

Міністерство освіти і науки України
Харківський національний автомобільно-дорожній університет
Кафедра інформаційних технологій та мехатроніки

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт з дисципліни
“Інформаційні технології”

**“Програмування на мові C++
у середовищі Microsoft Visual Studio 2010”**

для студентів напряму підготовки 6.050702 ” Електромеханіка ”,
галузь знань 0507 ” Електротехніка та електромеханіка ”

Розроблено та надруковано доц. Симбірським Г.Д.

Харків, 2018

Лабораторная работа №6
ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ СРЕДЫ VISUAL C++ 2010
ДЛЯ ОПЕРАЦИЙ С МНОГОМЕРНЫМИ МАССИВАМИ ДАННЫХ

Цель работы: получение навыков разработки программ для многомерных массивов данных в Visual C++.

1. Двухмерные массивы данных и их инициализация в Visual C++

Под размерностью массива понимают число индексов, которое необходимо указать для получения доступа к отдельному элементу массива. Массивы, рассмотренные в лабораторной работе № 5, например, были одномерными и требовали только одного индекса. Массивы с более чем одной размерностью, называются многомерными.

Самым простым многомерным массивом является двухмерный массив (матрица).

При объявлении массива указывается тип элементов массива, имя массива и его размер:

Тип ИмяМассива [Размер 1] [Размер 2];

Например, оператор

int A[3][8];

описывает целый двухмерный массив по имени **A** из 24-х целых чисел. В памяти будет зарезервировано место для 24-х целочисленных элементов массива (рис. 9.1).

В памяти компьютера двухмерный массив располагается непрерывно по строкам, то есть

A[0][0], A[0][1], A[0][2], A[0][3] ... A[0][8], A[1][0], A[1][1], A[1][2], A[1][8] ... A[0][8].

На рис. 1 приведена схема размещения элементов массива из 24-х целых чисел по имени **A** размерностью 3×8. В памяти будет зарезервировано место для 24-х целочисленных элементов массива, которые располагаются в **непрерывном (!!!)** блоке памяти.

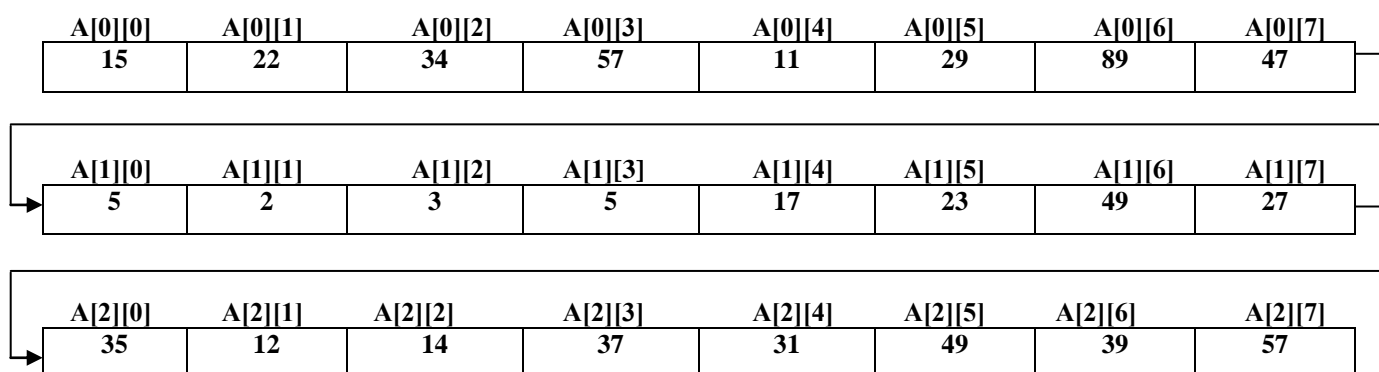


Рис. 1. Значения и расположение в памяти элементов массива **A[3][8]**

Массивы хранятся в памяти компьютера так, что самый правый индекс измеряется быстрее всего.

Возможна инициализации массива непосредственно в программном коде. В этом случае в фигурных скобках приводятся значения элементов массива (рис. 6.1) в следующем виде:

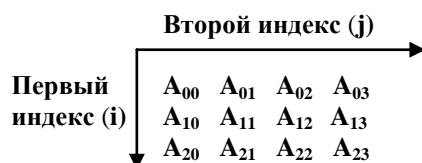
int A[2][3]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

или

int A[2][3]= {{1, 2, 3, 4} {5, 6, 7, 8} {9, 10, 11, 12}};

Количество инициализаторов не обязано совпадать с количеством элементов массива. Если инициализаторов меньше, то оставшиеся элементы не определены.

