

Понятие и свойства алгоритма

Понятие алгоритма

- Алгоритм – последовательность действий (план) для достижения цели за конечное число шагов
- Алгоритмизация – процесс разработки алгоритма для решения задачи

Свойства алгоритмов

- **Дискретность** – алгоритм состоит из набора отдельных законченных действий (шагов)
- **Детерминированность** / определенность - любое действие алгоритма должно быть строго и недвусмысленно определено в каждом случае
- **Универсальность** / массовость - один и тот же алгоритм можно использовать с разными исходными данными
- **Результативность** / конечность - при точном исполнении всех предписаний алгоритма процесс должен прекратиться за конечное число шагов с получением определенного ответа на вопрос задачи (либо вывода о том, что решения не существует)

Основные этапы технологического процесса решения задач с помощью ЭВМ

- 1 этап:** Постановка задачи и выбор метода решения (формальное математическое описание алгоритма)
- 2 этап:** Определение и описание входных, промежуточных и выходных данных, необходимых для решения задач.
- 3 этап:** Разработка алгоритма решения задач.
- 4 этап:** Кодирование описания данных и алгоритма (составление программы на выбранном языке программирования).
- 5 этап:** Отладка и тестирование программы с целью её проверки и доведения её в соответствии с поставленной задачей.
- 6 этап:** Выполнение и поддержка программы (создание новых версий в зависимости от новой техники).

Основные правила оформления схемы алгоритма

ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных, систем»

1. Символ предназначен для графической идентификации функции, которую он отображает, независимо от текста внутри этого символа.
2. Стандартное направление линий - слева направо и сверху вниз. Пересечение и ветвление линий не допускается.
3. Стрелки на прямых линиях можно не ставить. Если направление отличается от стандартного, то стрелки обязательны.
4. Линии должны подходить к символу слева, либо сверху, а исходить справа, либо снизу.
5. Допускается один вход и один выход из символа – кроме символов Модификация и Решение.

Обозначения элементов схемы алгоритма

Терминатор



Начало и конец алгоритма

Данные



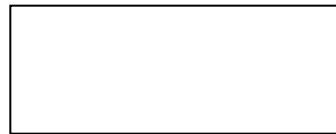
Ввод с клавиатуры и **вывод** на экран

Документ



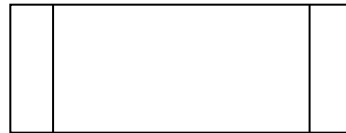
Вывод результатов на принтер

Процесс



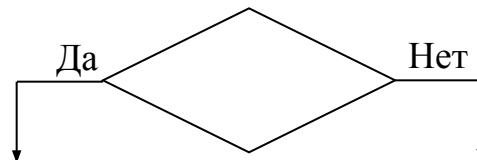
Вычислительные действия (одно или несколько действий, изменяющих решение или форму представления данных. Несколько блоков «процесс» можно объединить в один)

Предопределенный процесс



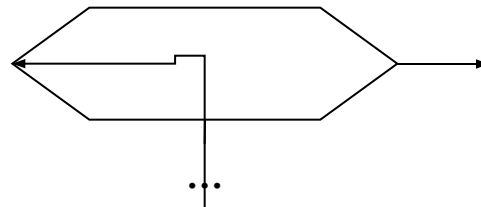
Подпрограмма, модуль или стандартная программа

Решение



Проверка условий (переход управления по условию, записанному внутри блока)

Модификация



Организация цикла (внутри блока записан параметр цикла, его начальное и конечное значение, шаг изменения)

Виды алгоритмов

```
graph TD; A[Виды алгоритмов] --> B[Алгоритмы линейной структуры]; A --> C[Алгоритмы ветвящейся структуры]; A --> D[Алгоритмы циклической структуры];
```

Алгоритмы линейной структуры

(линейные алгоритмы)

- последовательность действий, выполняющихся друг за другом в строгом порядке

Алгоритмы ветвящейся структуры

(ветвящиеся алгоритмы) -

последовательность действий с наличием хотя бы двух альтернативных путей выполнения алгоритма

Алгоритмы циклической структуры

(циклические алгоритмы) –

имеется многократное повторение некоторой группы действий

Пример линейного алгоритма: Вычисление площади круга

1 этап: постановка задачи $S = \pi R^2$

2 способ - схема алгоритма:

2 этап: входные и выходные данные

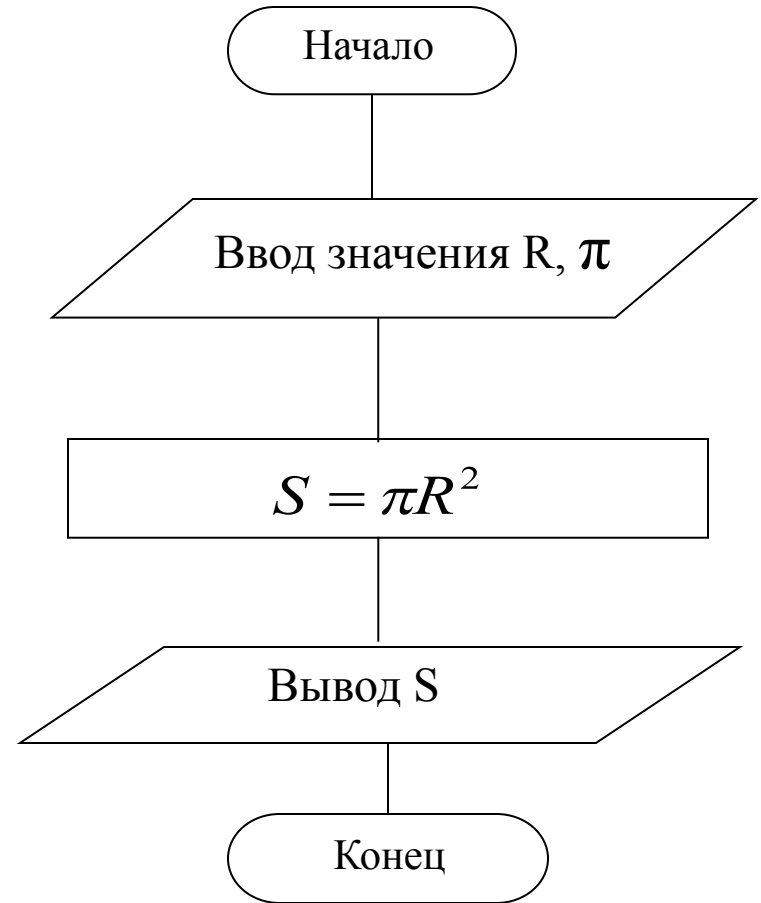
Входные данные: R, π

Выходные данные: S

3 этап: разработка алгоритма:

1 способ - словесное описание:

- 1) задать значения R, π
- 2) вычислить S по формуле
- 3) вывести значение S на экран



Типы ветвящихся алгоритмов

1. два альтернативных варианта
2. многовариантный выбор

Пример 1 ветвящегося алгоритма: Решение квадратного уравнения

1 этап: постановка задачи
 $ax^2+bx+c=0$

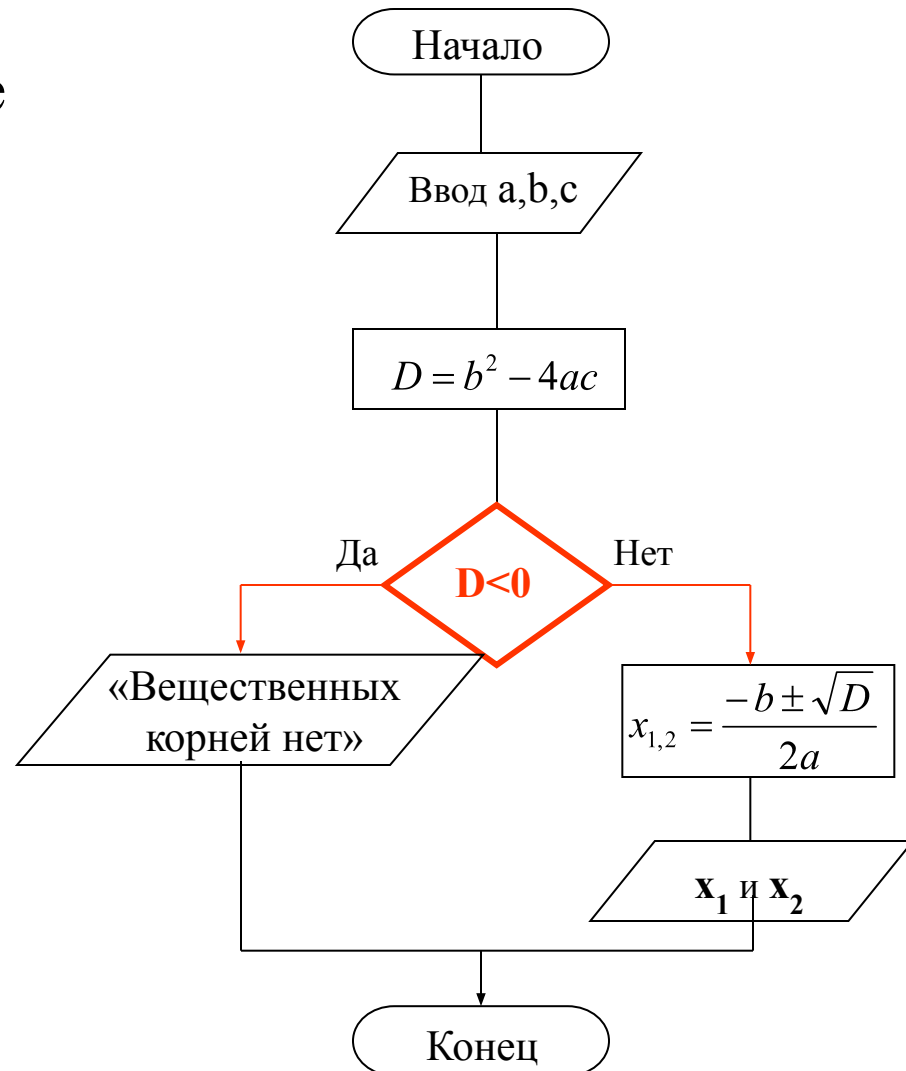
2 этап: входные и выходные данные
a, b, c - входные данные
D - промежуточные данные
x1, x2 – выходные данные

3 этап: разработка алгоритма:

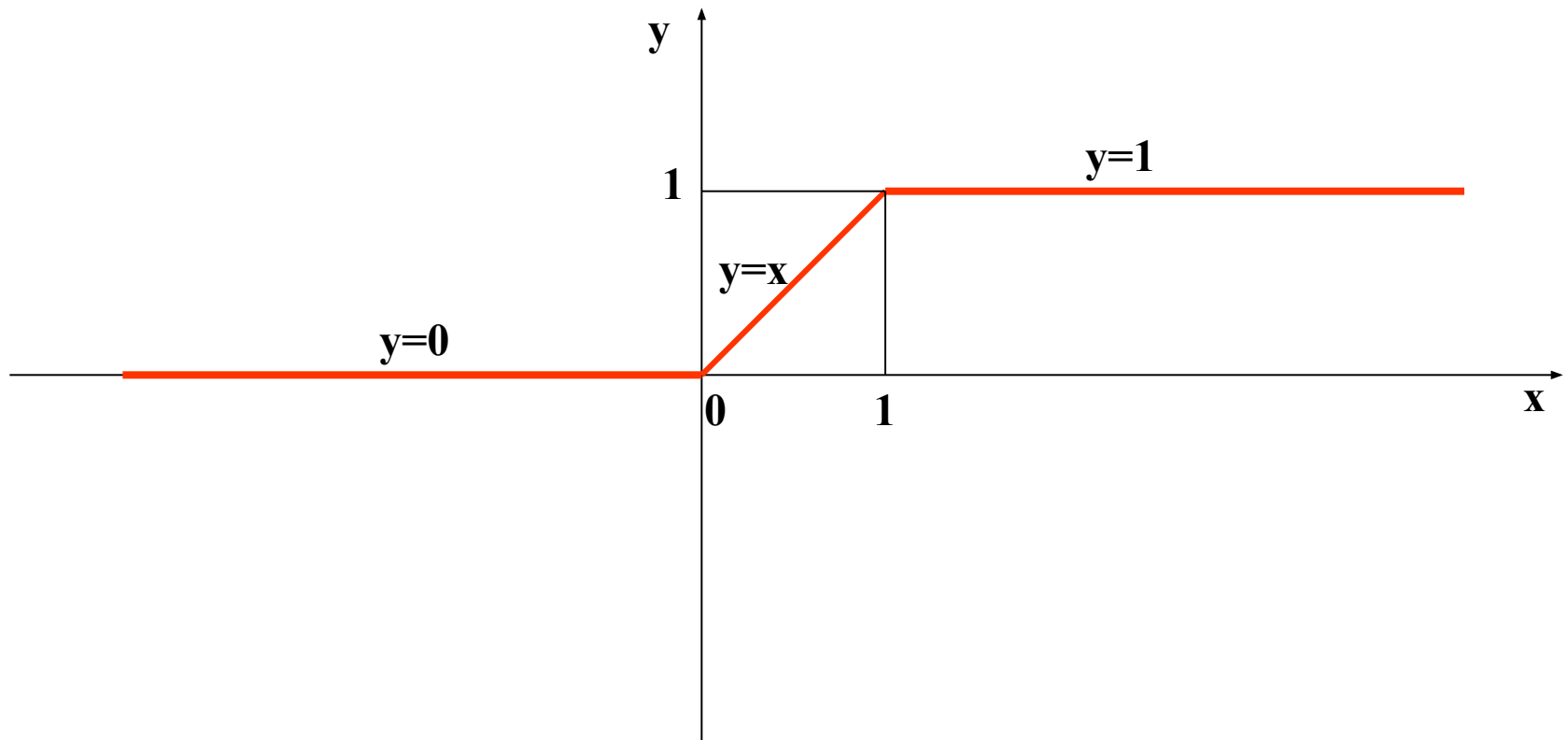
1 способ - словесное описание:

- 1) задать значения a, b, c
- 2) вычислить D по формуле
- 3) сравнить D с нулем:
 - а) если $D < 0$, вывести текст «Вещественных корней нет»
 - б) если $D \geq 0$, вычислить x_1, x_2 и вывести их значения на экран

2 способ - схема алгоритма:



Пример 2 ветвящегося алгоритма: Вычисление функции, заданной графически



1 этап: постановка задачи

$$\begin{cases} y=0, & x < 0 \\ y=x, & 0 \leq x \leq 1 \\ y=1, & x > 1 \end{cases}$$

2 этап: входные и выходные данные

x - входные данные

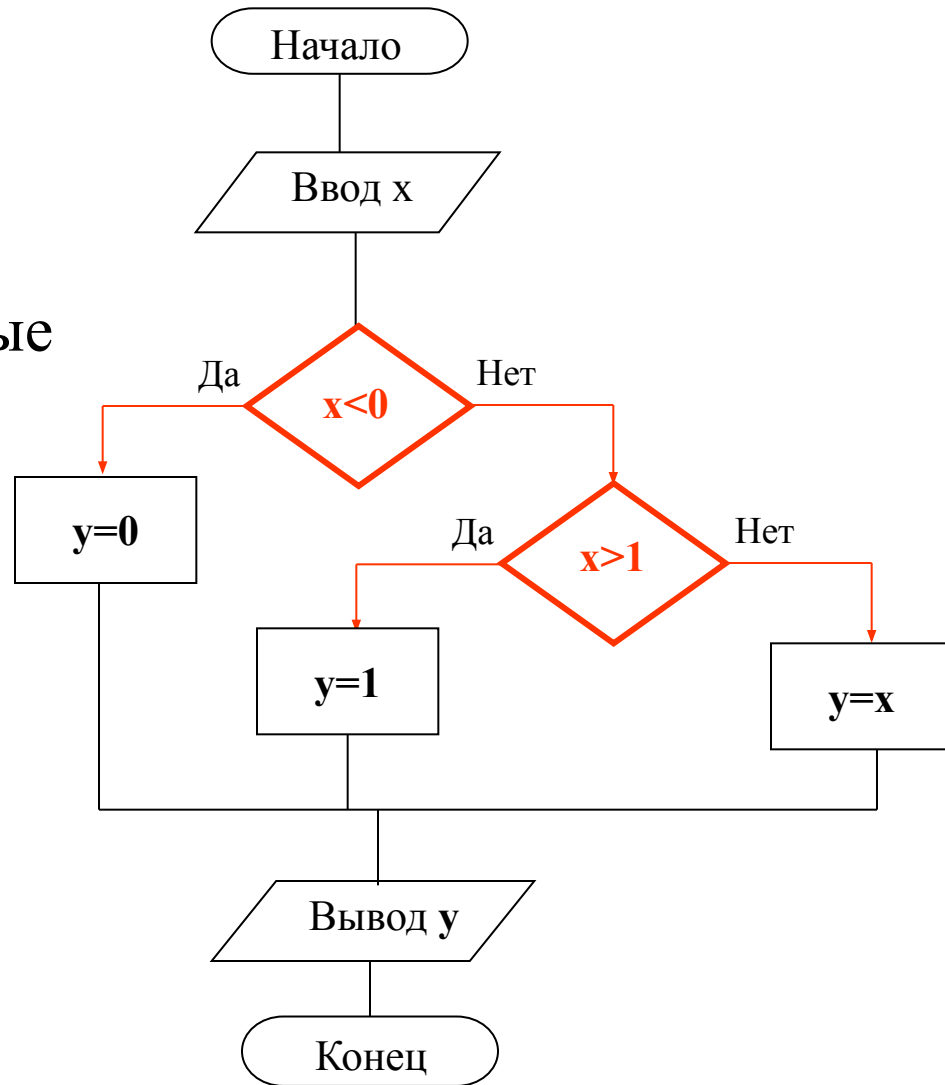
y - выходные данные

3 этап: разработка алгоритма:

1 способ - словесное описание:

- 1) задать значение **x**
- 2) сравнить **x** с нулем:
 - а) если $x < 0$, вычислить $y=0$
 - б) если нет - сравнить **x** с единицей:
 - если $x > 1$, вычислить $y=1$
 - если нет - вычислить $y=x$
- 3) вывести значение **y** на экран

2 способ - схема алгоритма:



Пример 3 ветвящегося алгоритма: Организация меню (многовариантный выбор)

Составить схему алгоритма следующей задачи:

1. На экран выводится окно с **текстом** меню:

1. **Вычисление площади круга**
2. **Вычисление площади квадрата**
3. **Вычисление площади трапеции**
4. **Выход**

Введите номер меню: _

2. С клавиатуры вводится **номер** меню.

