

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Пермский национальный исследовательский  
политехнический университет»

**ПРОГРАММНЫЕ СРЕДСТВА  
РЕАЛИЗАЦИИ АЛГОРИТМОВ.  
АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ  
ЗАДАЧ ПО ОБРАБОТКЕ МАССИВОВ**

*Методические указания  
к выполнению лабораторных работ  
по дисциплине «Информатика»*

Издательство  
Пермского национального исследовательского  
политехнического университета  
2015

УДК 004.421(072.8)

П78

Рецензент

канд. техн. наук, профессор *П.Н. Цылёв*  
(Пермский национальный исследовательский  
политехнический университет)

**Программные средства реализации алгоритмов. Алго-**  
П78 ритмизация и программирование задач по обработке массивов: метод. указания к выполнению лаб. работ по дисциплине «Информатика» / сост. И.Н. Шапова. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2015. – 35 с.

Рассмотрены вопросы алгоритмизации и программирования задач по обработке одномерных и двумерных массивов. Приведены блок-схемы типовых алгоритмов обработки массивов.

Методические указания предназначены для студентов горно-нефтяного факультета.

УДК 004.421(072.8)

© ПНИПУ, 2015

## ОГЛАВЛЕНИЕ

Введение .....	4
1. Алгоритмизация и программирование задач по обработке одномерных массивов .....	9
1.1. Вычисление суммы элементов одномерного массива .....	9
1.2. Формирование нового массива из элементов заданного массива, удовлетворяющих заданному условию, и подсчет их количества .....	10
1.3. Объединение двух массивов в один с чередованием исходных элементов .....	10
1.4. Поиск максимального (минимального) элемента в массиве с запоминанием его индекса .....	12
1.5. Упорядочивание массива .....	13
2. Алгоритмизация и программирование задач по обработке двумерных массивов.....	15
2.1. Вычисление суммы всех элементов двумерного массива .....	16
2.2. Вычисление следа матрицы .....	17
2.3. Вычисление сумм элементов для каждой строки матрицы.....	18
2.4. Транспонирование матрицы .....	20
2.5. Преобразование матрицы в одномерный массив .....	20
2.6. Умножение матрицы на вектор .....	21
2.7. Умножение матрицы на матрицу .....	23
2.8. Удаление $k$ -й строки матрицы .....	23
2.9. Включение строки в матрицу .....	23
2.10. Перестановка строк (столбцов) матрицы местами .....	25
Список литературы .....	26
Лабораторные работы.....	27
Лабораторная работа № 1.....	27
Лабораторная работа № 2.....	29
Лабораторная работа № 3.....	30
Контрольные вопросы .....	32
Контрольные задания .....	34

## ВВЕДЕНИЕ

**Массивы.** *Массивы* представляют собой упорядоченную совокупность данных, имеющую одно имя. Каждому элементу массива соответствует выражение порядкового типа (чаще – целое число), определяющее место этого элемента в массиве, которое называется *индексом*. *Размерность* массива – количество индексов, необходимое для однозначного доступа к элементу массива. Если для определения места элемента в массиве используется один индекс, то массив называют *одномерным* (вектором), два – *двумерным* (матрицей). В языке Паскаль индекс заключается в квадратные скобки. Индекс может быть константой –  $a[5]$ ,  $b[1,1]$ ; переменной –  $a[i]$ ,  $b[i,j]$ ; выражением –  $a[i+3]$ ,  $b[i+1,j+1]$ .

**Одномерные массивы.** Массивы описываются в разделе описания переменных в следующей форме:

```
var имя_массива: array[тип_индекса] of тип_элементов;
```

В качестве *типа индекса* чаще всего используется тип-диапазон –  $[n..k]$ .

Количество элементов в диапазоне определяется следующим образом:  $k-n+1$ .

*Пример.*

```
var a: array[1..10] of integer;
```

Здесь  $a$  – имя массива, элементы которого имеют базовый тип `integer`, первый элемент имеет индекс 1, индекс последнего элемента 10, всего 10 элементов.

На языке Паскаль есть возможность создавать свои типы данных, которые должны быть описаны в специальном разделе описания типов `type`.

*Пример.*

```
const n=10;
type vector=array[1..n] of real; { тип vector объединяет в себе все
var a, b: vector;                 одномерные массивы, состоящие
                                  из n действительных элементов }
```

Ввод и вывод массивов в Паскале осуществляется поэлементно, для чего необходимо организовать цикл.

*Примеры* ввода элементов одномерного массива (с клавиатуры и с помощью датчика случайных чисел):

```
for i:=1 to n do
  read(a[i]);
  randomize;
  for i:= 1 to n do
    begin
      a[i]:= 1 + random (100);
      write(a[i]:5, ' ');
    end;
```

*Пример* вывода элементов одномерного массива.

```
for i:=1 to n do
  write(a[i], ' ');
```

**Перебор элементов одномерного массива.** Элементы массива можно обрабатывать, двигаясь от начала массива к его концу или в обратном направлении:

```
for i:=1 to n do           for i:=n downto 1 do
  { обработка a[i] }       { обработка a[i] }
```

Можно обрабатывать элементы массива сразу по два элемента, двигаясь одновременно с обеих сторон:

```
i:=1;                     { задание нижней границы индекса }
j:=n;                     { задание верхней границы индекса }
```

```

while i<j do
begin
    {обработка a[i] и a[j]};
    i:=i+1; {движение слева направо, индекс увеличивается}
    j:=j-1; {движение справа налево, индекс уменьшается}
end;

```

Если необходимо перебирать только элементы массива с четным номером, то это может быть реализовано следующим образом:

*Вариант 1.*

```

i:=2;           {индекс начинает изменяться с четного
                числа 2}

while i<=n do
begin
    {обработка a[i]}; {величина шага, равная двум, обеспечивает
                       сохранение}
    i:=i+2;       {четности индекса}
end;

```

*Вариант 2.*

```

for i:=1 to n do {внутри цикла перебора индекса вложен}
if i mod 2 =0 then {обработка a[i]}; {оператор, проверяющий
                       его четность}

```

*Вариант 3.*

```

for i:=1 to n div 2 do {используется формула четного числа. По-
                       сколько}
{обработка a[2*i]}; {элементов с четным индексом – половина
                     от всего} {количества, то параметр цикла i
                     изменяется до n div 2}

```

**Двумерные массивы (матрицы).** Для работы с элементами двумерного массива нужно организовать два цикла. Каждый из них отвечает за перебор значений соответствующего индекса, т.е. полу-

чается структура вложенных циклов. Параметры внешнего и внутреннего циклов разные и изменяются не одновременно: при одном значении параметра внешнего цикла параметр внутреннего цикла принимает поочередно все значения.

