Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

Выполнила:

Студентка 1 курса 7 группы

Бабич Виолетта Станиславовна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| **2** | 1. Задан массив **A** из **n** элементов. Проверить, есть ли в нём отрицательные элементы. Если есть, то найти наибольшее значение **k**, при котором **A[k] < 0**.  2. Дана матрица. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент оказался в верхнем левом углу. |

№1

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

const int n = 10;

int A[n];

// Ввод элементов массива

cout << "Введите элементы массива A:" << endl;

for (int i = 0; i < n; ++i) {

cout << "A[" << i << "]: ";

cin >> A[i];

}

// Проверка наличия отрицательных элементов

bool hasNegative = false;

int maxNegativeIndex = -1;

// Проверяем все элементы и находим наибольший отрицательный и его индекс

for (int i = 0; i < n; ++i) {

if (A[i] < 0) {

hasNegative = true;

maxNegativeIndex = i;

}

}

// Вывод результата

if (hasNegative) {

cout << "В массиве есть отрицательные элементы." << endl;

cout << "Наибольший индекс k, при котором A[k] < 0: " << maxNegativeIndex << endl;

}

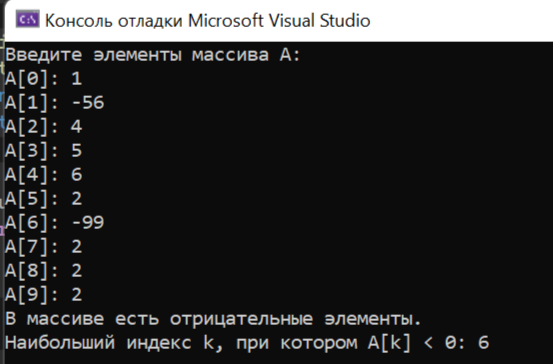
else {

cout << "В массиве нет отрицательных элементов." << endl;

}

return 0;

}



№2

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int n, m;

// Ввод размеров матрицы

cout << "Введите количество строк матрицы: ";

cin >> n;

cout << "Введите количество столбцов матрицы: ";

cin >> m;

// Проверка на допустимый размер матрицы

if (n <= 0 || m <= 0) {

cerr << "Некорректный размер матрицы." << endl;

return 1;

}

// Ввод элементов матрицы

vector<vector<int>> matrix(n, vector<int>(m));

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < n; ++i) {

for (int j = 0; j < m; ++j) {

cin >> matrix[i][j];

}

}

// Нахождение индексов наибольшего элемента

auto maxElement = max\_element(matrix[0].begin(), matrix[0].end());

int maxCol = distance(matrix[0].begin(), maxElement);

int maxRow = 0;

for (int i = 0; i < n; ++i) {

if (matrix[i][maxCol] > matrix[maxRow][maxCol]) {

maxRow = i;

}

}

// Перестановка строк и столбцов

swap(matrix[0], matrix[maxRow]);

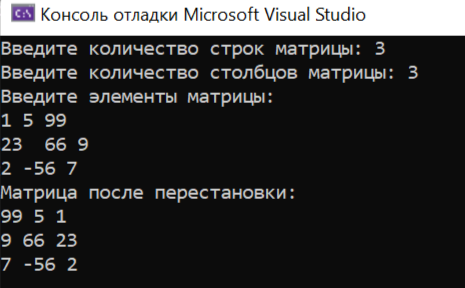
for (int i = 0; i < n; ++i) {

swap(matrix[i][0], matrix[i][maxCol]);

}

// Вывод измененной матрицы

cout << "Матрица после перестановки:" << endl;

 for (int i = 0; i < n; ++i) {

for (int j = 0; j < m; ++j) {

cout << matrix[i][j] << ' ';

}

cout << endl;

}

return 0;

}

**Дополнительные задания**

|  |  |
| --- | --- |
| **11** | 1. Задан массив **A** из **n** элементов. Проверить, есть ли в нём элементы, равные нулю. Если есть, найти наименьшее **k**, при котором **A[k] = 0**.  2. Для заданной матрицы размером 4 на 4 найти такие **k**, при которых **k**-я строка матрицы совпадает с **k**-м столбцом. |