3. В зависимости от введенного номера выполняется один из 4-х пунктов, причем после выполнения 1,2 или 3-го пунктов происходит возврат в окно меню.

1 этап: постановка задачи -
формулы площадей:

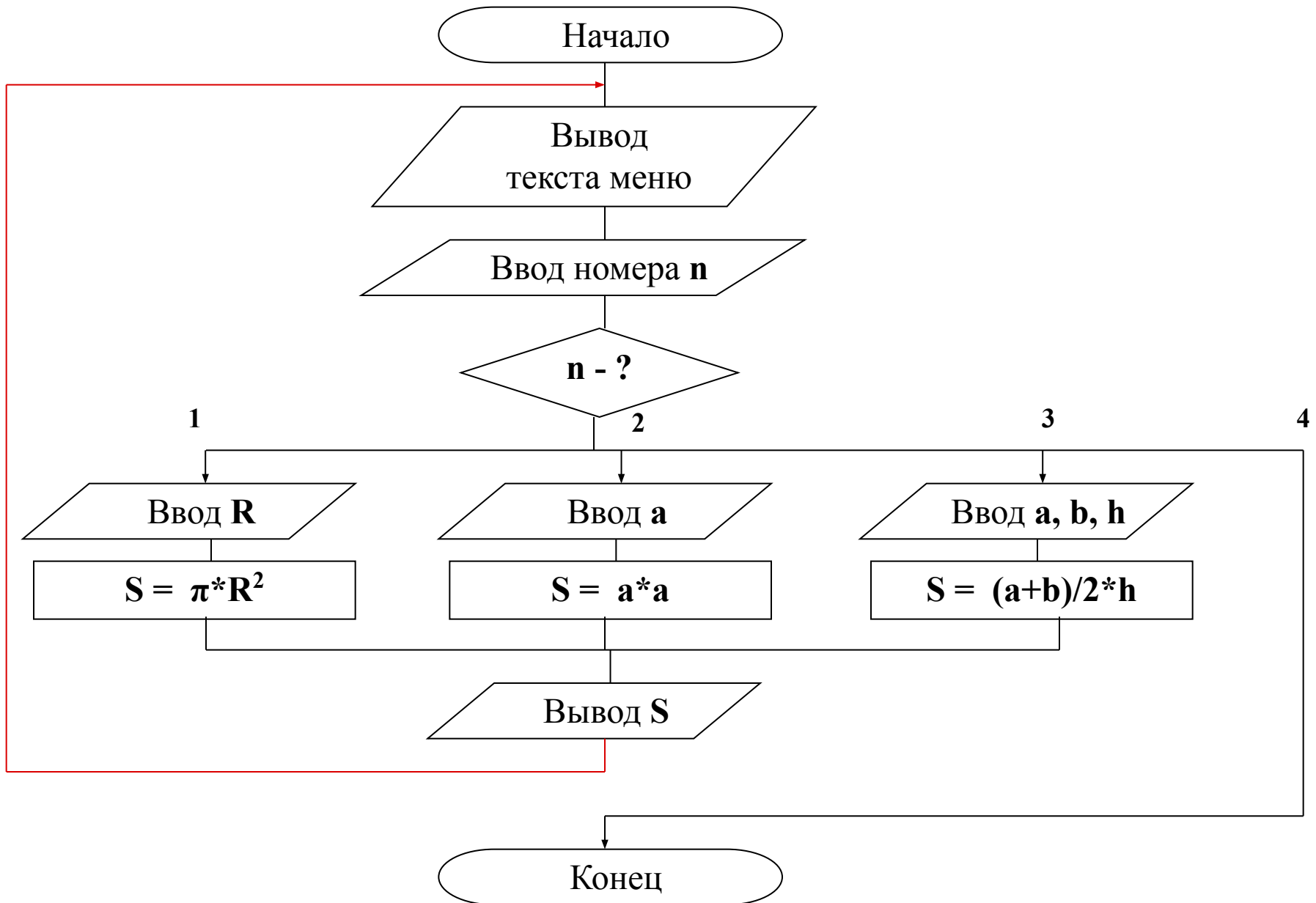
$$S = \pi * R^2$$

$$S = a * a$$

$$S = \frac{a + b}{2} h$$

2 этап: ВХОДНЫЕ данные - **R, a, b, h, n**
ВЫХОДНЫЕ данные - **S**

3 этап: схема алгоритма работы меню



Циклические алгоритмы

Организационно цикл состоит из следующих компонентов:

- подготовка цикла
- тело цикла
- условие продолжения цикла

Основные понятия:

- параметр цикла
- начальное и конечное значение параметра цикла
- шаг цикла

Пример 1 циклического алгоритма: Вычисление таблицы значений функции

1 этап Постановка задачи: Вычислить и вывести на экран таблицу значений X и Z , где $Z=X^2$ для $0,2 \leq X \leq 1,1$ с шагом $0,3$

2 этап Входные и выходные данные:

X - входные данные (1 ячейка ОЗУ)

Z - выходные данные (1 ячейка ОЗУ)

3 этап Разработка алгоритма

1 способ - словесное описание:

Вариант 1

- 1) $X = 0.2$
- 2) вычислить $Z = X^2$
- 3) вывести X и Z
- 4) изменить $X = X + 0.3$
- 5) если $X \leq 1.1$, то перейти к 2)
иначе перейти к п. 6)
- 6) конец

Вариант 2

- 1) $X = 0.2$
- 2) если $X \leq 1.1$, то перейти к п. 3)
иначе перейти к п. 7)
- 3) вычислить $Z = X^2$
- 4) вывести X и Z
- 5) изменить $X = X + 0.3$
- 6) перейти к п. 2)
- 7) конец

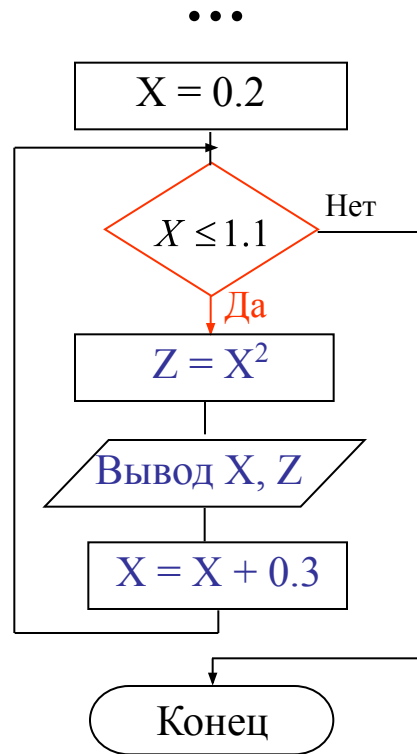
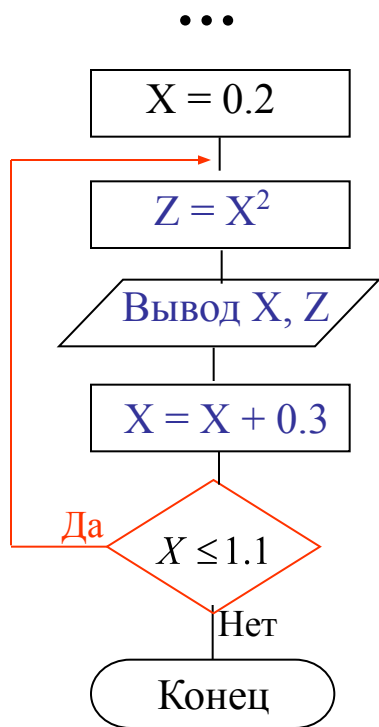
3 типа циклических алгоритмов:

1. Цикл с **постусловием**
2. Цикл с **предусловием**
3. Цикл со **счетчиком**

Схемы циклических алгоритмов 3-х типов

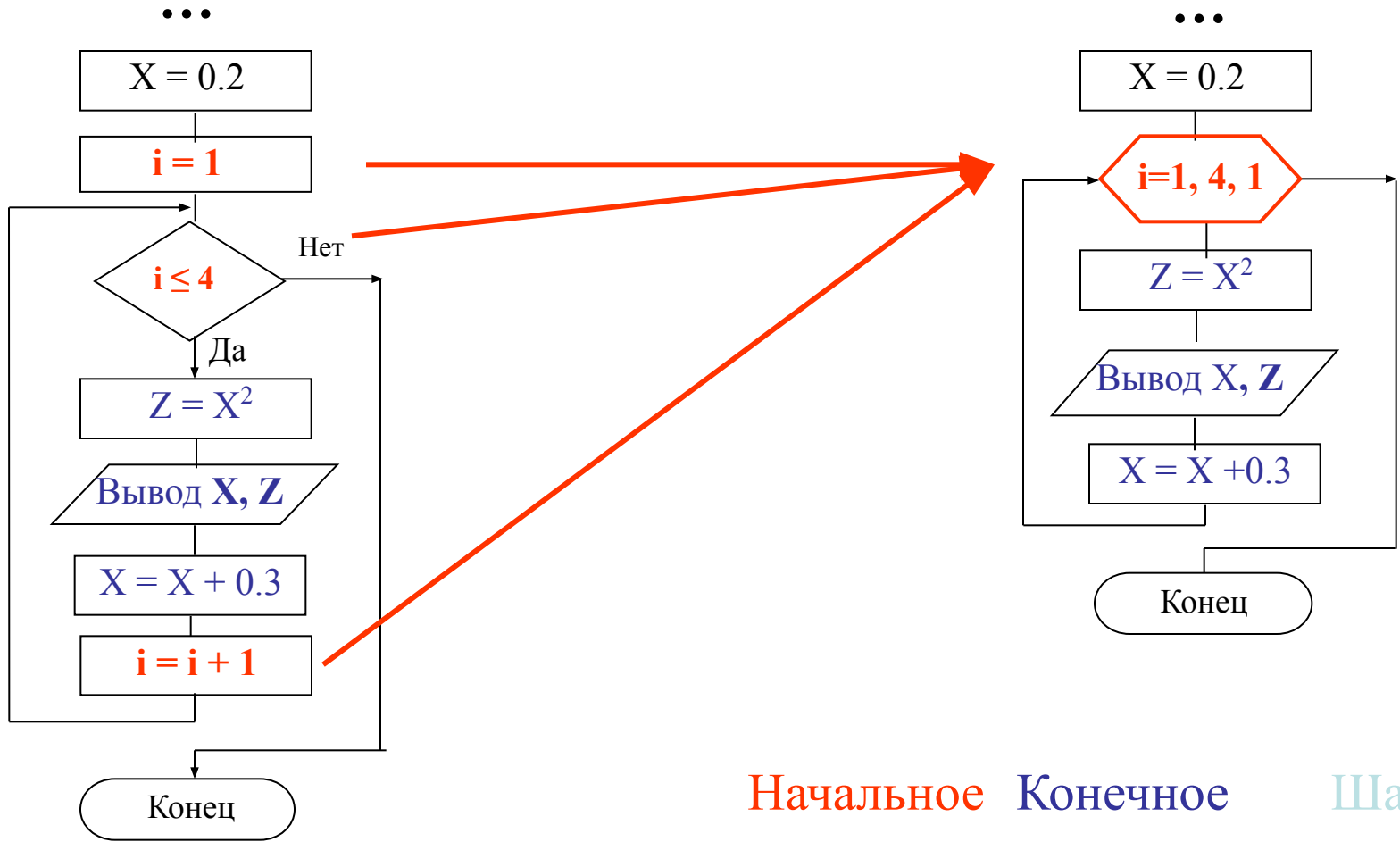
1-й тип: Цикл с постусловием

2-й тип: Цикл с предусловием

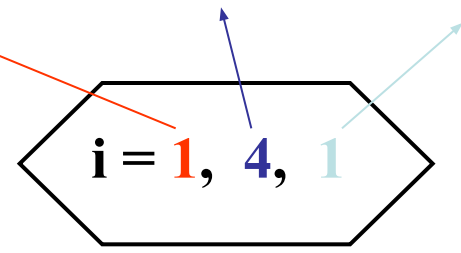


Здесь X – параметр цикла

3-й тип: Цикл со счетчиком



Начальное значение Конечное значение Шаг изменения



Здесь i – параметр цикла

Пример 2 циклического алгоритма (без организации массива)

1 этап: постановка задачи

С клавиатуры вводится последовательность чисел.
Найти сумму чисел до первого введенного «0».

2 этап: входные и выходные данные

A - входные данные (одна ячейка ОЗУ для ввода очередного числа)
S - выходные данные (одна ячейка ОЗУ для накопления суммы)

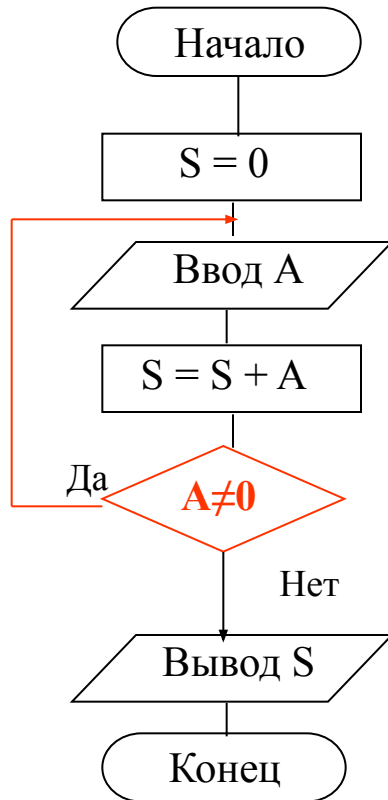
3 этап: разработка алгоритма:

- 1) начальное значение $S = 0$
- 2) в циклической части происходит накопление S:
рекуррентная формула суммы: $S = S + A$

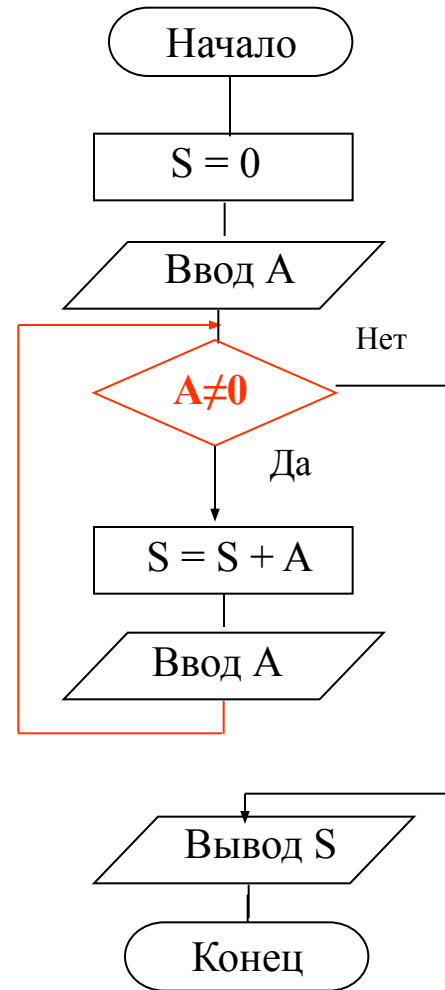
Замечание. Здесь заранее неизвестно, сколько раз выполнится цикл, поэтому можно применить **цикл с постусловием** или **цикл с предусловием**.

