.0000000479	0.9614997086
...



Разложение функции: алгоритм

$$\frac{1}{\sqrt[3]{1+x^3}} = 1 - \left[\frac{1}{3}x^3 - \frac{1 \cdot 4}{3^2 \cdot 2!}x^6 + \dots + (-1)^{n+1} \frac{1 \cdot 4 \cdot \dots \cdot (3n-2)}{3^n \cdot n!}x^{3n} + \dots \right].$$



Разложение функции: программа 1

ДЕМО



Разложение функции: программа 1

$$f(x) = \frac{1}{\sqrt[3]{1+x^3}}$$

$$S_n = (-1)^{n+1} \frac{1 \cdot 4 \cdot \dots \cdot (3n-2)}{3^n \cdot n!} x^{3n}$$

$$S = \sum_{n=0}^N S_n$$

```
#include <stdio.h>
#include <math.h>

void main() {
    double x, fx, Sn, S, eps, z, b;
    int n, i, a, p, f;

    printf("  x=");
    scanf("%lf", &x);
    printf("eps=");
    scanf("%lf", &eps);

    fx=1/pow(1+x*x*x, 1./3.);
    n=0;  Sn=1;  S=Sn;

    // Сюда вставить основной цикл

    printf("  fx=%12.10lf\n", fx);
    printf("  S=%12.10lf\n", S);
    printf("  n=%d\n", n);
}
```

```
// Основной цикл
while(fabs(fx-S) > eps) {
    n++; // Следующее слагаемое
    // Знак
    z=pow(-1.0, n);
    // Числитель
    a=1;
    for(i=1; i<=n; i++)
        a*=(3*i-2);
    // Факториал
    f=1;
    for(p=1; p<=n; p++)
        f*=p;
    // Знаменатель
    b=pow(3.0, n)*f;

    Sn = (z*a/b) * pow(x, 3.0*n);
    S += Sn;
```

Очевидное, но плохое решение

Разложение функции: суть

- Нетрудно заметить:

$$S_n = (-1)^{n+1} \frac{1 \cdot 4 \cdot \dots \cdot (3n - 2)}{3^n \cdot n!} x^{3n}$$

$$S_{n+1} = (-1)^{n+2} \frac{1 \cdot 4 \cdot \dots \cdot (3n - 2) \cdot (3(n + 1) - 2)}{3^{n+1} \cdot (n + 1)!} x^{3(n+1)}$$

$$S_{n+1} = -\frac{3n + 1}{3n + 3} x^3 \cdot S_n$$

Разложение функции: программа 2

ДЕМО



Разложение функции: программа 2

$$f(x) = \frac{1}{\sqrt[3]{1+x^3}}$$

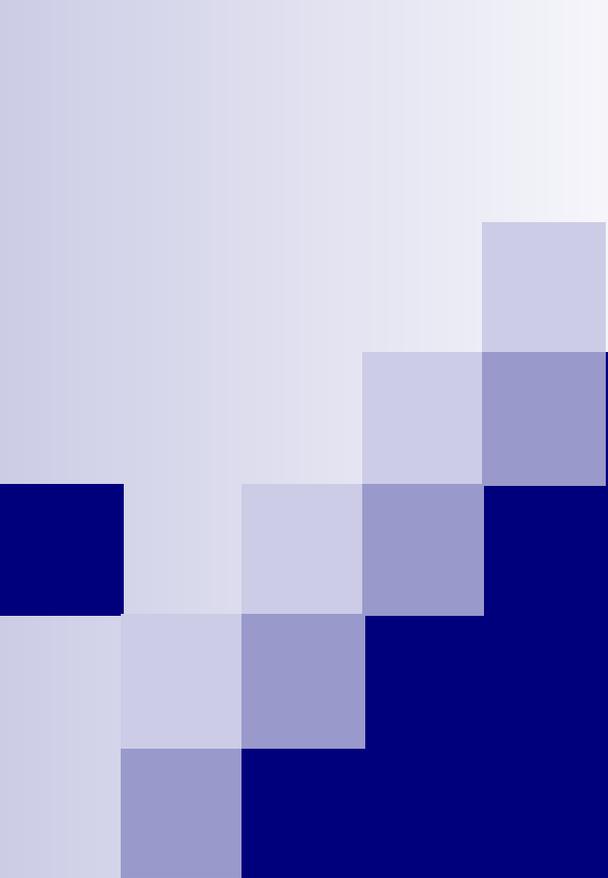
$$S_n = (-1)^{n+1} \frac{1 \cdot 4 \cdot \dots \cdot (3n-2)}{3^n \cdot n!} x^{3n}$$

