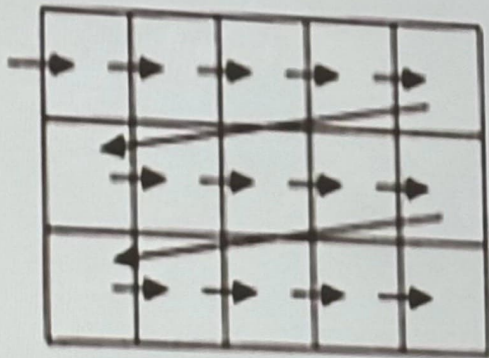


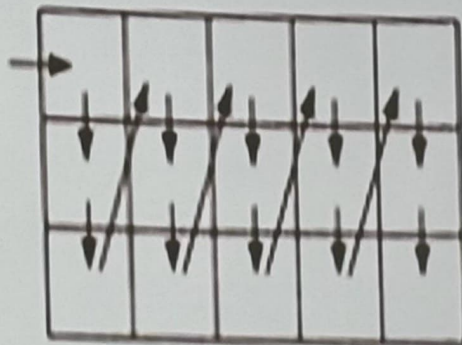
## Занятие 6. Матрицы. Вложенные циклы. Алгоритмы просмотра элементов

1. Определение матрицы. В общем понятии матрица - это двумерный массив. Для программирования все операции работы с матрицами имеют одну специфику: фактически, матрица - это массив одномерных массивов.

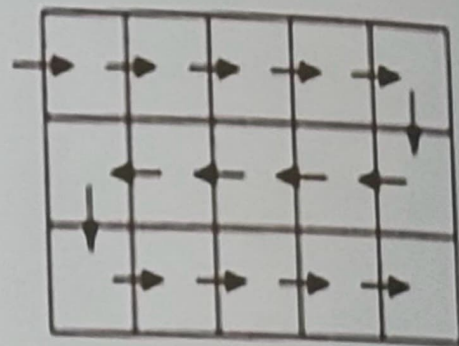
### 2. Разбор вариантов обхода матрицы.



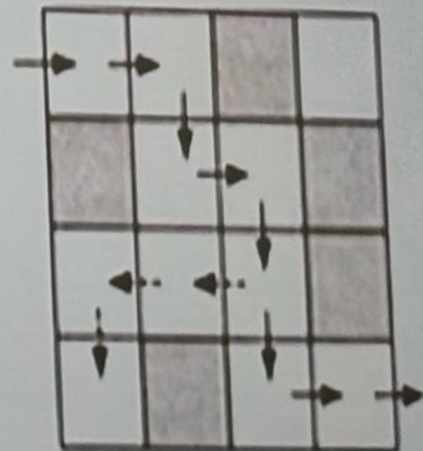
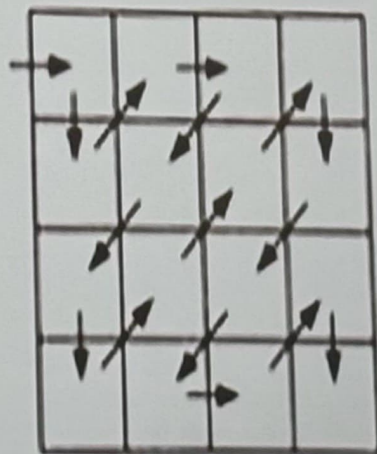
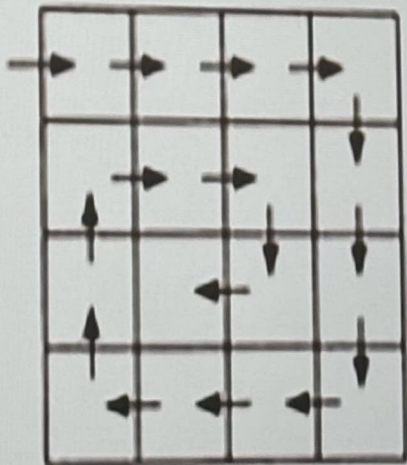
а



б



в



- Задача 1.** Определить максимальную сумму положительных элементов строк матрицы  $A(n,m)$ ,  $n, m \leq 20$ .
- Задача 2.** Разместить в начале массива различные элементы, встречающиеся в нем, и определить их количество.
- Задача 3.** Определить суммы элементов фрагментов, на которые матрицу делят диагонали, проведенные через элемент  $(k,l)$ .

