



ФГОБУ ВПО "СибГУТИ"
Кафедра вычислительных систем

ПРОГРАММИРОВАНИЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ "Конструкция ветвления"

Преподаватель:

Доцент Кафедры ВС, к.т.н.

Поляков Артем Юрьевич



Вычисление корней квадратного уравнения

$$a \cdot x^2 + b \cdot x + c = 0$$

$$D = b^2 - 4 \cdot a \cdot c$$

- 1) Если $D < 0$ – **корней нет**
- 2) Если $D = 0$ – **корень один** и вычисляется по формуле:

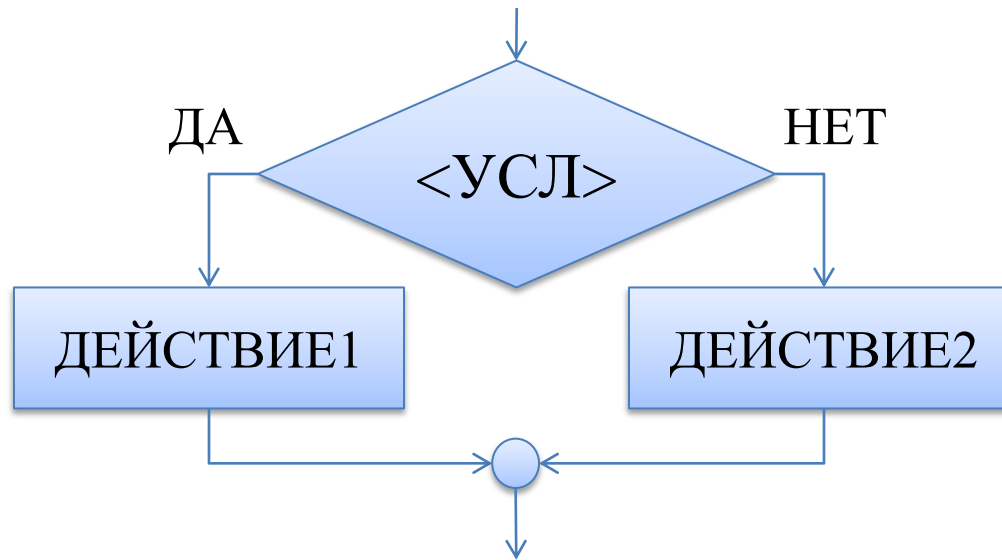
$$x = \frac{-b}{2a}$$

- 3) Если $D > 0$ – корней два, они вычисляются по формулам:

$$x_1 = \frac{-b - \sqrt{D}}{2a}; x_2 = \frac{-b + \sqrt{D}}{2a}$$



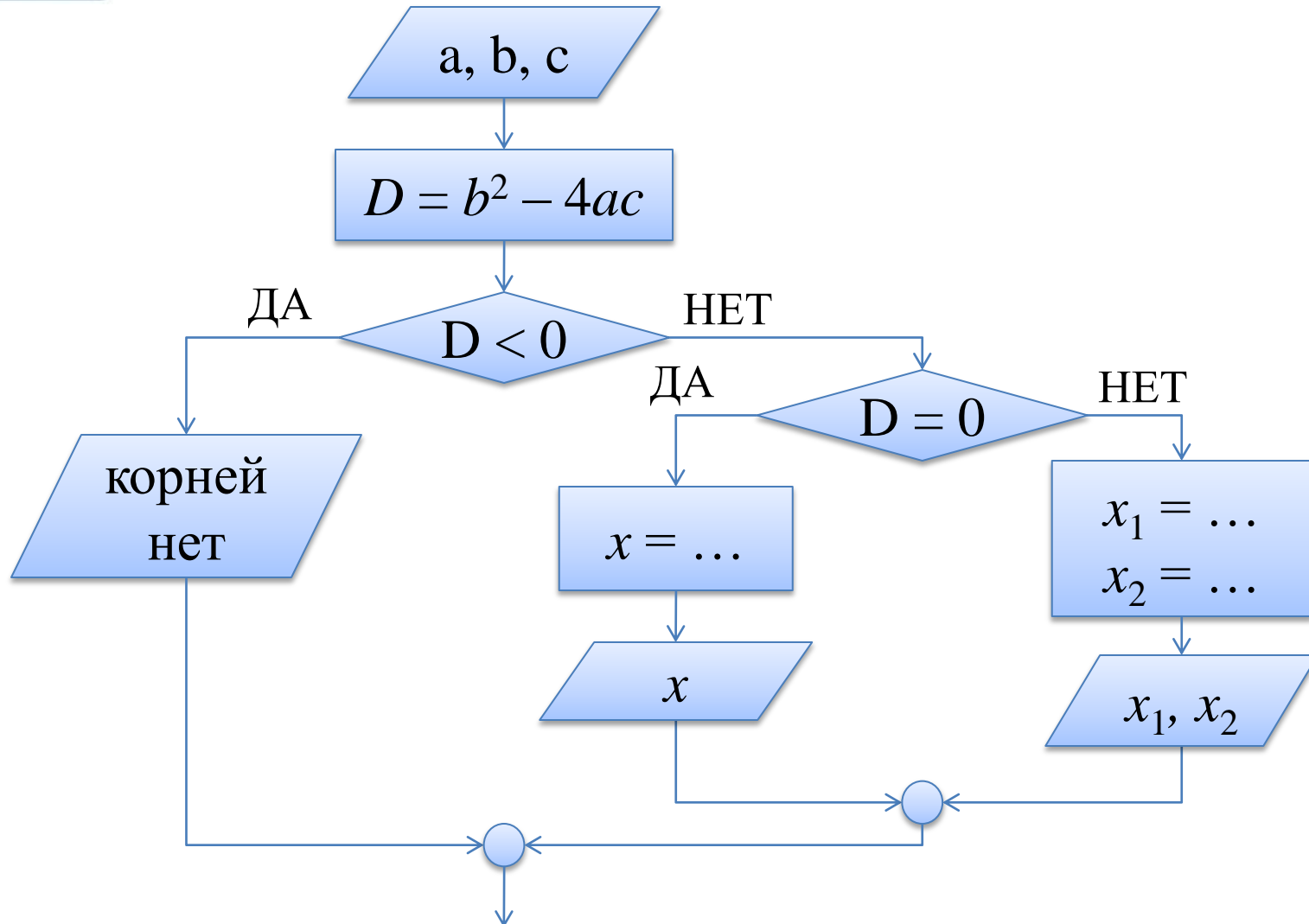
Конструкция if – ветвление



```
if ( <УСЛ> ) {  
    <ДЕЙСТВИЕ1>  
} else {  
    <ДЕЙСТВИЕ2>  
}
```



Алгоритм решения квадратного уравнения





Вычисление корней квадратного уравнения

roots.c

```
#include <stdio.h>
#include <math.h>
int main()
{
    float a, b, c, D, x1, x2;
    printf("Input a, b, c\n");
    scanf("%f %f %f", &a, &b, &c);
    D = b*b - 4*a*c;
    if( D < 0 ){
        printf("No real roots!\n");
    } else {
        if( D == 0 ){
            x1 = -b/(2*a);
            printf("x = %f\n", x1);
        } else {
            x1 = (-b - sqrt(D))/(2*a);
            x2 = (-b + sqrt(D))/(2*a);
            printf("x1 = %f, x2 = %f\n", x1, x2);
        }
    }
    return 0;
}
```



Компиляция roots.c

```
$ gcc -o roots roots.c  
/tmp/cckybsBN.o: In function `main':  
roots1.c:(.text+0xe1): undefined reference to `sqrt'  
roots1.c:(.text+0x10d): undefined reference to `sqrt'  
collect2: ld returned 1 exit status
```

- Функция sqrt расположена в библиотеке math.
- Несмотря на то, что в файле roots.c подключен заголовочный файл math.h подключение библиотеки не происходит.
- Непосредственное подключение должно производиться на этапе компиляции (в командной строке).

```
gcc -lm -o roots roots.c
```



Примеры входных данных

1)

$$1 \cdot x^2 + 3 \cdot x + 4 = 0$$

$$D = 3^2 - 4 \cdot 1 \cdot 4 = -7$$

*Действительных
корней нет*

2)

$$x^2 + 2 \cdot x + 1 = 0$$

$$D = 2^2 - 4 \cdot 1 \cdot 1 = 4 - 4 = 0$$

$$x = -2 / (2 \cdot 1) = -1$$

3)

$$4 \cdot x^2 + 5 \cdot x + 1 = 0$$

$$D = 5^2 - 4 \cdot 1 \cdot 4 = 9$$

$$x_1 = (-5 - 3) / 8 = -1$$

$$x_2 = (-5 + 3) / 8 = -1/4 = -0.25$$



Вычисление корней квадратного уравнения (2)

roots1.c

```
#include <stdio.h>
#include <math.h>
int main()
{
    float a, b, c, D, x1, x2;
    printf("Input a, b, c\n");
    scanf("%f %f %f", &a, &b, &c);
    D = b*b - 4*a*c;
    if( D < 0 ){
        printf("No real roots!\b");
    } else if( D == 0 ){
        x1 = -b/(2*a);
        printf("x = %f\n", x1);
    } else {
        x1 = (-b - sqrt(D))/(2*a);
        x2 = (-b + sqrt(D))/(2*a);
        printf("x1 = %f, x2 = %f\n", x1, x2);
    }
    return 0;
}
```




