Федеральное государственное бюджетное образовательное учреждение
высшего образования
Национальный исследовательский университет «МЭИ»

Типовой расчет «Процедуры, функции, модули»

Вариант 0

Задание выполнил: Фамилия И.О.
Студент группы А-00-26

Проверил: _____
*Оценка:*_____
*Замечания:*_____
_____
_____
_____
_____
_____
_____
_____
_____

Москва 2026 НИУ «МЭИ»

Пример решения задачи

Абстракция А0.

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

«Кратно пяти» → *целочисленная*

«Главная диагональ» → *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 некратные 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.

$$Kolich = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix}$$

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, $\|A_{ij}\|<=100$	цел	Двухмерный массив (5x5)	+XXX (:4)
	n	число строк и столбцов в матрице A, $0 < n \le 5$	цел	простая переменная	X (:1)
Выходные данные	A	Измененная матрица, $\|A_{ij}\|<=100$	цел	Двухмерный массив (5x5)	+XXX (:4)
	Kolich	Количество отрицательных элементов в каждом столбце, $0 \le Kolich_i \le 5$	цел	Одномерный массив (5)	X (:1)
	Nom	Номер первого столбца с максимальным количеством отрицательных элементов, $0 < Nom \le 5$	цел	простая переменная	X (:1)

Промежу-точные данные	dat	Файл с исходными данными с именем вида unit_dat#.txt	Текст.	файловая пер.	---
	res	Файл с отчетом с именем вида unit_res#.txt	Текст.	файловая пер.	---
	Flag	= 1, если есть на гл.диагонали есть элемент, кратный пяти, в противном случае 0	лог/цел	простая переменная	---
	ii	Строка с некорр. A[ii,jj], 0<ii≤ 5	цел	простая	X (:1)
	jj	Столбец с некорр. A[ii,jj], 0<jj≤ 5	цел	простая	X (:1)
	FlagA	= 1, если есть некорректные Aij, в противном случае 0	лог/цел	простая переменная	---

5.Входная форма. (Файлы unit_dat<№ теста>.txt)

Обр1
```
<n>
```
Обр2
```
<a[0][0]> <a[0][1]> ... <a[0][n-1]>
<a[1][0]> <a[1][1]> ... <a[1][n-1]>
. . . . . . . . . . . . . . . . . .
<a[n-1][0]> <a[n-1][1]> ... <a[n-1][n-1]>
```

6.Выходная форма (Файлы unit_res<№ теста>.txt)

15 пробелов перед текстом

Обр3
```
    Программа с модулем
```
Обр4
```
Исходная матрица A из <n> строк и <n> столбцов:
```
Обр5
```
<A[0][0]> <A[0][1]> ... <A[0][n-1]>
<A[1][0]> <A[1][1]> ... <A[1][n-1]>
. . . . . . . . . . . . . . . . . .
<A[n-1][0]> <A[n-1][1]> ... <A[n-1][n-1]>
```

Числа разделены пробелами

Обр6.1
```
Измененная матрица A
```
Обр6.2
```
<A[0][0]> <A[0][1]> ... <A[0][n-1]>
<A[1][0]> <A[1][1]> ... <A[1][n-1]>
. . . . . . . . . . . . . . . . . .
<A[n-1][0]> <A[n-1][1]> ... <A[n-1][n-1]>
```
Обр7.1
```
Количество отрицательных элементов по столбцам:
```
Обр7.2
```
<Kolich[0]> <Kolich[1]> . . . <Kolich[n-1]>
```
Обр8
```
Максимальное их число в столбце <Nom>
```

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) или (n>5) Или: не число, не целое число, отсутствует	Сообщение А1 «Некорректное число строк. Задача не решалась» И завершение работы программы с 1
2	Есть некорректный элемент в матрице	∃ii∃jj (A[ii,jj]>100) или ∃ii∃jj (A[ii,jj]<-100)	Сообщение А2 «Некорректный элемент матрицы A[<ii>, <jj>] = <A[ii-1,jj-1]> Задача не решалась» И завершение работы программы с 2
3,4	Файлы не открылись	Файлы не открылись	Сообщения А3 и А4 и завершение с 3 или 4

8. Функциональные тесты

№	Входные данные	Ожидаемый результат	Смысл теста
1а	n=0	Обр А1	Аномальная ситуация 1 (n<1)
1б	n=-1	Обр А1	Аномальная ситуация 1 (n<1)
1в	n='s'	Обр А1	Аномальная ситуация 1 симв.
2	n=6	Обр А1	Аномальная ситуация 1 (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 А2 A_{13}=-101	Аномальная ситуация 2 (A_{13}<-100) (A_{55}<-100) Ищет только первый из нескольких. Вывод номеров, а не индексов.
4	n=1 $A = \begin{pmatrix} 101 \end{pmatrix}$	Обр А2 A_{11}=101	Аномальная ситуация 2 (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, вне диагонали – есть. Данные взяты из примера 1. Есть отрицательные и положительные, есть нули вне гл.диагонали.
6	n=3; $A = \begin{pmatrix} 1 & 4 & 4 \\ 1 & 2 & 2 \\ 11 & -6 & -3 \end{pmatrix}$	$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	**Кратных пяти нет** вообще, есть отрицательные и положительные.
7	n=3; $A = \begin{pmatrix} 1 & -5 & 4 \\ 1 & 10 & 0 \\ 10 & -5 & -3 \end{pmatrix}$	$Kolich = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix}$ *Nom*=2	**На гл.диагонали есть элемент, кратный 5;** Во всех столбцах разное количество отрицательных элементов; данные взяты из примера 2
8	n=3; $A = \begin{pmatrix} 0 & 5 & 4 \\ 1 & 10 & 0 \\ 10 & 5 & 5 \end{pmatrix}$	$Kolich = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ *Nom*=1	**На гл.диагонали все элементы, кратны 5;** Во всех столбцах (равное) нет отрицательных элементов;
9	n=5; $A = \begin{pmatrix} 1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 1 & 2 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & -1 & -5 \end{pmatrix}$	$Kolich = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix}$ *Nom*=5	**На гл.диагонали все элементы, кратны 5;** Максимум отрицательных элементов в 5 столбце; Макс.выч.нагрузка
10а	Нет входного файла	А3	**Не удалось открыть вх.ф.**
10б	Выходной файл только для чтения	А4	**Не удалось создать вых.ф.**

