Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 17

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели и ссылки при работе с функциями»

Выполнила:

Студентка 1 курса 7 группы

Бабич Виолетта Станиславовна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| 2 | 1. В массиве из целых чисел найти в процентах частоту появления каждого из **k** наиболее часто встретившихся чисел (**k** – натуральное число, не превосходящее числа элементов массива).  2. Дана целочисленная прямоугольная матрица. Если есть в матрице нулевые элементы, то определить количество строк, содержащих их. |

№1

#include <iostream>

#include <algorithm>

#include <unordered\_map>

#include <vector>

using namespace std;

void findFrequency(const int\* arr, int size, int k, unordered\_map<int, double>& result) {

unordered\_map<int, int> frequencyMap;

// Заполнение frequencyMap частотами чисел

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

frequencyMap[arr[i]]++;

}

// Сортировка частот в порядке убывания

vector<pair<int, int>> sortedFrequency(frequencyMap.begin(), frequencyMap.end());

sort(sortedFrequency.begin(), sortedFrequency.end(),

[](const pair<int, int>& a, const pair<int, int>& b) {

return a.second > b.second;

});

// Находим частоту каждого из k наиболее часто встречающихся чисел в процентах

int totalElements = std::min(k, static\_cast<int>(sortedFrequency.size()));

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

result[sortedFrequency[i].first] = (static\_cast<double>(sortedFrequency[i].second) / size) \* 100;

}

}

int main() {

setlocale(LC\_ALL, "rus");

const int size = 10; // задаем размер массива

int arr[size] = { 1, 2, 2, 3, 3, 3, 4, 4, 4, 4 }; // задаем массив

int k = 3;

std::unordered\_map<int, double> result;

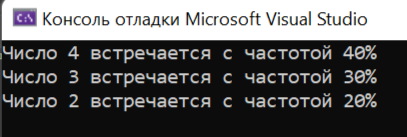
// подключаем функцию

findFrequency(arr, size, k, result);

// Вывод результатов

for (const auto& entry : result) {

cout << "Число " << entry.first << " встречается с частотой " << entry.second << "%" << endl;

 }

return 0;

}

№2

#include <iostream>

using namespace std;

//Если есть в матрице нулевые элементы, то определить количество строк, содержащих их.

void countRowsWithZeros(const int\* matrix, int rows, int cols, int& rowCount) {

rowCount = 0;

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

bool hasZero = false;

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

if (matrix[i \* cols + j] == 0) {

hasZero = true;

break;

}

}

if (hasZero) {

++rowCount;

}

}

}

int main() {

setlocale(LC\_ALL, "rus");

const int rows = 3; // Задем кол-во строк

const int cols = 4; // Задаем кол-во столбцов

int matrix[rows][cols] = {

{1, 2, 3, 4},

{5, 6, 7, 8},

{0, 9, 10, 11}

};

int rowCount;

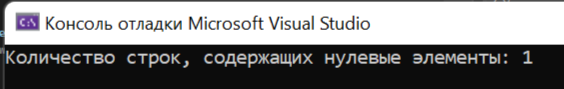
//Подключаем функцию

countRowsWithZeros(&matrix[0][0], rows, cols, rowCount);

cout << "Количество строк, содержащих нулевые элементы: " << rowCount << endl;

return 0;

}



|  |  |
| --- | --- |
| 11 | 1. Изменить одномерный массив, вычеркнув из него нечетные элементы.  2. Дана целочисленная квадратная матрица. Если она содержит отрицательные элементы, то определить сумму элементов в тех строках, где отрицательные элементы отсутствуют. |

№1

#include <iostream>

using namespace std;

int\* delete\_func(int\* a, int& n) { // создаем функцию для удаления нечетных элементов

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

if (a[i] % 2 != 0) {

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

a[j] = a[j + 1];

}

n--;

i--;

}

}

return a;

}

int main() { // функция main

setlocale(LC\_ALL, "rus");

int n;

cout << "Введите размер массива ";

cin >> n;

int\* a = new int[n];

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

// читаем введеный массив

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

cin >> a[i];

}

delete\_func(a, n); // вызываем функцию

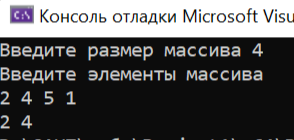
// выводим проебразованнный массив

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

cout << a[i] << ' ';

}

}



№2

#include <iostream>

using namespace std;

// создаем функцию, где, усли она содержит отрицательные элементы, то определить сумму элементов в тех строках, где отрицательные элементы отсутствуют.

void poisk\_sum(int\*\* a, int n) {

int s = 0, kol = 0;

bool t = false;

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

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

if (a[i][j] < 0) {

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

t = true;

break;

}

}

if (t == true) {

break;

}

}

if (t == true) {

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

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

if (a[i][j] >= 0) {

s += a[i][j];

kol++;

}

else {

break;

}

}

if (kol == n) {

cout << "Сумма элементов строки " << i << " равна " << s << endl;

}

s = 0;

kol = 0;

}

}

else {

cout << "В массиве нет отрицательных элемментов" << endl;

}

}

int main() {

setlocale(LC\_ALL, "rus");

int n, m;

cout << "Введите размер матрицы ";

cin >> n;

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

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

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

a[i] = new int[n];