Трассировка roots1 в GDB

1. При компиляции программы необходимо указать специальный ключ: **-g**

```
gcc -g -lm -o roots1 roots1.c
```

2. Запуск программы производится под контролем GDB:

```
$ gdb roots1
GNU gdb (GDB) 7.2-ubuntu
Copyright (C) 2010 Free Software Foundation, Inc.
...
Reading symbols from ../roots1...done.
(gdb)
```



Просмотр листинга программы (ИСХОДНЫХ КОДОВ)

```
(gdb) list
1      #include <stdio.h>
2      #include <math.h>
3
4      int main()
5      {
6          float a, b, c, D, x1, x2;
7          printf("Input a, b, c\n");
8          scanf("%f %f %f", &a, &b, &c);
9          D = b*b - 4*a*c;
10         if( D < 0 ){
(gdb) list
11             printf("No real roots!\b");
12         } else if( D == 0 ){
13             x1 = -b/(2*a);
14             printf("x = %f\n", x1);
15         } else {
16             x1 = (-b - sqrt(D))/(2*a);
17             x2 = (-b + sqrt(D))/(2*a);
18             printf("x1 = %f, x2 = %f\n", x1, x2);
19         }
20         return 0;
(gdb) list
21     }
22
(gdb) list
Line number 23 out of range; roots1.c has 22 lines.
```



Просмотр листинга main с начала

```
(gdb) list main
1      #include <stdio.h>
2      #include <math.h>
3
4      int main()
5      {
6          float a, b, c, D, x1, x2;
7          printf("Input a, b, c\n");
8          scanf("%f %f %f", &a, &b, &c);
9          D = b*b - 4*a*c;
10         if( D < 0 ){
(gdb)
```



Запуск программы №1

(для входных данных нет действ. корней)

```
(gdb) b main
```

```
Breakpoint 1 at 0x80484cd: file roots1.c, line 7.
```

```
(gdb) r
```

```
Starting program: .../src/roots1
```

```
Breakpoint 1, main () at roots1.c:7
```

```
7      printf("Input a, b, c\n");
```

```
(gdb) next
```

```
Input a, b, c
```

```
8      scanf("%f %f %f", &a, &b, &c);
```

```
(gdb) next
```

```
1 3 4
```

```
9      D = b*b - 4*a*c;
```

```
(gdb) next
```

```
10     if( D < 0 ){
```

```
(gdb) inspect D
```

```
$1 = -7
```

```
(gdb) next
```

```
11     printf("No real roots!\b");
```

```
(gdb) next
```

```
20     return 0;
```

```
(gdb) next
```

```
21 }
```

После запуска программа останавливается на первом операторе

Выполнение будет приостановлено, пока не будут введены данные

В данной точке происходит принятие решения о том какая из ветвей выполняется



Запуск программы №2 (один действительных корня)

```
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: . . . /src/roots1
Breakpoint 1, main () at roots1.c:7
7          printf("Input a, b, c\n");
(gdb) next
Input a, b, c
8          scanf("%f %f %f", &a, &b, &c);
(gdb) next
1 2 1
9          D = b*b - 4*a*c;
(gdb) next
10         if( D < 0 ){
(gdb) inspect D
$2 = 0
(gdb) next
12         } else if( D == 0 ){
(gdb) next
13             x1 = -b/(2*a);
(gdb) next
14             printf("x = %f\n", x1);
(gdb) inspect x1
$3 = -1
(gdb) n
x = -1.000000
```

Так как $D = 0$, то выполнение
будет продолжено по ветке
"else"



Запуск программы №3 (два действительный корень)