Двумерный массив  $A(m,n)$  с количеством строк, равным  $m$ , и количеством столбцов, равным  $n$ , содержащий элементы целого типа (integer), описывается в разделе описания переменных в следующей форме:

```
var a: array[1.. m,1..n] of integer;
```

К элементу двумерного массива обращаются следующим образом:  $a[i,j]$ , где  $i$  – номер строки,  $j$  – номер столбца. Обращаться к элементам двумерного массива можно как по строкам, так и по столбцам:

Обработка по строкам	Обработка по столбцам
<pre>for i:=1 to m do {перебор строк}   for j:=1 to n do {перебор столбцов}     {обработка a[i,j]}</pre>	<pre>for j:=1 to n do {перебор столбцов }   for i:=1 to m do {перебор строк }     {обработка a[i,j]}</pre>

*Примеры* ввода элементов двумерного массива (с клавиатуры и с помощью датчика случайных чисел).

```

randomize;
for i:=1 to m do
  for j:=1 to n do
    read(a[i,j]);
randomize;
for i := 1 to m do
  begin
    for j := 1 to n do
      begin
        a[i,j]:= 1 + random (100);
        write(a[i,j]:5, ' ');
      end;
    writeln;
  end;
```

*Пример* вывода элементов двумерного массива в виде матрицы.

```
for i:=1 to m do
  begin
    for j:=1 to n do
      write(a[i,j]:5, ' ');
    writeln;
  end;
```

Для двумерного массива можно использовать те же схемы перебора, что и для одномерного, но комбинаций здесь будет в два раза больше.

# 1. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ ЗАДАЧ ПО ОБРАБОТКЕ ОДНОМЕРНЫХ МАССИВОВ

Ввод и вывод элементов одномерного массива представляют собой циклический процесс. Параметром цикла является текущее значение индекса  $i$ , изменяющееся в общем случае от начального до конечного значения с шагом 1. Блок-схемы алгоритмов ввода и вывода элементов одномерного массива приведены соответственно на рис. 1, 2. Далее в блок-схемах рассматриваемых алгоритмов решения задач по обработке одномерных массивов ввод и вывод элементов показаны только блоком ввода/вывода, подразумевая при этом наличие циклической структуры.

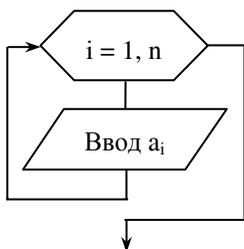


Рис. 1. Блок-схема алгоритма ввода элементов одномерного массива

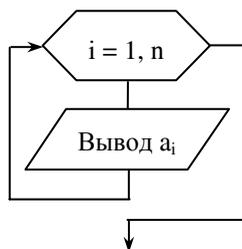


Рис. 2. Блок-схема алгоритма вывода элементов одномерного массива

## 1.1. Вычисление суммы элементов одномерного массива

Задан одномерный массив  $A(n)$ , состоящий из  $n$  элементов.

Вычислить сумму всех его элементов:  $S = \sum_{i=1}^n a_i$ . Вычисление сум-

мы выполняется по формуле:  $S = S + a_i$ .

Блок-схема алгоритма приведена на рис. 3.

## **1.2. Формирование нового массива из элементов заданного массива, удовлетворяющих заданному условию, и подсчет их количества**

Требуется из заданного массива  $A(n)$ , состоящего из  $n$  элементов, выбрать элементы, удовлетворяющие заданному условию  $a_i \leq t$ , и сформировать из них массив  $B$ . Учитывая, что не все элементы массива  $A$  войдут в массив  $B$ , необходимо выполнять подсчет количества  $j$  элементов, удовлетворяющих заданному условию. Блок-схема алгоритма приведена на рис. 4.

Особенностью решения этой задачи является то, что индексы элементов массивов  $A$  и  $B$  не совпадают, поэтому для обозначения индекса элементов массива  $B$  используют переменную  $j$ , значение которой изменяется на 1 перед занесением в массив  $B$  нового значения.

## **1.3. Объединение двух массивов в один с чередованием исходных элементов**

Требуется объединить два заданных массива  $A(n)$  и  $B(n)$ , содержащих по  $n$  элементов, в один массив  $C(2 \cdot n)$ , который будет содержать  $2n$  элементов, т.е. получить массив  $C = \{a_1, b_1, a_2, b_2, \dots, a_n, b_n\}$ .

Индекс элемента массива  $C$  зависит от индекса пересылаемого в него элемента массива  $A$  или  $B$  следующим образом:  $c_{2 \cdot i - 1} = a_i$ ,  $c_{2 \cdot i} = b_i$ ,  $i = 1, 2, 3, \dots, n$ .

Блок-схема алгоритма приведена на рис. 5.

