



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)**

ФАКУЛЬТЕТ
КАФЕДРА

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

О т ч е т

**по домашнему заданию №2(часть 2)
варианта №7**

Название лабораторной работы:

Обработка матриц

Дисциплина:

Основы программирования

Студент гр. **ИУ6-12**

(Подпись, дата)

Векшин Роман

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Черноусова Татьяна Геннадьевна

(И.О. Фамилия)

Часть 2. Обработка матриц

Составить программу, которая в квадратной матрице $F(n,n)$ $n \leq 8$ находит столбцы с равными суммами элементов. Дополнительных массивов для сохранения значений сумм не использовать. Вывести исходную матрицу, номера найденных столбцов и суммы их элементов. Если требуемые столбцы отсутствуют, то выдать соответствующее сообщение.

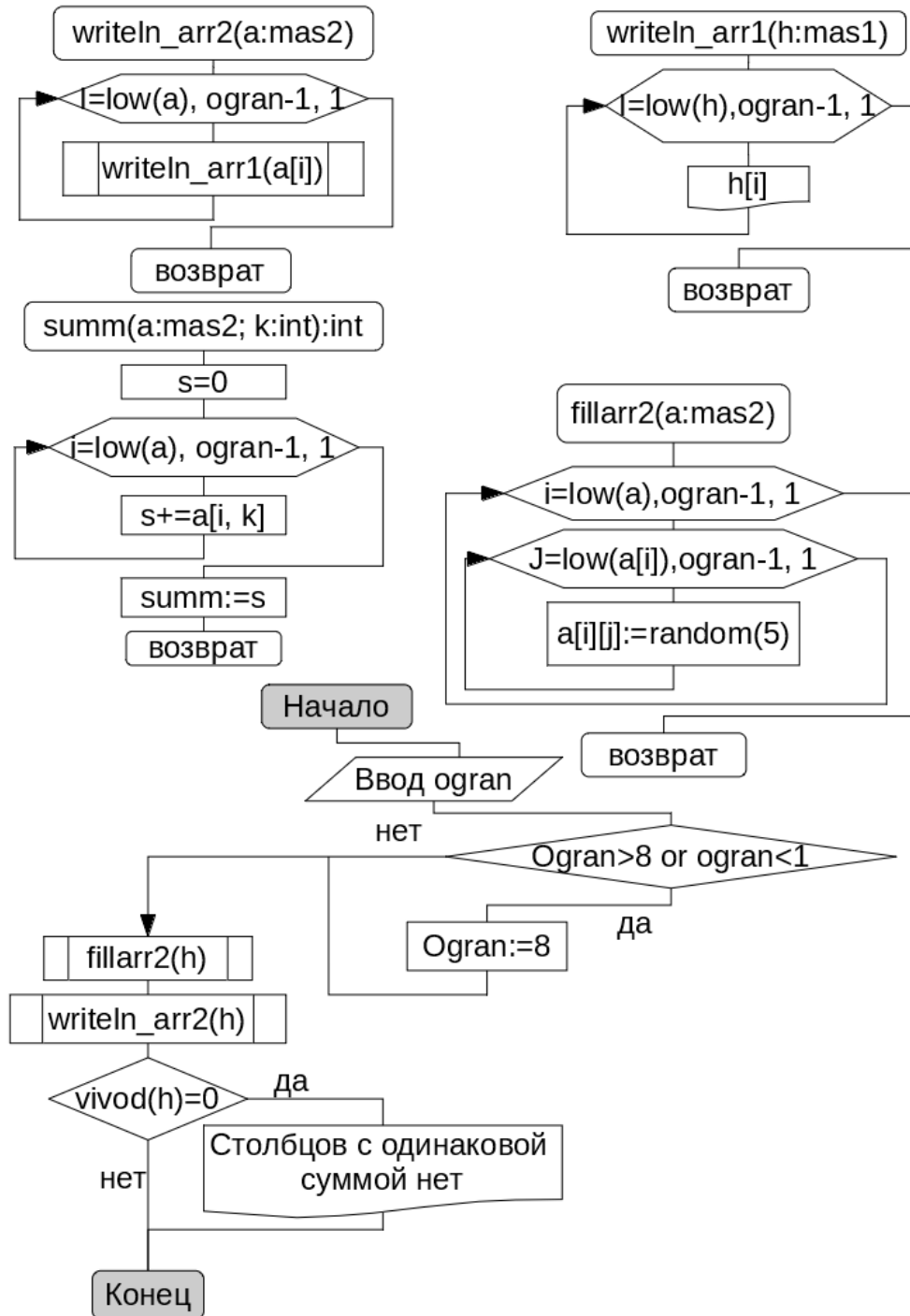


Рисунок 1-Схема алгоритма программы

Главная часть:

organ-переменная, хранящая размер массива. Вводится пользователем

h-двумерный массив

writeln_arr2:

a-локальная переменная двумерного массива

writeln_arr1:

h-локальная переменная одномерного массива

fillarr2:

i, j-переменные циклов для заполнения
a-локальная переменная двумерного массива

summ:

a-локальная переменная двумерного массива
k-номер столбца в массиве для подсчёта суммы
i-переменная цикла (номер строки)
s-переменная для хранения суммы