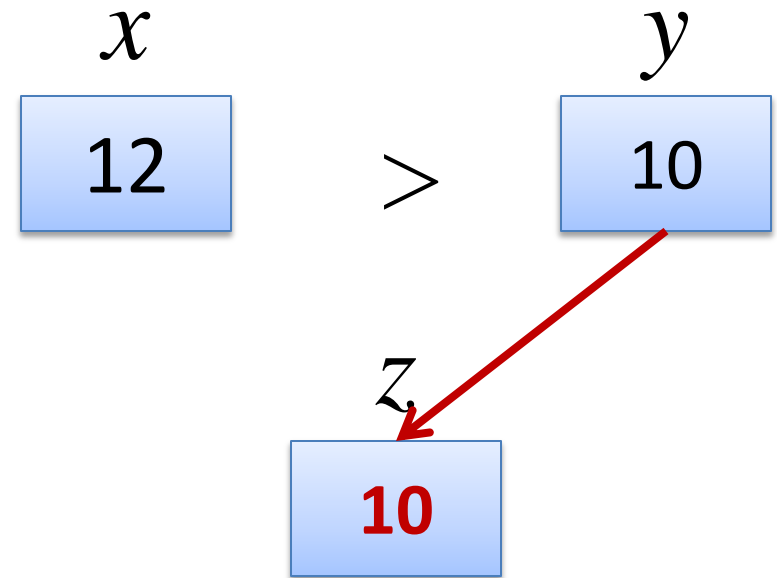
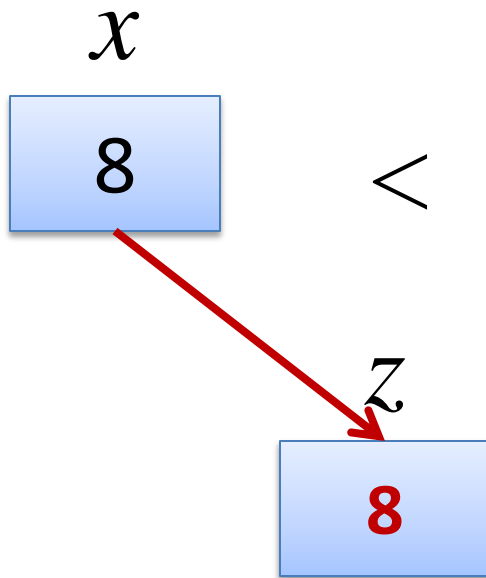
```
(gdb) r
. . .
Breakpoint 1, main () at roots1.c:7
7   printf("Input a, b, c\n");
. . .
8   scanf("%f %f %f", &a, &b, &c);
(gdb) next
4 5 1
9   D = b*b - 4*a*c;
. . .
(gdb) next
16 x1 = (-b - sqrt(D))/(2*a);
(gdb) inspect x1
$6 = -1.99953175
(gdb) inspect x2
$7 = 3.98799683e-34
(gdb) next
17 x2 = (-b + sqrt(D))/(2*a);
(gdb) next
18 printf("x1 = %f, x2 = %f\n", x1, x2);
(gdb) inspect x1
$10 = -1
(gdb) inspect x2
$11 = -0.25
```

```
(gdb) l main
1   #include <stdio.h>
2   #include <math.h>
3
4   int main()
5   {
6       float a, b, c, D, x1, x2;
7       printf("Input a, b, c\n");
8       scanf("%f %f %f", &a, &b, &c);
9       D = b*b - 4*a*c;
10      if( D < 0 ){
11          printf("No real roots!\n");
12      } else if( D == 0 ){
13          x1 = -b/(2*a);
14          printf("x = %f\n", x1);
15      } else {
16          x1 = (-b - sqrt(D))/(2*a);
17          x2 = (-b + sqrt(D))/(2*a);
18          printf("x1 = %f, x2 = %f\n",
19                x1, x2);
20      }
21      return 0;
```



Вычисление минимального из двух чисел (вариант 1)

$$\min(x, y) = \begin{cases} x, & x < y \\ y, & y < x \end{cases}$$





Программа вычисления минимума двух целых чисел

min.c

```
#include <stdio.h>
#include <math.h>
int main()
{
    int x, y, z;
    printf("Input x, y: ");
    scanf("%d %d", &x, &y);
    if( x < y ){
        z = x;
    }else{
        z = y;
    }
    printf("min(%d, %d) = %d\n", x, y, z);
    return 0;
}
```




Программа вычисления минимума двух целых чисел (2)

В программе допущена ошибка (подсвечено красным цветом).
Продемонстрировать как с помощью отладчика обнаружить
указанную ошибку.