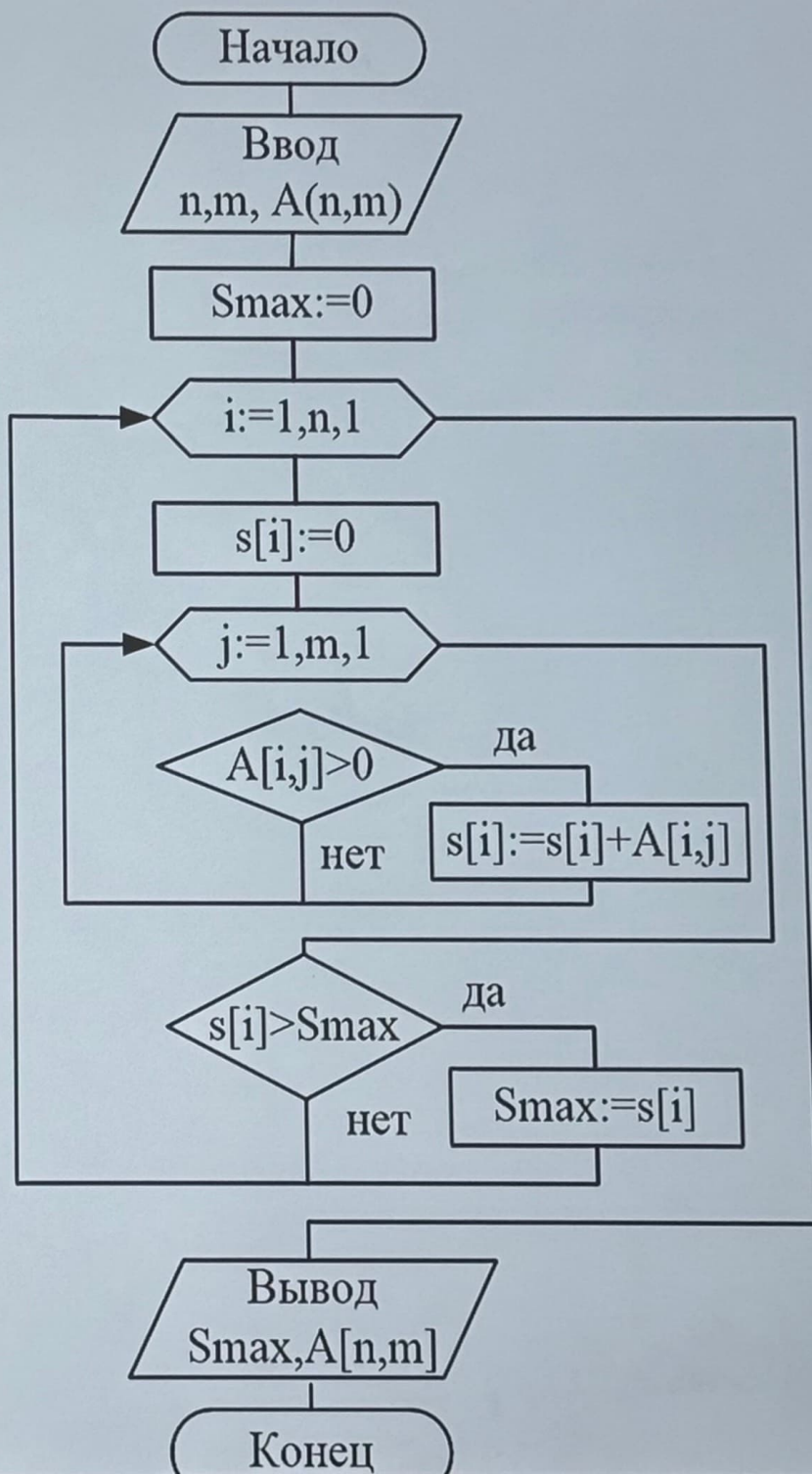
**Задача 1.** Определить максимальную сумму положительных элементов строк матрицы  $A(n,m)$ ,  $n, m \leq 20$ .

### *Разбор решения задачи 1*

В задаче рассматривается матрица, размещенная в *статической памяти*, т.е. памяти, выделяемой при компиляции программы. В то же время конкретный размер матрицы не задан. Следовательно необходимо предусмотреть описание матрицы максимально необходимого по условию задачи размера, например  $A(20,20)$ , а конкретный размер  $n$  матрицы вводить с клавиатуры.