Двухмерный массив, например, **int A[3][4]** можно представить в виде следующей матрицы:



Первый индекс – это номер строки в массиве, второй индекс – номер столбца.

В памяти компьютера элементы такой матрицы разместятся в таком порядке:

A₀₀, A₀₁, A₀₂, A₀₃, A₁₀, A₁₁, A₁₂, A₁₃, A₂₀, A₂₁, A₂₂, A₂₃.

Кроме этого, инициализация массивов возможна в процессе выполнения программы – путем записи данных в отведенные для массивов ячейки памяти.

При работе с массивами, в т. ч. одномерными, целесообразно использовать оператор цикла **for**, т.к. известен размер обрабатываемого массива (число элементов массива), т. е. число повторений цикла.

В языке C++ не проверяется выход индекса за пределы массива. Если массив **m[100]** целочисленный массив

```
int m[100]; ,
```

а в программе указано

```
x=m[200]; ,
```

то сообщение об ошибке не будет, а переменной **x** будет присвоено произвольное значение.

При обработке массивов в Visual C++ 2010 все действия в программе выполняются над элементами массива (!), а не над массивом в целом. При этом индекс элемента может быть задан либо его значением, либо выражением:

```
A[4], F[i+k+1]; .
```

Над массивами можно выполнять следующие действия:

1. Вводить массивы в память компьютера.
2. Выводить массивы на экран дисплея, на другое устройство или в файл.
3. Присваивать определенные значения элементам массивов.
4. Копировать массивы.
5. Переставлять элементы массивов.
6. Сортировать элементы массивов.

2. Консольный ввод и вывод двумерных массивов в среде Visual C++

Т. к. все действия необходимо выполнять над элементами двумерных массивов, то для ввода двумерного массива в память компьютера необходимо организовать его поэлементный ввод посредством оператора цикла **for** (т. к. известен размер массива).

Для консольного вывода двумерного массива также надо с помощью оператора цикла **for** организовать поэлементный вывод исследуемого массива.

Задание 6.1. Исследовать способы консольного ввода и вывода двумерных массивов. Необходимо ввести с клавиатуры матрицу **A** размерностью **M×N** в память компьютера и вывести эту информацию на дисплей.

Решение. Для придания программе большей универсальности зададим размерность исходной матрицы с запасом, а реальная размерность будет вводиться в каждом конкретном случае. Число строк обозначим **m**, а столбцов - **n**.

Поэлементный ввод матрицы **A** организован при помощи двух операторов цикла **for** – внешнего и внутреннего (вложенного). Внешний цикл организует перебор элементов матрицы **A** по строкам, а вложенный – по столбцам.

Поэлементный вывод исходной матрицы на экран будет производиться аналогичным образом.

Блок-схема алгоритма решения данной задачи приведена на рис. 2.