min_e.c

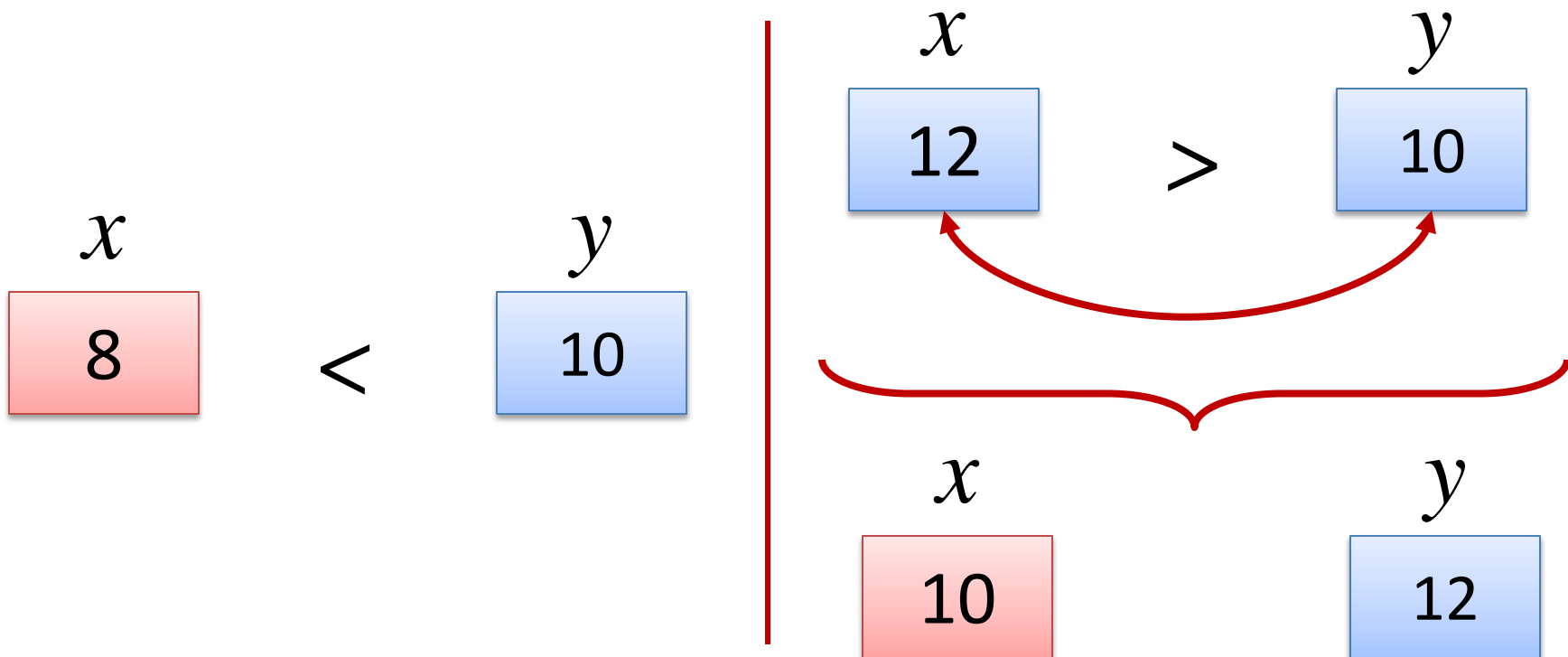
```
#include <stdio.h>
#include <math.h>
int main()
{
    int x, y, z;
    printf("Input x, y: ");
    scanf("%d %d", &x, &y);
    if( x > y ){
        z = x;
    }else{
        z = y;
    }
    printf("min(%d, %d) = %d\n", x, y, z);
    return 0;
}
```

При наборе текста программы знак "<" был по ошибке заменен на ">".
Ошибка проявится только если $x > y$!
Для отладки необходимо выполнить трассировку программы с проблемными данными