9. Метод

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

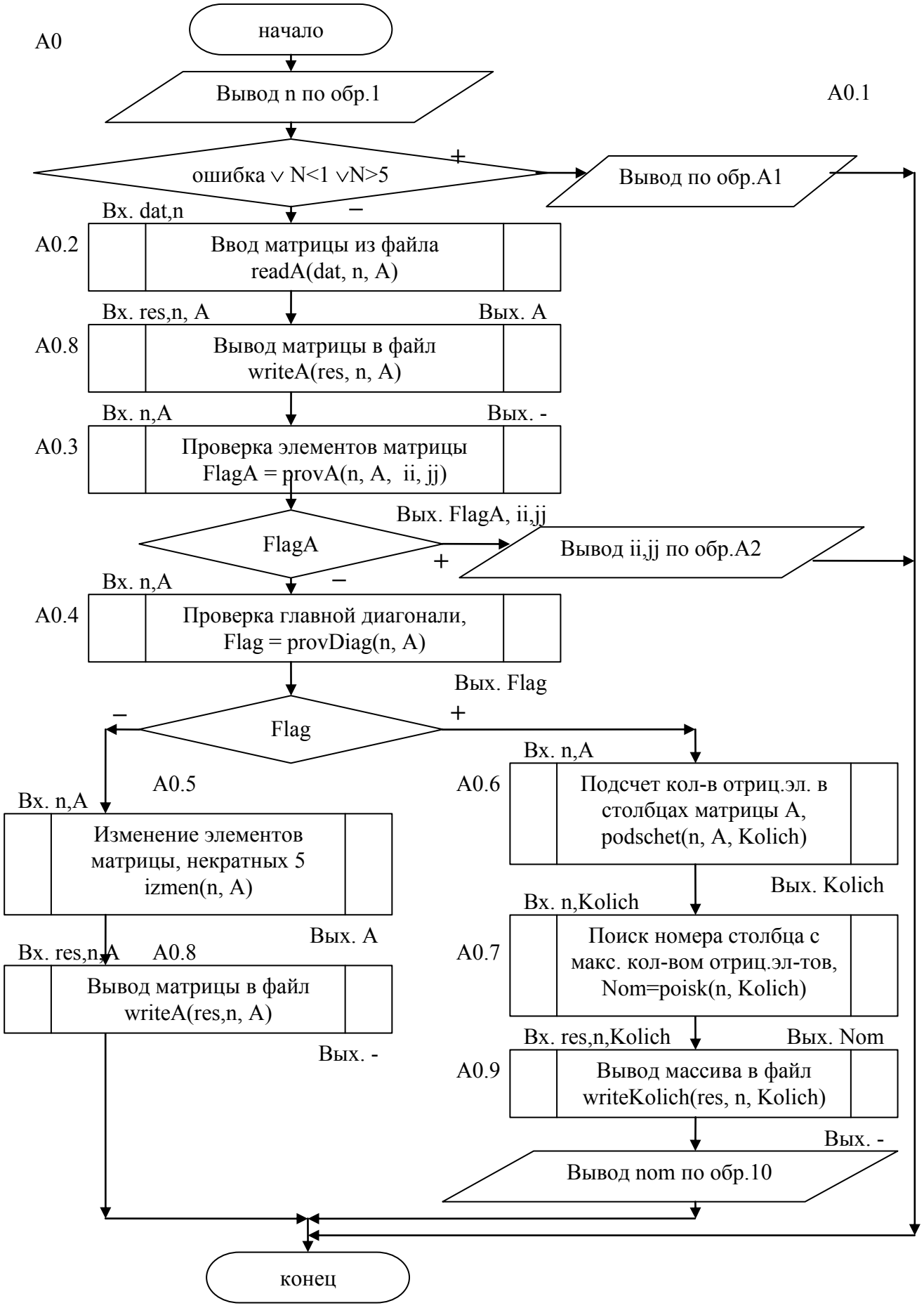
И разделим нашу задачу на десять подзадач:

1. Подзадача А0.1. Ввести из файла (обр.1) и проверить значение количества строк/столбцов (обр.А1)
2. Подзадача А0.2. Ввести матрицу из файла (обр.2).
3. Подзадача А0.3. Проверить значения элементов матрицы (обр.А2)
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 6.2)
9. Подзадача А0.9. Вывести количества отрицательных элементов (обр.7)
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 (А0.1, А0.2, А0.8, А0.9, А0.10) и логику решения задачи вцелом. Остальные пять задач оставим в виде абстракций. Все подзадачи кроме первой и последней выполним в виде процедур и функций, расположенных в отдельном модуле.

11. Алгоритм (на отдельном листе)

Без учета открытия/закрытия файлов


A0
начало
Вывод n по обр.1
A0.1
ошибка ∨ N<1 ∨N>5
+
Вывод по обр.A1
−
Вх. dat,n
A0.2
Ввод матрицы из файла
readA(dat, n, A)
Вх. res,n, A
Вых. A
A0.8
Вывод матрицы в файл
writeA(res, n, A)
Вх. n,A
Вых. -
A0.3
Проверка элементов матрицы
FlagA = provA(n, A, ii, jj)
Вых. FlagA, ii,jj
FlagA
+
Вывод ii,jj по обр.A2
−
Вх. n,A
A0.4
Проверка главной диагонали,
Flag = provDiag(n, A)
Вых. Flag
Flag
−
+
A0.5
Вх. n,A
Изменение элементов
матрицы, некратных 5
izmen(n, A)
Вых. A
Вх. res,n,A
A0.8
Вывод матрицы в файл
writeA(res,n, A)
Вых. -
Вх. n,A
A0.6
Подсчет кол-в отриц.эл. в
столбцах матрицы A,
podschet(n, A, Kolich)
Вых. Kolich
Вх. n,Kolich
A0.7
Поиск номера столбца с
макс. кол-вом отриц.эл-тов,
Nom=poisk(n, Kolich)
Вх. res,n,Kolich
Вых. Nom
A0.9
Вывод массива в файл
writeKolich(res, n, Kolich)
Вых. -
Вывод nom по обр.10
конец