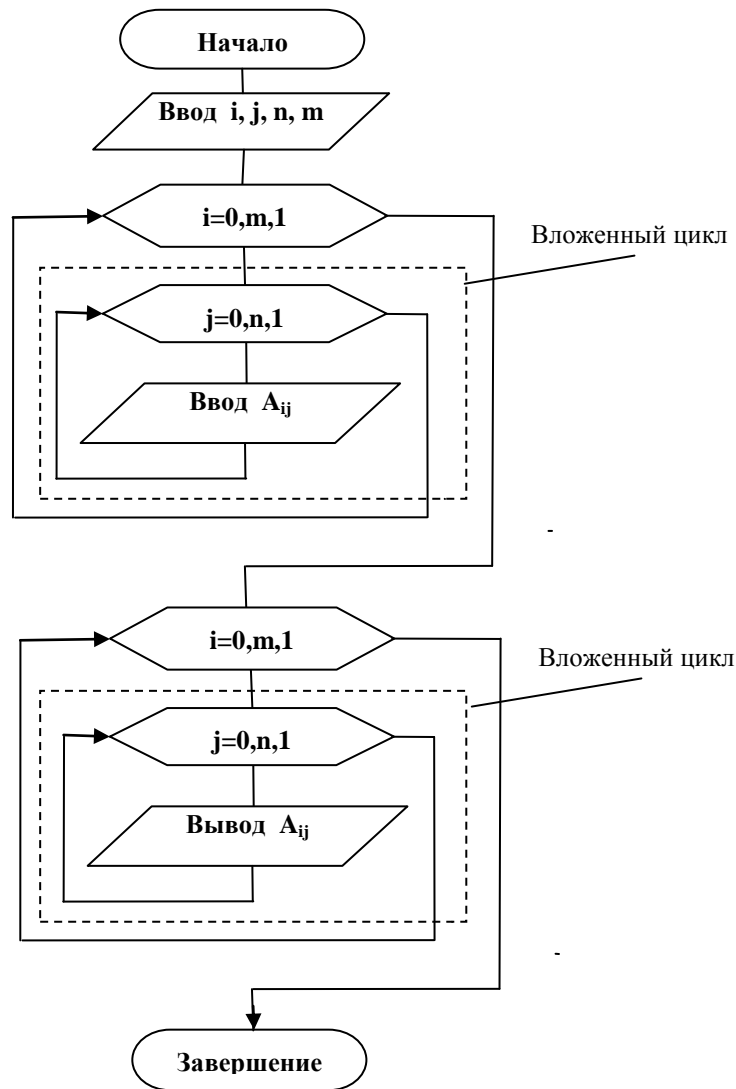


Рис. 2. Блок-схема алгоритма поэлементного ввода и вывода матрицы с использованием цикла **for**

Реализующая данный алгоритм программа имеет следующий вид:

```

#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    const int M=10,N=10; //Число строк и столбцов матрицы A с запасом
    int i, j, m, n, A[M][N]; //Объявление переменных и матрицы A
    cout<<"Vvedite chislo strok i stolbcov matricu:"<<endl;
    cin>>m>>n; //Ввод числа строк и столбцов исходной матрицы A
    cout<<" Vvedite postrochno elementu matricu:"<<endl;
    for(i=0; i<m; i++) //Внешний цикл ввода элементов матрицы A
        for(j=0; j<n; j++) //Вложенный цикл ввода элементов матрицы A
            cin>>A[i][j]; //Ввод элемента матрицы A
    cout<<"Matrica A"<<endl;
    for(i=0; i<m; i++) //Внешний цикл вывода элементов матрицы A
    { //Начало составного оператора для внешнего цикла for
        for(j=0; j<n; j++) //Вложенный цикл для вывода элементов матрицы A
            cout<<A[i][j]<<" "; //Вывод элемента матрицы A
        cout<<endl;
    } //Конец составного оператора для внешнего цикла for
    getch();
    return 0;
}

```

После запуска программы и ввода данных экран дисплея должен иметь вид:

Vvedite chislo strok i stolbcov matricu:

3 4

Vvedite postrochno elementu matricu:

1 2 3 4 5 6 7 8 9 10 11 12

Matrica A

1 2 3 4

5 6 7 8

9 10 11 12

Создайте в текстовом процессоре **Word** файл **Результат_Фамилия_Лр6**. Поля документа сделайте по 0,5 см.

Поместите окно DOS с результатами решения **Задания 6.1** в центральной части окна **Microsoft Visual Studio** ниже программного кода **Lr6-1.cpp** (см. рис. 1.2) и нажмите клавишу **<Prt Scr>**, после чего вставьте полученную копию экрана в файл **Результат_Фамилия_Лр6**. Над вставленным рисунком проставьте номер задания – **6-1**. Файл результатов не закрывайте до получения оценки за выполненную практическую часть работы в тетрадь с отчетом.

Закройте окно DOS, откройте пункт меню **Файл** и выполните команду **Закрывать решение**.

!Внимание! Результаты следующих заданий данной лабораторной работы сохраняйте строго в соответствии с приведенным выше порядком действий!

3. Исследование различных операций в двумерных массивах в среде Visual C++

Элементам массива могут быть присвоены значения выражений. При этом элементы массива и значения выражений должны иметь один и тот же тип.

Например, объявлен массив

```
double A[3][4];
```

тогда возможна запись

```
A[0][0]= 3.5;
```

```
A[1][3]= 0;
```

```
A[2][1]= a*x + b;
```

Копирование – это присваивание значений элементов одного массива элементам другого массива. При копировании оба массива должны иметь одинаковый размер и тип элементов. Копирование массива **A** в **B** будет иметь вид:

```
for (i=0; i<N; i++) B[i][j]= A[i][j];
```

Задание 6.2. Исследовать операцию присваивания двумерных массивов, а также операцию транспонирования матрицы. Задана матрица целых чисел **A** размерностью 3×4. Необходимо транспонировать матрицу **A**, т. е. поменять местами ее строки и столбцы, а затем рассчитать матрицу **B**, элементы которой определяются по формуле

$$B_{ij} = \sqrt[3]{A_{ij}^2} + \sin^2(A^3) .$$

Элементы исходной матрицы **A** ввести непосредственно в программном коде. Исходную матрицу **A**, транспонированную **AT** и полученную матрицу **B** вывести на экран. Для наглядности примем

$$A(3,4) = \begin{vmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{vmatrix} .$$

Блок-схема алгоритма решения данной задачи приведена на рис. 3.

