Функции и параметры

Пример

Пример задачи

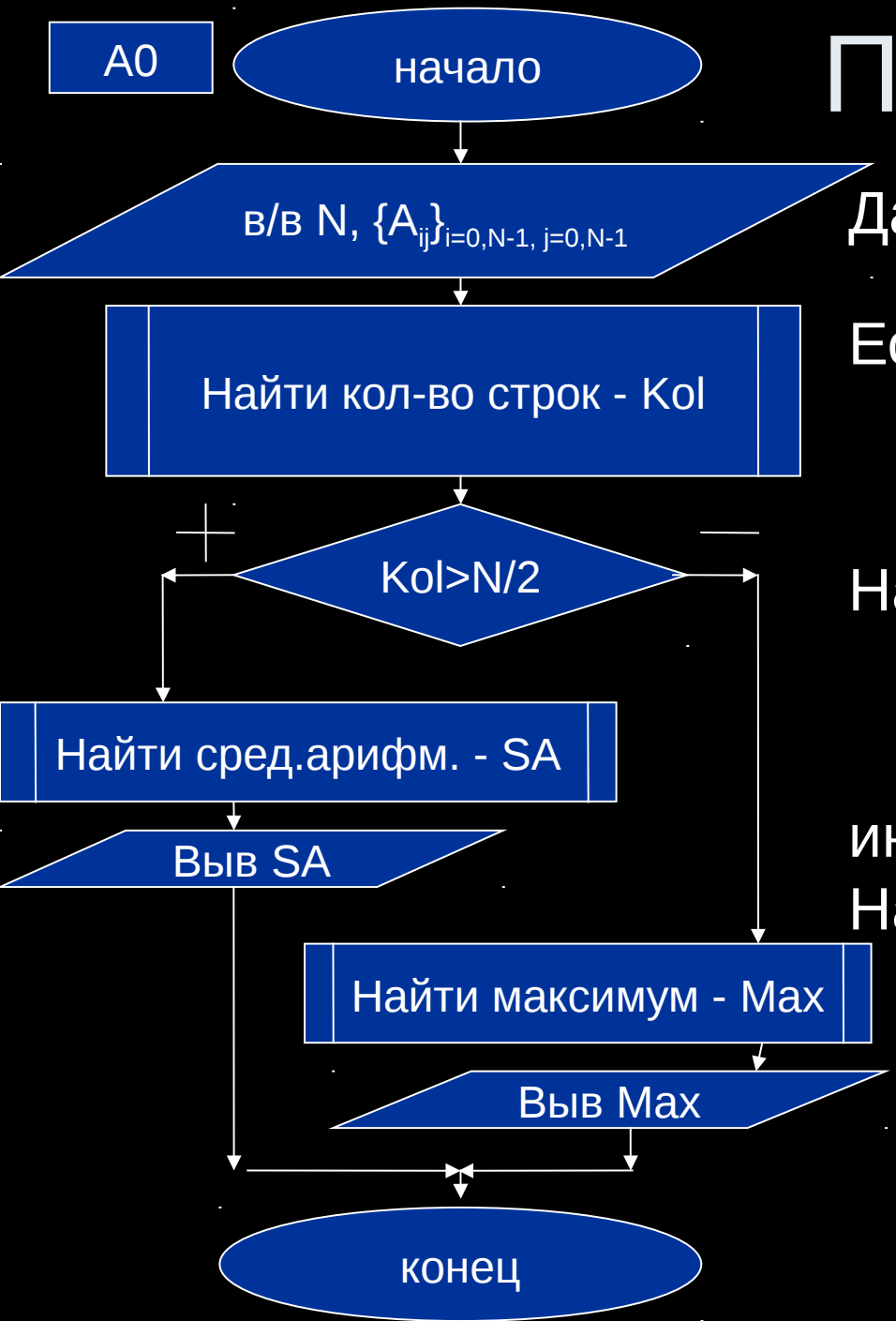
Дана квадратная матрица A размером N*N.

Если количество строк, в которых находятся только положительные элементы, больше N/2,

Найти среднее арифметическое значений всех положительных элементов,

иначе –

Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы **и на ней**.



Пример задачи

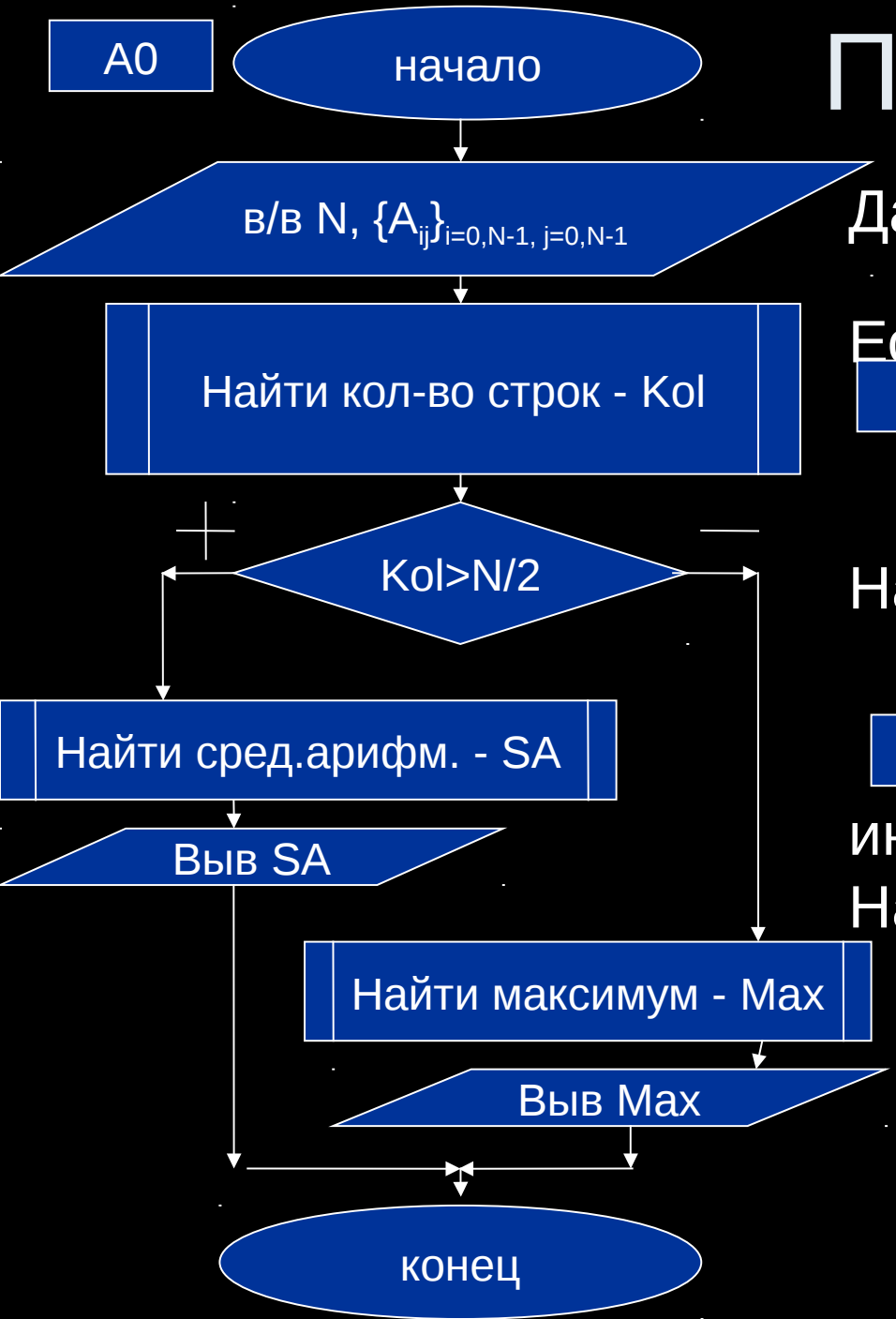
Дана квадратная матрица A размером N*N.

Если количество строк, в которых находятся только положительные элементы, больше N/2,

Найти среднее арифметическое значений всех положительных элементов,

иначе –

Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы **и на ней**.



Пример задачи

Дана квадратная матрица А размером N*N.

A0

Если количество строк, в которых находятся только положительные элементы больше N/2,