11. Программа на С (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 с основной функцией main  **********
#include "unit1.h"

int main(int argc, char * argv[]){
// описание переменных
    int A[Nmax][Nmax], Kolich[Nmax];
    int n, Nom, ii, jj;
    int flag, flagA;  // 0|1   лог
    FILE *dat, *res;

   dat=fopen(argv[1],"rt"); if (!dat) {printf("Ошибка 3 открытия вх.файла: %s", argv[1]); return 3;}
   res=fopen(argv[2],"wt"); if (!res) {printf("Ошибка 4 создания вых.ф.: %s", argv[2]); fclose(dat); return 4;}
   printf("%15s%s\n"," ", "Программа с одним модулем"); //Обр.3
                        //ввод n и его проверка----------------------------------------------------------------------------
   if (fscanf(dat, &n)!=1|| n<1 || n>Nmax) { //Обр.1
     printf(res, "Некорректное число строк.\nЗадача не решалась"); //Обр.A1
     fclose(dat);  fclose(res);
     return 1;
   }
                        //ввод и вывод матрицы A ---------------------------------------------------------------------------
   readA(dat, n, A); //Обр.2
   fclose(dat);
   printf("Исходная матрица из %d строк и %d столбцов:", n,n); //Обр.4
   writeA(res, n, A); //Обр.5
                         //----------------------решение задачи ------------------------------------------------------------
    FlagA=provA(n, A, &ii, &jj); // Проверка элементов матрицы
   if (FlagA ){
     printf (res, "Некорректный элемент матрицы");  //Обр.A2
     printf(res, "A[%d,%d] = %d", ii, jj, A[ii-1][jj-1]);
     printf (res, "Задача не решалась");
     fclose(res);
     return 2;
   }
                           //-------------------------------------------------------------------------------------------------
   Flag = provDiag(n, A);    //Проверка главной диагонали
   if (flag ) { //-----------------------------если есть кратный 5-----------------------------------------------------
      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
      printf(res, "Количество отрицательных элементов по столбцам:"); //Обр.7.1
      writeKolich(res, n, Kolich); //Обр.7.2
        //Вывод Nom по обр.8
       printf(res, "Максимальное их число в столбце %d", Nom); //Обр.8
    }
   else { //-----------------------------------------если нет кратных 5-----------------------------------
      izmen(n, A); //Изменение элементов матрицы, некратных 5
        //Вывод матрицы в файл
      printf(res, "Измененная матрица"); //Обр.6.1
      writeA(res,n, A); //Обр.6.2
    };
                           //-------------------------------------------------------------------------------------------------
   fclose(res);
   printf("Ok");
   return 0;
}
```

```
//**************Модуль unit1.h со всеми процедурами и функциями*********************
#ifndef _UNIT1
#define _UNIT1

#include <stdio.h>
```

```
#define Nmax 5
#define Amax 100

    //Ввод матрицы из файла, файл уже открыт
void readA(FILE* dat, const int n, int A[Nmax][Nmax]);
    //Вывод матрицы в файл, файл уже открыт
void writeA(FILE* res, const int n, int A[Nmax][Nmax]);
        //Вывод массива в файл, файл уже открыт
void writeKolich(FILE* res, const int n, int Kol[Nmax]);
```

Таблица соответствия параметров

Формальные параметры	Фактические параметры	Тип
res	res	FILE*
n	n	int
Kol	Kolich // &Kolich[0]	int*

```
    //Проверка элементов матрицы
int provA(const int n, int A[Nmax][Nmax], int*_ii, int *_jj);
```

Таблица соответствия параметров

Формальные параметры	Фактические параметры	Тип
n	n	int
A	A	int [][Nmax]
Лок.FlagA, результат	FlagA	int
_ii	&ii	int*
_jj	&jj	int*

```
    //Проверка главной диагонали
int provDiag(const int n, int A[Nmax][Nmax]);
    //Изменение элементов матрицы, некратных 5
void izmen(const int n, int A[Nmax][Nmax]);
    //Подсчет кол-в отриц.эл. в столбцах матрицы A
void podschet(const int n, int A[Nmax][Nmax], int Kol[]);
```

Таблица соответствия параметров

Формальные параметры	Фактические параметры	Тип
n	n	int
A	A	int [][Nmax]
Kol	Kolich // &Kollich[0]	int *

```
    //Поиск номера столбца с макс. кол-вом отриц.эл-тов
int poisk(const int n, int Kol[]);
```

Таблица соответствия параметров

Формальные параметры	Фактические параметры	Тип
n	n	int
Kol	Kolich // &Kollich[0]	int *
Лок.Nom, результат функ.	Nom	int

```
#endif
```

```
//****************Модуль unit1.c  со всеми процедурами и функциями**********************
#include <stdio.h>
#include <math.h>
#include "unit1.h"

     //Ввод матрицы из файла, файл уже открыт
void readA(FILE* dat, const int n, int A[Nmax][Nmax]){
  for (int i=0; i<n; i++)
    for (int j=0; j<n; j++)
      fscanf(dat, "%d", &A[i][j]);
  return;
}
     {Вывод матрицы в файл, файл уже открыт }
     //Вывод матрицы в файл, файл уже открыт
void writeA(FILE* res, const int n, int A[Nmax][Nmax]);
  for (int i=0; i<n; i++){
    for (int j=0; j<n; j++)
      fprintf(res, "%4d ", A[i][j]);
    fprintf(res,"\n");
  }
  return;
}
          //Вывод массива в файл, файл уже открыт
void writeKolich(FILE* res, const int n, int Kol[Nmax]);
  for (int j=0; j<n; j++)
    fprintf(res, "%d ", Kol[j]);
    fprintf(res, "\n");
  }
  return;
}

     //Проверка элементов матрицы
int provA(const int n, int A[Nmax][Nmax], int*_ii, int *_jj){
    int FlagA;
    printf(" Заглушка - Проверка элементов матрицы "'); //Заглушка
    // *_ii=1; *_jj=3; FlagA=1;  //тест 3
    // *_ii=1; *_jj=1; FlagA=1; {тест 4}
    FlagA=0; {остальные тесты}
    reurn FlagA;
}
     //Проверка главной диагонали
int provDiag(const int n, int A[Nmax][Nmax]){
    int  Flag;
    printf(" Заглушка - Проверка главной диагонали "); //Заглушка
    // Flag=1; //тест 7,8
    Flag=0; //остальные тесты
  return Flag;
}
     //Изменение элементов матрицы, некратных 5
void izmen(const int n, int A[Nmax][Nmax]){
    printf(" Заглушка - Изменение элементов матрицы, некратных 5"); //Заглушка
    for (int i=0; i<n; i++)
      for(int j=0; j<n; j++) A[i][j]=0;
    // A[0][1]=5; A[2][0]=10; A[2][1]=-5; //тест 5
    ; //остальные тесты
    return;
}
```

```
    //Подсчет кол-в отриц.эл. в столбцах матрицы A
void podschet(const int n, int A[Nmax][Nmax], int Kol[]);
   printf(" Заглушка - Подсчет кол-в отриц.эл. в столбцах матрицы A"); {Заглушка}
   // Kol[0]=0; Kol[1]=2; Kol[2]=1; //тест 7
   for (int i=0; i<n; i++) Kol[i]=0; {остальные тесты} ;
   return;
}