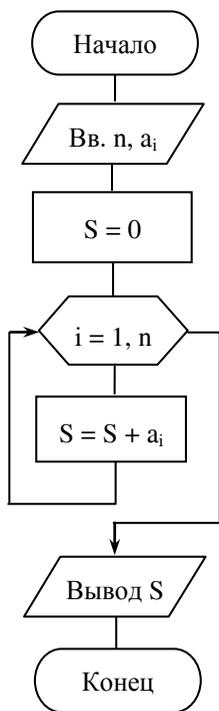


Рис. 3. Блок-схема алгоритма вычисления суммы элементов одномерного массива

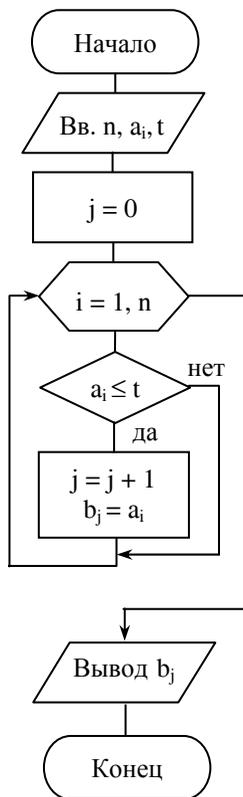


Рис. 4. Блок-схема алгоритма формирования нового массива из элементов заданного массива, удовлетворяющих заданному условию

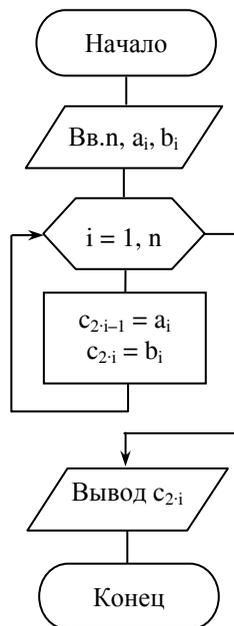


Рис. 5. Блок-схема алгоритма объединения двух массивов в один с чередованием исходных элементов

#### 1.4. Поиск максимального (минимального) элемента в массиве с запоминанием его индекса

Требуется из заданного массива  $A(n)$ , состоящего из  $n$  элементов, найти максимальный (минимальный) элемент и его индекс.

Блок-схема алгоритма поиска максимального элемента приведена на рис. 6, где  $p$  – максимальное значение из элементов массива,  $k$  – индекс максимального элемента.

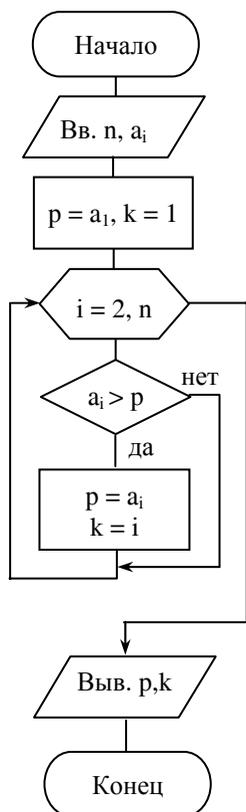


Рис. 6. Блок-схема алгоритма поиска максимального элемента в массиве

При поиске максимального элемента в массиве (рис. 6) вначале считаем кандидатом на максимальный первый элемент массива, затем в цикле сравниваем очередной элемент массива с кандидатом на максимальный, если очередной элемент больше кандидата, то меняем значение кандидата на значение данного элемента массива и фиксируем его индекс.

При поиске минимального элемента в массиве необходимо условие  $a_i > p$  на блок-схеме (см. рис. 6) заменить условием  $a_i < p$ .

Если в массиве несколько элементов имеют одинаковое максимальное (минимальное) значение, то в  $k$  запоминается индекс первого из них. Для того чтобы в  $k$  запоминался индекс последнего максимального элемента, нужно условие  $a_i > p$  заменить условием  $a_i \geq p$ , а для нахождения минимального – условием  $a_i \leq p$ .



*Пример* программы на языке Паскаль решения задачи поиска максимального элемента в массиве с запоминанием его индекса (рис. 7).

```
const n=10;
var a: array [1..n] of integer;
    p, k, i: integer;
begin
    for i:=1 to n do
        read(a[i]);
    p:=a[1]; k:=1;
    for i:=2 to n do
        if a[i]>p then
            begin p:=a[i]; k:=i; end;
    write ('max=', p, ' k=', k);
end.
```

## 2. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ ЗАДАЧ ПО ОБРАБОТКЕ ДВУМЕРНЫХ МАССИВОВ

Ввод и вывод элементов двумерного массива (матрицы) представляют собой циклический процесс, при этом нужно организовать два цикла, один из которых отвечает за перебор значений индекса строк, а второй – за перебор значений индекса столбцов.

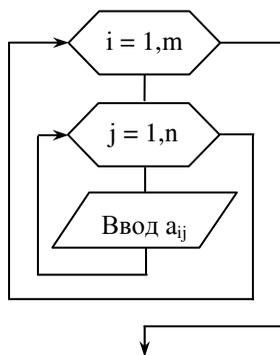


Рис. 8. Блок-схема алгоритма ввода элементов двумерного массива

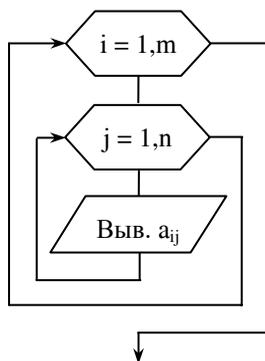


Рис. 9. Блок-схема алгоритма вывода элементов двумерного массива

Блок-схемы алгоритмов ввода и вывода элементов двумерного массива приведены соответственно на рис. 8, 9. Далее в блок-схемах рассматриваемых алгоритмов решения задач по обработке двумерных массивов ввод и вывод элементов показаны только блоком ввода/вывода, подразумевая при этом наличие циклических структур.

## 2.1. Вычисление суммы всех элементов двумерного массива

Задана матрица  $A(m,n)$ , состоящая из  $m$  строк и  $n$  столбцов.

Найти сумму всех ее элементов:  $S = \sum_{i=1}^m \sum_{j=1}^n a_{ij}$ . Вычисление суммы

выполняется по формуле:  $S = S + a_{ij}$ .

Блок-схема алгоритма приведена на рис. 10.

