

**Типовой расчет «Процедуры, функции, модули»**

**Вариант 0**

*Задание выполнил: Фамилия И.О.*

*Студент группы А-00-21*

*Проверил:* \_\_\_\_\_

*Оценка:* \_\_\_\_\_

*Замечания:* \_\_\_\_\_

---

---

---

---

---

---

---

---

---

---

## Пример решения задачи

## Абстракция А0.

**1.Условие.** Если на главной диагонали матрицы нет элементов, кратных пяти, заменить на нули все некрatные пяти элементы матрицы, в противном случае – найти число отрицательных элементов в каждом столбце матрицы и столбец с максимальным их числом.

«Кратно пяти»  $\rightarrow$  целочисленная

«Главная диагональ»  $\rightarrow$  квадратная

## 2.Уточненная постановка задачи.

Дана целочисленная квадратная матрица  $A$  из  $n$  строк и  $n$  столбцов. Если на главной диагонали матрицы нет ( $flag=False$ ) элементов, кратных пяти, заменить на нули все некрatные пяти элементы матрицы  $A$ , в противном случае ( $flag=True$ ) – найти количество отрицательных элементов ( $Kolich$ ) в каждом столбце матрицы и указать первый столбец ( $Nom$ ) с максимальным их числом.

## 3.Примеры.

Пример 1. Нет элементов, кратных 5, на гл.диагонали матрицы

$N = 3$

$$A = \begin{pmatrix} 1 & 5 & 4 \\ 1 & 2 & 0 \\ 10 & -5 & -3 \end{pmatrix}$$

Изменяем элементы матрицы

$$A = \begin{pmatrix} 0 & 5 & 0 \\ 0 & 0 & 0 \\ 10 & -5 & 0 \end{pmatrix}$$

Пример 2. Есть элементы, кратные 5, на гл.диагонали матрицы

$N = 3$

$$A = \begin{pmatrix} 1 & -5 & 4 \\ 1 & 10 & 0 \\ 10 & -5 & -3 \end{pmatrix}$$

Второй элемент главной диагонали (10) кратен 5, ищем количество отрицательных элементов в каждом столбце.

$Kolich = [2 \ 1]$

Максимальное число элементов во втором столбце

$Nom = 2$

## 4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура	Формат
Входные данные	A	Заданная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)	+XXX (:4)
	n	число строк и столбцов в матрице A, $0 < n \leq 5$	цел	простая переменная	X (:1)
Выходные данные	A	Измененная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)	+XXX (:4)
	Kolich	Количество отрицательных элементов в каждом столбце, $0 \leq kolich_i \leq 5$	цел	Одномерный массив (5)	X (:1)
	Nom	Номер первого столбца с максимальным количеством отрицательных элементов, $0 < Nom \leq 5$	цел	простая переменная	X (:1)

Промежуточные данные	dat	Файл с исходными данными с именем вида unit_dat#.txt	Текст.	файл	---
	res	Файл с отчетом с именем вида unit_res#.txt	Текст.	файл	---
	flag	= True, если есть на гл.диагонали есть элемент, кратный пяти, в противном случае False	лог	простая переменная	---
	ii	Строка с некорр. A[ <i>ii</i> , <i>jj</i> ], 0< <i>ii</i> ≤ 5	цел	простая	XX (:2)
	jj	Столбец с некорр. A[ <i>ii</i> , <i>jj</i> ], 0< <i>jj</i> ≤ 5	цел	простая	XX (:2)
	flagA	= True, если есть некорректные A <sub>ij</sub> , в противном случае False	лог	простая переменная	---

**5.Входная форма.** (Файлы unit\_dat<№ теста>.txt)

```

Обр1  <n>
      <a[1,1]> <a[1,2]> . . . <a[1,n]>
Обр2  <a[2,1]> <a[2,2]> . . . <a[2,n]>
      . . . . .
      <a[n,1]> <a[n,2]> . . . <a[n,n]>
    
```

**6.Выходная форма** (Файлы unit\_res<№ теста>.txt)

15 пробелов перед текстом

```

Обр3  Программа с модулем
Обр4  Исходная матрица из <n> строк и <n> столбцов:
      <a[1,1]> <a[1,2]> . . . <a[1,n]>
Обр5  <a[2,1]> <a[2,2]> . . . <a[2,n]>
      . . . . .
      <a[n,1]> <a[n,2]> . . . <a[n,n]>
Обр6  Некорректное число строк.
      Задача не решалась
Обр7  Некорректный элемент матрицы
      A[<ii>, <jj>] = <A[ii,jj]>
      Задача не решалась
Обр8.1 Измененная матрица
      <a[1,1]> <a[1,2]> . . . <a[1,n]>
Обр8.2 <a[2,1]> <a[2,2]> . . . <a[2,n]>
      . . . . .
      <a[n,1]> <a[n,2]> . . . <a[n,n]>
Обр9.1 Количество отрицательных элементов по столбцам:
Обр9.2 <kolich[1]> <kolich[2]> . . . <kolich[n]>
Обр10 Максимальное их число в столбце <Nom>
    
```

Числа разделены пробелами

**Имена входного и выходного файлов передаются как параметры программы:**

Первый – имя файла с исходными данными  
 Второй – имя файла для вывода исходных данных и результатов

## 7. Аномалии

№	Описание	Условие возникновения	Реакция
1	Некорректное число строк/столбцов	$(n < 1)$	Обр 6
2	Некорректное число строк/столбцов	$(n > 5)$	Обр 6
3	Есть некорректный элемент в матрице	$\exists i \exists j (A[i,j] < 100)$	Обр 7
4	Есть некорректный элемент в матрице	$\exists i \exists j (A[i,j] > 100)$	Обр 7

## 8. Функциональные тесты

№ теста	Входные данные	Ожидаемый результат	Смысл теста
1	N=0	Обр 6	Аномальная ситуация 1: $n < 1$
2	N=6	Обр 6	Аномальная ситуация 2: $n > 5$
3	N=5 $A = \begin{pmatrix} 100 & -100 & -101 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -102 \end{pmatrix}$	Обр 7 $A_{13} = -101$	Аномальная ситуация 3 ( $A_{13} < -100$ )
4	N=1 $A = \begin{pmatrix} 101 \end{pmatrix}$	Обр 7 $A_{11} = 101$	Аномальная ситуация 4 ( $A_{11} > 100$ )
5	n=3; $A = \begin{pmatrix} 1 & 5 & 4 \\ 1 & 2 & 0 \\ 10 & -5 & -3 \end{pmatrix}$	$A = \begin{pmatrix} 0 & 5 & 0 \\ 0 & 0 & 0 \\ 10 & -5 & 0 \end{pmatrix}$	Кратных пяти на гл.диагонали нет данные взяты из примера 1
6	n=5; $A = \begin{pmatrix} 1 & 2 & 3 & 4 & 6 \\ 7 & 8 & 9 & 11 & 12 \\ 13 & 14 & -1 & -2 & -2 \\ -4 & -6 & -7 & -8 & -9 \\ 11 & 9 & 8 & 7 & 6 \end{pmatrix}$	$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	Кратных пяти нет вообще, максимальное количество проверок и замен (м.в.н.)
7	n=3; $A = \begin{pmatrix} 1 & -5 & 4 \\ 1 & 10 & 0 \\ 10 & -5 & -3 \end{pmatrix}$	$Kolic_{\text{max}} = 2$ $Nom = 2$	На гл.диагонали есть элемент, кратный 5; Во всех столбцах разное количество отрицательных элементов; один максимум, данные взяты из примера 2
8	n=3; $A = \begin{pmatrix} 0 & 5 & 4 \\ 1 & 10 & 0 \\ 10 & 5 & 5 \end{pmatrix}$	$Kolic_{\text{max}} = 0$ $Nom = 1$	На гл.диагонали все элементы, кратны 5; Во всех столбцах нет (равное количество) отрицательных элементов;