2 варианта схемы алгоритма

Цикл с постусловием



Цикл с предусловием



Замечание: для другой задачи (например, произведение чисел до первого отрицательного) аналогичный **цикл с постусловием** даст **неверный ответ**.

Массивы

**Типовые алгоритмы по
обработке одномерных
массивов**

Понятие массива

Массив – это конечная, упорядоченная последовательность элементов одного типа, имеющая общее имя.

1. Элементы массива пронумерованы; обратиться к каждому элементу можно, указав его **индекс** (номер)
2. Количество индексов может быть разным (1, 2, 3,...), в зависимости от этого массивы могут быть **одномерные** и **многомерные**

Размер и размерность массива

- **Размер** – это количество элементов массива
- **Размерность** (ранг) определяется числом индексов элемента массива (одномерные, двумерные,...)

Обозначения 1-мерных массивов

1. Словесное описание:

Например, массив **A** из 10 целых чисел;

массив **R** из 15 вещественных чисел

2. Математическое описание:

- массив **A**: $\{ A_i \}_{i=1,10}$ или **A(10)**

- элементы массива **A**: $A_1 A_2 \dots A_i \dots A_{10}$

- массив **R**: $\{ R_i \}_{i=1,15}$ или **R(15)**

- элементы массива **R**: $R_1 R_2 \dots R_i \dots R_{15}$

- Замечание.**
1. **Размерность** массивов **A** и **R** одинакова и равна 1.
 2. **Размер** массивов разный (10 и 15 элементов)

Стандартные алгоритмы для обработки одномерных массивов

1. Нахождение **суммы** всех элементов массива (или части элементов)
2. Нахождение **произведения** всех элементов массива (или части элементов)
3. Подсчет **количества** положительных и отрицательных элементов в массиве
4. Нахождение **максимального** и **минимального** элемента в массиве
5. **Сортировка** массива (расположение элементов в порядке возрастания или убывания их значений)

Пример1: Сумма всех элементов массива А(10)

I этап:
$$S = \sum_{i=1}^{10} A_i$$

II этап: входные данные – массив А,
промежуточные данные - i
выходные данные – S

III этап: разработка алгоритма:

$$S = 0$$

$$S = S + A_1 = 0 + A_1 = A_1$$

$$S = S + A_2 = A_1 + A_2$$

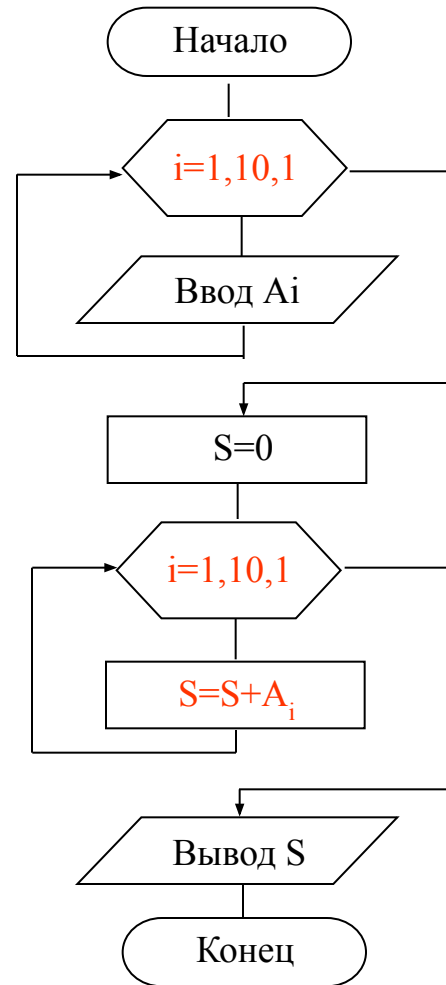
$$S = S + A_3 = (A_1 + A_2) + A_3$$

...

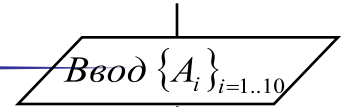
$$S = S + A_{10}$$

Рекуррентная формула суммы:

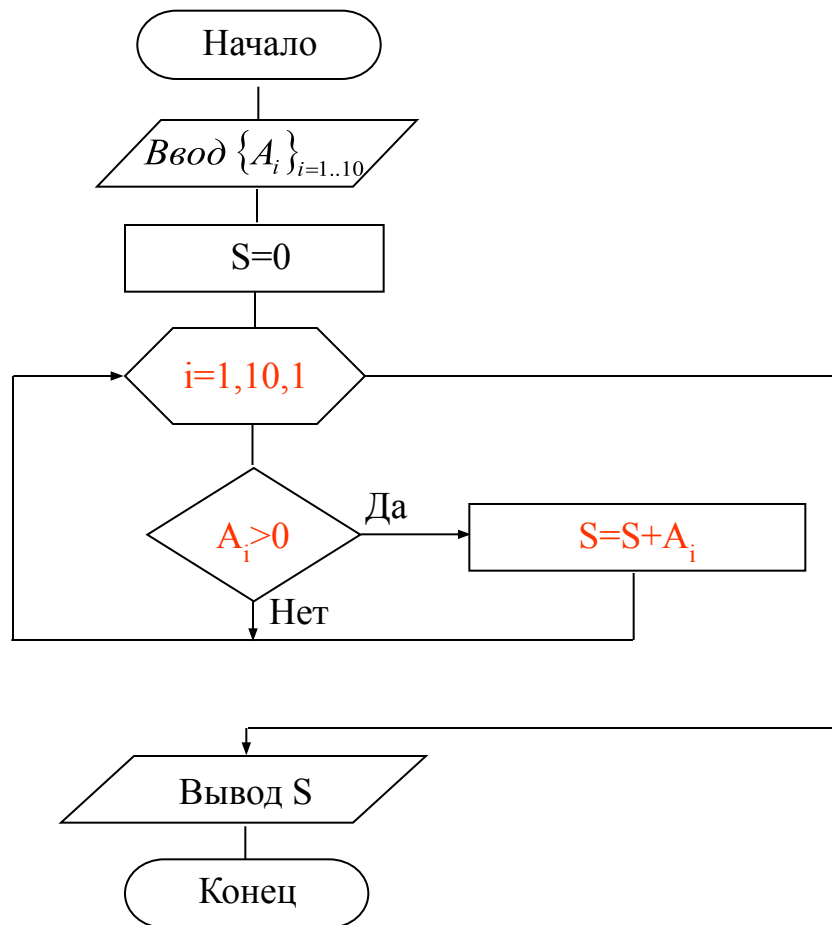
$$S = S + A_i$$



Краткая запись
(только для лекций)



Пример 2: Сумма положительных элементов массива $A(10)$



Пример 3: Произведение всех элементов массива A(10)

I этап: $PR = \prod_{i=1}^{10} A_i$

II этап: входные данные - массив A
промежуточные данные - i
выходные данные - PR

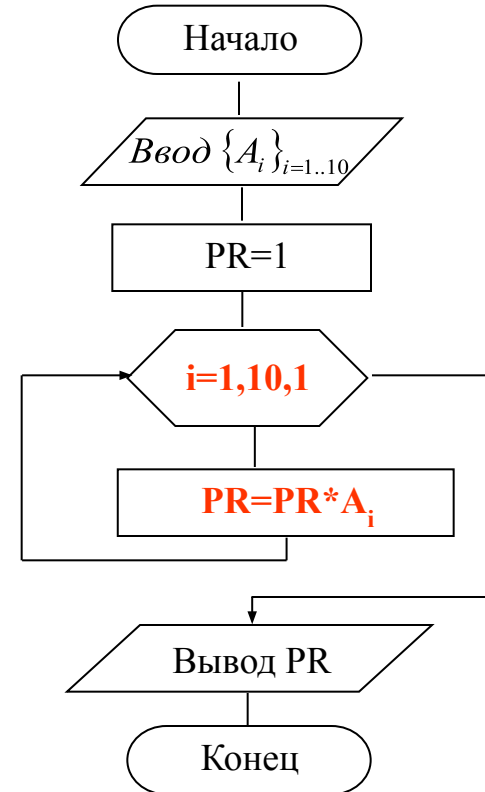
III этап: разработка схемы

PR=1

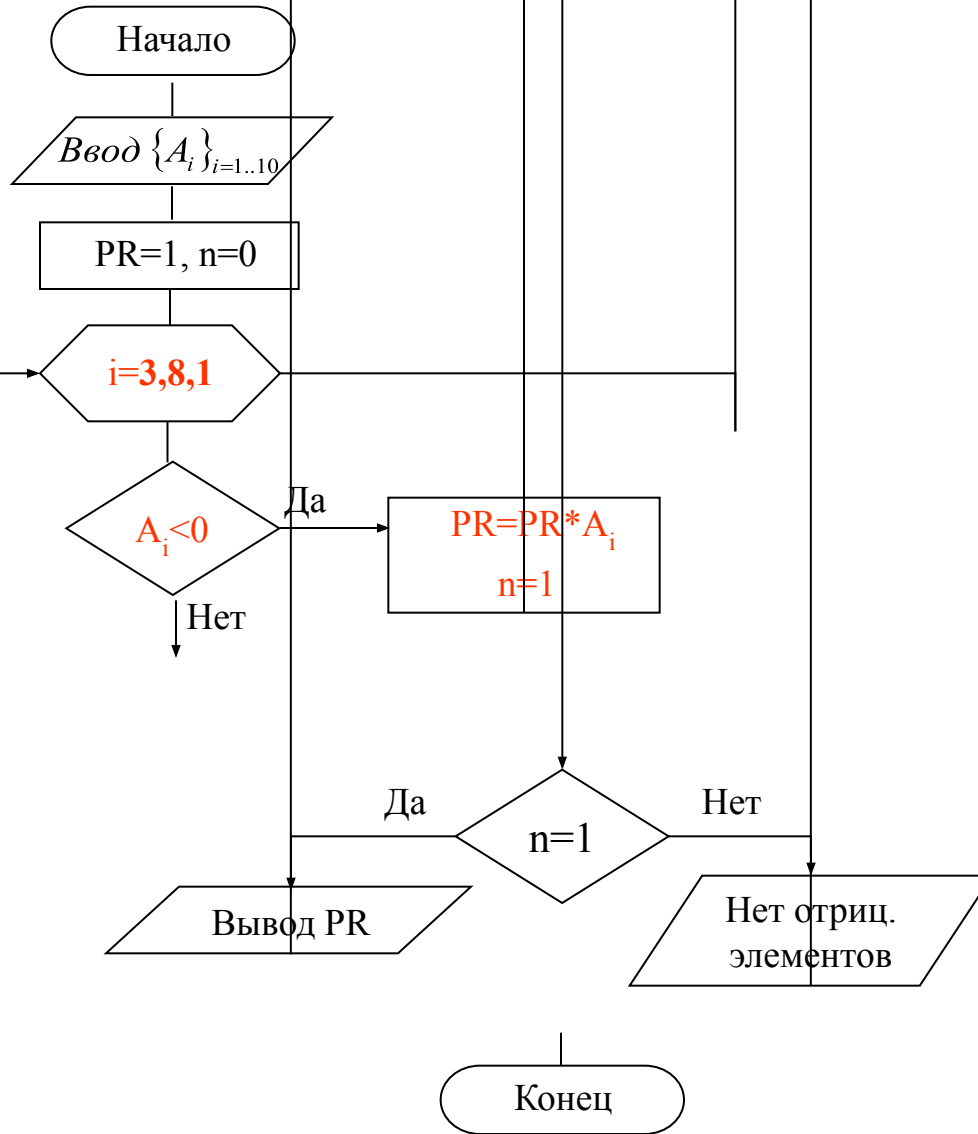
рекуррентная формула произведения:

$$PR = PR * A_i$$

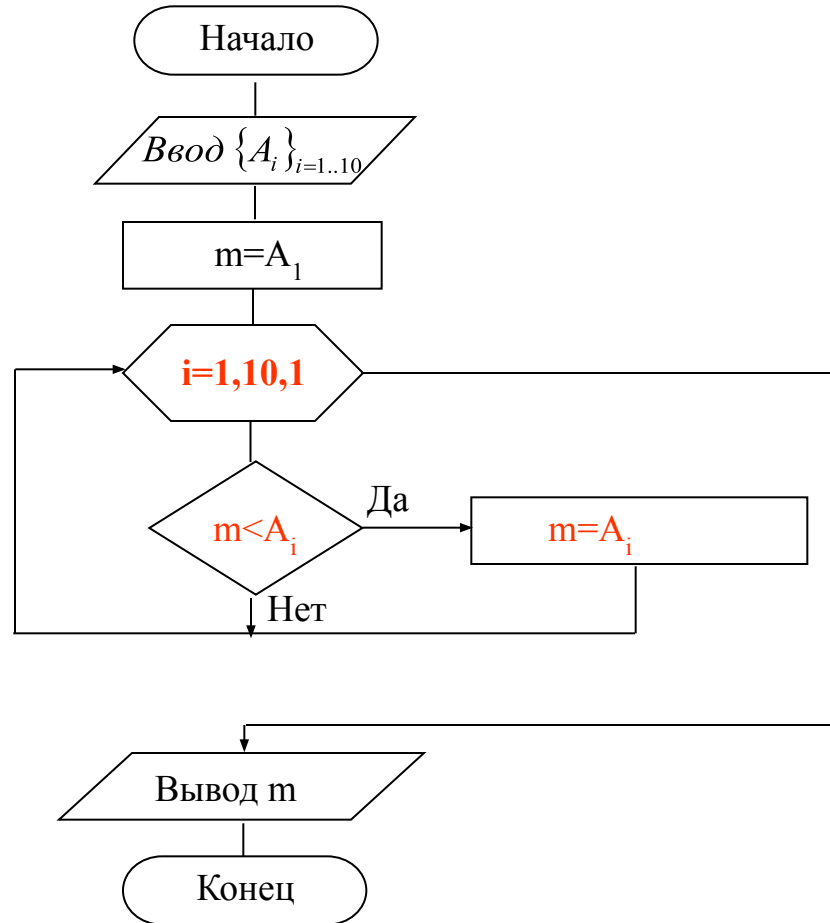
Схема алгоритма



Пример 4: Произведение отрицательных элементов массива А (с 3-го по 8-й)



Пример 5: Максимальный элемент массива А

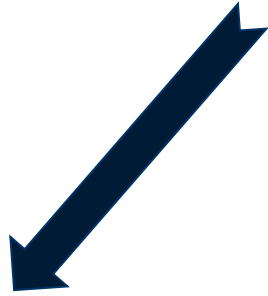


Примечание:

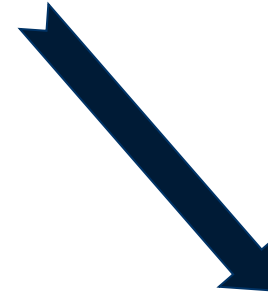
Можно ввести следующие дополнительные проверки:

- единственность максимума
- все элементы равны

Сортировка элементов массива



Алгоритмы выбора



Алгоритмы обмена

Алгоритм выбора

Исходный массив: **5 4 1 3 2**

n=5

1 этап - находим макс. элемент и меняем местами с последним элементом:

2 4 1 3 5

2 этап – (рассматриваем массив на единицу короче **n=4**) находим макс. элемент и меняем местами с последним элементом в укороченном массиве:

2 3 1 4 5

3 этап – (рассматриваем массив на единицу короче **n=3**) находим макс. элемент и меняем местами с последним элементом в укороченном массиве

2 1 3 4 5

4 этап – (рассматриваем массив на единицу короче **n=2**) находим макс. элемент и меняем местами с последним элементом в укороченном массиве

1 2 3 4 5

5 этап – (рассматриваем массив на единицу короче **n=1**) находим макс. элемент и меняем местами с самим собой

1 2 3 4 5

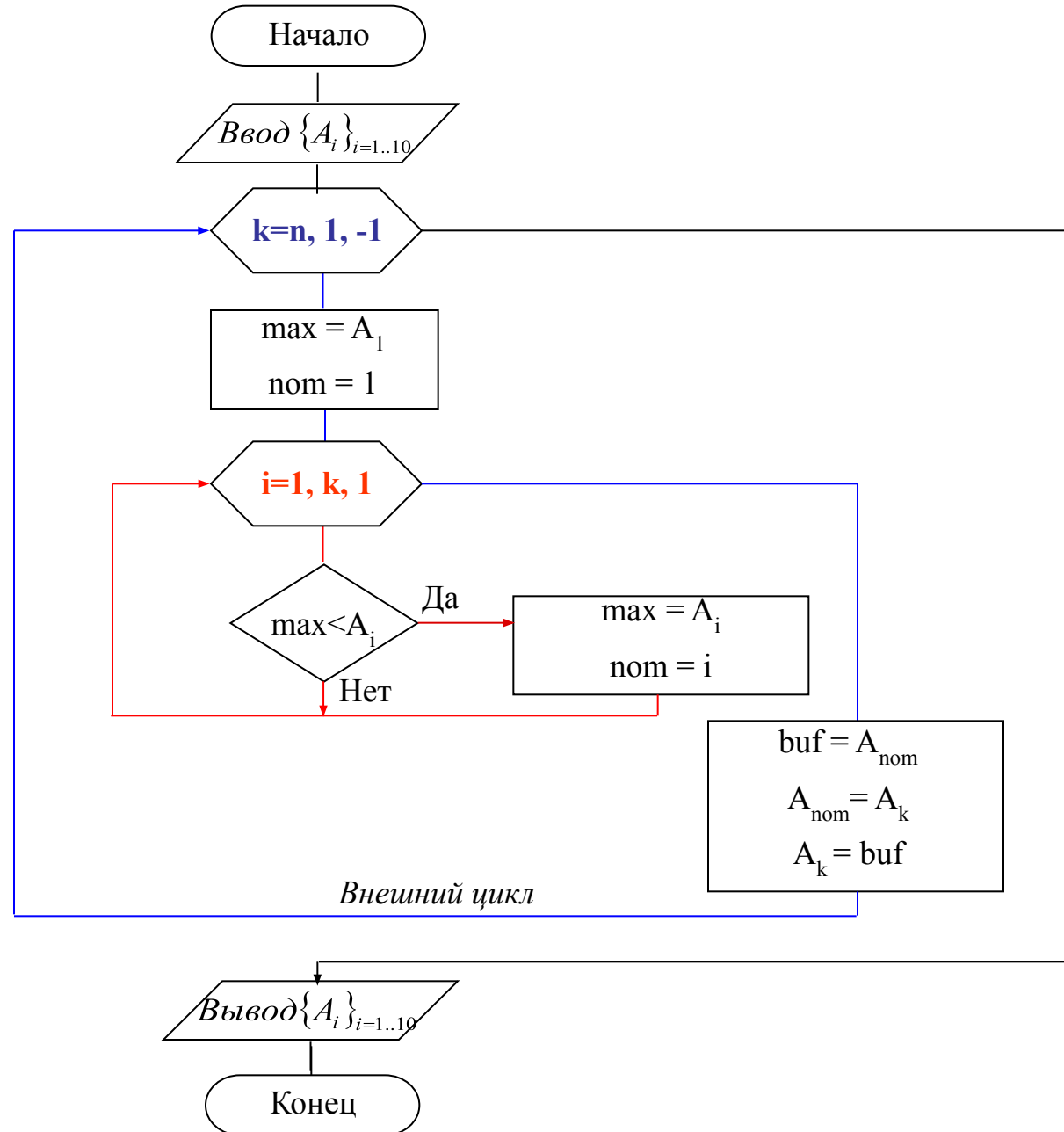
Вывод: 1) на каждом этапе повторяются одни и те же действия

2) этапы отличаются только количеством элементов массива

3) число этапов равно размеру массива

Пример сложного цикла, где во внутреннем цикле ищется максимальный элемент текущего фрагмента, а во внешнем цикле определяется количество этапов $(n-1)$ и происходит обмен максимального элемента и последнего элемента текущего фрагмента массива

Схема алгоритма метода выбора



Алгоритм обмена

Исходный массив: **5 4 1 3 2**

n=5

1 этап – сравниваем два соседних элемента: если предыдущий больше последующего, то меняем их местами:

4 5 1 3 2 → **4 1 5 3 2** → **4 1 3 5 2** → **4 1 3 2 5**

2 этап – (рассматриваем массив на единицу короче **n=4**) сравниваем два соседних элемента: если предыдущий больше последующего, то меняем их местами:

1 3 2 4 5

3 этап – (рассматриваем массив на единицу короче **n=3**) сравниваем два соседних элемента: если предыдущий больше последующего, то меняем их местами:

1 2 3 4 5

4 этап – (рассматриваем массив на единицу короче **n=2**) сравниваем два соседних элемента: если предыдущий больше последующего, то меняем их местами:

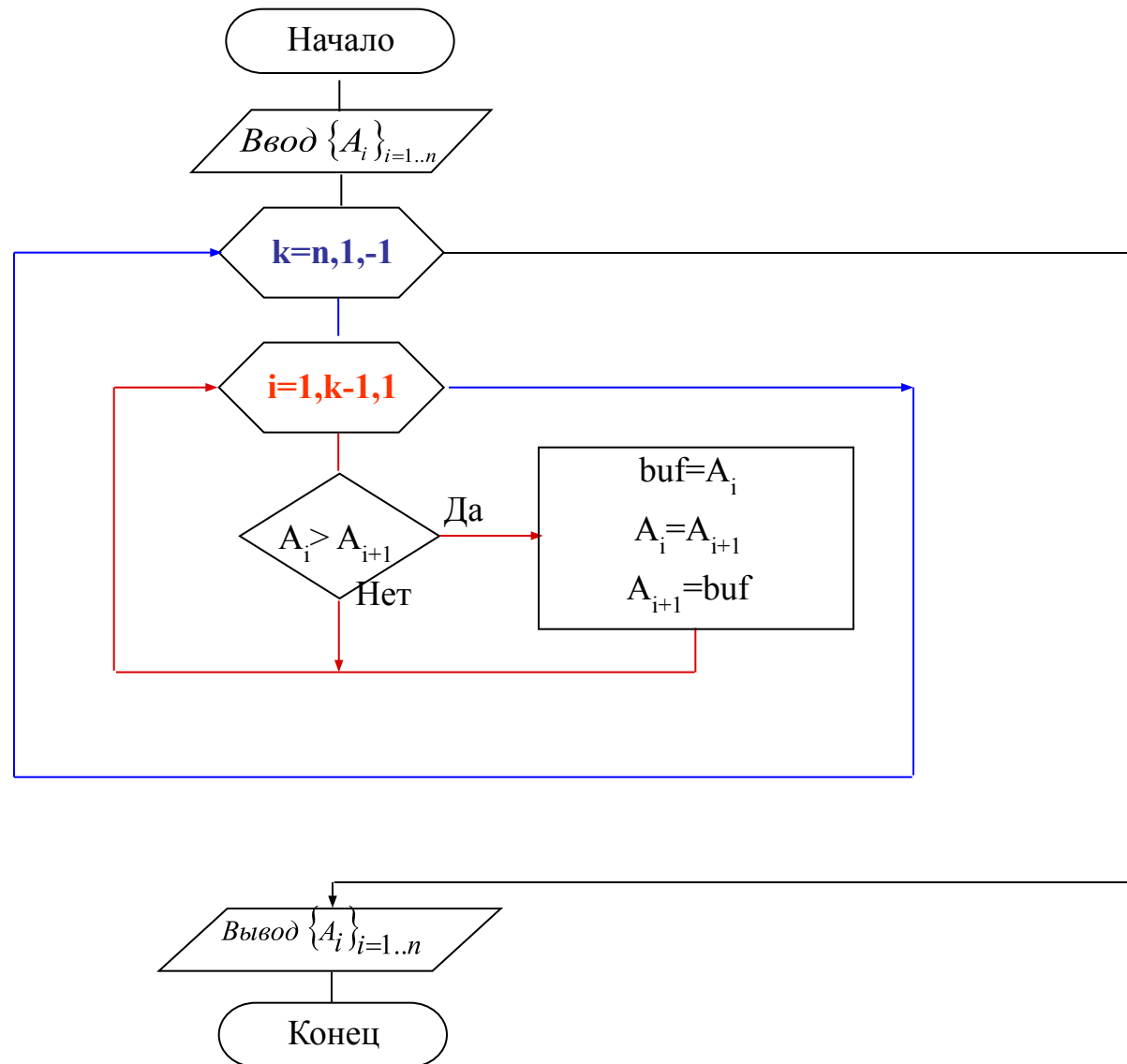
1 2 3 4 5

5 этап – (рассматриваем массив на единицу короче **n=1**) сравнения нет

1 2 3 4 5

Вывод: 1) на каждом этапе повторяются одни и те же действия
2) этапы отличаются только количеством элементов массива
3) число этапов равно размеру массива

Схема алгоритма метода обмена



Двумерные массивы (матрицы)

Типовые алгоритмы
обработки матриц

Обозначения 2-мерных массивов

1. Словесное описание:

Например, матрица $A(n,m)$ целых чисел

2. Математическое описание:

- матрица A : $\{ A_{ij} \}_{i=1,n \ j=1,m}$ или $A(n \times m)$

- элементы матрицы A :

$$\begin{matrix} A_{21} & A_{22} & A_{23} & \dots & A_{2m} \\ A_{31} & A_{32} & A_{33} & \dots & A_{3m} \end{matrix}$$

$$A_{n1} \ A_{n2} \ A_{n3} \ \dots \ A_{nm}$$

$$\left[\begin{matrix} A_{11} & A_{12} & A_{13} & \dots & A_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{matrix} \right]$$

Замечание. 1. **Размерность** массива A равна **2**.

2. **Размер** массива A - **$n \times m$** элементов

Пример обозначения 2-мерных массивов

1. Словесное описание:

Например, матрица $A(3,4)$ целых чисел

2. Математическое описание:

- матрица A : $\{ A_{ij} \}_{i=1,3 j=1,4}$ или $A(3 \times 4)$

- элементы матрицы A :

	A_{11}	A_{12}	A_{13}	A_{14}
A_{21}	A_{22}	A_{23}	A_{24}	
A_{31}	A_{32}	A_{33}	A_{34}	

Замечание. 1. **Размерность** массива A равна **2**.

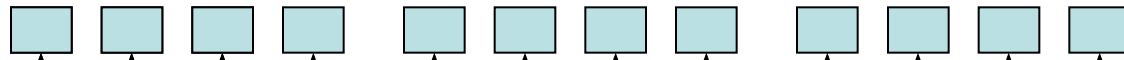
2. **Размер** массива A - **12** элементов

Распределение ОЗУ (матрицы)

Матрица $A(3 \times 4)$:

Номера ячеек:

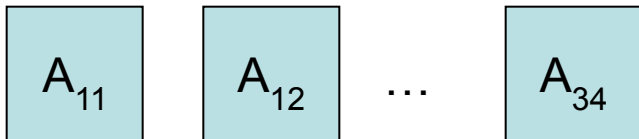
1 2 3 4 5 6 7 8 9 10 11 12



Имена ячеек:

A_{11} A_{12} A_{13} A_{14} A_{21} A_{22} A_{23} A_{24} A_{31} A_{32} A_{33} A_{34}

Краткая запись:

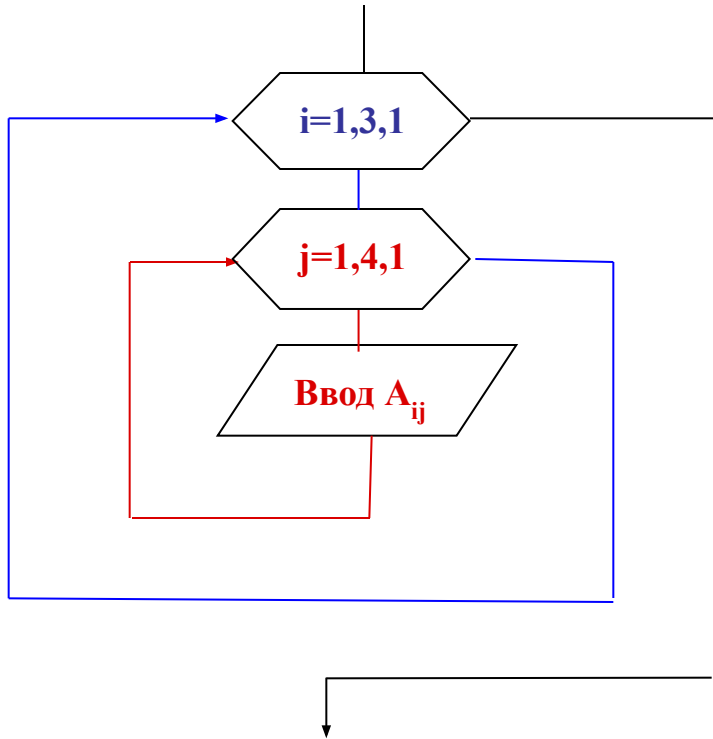


Стандартные алгоритмы для обработки матриц

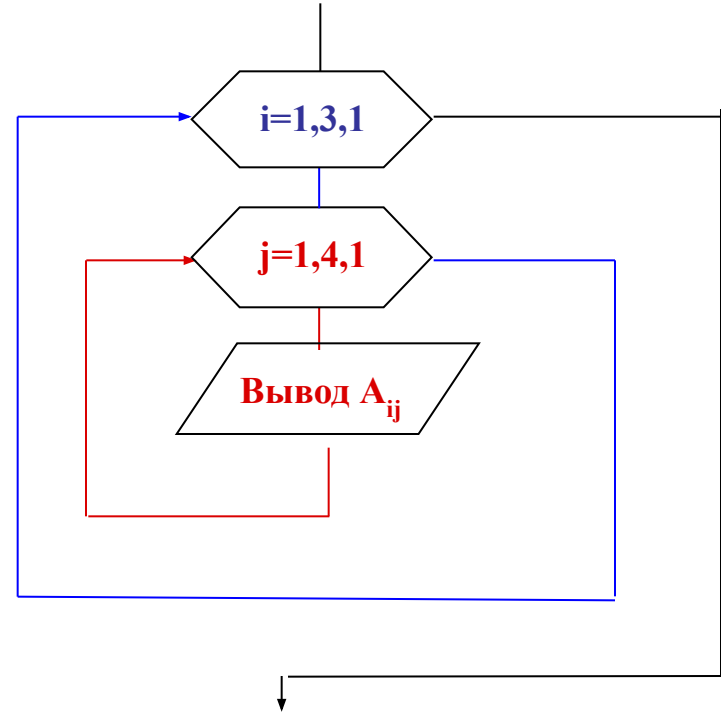
1. Вычисление **суммы** элементов матрицы
2. Вычисление **произведения** элементов матрицы
3. Подсчет **количества** элементов, удовлетворяющих условию
4. Нахождение **максимального и минимального** элемента в матрице
5. Вычисление суммы (произведения, максимального, минимального и т.п.) в каждой строке или каждом столбце или в **произвольной области** матрицы
6. Формирование **одномерного массива** из элементов матрицы

Ввод и вывод элементов матрицы A(3x4)

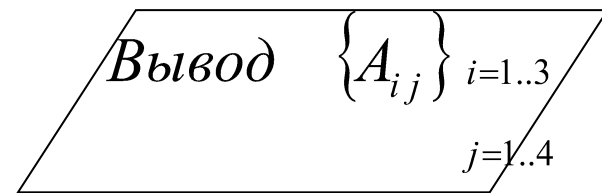
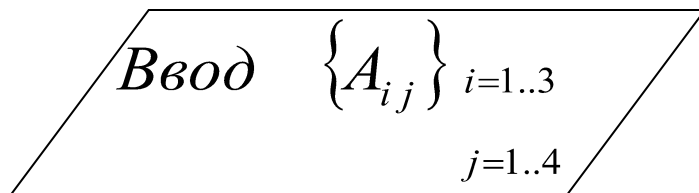
Ввод элементов