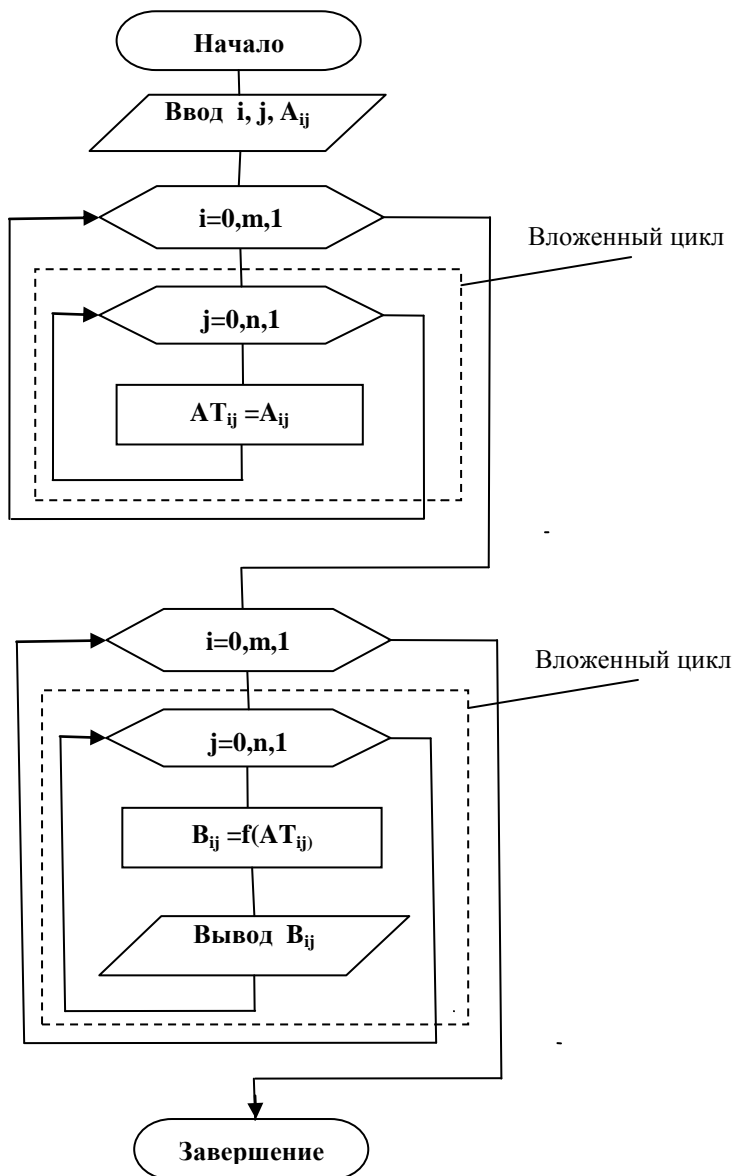


Рис. 3. Блок-схема алгоритма транспонирования и присваивания матрицы

Реализующая данный алгоритм программа имеет следующий вид:

```

#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int i, j;
    double AT[4][3], B[4][3]; //Объявление переменных и матриц A и B
    double A[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}; //Инициализация исходной матрицы A
    cout<<"Matrica A"<<endl; //Вывод названия исходной матрицы

    for(i=0; i<3; i++) //Внешний цикл вывода элементов матрицы A
    {
        for(j=0; j<4; j++) //Вложенный цикл для вывода элементов матрицы A
            cout<<A[i][j]<<" "; //Вывод элементов исходной матрицы A
        cout<<endl;
    }

    //Транспонирование исходной матрицы
    for(i=0; i<3; i++) //Внешний цикл для транспонирования
        for(j=0; j<4; j++) // Вложенный цикл для транспонирования
            AT[j][i]=A[i][j]; //Операция транспонирования матрицы A
    cout<<"Matrica AT = A transponirovannaya"<<endl; //Вывод названия
  
```

```

for(i=0; i<4; i++) //Начало цикла
{
    for(j=0; j<3; j++) //Вложенный цикл для вывода матрицы AT
        cout<<AT[i][j]<<" "; //Вывод элементов матрицы AT
    cout<<endl;
}
cout<<"Matrica B = f(AT)"<<endl; //Вывод названия

//Вычисление матрицы B
for(i=0; i<4; i++) //Внешний цикл для поэлементного вычисления матрицы B
{
    for(j=0; j<3; j++) // Вложенный цикл для поэлементного вычисления матрицы B
    {
        B[i][j]=5*pow(sin(pow(AT[i][j],3)),2)+pow((AT[i][j]*AT[i][j]),1./3.); // Bij = f(ATji)
        cout<<B[i][j]<<" "; //Вывод элементов матрицы B
    }
    cout<<endl;
}
getch();
return 0;
}

```

После запуска программы на выполнение экран дисплея должен иметь вид:

```

Matrica A
1 2 3 4
5 6 7 8
9 10 11 12
Matrica AT = A transponirovannaya
1 5 9
2 6 10
3 7 11
4 8 12
Matrica B = f(AT)
4.54037 4.82155 4.43915
6.48155 5.72441 8.06024
6.65336 5.09899 8.64416
6.75208 4.03162 5.31802

```

Исследуем примеры программ, которые выполняют различные действия над двумерными массивами.