$$S = \sum_{n=0}^N S_n$$

```
#include <stdio.h>
#include <math.h>
#define MAX_ITER 1000.0
void main() {
    double x,fx,Sn,S,eps,n;
    printf("x=");    scanf("%lf",&x);
    printf("eps=");  scanf("%lf",&eps);
    fx=1/pow(1+x*x*x,1./3.);
    Sn=1; S=Sn; n=0;
    while(fabs(fx-S) > eps && n < MAX_ITER) {
        Sn*=- (3*n+1)*x*x*x/(3*n+3);
        S+=Sn;
        n+=1;
    }
    printf("fx=%12.10lf\n",fx);
    printf(" S=%12.10lf\n",S);
    printf(" n=%1.0lf\n",n);
}
```

$$S_{n+1} = -\frac{3n+1}{3n+3} x^3 \cdot S_n$$

Хорошее решение



Задача о площади пересечения прямоугольников

- Постановка задачи
- Алгоритм
- Программа

Площадь пересечения прямоугольников: постановка задачи

- В программу последовательно поступают габариты прямоугольников a, b, c, d ($a \leq x \leq b, c \leq y \leq d$). Требуется, не запоминая габаритов всех прямоугольников, найти площадь их пересечения.

$$A = \max(a_i)$$

$$B = \min(b_i)$$

$$C = \max(c_i)$$

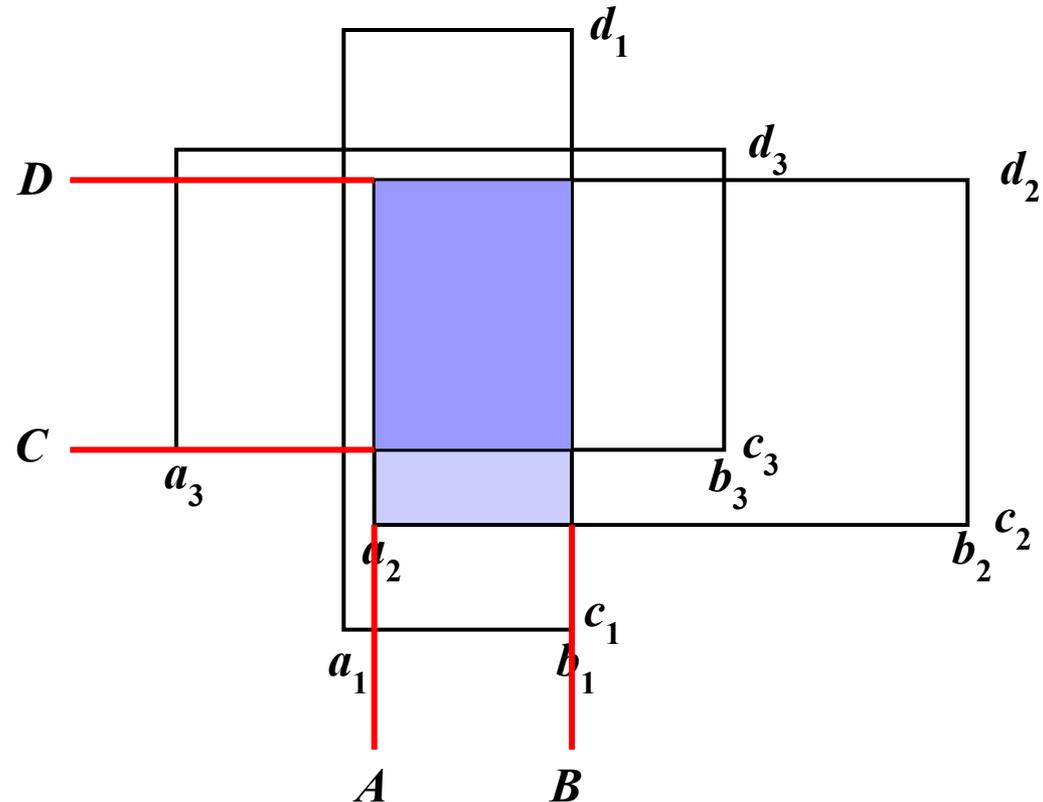
$$D = \min(d_i)$$

Если $A < B$ и $C < D$, то