## 9. Метод

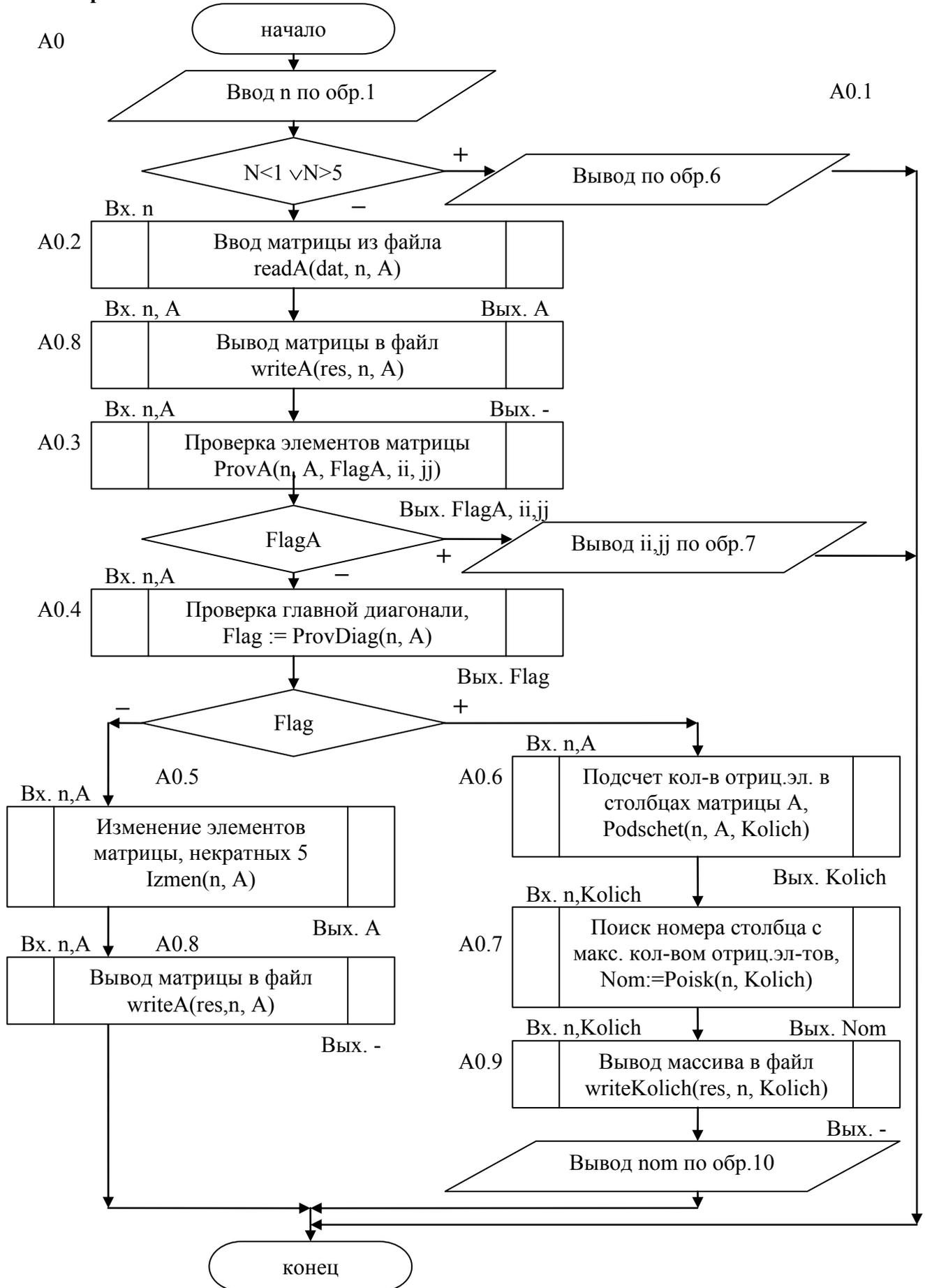
Отделим ввод-вывод от обработки данных (собственно решения задачи).

То есть, разделим нашу задачу на десять подзадач:

1. Подзадача А0.1. Ввести из файла (обр.1) и проверить значение количества строк/столбцов (обр.6)
2. Подзадача А0.2. Ввести матрицу из файла (обр.2).
3. Подзадача А0.3. Проверить значения элементов матрицы (обр.7)
4. Подзадача А0.4. Проверить отсутствие на главной диагонали элементов, кратных пяти
5. Подзадача А0.5. Изменить матрицу, заменив все элементы некратные пяти на 0
6. Подзадача А0.6. Подсчитать в каждом столбце количество отрицательных элементов
7. Подзадача А0.7. Найти номер столбца с максимальным количеством отрицательных элементов
8. Подзадача А0.8. Вывести матрицу (обр.5 и 8.2)
9. Подзадача А0.9. Вывести количества отрицательных элементов (обр.9)
10. Подзадача А0.10. Вывести номер столбца с максимальным количеством отрицательных элементов (обр.10)

Сейчас, на нулевом уровне, опишем ввод-вывод простых переменных и массивов (A0.1, A0.2, A0.8, A0.9, A0.10) и логику решения задачи в целом. Остальные пять задач оставим в виде абстракций. Все подзадачи кроме первой и последней выполним в виде процедур и функций, расположенных в отдельном модуле.