Задание 6.3. Самостоятельно разработать программу для вычисления матрицы Y размера $M \times N$ по формуле для своего варианта (таб. 6.1). Определите свой номер варианта как номер компьютера. По аналогии с Заданием 6.1 задать с запасом $M=10$ и $N=10$. Размер матрицы (3×4) и ее элементы (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) ввести с клавиатуры. Вывести на экран исходную и вычисленную матрицы.

Перед вводом программного кода самостоятельно разработать и начертить в отчете по лабораторной работе блок-схему алгоритма решения данной задачи и записать формулу для расчета на языке C++.

Таблица 6.1. Исходные данные и формулы для расчета Y (Задание 6.3)

№ вар.	$[Y]=f([X])$	№ вар.	$[Y]=f([X])$
1	$y_{ij} = \begin{cases} \cos^2 x_{ij} + 2 \sin x_{ij}, & \text{если } x < 1; \\ 0,25x_{ij}^3 - 3x_{ij}, & \text{если } x \geq 1; \end{cases}$	10	$y_{ij} = \begin{cases} \log_3 5x_{ij}^2 + \sqrt{x_{ij}^3}, & \text{если } x < 0; \\ e^{2x_{ij}} + 0,3x_{ij}^3, & \text{если } x \geq 0; \end{cases}$
2	$y_{ij} = \begin{cases} \log_3 5x_{ij}^2 + \sqrt{x_{ij}^3}, & \text{если } x < 0; \\ e^{2x_{ij}} + 0,23x_{ij}^6, & \text{если } x \geq 0; \end{cases}$	11	$y_{ij} = \begin{cases} 1,56 + \sin^2 x_{ij}, & \text{если } x < 1 \\ 0,19x_{ij}^3 - x_{ij}, & \text{если } x \geq 1 \end{cases}$
3	$y_{ij} = \begin{cases} 1,56 + \sin^2 x_{ij}, & \text{если } x < 1; \\ 0,23x_{ij}^3 - x_{ij}, & \text{если } x \geq 1; \end{cases}$	12	$y_{ij} = \begin{cases} 1,2x_{ij}^2 + 2\sqrt[3]{3x_{ij}}, & \text{если } x < 1 \\ \cos^3 x_{ij}, & \text{если } x \geq 1 \end{cases}$
4	$y_{ij} = \begin{cases} 0,12x_{ij}^2 + 2\sqrt[3]{3x_{ij}}, & \text{если } x < 1; \\ \sin^3 x_{ij}, & \text{если } x \geq 1; \end{cases}$	13	$y_{ij} = \begin{cases} \cos^2 x_{ij} + 2 \sin x_{ij}, & \text{если } x < 1; \\ 0,32x_{ij}^2 - 3x_{ij}, & \text{если } x \geq 1; \end{cases}$

5	$y_{ij} = \begin{cases} \cos^2 x_{ij} + 2 \sin x_{ij}, & \text{если } x < 1; \\ 0,14x_{ij}^3 - 3x_{ij}, & \text{если } x \geq 1; \end{cases}$	14	$y_{ij} = \begin{cases} 2x_{ij}^3 + 4 \sin x_{ij}, & \text{если } x < 0 \\ \sin^2 x_{ij}^3, & \text{если } x \geq 0 \end{cases}$
6	$y_{ij} = \begin{cases} 2x_{ij}^3 + 4 \sin x_{ij}, & \text{если } x < 0; \\ \cos^2 x_{ij}^3, & \text{если } x \geq 0; \end{cases}$	15	$y_{ij} = \begin{cases} 2 \cos x_{ij}^3 + 4 \sin x_{ij}, & \text{если } x < 0; \\ x_{ij}^6, & \text{если } x \geq 0; \end{cases}$
7	$y_{ij} = \begin{cases} 2 \cos x_{ij}^3 + 4 \sin x_{ij}, & \text{если } x < 0; \\ x_{ij}^6, & \text{если } x \geq 0; \end{cases}$	16	$y_{ij} = \begin{cases} 5x_{ij}^2 + \ln x_{ij}, & \text{если } x < 0; \\ 6,8^4 - \sqrt{x_i}, & \text{если } x \geq 0; \end{cases}$
8	$y_{ij} = \begin{cases} 5x_{ij}^2 + \ln x_{ij}, & \text{если } x < 0; \\ \lg x_{ij}^4 + 3x_{ij}^6, & \text{если } x \geq 0; \end{cases}$	17	$y_{ij} = \begin{cases} 0,34 \sin x_{ij}^2 + 2\sqrt[3]{12,1x_{ij}}, & \text{если } x < 1; \\ \cos^3 x_i, & \text{если } x \geq 1; \end{cases}$
9	$y_{ij} = \begin{cases} 0,234x_{ij}^2 + 2\sqrt[3]{12,1x_{ij}}, & \text{если } x < 1; \\ \sin^3 x_{ij}, & \text{если } x \geq 1; \end{cases}$	18	$y_{ij} = \begin{cases} \cos^2 x_{ij} + 2 \sin x_{ij}, & \text{если } x < 1; \\ 0,25x_{ij}^3 - 3x_{ij}, & \text{если } x \geq 1; \end{cases}$