A0.1 A0.2 A0.3

Найти среднее арифметическое значений всех положительных элементов,

A0.1.1 - Проверка строки

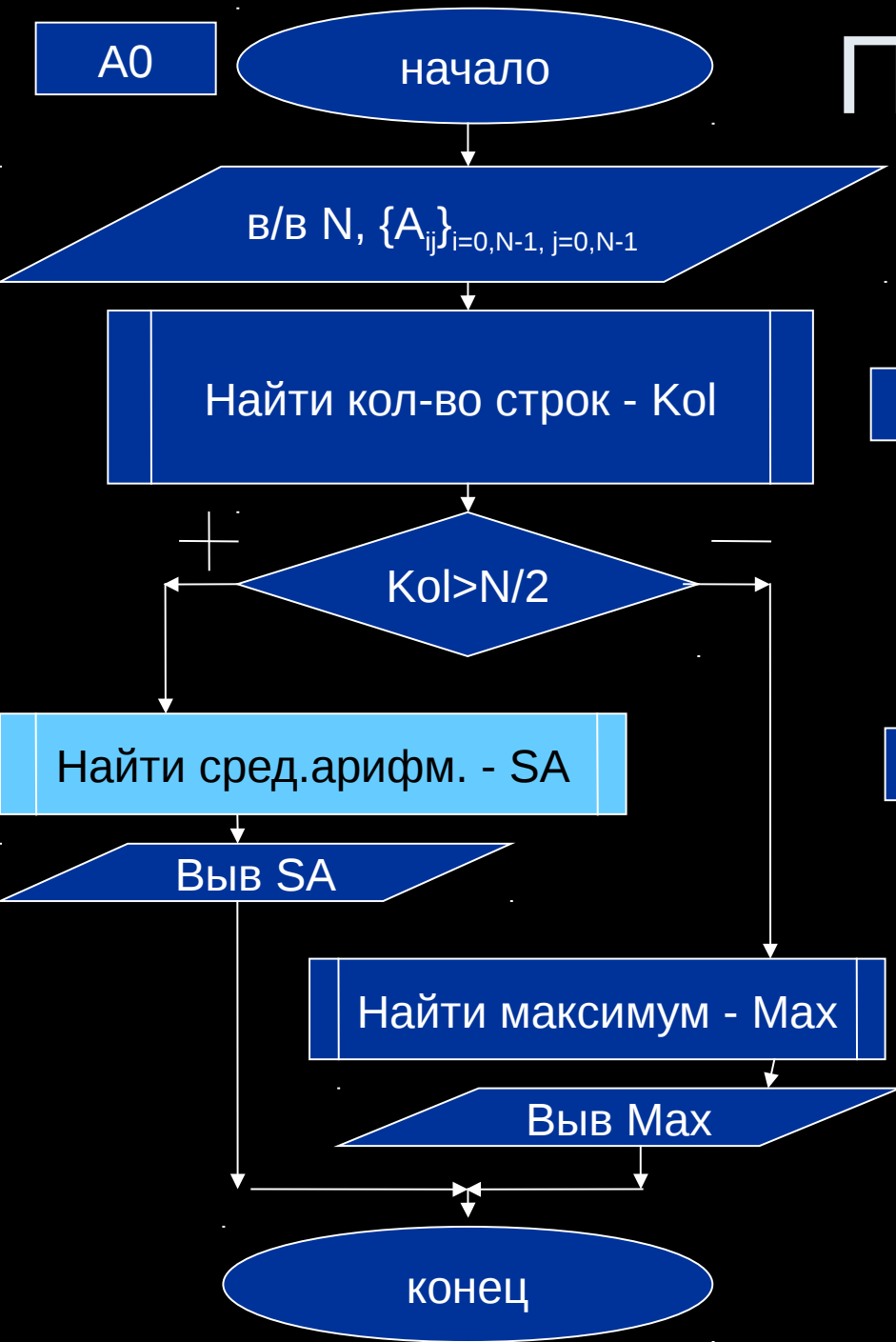
A0.2

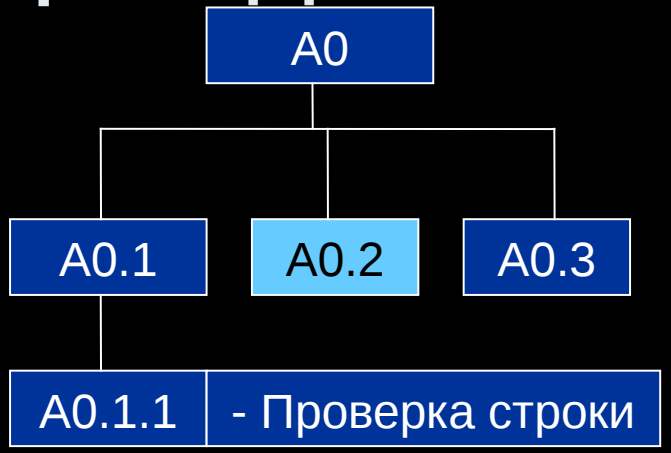
иначе –

Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы **и на ней**.

A0.3

Пример задачи





Дана квадратная матрица A размером N*N.

Если количество строк, в которых находятся только положительные элементы, больше N/2,

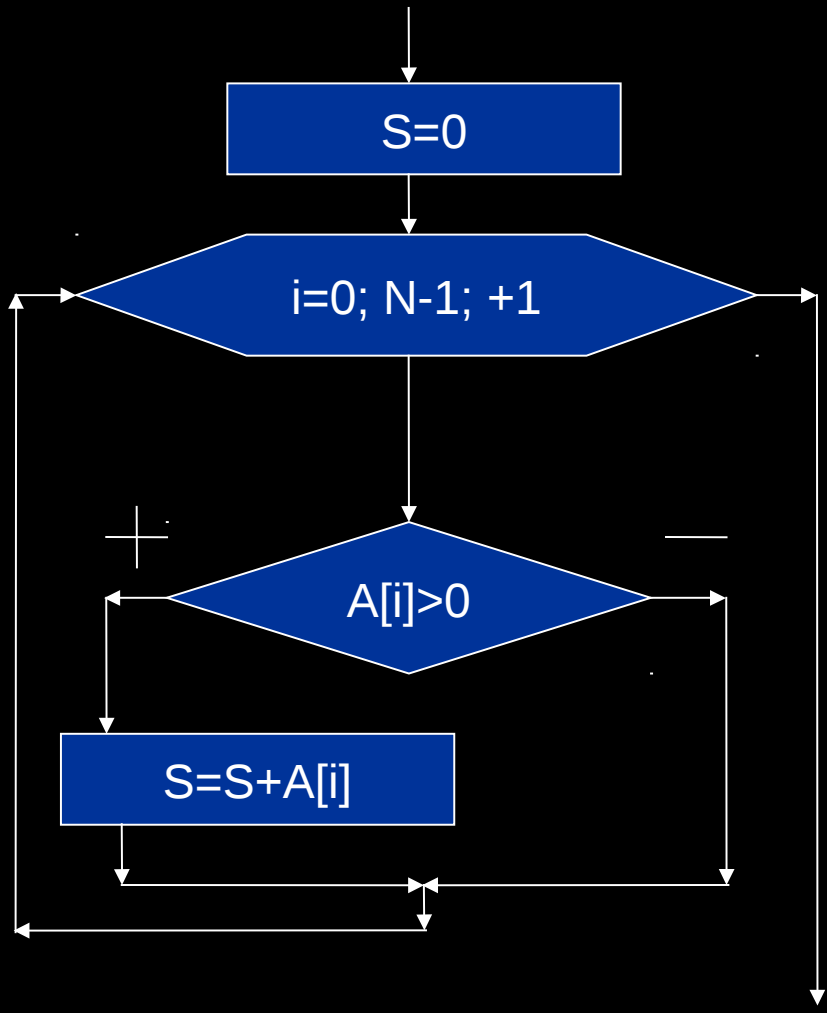
Найти среднее арифметическое значений всех положительных элементов,

иначе –

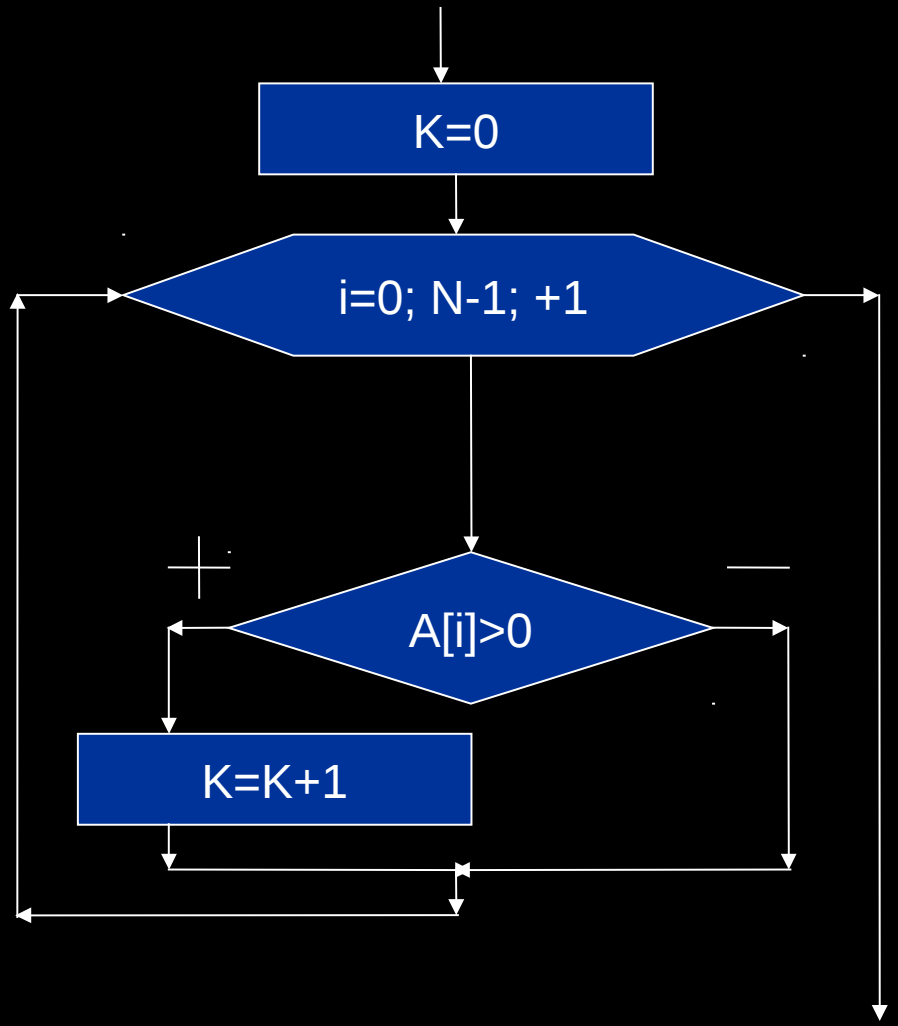
Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы **и на ней**.