### 10. Алгоритм



**11. Программа на Delphi (с процедурами-заглушками, кроме процедур ввода и вывода).**

Имена входного и выходного файлов передаются как параметры программы:

Первый – имя файла с исходными данными

Второй – имя файла для вывода исходных данных и результатов

*(Если выделено достаточное колич-во процедур, основная программа уместится на лист, как и блок-схема)*

\*\*\*\*\*Файл основной программы PFU.dpr\*\*\*\*\*

```

program PFU;
{$APPTYPE CONSOLE}
uses ProfUN; {там все константы, типы, процедуры и функции}

var {раздел описания переменных -----}
    A: Matrix;
    Kolich: MasKol;
    n, Nom, ii, jj: byte;
    flag, flagA: Boolean;
    dat, res: TextFile;

begin {раздел операторов -----}
    assignFile(dat, ParamStr(1)); reset(dat);
    assignFile(res, ParamStr(2)); rewrite(res);
    writeln(' ':15, 'Программа с модулем'); {Обр.3}
    {ввод n и его проверка-----}
    readln(dat, n); {Обр.1}
    if (N<1) OR (N>Nmax) then
    begin
        writeln(res, 'Некорректное число строк. '#13#10, ' Задача не решалась'); {Обр.6}
        CloseFile(dat); CloseFile(res);
        Exit;
    end;
    {ввод и вывод матрицы A -----}
    readA(dat, n, A); {Обр.2}
    CloseFile(dat);
    writeln('Исходная матрица из ',n,' строк и ',n,' столбцов:'); {Обр.4}
    writeA(res, n, A); {Обр.5}
    {-----решение задачи -----}
    ProvA(n, A, FlagA, ii, jj); { Проверка элементов матрицы }
    If FlagA then
    Begin
        Writeln(res, 'Некорректный элемент матрицы'); {Обр.7}
        Writelnres, 'A[', ii, ', ', jj, ' ] = ', A[ii,jj]);
        Writeln(res, 'Задача не решалась');
        CloseFile(res);
        Exit;
    end;
    {-----}
    Flag := ProvDiag(n, A); {Проверка главной диагонали}
    If flag then {-----если есть кратный 5-----}
    Begin
        Podschet(n, A, Kolich); {Подсчет кол-ва отриц.эл. в столбцах матрицы A}
        Nom:=Poisk(n, Kolich); {Поиск номера столбца с макс. кол-вом отриц.эл-тов}
        {Вывод массива в файл}
        writeln(res, 'Количество отрицательных элементов по столбцам:'); {Обр.9.1}
        writeKolich(res, n, Kolich); {Обр.9.2}
        {Вывод nom по обр.10}
        writeln(res, 'Максимальное их число в столбце ', Nom); {Обр.10}
    End
    Else {-----если нет кратных 5-----}
    Begin
        Izmen(n, A); {Изменение элементов матрицы, некратных 5}
        {Вывод матрицы в файл}
        writeln(res, 'Измененная матрица'); {Обр.8.1}
        writeA(res,n, A); {Обр.8.2}
    End;
    {-----}
    CloseFile(res);
End.

```

\*\*\*\*\*Модуль ProFUn.pas со всеми процедурами и функциями\*\*\*\*\*

Unit ProFUn;  
Interface

Const  
Nmax = 5;

Type  
Matrix = array [1..Nmax, 1..Nmax] of ShortInt;  
MasKol = array [1..Nmax] of byte;

{Ввод матрицы из файла, файл уже открыт}

Procedure readA(var dat: TextFile; const n: byte; out A: Matrix);

{Вывод матрицы в файл, файл уже открыт}

Procedure writeA(var res: TextFile; const n: byte; var A: Matrix);

{Вывод массива в файл, файл уже открыт }

Procedure writeKolich(var res: TextFile; const n: byte; var Kol: MasKol);

Таблица соответствия параметров		
Формальные параметры	Фактические параметры	Тип
res	res	TextFile
n	n	Byte
Kol	Kolich	MasKol

{Проверка элементов матрицы}

Procedure ProvA(const n:byte; var A: Matrix; out FlagA: Boolean; out ii, jj: byte);

Таблица соответствия параметров		
Формальные параметры	Фактические параметры	Тип
n	n	Byte
A	A	Matrix
FlagA	FlagA	Boolean
ii	ii	byte
jj	jj	byte

{Проверка главной диагонали}

Function ProvDiag(const n:byte; var A: Matrix): Boolean;

{Изменение элементов матрицы, некратных 5}

Procedure Izmen(const n:byte; var A: Matrix);

{Подсчет кол-в отриц.эл. в столбцах матрицы A}

Procedure Podschet(const n:byte; var A: Matrix; out Kol: MasKol);

Таблица соответствия параметров		
Формальные параметры	Фактические параметры	Тип
n	n	Byte
A	A	Matrix
Kol	Kolich	MasKol

{Поиск номера столбца с макс. кол-вом отриц.эл-тов}

Function Poisk(const n:byte; var Kol: MasKol):byte;

Таблица соответствия параметров		
Формальные параметры	Фактические параметры	Тип
n	n	Byte
Kol	Kolich	MasKol
Poisk (значение функции)	Nom	Byte

```
{Ввод матрицы из файла, файл уже открыт }
Procedure readA(var dat: TextFile; const n: byte; out A: Matrix);
Var i,j: byte;
Begin
  For i:=1 to n do
    begin
      For j:=1 to n do
        Read(dat, A[i,j]);
      Readln(dat);
    end;
End;
```

```
{Вывод матрицы в файл, файл уже открыт }
Procedure writeA(var res: TextFile; const n: byte; var A: Matrix);
Var i,j: byte;
Begin
  For i:=1 to n do
    begin
      For j:=1 to n do
        Write(res, A[i,j]:3, ' ');
      Writeln(res);
    end;
End;
```

```
{Вывод массива в файл, файл уже открыт }
Procedure writeKolich(var res: TextFile; const n: byte; var Kol: MasKol);
Var j: byte;
Begin
  For j:=1 to n do
    Write(res, Kol[j], ' ');
    Writeln(res);
  end;
End;
```

```
{Проверка элементов матрицы}
Procedure ProvA(const n:byte; var A: Matrix; out FlagA: Boolean; out ii, jj: byte);
{описания локальных переменных}
Begin
```

```
{Заглушка} writeln(' Заглушка - Проверка элементов матрицы ');
{тест 3} // ii:=1; jj:=3; FlagA:=true;
{тест 4} // ii:=1; jj:=1; FlagA:=true;
{остальные тесты} FlagA:=False;
```

```
End;
```

```
{Проверка главной диагонали}
Function ProvDiag(const n:byte; var A: Matrix): Boolean;
Var Flag: Boolean;
{описания других локальных переменных}
Begin
```

```
{Заглушка} writeln(' Заглушка - Проверка главной диагонали ');
{тест 7,8} // Flag:=True;
{остальные тесты} Flag:=False;
```

```
ProvDiag:=Flag;
End;
```

{Изменение элементов матрицы, некратных 5}

Procedure Izmen(const n:byte; var A: Matrix);

Var i, j: byte;

{описания других локальных переменных}

Begin

```
{Заглушка} writeln(' Заглушка - Изменение элементов матрицы, некратных 5');
{тест 5} // for i:=1 to 3 do for j:=1 to 3 do A[i,j]:=0; A[1,2]:=5; A[3,1]:=10; A[3,2]:=-5;
{тест 6} // for i:=1 to 3 do for j:=1 to 3 do A[i,j]:=0;
{остальные тесты} ;
```