Задание 6.4. Исследовать операцию поиска максимального элемента матрицы. Задана матрица **A** из 20 произвольных чисел размера 4×5. Необходимо найти максимальный элемент матрицы **A** и номера его строки и столбца (индексы элемента матрицы). Элементы матрицы **A** ввести в программе прямым образом. На экран вывести исходную матрицу **A**, максимальный элемент и его индексы.

Блок-схема алгоритма решения данной задачи приведена на рис. 4.

Для решения задачи необходимо ввести и отладить следующий программный код:

```
#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
int i, j, m, n; // Объявление переменных
int A[4][5]={1,2,18,3,20,4,5,6,7,8,9,10,19,11,12,13,14,15,16,17}; //Инициализация матрицы A
int B;
cout<<"Matrica A"<<endl;
for(i=0; i<4; i++) //Внешний цикл вывода матрицы A
{
for(j=0; j<5; j++) //Внутренний цикл вывода матрицы A
cout<<A[i][j]<<"\t"; //Вывод элемента матрицы A
cout<<endl;
}
B=A[0][0]; //Подготовка к поиску максимального элемента
m=0; n=0;
for(i=0; i<4; i++) //Внешний цикл по перебору элементов матрицы A
for(j=0; j<5; j++) //Внутренний цикл по перебору элементов матрицы A
if(A[i][j]>B) //Выбор максимального элемента
{
B=A[i][j]; //Запоминание в B большего элемента Aij
n=i; //Запоминание в n номера строки i элемента Aij
m=j; //Запоминание в m номера столбца j элемента Aij
}
cout<<endl<<"Max= "<<B; //Вывод максимального элемента
cout<<" Stroka " <<n+1; //Вывод номера строки
cout<<" Stolbec " <<m+1<<endl; //Вывод номера столбца
getch();
return 0;
}
```