Так как матрица вводится, суммируется и выводится построчно, все просмотры можно выполнить в одном цикле. Вероятность внесения ошибок в программу уменьшится, если это решение выполнять поэтапно: ввод матрицы, нахождение сумм элементов строк и выбор максимальной суммы, вывод матрицы и результата (рисунок 1).





**Рисунок 1** – Схема алгоритма решения задачи 1

**Внимание!** Если элементы матрицы – вещественные числа, то при задании  $S_{\max}=0$ , может возникнуть ситуация, что подсчитанная сумма в строке будет очень близка к 0 и если задать слишком ограниченный параметр поля вывода ответа, то можно получить такую ситуацию, что ответ выведется неверным, равным 0, но решение таковым в реальности не будет. Либо может возникнуть ситуация с выводом индекса строки с максимальной суммой элементов- такую строку можно пропустить из-за близости ответа к нулю.

Поэтому, лучше использовать разницу значений по модулю, сравниваемую с  $\epsilon$  (погрешностью вычислений):  $|S_{\max}-S[i]| < \epsilon$  и задать  $S_{\max} = -1e10$ .

**Задача 2.** Разместить в начале массива различные элементы, встречающиеся в нем, и определить их количество.

**Разбор решения задачи 2**

Всего можно предложить более 6 алгоритмов решения задачи

Для реализации выбираем вариант, который является наиболее эффективным по времени решения и используемой памяти. Алгоритм заключается в следующем (рисунок 2):

- первый элемент массива остается на месте.



разницу значений по модулю, сравниваемую с  $\epsilon$ ps  
количество вычислений):  $|S_{\max} - S[i]| < \epsilon$ ps и задать  $S_{\max} = -1e10$ .

**Задача 2.** Разместить в начале массива различные элементы, встречающиеся в нем, и определить их количество.

### *Разбор решения задачи 2*

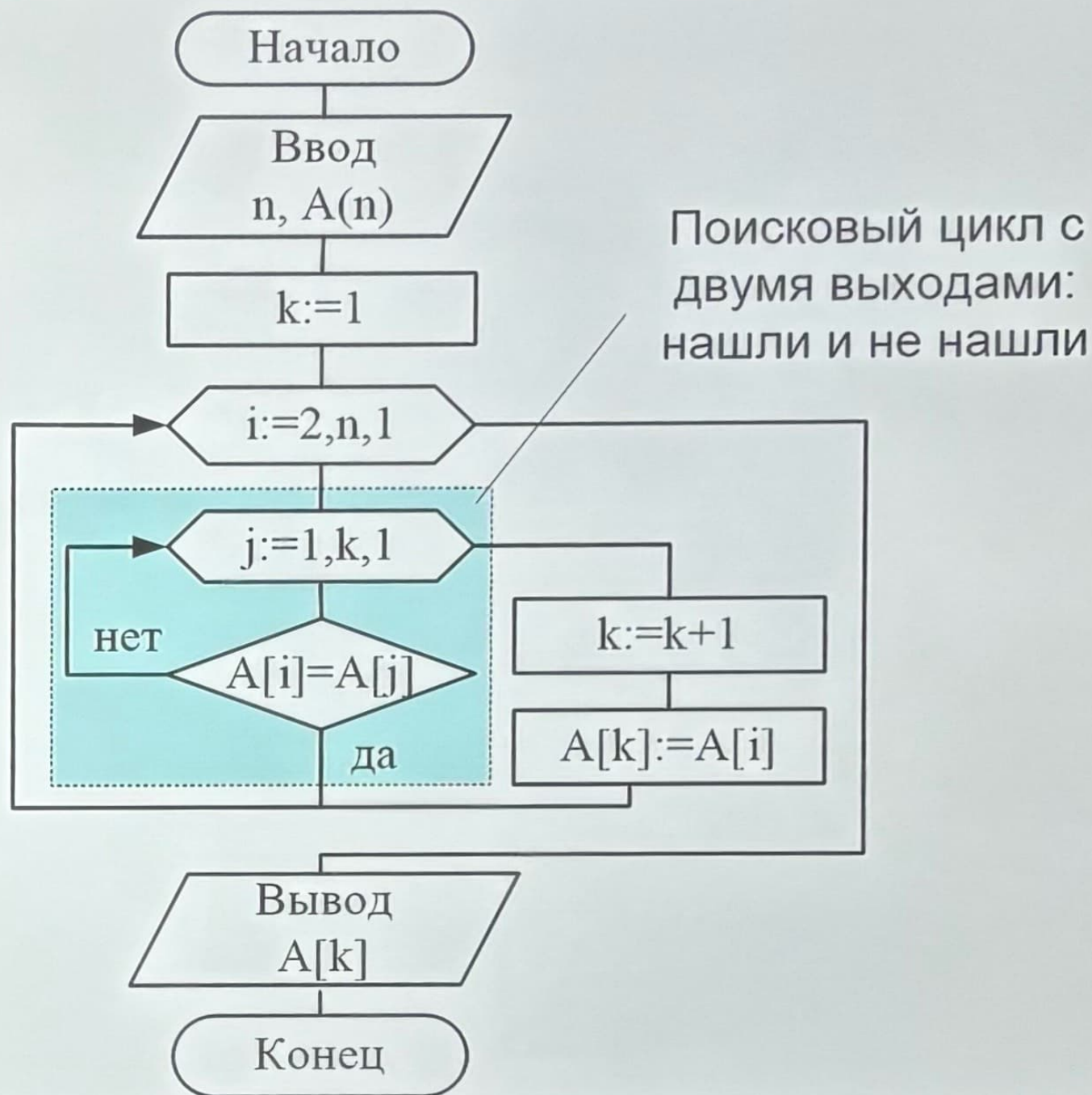
Всего можно предложить более 6 алгоритмов решения задачи