    //Поиск номера столбца с макс. кол-вом отриц.эл-тов
int poisk(const int n, int Kol[]){
   int Nom:;
    printf(" Заглушка - Поиск номера столбца с макс. кол-вом отриц.эл-тов"); //Заглушка
   // Nom=2; //тест 7
   // Nom=1; //тест 8
   Nom=5 ;
   return Nom;
}
```

Простой без проверок ввод и вывод раскрываем сразу.
Далее рассмотрим каждую из (нераскрытых) 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 1 и найти номер строки (ii) и столбца (jj) первого такого элемента, иначе присвоить FlagA значение 0.

3.Примеры.

Тест 3 есть некорректный элемент A[1][3]=-101, FlagA= 1
Тест 4 есть некорректный элемент A[1][1]=101, FlagA= 1
Тесты 5-7 нет некорректных элементов, FlagA= 0

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	A	Заданная матрица, $\|A_{ij}\|<=100$	цел	Двухмерный массив (5x5)
	n	число строк и столбцов в матрице A, $0 < n \le 5$	цел	простая переменная
Выходные данные	ii	Строка с некорр. A[ii,jj], $0<ii\le 5$	цел	простая
	jj	Столбец с некорр. A[ii,jj], $0<jj\le 5$	цел	простая
	FlagA	= 1, если есть некорректные Aij, в противном случае 0	Лог|/ цел	простая переменная
Промежу-точные данные	i	Индекс текущей строки, $0 \le i \le 5$	цел	простая переменная
	j	Индекс текущего столбца, $0 \le j \le 5$	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= 1
Тест 4 есть некорректный элемент A[1][1]=101, FlagA= 1
Тесты 5-9 нет некорректных элементов, FlagA= 0

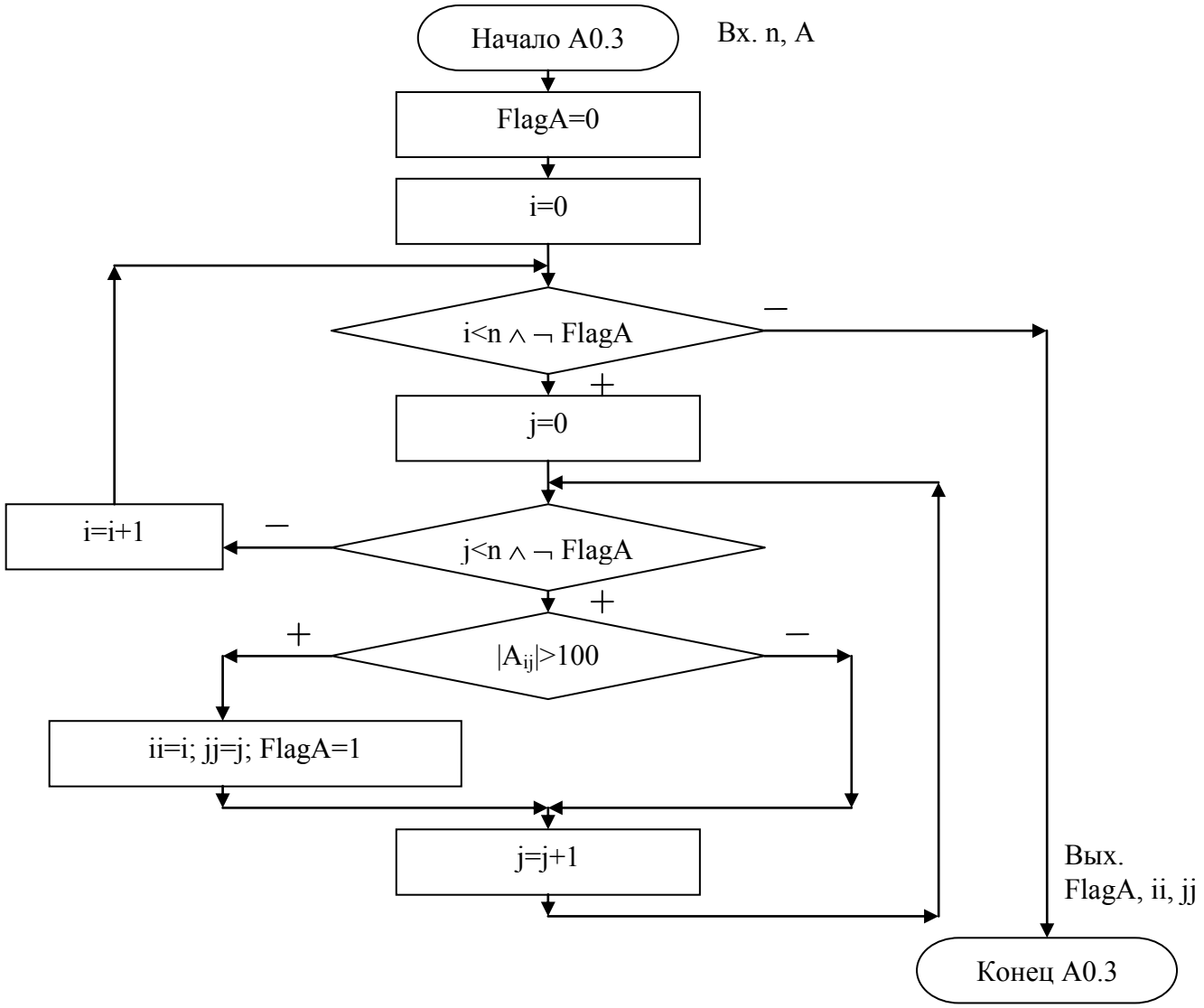
9. Метод (задача типа 1) из файла VseOdin.doc)

Пусть FlagA= { 1, если есть в матрице элемент $|A_{i,j}|>100$;
0, если нет такого элемента

Предположим сначала, что такого элемента в матрице нет (FlagA=0).
Затем будем просматривать строки, начиная с первой (i=0)
Пока не просмотрены все (i<n) и не найден такой элемент (FlagA==0)
    В каждой строке будем просматривать элементы, начиная с первого (j=0)
    Пока не просмотрены все (j<n) и не найден такой элемент (FlagA==0)
        Если рассматриваемый элемент A_{ij} по модулю больше 100
        То
            искомый элемент найден! (FlagA=1)
            Запоминаем его местонахождение (ii=i+1; jj=j+1)
        Переходим к следующему элементу в строке (j=j+1)
    Переходим к следующей строке (i=i+1)

Найденные значения FlagA, ii, jj будут искомыми.

10. Алгоритм



11. Программа на С. Функция с результатом

Таблица соответствия параметров

Формальные параметры	Фактические параметры	Тип
n	n	int
A	A	int [][Nmax]
Лок.FlagA, результат	FlagA	int
_ii	&ii // адрес ii	int*
_jj	&jj // адрес jj	int*