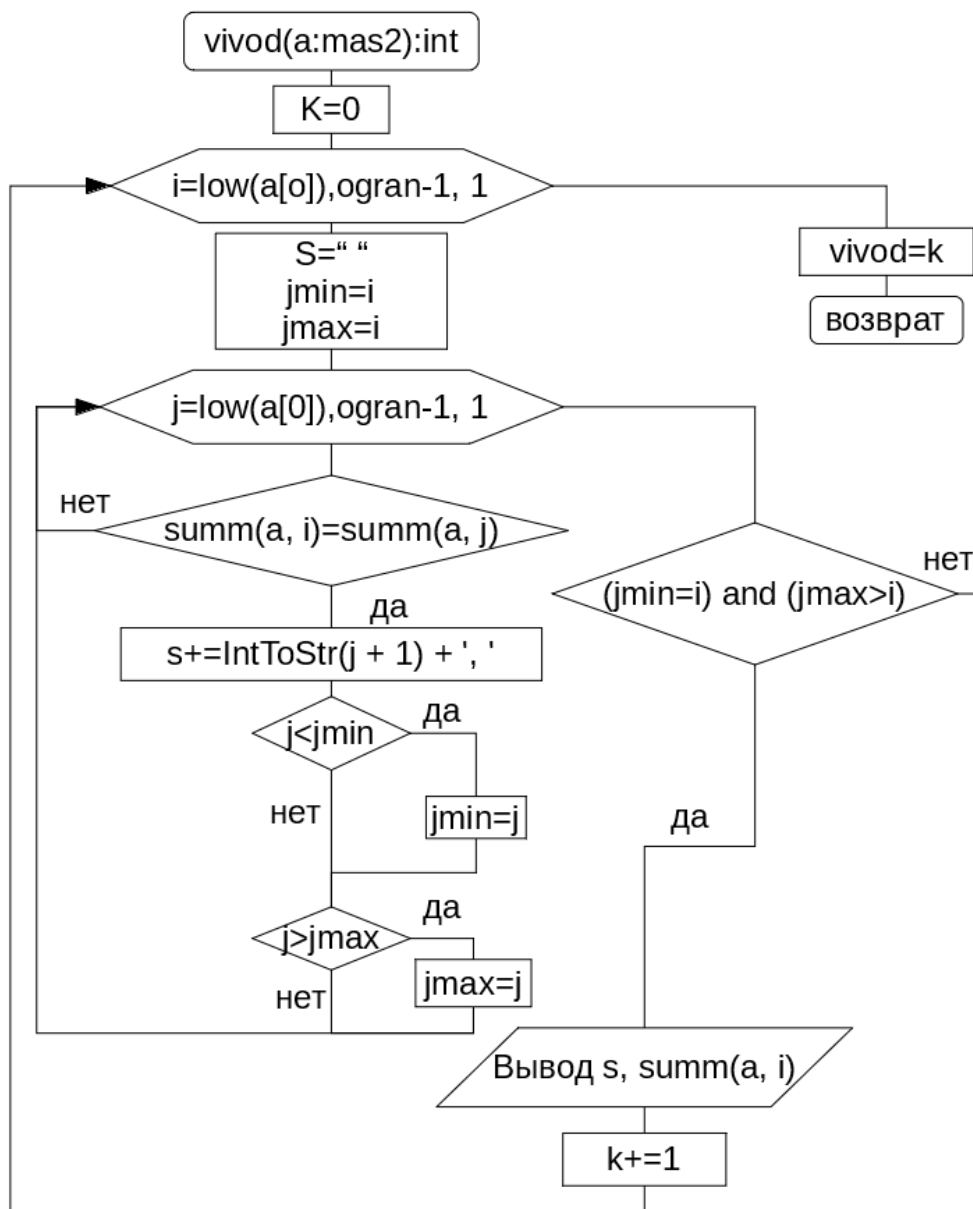


Рисунок 2-Схема алгоритма программы

k-переменная для подсчёта «пар» строк с одинаковой суммой

i-переменная цикла (индекс столбца)

j-переменная цикла, индекс столбца для сравнения суммы с суммой столбца i

jmin-в этой переменной записывается минимальный номер столбца с нужной нам суммой

jmax-в этой переменной записывается максимальный номер столбца с нужной нам суммой

s-строка, хранящая в себе номера столбцов с одинаковой суммой

Код программы

```
program project1;

{$APPTYPE CONSOLE}
uses
  SysUtils;

const
  n = 8;

type
  mas1 = array[0..n - 1] of integer;
  mas2 = array[0..n - 1] of mas1;
var
  ogran: byte;
  ///Построчный вывод матрицы
  procedure writeln_arr2(const a: mas2);

  ///Процедура вывода одномерного массива
  procedure writeln_arr1(const h: mas1);
  var
    i: byte;
  begin
    for i := low(h) to ogran - 1 do
      Write(h[i]: 2);
      writeln;
    end;

  var
    i: byte;
  begin
    Write(' ');
    for i := 1 to ogran do
      Write(i: 2);
      writeln();
    for i := low(a) to ogran - 1 do
      begin
        Write(i + 1: 2, '|');
        writeln_arr1(a[i]);
      end;
    end;
  end;

  ///Возвращает сумму элементов столбца k
  function summ(const a: mas2; const k: integer): integer;
  var
    s: integer = 0;
    i: byte;
  begin
    for i := low(a) to ogran - 1 do
      s += a[i, k];
