Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

Лабараторная работа №15

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Дынамічнае выдзяленне памяці»

Выканала:

Студэнтка 1 курса 6 группы

Літвінчук Дар'я Валер'еўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

**Варыянт 10**

|  |  |
| --- | --- |
| **10** | 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму положительных элементов массива и произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.  2. Найти в матрице первую строку, все элементы которой отрицательны. Увеличить все элементы матрицы на значение первого элемента найденной строки. |

#include <iostream>

#include <Windows.h>

int main()

{

using namespace std;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

float\* A;

float sum = 0, mult = 1;

int min = 0, max = 0;

cout << "Введите размер массива: ";

cin >> n;

A = (float\*)malloc(n \* sizeof(int)); //выделение памяти и проверка,

cout << "Введите массив: ";

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

{

cin >> A[i]; //вводим массива

}

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

{

if (A[i] > 0) //находим сумму положительных элементов

sum += A[i];

}

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

{

if (abs(A[i]) < abs(A[min]))

min = i;//номер элемента минимального по модулю

if (abs(A[i]) > abs(A[max]))

max = i; //номер элемента максимального по модулю

}

if (min > max) //если максимальный элемент идёт раньше минимального

swap(min, max);

for (int i = min + 1; i < max; i++)//элементы между минимальных и максимальным элементом

{

if (A[i]!=0)

mult \*= A[i];//находим произведение

}

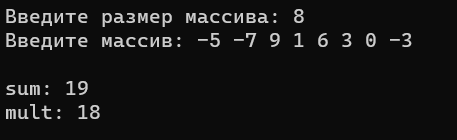
cout << endl;

cout << "sum: " << sum << endl;

cout << "mult: " << mult << endl;

free(A);//освобождение динамической памяти

}



#include <iostream>

#include <Windows.h>

int main()

{

using namespace std;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m, str\_num = 0, i, j, el;

bool flag = false;

int\*\* A;

cout << "введите количество строк n:";

cin >> n;

cout << "введите количество столбцов m:";

cin >> m;

A = new int\* [n];//выделение динам.памяти для двумерного массива

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

A[i] = new int[m];

}

cout << "введите символы: ";//вводим элементы для массива

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

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

cin >> A[i][j];

}

}

cout << "\n";

cout << " Первоначальная матрица:" << "\n";//выводим первоночальный массив

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

cout << "\n";

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

cout << A[i][j] << "\t";

}

}

for (i = 0; i < n; i++) {//ищем первую строку все элементы которой отрицательны

flag = false;

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

if (A[i][j] > 0) {//присваиваим элементы значение true, если он положительный

flag = true;

}

}

if (flag == false) {

el = A[i][0];

str\_num = i;

break;//досрочно заканчиваем программу, если нашли нужную строку

}

}

cout << "\n" << " Номер строки все элементы которой отрицатьельны: " << str\_num + 1;

cout << "\n \n Изменённая матрица:" << "\n";//выводим мзменённый массив

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

cout << "\n";;

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

A[i][j] += el;//увеличиваем каждый элемент массива на первый элемент найденной строки

cout << A[i][j] << "\t";

}

}

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

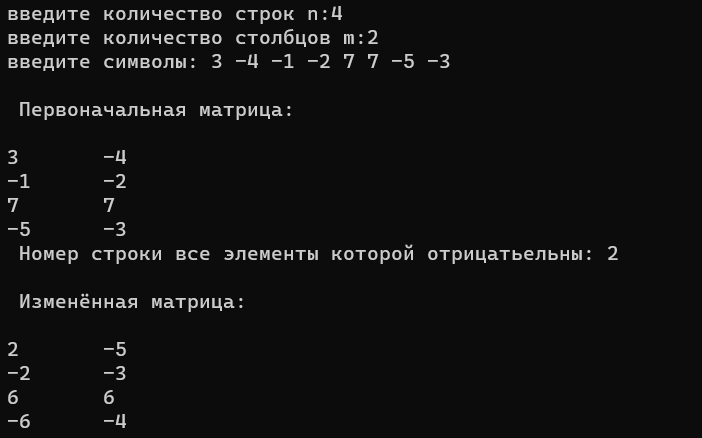
delete[]A[i]; //освобождение памяти

}

delete[]A; //освобождение памяти

cout << "\n";