№1

#include <iostream>

using namespace std;

int main() {

setlocale(0, "rus");

int n;

cout << "Введите размер массива: ";

cin >> n;

int\* arr = new int[n]; // Создаем динамический массив

cout << "Введите элементы массива: ";

for (int i = 0; i < n; ++i) {

std::cin >> arr[i];

}

int zeroIndex = -1; // Индекс первого встреченного элемента, равного нулю

// Поиск элементов, равных нулю

for (int i = 0; i < n; ++i) {

if (arr[i] == 0) {

zeroIndex = i;

break; // Прерываем цикл при нахождении первого нулевого элемента

}

}

if (zeroIndex != -1) {

cout << "Есть элементы, равные нулю." << std::endl;

cout << "Наименьший индекс k, при котором A[k] = 0: " << zeroIndex << std::endl;

}

else {

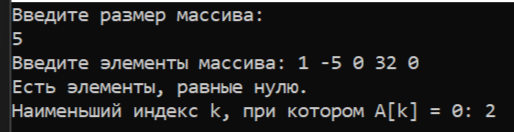
cout << "В массиве нет элементов, равных нулю." << std::endl;

}

delete[] arr; // Освобождаем память

return 0;

}



№2

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, s = 0;

cout << "Введите размер массива: ";

cin >> n;

bool t = false;

// Выделение памяти под двумерный массив

int\*\* a = new int\* [n];

for (int i = 0; i < n; i++) {

a[i] = new int[n];

// Заполнение массива значениями, вводимыми с клавиатуры

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

cin >> a[i][j];

}

}

// Поиск одинаковых элементов в строке и столбце с одинаковым индексом

for (int i = 0; i < n && !t; i++) {

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

if (a[i][j] != a[j][i]) {

break;

}

s++;

}

// Если количество совпадающих элементов равно порядку матрицы

if (s == n) {

cout << "Индекс равных строки и столбца = " << i << endl;

t = true;

}

s = 0;

}

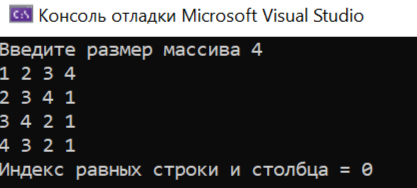
// Вывод сообщения, если не найдены строки и столбцы с одинаковым индексом

if (!t) {

cout << "Никакие строка и столбец с одинаковым индексом не равны" << endl;

}

// Освобождение выделенной памяти

 for (int i = 0; i < n; i++) {

delete[] a[i];

}

delete[] a;

return 0;

}

|  |  |
| --- | --- |
| **8** | 1. В одномерном массиве, состоящем из **k** целых элементов, вычислить количество положительных элементов массива и сумму элементов массива, расположенных после последнего элемента, равного нулю.  2. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. |

№1

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int k;

// Ввод размера массива

cout << "Введите размер массива: ";

cin >> k;

int\* arr = new int[k];

// Ввод элементов массива

cout << "Введите элементы массива: ";

for (int i = 0; i < k; ++i) {

cin >> arr[i];

}

int positiveCount = 0;

int sumAfterLastZero = 0;

bool zeroFound = false;

// Перебор элементов массива

for (int i = 0; i < k; ++i) {

// Подсчет положительных элементов

if (arr[i] > 0) {

positiveCount++;

}

// Поиск последнего элемента, равного нулю

if (arr[i] == 0) {

zeroFound = true;

// Обнуление счетчика суммы после нуля

sumAfterLastZero = 0;

}

// Суммирование элементов после последнего нуля

if (zeroFound) {

sumAfterLastZero += arr[i];

}

}

// Вывод результата

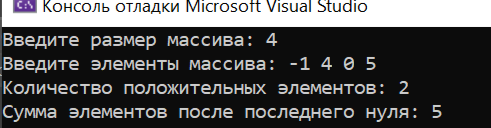
cout << "Количество положительных элементов: " << positiveCount << endl;

cout << "Сумма элементов после последнего нуля: " << sumAfterLastZero << endl;

delete[] arr;

return 0;

}



№2

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int rows, cols;

// Ввод размеров матрицы

cout << "Введите количество строк матрицы: ";

cin >> rows;

cout << "Введите количество столбцов матрицы: ";

cin >> cols;