Для реализации выбираем вариант, который является наиболее эффективным по времени решения и используемой памяти. Алгоритм заключается в следующем (рисунок 2):

- первый элемент массива остается на месте,
- затем каждый следующий элемент сравнивается с уже найденными различными,
- если такого элемента в перечне найденных еще нет, он добавляется к списку.

На рисунке желтым помечен сравниваемый элемент, а зеленым – уже найденные различные элементы.

цикл, который характеризуется двумя выходами.





управления не реализуется ни Циклом-пока, ни Счетным циклом, ни Циклом-до, т.е. реализуется неструктурно, с использованием операторов goto или break.

Преобразование неструктурной реализации в структурную показано на рисунке 4.

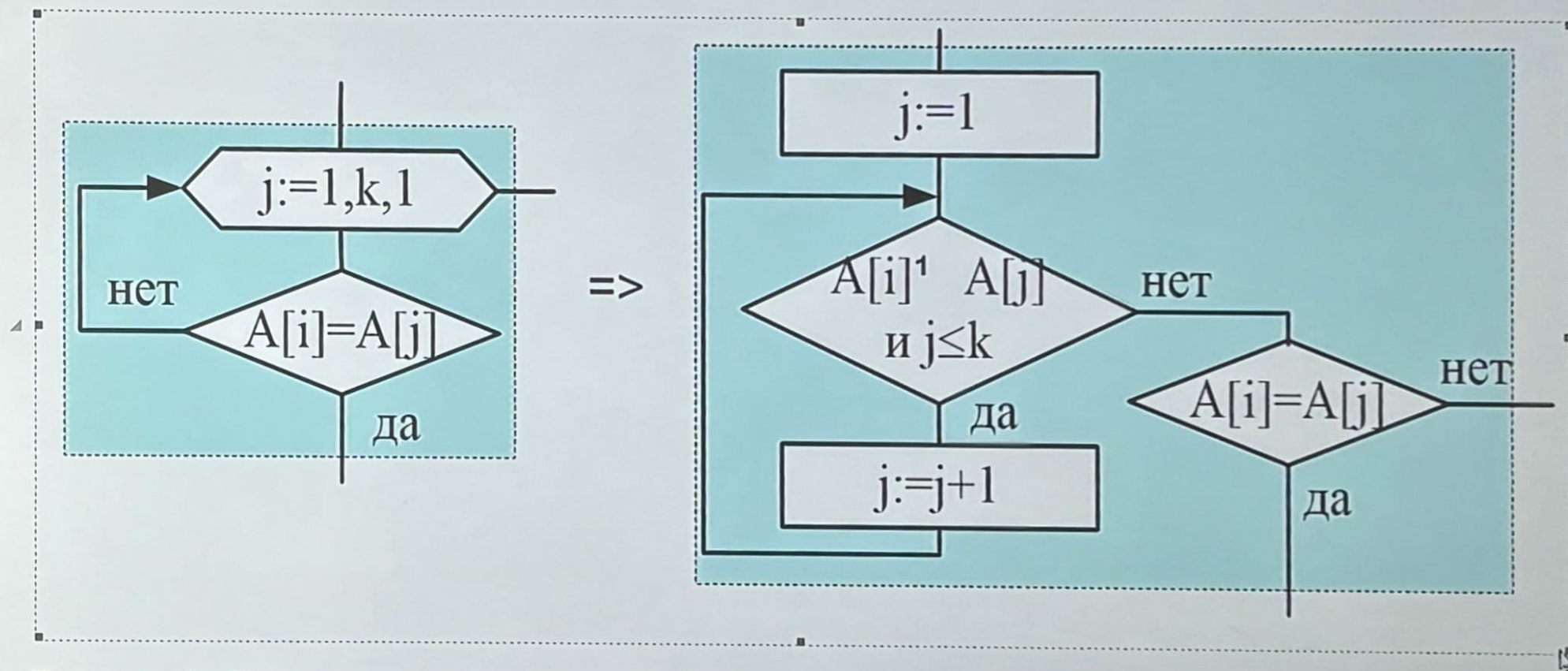
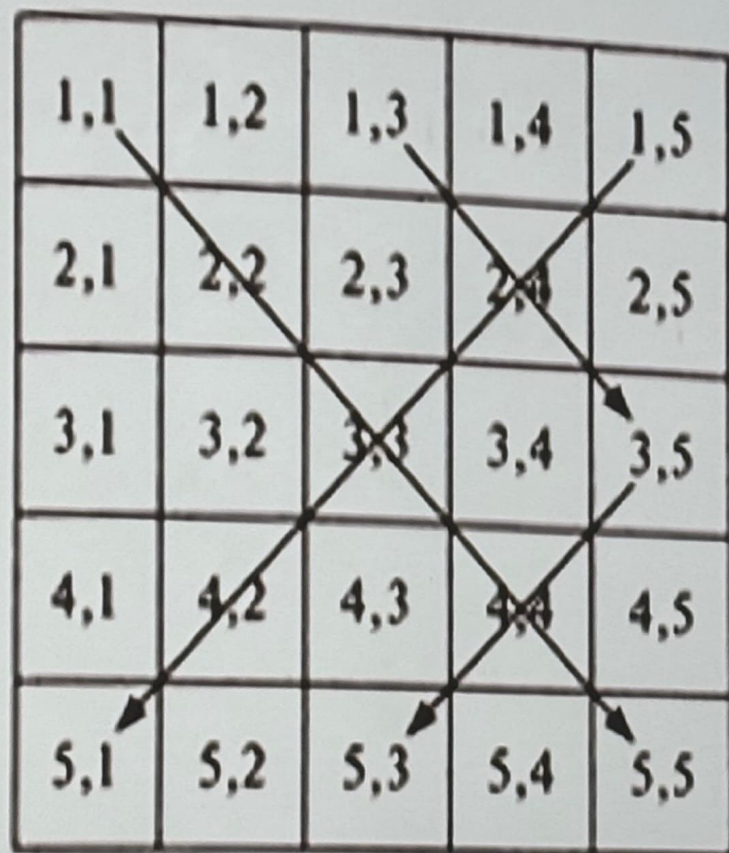


Рисунок 4 – Неструктурный и структурный варианты поискового цикла

**Задача 3.** Определить суммы элементов фрагментов, на которые матрицу делят диагонали, проведенные через элемент  $(k, l)$ .



1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,5

Побочная диагональ:  $i + j = n + 1$

Закономерность формирования индексов диагоналей, проходящих через элемент  $[p, k]$ :

а) параллельно главной:  $i - j = p - k$ ;

б) параллельно побочной:  $i + j = p + k$ .

