|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | ***ИУК «Информатика и управление»*** |
| **КАФЕДРА** | ***ИУК4 «Программное обеспечение ЭВМ,*** |
|  | ***информационные технологии»*** |

**ДОМАШНЯЯ РАБОТА №1**

**«Обработка двумерных массивов»**

**ДИСЦИПЛИНА: «Высокоуровневое программирование»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-11Б | |  |  | ( | Суриков Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Пчелинцева Н.И. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

**Цель:** Получить практические навыки по созданию двумерных массивов и их обработки на языке С++.

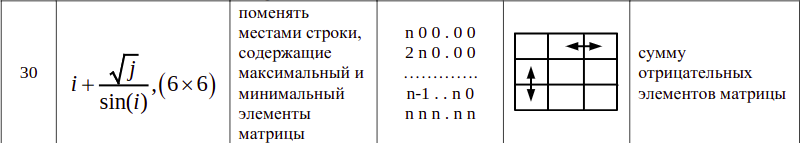
**Задачи:**

1. Изучить синтаксис и важнейшие особенности создания и обработки двумерных массивов.
2. Изучить основные операции над двумерными массивами.
3. Научиться составлять и реализовывать алгоритмы для вычислений над двумерными массивами.
4. Научиться оформлять алгоритмы подпрограмм, в которых используются двумерные массивы, с помощью блок-схем.

**Общее задание:**

1. Получить матрицу А (m\*n) образованную по соответствующему закону (размер матрицы и закон, по которому определяется каждый элемент матрицы А, для каждого варианта задания приведены в столбце 2 таблицы).
2. Получить матрицу В, осуществив соответствующие преобразования над матрицей А (преобразование, которое необходимо сделать над матрицей А, для каждого варианта задания приведено в столбце 3 таблицы).
3. Построить матрицу С (6\*6) соответствующего вида (вид матрицы С для каждого варианта задания приведен в столбце 4 таблицы).
4. Вычислить D = B \* C.
5. Получить матрицу Е, переставив блоки матрицы D (необходимые перестановки для каждого варианта задания приведены в столбце 5 таблицы).
6. Сделать соответствующие вычисления для матрицы Е (требуемые вычисления приведены в столбце 6 таблицы).

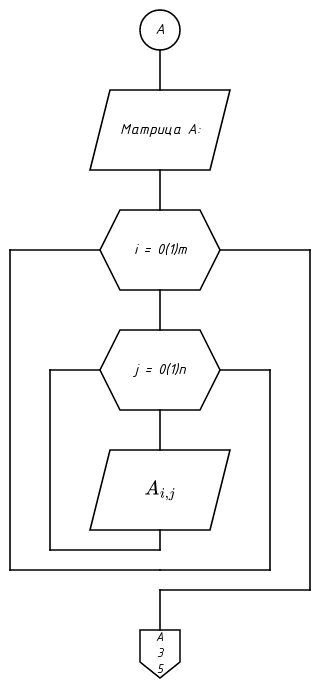
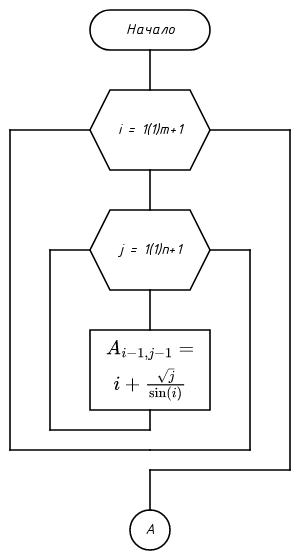
На экран необходимо последовательно вывести матрицы А, В, С, D, Е и результат вычисления пункта 6 задания.



**Вариант №30**

**Пункт 1:**

*Блок схема:*



*Листинг фрагмента программы на C++:*

#include <iostream>

#include <cmath>

#include <iomanip>

#define PI 3.14159265

**using** **namespace** **std**;

**int** main()

{

**const** **int** m = 6;

**const** **int** n = 6;

// Строим матрицу А

**int** A[m][n]{};

**for** (**int** i = 1; i < m + 1; i++)

{

**for** (**int** j = 1; j < n + 1; j++)

{

A[i - 1][j - 1] = i + sqrt(j) / sin(i \* PI / 180);

}

}

cout << "Матрица A:" << endl;

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < n; j++)

{

cout << setw(5) << A[i][j] << " ";

}

cout << endl;

}

cout << endl;