}



**Варыянт 11**

|  |  |
| --- | --- |
| **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. |

#include <iostream>

#include <Windows.h>

int main()

{

using namespace std;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int \*A , n;

bool t = false; // создаём флажок для проверки наличия нуля в массиве

cout << "Введите размер массива: ";

cin >> n;

A = new int[n]; //выделение динам.памяти

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

cin >> \*(A + i);

}

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

if (\*(A + i) == 0) { // ищем элемент равный нулю

cout << "Элемент с индексом " << i << " равен 0" << endl; // выводим его индекс

t = true; // помечаем, что такой элемент есть

break;

}

}

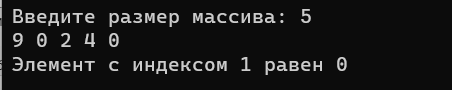
if (t == false) { // если не нашли элемент = 0, выводим сообщение об этом

cout << "В массиве нет элементов равных нулю" << endl;

}

delete[] A; // освобождаем память

}



#include <iostream>

#include <Windows.h>

int main()

{

using namespace std;

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n = 4, s = 0;

bool t = false;

int\*\* A = new int\* [n]; // выделение динамической памяти

cout << "введите массив: " << endl;//вводим элементы для массива

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

A[i] = new int[n]; // формируем массив массивов

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

cin >> A[i][j];

}

}

for (int i = 0; i < n; i++) { // ищем одинаковые элементы в строке и столбце с одним индексом

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

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

break;

}

else if (A[i][j] == A[j][i]) {

s += 1; // считаем количество таких элементов

}

}

if (s == n) { // если количество найденных элементов совпало с порядком матрицы

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

t = true;

}

s = 0;

}

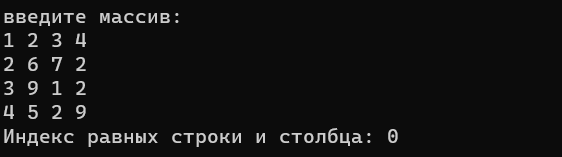
if (t == false) {

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

}

delete[] A; // освобождаем память

}



**Варыянт 12**

|  |  |
| --- | --- |
| **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. |

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, count = 0;

cout << "Введите размер массива: ";

cin >> n;

int\* a = new int[n]; // выделение динамической памяти

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

cin >> a[i]; // считываем массив

}

int max = a[0]; // присваиваем переменной max значение 1 элемента массива

for (int i = 0; i < n; i++) { // ищем максимальный элемент в массиве

if (a[i] > max) {

max = a[i];

}

}

for (int i = 0; i < n; i++) { // считаем сколько раз встречается максимальный элемент в массиве

if (a[i] == max) {

count += 1;

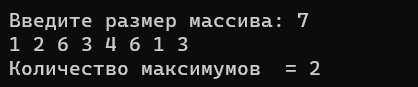
}

}

cout << "Количество максимумов = " << count << endl;

delete[] a; // освобождаем память

}



#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m, num = 0;

bool t = false;

cout << "Введите размер массива: ";

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

int\*\* a = new int\* [n]; // выделение динамической памяти под внешний массив

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

a[i] = new int[m];

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

cin >> a[i][j]; // считываем массив

}

}

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

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

if (a[i][j] > 0) { //ищем положительный элемент

t = true; // поднимаем флажок, что в массиве есть хотя бы 1 положительный элемент

num = i; // помечаем первую строку, в которой находится положительный элемент

cout << "В строке с индексом " << i << " есть хотя бы 1 положительное число" << endl; // выводим в какой строке он находится

break; // прерываем внутренний цикл

}

}

if (t == true) { // если мы уже нашли такой элемент, то и внешний цикл тоже прерываем

break;