End;

{Подсчет кол-в отриц.эл. в столбцах матрицы A}

Procedure Podschet(const n:byte; var A: Matrix; out Kol: MasKol);

Var j:byte;

{описания других локальных переменных}

Begin

```
{Заглушка} writeln(' Заглушка - Подсчет кол-в отриц.эл. в столбцах матрицы A');
{тест 7} // Kol[1]:=0; Kol[2]:=2; Kol[3]:=1;
{тест 8} // for j:=1 to 3 do Kol[j]:=0;
{остальные тесты} ;
```

End;

{Поиск номера столбца с макс. кол-вом отриц.эл-тов}

Function Poisk(const n:byte; var Kol: MasKol):byte;

Var

Nom: byte;

{описания других локальных переменных}

Begin

```
{Заглушка} writeln(' Заглушка - Поиск номера столбца с макс. кол-вом отриц.эл-тов');
{тест 7} // Nom:=2;
{тест 8} // Nom:=1;
{остальные тесты} ;
```

Poisk:=Nom;

End;

End.

Далее рассмотрим каждую из подзадач подробнее.

## Абстракция А0.3

**1. Условие.** Проверить, есть ли в матрицы элементы, абсолютная величина которых больше 100

**2. Уточненная постановка задачи.**

Дана целочисленная квадратная матрица  $A$  из  $n$  строк и столбцов. Проверить (FlagA), есть ли в матрицы элементы, абсолютная величина которых больше 100. Если есть, то присвоить FlagA значение True и найти номер строки ( $ii$ ) и столбца ( $jj$ ) первого такого элемента, иначе присвоить FlagA значение False.

**3. Примеры.**

Тест 3 есть некорректный элемент  $A[1,3]=-101$ , FlagA= True

Тест 4 есть некорректный элемент  $A[1,1]=101$ , FlagA= True

Тесты 5-8 нет некорректных элементов, FlagA= False

**4. Таблица данных**

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	A	Заданная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)
	n	число строк и столбцов в матрице A, $0 < n \leq 5$	цел	простая переменная
Выходные данные	ii	Строка с некорр. $A[ii,jj]$ , $0 < ii \leq 5$	цел	простая
	jj	Столбец с некорр. $A[ii,jj]$ , $0 < jj \leq 5$	цел	простая
	flagA	= True, если есть некорректные $A_{ij}$ , в противном случае False	лог	простая переменная
Промежуточные данные	i	Номер текущей строки, $0 < i \leq 6$	цел	простая переменная
	j	Номер текущего столбца, $0 < j \leq 6$	цел	простая переменная

**5. Входная форма.**

нет ввода/вывода

**6. Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тест 3 есть некорректный элемент  $A[1,3]=-101$ , FlagA= True

Тест 4 есть некорректный элемент  $A[1,1]=101$ , FlagA= True

Тесты 5-8 нет некорректных элементов, FlagA= False

**9. Метод (задача типа 1) из файла Sem-3.pdf)**

Пусть  $FlagA = \begin{cases} \text{Истина, если есть в матрице элемент } |A_{i,j}| > 100; \\ \text{Ложь, если нет такого элемента} \end{cases}$

Предположим сначала, что такого элемента в матрице нет (FlagA:=False).

Затем будем просматривать строки, начиная с первой ( $i:=1$ )

Пока не просмотрены все ( $i \leq n$ ) и не найден такой элемент (FlagA=False)

В каждой строке будем просматривать элементы, начиная с первого ( $j:=1$ )

Пока не просмотрены все ( $j \leq n$ ) и не найден такой элемент (FlagA=False)

Если рассматриваемый элемент  $A_{ij}$  по модулю больше 100

То

искомый элемент найден! (FlagA:=True)

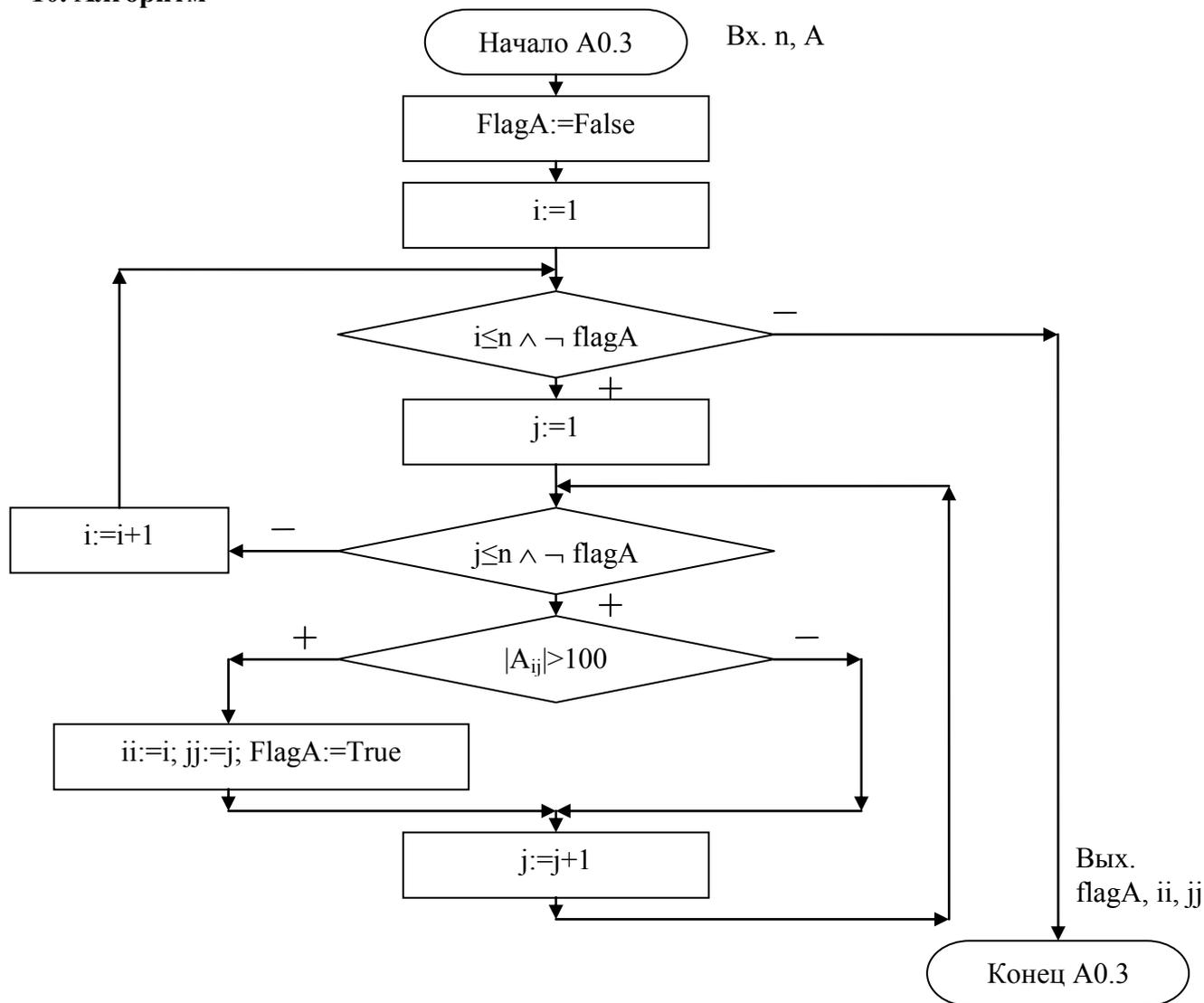
Запоминаем его местонахождение ( $ii:=i$ ;  $jj:=j$ )

