Примеры разработки программ-функций в системе MATHCAD

Лекция 7

Пример: задача 1.4.2.30

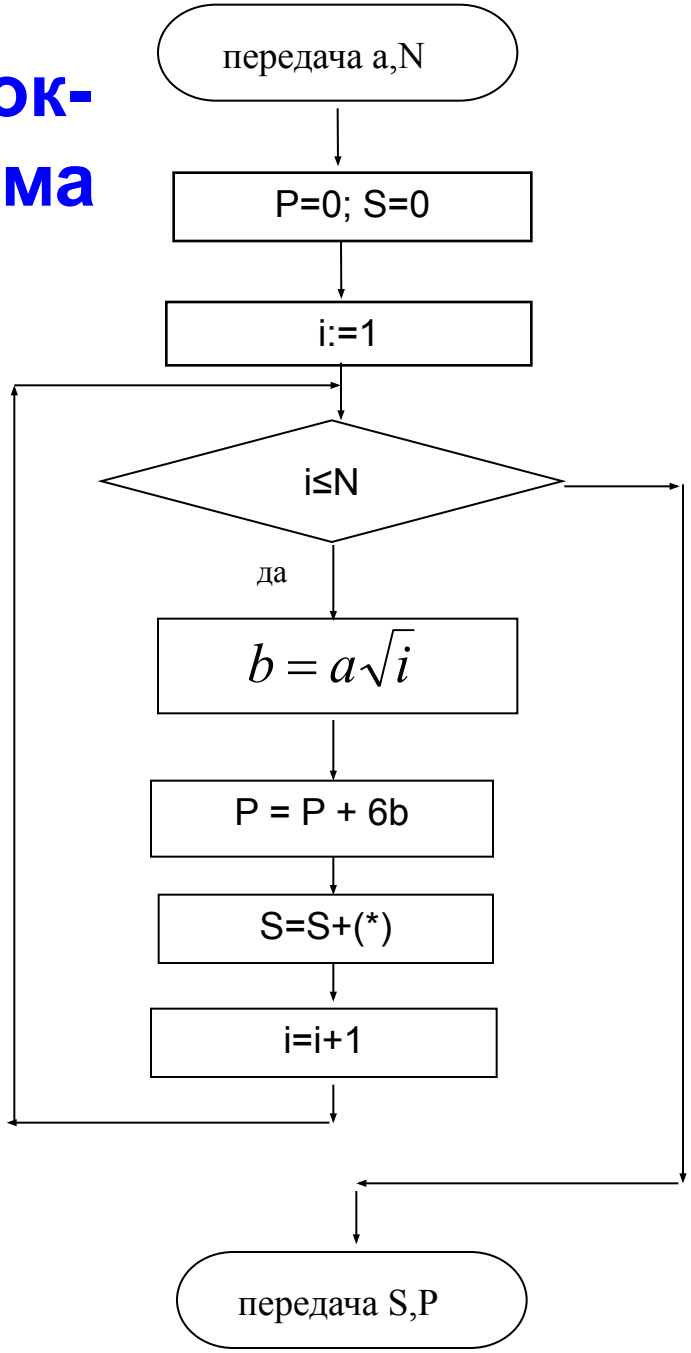
Условие

Найти суммарную площадь и сумму периметров *N* правильных шестиугольников, длины сторон которых образуют последовательность: $a, a\sqrt{2}, ..., a\sqrt{N}$.

Состав данных

Имя	Смысл	Тип	Структура
Исходные данные			
a	Длина стороны 1-го шестиугольника	Веществ.	Простая переменная
N	Количество шестиугольников	Целый	Простая переменная
Выходные данные			
P	Сумма периметров шестиугольников	Веществ.	Простая переменная
S	Сумма площадей шестиугольников	Веществ.	Простая переменная
Промежуточные данные			
i	Счетчик шестиугольников	Целый	Простая переменная
b	Длина стороны текущего шестиугольника	Веществ.	Простая переменная

Блок-схема



Определение функции

```
SummaS_P_6_ugol(a,N) := | S ← 0
                        | P ← 0
                        | for i ∈ 1..N
                        |     b ← a·√i
                        |     P ← P + 6·b
                        |     S ← S + 3·(√3/2)·b^2
                        | (S P)
```

Примеры вызова функции

$$Z := SummaS_P_6_ugol(1,5)$$

$$Z = \begin{pmatrix} 21.778 \\ 8.382 \end{pmatrix}$$

$$SummaS_P_6_ugol(3,6) = \begin{pmatrix} 84.426 \\ 32.495 \end{pmatrix}$$

Периметр шестиугольника: p=6b; площадь шестиугольника:

$$s = \frac{3\sqrt{3}}{2} b^2 \quad (*)$$

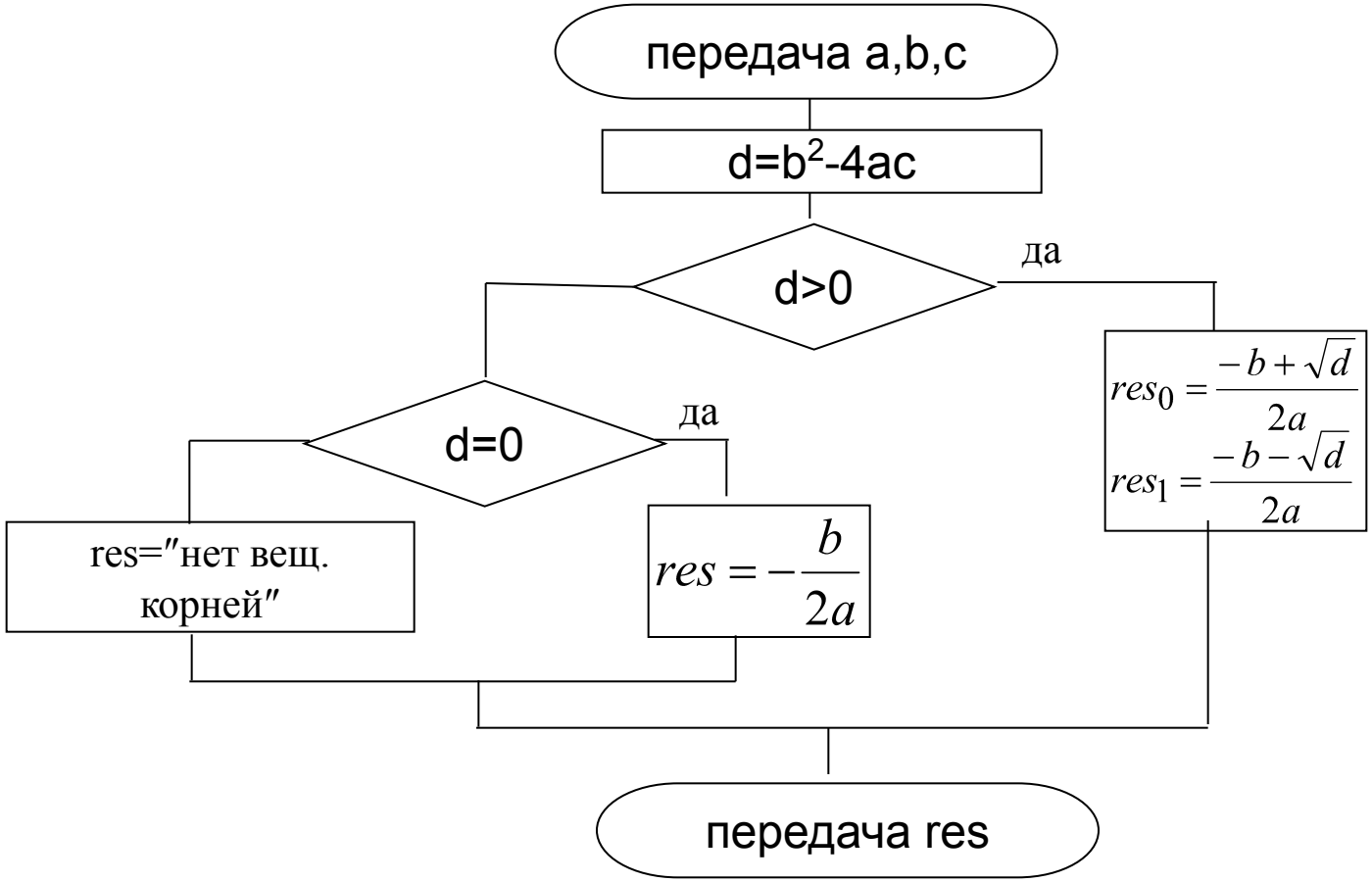
Найти вещественные корни квадратного уравнения $ax^2+bx+c=0$

Состав данных

Имя	Смысл	Тип	Структура
	Исходные данные		
a	коэффициенты уравнения	Веществ.	Простая переменная
b			
c			
	Выходные данные		
res	Корни уравнения	Веществ.	Одномерный массив из двух элементов
	Сообщение об отсутствии вещ. корней	строковый	Простая переменная
	Промежуточные данные		
d	Дискриминант уравнения $d=b^2-4ac$	Целый	Простая переменная

альтернатива

Блок-схема алгоритма нахождения вещественных корней квадратного уравнения



Программа-функция вычисления вещественных корней квадратного уравнения

```
kv_ur1(a,b,c) := | d ← b^2 − 4·a·c
                 | if d > 0
                 |     | res_0 ← (−b + √d) / (2·a)
                 |     | res_1 ← (−b − √d) / (2·a)
                 | otherwise
                 |     | res ← −b / (2·a)   if d = 0
                 |     | res ← "нет вещественных корней"   otherwise
                 | res
```

Обратите внимание:
и тип, и структура переменной res выбирается динамически, в зависимости от значения d