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

cin >> a[i][j];

}

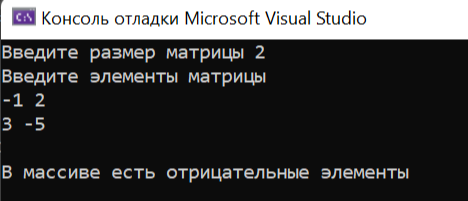
}

cout << endl;

// вызываем функцию для поиска отрицательного элемента

poisk\_sum(a, n);

}



|  |  |
| --- | --- |
| 12 | 1. Найти количество отрицательных элементов, стоящих на чётных местах в одномерном массиве.  2. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое. |

№1

#include <iostream>

using namespace std;

// функция для подсчета отрицательных элементов, которые стоят на четных местах в массиве

int poisk(int\* a, int n) {

int s = 0;

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

if ((i + 1) % 2 == 0 and a[i] < 0) {

s++;

}

}

return s;

}

int main() {

setlocale(LC\_ALL, "rus");

int n;

cout << "Введите размер массива ";

cin >> n; // считываем размер массива

int\* a = new int[n];

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

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

cin >> a[i];

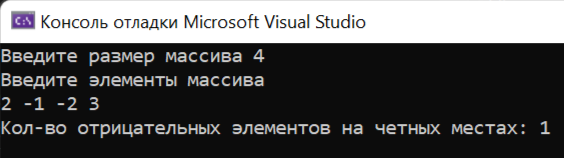
}

// вызаваем функцию для подсчета отрицательных элементов

int kol = poisk(a, n);

cout << "Кол-во отрицательных элементов на четных местах: " << kol << endl; // выводим количество таких элементов

}



№2

#include <iostream>

using namespace std;

// создаем функцию, которая проверяет, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер.Все элементы столбца с таким же номером уменьшить вдвое.

double\*\* zamena(double\*\* a, int n, int m) {

int s = 0;

bool t = false;

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

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

if (a[i][j] < 0) {

cout << "строка, которая содержит отрицательный элемент: " << i + 1 << endl;

s = i;

t = true;

break;

}

}

if (t == true) {

break;

}

}

if (t == true) {

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

a[i][s] /= 2;

}

}

else {

cout << "нет отрицательных элементов" << endl;

}

return a;

}

int main() {

setlocale(LC\_ALL, "rus");

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m;

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

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

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

a[i] = new double[m];

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

cin >> a[i][j];

}

}

cout << endl;

// вызываем функцию

zamena(a, n, m);

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

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

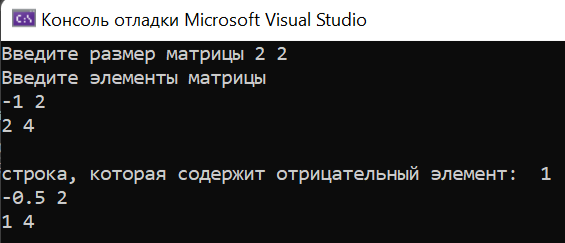
cout << a[i][j] << ' ';

}

cout << endl;

}

}



|  |  |
| --- | --- |
| 14 | 1. В одномерном массиве найти минимальный и максимальный элементы. Вычислить их разность.  2. Если в матрице имеется столбец, все элементы которого положительны, то знаки элементов предыдущего столбца изменить на противоположные. |

№1

#include <iostream>

using namespace std;

// поиск максимума

int poisk\_max(int\* a, int n) {

int max = a[0];

for (int i = 1; i < n; i++) {

if (a[i] > max) {

max = a[i];

}

}

return max;

}

// поиск минимума

int poisk\_min(int\* a, int n) {

int min = a[0];

for (int i = 1; i < n; i++) {

if (a[i] < min) {

min = a[i];

}

}

return min;

}

int main() {

setlocale(LC\_ALL, "rus");

int n;

cout << "Введите размер массива ";

cin >> n;

int\* a = new int[n];

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

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

cin >> a[i];

}

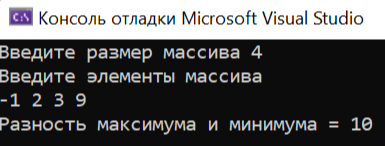
// вызываем функции

int max = poisk\_max(a, n);

int min = poisk\_min(a, n);

cout << "Разность максимума и минимума = " << (max - min) << endl;

}



№2

#include <iostream>

#include <iomanip>

using namespace std;

// Если в матрице имеется столбец, все элементы которого положительны, то знаки элементов предыдущего столбца изменить на противоположные.

int\*\* zamena(int\*\* a, int n, int m) {

int s = 0;

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

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

c[i] = new int[m];

}

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

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

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

s++;

}

else {

break;

}

}

if (s == n and i == 0) {

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

c[j][m - 1] = (-a[j][m - 1]);

}

}

else if (s == n) {

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

c[j][i - 1] = (-a[j][i - 1]);

}

}

else {

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

c[j][i - 1] = a[j][i - 1];

}

}

s = 0;

}

return c;

}

int main() {

setlocale(LC\_ALL, "rus");

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m;

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

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

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

a[i] = new int[m];

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

cin >> a[i][j];

}

}

cout << endl;

// вызываем функцию

a = zamena(a, n, m);

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

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

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

}

cout << endl;

}

}

