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:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Преподаватель: асс. Андронова М.В.

2023, Минск

|  |  |
| --- | --- |
| **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 <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

cout << "Введите размер массива ";

cin >> n; // считываем размер массива

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

int \*a = new int[n]; // создаём указатель на массиве размером n

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;

}

}

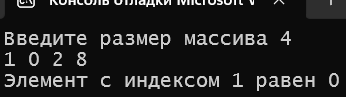
if (t == false) { // если не нашли элемент = 0, выводим сообщение об этом

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

}

delete[] a; // освобождаем память

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, s = 0;

cout << "Введите размер массива ";

cin >> n; // считываем размер массива

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

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

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;

}

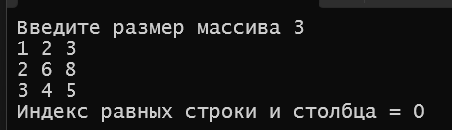
if (t == false) {

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

}

delete[] a; // освобождаем память

}



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

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, k = 0;

cout << "Введите размер массива ";

cin >> n; // считываем размер массива

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

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

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

cin >> a[i];

}

for (int i = 0; i < n; i++) { // ищем отрицательные элементы

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

k = i;

t = true; // обозначаем наличие хотя бы 1 отрицательного элемента

}

}

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

cout << "Отрицательных элементов нет " << endl;

}

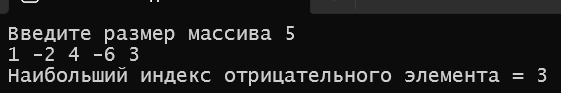
else { // иначе выводим индекс этого элемента

cout << "Наибольший индекс отрицательного элемента = " << k << endl;

}

delete[] a; // освобождаем память

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, k = 0, x = 0, y = 0;

cout << "Введите размер массива ";

cin >> n; // 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[n]; // выделение динамической памяти под внутренний массив

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

cin >> a[i][j];

}

}

int max = a[0][0];

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

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

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

max = a[i][j];

x = i; // сохраняем индекс строки

y = j; // сохраняем индекс столбца

}

}

}

if (x != 0 or y != 0) { // если он уже не находится на нужной позиции

for (int i = 0; i < n; i++) { // меняем строки местами

k = a[0][i];

a[0][i] = a[x][i];

a[x][i] = k;

}

for (int i = 0; i < n; i++) { // меняем столбцы местами

k = a[i][0];

a[i][0] = a[i][y];

a[i][y] = k;

}

}

cout << endl;

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

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

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

}

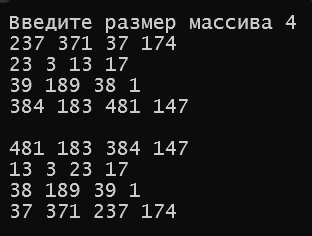
cout << endl;

}

delete[] a; // освобождаем память

}

|  |  |
| --- | --- |
| **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 <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.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;

}

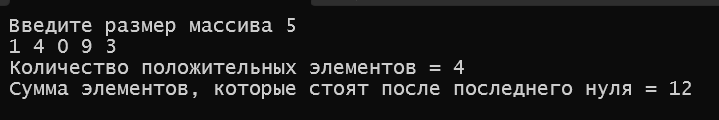
else { // иначе выводим сумму элементов, стоящих после нуля

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

}

delete[] a; // освобождаем память

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.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;

}

}

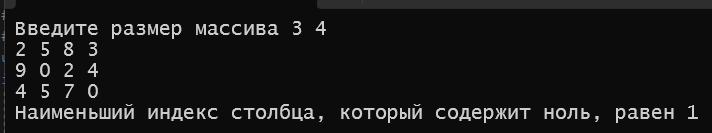
if (t == false) { // если мы не нашли такого столбца, то выводим сообщение об этом

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

}

delete[] a; // освобождаем память

}



|  |  |
| --- | --- |
| **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 <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, kol = 0;

cout << "Введите размер массива ";

cin >> n; // считываем размер массива

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) {

kol += 1;

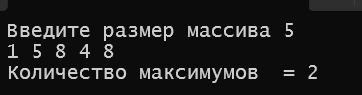
}

}

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

delete[] a; // освобождаем память

}



#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

#include <conio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, k, num = 0;

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++) {

a[i] = new int[k];

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

cin >> a[i][j]; // считываем массив

}

}

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

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

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

}

cout << "Полученный массив " << endl;

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

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

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

}

cout << endl;

}

}

delete[] a; // освобождаем память

}