Примеры вызовов программы-функции kv_ur1

$$kv_ur1(1,5,-4) = \begin{pmatrix} 0.702 \\ -5.702 \end{pmatrix}$$

kv_ur1(1, −2, 5) = "нет вещественных корней"

kv_ur1(1, −2, 1) = 1

Пример решения задачи со сложным условием продолжения цикла

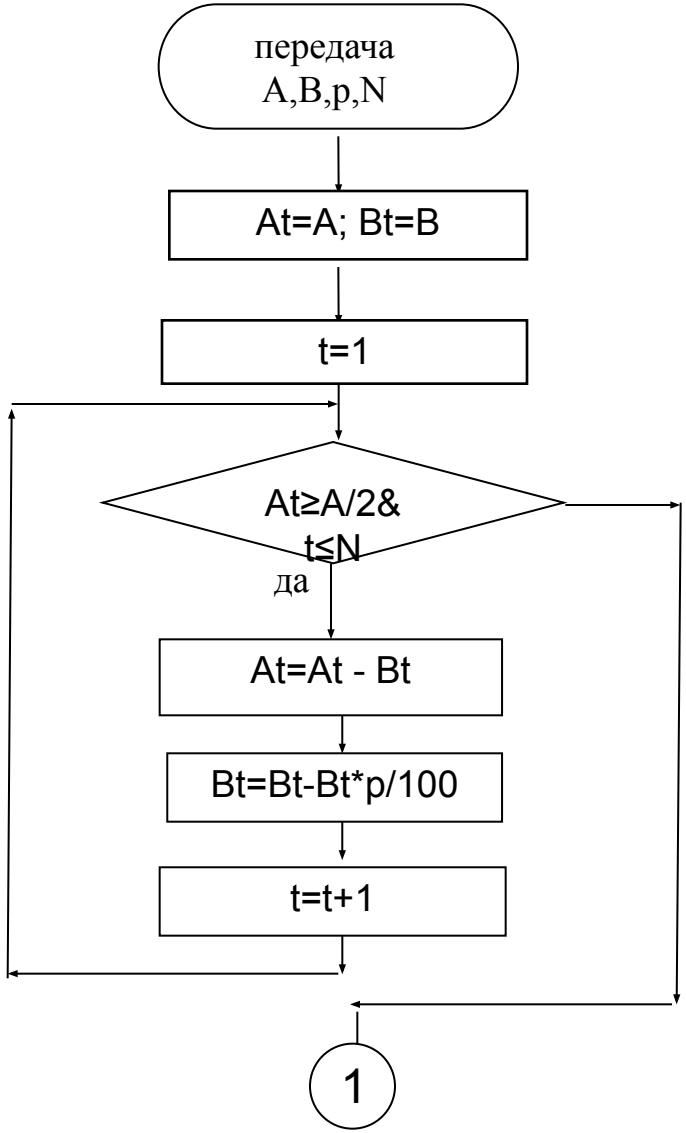
Условие

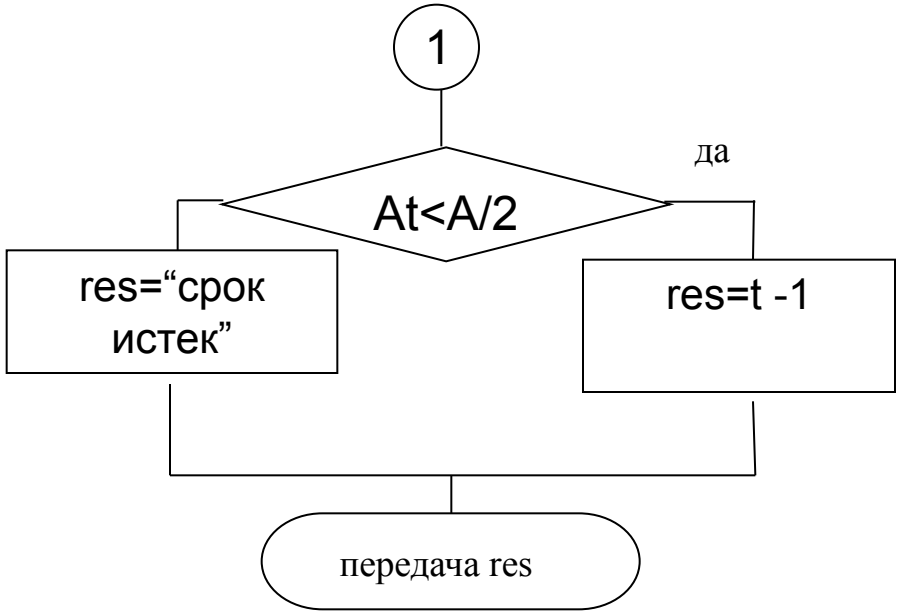
A - начальная стоимость оборудования. В первый год эксплуатации стоимость оборудования снижается на *B* руб., а в каждый следующий год снижение стоимости уменьшается на *p* % (относительно предыдущего года). Определить, через сколько лет стоимость оборудования станет меньше *A*/2. Рассматривать срок не более *N* лет.

Состав данных

Имя	Смысл	Тип	Структура
Исходные данные			
A	начальная стоимость оборудования	вещественный	простая переменная
B	начальное снижение стоимости	вещественный	простая переменная
p	процентное уменьшение снижения стоимости	вещественный	простая переменная
N	максимальное число лет	целый	простая переменная
Выходные данные			
res	искомый номер года сообщение «срок истек»	целый строковый	простая переменная
Промежуточные переменные			
t	текущий номер года	целый	простая переменная
At	текущая стоимость оборудования	вещественный	простая переменная
Bt	текущее ежегодное снижение стоимости	вещественный	простая переменная

Блок-схема алгоритма





Программа-функция

z1_3_4(A,B,p,N):=

At←A

Bt←B

t←1

while $(At \geq \frac{A}{2}) \wedge (t \leq N)$

  At←At-Bt

  Bt←Bt+0.01·Bt·p

  t←t+1

res ←t-1 if $At < \frac{A}{2}$

res←"срок прошел" otherwise

res

Вызов функции:

z1_3_4(1000,100,1,10)=5 z1_3_4(1000,100,1,10)="срок прошел"

y:=z1_3_4(1000,100,20,10) y=3

Решение задачи методом флажка

Промежуточная переменная:

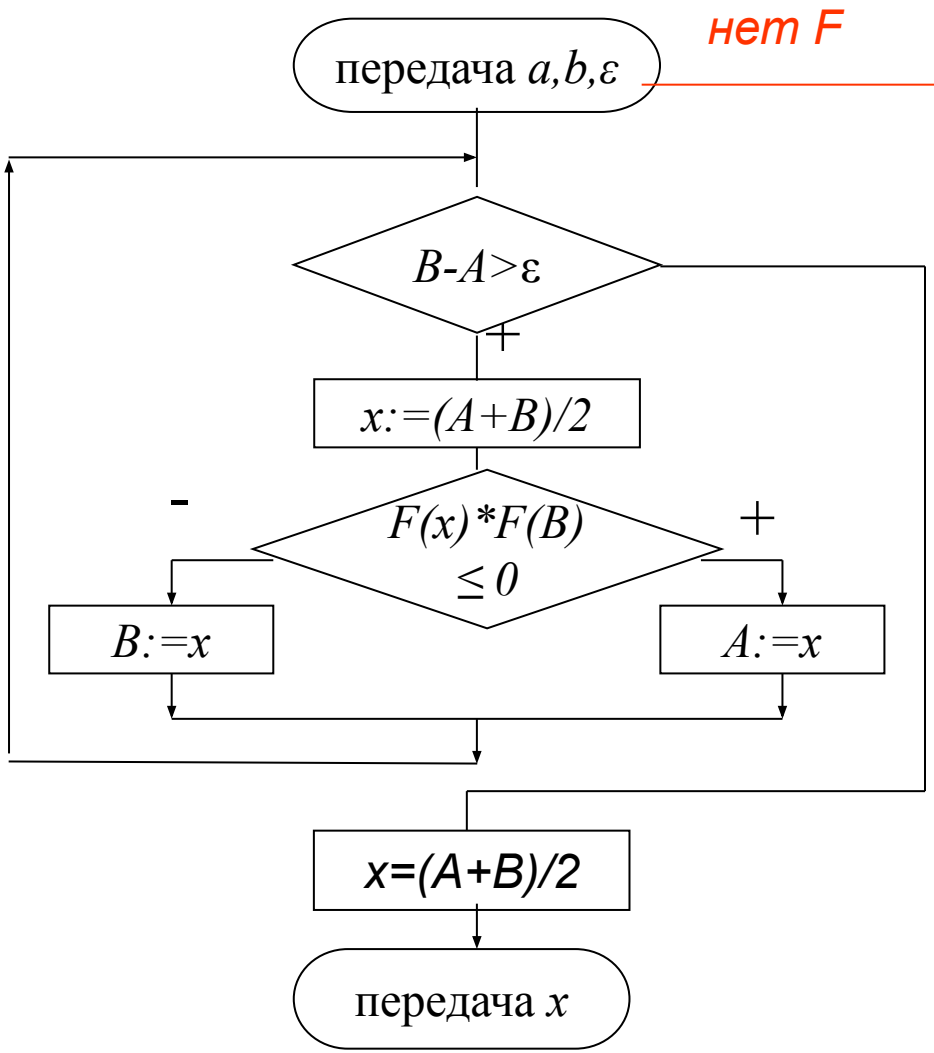
$$f = \begin{cases} 1, \text{если цикл надо продолжить} \\ 0, \text{если надо выйти из цикла} \end{cases}$$

Блок-схема алгоритма с использованием метода флажка

Программа-функция (метод флажка)

```
z1_3_4_v2(A,B,p,N):= | At←A
                     | Bt←B
                     | f←1
                     | t←1
                     | while f
                     |     if At<A/2
                     |         res←t-1
                     |         f←0
                     |     otherwise
                     |         if t>N
                     |             res←″срок прошел″
                     |             f←0
                     |         otherwise
                     |             At←At - Bt
                     |             Bt←Bt-0.01 · p · Bt
                     |             t←t+1
                     | res
```

Решение уравнения методом деления отрезка пополам



Как в системе Mathcad определить функцию, параметром которой является функция?

Программа-функция

$F(x):=x^2-0.25$

$$\text{Root1}(a,b,e):=\left|\begin{array}{l}\text{while } b-a>e\\ \quad\left|\begin{array}{l}x\leftarrow\dfrac{a+b}{2}\\ a\leftarrow x \quad \text{if } F(x)\cdot F(b)\le 0\\ b\leftarrow x \text{ otherwise}\end{array}\right.\\ x\leftarrow\dfrac{a+b}{2}\\ x\end{array}\right.$$

Функция Root1 вызывает только функцию F! Нет возможности подставить другую функцию!

Вызов программы-функции:

Root1(0.1,1,10-3)=0.5

Встроенная функция root

скаляры

root(*Выражение*, *Имя_переменной*) – возвращает значение переменной, при котором выражение равно 0, с точностью (погрешностью), задаваемой системной переменной TOL.