A0.2 - Найти сред арифм Aij>0

Найти сумму

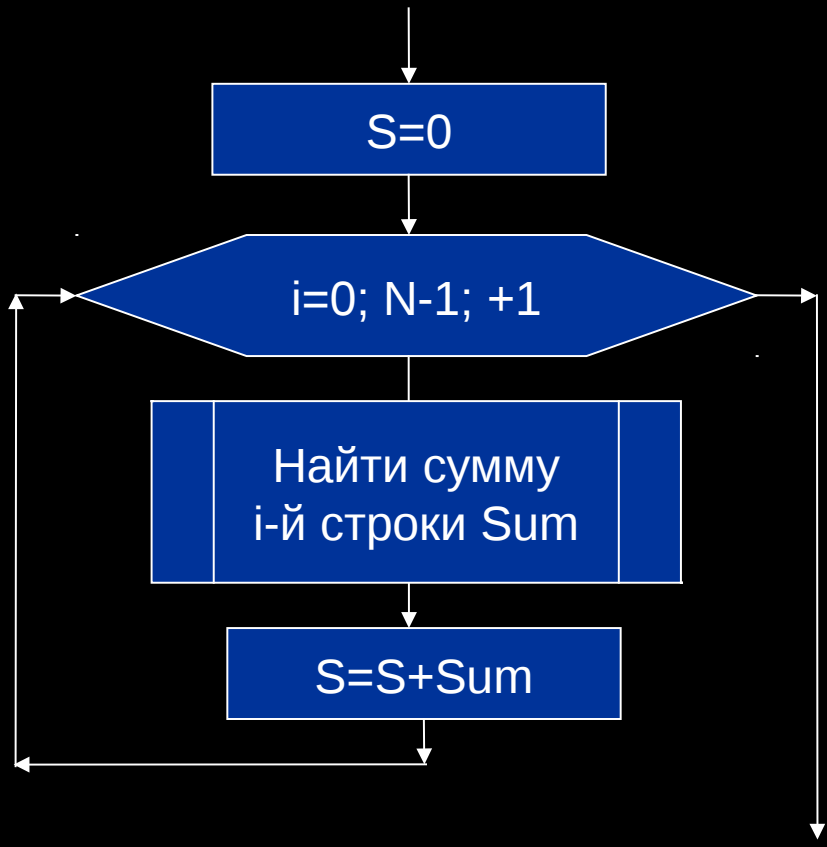


Найти количество

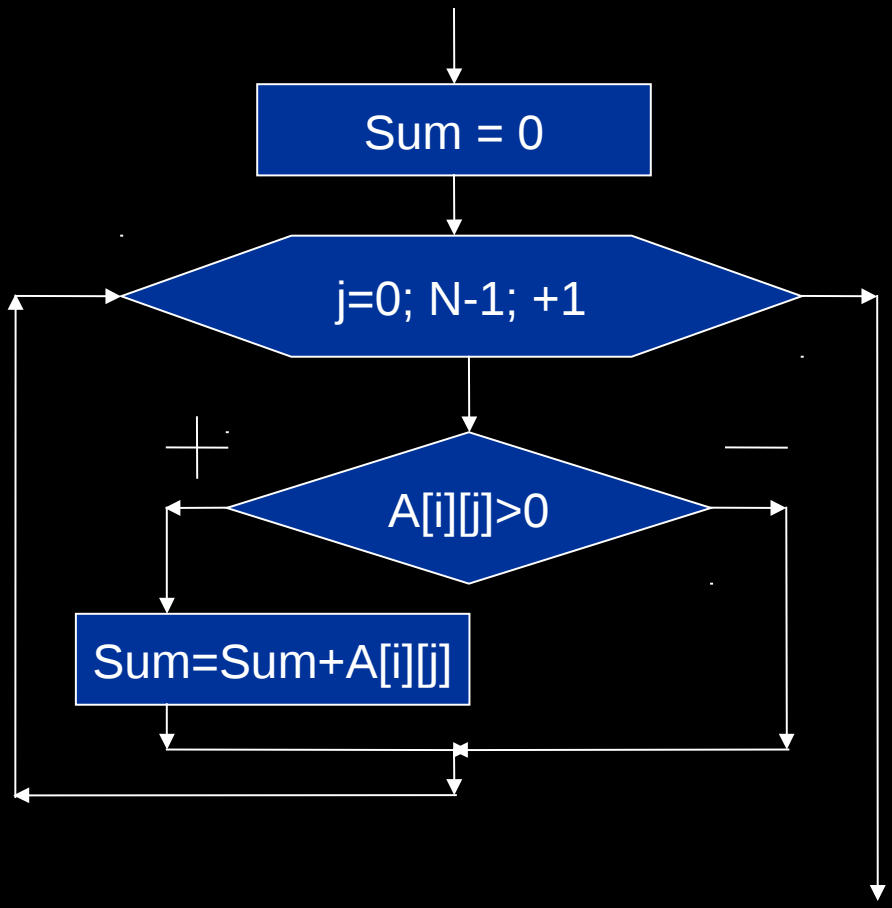


Из spkmmu.doc – для ОДНОМЕРНОГО массива

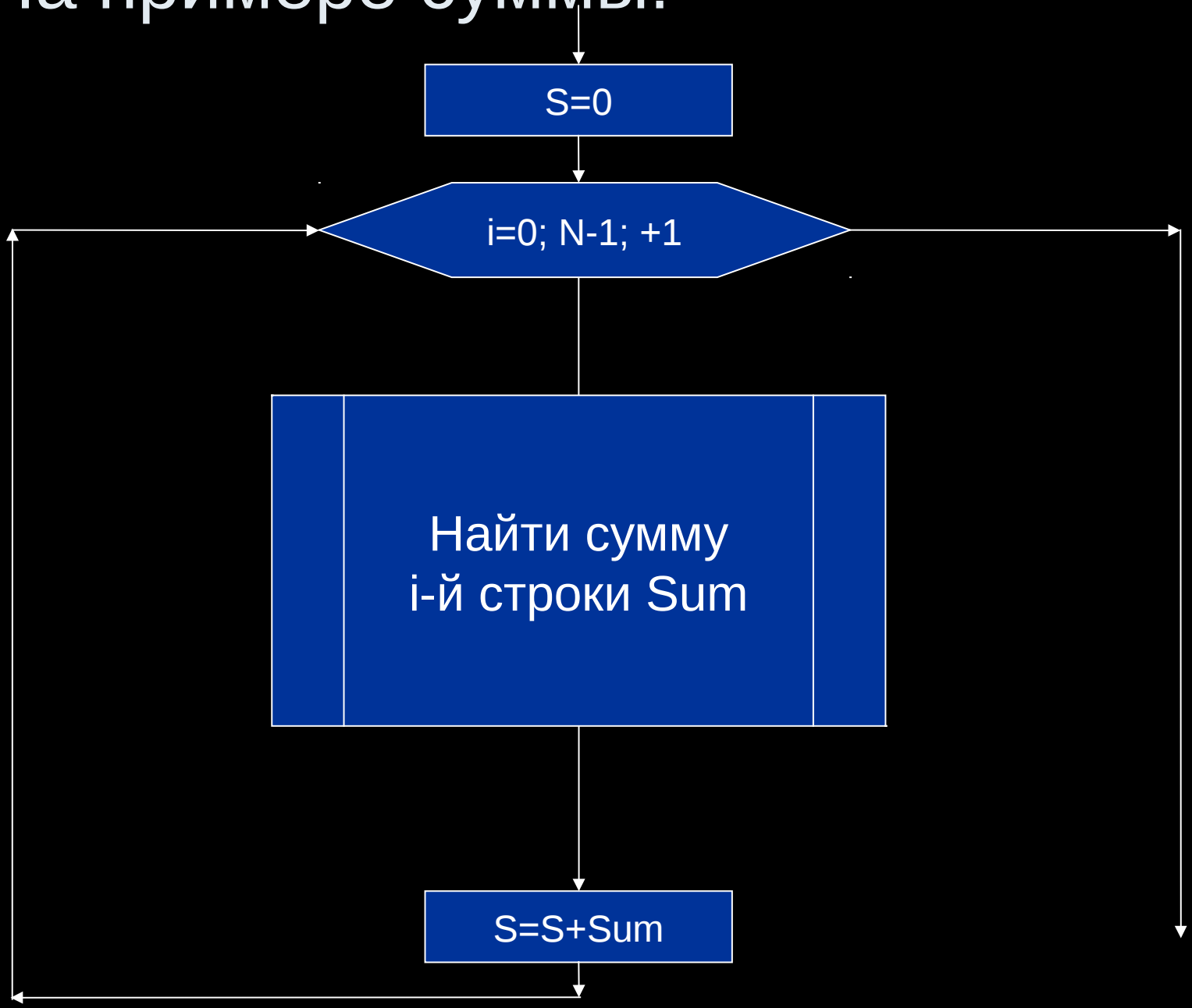
На примере суммы:



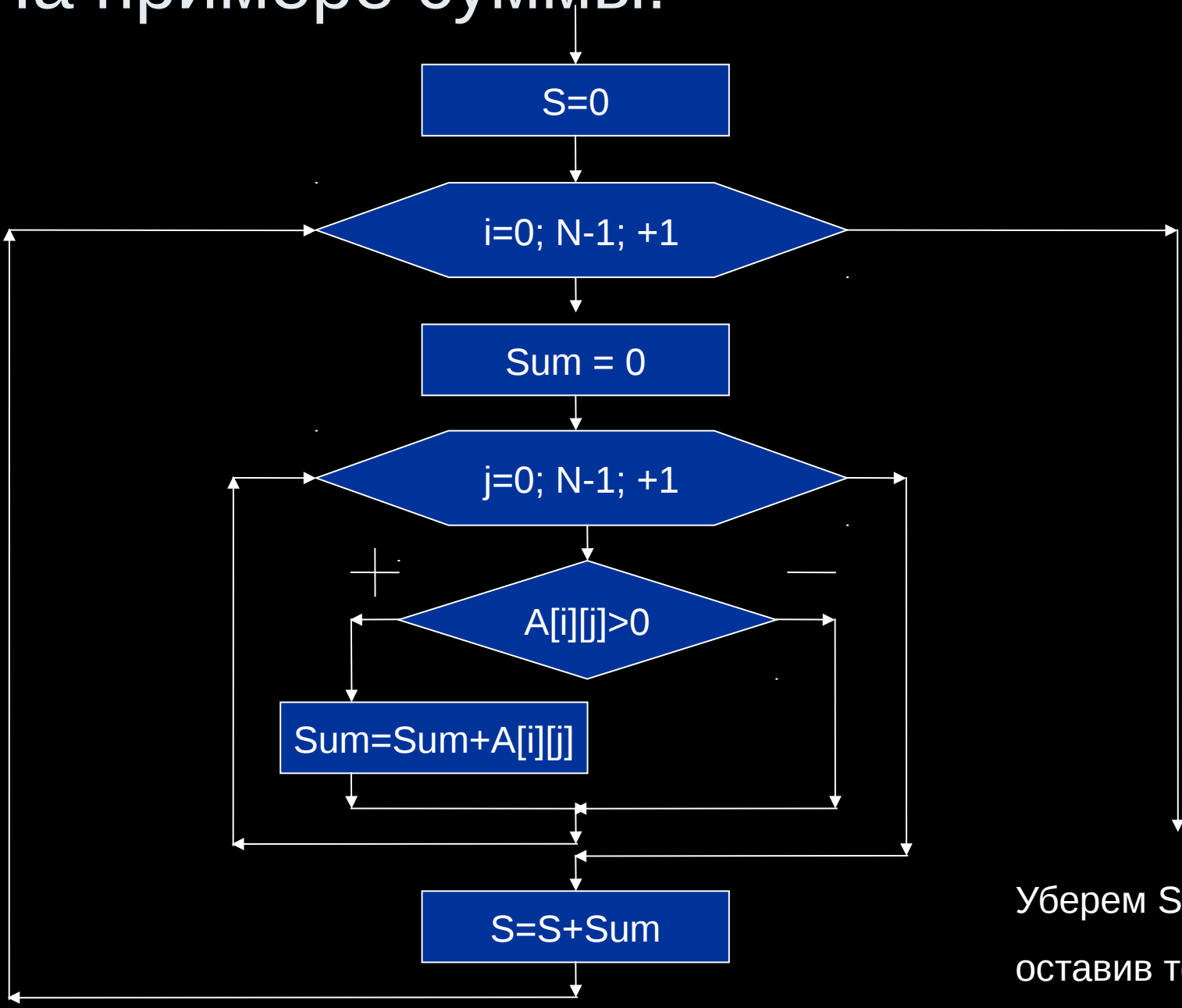
Сумма строки i



На примере суммы:



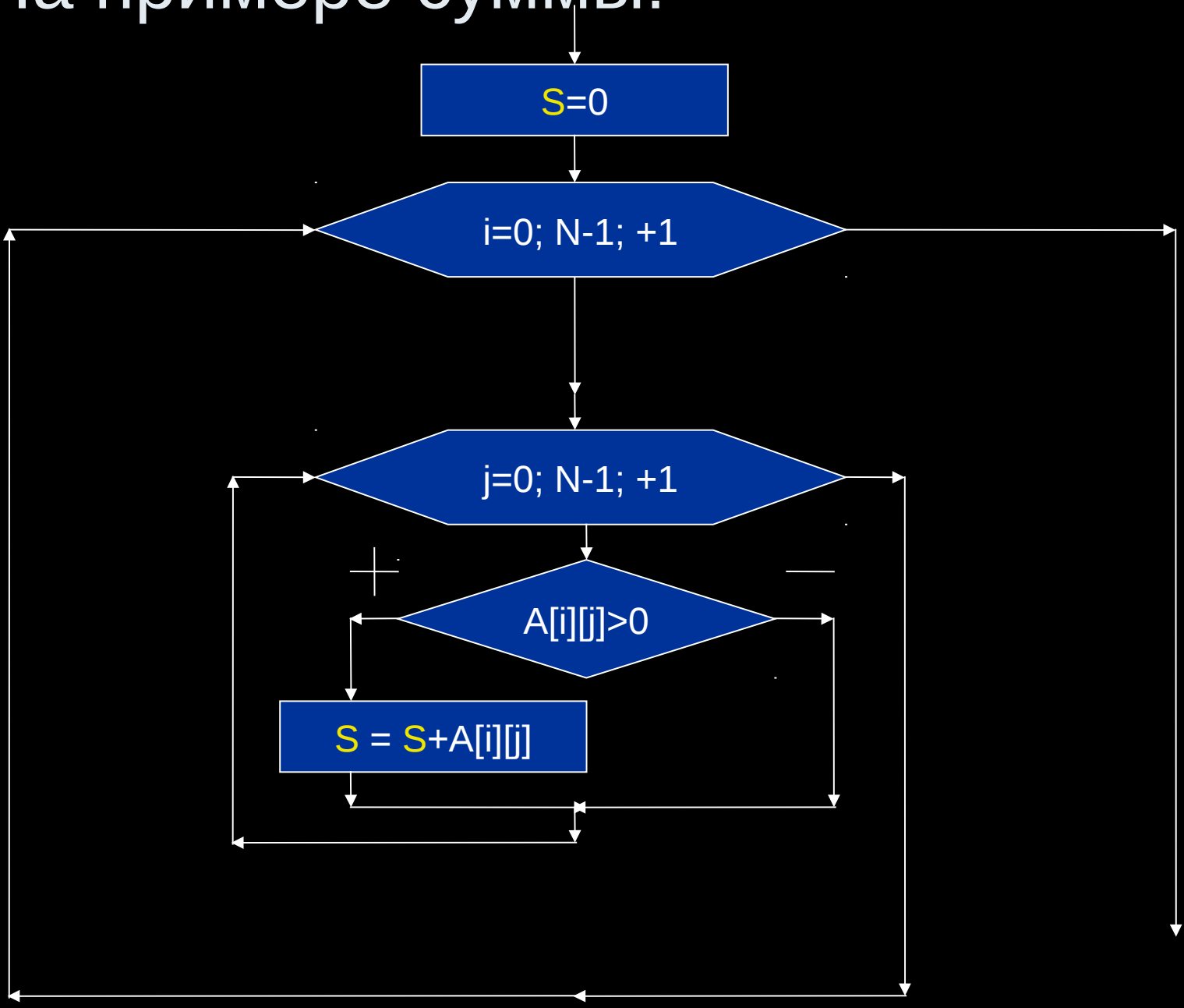
На примере суммы:



Уберем Sum,

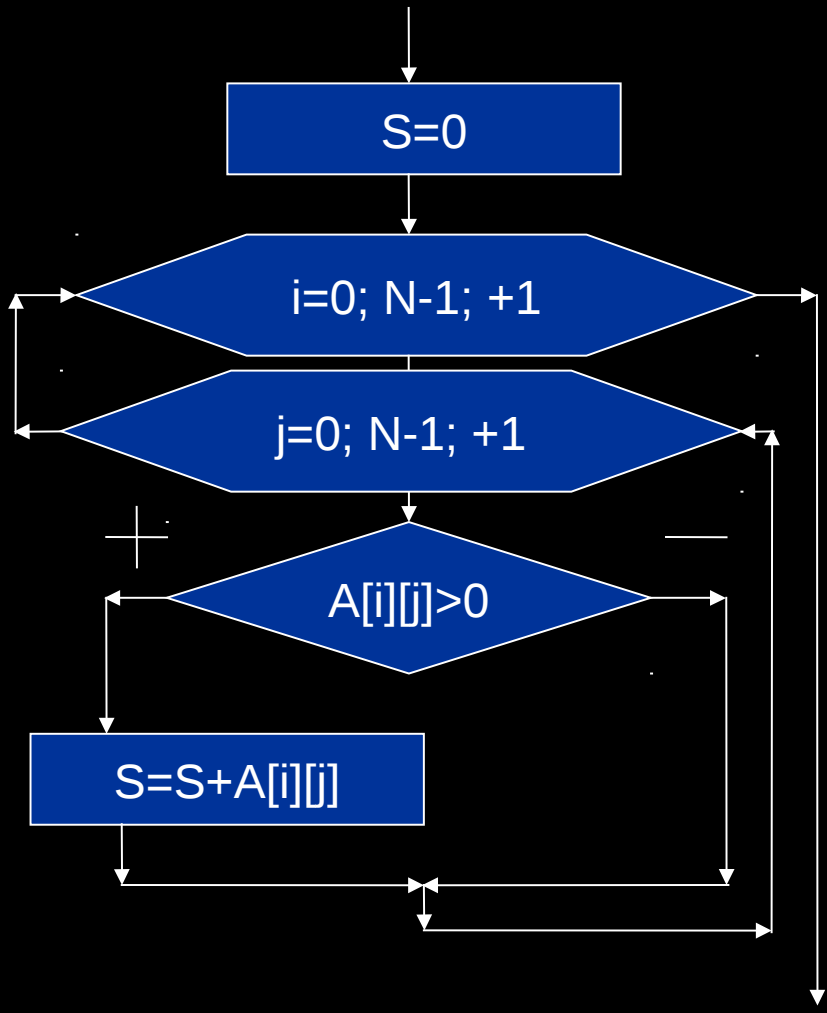
оставив только S

На примере суммы:

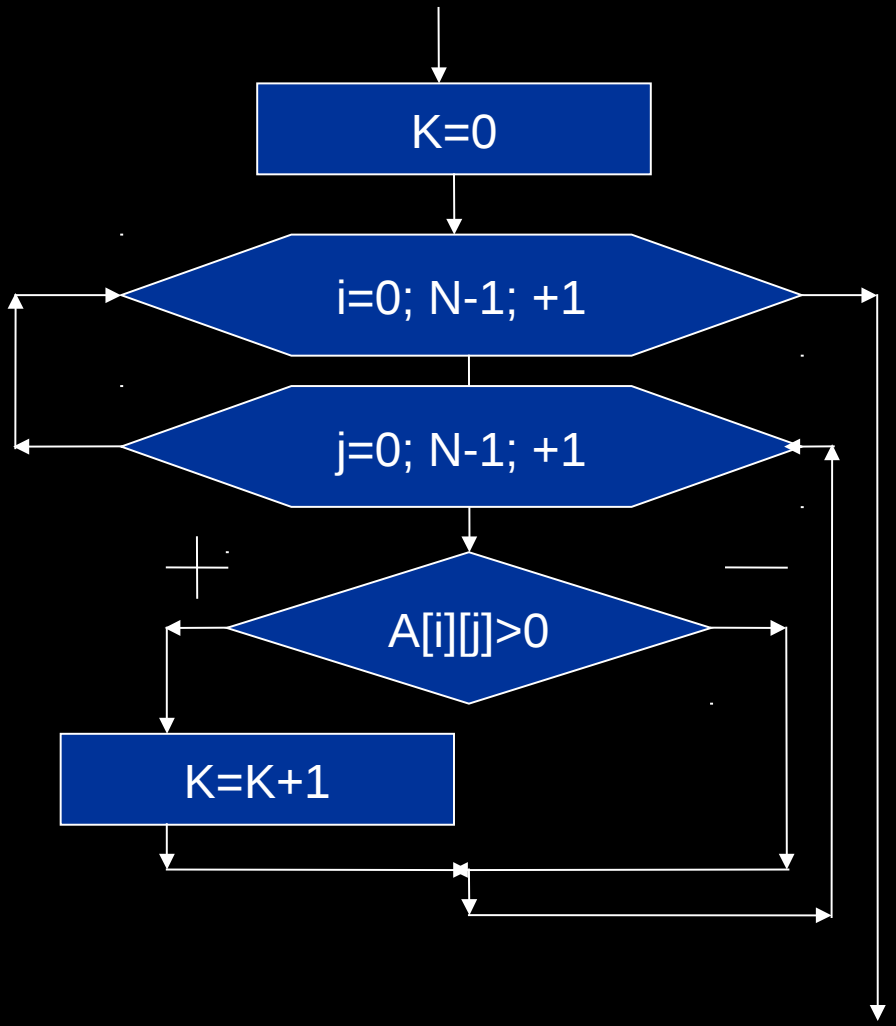


A0.2 - Найти сред арифм Aij>0

Найти сумму - S



Найти количество - K



Т.о. для матрицы – удваиваем циклы

A0.2 - Найти сред арифм Aij>0



Найти сумму S и

Найти количество K одновременно

K>0, т.к. строк с положительными элементами > N/2

A0.2 - Найти сред.арифм Aij>0

```
void FindSA( _______параметры_____________){
 _________локальные переменные_____________
   S=0;    K=0;
   for (int i=0; i< N;i++)
      for (int j=0; j<N; i++)
         if (A[i][j]>0){
            S=S+A[i][j];
            K=K+1;
         }
   SA=S/K;
}
```