```
//Проверка элементов матрицы. FlafA – локальная переменная и результат, индексы по адресу переданы
int provA(const int n, int A[Nmax][Nmax], int*_ii, int *_jj){
 int FlagA=0;
 i=0;
 while (i<n && !FlagA ){
    j=0;
    while (j<n && !FlagA ){
       if (abs(A[i][j]) > Amax ) {
          *_ii=i; *_jj=j; // изменяем значения по переданным адресам
          FlagA=1;
       }
       j++;
    }
    i++;
 }
 return FlagA;
}
```

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= 0

Тест 7,8,9 есть на главной диагонали матрицы элементы, кратные 5, Flag= 1

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	A	Заданная матрица, $\|A_{ij}\|<=100$	цел	Двухмерный массив (5x5)
	n	число строк и столбцов в матрице A, $0 < n \le 5$	цел	простая переменная
Выходные данные	**Flag**	= 1, если есть кратные 5 на гл.диагонали, в противном случае 0	лог/ цел	простая переменная

Промежу-точные данные	i	Индекс текущей строки, $0 \le i \le 5$	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= 0

Тест 7,8,9 есть на главной диагонали матрицы элементы, кратные 5, Flag= 1

9. Метод (задача типа 1) из файла VseOdin.doc)

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

Пусть Flag= { 1, если есть в на гл.диагонали элемент $A_{i,i}$, кратный 5;

0, если нет такого элемента

Предположим сначала, что такого элемента в матрице нет (Flag=0).

Затем будем просматривать строки, начиная с первой (i=0)

Пока не просмотрены все (i<n) и не найден такой элемент (FlagA==0)

{ В каждой строке будем проверять один элемент A_{ii}

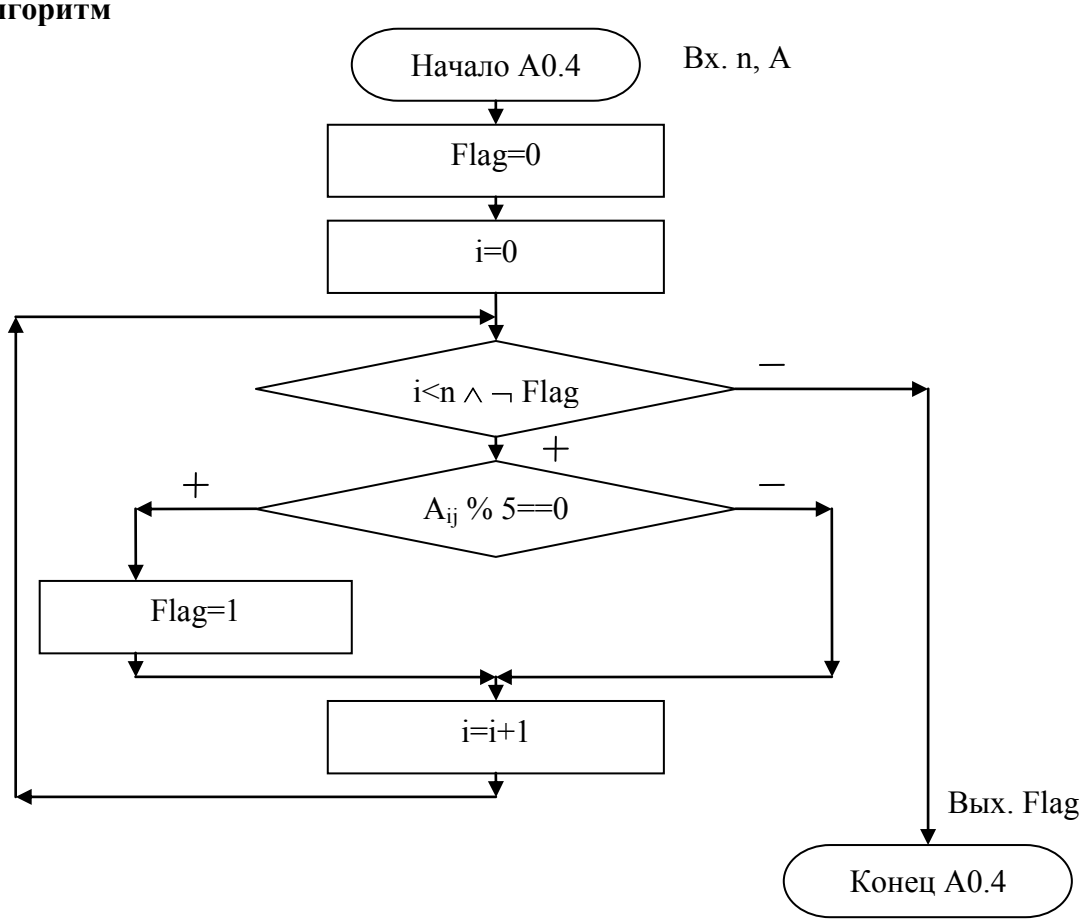
Если остаток от деления A_{ij} на 5 равен 0

То искомый элемент найден! (Flag=1)

Переходим к следующей строке (i=i+1)

Найденное значение Flag будет искомым.

10. Алгоритм



11. Программа на С. Функция с результатом

```
    //Проверка главной диагонали.  Flag – локальная переменная и результат функции
int provDiag(const int n, int A[Nmax][Nmax]) {
   int Flag=0, i=0;

   while (i<n &&!Flag ){
    if (A[i][i] % 5 == 0 ) Flag=1;   // или можно Flag= A[i][i] % 5 == 0;
    i++;
   }