Вывод элементов



Краткая запись (только для лекций):



Пример1: Сумма всех элементов матрицы А(3x4)

I этап:

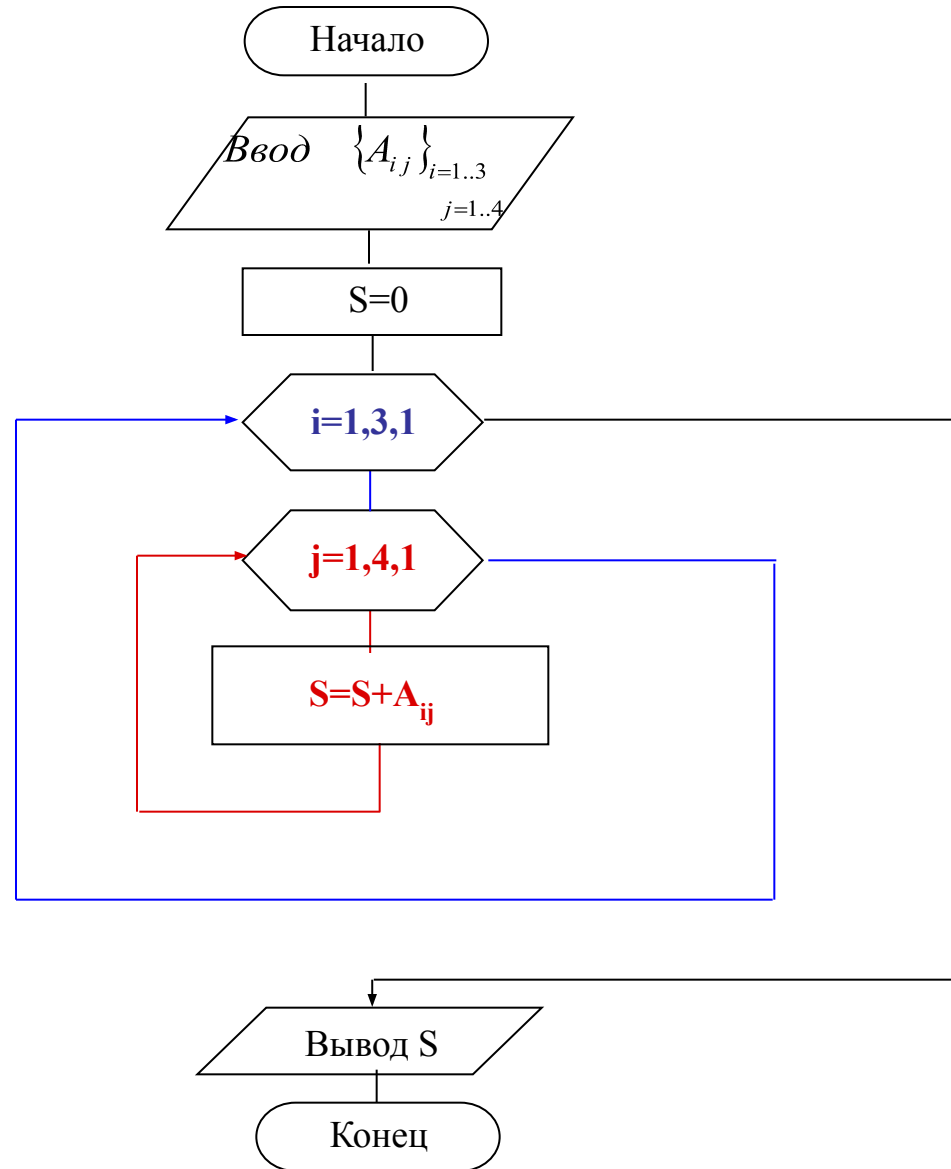
$$S = \sum_{i=1}^3 \sum_{j=1}^4 A_{ij}$$

II этап: входные данные – матрица А,
промежуточные данные – i, j
выходные данные – сумма S

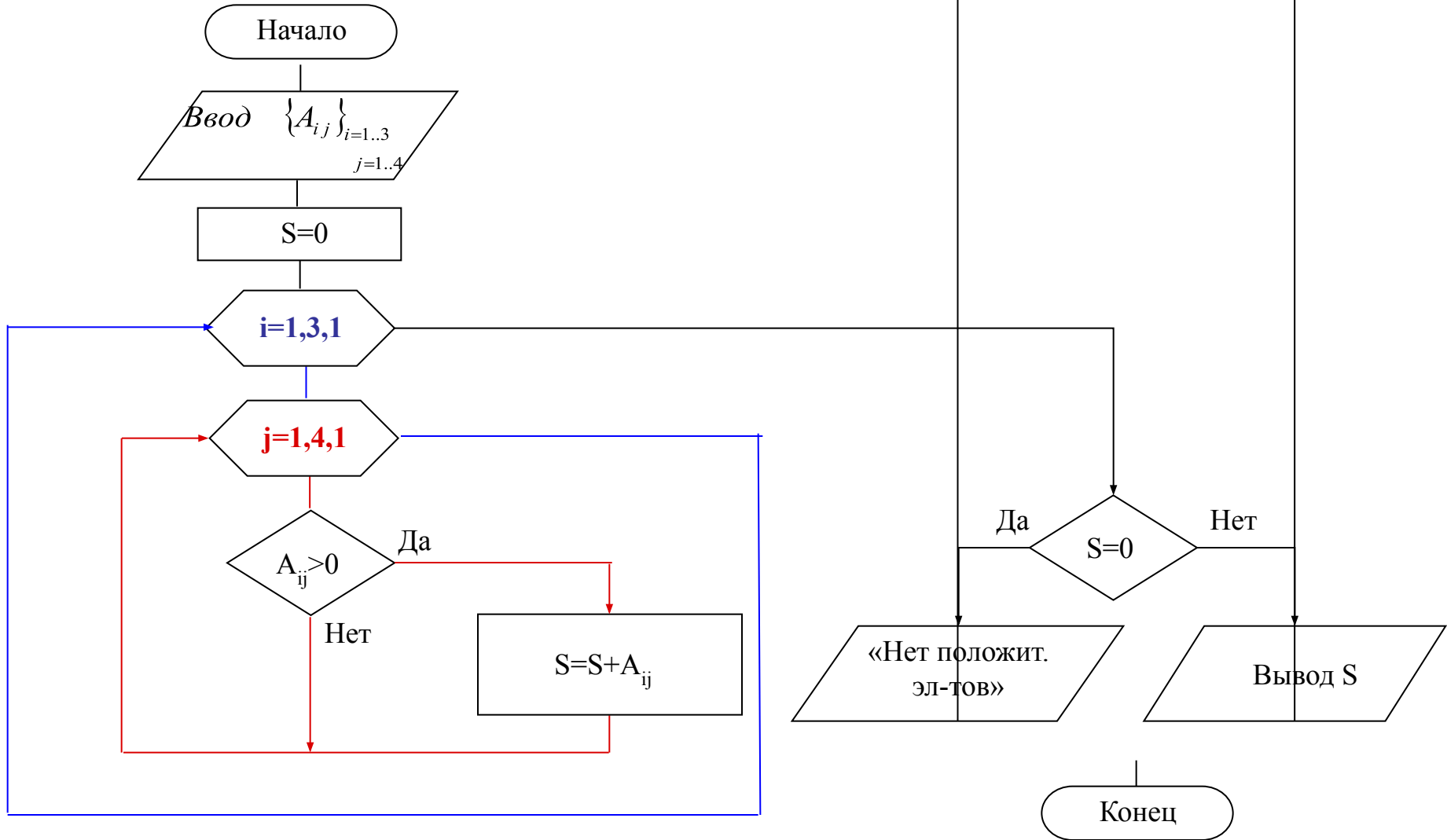
III этап: разработка алгоритма:

Рекуррентная формула суммы:

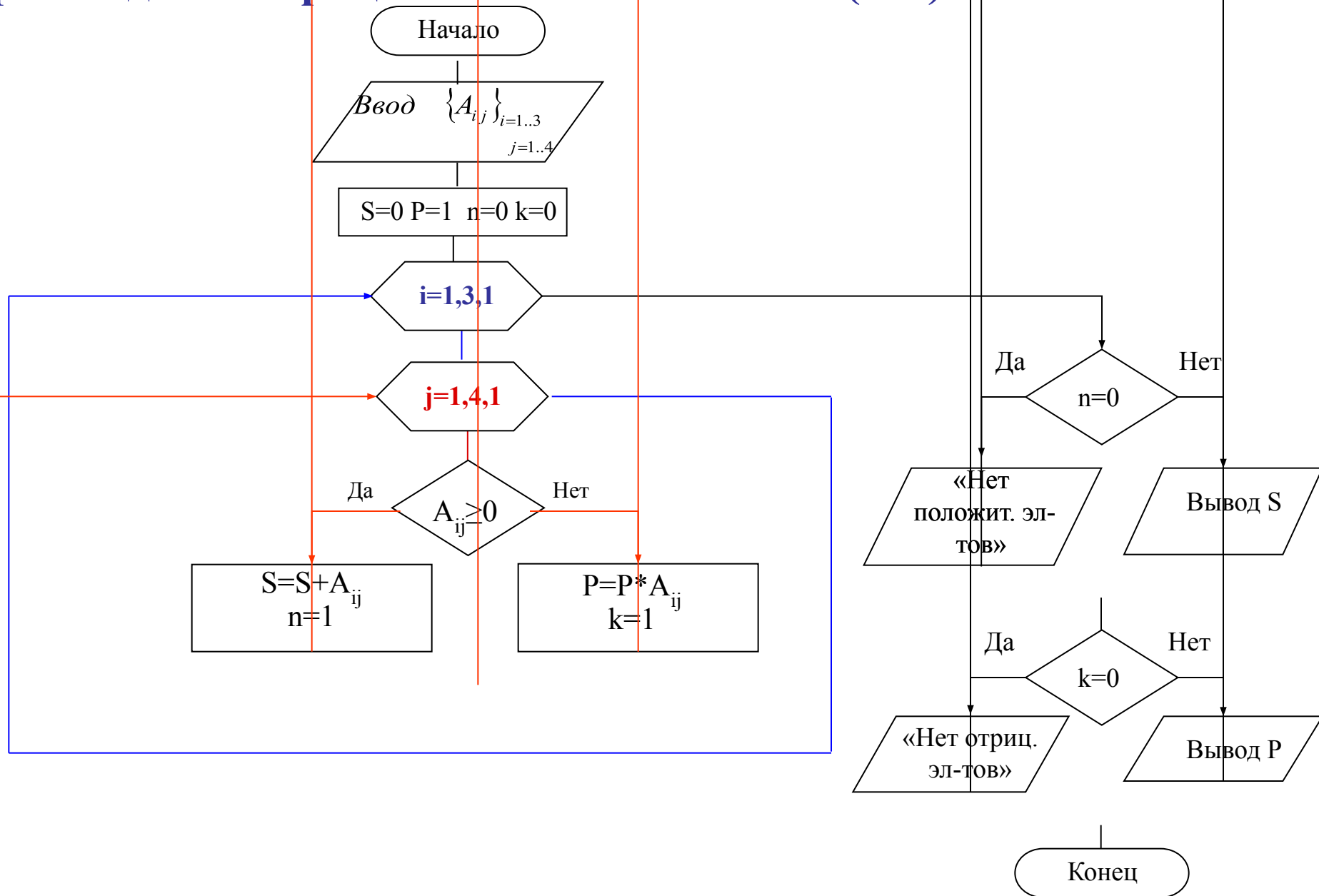
$$S = S + A_{ij}$$



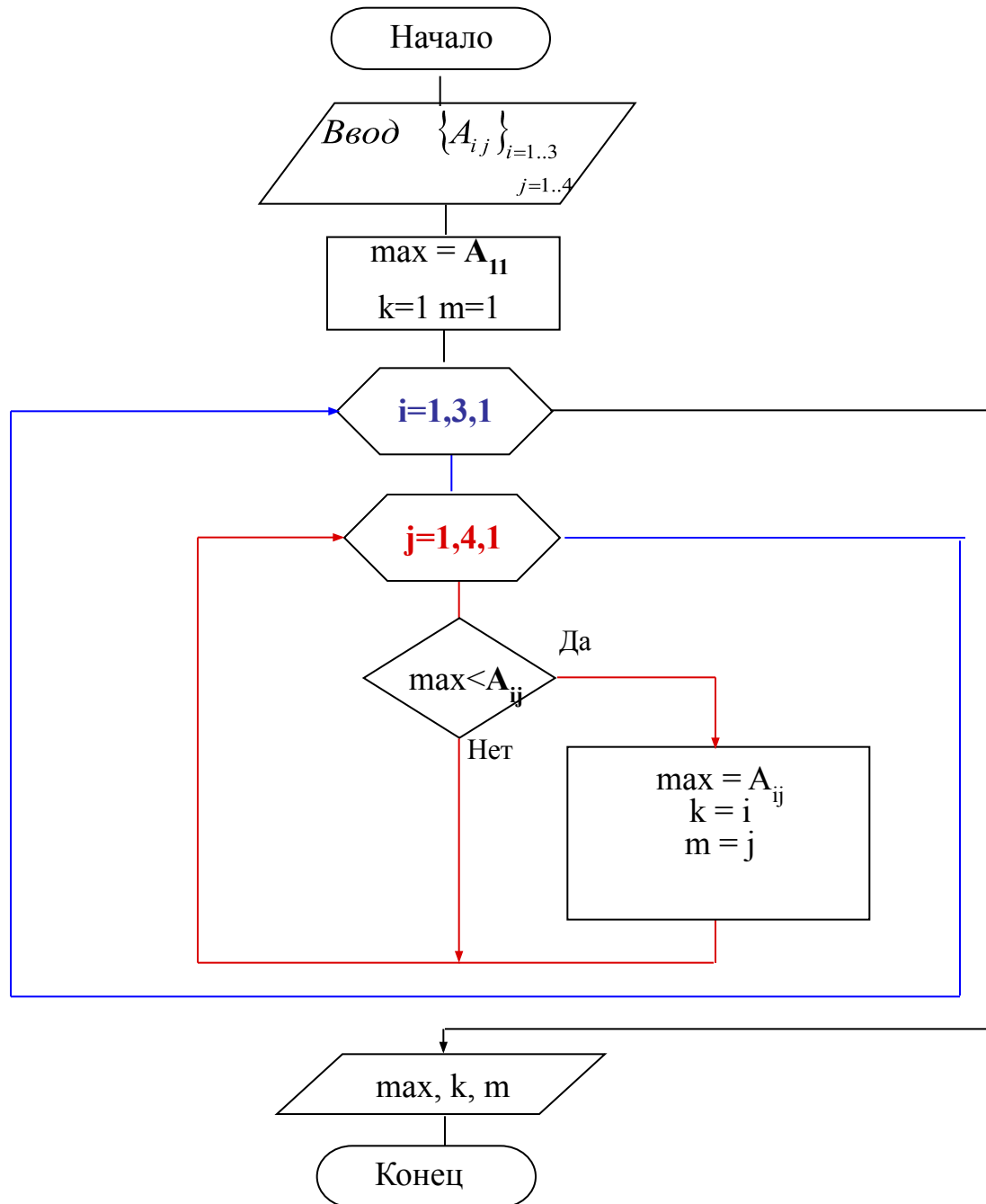
Пример 2: Сумма положительных элементов матрицы A (3x4)



Пример 3: Сумма положительных и произведение отрицательных элементов $A(3 \times 4)$



Пример 4: Максимальный элемент матрицы $A(3 \times 4)$ и его номер

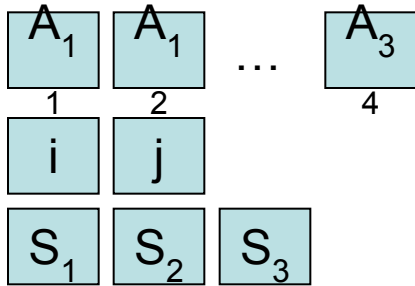


Пример 5: Сумма элементов каждой строки матрицы A(3x4)