	вх	пром	вых	
S	-	+	-	лок
K	-	+	-	лок
i	-	+	-	лок
N	+	-	-	По знач.
j	-	+	-	лок
A	+	-	-	По знач.
SA	-	+	+	По адресу

Параметры и локал.переменные

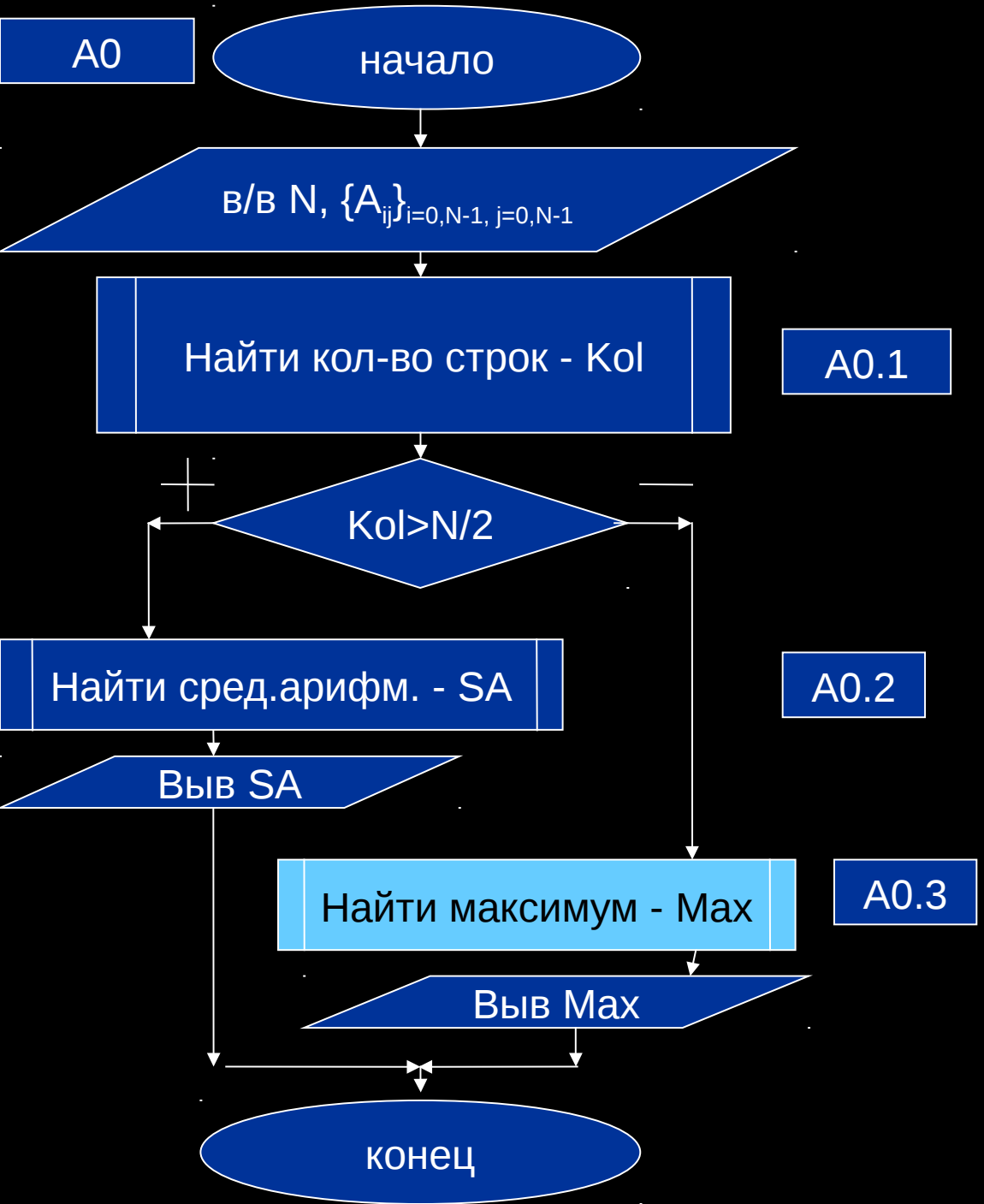
вх	пром	вых		
-	+	-	лок	0) Локальная переменная
+	+	-	парам	1) По умолчанию, передача по значению (создается копия)
+	+	+	парам	2) Передача по адресу/ссылке
+	-	-	парам	3) с const - Запрет на изменение, или как 1) (копия)
-	+	+	парам	4) Передача по адресу/ссылке Либо лок.перем..+результат

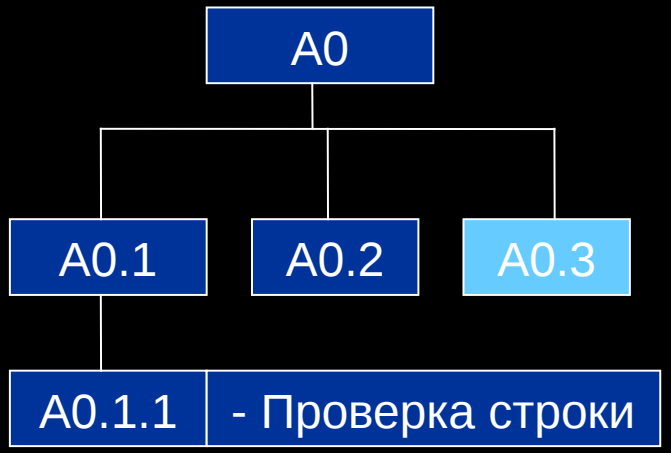
A0.2 - Найти сред.арифм Aij>0

```
#define Nmax 10
void FindSA(const int N, int A[][Nmax], double *SA){
 int S,K; //локальные переменные

   S=0;    K=0;
   for (int i=0; i< N;i++)
      for (int j=0; j<N; i++)
         if (A[i][j]>0){
            S=S+A[i][j];
            K=K+1;
         }
   *SA=(double)S/K;
}
```

	вх	пром	вых	
S	-	+	-	лок
K	-	+	-	лок
i	-	+	-	лок
N	+	-	-	const/-
j	-	+	-	лок
A	+	-	-	const/-





Дана квадратная матрица A размером N*N.

Если количество строк, в которых находятся только положительные элементы, больше N/2,

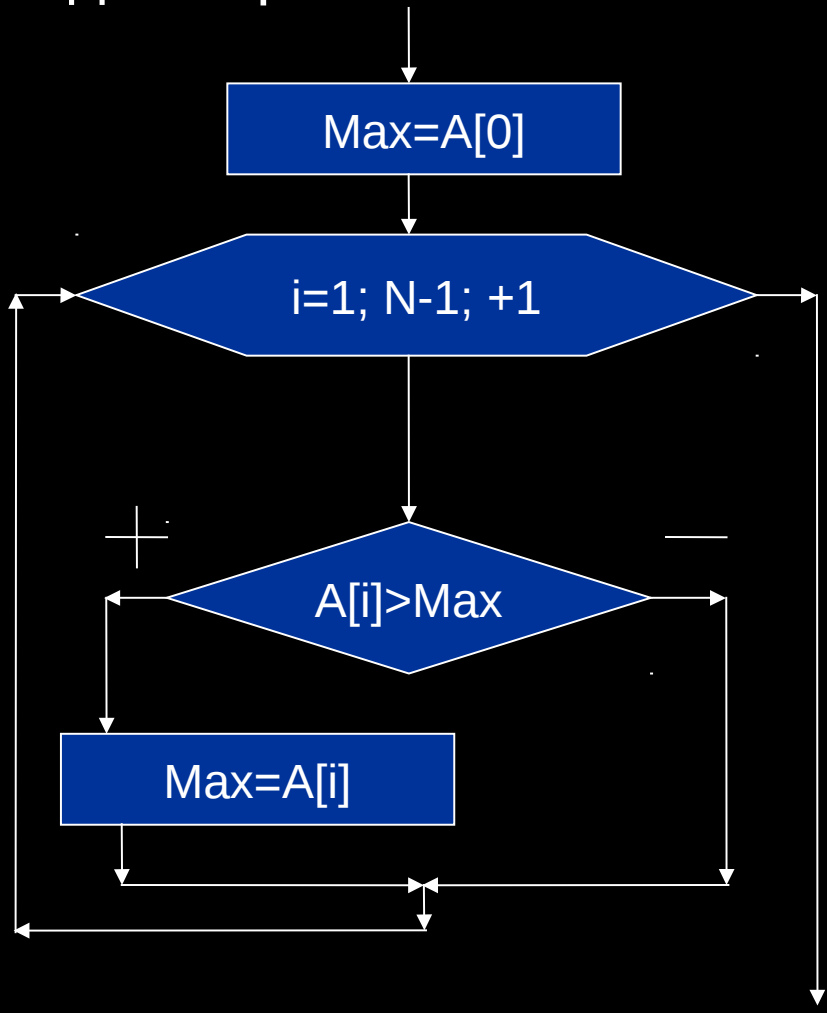
Найти среднее арифметическое значений всех положительных элементов,