Переходим к следующему элементу в строке ( $j:=j+1$ )

Переходим к следующей строке

Найденные значения FlagA, ii, jj будут искомыми.

## 10. Алгоритм



## 11. Программа на паскале. Процедура общего вида

```

{Проверка элементов матрицы}
Procedure ProvA(const n:byte; var A: Matrix; out FlagA: Boolean; out ii, jj: byte);
var {описания локальных переменных}
  i, j: byte;
Begin
  FlagA:=False;
  i:=1;
  while (i<=n) and not flagA do
  begin
    j:=1;
    while (j<=n) and not flagA do
    begin
      if abs(A[i,j]) > 100 then
      begin
        ii:=i; jj:=j; flagA:=True
      end;
      inc(j);
    end;
    inc(i);
  end;
End;

```

**Абстракция А0.4**

**1.Условие.** Проверить, есть ли на главной диагонали матрицы элементы, кратные 5.

**2.Уточненная постановка задачи.**

Дана целочисленная квадратная матрица  $A$  из  $n$  строк и столбцов. Проверить (flag), есть ли на главной диагонали матрицы элементы, кратные 5.

**3.Примеры.**

Тест 5,6 нет на главной диагонали матрицы элементов, кратных 5, Flag= False

Тест 7,8 есть на главной диагонали матрицы элементы, кратные 5, Flag= True

**4. Таблица данных**

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	A	Заданная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)
	n	число строк и столбцов в матрице A, $0 < n \leq 5$	цел	простая переменная
Выходные данные	flagA	= True, если есть кратные 5 на гл. диагонали, в противном случае False	лог	простая переменная

Промежуточные данные	i	Номер текущей строки, $0 < i \leq 6$	цел	простая переменная

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тест 5,6 нет на главной диагонали матрицы элементов, кратных 5, Flag= False

Тест 7,8 есть на главной диагонали матрицы элементы, кратные 5, Flag= True

**9. Метод (задача типа 1) из файла Sem-3.pdf)**

У элементов, лежащих на главной диагонали индексы равны ( $i=j$ ).

Пусть Flag=  $\begin{cases} \text{Истина, если есть в на гл. диагонали элемент } A_{i,i}, \text{ кратный } 5; \\ \text{Ложь. если нет такого элемента} \end{cases}$

Предположим сначала, что такого элемента в матрице нет (Flag:=False).

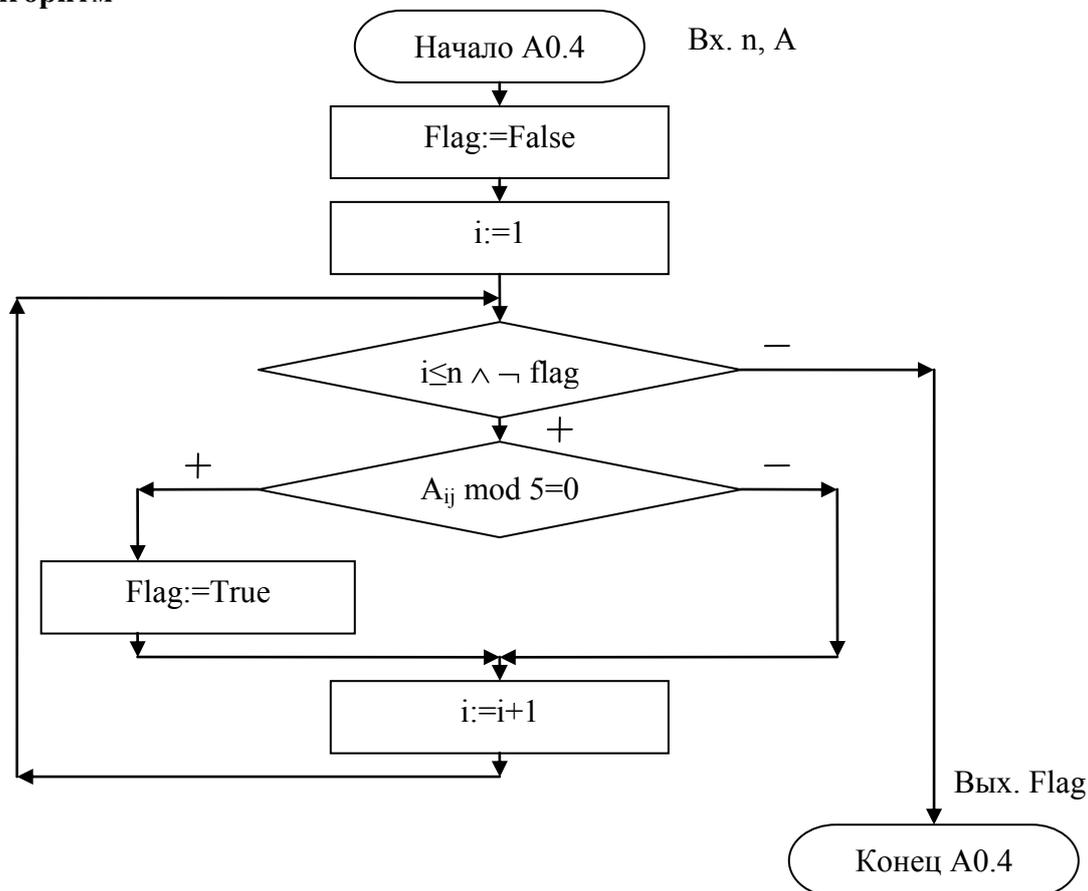
Затем будем просматривать строки, начиная с первой ( $i:=1$ )

Пока не просмотрены все ( $i \leq n$ ) и не найден такой элемент (FlagA=False)

$\left\{ \begin{array}{l} \text{В каждой строке будем проверять один элемент } A_{ii} \\ \text{Если остаток от деления } A_{ij} \text{ на } 5 \text{ равен } 0 \\ \quad \text{То искомый элемент найден! (Flag:=True)} \\ \text{Переходим к следующей строке (i:=i+1)} \end{array} \right.$

Найденное значение Flag будет искомым.