   return Flag;
}
```

Абстракция А0.5

1.Условие. Заменить в матрице все элементы, некратные пяти, нулями.

2.Уточненная постановка задачи.

Дана **целочисленная квадратная** матрица **A** из **n** строк и столбцов. Изменить матрицу, заменив все элементы некратные пяти нулями.

3.Примеры.

Тест 5 есть и кратные и некратные пяти

Тест 6 нет кратных пяти – заменяются все

4. Таблица данных

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

Промежу-точные данные	i	Индекс текущей строки, $0 \le i \le 5$	цел	простая переменная
	j	Индекс текущего столбца, $0 \le j \le 5$	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 некратные пяти

Тест 6 нет кратных пяти – заменяются все

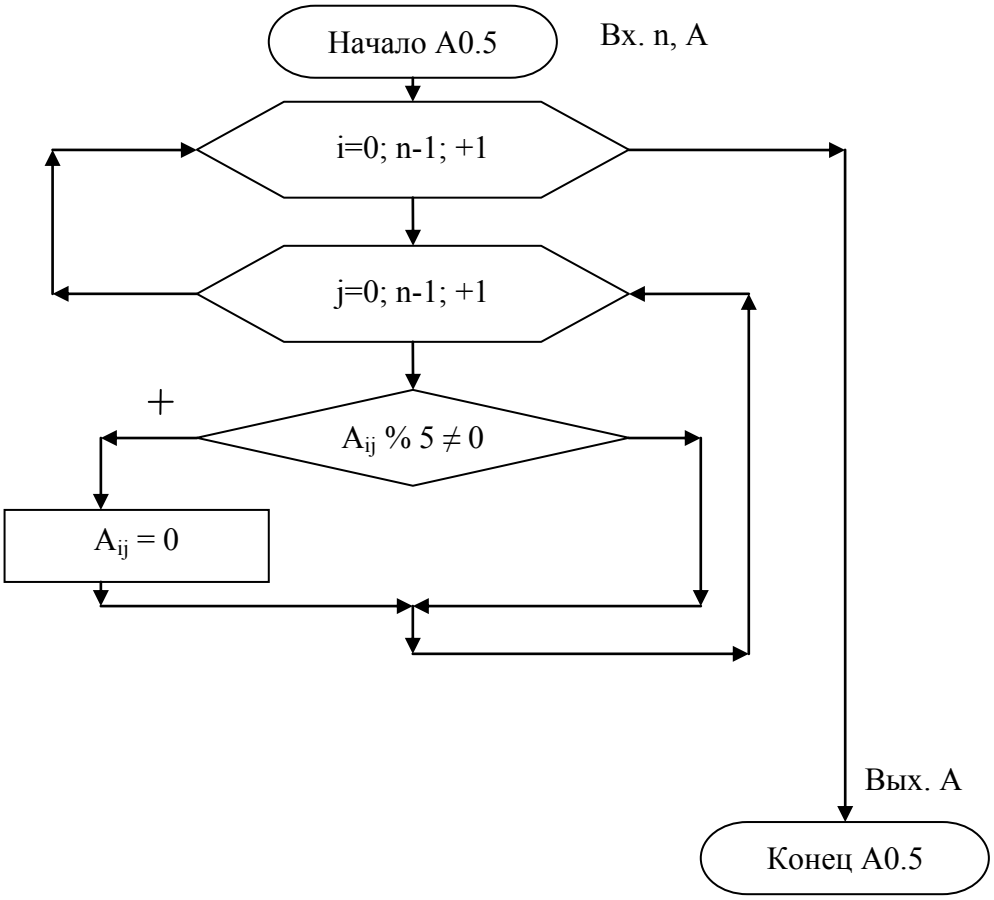
9. Метод

Будем просматривать строки, начиная с первой по последнюю (i=0;n-1;+1)

- В каждой строке будем просматривать все элементы, начиная с первого до последнего (j=0; n-1;+1)
  - Каждый элемент будем проверять
  - Если остаток от деления A_{ij} на 5 не равен 0
    - То меняем на ноль (A[i][j]=0)
  - Переходим к следующему элементу в строке (j=j+1)
- Переходим к следующей строке (i=i+1)

Матрица изменена

10. Алгоритм



11. Программа на С. Функция без результата

```
    //Изменение элементов матрицы, некратных 5. Изменение элементов по адресу A
void izmen(const int n, int A[Nmax][Nmax]){
   for (int i=0; i<n; i++)
     for (int j=0; j<n; j++)
       if (A[i][j] % 5 !=0) A[i][j]=0;
   return;
}
```

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

Тест 9 во всех столбцах есть отрицательные, в каждом больше, чем в предыдущем

4. Таблица данных

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

Промежу-точные данные	j	Индекс текущего столбца, $0 \le j \le 5$	цел	простая переменная
	i	Индекс элемента в столбце, $0 \le i \le 5$	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

Тест 9 во всех столбцах есть отрицательные, в каждом больше, чем в предыдущем

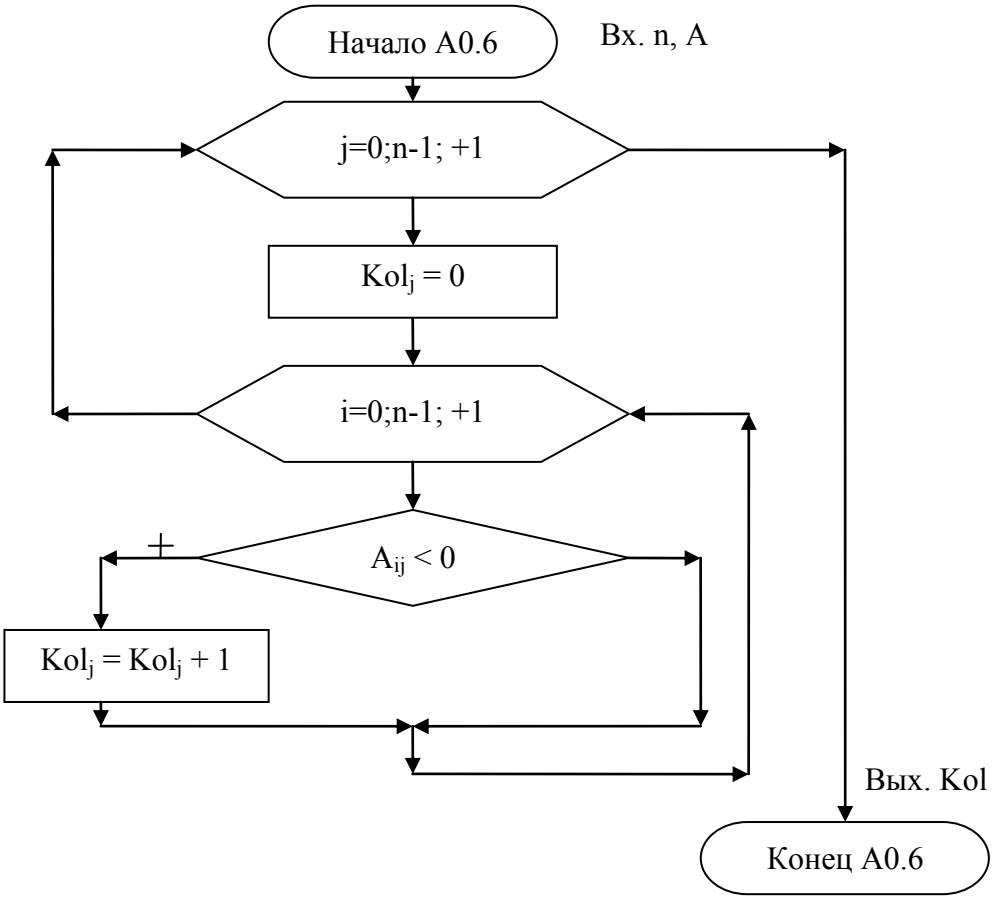
9. Метод

Будем просматривать столбцы, начиная с первого до последнего (j=0;n-1;+1)

- В каждой столбце будем считать количество Kol_j,
- Начальное значение 0 (Kol[j]=0;)
- Проверим элементы в столбце с первого до последнего (i=0;n-1;+1)
  - Если элемент отрицательный (A[i][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 С. Функция без результата возвращаемого