иначе –

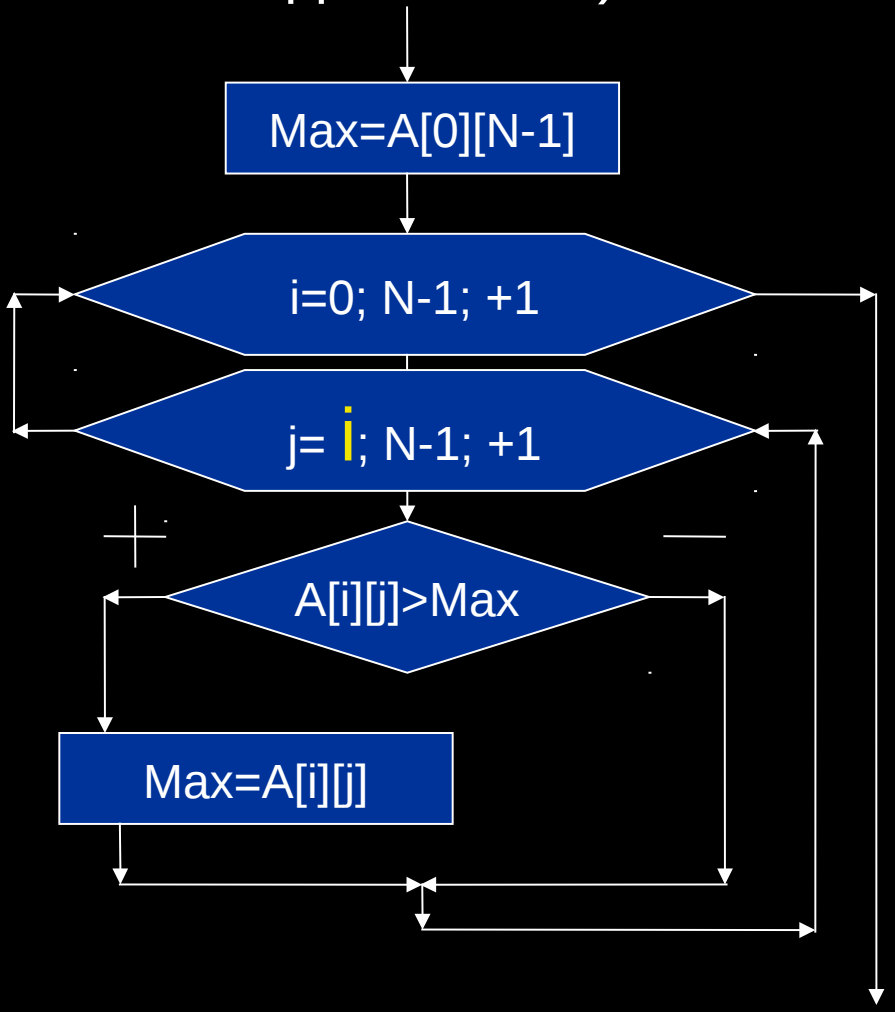
Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы и на ней.

A0.3 - Найти Max Aij | i<=j

Найти максимум в одномерном массиве:



Для матрицы (выше глав. диагонали):



Из spkmmu.doc

A0.3 - Найти Max Aij | i<=j

```
void FindMax( _______параметры_____________){
  _________локальные переменные_____________

   Max=A[0][N-1];
   for (int i=0;i<N;i++)
      for (int j=i; j<N;j++)
         if (A[i][j]>Max )
            Max=A[i][j];
}
```

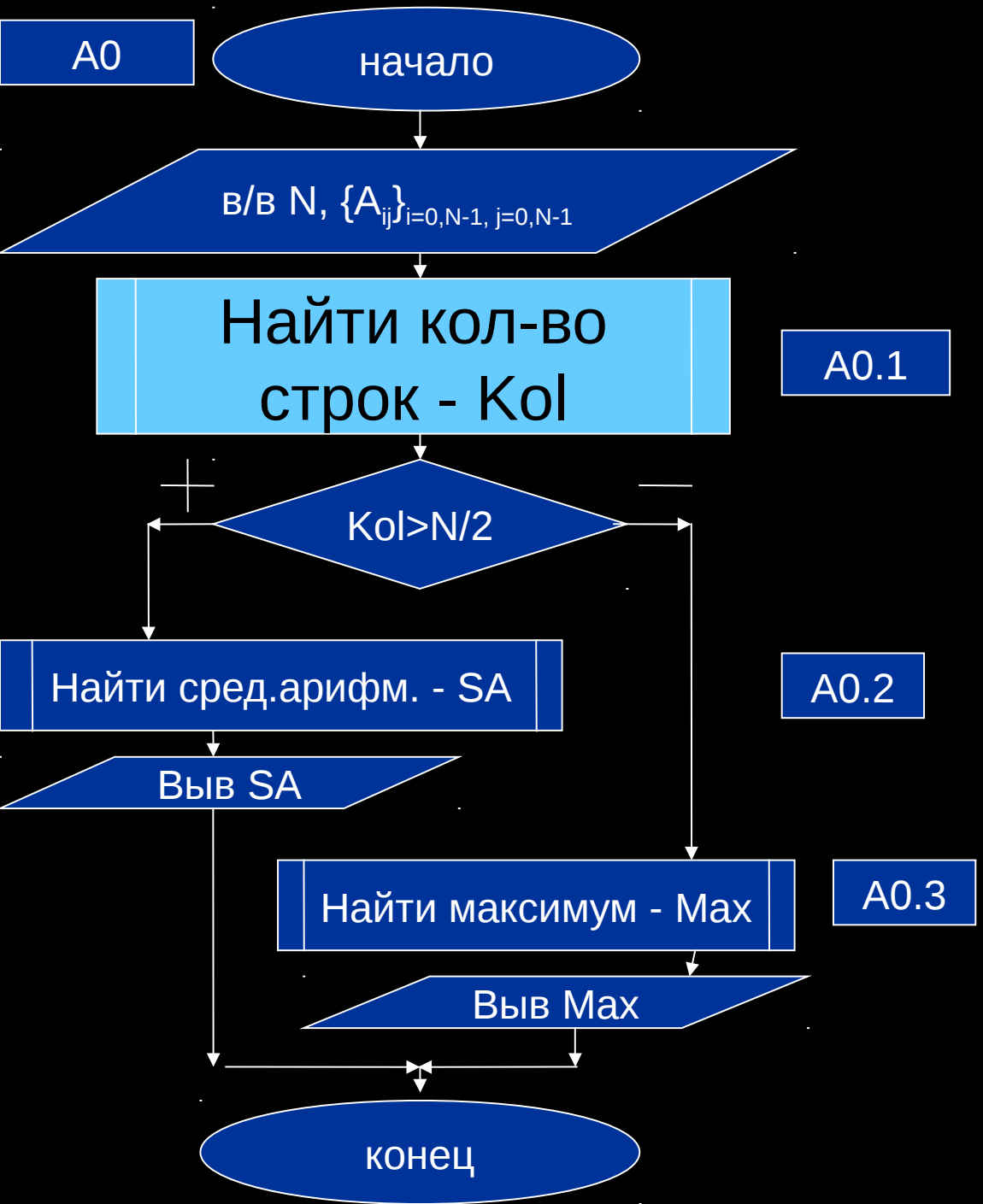
	вх	пром	вых	
Max	-	+	+	По адресу
i	-	+	-	лок
N	+	-	-	const/-
j	-	+	-	лок
A	+	-	-	const/-

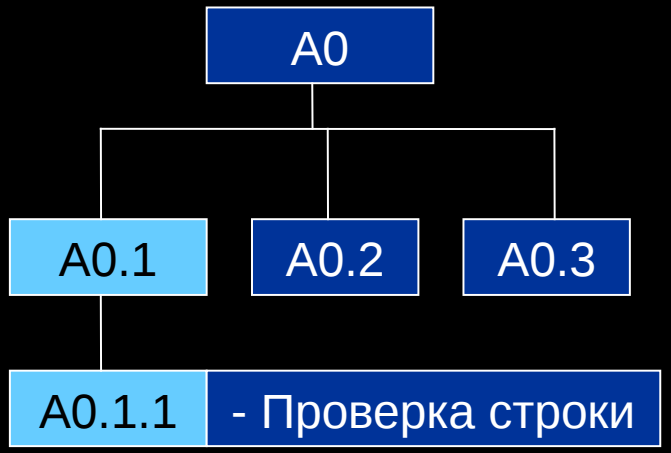
A0.3 - Найти Max Aij | i<=j

```
#define Nmax 10
void FindMax( const int N, const int A[][Nmax], int *Max){
  ///локальные переменные i,j

   *Max=A[0][N-1];
   for (int i=0;i<N;i++)
      for (int j=i; j<N;j++)
         if (A[i][j]>*Max )
            *Max=A[i][j];
}
```

	вх	пром	вых	
Max	-	+	+	По адресу
i	-	+	-	лок
N	+	-	-	const/-
j	-	+	-	лок
A	+	-	-	const/-





Дана квадратная матрица A размером N*N.

Если количество строк, в которых находятся только положительные элементы, больше N/2,

Найти среднее арифметическое значений всех положительных элементов,

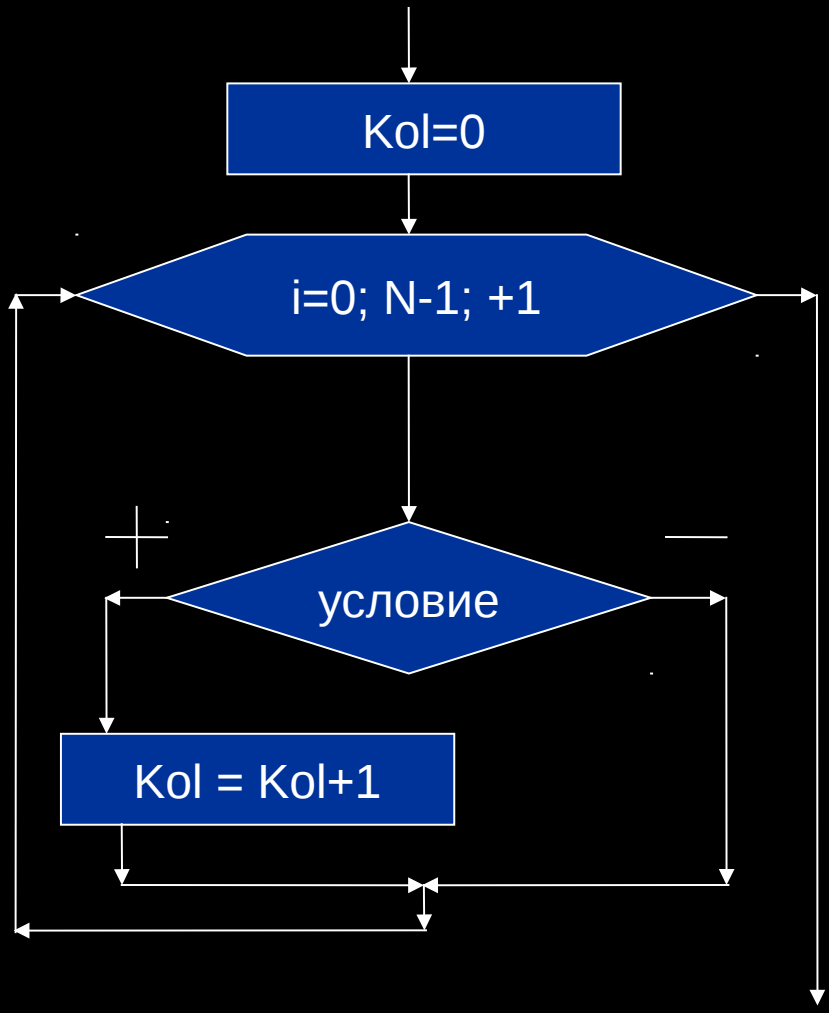
иначе –

Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы **и на ней**.