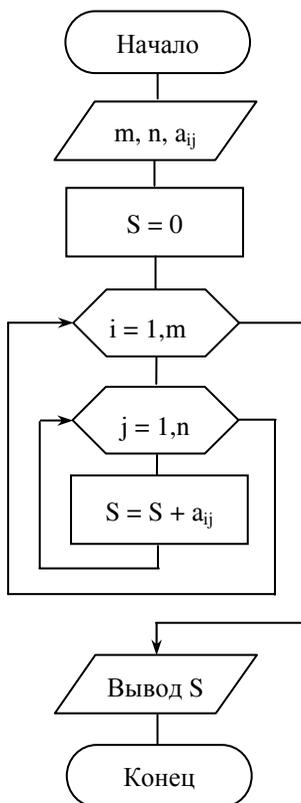


Рис. 10. Блок-схема алгоритма вычисления суммы элементов двумерного массива

## 2.2. Вычисление следа матрицы

Следом квадратной матрицы  $A(m,m)$  является сумма элементов матрицы, расположенных на ее главной диагонали, которая вычисляется по формуле:

$$S = \sum_{i=1}^n a_{ii}.$$

Для вычисления суммы по приведенной формуле достаточно организовать один цикл, так как индексы у элементов главной диагонали матрицы равны. Блок-схема алгоритма приведена на рис. 11.

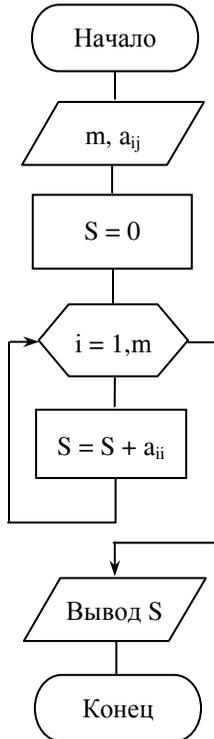


Рис. 11. Блок-схема алгоритма вычисления следа матрицы

### 2.3. Вычисление сумм элементов для каждой строки матрицы

Суммы элементов строк матрицы  $A(m,n)$  вычисляются по формуле:

$$d_i = \sum_{j=1}^n a_{ij}, \quad i = 1, 2, \dots, m.$$

Результатом является одномерный массив  $D(m)$ , содержащий  $m$  элементов.

Блок-схема алгоритма приведена на рис. 12.

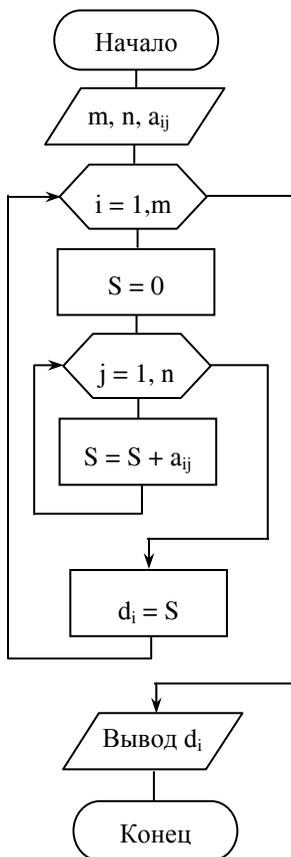


Рис. 12. Блок-схема алгоритма вычисления сумм элементов для каждой строки матрицы

*Пример* программы на языке Паскаль вычисления следа матрицы (см. рис. 11).

```
const m=10;
var a: array [1..m, 1..m] of integer;
    S: integer;
begin
  randomize;
  for i := 1 to m do
    begin
      for j := 1 to m do
        begin
          a[i,j]:= 1 + random (100); {ввод элементов матрицы с помощью}
          write(a[i,j]:5,' ')      {датчика случайных чисел}
        end;
      writeln;
    end;
  S:=0;
  for i:=1 to m do  достаточно одного цикла, так как у элементов главной}
    S:=S + a[i,i]; {диагонали матрицы значения индексов равны}
  write('S=', S);
end.
```

*Пример* программы на языке Паскаль вычисления сумм элементов для каждой строки матрицы (см. рис. 12).

```
const m=10; n=10;
var a: array [1..m, 1..n] of integer;
    S: integer;
begin
```

```

for i:=1 to m do
for j:=1 to n do
    read(a[i,j]); {ввод элементов матрицы с клавиатуры}
for i:=1 to m do
    begin
        S:=0;
        for j:=1 to n do
            S:=S + a[i,j]; {вычисление суммы элементов каж-
                дой строки}
        d[i]:=S;          {матрицы}
    end;
for i:=1 to m
    writeln(' d[', i, ']= ', d[i]);
end.

```

## 2.4. Транспонирование матрицы

Транспонированной матрицей  $V(n,m)$  от матрицы  $A(m,n)$  называется матрица, у которой столбцы заменены строками, а строки – столбцами. Элементы транспонированной матрицы вычисляются по формуле:

$$b_{ji} = a_{ij}, i = 1, 2, \dots, m; j = 1, 2, \dots, n.$$

Блок-схема алгоритма приведена на рис. 13.

## 2.5. Преобразование матрицы в одномерный массив

Необходимо из элементов матрицы  $A(m,n)$  получить одномерный массив  $B(l)$ , где  $l = m \cdot n$ , элементы которого получаются из строк исходной матрицы:

$$b_1 = a_{11}, b_2 = a_{12}, \dots, b_n = a_{1n}, b_{n+1} = a_{21}, \dots, b_{m \cdot n} = a_{mn}$$

по формуле  $b_k = a_{ij}$ .

Блок-схема алгоритма приведена на рис. 14.