## 10. Алгоритм



## 11. Программа на паскале. Функция

{Проверка главной диагонали}

Function ProvDiag(const n:byte; var A: Matrix): Boolean;

Var {описания локальных переменных}

Flag: Boolean;

i: byte;

Begin

Flag:=False;

I:=1;

While (i<=n) and not flag do

Begin

If A[i,i] mod 5 = 0 then flag:=true; // или можно в итерационном цикле flag:= A[i,i] mod 5 = 0;

Inc(i);

End;

ProvDiag:=Flag;

End;

**Абстракция А0.5**

1. Условие. Заменить в матрице все элементы, не кратные пяти, нулями.

2. Уточненная постановка задачи.

Дана целочисленная квадратная матрица  $A$  из  $n$  строк и столбцов. Изменить матрицу, заменив все элементы, не кратные пяти, нулями.

3. Примеры.

Тест 5 есть и кратные и не кратные пяти

Тест 6 нет кратных пяти – заменяются все

**4. Таблица данных**

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	A	Заданная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)
	n	число строк и столбцов в матрице A, $0 < n \leq 5$	цел	простая переменная
Выходные данные	A	Измененная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)

Промежуточные данные	i	Номер текущей строки, $0 < i \leq 6$	цел	простая переменная
	j	Номер текущего столбца, $0 < j \leq 6$	цел	простая переменная

**5. Входная форма.**

нет ввода/вывода

**6. Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тест 5 есть и кратные и не кратные пяти

Тест 6 нет кратных пяти – заменяются все

**9. Метод**

Будем просматривать строки, начиная с первой до последнюю ( $i:=1;+1;n$ )

В каждой строке будем просматривать все элементы, начиная с первого до последнего ( $j:=1;+1;n$ )

Каждый элемент будем проверять

Если остаток от деления  $A_{ij}$  на 5 не равен 0

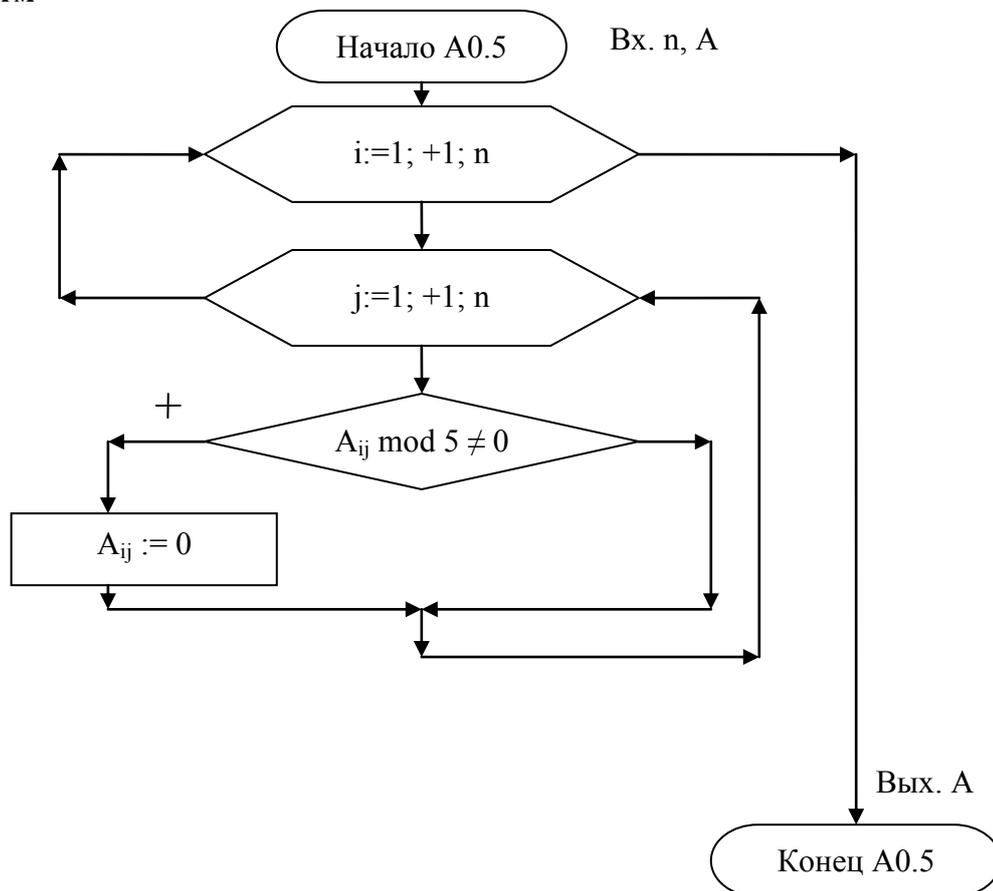
То меняем его на ноль ( $A[i,j]:=0$ )

Переходим к следующему элементу в строке

Переходим к следующей строке ( $i:=i+1$ )

Матрица изменена

## 10. Алгоритм



## 11. Программа на паскале. Процедура общего вида

{Изменение элементов матрицы, некратных 5}

Procedure Izmen(const n:byte; var A: Matrix);

Var

i, j: byte;

Begin

For i:=1 to n do

For j:=1 to n do

If A[i,j] mod 5 <> 0 then A[i,j]:=0;

End;

**Абстракция А0.6**

**1.Условие.** Найти количество отрицательных элементов в каждом столбце матрицы.

**2.Уточненная постановка задачи.**

Дана целочисленная квадратная матрица  $A$  из  $n$  строк и столбцов. Создать новый массив  $Kol$  из  $n$  элементов, каждому элементу  $Kol_j$  которого присвоить значение количества отрицательных элементов в соответствующем столбце ( $j$ ) матрицы  $A$ .

**3.Примеры.**

Тест 7 есть отрицательные элементы в столбцах. Во всех разное количество

Тест 8 нет отрицательных элементов

**4. Таблица данных**

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	A	Заданная матрица, $ A_{ij}  \leq 100$	цел	Двухмерный массив (5x5)
	n	число строк и столбцов в матрице A, $0 < n \leq 5$	цел	простая переменная
Выходные данные	Kol	Количество отрицательных элементов в каждом столбце, $0 \leq kol_j \leq 5$	цел	Одномерный массив (5)

Промежуточные данные	j	Номер текущего столбца, $0 < j \leq 6$	цел	простая переменная
	i	Номер элемента в столбце, $0 < i \leq 6$	цел	простая переменная

Промежуточные данные заполняются не сразу, а по мере необходимости иметь дополнительные переменные

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тест 7 есть отрицательные элементы в столбцах. Во всех разное количество

Тест 8 нет отрицательных элементов ( во всех столбцах равное количество)

**9. Метод**

Будем просматривать столбцы, начиная с первого до последнего ( $j:=1;+1;n$ )

