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:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| 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. |

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int\* delete\_func(int\* a, int &n) { // функция для удаления нечётных 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

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

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++) { // считываем массив

cin >> a[i];

}

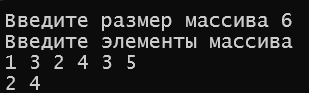
delete\_func(a, n); // вызываем функцию удаления нечётных элементов

for (int i = 0; i < n; i++) { // выводим получившийся массив

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

}

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

void poisk\_sum(int \*\*a, int n) { // функция поиска отрицательного элемента, и, если находит, ищет сумму элементов тех строк, где нет отрицательных элементов

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 с индексом " << i << " равна " << s << endl;

}

s = 0;

kol = 0;

}

}

else {

cout << "Ни одна из строк не содержит отрицательные элементы" << endl;

}

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m;

cout << "Введите размер матрицы ";

cin >> n; // считываем размер матрицы

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

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

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

a[i] = new int[n];

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

cin >> a[i][j];

}

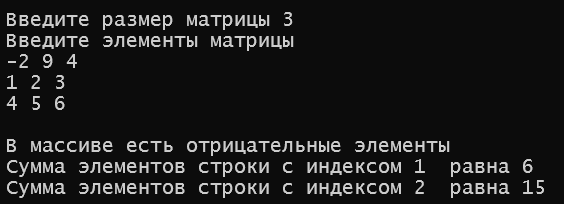
}

cout << endl;

poisk\_sum(a, n); // вызываем функцию поиска отрицательного элемента, и, если находим, ищем сумму элементов тех строк, где нет отрицательных элементов

}

|  |  |
| --- | --- |
| 16 | 1. В одномерном массиве найти максимальный из отрицательных элементов и поменять его местами с последним элементом массива.  2. Проверить, все ли строки матрицы содержат хотя бы один отрицательный элемент. Если да, то изменить знаки всех элементов матрицы на обратные. |



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int\* poisk\_max\_and\_zamena(int\* a, int n) { // функция, которая ищет максимальный отрицательный элемент и меняет его местами с последним элементом

int max = 0;

int pos = -1;

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

if (a[i] < 0 and max == 0) {

max = a[i];

pos = i;

}

else if (a[i] < 0 and a[i] > max) {

max = a[i];

pos = i;

}

}

if (max == 0) {

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

return a;

}

int t = a[pos];

a[pos] = a[n - 1];

a[n - 1] = t;

return a;

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

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++) { // считываем массив

cin >> a[i];

}

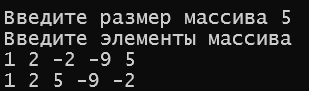
poisk\_max\_and\_zamena(a, n); // вызываем функцию поиска максимального отрицательного элемента и меняем его с последним

for (int i = 0; i < n; i++) { // выводим получившийся массив

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

}

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int \*\*poisk\_and\_zamena(int \*\*a, int n, int m) { // функция, которая проверяет наличие во всех строках хотя бы 1 отрицательного элемента, и, если во всех строках есть такой, меняет знаки всех элементов на противоположные

int s = 0;

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

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

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

s++;

break;

}

}

}

if (s == n) {

cout << "В каждой строке есть хотя бы 1 отрицательный элемент " << endl;

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

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

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

}

}

}

else {

cout << "Не в каждой строке есть отрицательный элемент" << endl;

}

return a;

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m; // считываем размер матрицы

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

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

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

a[i] = new int[m];

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

cin >> a[i][j];

}

}

poisk\_and\_zamena(a, n, m); // вызываем функцию, которая проверяет наличие во всех строках хотя бы 1 отрицательного элемента, и, если во всех строках есть такой, меняет знаки всех элементов на противоположные

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

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

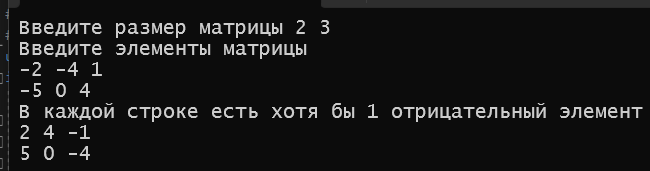
cout << a[i][j] << ' ';

}

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. |

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int poisk\_max(int \*a, int n) { // функция поиска максимума

int max = a[0];

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

if (a[i] > max) {

max = a[i];

}

}

return max;

}

int poisk\_min(int\* a, int n) { // функция поиска минимума

int min = a[0];

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

if (a[i] < min) {

min = a[i];

}

}

return min;

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

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++) { // считываем массив

cin >> a[i];

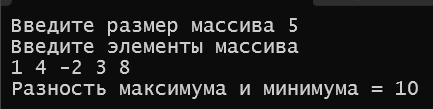
}

int max = poisk\_max(a, n); // вызываем функцию поиска максимума

int min = poisk\_min(a, n); // вызываем функцию поиска минимума

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

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int \*\*zamena(int \*\*a, int n, int m) { // функция замены элементов предыдущего столбца на противоположные при условии, что все элементы текущего положительные

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() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m; // считываем размер матрицы

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

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

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

a[i] = new int[m];

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

cin >> a[i][j];

}

}

cout << endl;

a = zamena(a, n, m); // вызываем функцию замены элементов предыдущего столбца на противоположные при условии, что все элементы текущего положительные

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

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

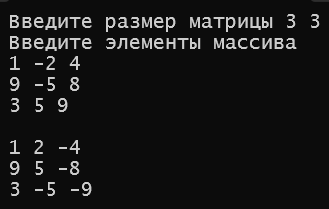
cout << a[i][j] << ' ';

}

cout << endl;

}

}



|  |  |
| --- | --- |
| 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. |

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int poisk(int \*a, int n) { // функция подсчёта количества отрицательных элементов, которые стоят на чётных местах

int s = 0;

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

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

s++;

}

}

return s;

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

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++) { // считываем массив

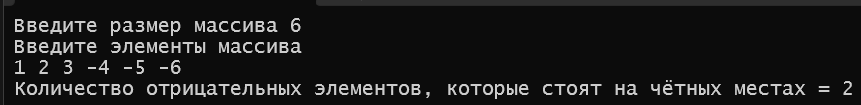
cin >> a[i];

}

int kol = poisk(a, n); // вызываем функцию подсчёта количества отрицательных элементов, которые стоят на чётных местах

cout << "Количество отрицательных элементов, которые стоят на чётных местах = " << kol << endl; // выводим количество таких элементов

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

double \*\*zamena(double \*\*a, int n, int m) { // функция поиска индекса первой строки с отрицательным элементом и уменьшения значения элементов столбца с таким же индексом, как и эта строка, в 2 раза

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 << "Первая строка, которая содержит отрицательный элемент имеет индекс " << i << 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 << "Ни одна из строк не содержит отрицательные элементы" << endl;

}

return a;

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m; // считываем размер матрицы

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

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

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

a[i] = new double[m];

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

cin >> a[i][j];

}

}

cout << endl;

zamena(a, n, m); // вызываем функцию поиска индекса первой строки с отрицательным элементом и уменьшения значения элементов столбца с таким же индексом, как и эта строка, в 2 раза

for (int i = 0; i < n; i++) { // выводим полученную матрицу

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

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

}

cout << endl;

}

}