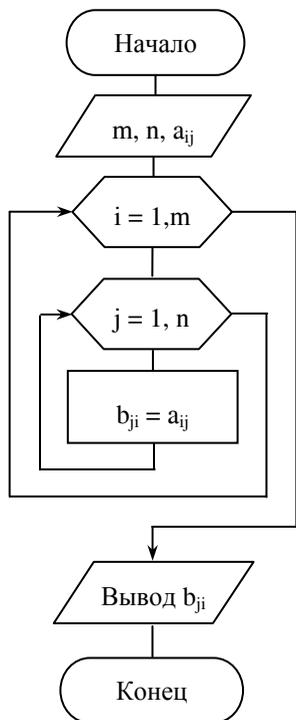


Рис. 13. Блок-схема алгоритма транспонирования матрицы

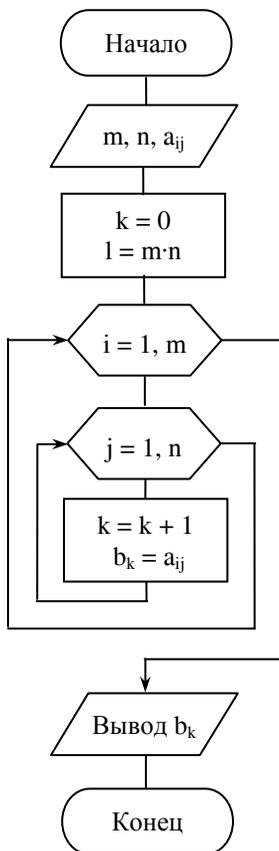


Рис. 14. Блок-схема алгоритма преобразования матрицы в одномерный массив

## 2.6. Умножение матрицы на вектор

Произведением матрицы  $A(m,n)$  на вектор-столбец  $V(n)$  является вектор  $C(m)$ , каждый элемент которого вычисляется по формуле:

$$c_i = \sum_{j=1}^n a_{ij} \cdot b_j, i = 1, 2, \dots, m.$$

Блок-схема алгоритма приведена на рис. 15.

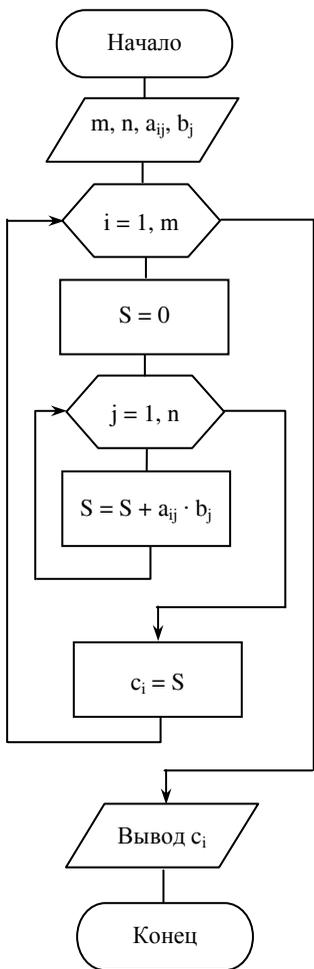


Рис. 15. Блок-схема алгоритма умножения матрицы на вектор

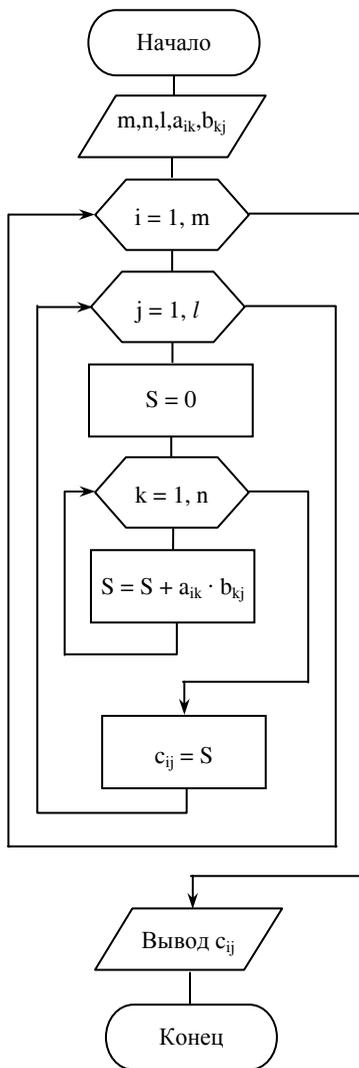


Рис. 16. Блок-схема алгоритма умножения матрицы на матрицу

## 2.7. Умножение матрицы на матрицу

Произведением матрицы  $A(m,n)$  на матрицу  $B(n,l)$  является матрица  $C(m,l)$ , каждый элемент которой вычисляется по формуле:

$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}, i = 1, 2, \dots, m; j = 1, 2, \dots, l.$$

Умножить можно только те матрицы, у которых число столбцов первой матрицы равно числу строк второй матрицы. Результатом является матрица, число строк которой равняется числу строк первой матрицы, а число столбцов равно числу столбцов второй матрицы.

Блок-схема алгоритма приведена на рис. 16.

## 2.8. Удаление $k$ -й строки матрицы

Для удаления  $k$ -й строки матрицы  $A(m,n)$  необходимо все строки, начиная с  $(k+1)$ -й переместить вверх по формуле:  
 $a_{ij} = a_{i+1,j}, i = k, \dots, m; j = 1, 2, \dots, n.$

Результатом является матрица  $A(m-1,n)$ , количество строк которой на единицу меньше, чем в исходной. Блок-схема алгоритма приведена на рис. 17.

## 2.9. Включение строки в матрицу

Необходимо в матрицу  $A(m,n)$  на место  $k$ -й строки вставить новую строку, представленную вектором  $C(n)$ .

Задача решается в два этапа:

- 1) строки, начиная с  $k$ -й, смещаются вниз на одну строку;
- 2) на место  $k$ -й строки вставляется вектор  $C(n)$ .

Смещение строк на одну вниз выполняется по формуле:

$$a_{i+1,j} = a_{i,j}, \text{ начиная с последней строки: } i = m, m-1, \dots, k.$$

Включение элементов вектора  $C(n)$  на место  $k$ -й строки матрицы выполняется по формуле:  $a_{kj} = c_j, j = 1, 2, \dots, n$ .

Результатом является матрица  $A(m+1, n)$ , количество строк в которой на одну больше, чем в исходной матрице. Блок-схема алгоритма приведена на рис. 18.