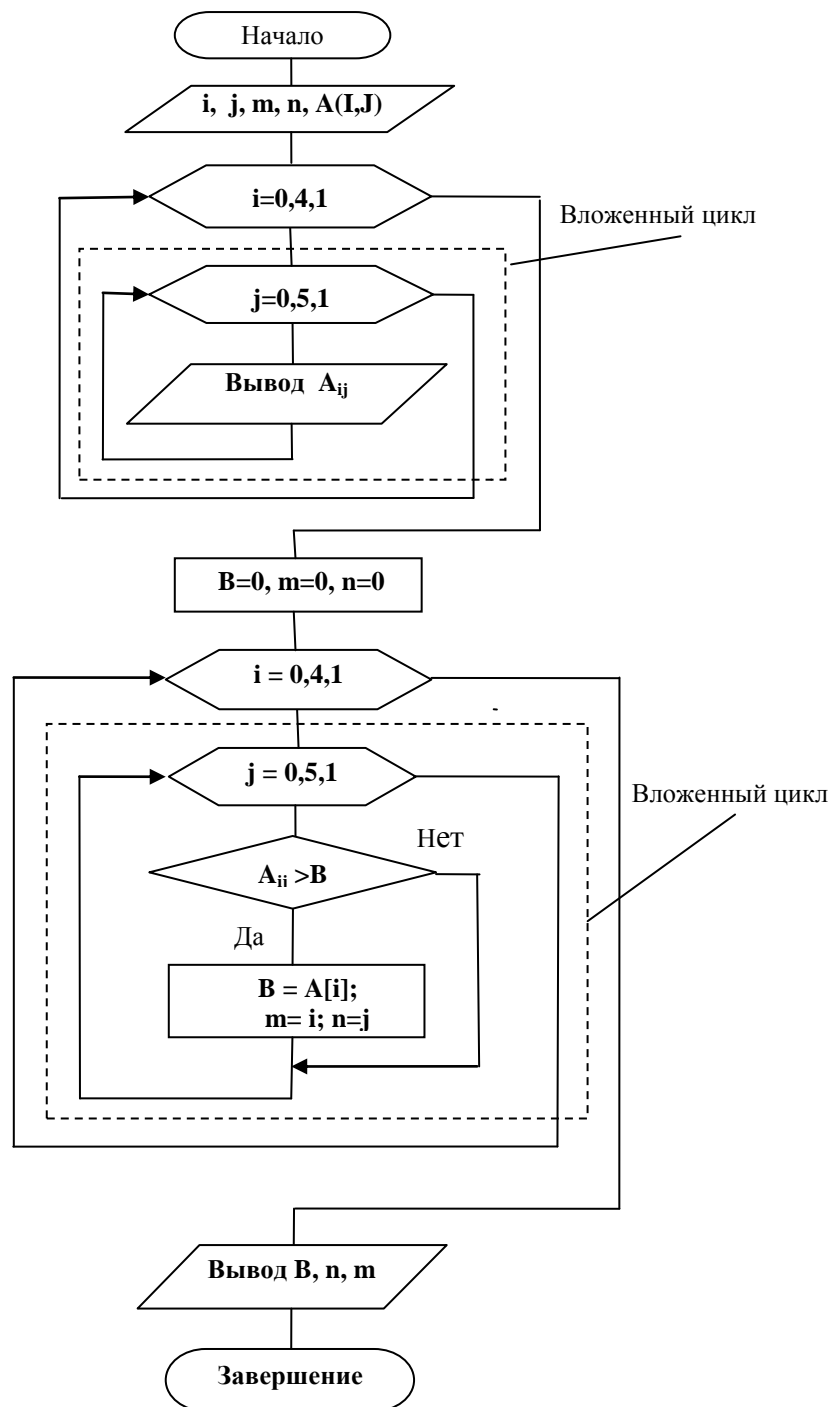



Рис. 4. Блок-схема алгоритма поиска максимального элемента матрицы

Экран после выполнения программы должен иметь следующий вид:

```

Matrica A
1   2   18  3   20
4   5   6   7   8
9   10  19  11  12
13  14  15  16  17
Max= 20   Stroka 1   Stolbec 5
  
```

Задание 6.5. Исследовать способы ввода и вывода двумерных массивов, а также следующие действия с матрицей: найти минимальный элемент каждой строки матрицы **A**, составить из них вектор **R** (одномерный массив) и определить суммы его четных и нечетных элементов. Элементы заданной матрицы из 20 произвольных чисел размера 4×5 (см. в результатах) ввести с клавиатуры. На экран вывести исходную матрицу **A**, вектор **R** и вычисленные суммы.

Блок-схема алгоритма решения данной задачи не приводится, т. к. она аналогична блок-схеме на рис. 4. Для решения задачи необходимо ввести и отладить следующий программный код:

```

#include "stdafx.h"
#include <conio.h>
#include "iostream"
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
const int M=10, N=10; //Максимальное число строк и столбцов матрицы
int A[M][N], R[M]; //Декларирование матрицы A и вектора R
int i, j, k, m, S1, S2; // Декларирование индексов и вспомогательных переменных
cout<<"Vvedite chislo strok i stolbcov matricu A :"<<endl;
cin>>m>>k; //Ввод реальных размеров исходной матрицы A
cout<<"Vvedite postrochno elementu matricu A :"<<endl;
for(i=0; i<m; i++) //Внешний цикл вывода элементов матрицы A
for(j=0; j<k; j++)
cin>>A[i][j]; //Ввод элементов исходной матрицы A

//Поиск минимального элемента каждой строки исходной матрицы A
for(i=0; i<m; i++)
{
R[i]=A[i][0]; //Начало составного оператора в цикле
for(j=0; j<k; j++) //Задание начального значения минимального элемента строки
if(A[i][j]<R[i]) //Определение минимального элемента
R[i]=A[i][j]; //каждой строки исходной матрицы A
} //Запись его в массив R
//Конец составного оператора в цикле
S1=0; //Задание начальных значений сумм четных
S2=0; //и нечетных элементов вектора R

//Операции по определению суммы четных и нечетных элементов массива R
for(i=0; i<m; i++)
{
if(R[i]%2==0) //Начало составного оператора в цикле
S2=S2+R[i]; //Условие – есть ли остаток от деления на 2 (четное или нет)
else S1=S1+R[i]; //Определение суммы четных элементов вектора R
} //Определение суммы нечетных элементов вектора R
//Конец составного оператора в цикле
cout<<"Matrica A"<<endl;
for(i=0; i<m; i++) //Внешний цикл вывода элементов матрицы A
{ //Начало составного оператора в цикле
for(j=0; j<k; j++)
cout<<A[i][j]<<" "; //Вывод элементов матрицы A
cout<<endl; //Конец составного оператора в цикле
}
cout<<endl<<"Vektor R"<<endl;
for(i=0; i<m; i++)
cout<<R[i]<<" "; //Вывод вектора R
cout<<endl;
cout<<"S1= "<<S1<<endl; //Вывод суммы S1 четных элементов
cout<<"S2= "<<S2<<endl; //Вывод суммы S2 нечетных элементов
getch();
return 0;
}