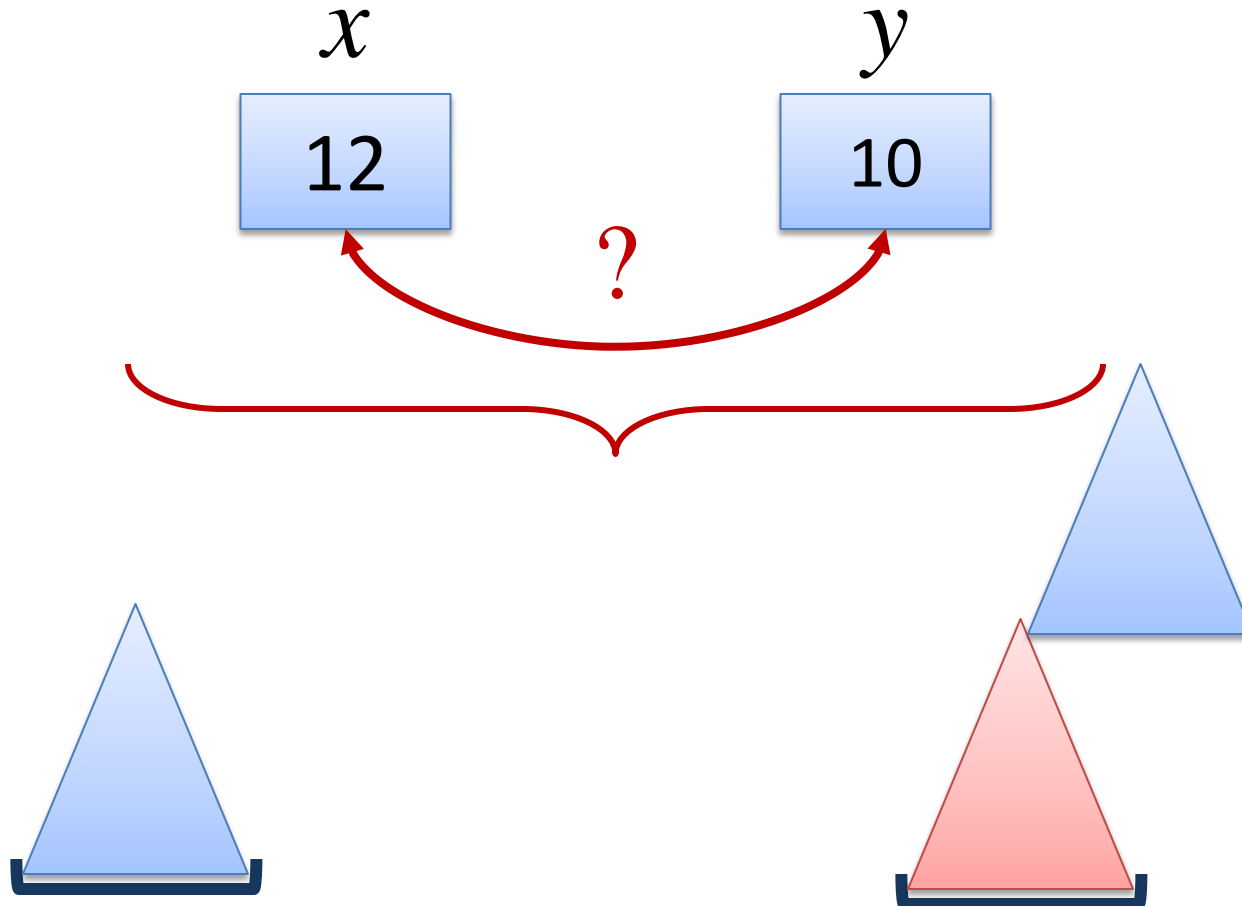
Вычисление минимального из двух чисел (вариант 2 – минимум в x)