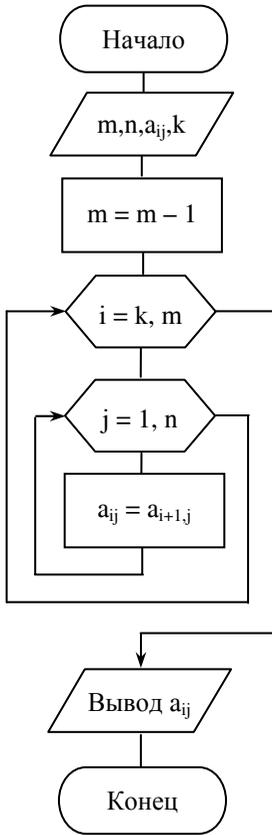


Рис. 17. Блок-схема алгоритма удаления  $k$ -й строки матрицы

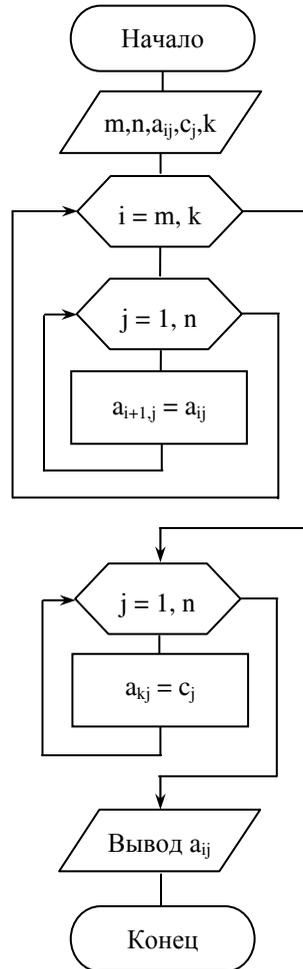


Рис. 18. Блок-схема алгоритма включения строки в матрицу

## 2.10. Перестановка строк (столбцов) матрицы местами

В исходной матрице  $A(m,n)$  необходимо поменять местами строку  $k$  со строкой  $l$ . Элементы меняются местами с помощью дополнительной переменной  $p$ .

Блок-схема алгоритма приведена на рис. 19.

Блок-схема алгоритма перестановки местами  $k$ -го столбца со столбцом  $l$  приведена на рис. 20.

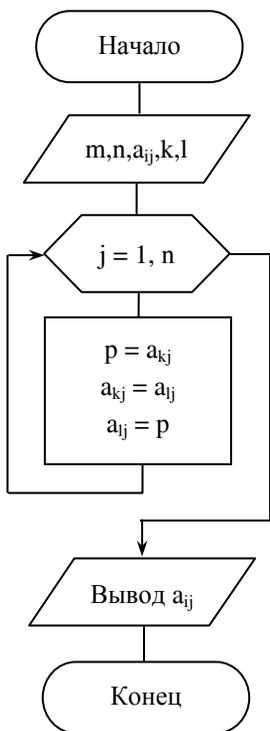


Рис. 19. Блок-схема алгоритма перестановки  $k$ -й и  $l$ -й строк матрицы местами

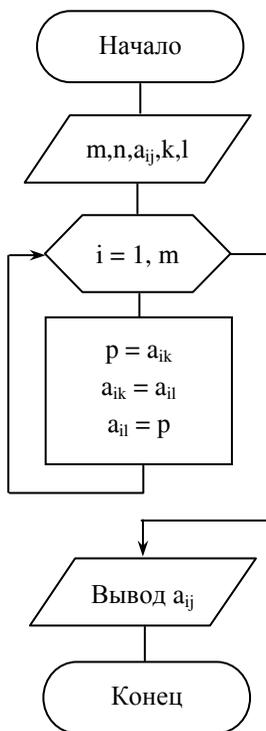


Рис. 20. Блок-схема алгоритма перестановки  $k$ -го и  $l$ -го столбцов матрицы местами

## СПИСОК ЛИТЕРАТУРЫ

1. Аляев Ю.А., Козлов О.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic: учеб.-справ. пособие. – М.: Финансы и статистика, 2007. – 319 с.
2. Аляев Ю.А., Гладков В.П., Козлов О.А. Практикум по алгоритмизации и программированию на языке Паскаль: учеб. пособие. – М.: Финансы и статистика, 2007. – 527 с.
3. Семакин И.Г., Шестаков А.П. Основы программирования: учеб. – 5-е изд. – М.: Академия, 2006. – 431 с.

## ЛАБОРАТОРНЫЕ РАБОТЫ

Порядок выполнения лабораторных работ:

1. Проанализировать задание, установить какие величины являются входными параметрами, какие – выходными.
2. Составить блок-схему алгоритма.
3. По разработанному алгоритму составить программу на языке Паскаль.
4. В редакторе среды программирования Pascal ABC ввести программу и отредактировать ее.
5. Осуществить компиляцию и отладку программы.
6. Получить результаты работы программы.
7. Оформить отчет по работе и ответить на контрольные вопросы.

### Лабораторная работа № 1

#### Программирование типовых алгоритмов обработки одномерных массивов

**Цель работы:** научиться использовать языки и системы программирования для решения задач – освоить типовые алгоритмы и приемы программирования при работе с одномерными массивами.

Указания к выполнению задания: исходные данные, т.е. элементы заданного массива вывести вместе с полученными результатами.

1. Найти сумму элементов одномерного массива, состоящего из 10 элементов. Разделить каждый элемент исходного массива на полученное значение. Результат получить в том же массиве. Вывести в одной строке.

2. Вычислить сумму и разность двух заданных одномерных массивов, состоящих из 5 элементов. Результат напечатать в виде двух параллельных столбцов.

3. Задан массив, состоящий из 5 элементов. Вычислить значения функции  $y = x^2$  при значениях аргумента, заданных в массиве  $x$ , и поместить их в массив  $y$ . Напечатать массивы  $x$  и  $y$  в виде двух столбцов.