Функция **root** решает уравнение:

***Выражение*=0**

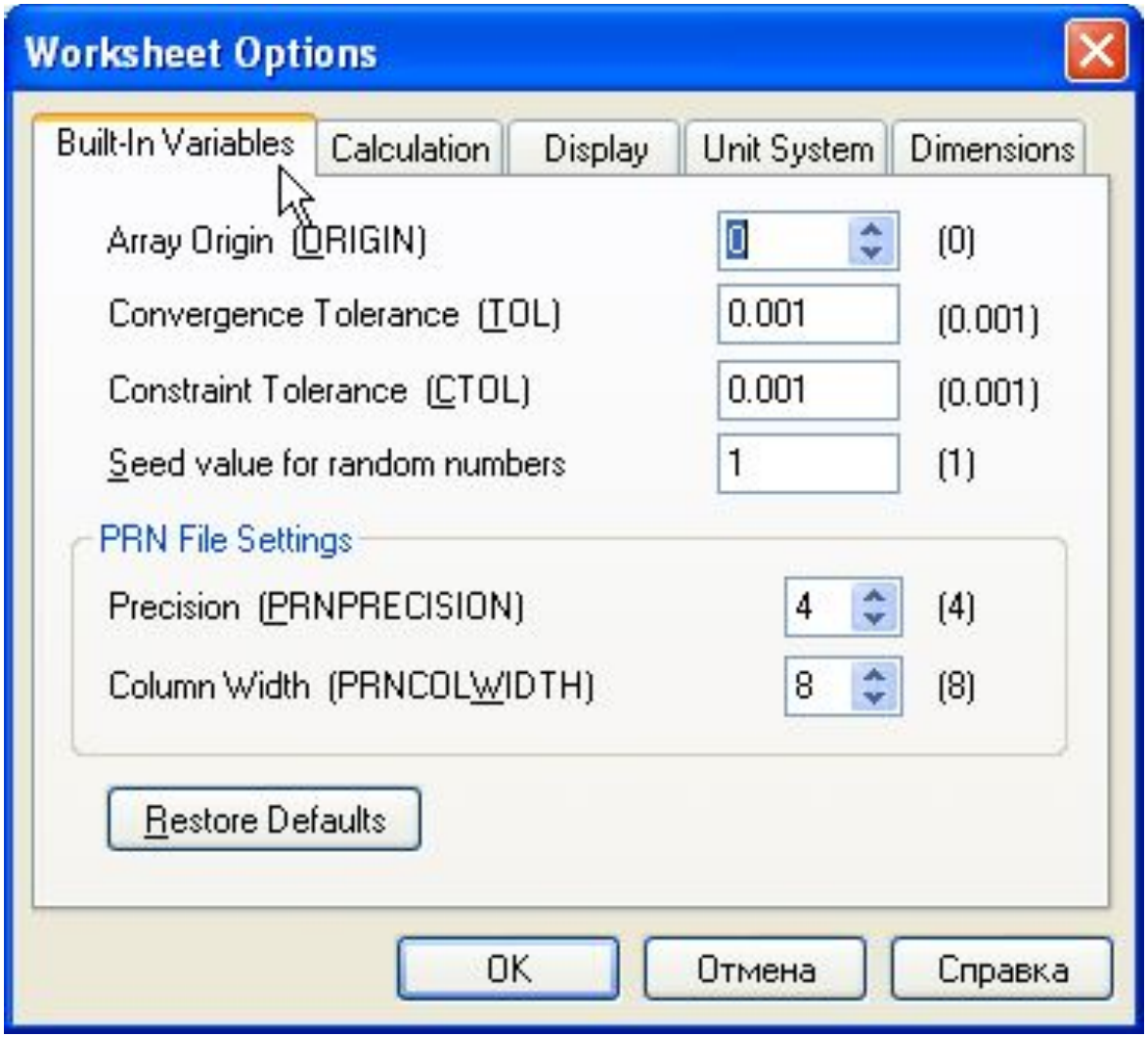
Например,
F(x)=0

Или
$x^3-6x^2+21x-52=0$

До вызова функции root переменной **x** должно быть присвоено начальное значение (начальное приближение корня).

Установка значений встроенных переменных

Tools / Worksheet Options / Built-in Variables (Сервис / Опции документа / Встроенные переменные)