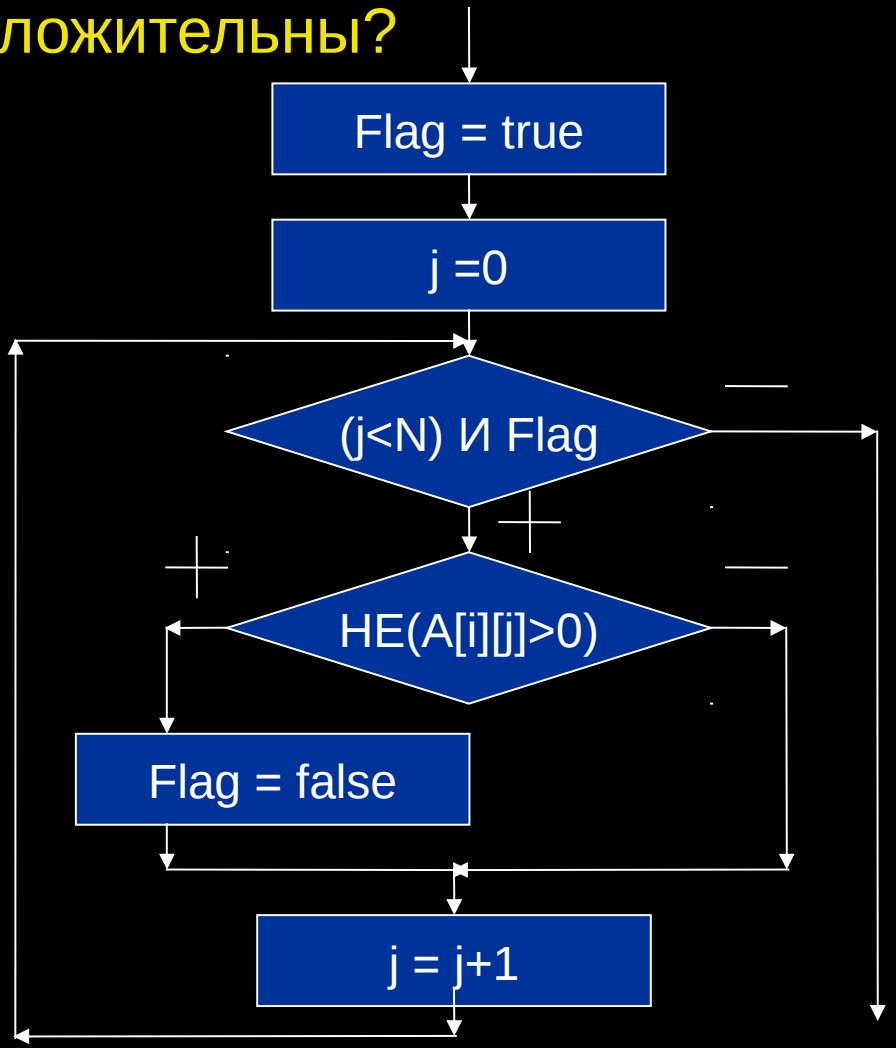
A0.1 - Найти количество строк,

в которых находятся только положительные элементы

Найти количество



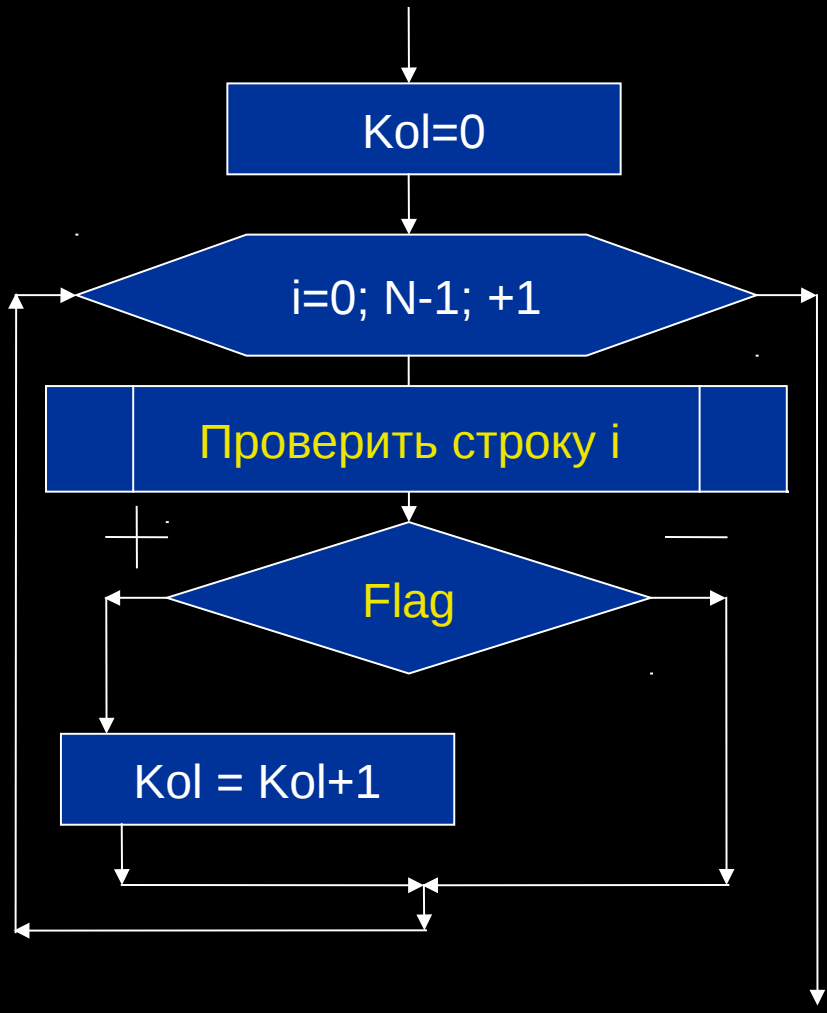
Проверить строку – ВСЕ ЛИ положительны?



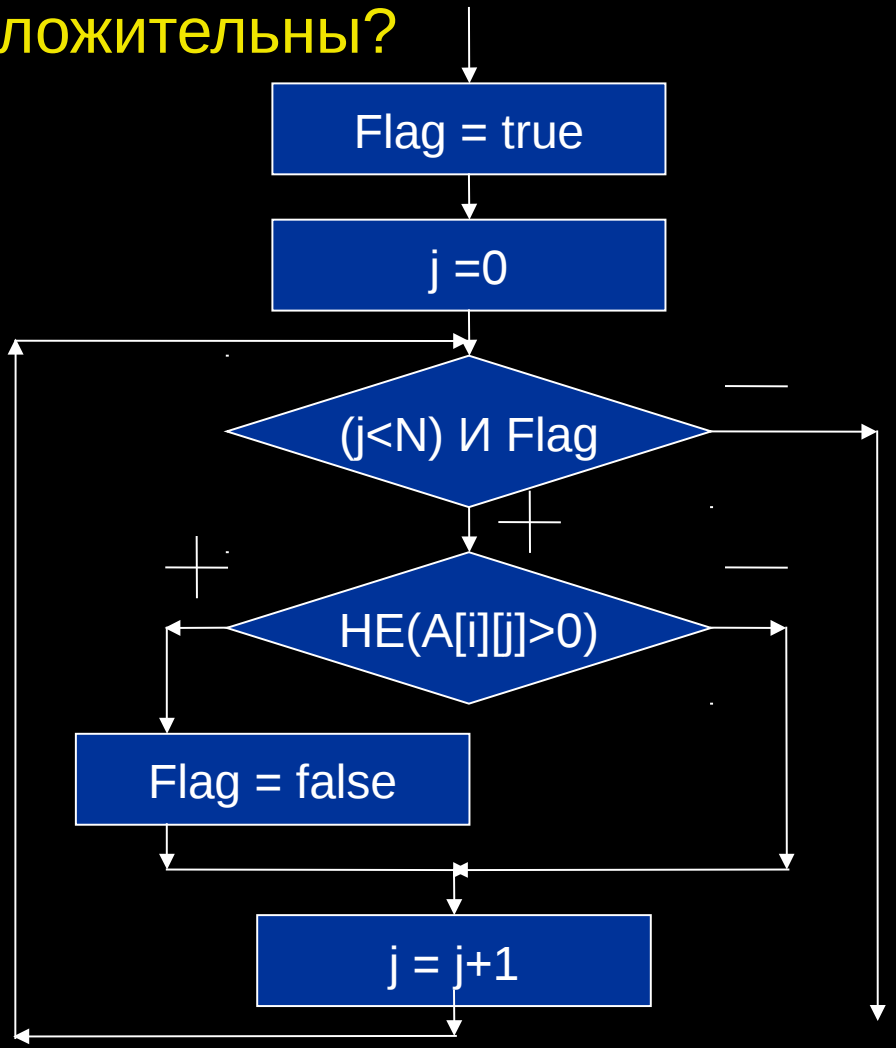
A0.1 - Найти количество строк,

в которых находятся только положительные элементы

Найти количество



Проверить строку – ВСЕ ЛИ положительны?