4. Найти среднее арифметическое значение элементов массива, состоящего из 5 элементов. Преобразовать исходный массив, вычитая из каждого элемента среднее значение. Исходный и полученный массивы напечатать в виде двух параллельных столбцов.

5. Определить среднее арифметическое значение элементов массива, состоящего из 6 элементов. Найти далее индекс элемента массива, наиболее близкого к среднему значению, т.е. найти минимальную по абсолютной величине разность элементов массива и среднего арифметического значения ( $|a_i - D|$ , где  $D$  – среднее арифметическое значение).

6. Задан массив, состоящий из 10 элементов. Сформировать два массива по 5 элементов каждый, включая в первый элементы исходного массива с четными индексами, а во второй – с нечетными.

7. Вычислить минимальный элемент массива, состоящего из 8 элементов, и его номер. Преобразовать исходный массив, вычитая из каждого элемента минимальное значение.

8. Найти максимальный и минимальный элементы массива, состоящего из 8 элементов, и поменять их местами. Полученный массив напечатать в виде столбца.

9. Объединить два заданных массива  $A$  и  $B$ , содержащих по 5 элементов, в один массив  $C = \{a_1, b_1, a_2, b_2, \dots, a_5, b_5\}$ . Полученный массив вывести в одной строке.

10. Переписать подряд в массив  $Y$  положительные и в массив  $Z$  отрицательные элементы массива  $X$ , состоящего из 6 элементов,  $X = \{-1, 1, 2, -2, -3, 3\}$ . Результат (массивы  $Y$  и  $Z$ ) напечатать в виде двух параллельных столбцов.

## Лабораторная работа № 2

### Программирование типовых алгоритмов обработки двумерных массивов

**Цель работы:** научиться использовать языки и системы программирования для решения задач – освоить типовые алгоритмы и приемы программирования при работе с двумерными массивами (матрицами).

Указания к выполнению задания: значения элементов матриц задать с помощью датчика случайных чисел. Исходные данные, т.е. элементы заданного массива, вывести вместе с полученными результатами.

1. Задана матрица  $A(4, 4)$ . Вычислить наибольшие элементы каждой строки матрицы. Результат получить в одномерном массиве  $B(4)$ . Определить номер строки, в которой значение наибольшего элемента максимально.

2. Задана матрица  $A(5, 4)$ . Вычислить суммы элементов каждой строки матрицы. Результат получить в одномерном массиве  $B(5)$ . Определить номер строки с наибольшим значением вычисленной суммы.

3. Задана матрица  $A(4, 3)$ . Вычислить наименьший элемент матрицы и номера строки и столбца, в которых он расположен. Удалить строку с наименьшим элементом из матрицы. Полученную матрицу напечатать по строкам.

4. Задана матрица  $A(5, 4)$ . Вычислить наибольший элемент матрицы и номера строки и столбца, в которых он расположен. Записать нули в строку и столбец, в которых он находится. Полученную матрицу напечатать по строкам.

5. Задана матрица  $A(4, 3)$ . Вычислить число положительных, отрицательных и нулевых элементов матрицы (предполагается наличие каждого из них). Заменить все отрицательные элементы матрицы на положительные с тем же абсолютным значением. Полученную матрицу напечатать по строкам.

6. Задана матрица  $A(3, 5)$ . Упорядочить по возрастанию элементы той строки, где находится наибольший элемент матрицы. Полученную матрицу напечатать по строкам.

7. Перемножить матрицы  $A(4, 3)$  и  $B(3, 5)$ . Элементы результирующей матрицы вычислить по формуле:  $C_{ik} = \sum_{j=1}^m a_{ij} \cdot b_{jk}$ ;  $i = 1, 2, \dots, n$ ;  $k = 1, 2, \dots, l$ . Полученную матрицу напечатать по строкам.

8. Упорядочить элементы каждой строки заданной матрицы  $A(5, 4)$ , расположив их в матрице  $B(5, 4)$  по убыванию. Полученную матрицу напечатать по строкам.

9. Матрицу  $A(5, 4)$  преобразовать в матрицу  $B(4, 3)$ , вычеркнув строку и столбец, на пересечении которых расположен наименьший элемент матрицы  $A$ . Полученную матрицу  $B$  напечатать по строкам.

10. Задана матрица  $A(5, 5)$ . Вычислить сумму элементов верхней треугольной матрицы, включая элементы ее главной диагонали. Преобразовать исходную матрицу, присвоив всем элементам главной диагонали значения, равные 1. Полученную матрицу напечатать по строкам.

### Лабораторная работа № 3

#### Программирование преобразования и построения матриц

**Цель работы:** научиться использовать языки и системы программирования для решения задач – освоить приемы программирования преобразования и построения матриц.

Указания к выполнению задания: значения элементов матриц задать с помощью датчика случайных чисел. Исходные данные, т.е. элементы заданного массива, вывести вместе с полученными результатами.

1. Дана действительная квадратная матрица  $A(6, 6)$ . Преобразовать матрицу по правилу: строку с номером  $n=3$  сделать столбцом

с номером  $n=3$ , а столбец с номером  $n=3$  сделать строкой с номером  $n=3$ .

2. Даны действительные числа  $a_1, \dots, a_6$ , действительная квадратная матрица  $D(6, 6)$ . Получить действительную матрицу  $B(6, 7)$ , вставив в исходную матрицу между пятым и шестым столбцами новый столбец с элементами  $a_1, \dots, a_6$ .

3. Дана целочисленная матрица  $A(6, 9)$ . Найти матрицу, получающуюся из данной перестановкой столбцов – первого с последним, второго с предпоследним и т.д.

4. Дана целочисленная матрица  $A(7, 8)$ . Найти матрицу, получающуюся из данной перестановкой строк – первой с последней, второй с предпоследней и т.д.