В каждой столбце будем считать количество  $Kol_j$ ,

Начальное значение 0 ( $Kol[j]:=0;$ )

Проверим элементы в столбце с первого до последнего ( $i:=1;+1;n$ )

Если элемент отрицательный ( $Kol[j]<0$ )

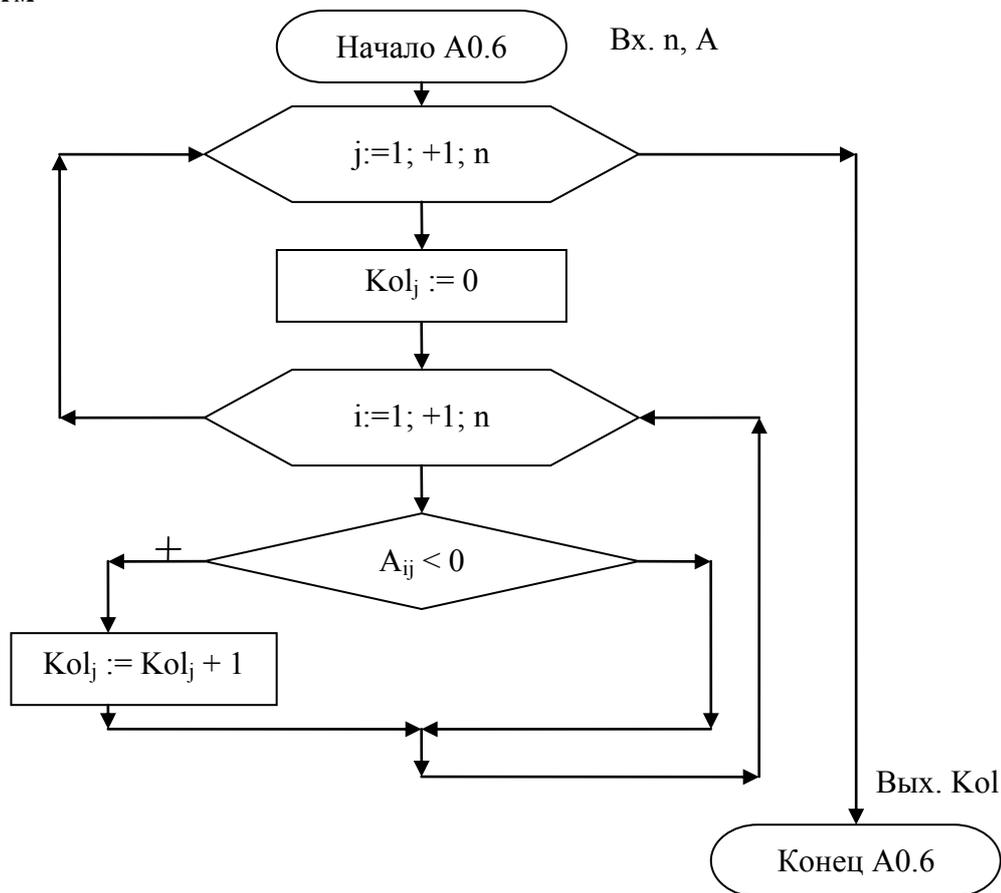
То увеличиваем количество ( $Kol[j]:=Kol[j]+1$ )

Переходим к следующему элементу в столбце

Переходим к следующему столбцу

Массив заполнен

## 10. Алгоритм



## 11. Программа на паскале. Процедура общего вида

{Подсчет кол-в отриц.эл. в столбцах матрицы A}

Procedure Podschet(const n:byte; var A: Matrix; out Kol: MasKol);

Var  
i,j:byte;

Begin

For j:=1 to n do

Begin

Kol[j]:=0;

For i:=1 to n do

If A[i,j]<0 then inc(Kol[j]);

End;

End;

**Абстракция А0.7**

**1.Условие.** Найти номер элемента массива с максимальным значением.

**2.Уточненная постановка задачи.**

Дан целочисленный одномерный массив **Kol** из **n** элементов. Найти номер (**Nom**) первого элемента с максимальным значением.

**3.Примеры.**

Тест 7 Все элементы разные,  $Nom=2$

Тест 8 Все элементы одинаковые, максимум – первый из них  $Nom=1$

**4. Таблица данных**

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	Kol	Количество отрицательных элементов в каждом столбце, $0 \leq kol_j \leq 5$	цел	Одномерный массив (5)
	n	число строк и столбцов в матрице A, $0 < n \leq 5$	цел	простая переменная
Выходные данные	Nom	Номер первого столбца с максимальным количеством отрицательных элементов, $0 < Nom \leq 5$	цел	простая переменная

Промежуточные данные	k	Номер текущего элемента, $0 < k \leq 6$	цел	простая переменная

**5.Входная форма.**

нет ввода/вывода

**6.Выходная форма**

нет ввода/вывода

**7. Аномалии**

нет ввода/вывода

**8. Тесты**

Тест 7 Все элементы разные,  $Nom=2$

Тест 8 Все элементы одинаковые, максимум – первый из них  $Nom=1$

**9. Метод**

Пусть максимальное значение находится в первом элементе массива. ( $Nom:=1$ )

Будем просматривать все остальные элементы со второго до последнего ( $k:=2;+1;n$ )

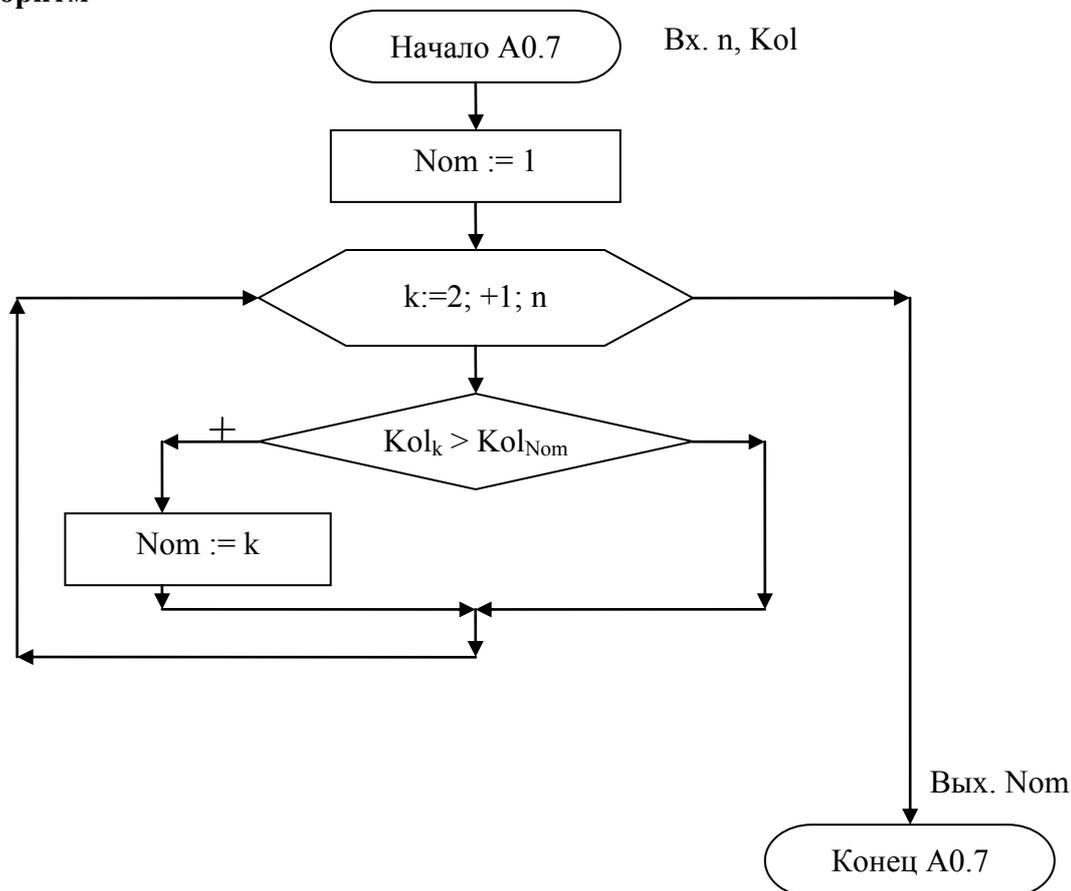
Если найдем элемент больше ( $Kol[k]>Kol[Nom]$ ),