```
//Подсчет кол-в отриц.эл. в столбцах матрицы А. Изменяем значения по адресу Kol
void podschet(const int n, int A[Nmax][Nmax], int Kol[]);
```

Таблица соответствия параметров

Формальные параметры	Фактические параметры	Тип
n	n	int
A	A	int [][Nmax]
Kol	Kolich // &Kollich[0]	int * |int []

```
for (int j=0; j<n; j++){
  Kol[j]=0;
  for( i=0; i<n; i++)
    if (A[i][j]<0) Kol[j]++;
return;
}
```

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+1**) **первого** элемента с максимальным значением.

3.Примеры.

Тест 7 Все элементы разные, Nom=2

Тест 8 Все элементы одинаковые, Nom=1

Тест 9 Последний – максимальный, Nom=5

4. Таблица данных

Класс	Имя	Описание (смысл), диапазон, точность	Тип	Структура
Входные данные	Kol	Количество отрицательных элементов в каждом столбце, $0 \le kol_j \le 5$	цел	Одномерный массив (5)
	n	число строк и столбцов в матрице A, $0 < n \le 5$	цел	простая переменная
Выходные данные	Nom	Возвращаем **номер** (Nom+1), В коде **индекс** первого столбца с максимальным количеством отрицательных элементов, $0 \le Nom \le 5$	цел	простая переменная

Промежу-точные данные	k	Индекс текущего элемента, $0 \le k \le 5$	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, Nom=1

Тест 9 Последний – максимальный, Nom=5

9. Метод

Nom – индекс в коде, в ответе – номер (Nom+1).

Пусть максимальное значение находится в первом элементе массива. (Nom=0)

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

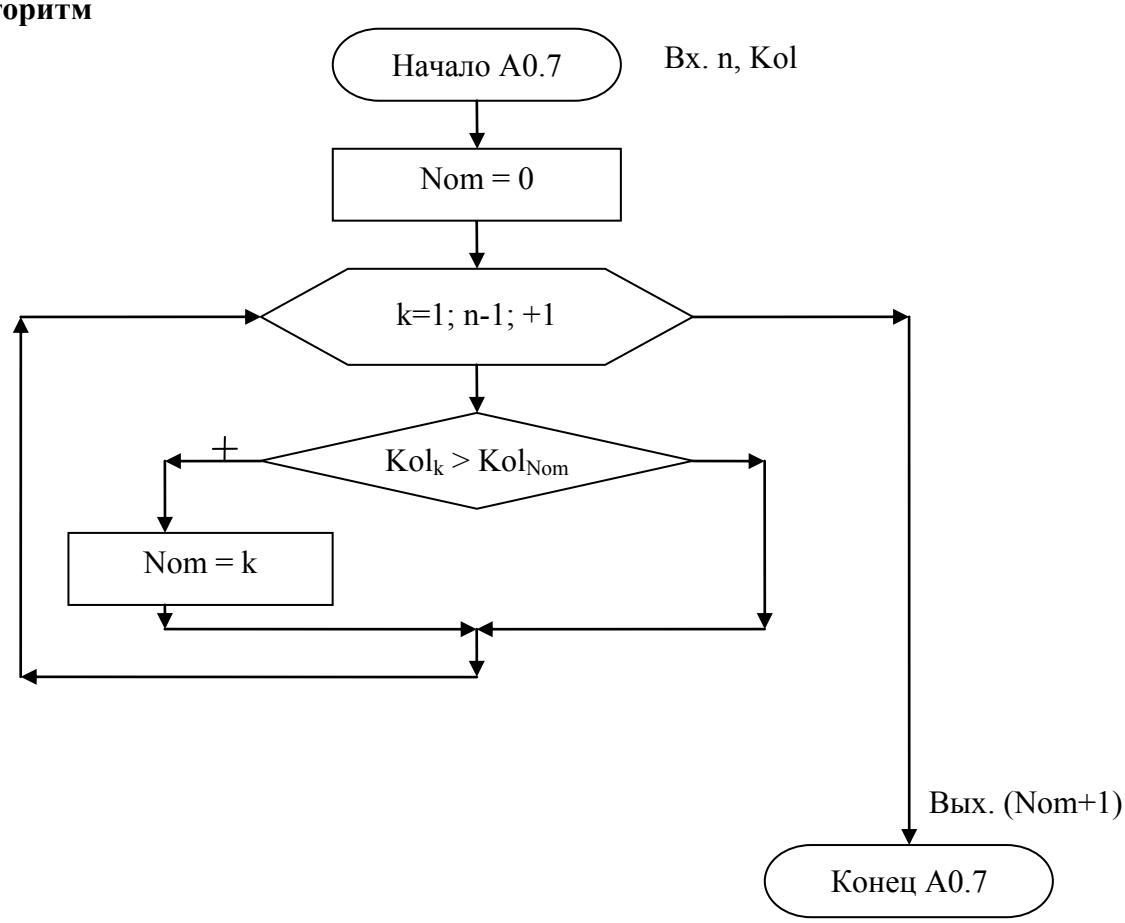
    Если найдем элемент больше (Kol[k]>Kol[Nom-1]),

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

    Переходим к следующему элементу

Текущее значение (Nom+1) – искомое

10. Алгоритм



11. Программа на С. Функция с возвращаемым результатом