*Результат выполнения:*

*Матрица A:*

*58 82 100 115 129 141*

*30 42 51 59 66 72*

*22 30 36 41 45 49*

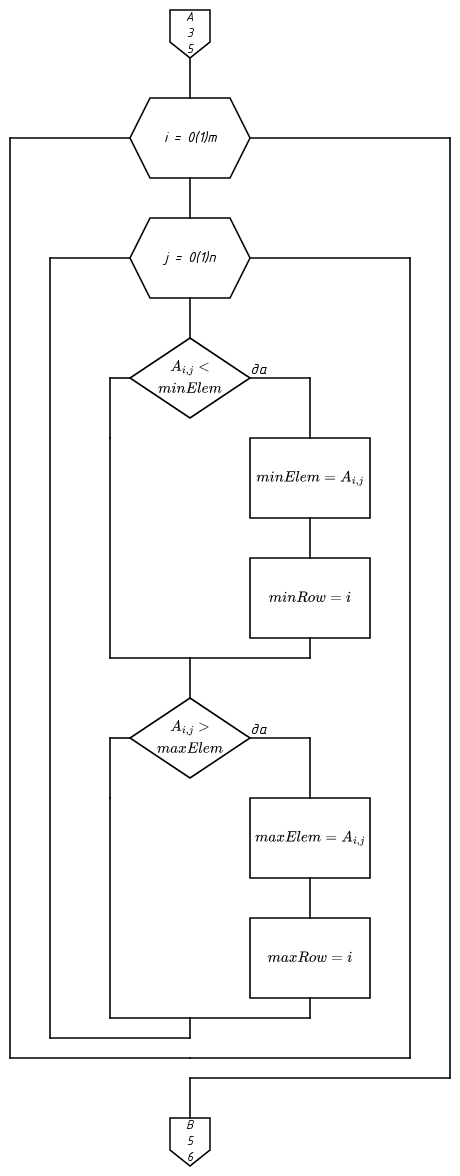
*18 24 28 32 36 39*

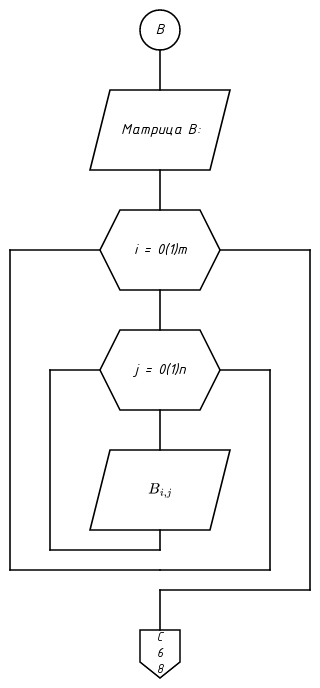
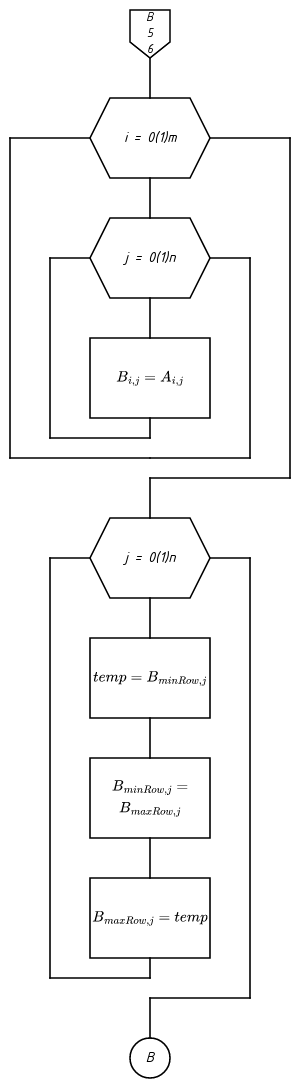
*16 21 24 27 30 33*

*15 19 22 25 27 29*

**Пункт 2:**

*Блок схема:*





*Листинг фрагмента программы на C++:*

// Получаем индексы максимального и минимального элементов A

**int** minRow{}, maxRow{};

**int** minElem = A[0][0], maxElem = A[0][0];

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < n; j++)

{

**if** (A[i][j] < minElem)

{

minElem = A[i][j];

minRow = i;

}

**if** (A[i][j] > maxElem)

{

maxElem = A[i][j];

maxRow = i;