    summ := s;
  end;

  ///Заполнение матрицы
  procedure fillarr2(var a: mas2);
  var
    i, j: byte;
  begin
    randomize;
    for i := low(a) to ogran - 1 do
      for j := low(a[i]) to ogran - 1 do
        a[i][j] := random(10) - 5;
      end;
    end;

  ///Функция выводит информацию о столбцах и возвращает кол-во "пар" с один. суммой
  function vivod(const a: mas2): integer;
  var
    jmin, jmax, k, i, j: integer;
    s: string;
  begin
    k := 0;
    for i := low(a[0]) to ogran - 1 do
```

```

begin
  s := '';
  jmin := i;
  jmax := i;
  for j := low(a[0]) to ogran - 1 do
    if (summ(a, i) = summ(a, j)) then
      begin
        s += IntToStr(j + 1) + ', ';
        if j < jmin then
          jmin := j;
        if j > jmax then
          jmax := j;
      end;

    if (jmin = i) and (jmax > i) then
      begin
        writeln(utf8toansi('Столбцы {', copy(s, 1, length(s) - 2),
          utf8toansi('} с одинаковой суммой: '),
          summ(a, i)));
        k += 1;
      end;
    end;
    vivod := k;
  end;

var
  h: mas2;
begin
  writeln(utf8toansi('Введите размер массива'));
  readln(ogran);
  if (ogran > 8) or (ogran < 1) then
    ogran := 8;
  fillarr2(h);
  writeln(utf8toansi('Сгенерированный массив:'));
  writeln_arr2(h);
  if (vivod(h) = 0) then
    writeln(utf8toansi('Столбцов с одинаковой суммой нет'));
  readln();
end.