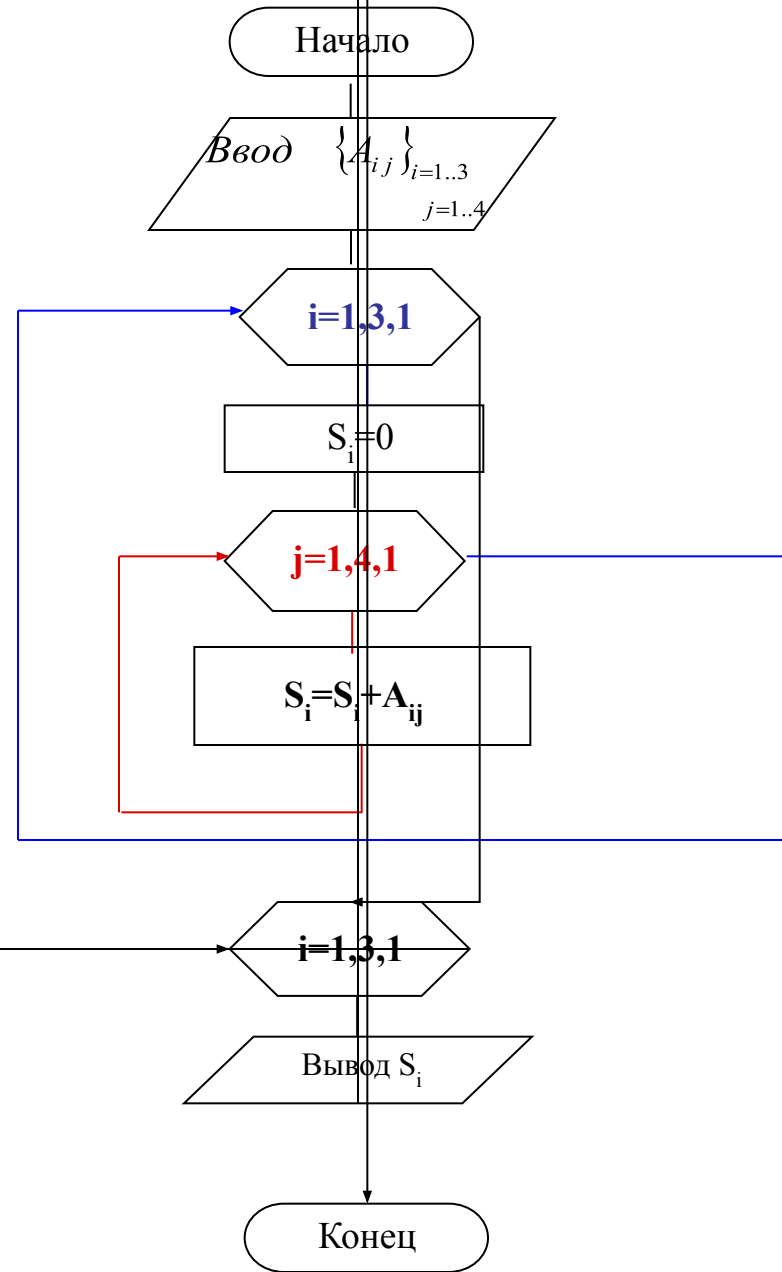
I этап:
$$S_i = \sum_{j=1}^4 A_{ij}$$

II этап: входные данные – матрица A,
промежуточные данные – i, j
выходные данные – массив S_i

Распределение ячеек ОЗУ:



III этап: разработка алгоритма:



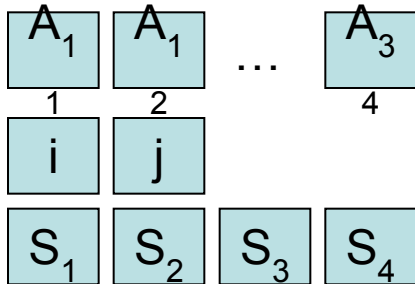
Пример 6: Сумма элементов каждого столбца матрицы A(3x4)

I этап:

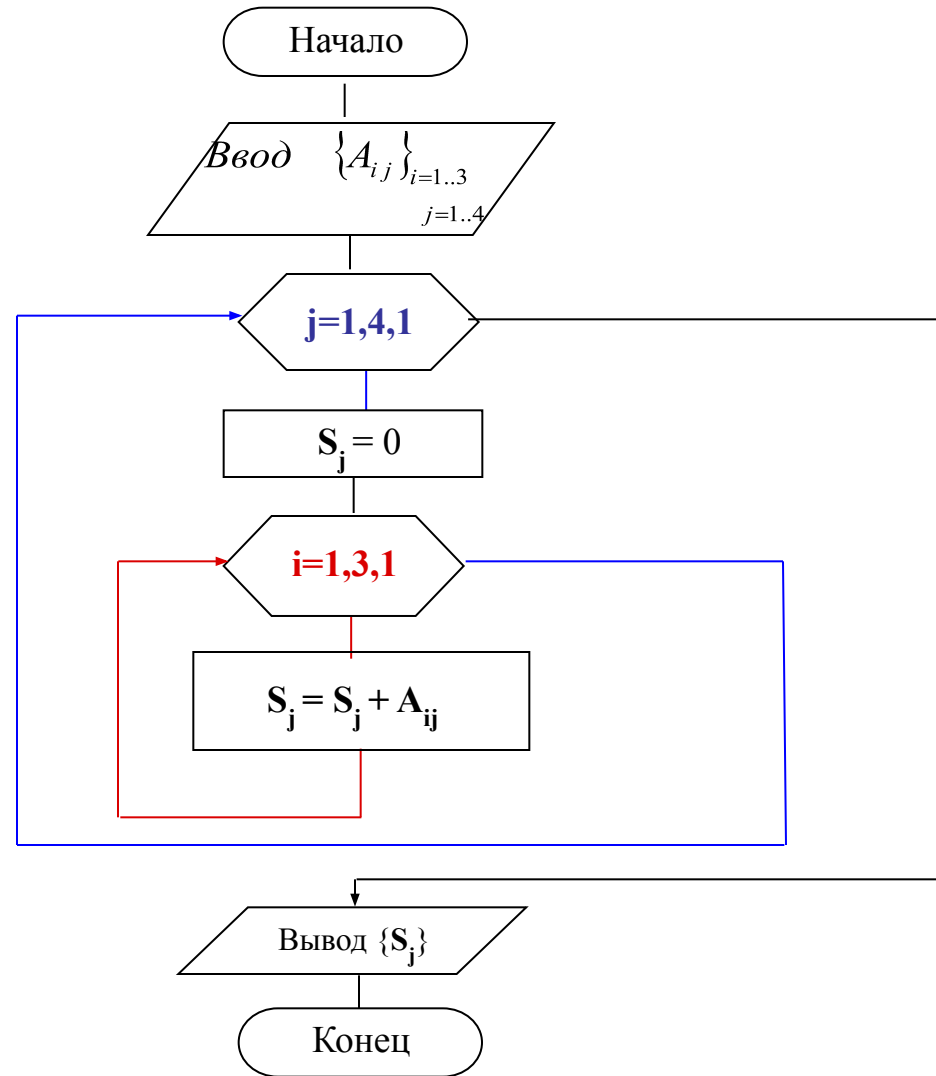
$$S_j = \sum_{i=1}^3 A_{ij}$$

II этап: входные данные – матрица A_{ij} ,
промежуточные данные – i, j
выходные данные – массив S_i

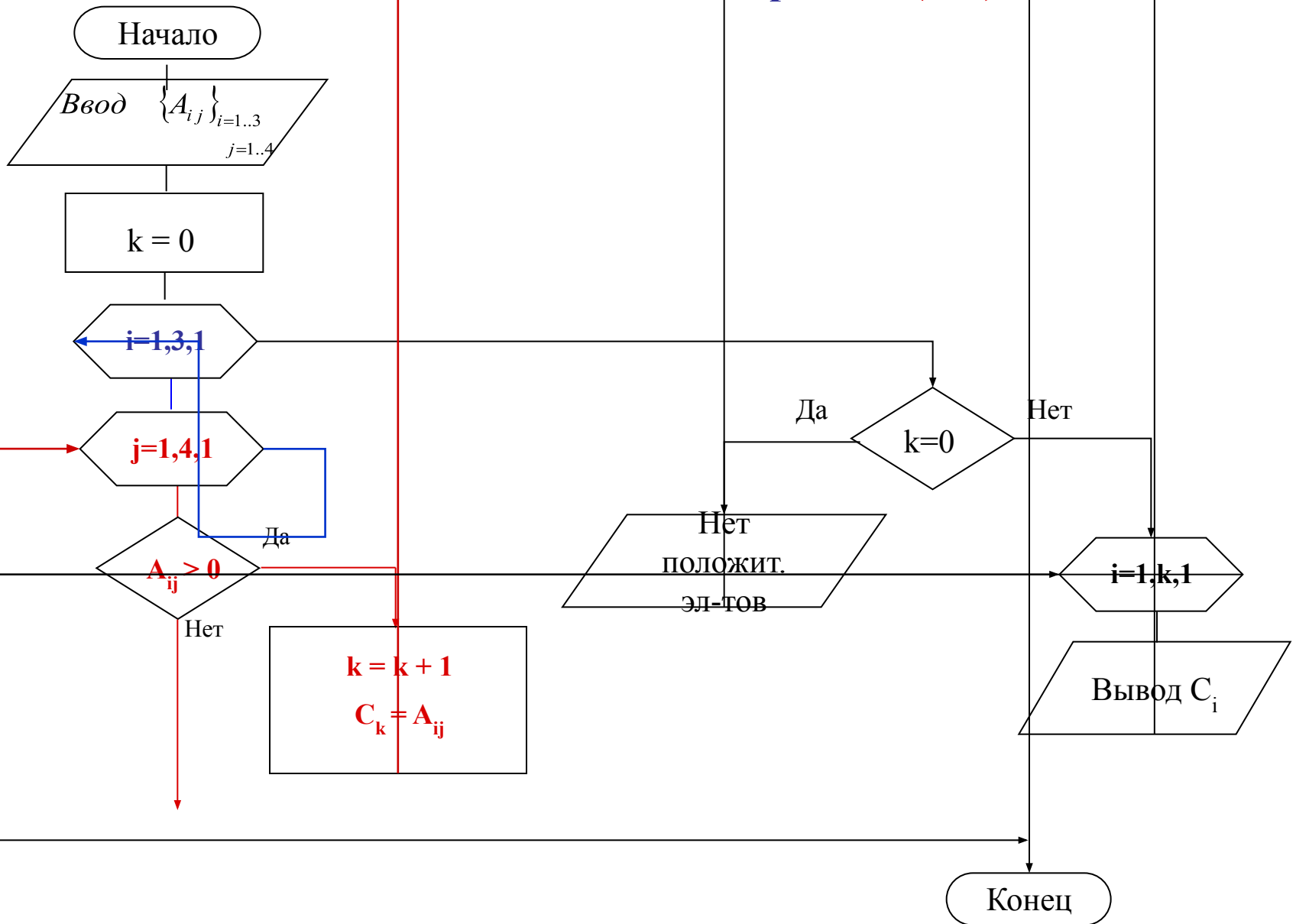
Распределение ячеек ОЗУ:



III этап: разработка алгоритма:



Пример 7: Формирование одномерного массива C из положительных элементов матрицы $A(3 \times 4)$



Лекция № 6

Алгоритмы обработки квадратных матриц

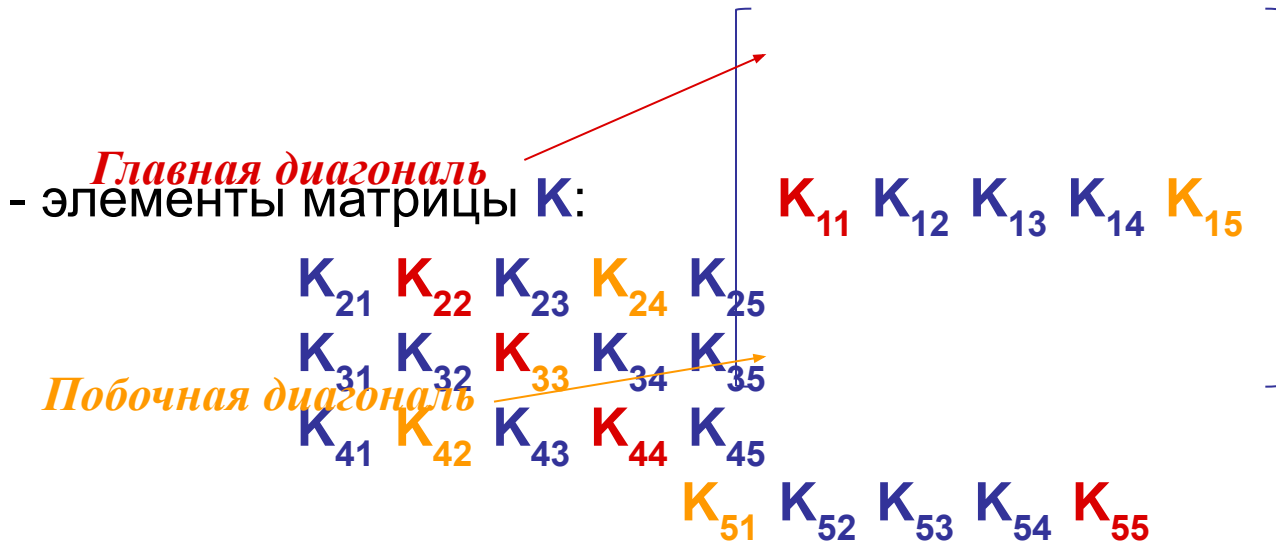
Квадратные матрицы

Число строк = числу столбцов

Обозначения квадратных матриц

1. **Словесное описание:** Матрица $A(n,n)$ целых чисел
2. **Математическое описание:** матрица A : $\{ A_{ij} \}_{i=1,n j=1,n}$ или $A(n \times n)$

- например, матрица $K(5,5)$: $\{ K_{ij} \}_{i=1,5 j=1,5}$ или $K(5 \times 5)$

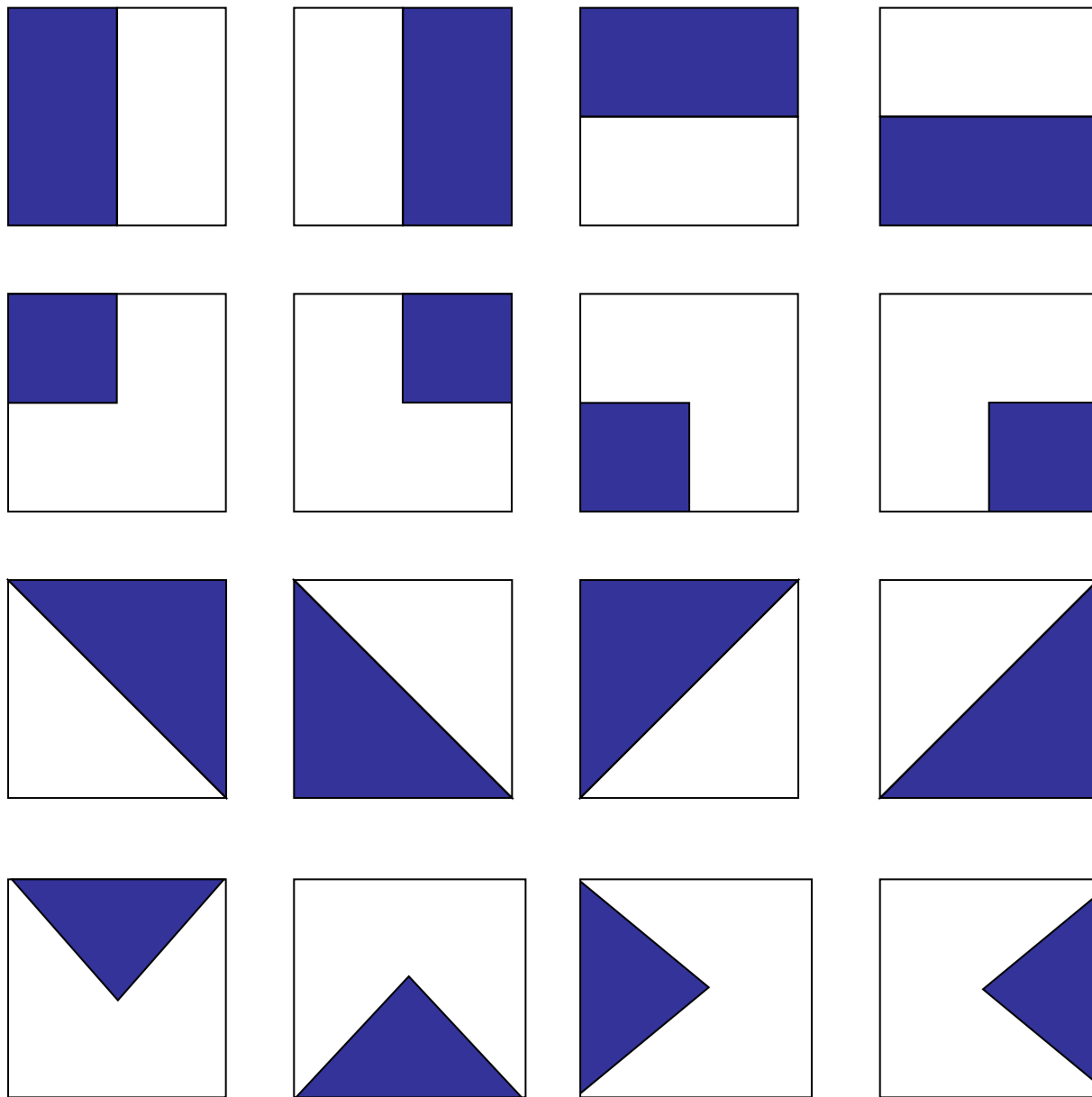


- Замечание.**
1. **Размерность** массива K равна **2**.
 2. **Размер** массива K - **$5 \times 5 = 25$** элементов

Стандартные алгоритмы для обработки квадратных матриц

1. Вычисление суммы, произведения, количества элементов в указанной области, удовлетворяющих условию
2. Нахождение максимального и минимального элемента в указанной области матрицы
3. Вычисление суммы (произведения, максимального, минимального и т.п.) в каждой строке или каждом столбце указанной области матрицы
4. Формирование одномерного массива из элементов указанной области матрицы

Примеры областей в матрицах



Возможны
и другие
варианты!