То изменим значение номера максимума на текущее ( $Nom:=k$ )

Переходим к следующему элементу

Текущее значение  $Nom$  – искомое

## 10. Алгоритм



## 11. Программа на паскале. Функция

```

{Поиск номера столбца с макс. кол-вом отриц.эл-тов}
Function Poisk(const n:byte; var Kol: MasKol):byte;
Var
  Nom: byte;
  k: byte; {описания других локальных переменных}
Begin
  Nom:=1;
  For k:=2 to n do
    If Kol[k]> Kol[Nom] then Nom:=k;

  Poisk:=Nom;
End;
  
```

После написания текста программы можно перейти к написанию **структурных тестов**.

Для этого составим таблицу, в которой перечислим все структуры ветвления и циклов в порядке их появления в программе.

И укажем плюсом ситуации, которые уже случались в функциональных тестах (1-8), чтобы найти строки без плюсов и создать для них недостающие тесты:

Структурные тесты. Выявление случаев, не покрываемых функциональными тестами (1-8):

Тест№	1	2	3	4	5	6	7	8	Новые тесты				
	9	10	11	12									
Структура													
If (N<1) or (N>5)	Невозможно одновременное выполнение этих условий												
N<1 истина N>5 истина	Невозможно одновременное выполнение этих условий												
N<1 истина N>5 ложь (N=5)	Невозможно одновременное выполнение этих условий												
N<1 истина N>5 ложь (N<5)	N=0	-	-	-	-	-	-	-	-	-	-	-	-
N<1 ложь (N=1) N>5 истина	Невозможно одновременное выполнение этих условий												
N<1 ложь (N>1) N>5 истина	-	N=6	-	-	-	-	-	-	-	-	-	-	-
N<1 ложь (N=1) N>5 ложь (N=5)	Невозможно одновременное выполнение этих условий												
N<1 ложь (N>1) N>5 ложь (N=5)			N=5			N=5							
N<1 ложь (N=1) N>5 ложь (N<5)				N=1							1	1	
N<1 ложь (N>1) N>5 ложь (N<5)					N=3		N=3	N=3	3				2
Ввод/вывод матрицы For i:=1 to n do	Н	Н											
0 раз (i>n)	Е	Е	Невозможно, т.к выше проверено, что n≥1										
1 раз (i=n)	Т	Т		1							1	1	
более 1 раза (i<n)	В	В	5		3	5	3	3	3				2
For j:=1 to n do	В	В											
0 раз	О	О	Невозможно, т.к выше проверено, что n≥1										
1 раз	Д	Д		1							1	1	
более 1 раза	А	А	5		3	5	3	3	3				2
Проверка элементов while (i<=n) and not flagA do	И	И											
(i<=n) истина (i<n) flagA ложь	П	П	+	-	+	+	+	+					
(i<=n) истина (i=n) flagA ложь	Р	Р	-	+	+	+	+	+					
(i<=n) истина (i<n) flagA истина	В	В	+	-	-	-	-	-					
(i<=n) истина (i=n) flagA истина	Е	Е	-	-	-	-	-	-	+	-	-	-	
(i<=n) ложь (i>n) flagA ложь	Р	Р	-	-	+	+	+	+					
(i<=n) ложь (i>n) flagA истина	М	М	-	+	-	-	-	-					
while (j<=n) and not flagA do	Т	Т											
(j<=n) истина (j<n) flagA ложь	Р	Р	+	-	+	+	+	+					
(j<=n) истина (j=n) flagA ложь	И	И	-	+	+	+	+	+					
(j<=n) истина (j<n) flagA истина	Ц	Ц	+	-	-	-	-	-					
(j<=n) истина (j=n) flagA истина	Ы,	Ы,	-	-	-	-	-	-	+	-	-	-	
(j<=n) ложь (j>n) flagA ложь			-	-	+	+	+	+					
(j<=n) ложь (j>n) flagA истина			-	+	-	-	-	-					
if abs(A[i,j]) > 100 then	И	И											
abs(A[i,j]) > 100	З	З	-101	101						101			
abs(A[i,j]) < 100	А	А	-		+	+	+	+			+	+	+
abs(A[i,j]) = 100	Д	Д	100							100			
	А	А	-100										
If FlagA then	Ч	Ч											
FlagA истина	А	А	+	+						+			



## 8. Недостающие структурные тесты (остальное пересекается с функциональными)

№ теста	Входные данные	Ожидаемый результат	Смысл теста
9	$N=3$ $A = \begin{pmatrix} 100 & 100 & 100 \\ 100 & 101 & 100 \\ 100 & 100 & 100 \end{pmatrix}$	Обр 7 $A_{22}=101$	Аномальная ситуация 4 ( $A_{22}>100$ ) В предпоследней строке и столбце
10	$n=1;$ $A = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$	$A = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$	Единственный элемент главной диагонали НЕ кратен 5, одна замена, 1 раз выполняется цикл при изменении матрицы
11	$n=1;$ $A = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$	Kolich = (0) Nom=1	Единственный (и последний) элемент главной диагонали кратен 5, 0 раз выполняется цикл в поиске максимума, 1 раз выполняется цикл при выводе количеств
12	$n=2;$ $A = \begin{pmatrix} 1 & 0 \\ 5 & -5 \end{pmatrix}$	Kolich = (0 1) Nom=2	На главной диагонали есть кратный 5 элемент, 1 раз выполняется цикл в поиске максимума,