}

}

}

// Копируем матрицу A в B

**int** B[m][n]{};

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < n; j++)

{

B[i][j] = A[i][j];

}

}

// Меняем местами строки с минимальным и максимальным элементами

**for** (**int** j = 0; j < n; j++)

{

**int** temp = B[minRow][j];

B[minRow][j] = B[maxRow][j];

B[maxRow][j] = temp;

}

cout << "Матрица B:" << endl;

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < n; j++)

{

cout << setw(5) << B[i][j] << " ";

}

cout << endl;

}

cout << endl;

*Результат выполнения:*

*Матрица B:*

*15 19 22 25 27 29*

*30 42 51 59 66 72*

*22 30 36 41 45 49*

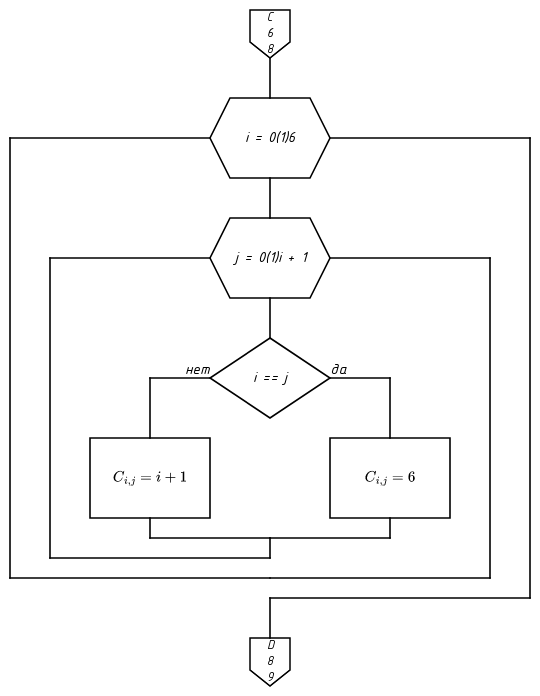
*18 24 28 32 36 39*

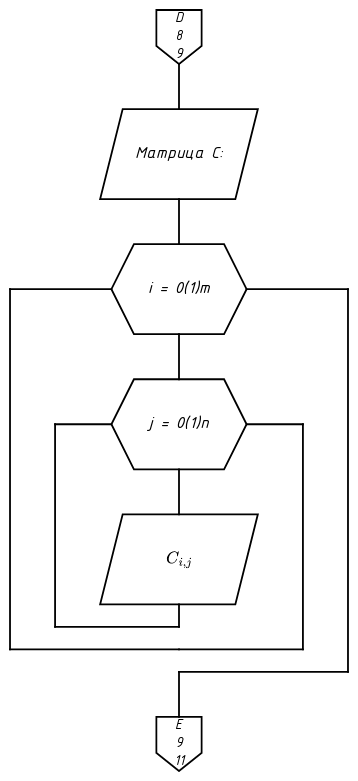
*16 21 24 27 30 33*

*58 82 100 115 129 141*

**Пункт 3:**

*Блок схема:*





*Листинг фрагмента программы на C++:*

// Строим матрицу С

**int** C[6][6]{};

**for** (**int** i = 0; i < 6; i++)

{

**for** (**int** j = 0; j <= i; j++)

{

**if** (i == j)

C[i][j] = 6;

**else**

C[i][j] = i + 1;

}

}

cout << "Матрица C:" << endl;

**for** (**int** i = 0; i < 6; i++)

{

**for** (**int** j = 0; j < 6; j++)

{

cout << setw(5) << C[i][j] << " ";

}

cout << endl;

}

cout << endl;