}

}

if (t == false) { // если нет положительных элементов, то сообщаем об этом

cout << "Ни в одной из строк нет положительных элементов" << endl;

}

else { // если есть

if (num == 0) {

num = n - 1;

}

else {

num -= 1;

}

for (int i = 0; i < m; i++) { // меняем у предыдущей строки знаки(для 1 строки меняем у последней)

a[num][i] = -a[num][i];

}

cout << "Полученный массивЖ " << endl;

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

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

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

}

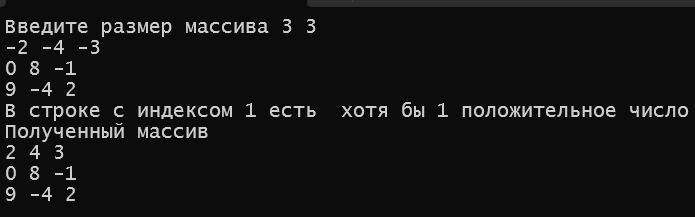
cout << endl;

}

}

delete[] a; // освобождаем память

}



**Варыянт 8**

|  |  |
| --- | --- |
| **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. |

#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int k, sum = INFINITE, kol = 0;

bool t = false; // флажок для пометки найденного нуля

cout << "Введите размер массива ";

cin >> k; // считываем размер массива

int \*a = new int[k]; // выделение динамической памяти

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

cin >> a[i];

}

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

if (a[i] > 0) { // считаем количество положительных элементов

kol += 1;

}

if (a[i] == 0 and t == false) { // помечаем, что ноль есть

sum = 0;

t = true;

}

else if (a[i] == 0 and t == true) { // обнуляем сумму, если нашли новый ноль

sum = 0;

}

if (t == true and a[i] != 0) { // считаем сумму элементов, расположенных после нуля

sum += a[i];

}

}

cout << "Количество положительных элементов = " << kol << endl; // выводим количество положительных элементов

if (sum == INFINITE) { // если в массиве нет нулей, то выводим сообщение об этом

cout << "В массиве нет элементов равных нулю" << endl;

}

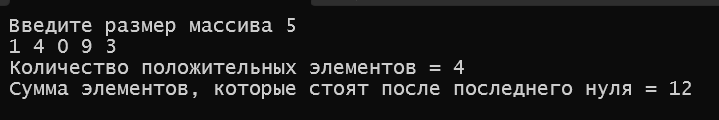
else { // иначе выводим сумму элементов, стоящих после нуля

cout << "Сумма элементов, которые стоят после последнего нуля = " << sum << endl;

}

delete[] a; // освобождаем память

}



#include <iostream>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int k, n;

bool t = false; // флажок для пометки найденного нуля

cout << "Введите размер массива ";

cin >> n >> k; // считываем размер массива

int \*\*a = new int\*[n]; // выделение динамической памяти под внешний массив

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

a[i] = new int[k]; // выделение динамической памяти под внутренний массив

for (int j = 0; j < k; j++) { // считываем массив

cin >> a[i][j];

}

}

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

for (int j = 0; j < n; j++) { // ищем нулевой элемент

if (a[j][i] == 0) { // если нашли

t = true; // подимаем флажок, что элемент найден

cout << "Наименьший индекс столбца, который содержит ноль, равен " << i << endl; // выводим индекс первого такого столбца

break; // прерываем внутренний цикл

}

}

if (t == true) { // если мы нашли такой столбец, то нам нет смысла проверять другие, поэтому прерываем внешний цикл

break;

}

}

if (t == false) { // если мы не нашли такого столбца, то выводим сообщение об этом

cout << "Нет ни одного столбца, который бы содержал 0" << endl;

}

delete[] a; // освобождаем память

}