```

В результате выполнения вышеприведенной программы получим следующие результаты:

```

Vvedite chislo strok i stolbcov matricu A :
4 5
Vvedite postrochno elementu matricu A :
1 2 18 3 20
4 5 6 7 8
9 10 19 11 12
13 14 17 15 16
Matrica A
1      2      18      3      20
4      5      6      7      8
9      10     19     11     12
13     14     17     15     16
Vektor R
1      4      9      13
S1= 23
S2= 4

```

Задание 6.6. Задана квадратная матрица 4x4 (элементы – числа подряд от 1 до 16). Самостоятельно разработать программу в соответствии со своим вариантом (таб. 6.2). Номер варианта определяется по номеру компьютера. Ввод исходной матрицы запрограммировать с клавиатуры. Исходную матрицу, полученную матрицу и другие результаты вывести на экран.

Перед вводом программного кода самостоятельно разработать и начертить в отчете по лабораторной работе блок-схему алгоритма решения данной задачи и записать формулу для расчета на языке C++.

Таблица 6.2

Задание для самостоятельного программирования

№	Условие задания
1	Заменить нолями все элементы главной диагонали заданной матрицы и нижележащие элементы
2	Найти сумму всех ее элементов и заменить ею элементы главной диагонали заданной матрицы
3	Найти сумму и произведение всех отрицательных элементов заданной матрицы
4	Найти сумму минимальных элементов каждого столбца заданной матрицы
5	Найти среднее арифметическое максимального и минимального значений элементов заданной матрицы
6	Найти среднее арифметическое каждого из столбцов заданной матрицы
7	Получить новую матрицу путем деления всех элементов заданной матрицы на ее максимальный элемент
8	Получить новую матрицу путем деления всех элементов заданной матрицы на ее минимальный элемент
9	Найти сумму и произведение всех негативных элементов заданной матрицы
10	Найти сумму первых четырех элементов заданной матрицы и заменить ею элементы первой строки.
11	Найти сумму максимальных элементов каждого столбца заданной матрицы
12	Получить новую матрицу вычитанием из каждого элемента заданной матрицы ее минимального элемента
13	Заменить элементы первой строки заданной матрицы элементами ее второго столбца
14	Найти сумму элементов четных столбцов заданной матрицы

Задание 6.7. Задана квадратная матрица 4x4 (элементы – числа подряд от 1 до 16). Составить программу в соответствии со своим вариантом (таб. 6.1). Номер варианта определяется по номеру компьютера плюс два. Ввод исходной матрицы осуществить в программе. Исходную матрицу, полученную матрицу и другие результаты вывести на экран.

Перед вводом программного кода самостоятельно разработать и начертить в отчете по лабораторной работе блок-схему алгоритма решения данной задачи и записать формулу для расчета на языке C++.

Контрольные вопросы

1. Раскрыть понятие многомерного массива?
2. Как объявляются многомерные массивы?
3. Как выполняются внешние и вложенные циклы **for**?
4. Как обратиться к элементу матрицы?
5. Как выполняются различные операции над матрицами?
4. Как в программе на C++ будет выведен на экран двухмерный массив $A[k][k]$ оператором `for(i=0; i<k; i++) for(j=0; j<k; j++) cout<<A[i][j]<<endl; ?`
 - 1) В виде строки
 - 2) В виде столбца
 - 3) В виде матрицы
6. Как будет выведен на экран двухмерный массив в фрагменте программы на C++?


```
int i, j, k, m;
for(i=0; i<k; i++)
  for (j=0; j<m; j++)
    cout<<A[i][j]<<" ";
```

 - 1) В виде строки
 - 2) В виде столбца
 - 3) В виде матрицы
8. Что будет выведено на экран в фрагменте программы на C++?


```
for(i=0; i<n; i++)
  for (j=0; j<n; j++)
    cout<<A[0][j];
```

 - 1) Первая строка матрицы, повторенная **n** раз
 - 2) Первый столбец матрицы, повторенный **n** раз
 - 3) Полная матрица