Примеры решения уравнения с помощью root

Пример 1

$F(x):=x^3-6\cdot x^2+21\cdot x-52$

$x:=0$

$x1:=root(F(x),x)$ $x1=4$

root может вычислять как вещественные, так и комплексные корни

Пример 2

$G(a,x):=root(exp(x)-a\cdot x^2, x)$

$a:=1..5$

$x_0:=0$ $x_a:=G(a, x_{a-1})$

a=

1
2
3
4
5

x=
0
-0.704
-0.54
-0.459
-0.408
-0.371

Примеры программ с массивами

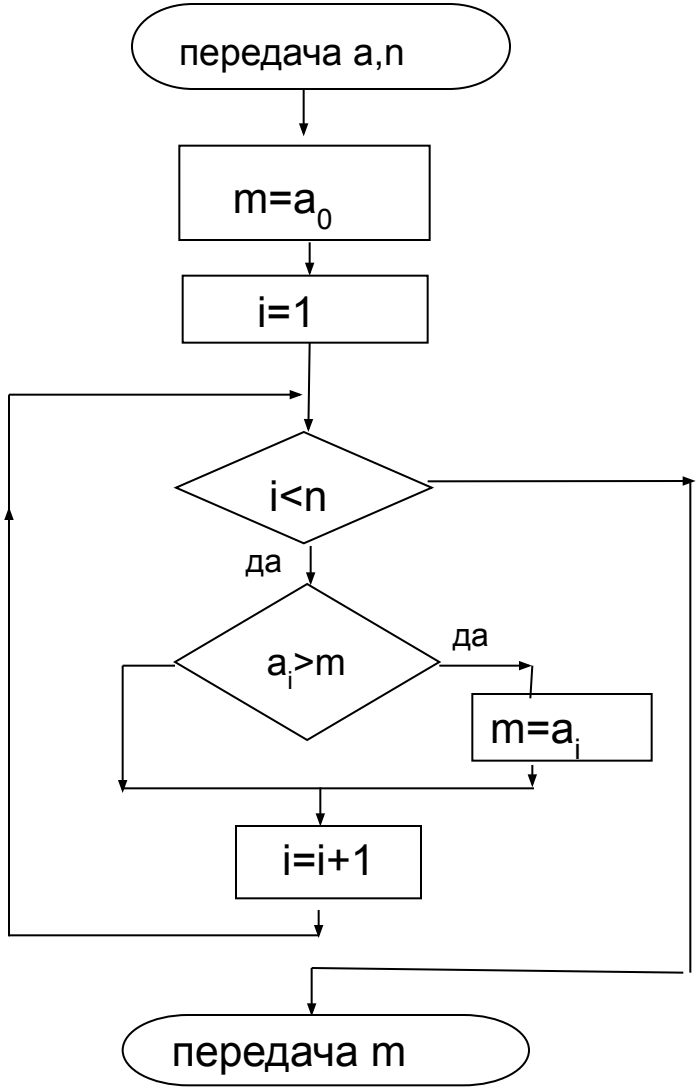
Дан массив **a** из **n** элементов. Найти максимальное значение элементов массива.

Состав данных

Имя	Смысл	Тип	Структура
Исходные данные			
n	число элементов массива	целый	простая переменная
a	заданный массив	вещественный	одномерный массив
Выходные данныв			
m	максимальное значение элементов массива	вещественный	простая переменная
Промежуточные данные			
i	счетчик элементов массива	целый	простая переменная

Блок-схема алгоритма

Системная (встроенная) переменная ORIGIN=0



Код программы-функции

$$mx(a,n) := \left| \begin{array}{l} m \leftarrow a_0 \\ \text{for } i \in 1..n-1 \\ \quad m \leftarrow a_i \text{ if } m < a_i \\ m \end{array} \right.$$

Вызов программы-функции

$$a = \begin{pmatrix} 2 \\ -1 \\ 6 \\ 0 \end{pmatrix}$$

$$mx(a,4) = 6$$

С помощью панели «Матрицы»