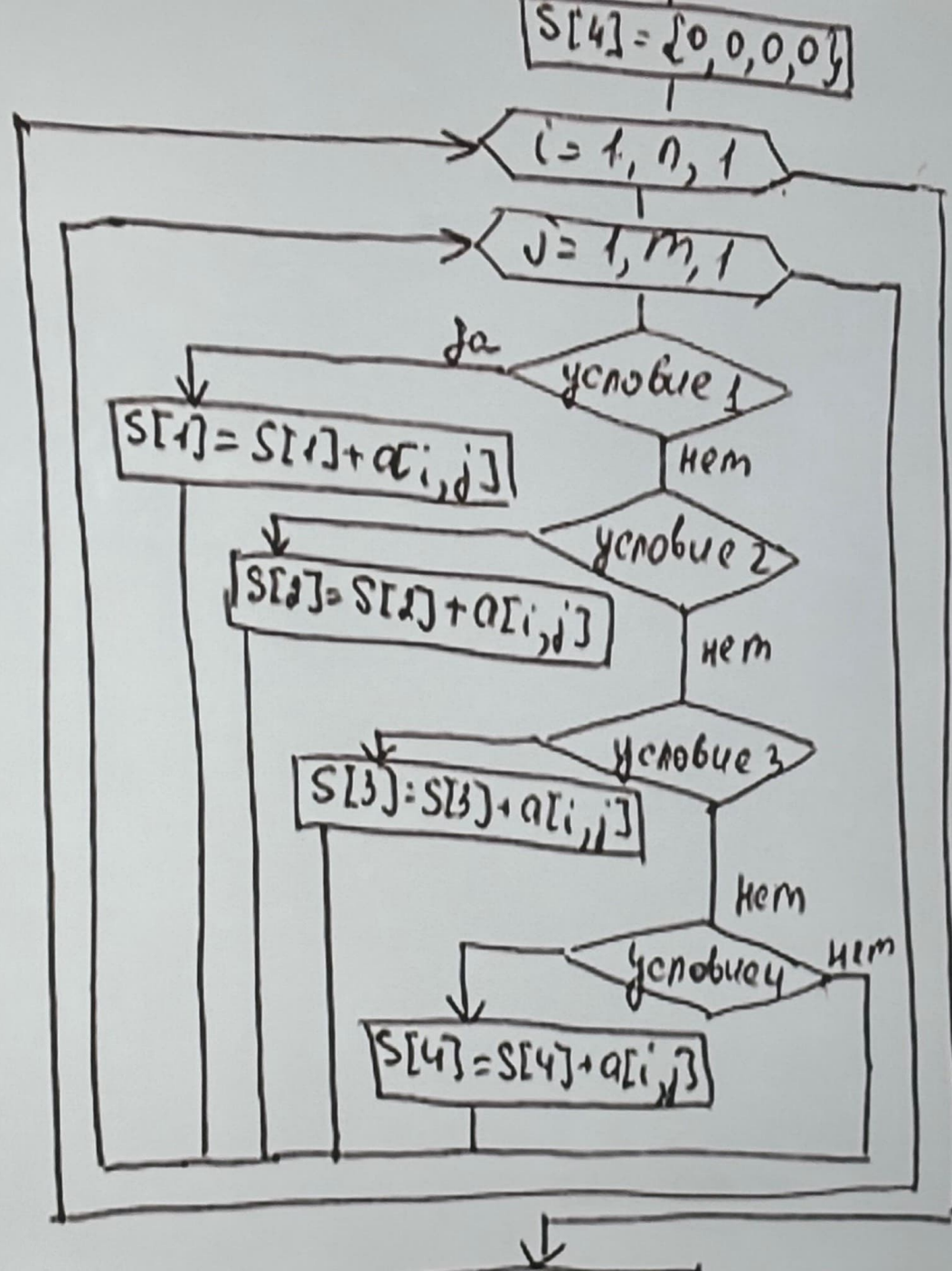
Количество элементов диагонали:

а) параллельной главной:  $s = n - |p - k|$ ;

б) параллельной побочной:  $s = n - |n + 1 - (p + k)|$ .

Главная диагональ:  $i = j$





Условие 1

$$(i+j) < (k+l)$$

$$\text{и } (i-j) < (k-l)$$

Условие 2

$$(i+j) < (k+l)$$

$$\text{и } (i-j) > (k-l)$$

Условие 3

$$(i+j) > (k+l)$$

$$\text{и } (i-j) < (k-l)$$

Условие 4

$$(i+j) > (k+l)$$

$$\text{и } (i-j) > (k-l)$$