*Результат выполнения:*

*Матрица C:*

*6 0 0 0 0 0*

*2 6 0 0 0 0*

*3 3 6 0 0 0*

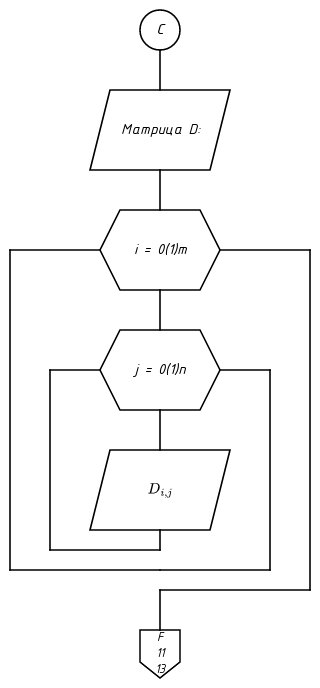
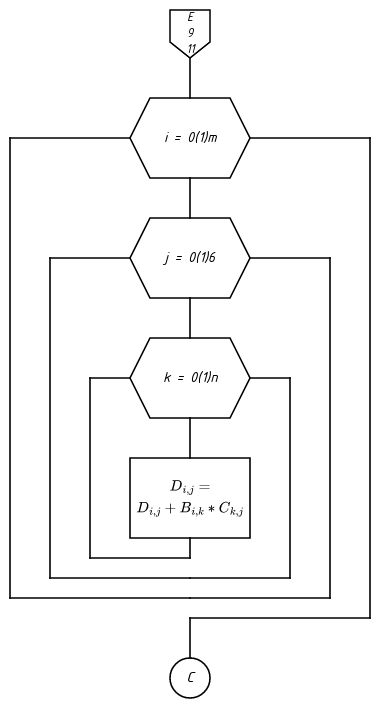
*4 4 4 6 0 0*

*5 5 5 5 6 0*

*6 6 6 6 6 6*

**Пункт 4:**

*Блок схема:*



*Листинг фрагмента программы на C++:*

// Вычисляем D=B\*C

**int** D[m][6]{};

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < 6; j++)

{

**for** (**int** k = 0; k < n; k++)

{

D[i][j] += B[i][k] \* C[k][j];

}

}

}

cout << "Матрица D:" << endl;

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < 6; j++)

{

cout << setw(5) << D[i][j] << " ";

}

cout << endl;

}

cout << endl;

*Результат выполнения:*

*Матрица D:*

*603 589 541 459 336 174*

*1415 1403 1304 1116 828 432*

*983 971 899 765 564 294*

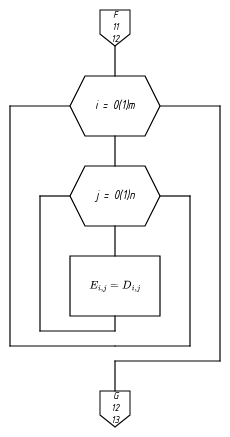
*782 770 710 606 450 234*

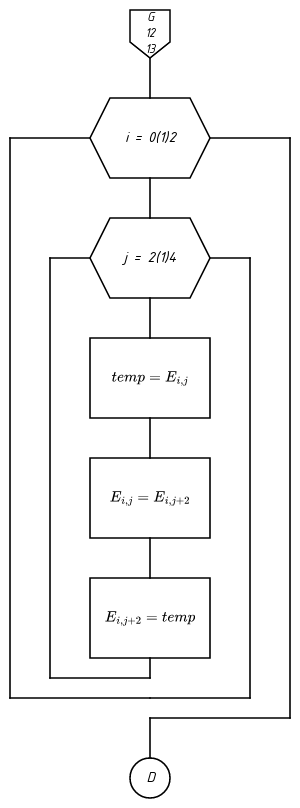
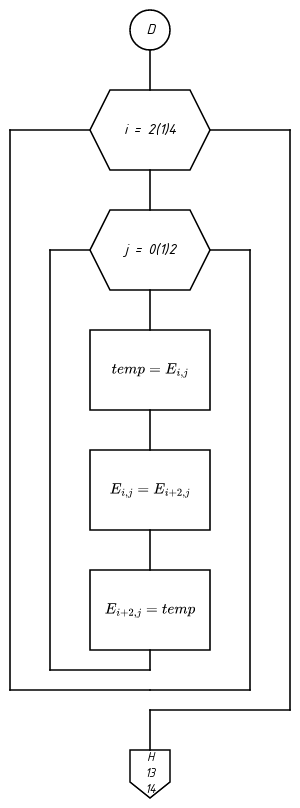
*666 654 600 510 378 198*

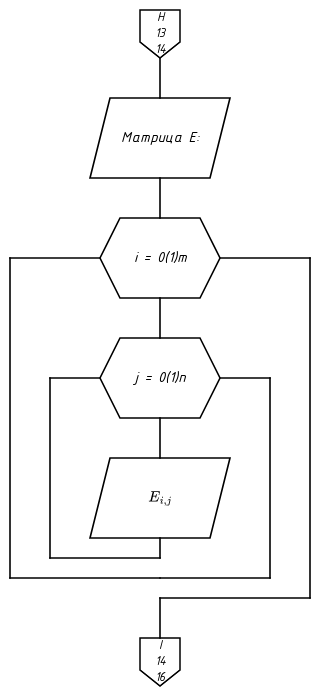
*2763 2743 2551 2181 1620 846*

**Пункт 5:**

*Блок схема:*





*Листинг фрагмента программы на C++:*

// Копируем матрицу D в E

**int** E[6][6]{};

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < n; j++)