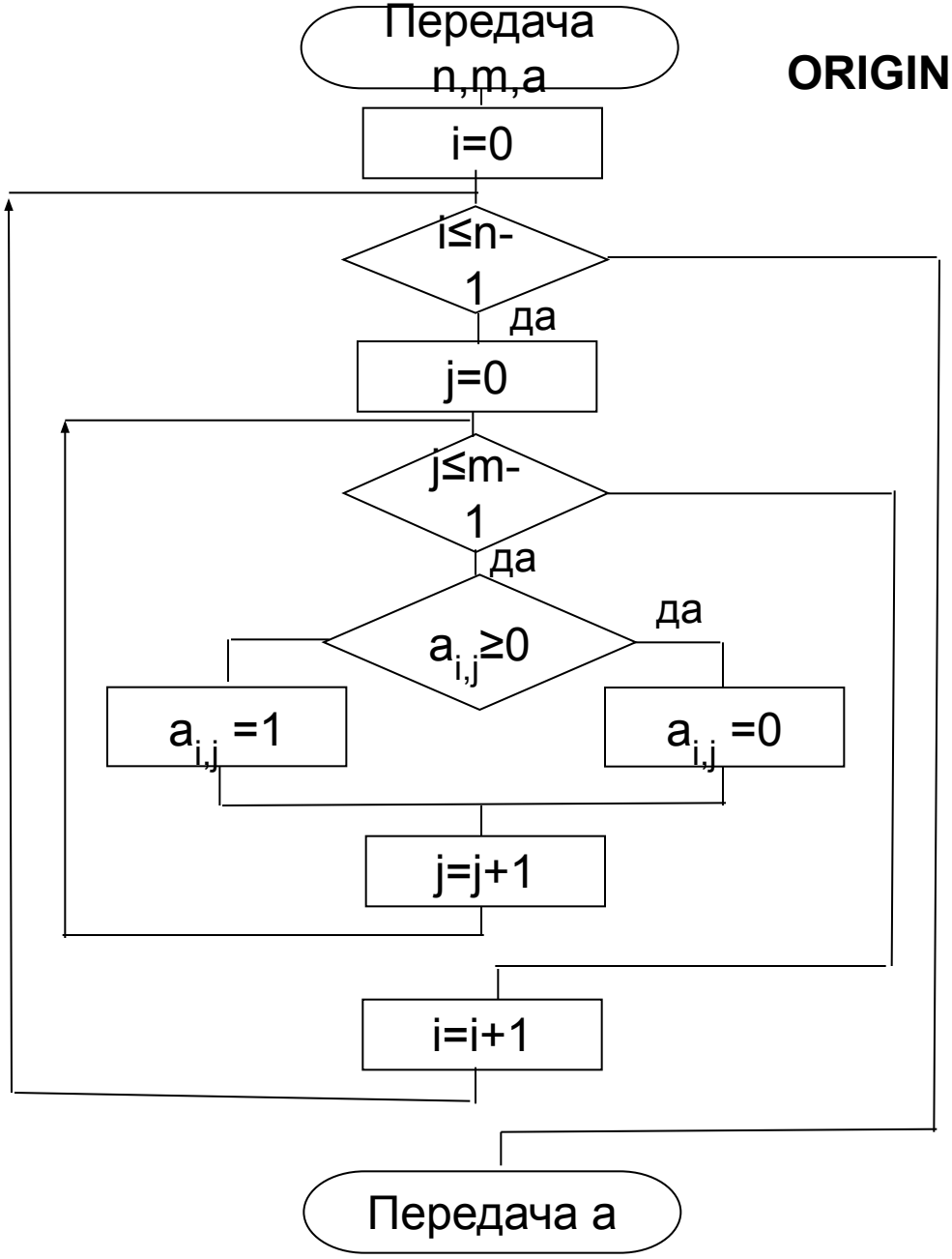
Примеры программ с массивами

Дана матрица **a** из **n** строк и **m** столбцов. Неотрицательные элементы матрицы заменить на 0, а отрицательные – на 1.

Состав данных

Имя	Смысл	Тип	Структура
Исходные данные			
n	число строк матрицы	целый	простая переменная
m	число столбцов матрицы		число строк матрицы
a	заданная матрица	вещественный	двумерный массив
Выходные данныв			
a	Измененная матрица	вещественный	двумерный массив
Промежуточные данные			
i	счетчик строк	целый	простая переменная
j	счетчик столбцов	целый	простая переменная

Блок-схема алгоритма



Программа-функция и ее вызов

ORIGIN=0

zamena(n,m,a):=
- for i∈0..n-1
  - for j∈0..m-1
    - $a_{i,j} \leftarrow 0$ if $a_{i,j} \geq 0$
    - $a_{i,j} \leftarrow 1$ otherwise
  - a

$$D := \begin{pmatrix} -3 & 0 & 2 \\ 1 & -2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

T:=zamena(3,3.D)

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Встроенные функции, определяющие размеры массивов

rows(**A**) – возвращает число строк в матрице **A.** Если **A** – скаляр, возвращается 0.

cols(**A**) – возвращает число столбцов в матрице A. Если **A** – скаляр, возвращается 0.

length(**v**) – возвращает число элементов вектора **v.**

last(**v**) – возвращает индекс последнего элемента в векторе **v.**

Улучшенный код

```
Fun2(a) :=
    res ← 0
    n ← rows(a)
    m ← cols(a)
    if ORIGIN = 0
        i1 ← 0
        j1 ← 0
        n1 ← n − 1
        m1 ← m − 1
    otherwise
        if ORIGIN = 1
            i1 ← 1
            j1 ← 1
            n1 ← n
            m1 ← m
        res ← "error!"   otherwise
    for i ∈ i1..n1
      for j ∈ j1..m1
          a_{i,j} ← 0   if a_{i,j} ≥ 0
          a_{i,j} ← 1   otherwise
    res ← a   if res = 0
    res
```

⊕1. Нет зависимости от значения ORIGIN.

⊕2. Нет необходимости задавать размеры матрицы

Еще более совершенный код

```
Fun3(a):=
    n←rows(a)
    m←cols(a)
    i1 ←ORIGIN
    n1←n+ORIGIN-1
    j1 ←ORIGIN
    m1←m+ORIGIN-1
    for i∈ i1..n1
        for j∈ j1..m1
            a[i,j] ←0   if a[i,j]≥0
            a[i,j] ←1   otherwise
    a
```

⊕1. Нет зависимости от значения ORIGIN (может быть ORIGIN≠0, ORIGIN≠1).

⊕2. Еще короче.

Цикл for «для каждого элемента массива»

```
Sum(a):= | s←0
         | for v∈ a
         |     | s←s+v
         | s
```

НО: изменение v не приводит к изменению элемента матрицы!

$$x = \begin{pmatrix} 1 \\ 4 \\ 5 \end{pmatrix} \quad A = \begin{pmatrix} -1 & -2 & -3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

Одна функция и для вектора, и для матрицы

Sum(x)=10 Sum(A)=6

Выделение подпрограмм в процессе решения задачи

Условие

Проверить, упорядочены ли по возрастанию элементы третьего столбца матрицы A(4×6)и элементы пятого столбца матрицы B(5×5)

Решали эту задачу на языке MATLAB

1 шаг решения: выбираем, какую подпрограмму (в MATHCAD программу-функцию) надо разработать: подпрограмма должна вызываться несколько раз с разными данными.