Определение областей

1. Выше главной диагонали

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$i=1 \quad j=1,2,3,4,5$
 $i=2 \quad j=2,3,4,5$
 $i=3 \quad j=3,4,5$
 $i=4 \quad j=4,5$
 $i=5 \quad j=5$

$A(5,5) \quad i = 1, 5, 1$
 $j = i, 5, 1$

$A(n,n) \quad i = 1, n, 1$
 $j = i, n, 1$

2. Ниже главной диагонали

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$i=1 \quad j=1$
 $i=2 \quad j=1,2$
 $i=3 \quad j=1,2,3$
 $i=4 \quad j=1,2,3,4$
 $i=5 \quad j=1,2,3,4,5$

$A(5,5) \quad i = 1, 5, 1$
 $j = 1, i, 1$

$A(n,n) \quad i = 1, n, 1$
 $j = 1, i, 1$

3. Выше побочной диагонали

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$i=1 \quad j=1,2,3,4,5$
 $i=2 \quad j=1,2,3,4$
 $i=3 \quad j=1,2,3$
 $i=4 \quad j=1,2$
 $i=5 \quad j=1$

$A(5,5) \quad i = 1, 5, 1$
 $j = 1, 6-i, 1$

$A(n,n) \quad i = 1, n, 1$
 $j = 1, n+1-i, 1$

4. Ниже побочной диагонали

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$i=1 \quad j=5$
 $i=2 \quad j=4,5$
 $i=3 \quad j=3,4,5$
 $i=4 \quad j=2,3,4,5$
 $i=5 \quad j=1,2,3,4,5$

$A(5,5) \quad i = 1, 5, 1$
 $j = 6-i, 5, 1$

$A(n,n) \quad i = 1, n, 1$
 $j = n+1-i, n, 1$

5. Верхний треугольник

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$i=1 \quad j=1,2,3,4,5$
 $i=2 \quad j=2,3,4$
 $i=3 \quad j=3$

$A(5,5) \quad i = 1, 3, 1$
 $j = i, 6-i, 1$

$A(n,n) \quad i = 1, n/2, 1$
 $j = i, n+1-i, 1$

6. Нижний треугольник

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$i=3 \quad j=3$
 $i=4 \quad j=2,3,4$
 $i=5 \quad j=1,2,3,4,5$

$A(5,5) \quad i = 3, 5, 1$
 $j = 6-i, i, 1$

$A(n,n) \quad i = n/2, n, 1$
 $j = n+1-i, i, 1$

7. Левый треугольник

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$j=1 \quad i=1,2,3,4,5$
 $j=2 \quad i=2,3,4$
 $j=3 \quad i=3$

$A(5,5) \quad j = 1, 3, 1$
 $i = j, 6-j, 1$

$A(n,n) \quad j = 1, n/2, 1$
 $i = j, n+1-j, 1$

8. Правый треугольник

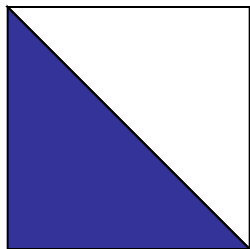
A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

$j=3 \quad i=3$
 $j=4 \quad i=2,3,4$
 $j=5 \quad i=1,2,3,4,5$

$A(5,5) \quad j = 3, 5, 1$
 $i = 6-j, j, 1$

$A(n,n) \quad j = n/2, n, 1$
 $i = n+1-j, j, 1$

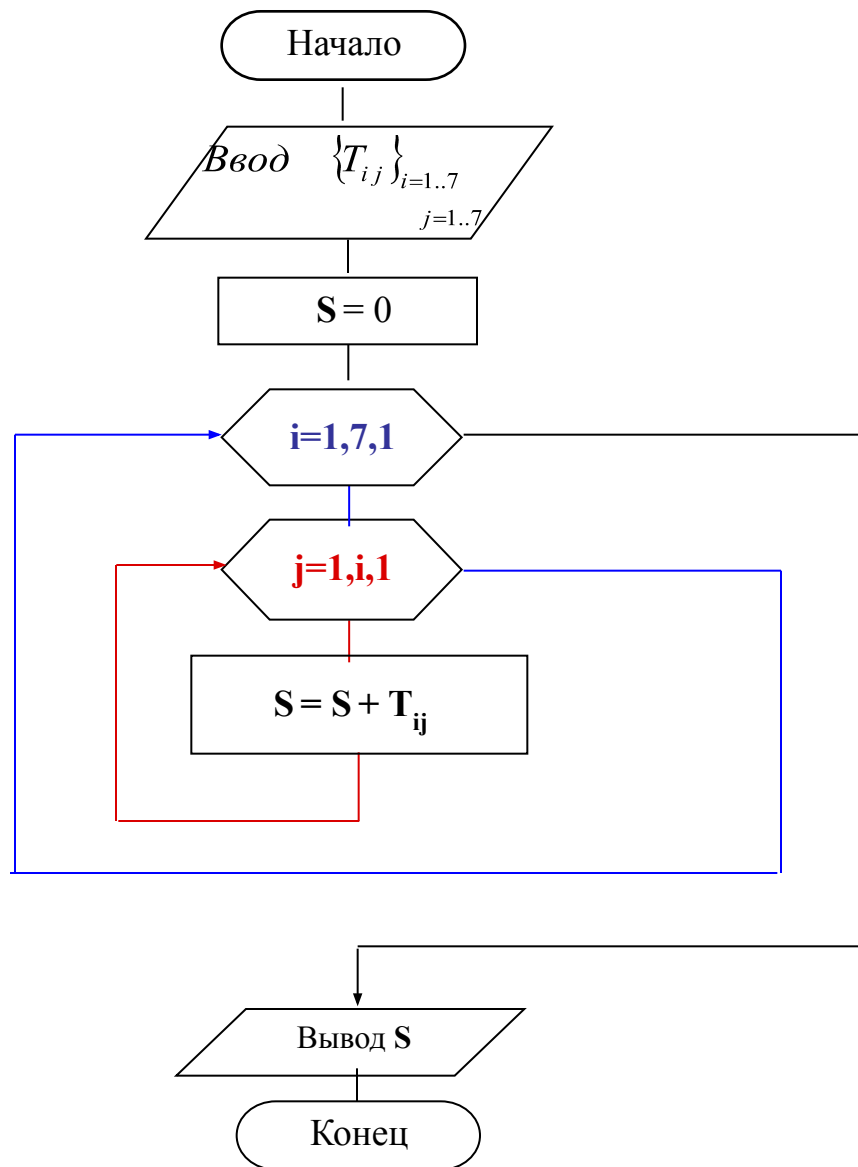
Пример 1: Сумма элементов в заштрихованной области матрицы T(7,7)



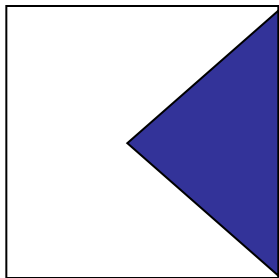
I этап:
$$S = \sum_{i=1}^7 \sum_{j=1}^i T_{ij}$$

II этап: входные данные – матрица T,
промежуточные данные – i, j
выходные данные – S

III этап: разработка алгоритма:



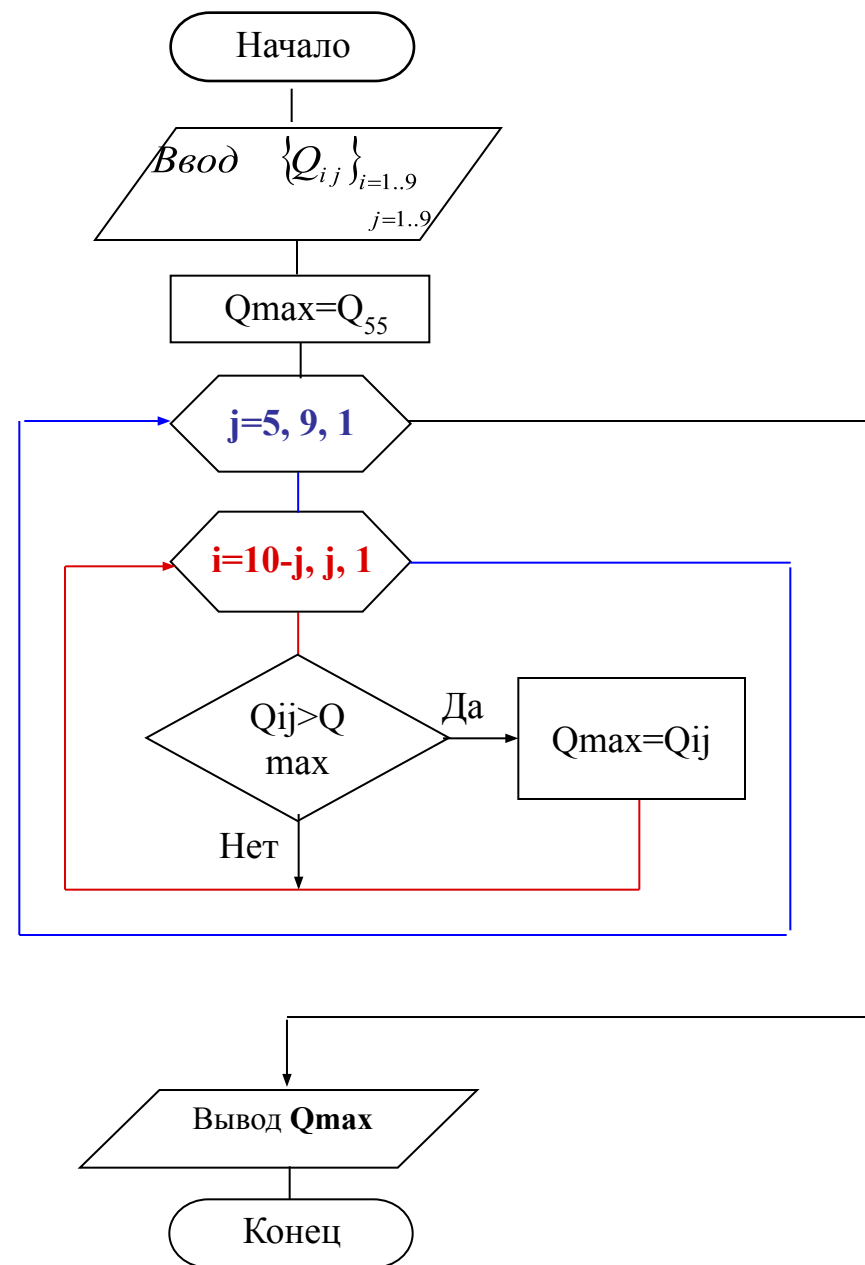
Пример 2: Максимальный элемент в заштрихованной области матрицы $Q(9,9)$



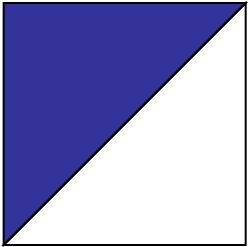
I этап: Q_{\max} среди Q_{ij} в заштрих. области
($j=5,9,1, i=10-j,j,1$)

II этап: входные данные – матрица Q ,
промежуточные данные – i, j
выходные данные – Q_{\max}

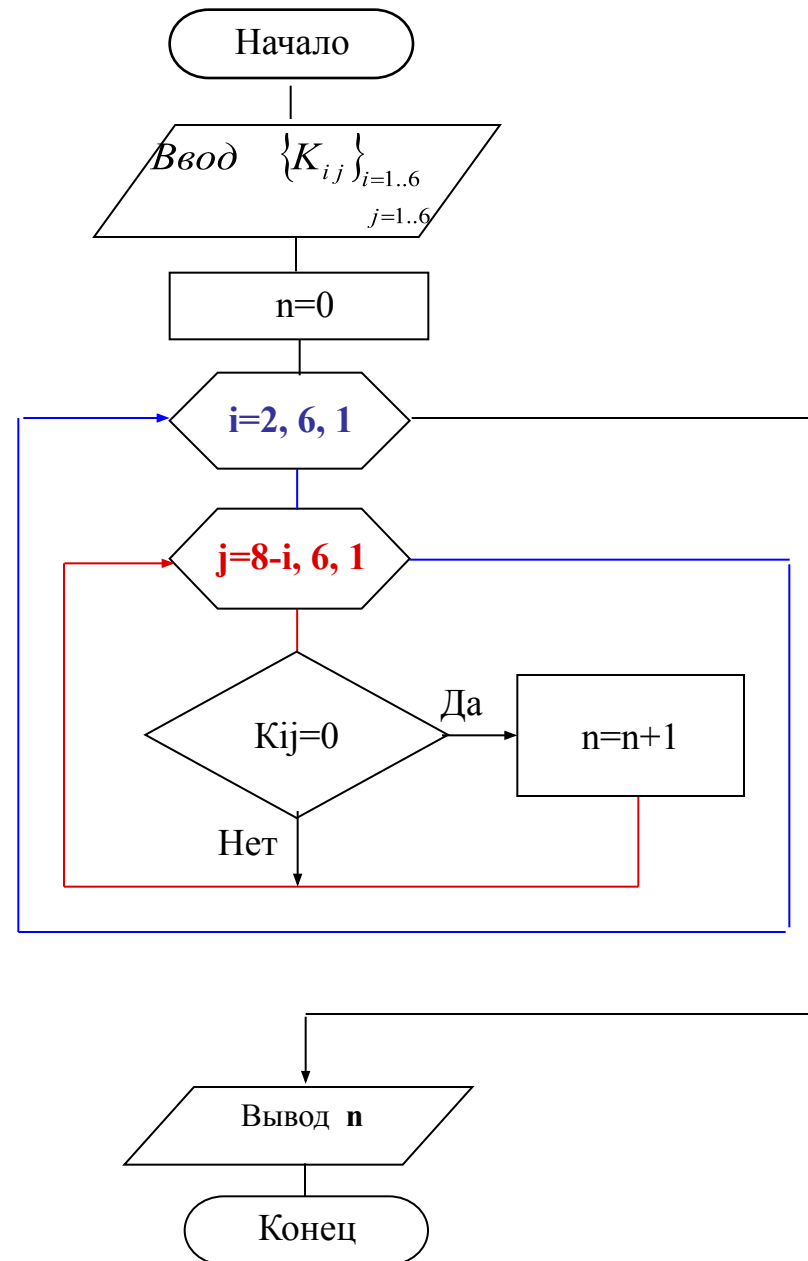
III этап: разработка алгоритма:



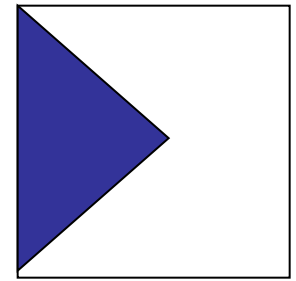
Пример 3: Количество нулевых элементов вне заштрихованной области матрицы $K(6,6)$



K_{11}	K_{12}	K_{13}	K_{14}	K_{15}	K_{16}
K_{21}	K_{22}	K_{23}	K_{24}	K_{25}	K_{26}
K_{31}	K_{32}	K_{33}	K_{34}	K_{35}	K_{36}
K_{41}	K_{42}	K_{43}	K_{44}	K_{45}	K_{46}
K_{51}	K_{52}	K_{53}	K_{54}	K_{55}	K_{56}
K_{61}	K_{62}	K_{63}	K_{64}	K_{65}	K_{66}



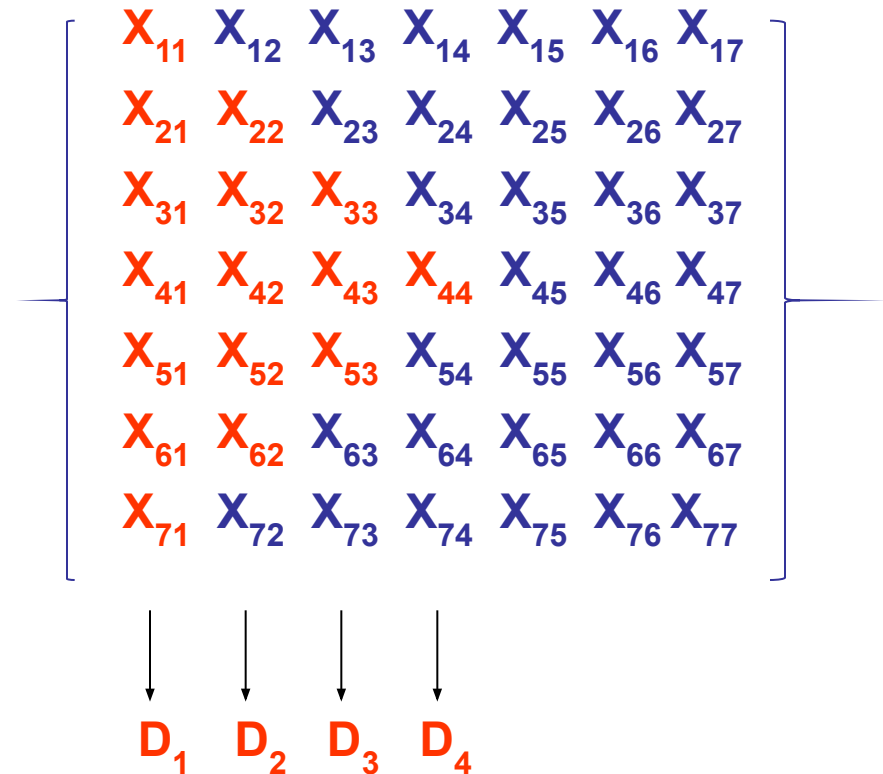
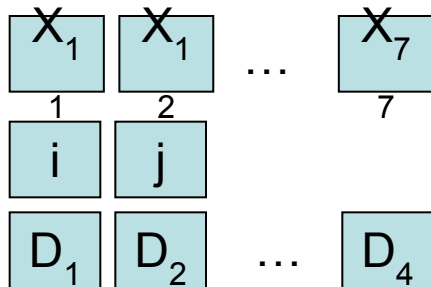
Пример 4: Сформировать одномерный массив D из количеств положительных элементов каждого столбца в заштрихованной области матрицы X(7,7)



I этап: Заштрих. область
($j=1..4, i=j, 8-j$)

II этап: входные данные – матрица X,
промежуточные данные – i, j
выходные данные – массив D

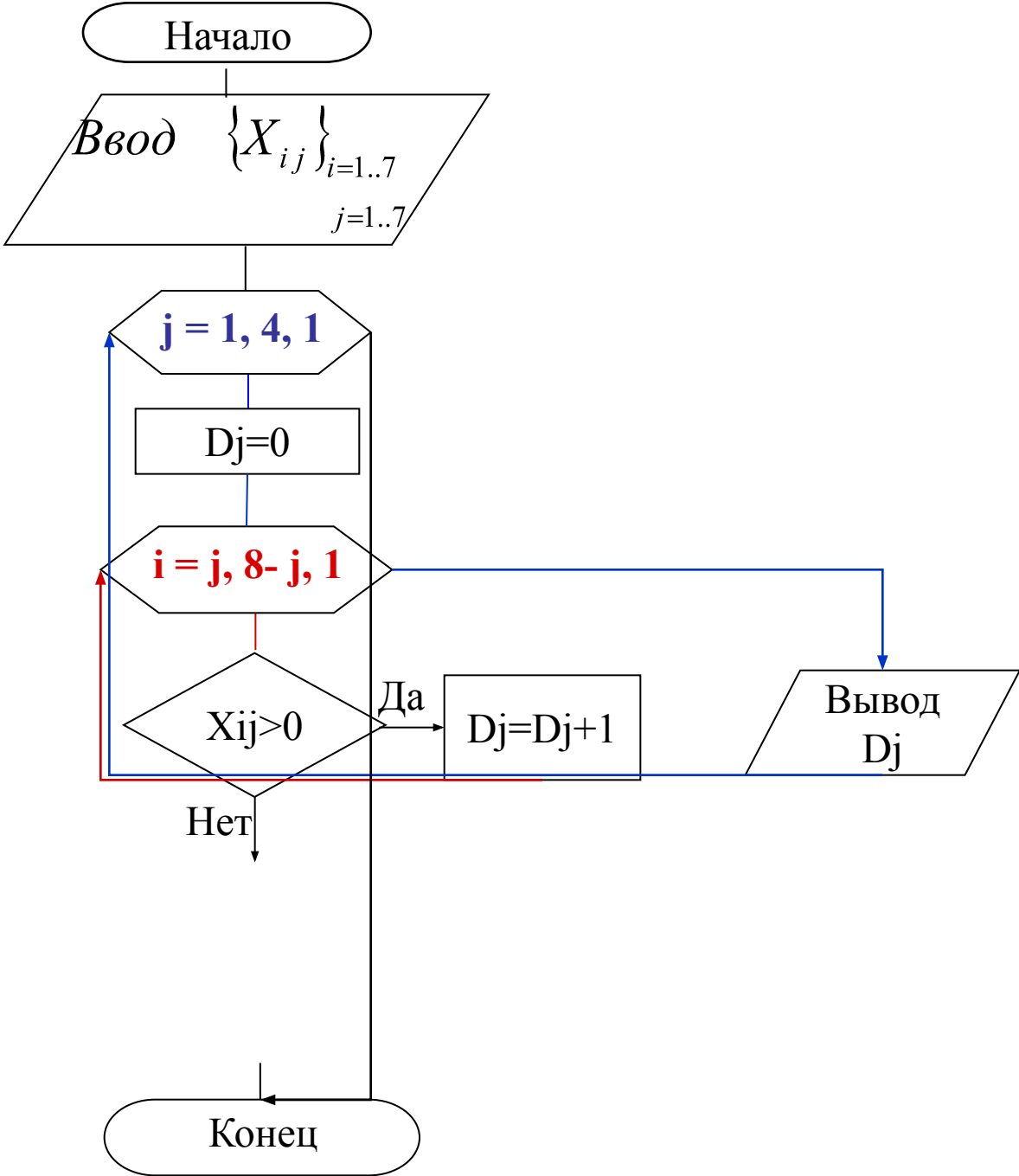
Распределение ячеек ОЗУ:



III этап: Словесное описание

Просматривать по столбцам элементы заштрихованной области. В начале просмотра столбца присвоить $D_j=0$. Если рассматриваемый элемент X_{ij} положительный, то увеличить D_j на 1.

Схема алгоритма:



Чтение алгоритмов

Пример

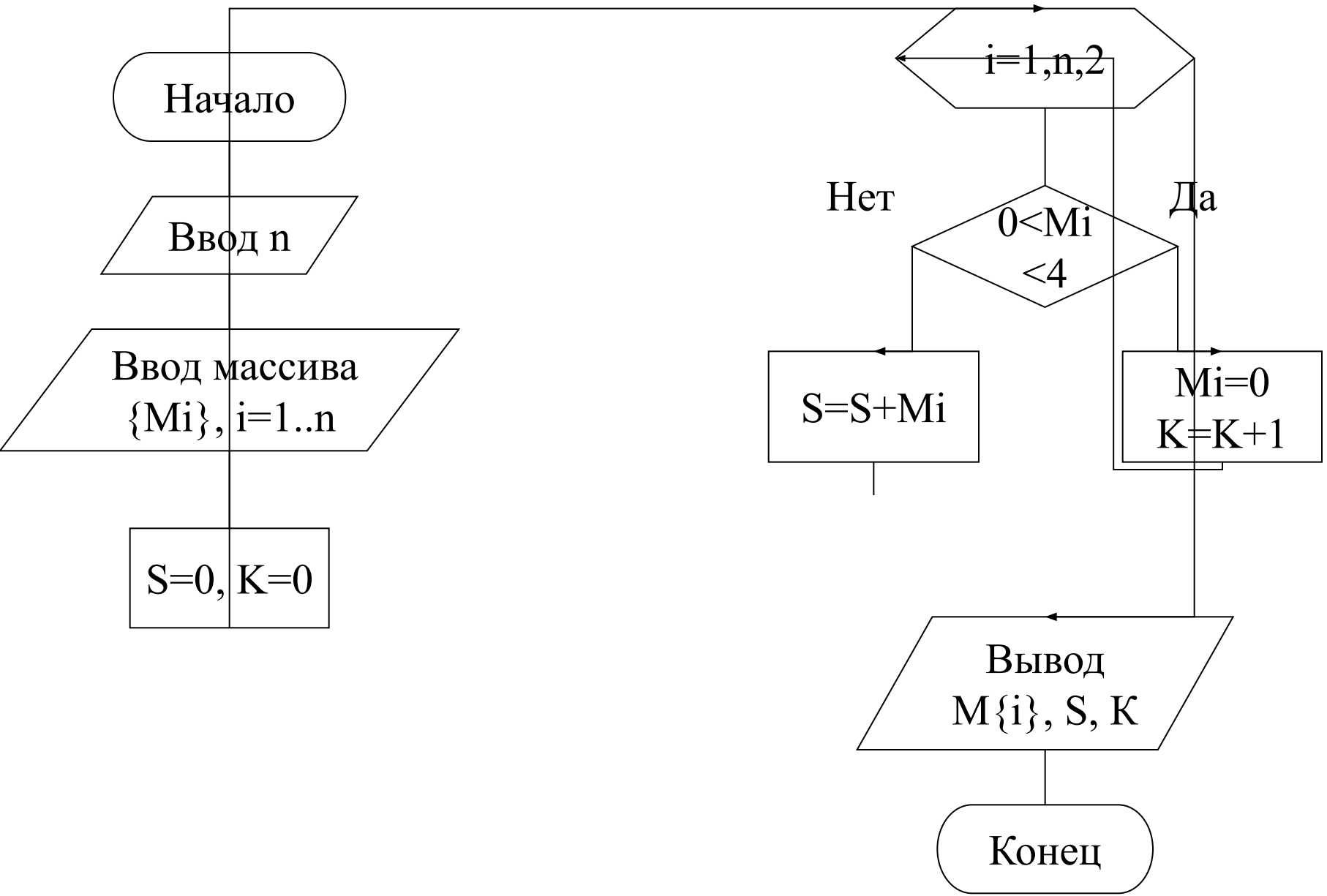
Дано:

1. Схема алгоритма
2. Входной поток данных:
8 -4 -1 3 -2 1 -1 5 6

Задача:

1. Сформулировать постановку задачи
2. Нарисовать все состояния ячеек ОЗУ
3. Определить значения выходных данных

Схема алгоритма:



Подход к решению

- Определить тип задачи (сумма, произведение, количество, max, min, ветвление и т.п.):
 - Что выводится (выходные данные)?
 - Какие начальные значения?
 - Как вычисляются выходные данные?
- Выписать все переменные, задействованные в задаче; нарисовать ячейки ОЗУ
- Определить состояния ячеек ОЗУ (ввод данных, каждый шаг каждого цикла вычислений, вывод данных)
- Записать значения выходных данных

Решение

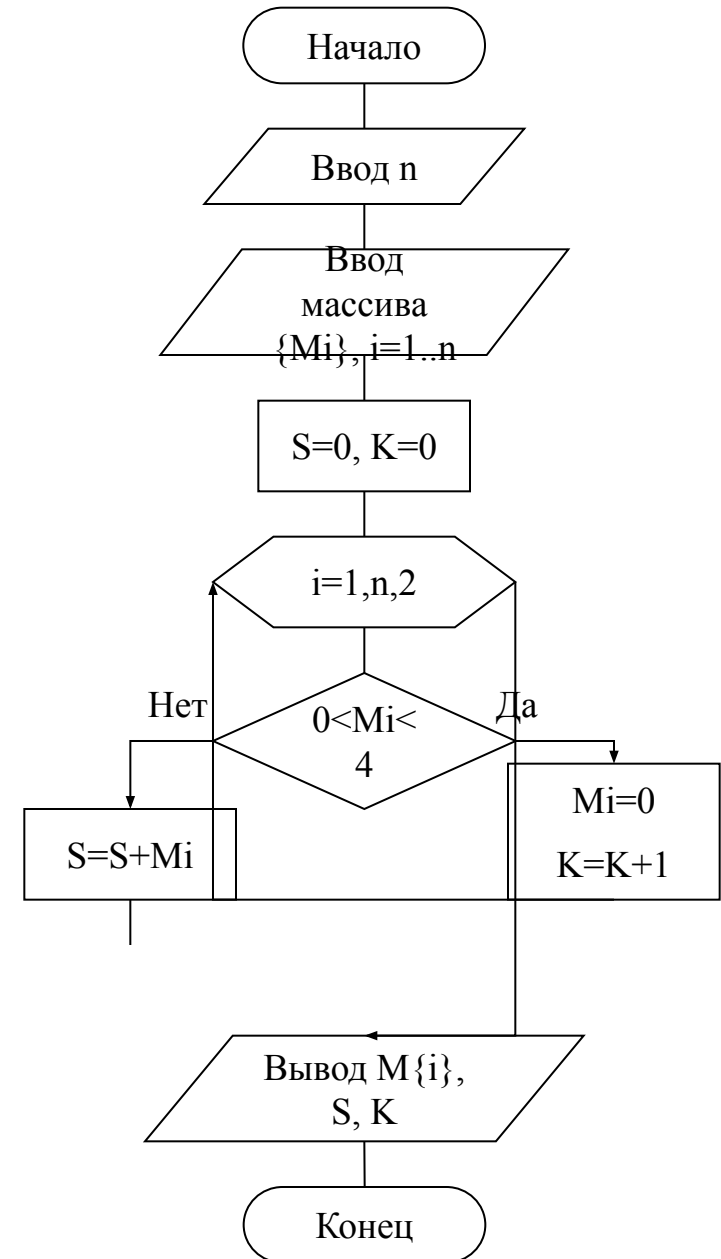
1. Сформулировать постановку задачи

Ввести массив заданного размера.

Среди элементов с нечетными индексами:

- заменить на 0 все элементы от 0 до 4 и посчитать количество таких элементов,
- определить сумму элементов, не принадлежащих этому интервалу.

Вывести массив, сумму и количество.



2. Нарисовать все состояния ячеек ОЗУ

n	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇	M ₈	S	K	i
-	-	-	-	-	-	-	-	-	-	-	-

} Первоначально

2. (продолжение)

n	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇	M ₈	S	K	i
8	-4	-1	3	-2	1	-1	5	6	0	0	8

Было перед
основным
циклом

Примечание:

В зависимости от среды и языковой конструкции, используемой при программировании алгоритма, возможны следующие значения переменной **i** после выполнения цикла:

i=8 (конечное значение в цикле)

i не определено (и др.)

3. Определить значения выходных данных

Выходные данные:

$\{M_i\}$: -4 -1 0 -2 0 -1 5 6

$S = 1$

$K = 2$