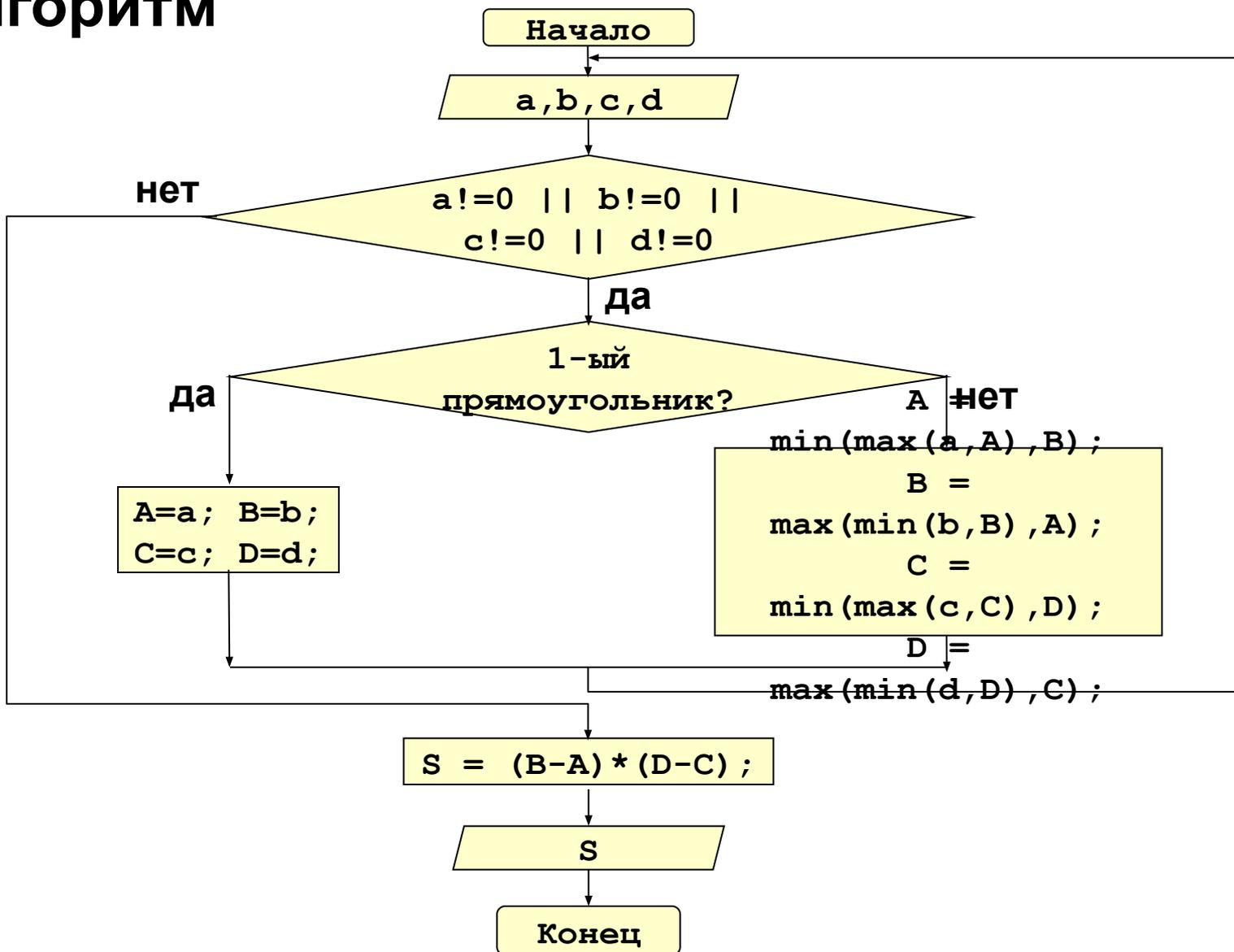
$$S = (B - A) \cdot (D - C),$$

иначе

$$S = 0.$$



Площадь пересечения прямоугольников: алгоритм



Площадь пересечения прямоугольников: программа

ДЕМО



Площадь пересечения прямоугольников: программа

```
#include <stdio.h>
#define min(x,y) ((x)<(y))?(x):(y)
#define max(x,y) ((x)>(y))?(x):(y)

void main() {
    float a,b,c,d,A,B,C,D,S=0;
    int first=1;
    do {
        printf("a, b, c, d:\n");
        scanf("%f%f%f%f",&a,&b,&c,&d);
        if (first) {
            A=a; B=b; C=c; D=d; first=0;
        } else {
            A = min(max(a,A),B);
            B = max(min(b,B),A);
            C = min(max(c,C),D);
            D = max(min(d,D),C);
        }
    } while(a!=0 || b!=0 || c!=0 || d!=0);
    S = (B-A)*(D-C);
    printf("S = %f\n",S);
}
```

Вопросы?

- Задача о разложении функции
 - Постановка задачи
 - Алгоритм
 - Программа 1
 - Замечание об эффективности
 - Программа 2
- Задача о площади пересечения прямоугольников
 - Постановка задачи
 - Алгоритм
 - Программа



Дубовая роща. Девочка и банан