```
    //Поиск индекса первого столбца с макс. кол-вом отриц.эл-тов, возвращение номера
int poisk(const int n, int Kol[]){
  int Nom=0;
  for (int k=1; k<n; k++)
    if (Kol[k]> Kol[Nom]) Nom=k;

  return Nom+1;
}
```

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.

Структурные тесты. Выявление случаев, не покрываемых функциональными тестами (1-10):

Тест№ / Структура	1	2	3	4	5	6	7	8	10	**9**	**11**	**12**	**13**
if (!dat)													
dat ==NULL									**10а**				
dat =/= NULL	+	+	+	+	+	+	+	+		+	+	+	+
if (!res)													
res ==NULL									**10б**				
res =/= NULL	+	+	+	+	+	+	+	+		+	+	+	+
if (fscanf(dat, &n)!=1|| n<1 || n>Nmax)	(ввод и при успехе проверка n)								Нет ввода				
fscanf(dat, &n)!=1	1в								-	-			
fscanf(dat, &n)==1 далее:									-	-			
N<1 истина N>5 истина	Не возможно одновременное выполнение этих условий								-	-			
N<1 истина N>5 ложь (N=5)	Не возможно одновременное выполнение этих условий								-	-			
N<1 истина N>5 ложь (N<5)	0 1а, -1 1б	-	-	-	-	-	-	-	-	-	-		
N<1 ложь (N=1) N>5 истина	Не возможно одновременное выполнение этих условий												
N<1 ложь (N>1) N>5 истина	-	N=6	-	-	-	-	-	-	-	-	-		
N<1 ложь (N=1) N>5 ложь (N=5)	Не возможно одновременное выполнение этих условий												
N<1 ложь (N>1) N>5 ложь (N=5)			N=5							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	N=3				2	
Ввод/вывод матрицы for(int i=0; i<n-1; i++)	НЕТ ВВОДА ИПРОВЕРКИ МАТРИЦЫ	НЕТ ВВОДА ИПРОВЕРКИ МАТРИЦЫ											
0 раз (i=0>=n-1)			Не возможно, n≥1										
1 раз (n=1)				1					-		1		1
более 1 раза (n>1)			5		3	3	3	3	-	5		2	
for(int j=0; j<n-1; j++)													
0 раз (j=0>=n-1)			Не возможно, n≥1										
1 раз (n=1)				1					-	-	1		1
более 1 раза (n>1)			5		3	3	3	3	-	5-		2	
Проверка элементов while (i<n && !FlagA)													
(i<n) истина (i<n) FlagA = 0 (!FlagA истина)			+	+	+	+	+	+		+			
(i<n) истина (i<n) FlagA =1 (!FlagA ложь)			+	-	-	-	-	-		-			
(i<n) ложь (i=n) FlagA = 0 (!FlagA истина)			-	-	+	+	+	+		+			
(i<n) ложь (i=n) FlagA =1 (!FlagA ложь)			-	+	-	-	-	-		-			
i>n невозможно при шаге +1													
while (j<n && !FlagA)													
(j<n) истина (j<n) FlagA =0 (!FlagA истина)			+	-	+	+	+	+		+			
(j<n) истина (j<n) FlagA =1 (!FlagA ложь)			+	-	-	-	-	-		-			
(j<n) ложь (j=n) FlagA =0 (!FlagA истина)			-	-	+	+	+	+		+			
(j<n) ложь (j=n) FlagA =1 (!FlagA ложь)			-	+	-	-	-	-					
j>n невозможно при шаге +1													
if (abs(A[i,j]) > 100)													
abs(A[i,j]) > 100			-101	101									
abs(A[i,j]) < 100			-		+	+	+	+	-	+	+	+	

abs(A[i,j]) = 100			100 -100						-				
if (FlagA)													
FlagA =1 истина			+	+					+				
FlagA =0 ложь					+	+	+	+		+	+	+	
Проверка гл.диагонали while (i<n && !Flag)			З А Д А Ч А	З А Д А Ч А									
(i<n) истина(i<n) Flag =0 (!Flag истина)					+	+	+	+					
(i<n) истина (i<n) Flag =1 (!Flag ложь)					-	-	+	+					
(i<n) ложь (i=n) Flag =0 (!Flag истина)			Н	Н	+	+	-	-					
(i<n) ложь (i=n) Flag =1 (!Flag ложь)			Е	Е	-	-	-	-		+			
if (A[i][i] % 5 == 0)			Р	Р									
A[i][i] % 5 == 0			Е	Е	-	-	+	+		+	+	+	
A[i][i] % 5 ≠ 0			Ш	Ш	+	+	+	-		+		+	
if (Flag)			А	А									
Flag истина 1			Л	Л			+	+		+	+	+	
Flag ложь 0			А	А	+	+				+			
Изменение матрицы for (i=0;i<n;i++)			С Ь	С Ь									
0 раз (i=0>=n)					Не возможно, n≥1								
1 раз (n=1)													+
Более 1 раза (n>1)					3	3							
for(j=0;j<n;j++)													
0 раз (j=0>=n)					Не возможно, n≥1								
1 раз (n=1)													+
более 1 раза (n>1)					3	3							
Вывод изменен. матрицы for (i=0;i<n;i++)					Не возможно, n≥1								
0 раз (i=0>=n)													+
1 раз (n=1)					3	3							
Более 1 раза (n>1)													
for(j=0;j<n;j++)													
0 раз (j=0>=n)					Не возможно, n≥1								
1 раз (n=1)													+
более 1 раза (n>1)					3	3							
Подсчёт количеств отриц. for (i=0;i<n;i++)													
0 раз (i=0>=n)					Не возможно, n≥1								-
1 раз (n=1)											+		
Более 1 раза (n>1)							3	3				2	-
for(j=0;j<n;j++)													
0 раз (j=0>=n)					Не возможно, n≥1								
1 раз (n=1)											+		-
более 1 раза (n>1)							3	3				2	-
if (A[i][j]<0)													
A[i][j]<0							+			+		+	-
A[i][j]>0							+	+		+	+	+	-
A[i][j]==0							+	+		+		+	-
Поиск столбца с макс. for (k=1;k<n; k++)													
0 раз (k=1>=n)					-	-					+	+	-
1 раз (n=2)					-	-						+	-
более 1 раза (n>2)					-	-	3	3					-
if (Kol[k]> Kol[Nom])													
`Kol[k]> Kol[Nom]`							+	-		+	-	+	
`Kol[k]= Kol[Nom]`							-	+			-		
`Kol[k]< Kol[Nom]`							+	-			-		
`Вывод кол-ва Kolich` for(j=0;j<n;j++)													
0 раз (j=0>=n)					Не возможно, n≥1								

1 раз (n=1)											1		-
более 1 раза (n>1)							3	3				2	-

Выявлено 9 строк в таблице, непокрытых имеющимися 10 тестами. Добавим еще 3 теста 11,12,13.

8. Недостающие структурные тесты

№ теста	Входные данные	Ожидаемый результат	Смысл теста
11	n=1; $A = (5)$	Kolich = (0) Nom=1	Единственный элемент главной диагонали кратен 5 Максимум без сравнений
12	n=2; $A = \begin{pmatrix} 1 & 0 \\ 5 & -5 \end{pmatrix}$	Kolich = (0 1) Nom=2	На главной диагонали есть кратный 5 элемент, Во втором столбце отрицательных элементов больше Максимум 1 сравнение
13	n=1; $A = (1)$	$A = (0)$	Единственный элемент главной диагонали НЕ кратен 5, одна замена