$$\min(x, y) = \begin{cases} x, & x < y \\ y, & y < x \end{cases}$$



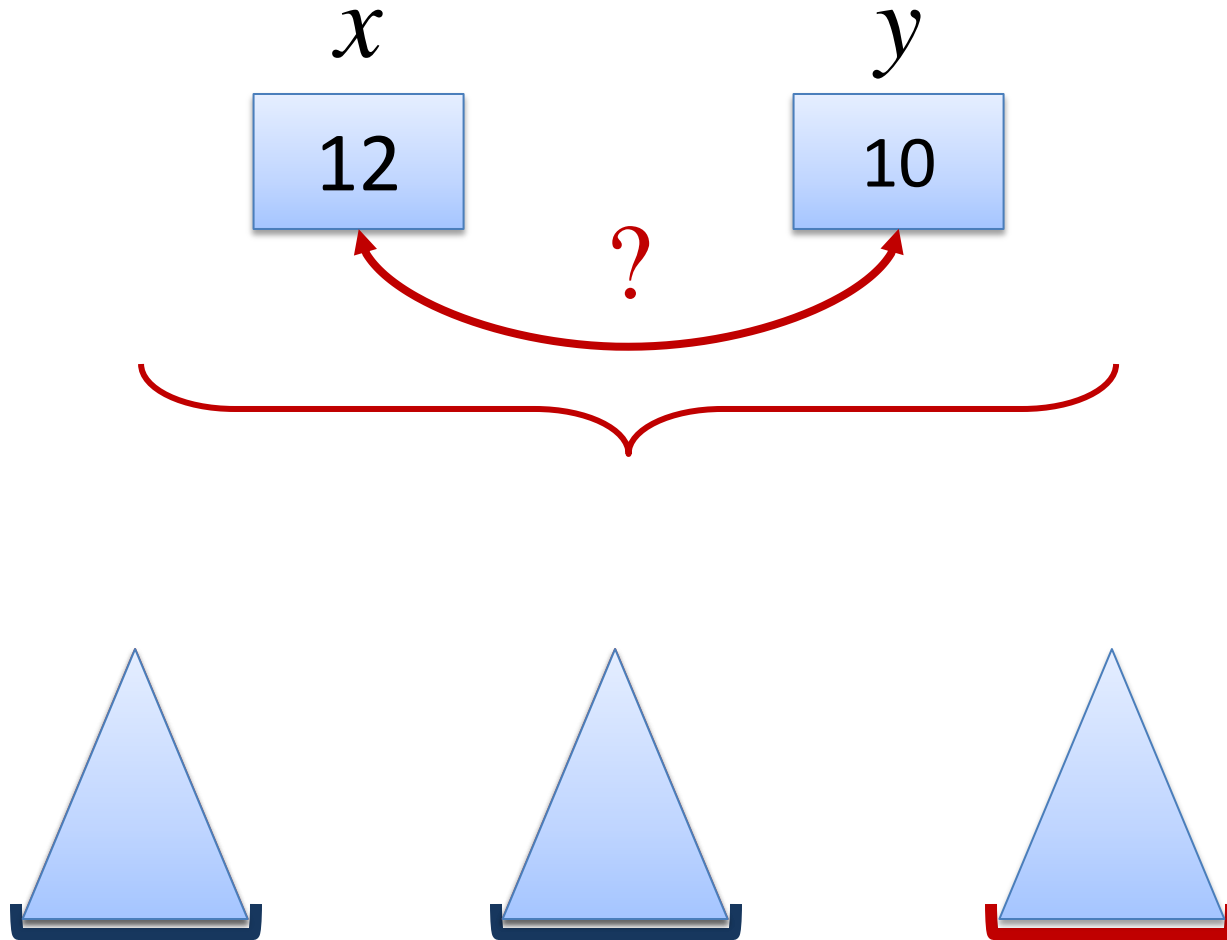


Обмен значений двух ячеек



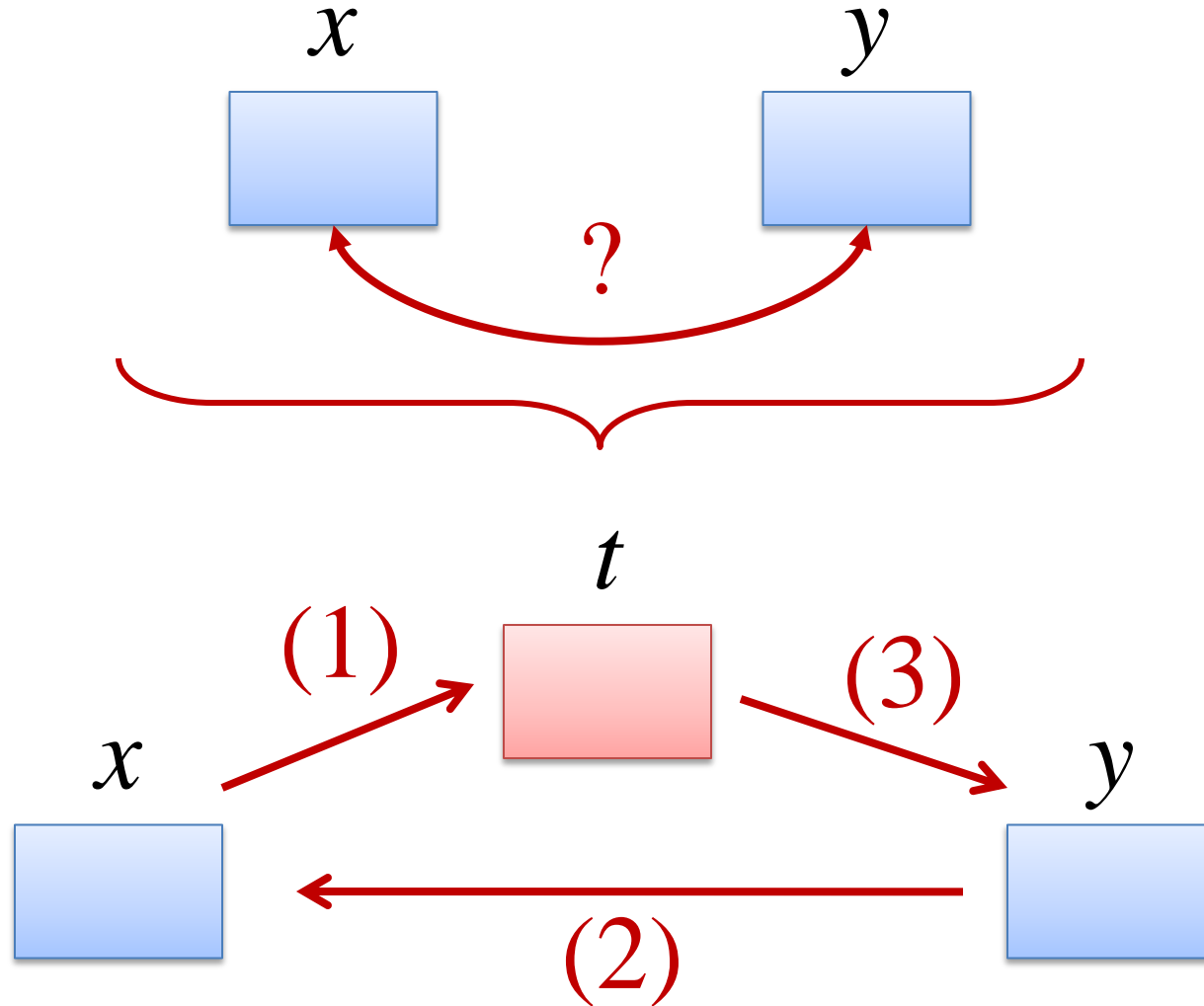


Обмен значений двух ячеек



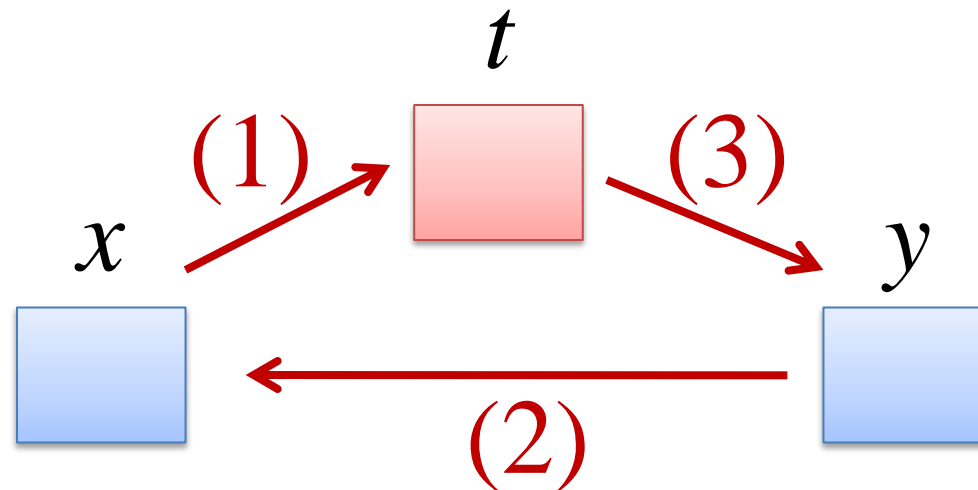


Обмен значений двух ячеек





Обмен значений двух ячеек



**Последовательность шагов: $x=5$, $y = 10$.
АЛГОРИТМ?**



Программа вычисления минимума двух целых чисел (с ошибкой!)

min2_er.c

```
#include <stdio.h>
#include <math.h>
int main()
{
    int x, y, t;
    printf("Input x, y: ");
    scanf("%d %d", &x, &y);
    printf("min(%d, %d) = ", x, y);
    if( x > y ){
        x = y;
        t = x;
        y = t;
    }
    printf("x=%d, y=%d\n", x, y);
    return 0;
}
```

В программе допущена
ошибка.
Продемонстрировать как с
помощью отладчика
обнаружить указанную
ошибку.



Программа вычисления минимума двух целых чисел (исправленная) `min2.c`

```
#include <stdio.h>
#include <math.h>
int main()
{
    int x, y, t;
    printf("Input x, y: ");
    scanf("%d %d", &x, &y);
    printf("min(%d, %d) = ", x, y);
    if( x > y ){
        t = x;
        x = y;
        y = t;
    }
    printf("%d\n", x);
    return 0;
}
```




Задача вычисления минимального числа в последовательности произвольной длины

Дано:

На вход с клавиатуры вводятся последовательность
 $a = \{ a_1, a_2, \dots, a_{n-1}, a_n \}$, состоящая из n чисел.

Требуется:

Определить значение наименьшего числа в
последовательности

АЛГОРИТМ?