{

E[i][j] = D[i][j];

}

}

// Меняем 2 и 3 блоки

**for** (**int** i = 0; i < 2; i++)

{

**for** (**int** j = 2; j < 4; j++)

{

**int** temp = E[i][j];

E[i][j] = E[i][j + 2];

E[i][j + 2] = temp;

}

}

// Меняем 4 и 7 блоки

**for** (**int** i = 2; i < 4; i++)

{

**for** (**int** j = 0; j < 2; j++)

{

**int** temp = E[i][j];

E[i][j] = E[i + 2][j];

E[i + 2][j] = temp;

}

}

cout << "Матрица E:" << endl;

**for** (**int** i = 0; i < 6; i++)

{

**for** (**int** j = 0; j < 6; j++)

{

cout << setw(5) << E[i][j] << " ";

}

cout << endl;

}

cout << endl;

*Результат выполнения:*

*Матрица E:*

*603 589 336 174 541 459*

*1415 1403 828 432 1304 1116*

*666 654 899 765 564 294*

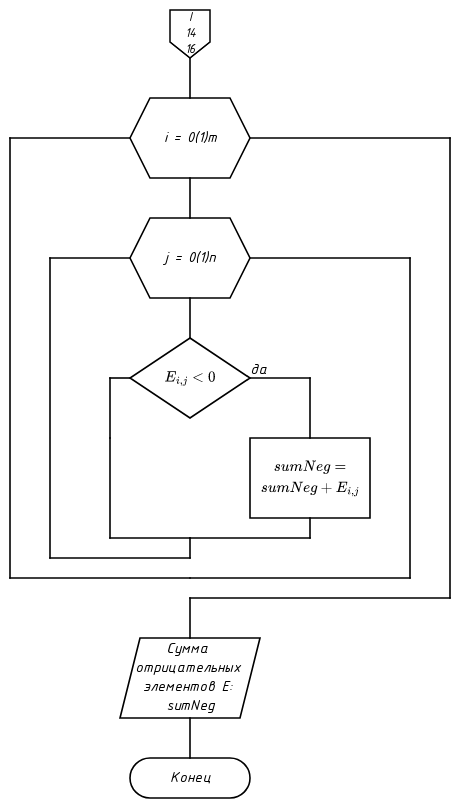
*2763 2743 710 606 450 234*

*983 971 600 510 378 198*

*782 770 2551 2181 1620 846*

**Пункт 6:**

*Блок схема:*



*Листинг фрагмента программы на C++:*

// Делаем вычисления для матрицы Е

**int** sumNeg{};

**for** (**int** i = 0; i < m; i++)

{

**for** (**int** j = 0; j < n; j++)

{

**if** (E[i][j] < 0)

sumNeg += E[i][j];

}

}

cout << "Сумма отрицательных элементов матрицы E: " << sumNeg << endl;

**return** 0;

}

*Результат выполнения:*

*Сумма отрицательных элементов матрицы E: 0*

**Вывод**

В ходе данной домашней работы я получил практические навыки по созданию двумерных массивов и их обработки на языке С++. Я изучил синтаксис и важнейшие особенности создания и обработки двумерных массивов, основные операции над двумерными массивами, научился составлять и реализовывать алгоритмы для вычислений над двумерными массивами, оформлять алгоритмы подпрограмм, в которых используются двумерные массивы, с помощью блок-схем.

**Литература**

1. Курс лекций доцента кафедры ФН1-КФ Пчелинцевой Н.И.
2. Программирование на языке высокого уровня С/С++ [Электронный ресурс]: конспект лекций / – Электрон. текстовые данные. – М.: Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2016. – 140 c. – Режим доступа: http://www.iprbookshop.ru/48037.
3. Зырянов, К. И. Программирование на C++: учебное пособие / К. И. Зырянов, Н. П. Кисленко. — Новосибирск: Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2017. — 129 c. — ISBN 978-5-7795-0817- 9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/85873.html
4. Фридман, А. Л. Язык программирования C++: учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 c. — ISBN 978-5-4497-0920-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/102076.html