Для нашей задачи: функция определяет, упорядочены ли по возрастанию q-го столбца произвольной матрицы X.
Вход функции: матрица X, номер столбца q.
Выход функции : F – признак упорядоченности элементов q-го столбца:

$$F = \begin{cases} 1, \text{если элементы q - го столбца упорядочены;} \\ 0, \text{если нет.} \end{cases}$$

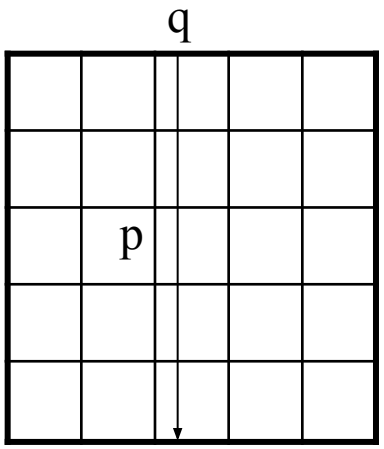
2 шаг - Пишем заголовок функции:

upor_stolb(X,q):=

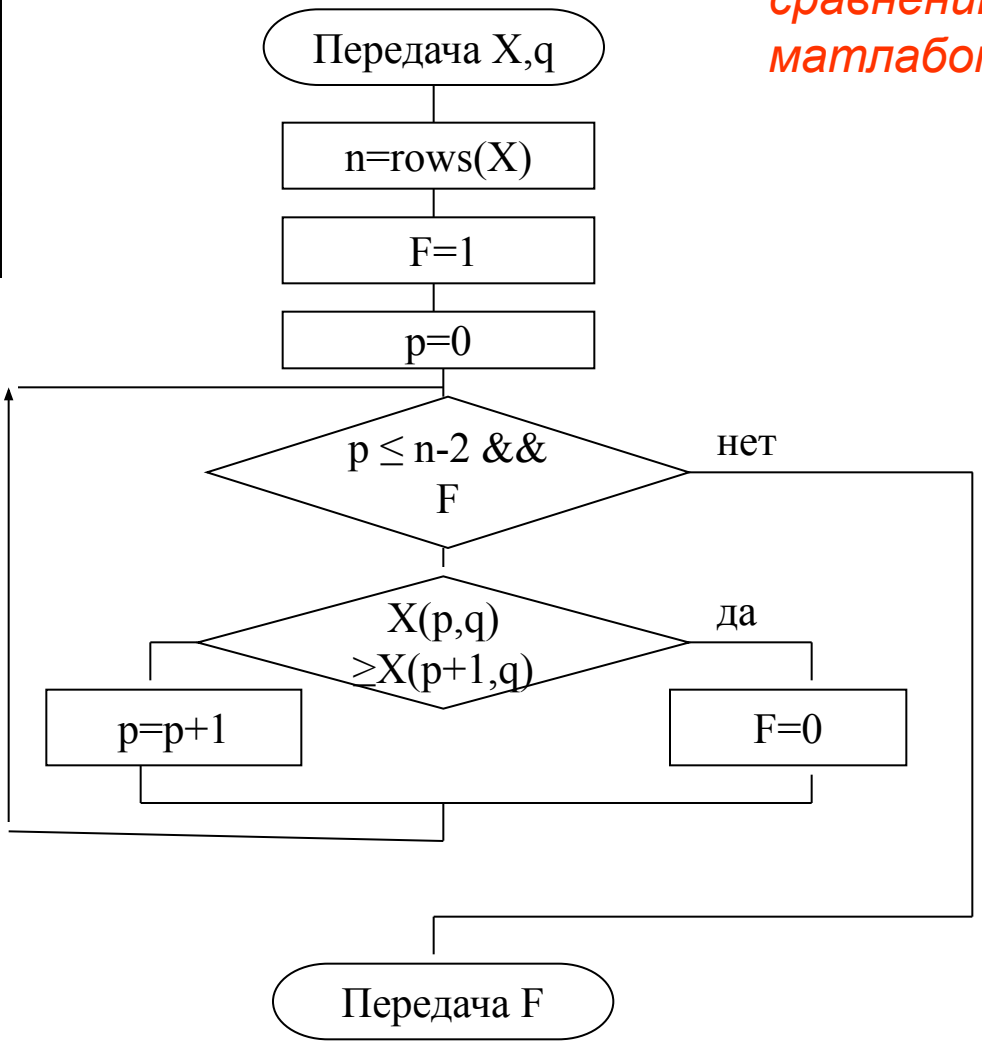
3 шаг. Состав данных функции upor_stolb

Имя	Смысл	Тип	Структура
Входные данные (передаваемые из вызывающего кода в функцию)			
X	Заданная матрица	вещественный	двумерный массив
q	Номер столбца	целый	простая переменная
Выходные данные (передаваемые из функции вызывающий код)			
F	Признак упорядоченности столбца	целый (логический)	простая переменная
Промежуточные данные			
n	Число строк матрицы	целый	простая переменная
m	Число столбцов матрицы	целый	простая переменная
p	Счетчик строк матрицы	целый	простая переменная

4 шаг. Блок-схема алгоритма функции



Что исправили по сравнению с матлабом?



5 шаг. Код функции upor_stolb

Upor_stolb(X,q):=

n ← rows(X)

F←1

p←0

while (p≤n-2) ^ F

F←0 if $X_{p,q} > X_{p+1,q}$

p←p+1 otherwise

F

Вызов функции – самостоятельно!

Следующая лекция: локальные и глобальные переменные