// Проверка на корректность размеров

if (rows <= 0 || cols <= 0) {

cout << "Размеры матрицы должны быть положительными числами." << endl;

return 1;

}

int\*\* matrix = new int\* [rows];

// Ввод элементов матрицы

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < rows; ++i) {

matrix[i] = new int[cols];

for (int j = 0; j < cols; ++j) {

cin >> matrix[i][j];

}

}

int firstColumnWithZero = -1;

// Поиск столбца с хотя бы одним нулевым элементом

for (int j = 0; j < cols; ++j) {

for (int i = 0; i < rows; ++i) {

if (matrix[i][j] == 0) {

firstColumnWithZero = j;

break;

}

}

if (firstColumnWithZero != -1) {

break;

}

}

// Вывод результата

if (firstColumnWithZero != -1) {

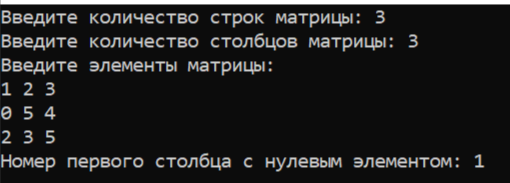
cout << "Номер первого столбца с нулевым элементом: " << firstColumnWithZero + 1 << endl;

}

else {

cout << "В матрице нет столбцов с нулевым элементом." << endl;

}

 // Освобождение памяти

for (int i = 0; i < rows; ++i) {

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

|  |  |
| --- | --- |
| **12** | 1. Задан массив **A** из **n** элементов. Подсчитать, сколько раз встречается в нем максимальное число.  2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные. |

№1

#include <iostream>

#include <climits>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int n;

// Ввод размера массива

cout << "Введите размер массива: ";

cin >> n;

int\* array = new int[n];

// Ввод элементов массива

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < n; ++i) {

cin >> array[i];

}

// Поиск максимального элемента и подсчет его количества

int maxElement = INT\_MIN;

int countMax = 0;

for (int i = 0; i < n; ++i) {

if (array[i] > maxElement) {

maxElement = array[i];

countMax = 1;

}

else if (array[i] == maxElement) {

countMax++;

}

}

// Вывод результата

cout << "Максимальное число: " << maxElement << endl;

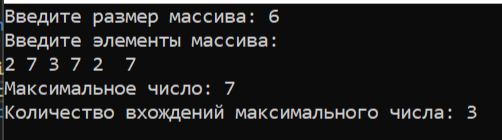
cout << "Количество вхождений максимального числа: " << countMax << endl;

// Освобождение памяти

delete[] array;

return 0;

}



№2

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int rows, cols;

// Ввод размеров матрицы

cout << "Введите количество строк матрицы: ";

cin >> rows;

cout << "Введите количество столбцов матрицы: ";

cin >> cols;

// Проверка на корректность размеров

if (rows <= 0 || cols <= 0) {

cout << "Размеры матрицы должны быть положительными числами." << endl;

return 1;

}

int\*\* matrix = new int\* [rows];

// Ввод элементов матрицы

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < rows; ++i) {

matrix[i] = new int[cols];

for (int j = 0; j < cols; ++j) {

cin >> matrix[i][j];

}

}

bool foundPositiveRow = false;

int positiveRowNumber = -1;

// Поиск строки с положительным элементом

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

if (matrix[i][j] > 0) {

foundPositiveRow = true;

positiveRowNumber = i;

break;

}

}

if (foundPositiveRow) {

break;

}

}

// Вывод результата

if (foundPositiveRow) {

cout << "Найдена строка с положительным элементом. Номер строки: " << positiveRowNumber + 1 << endl;

// Изменение знаков элементов предыдущей строки

if (positiveRowNumber > 0) {

for (int j = 0; j < cols; ++j) {

matrix[positiveRowNumber - 1][j] \*= -1;

}

cout << "Знаки элементов предыдущей строки изменены на противоположные." << endl;

}

}

else {

cout << "В матрице нет строк, содержащих положительный элемент." << endl;

}

// Освобождение памяти

for (int i = 0; i < rows; ++i) {

delete[] matrix[i];

}

delete[] matrix;

return 0;

}