A0.1.1 – Проверить строку i

```
#define Nmax 10
void Check_i(const int N, const int i, int A[][Nmax],bool*Flag){
 int j;

   *Flag=true;    j=0;
   while (j<N && *Flag){

      if (! (A[i][j]>0) )
         *Flag= false;
      j++;
   }
}
```

	вх	пром	вых	
Flag	-	+	+	По адресу
i	+	-	-	const/-
N	+	-	-	const/-
j	-	+	-	лок

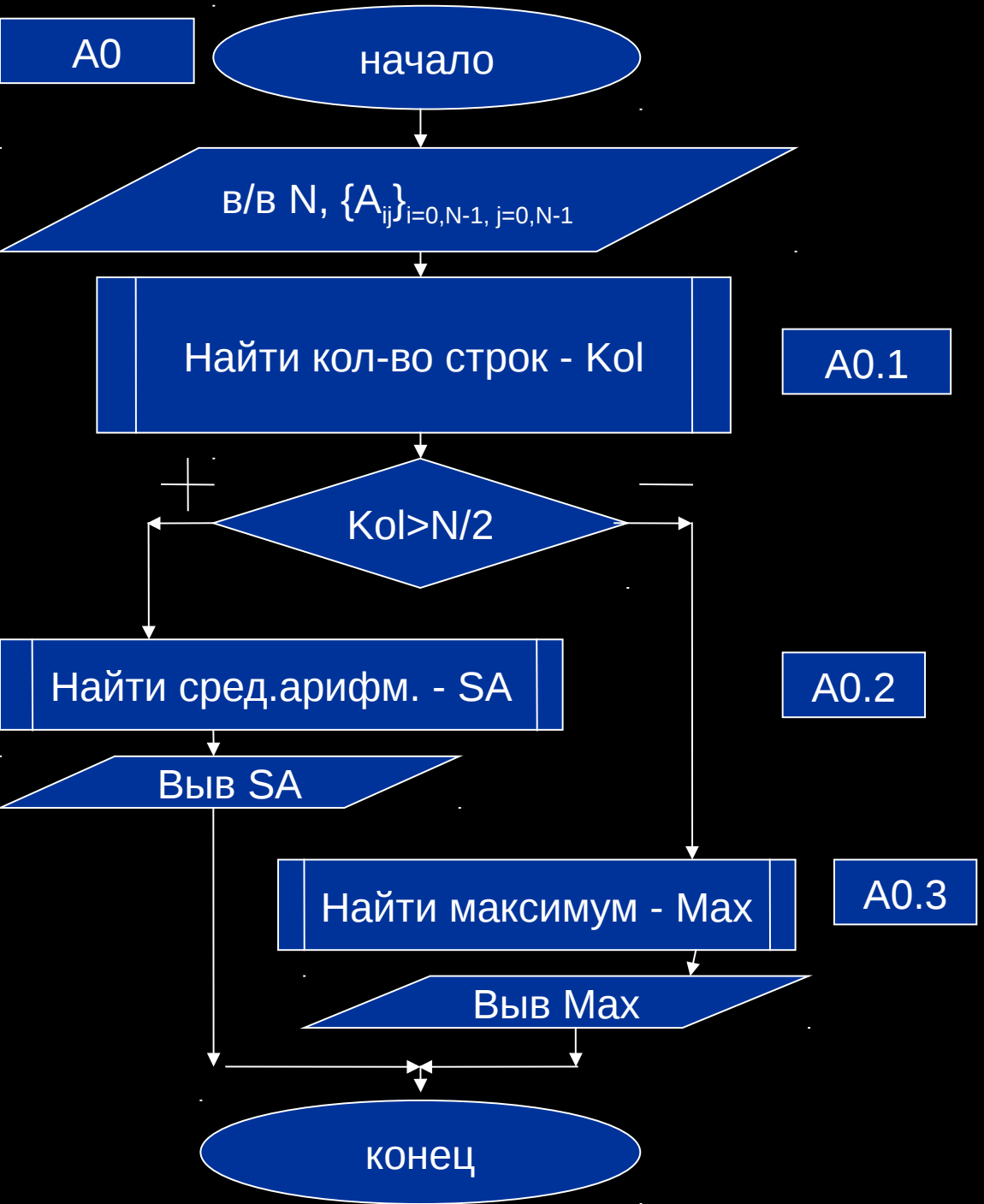
A0.1 – Найти количество строк

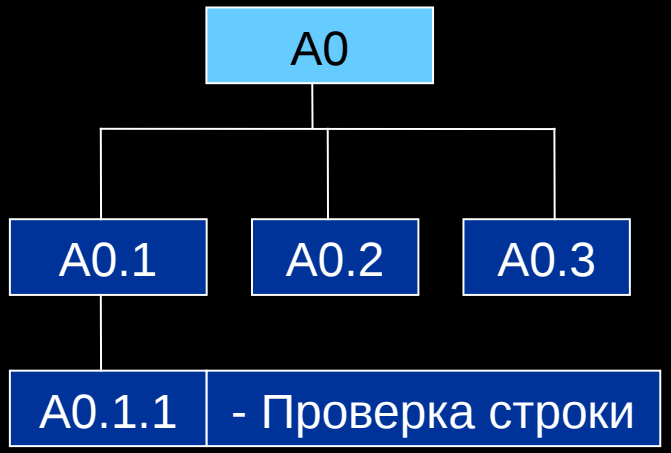
```
#define Nmax 10
void FindKol(  const int N, int A[][Nmax], int *Kol){
   int i; bool Flag;

   *Kol=0;
   for (i=0; i< N;i++){

      Check_i( N,i, A, &Flag);
      if (Flag)
         (*Kol)++;
   }
}
```

	вх	пром	вых	
Kol	-	+	+	По адресу
Flag	-	+	-	лок
i	-	+	-	лок
N	+	-	-	const/-
A	+			const/





Дана квадратная матрица A размером N*N.

Если количество строк, в которых находятся только положительные элементы, больше N/2,

Найти среднее арифметическое значений всех положительных элементов,

иначе –

Найти максимальное значение среди элементов, лежащих выше главной диагонали матрицы **и на ней**.

A0 – Основная программа

```
#include <...>

#define Nmax 10

void Check_i( const int N, const int i,  int A[][Nmax], bool *Flag){...}

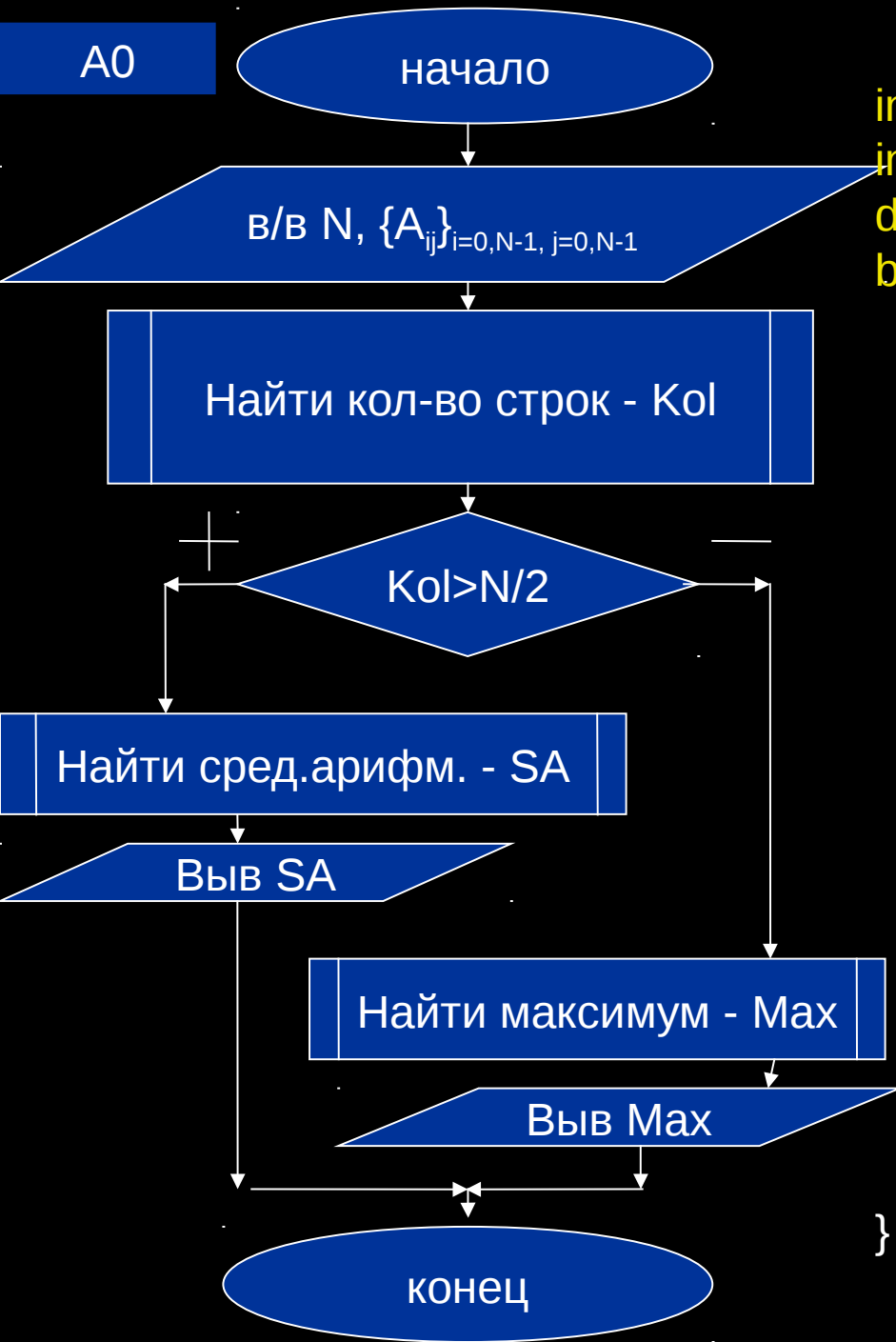
void FindKol( const int N, int A[][Nmax], int *Kol){...}

void FindMax(const int N, int A[][Nmax], int *Max){...}

void FindSA(const int N, int A[][Nmax], double *SA){...}


void VvodVyvodNA( int *N, int A[][Nmax]){...}


int main(void){
int N, A[Nmax][Nmax], Kol, Max; double SA; bool Flag; …
}
```




```
int main(void){
int N, A[Nmax][Nmax], Kol, Max;
double SA;
bool Flag;


   VvodVyvodNA( &N, A);
   FindKol( N, A, &Kol);
   if (Kol > N/2 ){

      FindSA( N, A, &SA);
      printf("Kol> N/2, Sa=%7.3lf", SA);
    }
   else{

      FindMax( N, A, &Max);
      printf("Kol<=N/2, Max=%d',Max);
    }

  return 0;

}
```