```

Тестирование

Таблица 1- Тестирование программы

Исходные данные	Ожидаемый результат	Полученный результат
4 1 0 4	Нет столбцов	Столбцов с одинаковой суммой нет
2 1 0 3 2 1 0 3 2	1 2	Столбцы {1, 2} с одинаковой суммой: 5
0 2 0 4 0 1 4 1 2 3 3 2 1 0 2 1	Нет столбцов	Столбцов с одинаковой суммой нет
0 1 4 4 1 4 4 3 2 4 4 0 0 0 4 1 2 1 2 3 2 4 4 3 2	1, 2, 4	Столбцы {1, 2, 4} с одинаковой суммой: 11
1 4 1 -5 0 1 -4 -2 0 -1 -1 -4 1 2 -1 0 -4 -5 -3 1 3 -3 2 0 2	Нет столбцов	Столбцов с одинаковой суммой нет
1 0 1-2 4 3-2 1 -1 2 2-5-3 0 4 4 1-1 2 0-3 0 1-4 -4 1-1 1-4 2 2-4 4-5-5 4-4-5-5-2 -3 0 2 2 4 2 4-3 0 3 1 2-5 1 4 3 -1-3-3-3-3 3 0-3	1 2 3 4	Столбцы {1, 2} с одинаковой суммой: -3 Столбцы {3, 4} с одинаковой суммой: -1

Примеры работы программы

C:\Users\Asus\Desktop\ОП\ДЗ\ДЗ2\Задач

Сгенерированный массив:

4 1

0 4

Столбцов с одинаковой суммой нет

Рисунок 3-Пример работы программы 1

C:\Users\Asus\Desktop\ОП\ДЗ\ДЗ2\Задача2\proj

Сгенерированный массив:

2 1 0

3 1 2

0 3 2

Столбцы {1, 2} с одинаковой суммой: 5

Рисунок 4-Пример работы программы 2

C:\Users\Asus\Desktop\ОП\ДЗ\ДЗ2\Задач

Сгенерированный массив:

0 2 0 4

0 1 4 1

2 3 3 2

1 0 2 1

Столбцов с одинаковой суммой нет

Рисунок 5-Пример работы программы 3

C:\Users\Asus\Desktop\ОП\ДЗ\ДЗ2\Задача2\project1.ex

Сгенерированный массив:

0 1 4 4 1

4 4 3 2 4

4 0 0 0 4

1 2 1 2 3

2 4 4 3 2

Столбцы {1, 2, 4} с одинаковой суммой: 11

Рисунок 6-Пример работы программы 4

C:\Users\Asus\Desktop\ОП\ДЗ\ДЗ2\Задача2

Введите размер массива

5

Сгенерированный массив:

1 2 3 4 5

1| 1 4 1-5 0

2| 1-4-2 0-1

3| -1-4 1 2-1

4| 0-4-5-3 1

5| 3-3 2 0 2

Столбцов с одинаковой суммой нет

Рисунок 7-Пример работы программы

```
C:\Users\Asus\Desktop\ОП\ДЗ\Д32\Задача2\project1.exe
Введите размер массива
8
Сгенерированный массив:
      1 2 3 4 5 6 7 8
1| 1 0 1-2 4 3-2 1
2|-1 2 2-5-3 0 4 4
3| 1-1 2 0-3 0 1-4
4|-4 1-1 1-4 2 2-4
5| 4-5-5 4-4-5-5-2
6|-3 0 2 2 4 2 4-3
7| 0 3 1 2-5 1 4 3
8|-1-3-3-3-3 3 0-3
Столбцы {1, 2} с одинаковой суммой: -3
Столбцы {3, 4} с одинаковой суммой: -1
```

Рисунок 8-Пример работы программы

Вывод

- 1) Разработана схема алгоритма программы в среде LibreOffice Draw(см. рис. 1, рис. 2).
- 2) Создан код программы по схеме алгоритма в среде Lazarus.
- 3) Проведено тестирование(см. табл. 1).
- 4) Тестирование показало корректность работы программы(см. Рис. 3-8).