5. Дана действительная матрица  $D(6, 7)$ , действительные числа  $a_1, \dots, a_7, b_1, \dots, b_7$ , натуральные числа  $p=3, q=5$ . Образовать новую матрицу  $C(7, 8)$  вставкой после строки с номером  $p$  данной матрицы новой строки с элементами  $a_1, \dots, a_7$  и последующей вставкой после столбца с номером  $q$  нового столбца с элементами  $b_1, \dots, b_7$ .

6. В данной действительной матрице  $A(5, 7)$  поменять местами: строки с номерами 2 и 4; столбцы с номерами 3 и 5.

7. В данной действительной квадратной матрице  $A(6, 6)$  найти наибольший по модулю элемент. Получить матрицу  $B(5, 6)$  путем удаления из исходной матрицы строки с найденным наибольшим значением.

8. Дана действительная матрица  $A(6, 7)$ . Получить транспонированную матрицу  $B(7, 6)$ .

9. Дана действительная матрица  $A(7, 8)$ . Необходимо из ее элементов получить одномерный массив  $B(7 \cdot 8)$ , элементы которого получаются переписыванием матрицы  $A(7, 8)$  по строкам.

10. Даны действительная матрица  $A(6, 7)$  и вектор-столбец  $B(7)$ . Получить в результате умножения исходной матрицы  $A(6, 7)$  на вектор-столбец  $B(7)$  вектор  $C(6)$ , каждый элемент которого вы-

числяется по формуле: 
$$C_i = \sum_{j=1}^7 a_{ij} \cdot b_j, \quad i = 1, 2, \dots, 6.$$

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что называется массивом?
2. Что называется размерностью массива?
3. Какого типа могут быть элементы массива?
4. Какого типа могут быть индексы элементов массива?
5. Как объявить в программе одномерный и двумерный массив?
6. Какими способами может быть заполнен массив? Приведите примеры.
7. Как осуществить вывод элементов одномерного массива в строку?
8. Как осуществить вывод элементов одномерного массива в столбец?
9. Каким образом можно организовать ввод (вывод) элементов двумерных массивов?
10. Как осуществить вывод элементов двумерного массива в виде матрицы?
11. Каким образом в программе организовать обработку массива по строкам (столбцам)?
12. Алгоритм вычисления суммы элементов одномерного массива.
13. Алгоритм формирования нового массива из элементов заданного массива, удовлетворяющих заданному условию, и подсчет их количества.
14. Алгоритм объединения двух массивов в один с чередованием исходных элементов.
15. Алгоритм поиска максимального (минимального) элемента в массиве с запоминанием его индекса.
16. Алгоритм упорядочивания массива.
17. Алгоритм вычисления суммы всех элементов двумерного массива.
18. Алгоритм вычисления следа матрицы.

19. Алгоритм вычисления сумм элементов для каждой строки матрицы.
20. Алгоритм транспонирования матрицы.
21. Алгоритм преобразования матрицы в одномерный массив.
22. Алгоритм умножения матрицы на вектор.
23. Алгоритм умножения матрицы на матрицу.
24. Алгоритм удаления  $k$ -й строки матрицы.
25. Алгоритм включения строки в матрицу.
26. Алгоритм перестановки строк (столбцов) матрицы местами.

**КОНТРОЛЬНЫЕ ЗАДАНИЯ**  
**по теме «Программные средства реализации алгоритмов.**  
**Алгоритмизация и программирование задач**  
**по обработке массивов»**

1. Дан одномерный массив  $A(n)$ . Найти  $\max(a_2, a_4, \dots, a_{2k}) + \min(a_1, a_3, \dots, a_{2k+1})$ .

2. Дана последовательность действительных чисел  $a_1, a_2, \dots, a_n$ . Указать те ее элементы, которые принадлежат отрезку  $[c, d]$ .

3. Дана последовательность целых положительных чисел  $a_1, a_2, \dots, a_n$ . Найти произведение только тех из них, которые больше заданного числа  $m$ . Если таких чисел нет, то выдать сообщение об этом.

4. Даны действительные числа  $a_1, a_2, \dots, a_n$ . Среди них есть положительные и отрицательные. Заменить нулями те числа, величина которых по модулю больше максимального числа, т.е.  $|a_i| > \max\{a_1, a_2, \dots, a_n\}$ .

5. Даны действительные числа  $a_1, a_2, \dots, a_{2n}$ . Найти  $\max(a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1})$ .

6. В последовательности действительных чисел  $a_1, a_2, \dots, a_n$  есть только положительные и отрицательные элементы. Вычислить произведение отрицательных элементов  $P_1$  и произведение положительных элементов  $P_2$ . Сравнить модуль  $P_2$  с модулем  $P_1$ , указать, какое из произведений по модулю больше.

7. Даны целые положительные числа  $a_1, a_2, \dots, a_n$ . Найти среди них те, которые являются квадратами некоторого числа  $m$ . Если таких чисел нет, то выдать сообщение об этом.

8. Даны действительные числа  $a_1, a_2, \dots, a_{2n}$ . Найти  $\min(a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1})$ .

9. Дана последовательность целых чисел  $a_1, a_2, \dots, a_n$ . Найти произведение только тех из них, величина которых по модулю меньше заданного числа  $m$ . Если таких чисел нет, то выдать сообщение об этом.

10. Дана последовательность действительных чисел  $a_1, a_2, \dots, a_n$ . Найти сумму элементов, которые принадлежат отрезку  $[c, d]$ .

Учебное издание

ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ АЛГОРИТМОВ.  
АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ ЗАДАЧ  
ПО ОБРАБОТКЕ МАССИВОВ

Методические указания  
к выполнению лабораторных работ  
по дисциплине «Информатика»

Составитель *И.Н. Шапова*

Корректор *И.Н. Жеганина*

---

Подписано в печать 12.08.15. Формат 60×90/16.  
Усл. печ. л. 2,25. Тираж 100 экз. Заказ № 142/2015.

---

Издательство  
Пермского национального исследовательского  
политехнического университета.  
Адрес: 614990, г. Пермь, Комсомольский пр., 29, к. 113.  
Тел. (342) 219-80-33.