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»

Выполнил:

Студент 1 курса 7 группы

Ленкевич Павел Андреевич

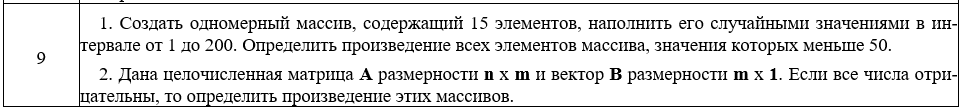
Преподаватель: асс. Андронова М.В.

2023, Минск

5. В соответствии со своим вариантом написать главную функцию, в которой имеются вызовы ***функций пользователя***, реализующих задачи из таблицы ниже. Ввод исходных данных и вывод результатов осуществить в главной функции, при этом использовать***динамические* *массивы***.

Для передачи параметров в функции пользователя и возвращения результатов применить ***указатели*** и ***ссылки***.

**Вариант 9.**

****

**Задача 1.**

#include <iostream> // Подключение библиотек

#include <ctime>

using namespace std; // Позволяет не писать std перед операторами ввода-вывода

void task1() {

int\* arr = new int[15]; // Создание динамического массива размером 15

srand(time(0));

int product = 1;

for (int i = 0; i < 15; i++) {

arr[i] = rand() % 200 + 1; // Заполнение массива случайными числами от 1 до 200

if (arr[i] < 50) { // Если элемент массива меньше 50, умножаем его на product

product \*= arr[i];

}

}

cout << "Произведение элементов массива, значения которых меньше 50: " << product << endl; // Вывод произведения элементов массива, значения которых меньше 50

delete[] arr;// Освобождение памяти

}

int main() {

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

task1(); // Вызов функции для решения задачи 1

return 0;

}

****

**Задача 2.**

#include <iostream> // Подключение библиотек

using namespace std; // Позволяет не писать std перед операторами ввода-вывода

void task2(int\*\* A, int\* B, int n, int m) {

int\* result = new int[n]; // Создание динамического массива для хранения результата

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

result[i] = 0;

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

if (A[i][j] < 0 && B[j] < 0) { // Если элементы матрицы A и вектора B отрицательны, умножаем их и добавляем к result[i]

result[i] += A[i][j] \* B[j];

}

}

}

cout << "Произведение матрицы A и вектора B: "; // Вывод результата

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

cout << result[i] << " ";

}

cout << endl;

delete[] result; // Освобождение памяти

}

int main() {

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

int n = 3, m = 3;

int\*\* A = new int\* [n]; // Создание и заполнение матрицы A

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

A[i] = new int[m];

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

A[i][j] = -5; // Заполните матрицу A

}

}

int\* B = new int[m]; // Создание и заполнение вектора B

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

B[i] = -3; // Заполните вектор B

}

task2(A, B, n, m); // Вызов функции для решения задачи 2

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

delete[] A[i];

}

delete[] A;

delete[] B;

return 0;

}

****

**Объединение:**

#include <iostream> // Подключение библиотек

#include <ctime>

using namespace std; // Позволяет не писать std перед операторами ввода-вывода

// Задача 1

void task1() {

int\* arr = new int[15]; // Создание динамического массива размером 15

srand(time(0));

int product = 1;

for (int i = 0; i < 15; i++) {

arr[i] = rand() % 200 + 1; // Заполнение массива случайными числами от 1 до 200

if (arr[i] < 50) { // Если элемент массива меньше 50, умножаем его на product

product \*= arr[i];

}

}

cout << "Произведение элементов массива, значения которых меньше 50: " << product << endl; // Вывод произведения элементов массива, значения которых меньше 50

delete[] arr;// Освобождение памяти

}

// Задача 2

void task2(int\*\* A, int\* B, int n, int m) {

int\* result = new int[n]; // Создание динамического массива для хранения результата

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

result[i] = 0;

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

if (A[i][j] < 0 && B[j] < 0) { // Если элементы матрицы A и вектора B отрицательны, умножаем их и добавляем к result[i]

result[i] += A[i][j] \* B[j];

}

}

}

cout << "Произведение матрицы A и вектора B: "; // Вывод результата

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

cout << result[i] << " ";

}

cout << endl;

delete[] result; // Освобождение памяти

}

int main() {

setlocale(LC\_CTYPE, "Russian"); // Подключает русский язык

task1(); // Вызов функции для решения задачи 1

int n = 3, m = 3;

int\*\* A = new int\* [n]; // Создание и заполнение матрицы A

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

A[i] = new int[m];

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

A[i][j] = -5; // Заполните матрицу A

}

}

int\* B = new int[m]; // Создание и заполнение вектора B

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

B[i] = -3; // Заполните вектор B

}

task2(A, B, n, m); // Вызов функции для решения задачи 2

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

delete[] A[i];

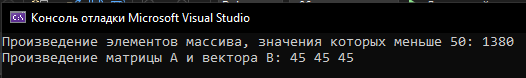
}

delete[] A;

delete[] B;

return 0;

}

****

5 вариант

1. В одномерном массиве, содержащем 15 элементов, посчитать и вывести сумму всех четных элементов массива.

Исходный код:

#include <iostream>

using namespace std;

int is\_elem(int\* , int , int);

void main()

{

setlocale(LC\_CTYPE, "Rus");

int t, even = 0;

int A[15];

cout << "Ведите массив из 15 символов\n";

for (int i = 0; i < 15; i++)

cin >> \*(A + i);

t = is\_elem(A, sizeof(A) / sizeof(int), even); // вызов функции

cout << "Сумма чётных элементов массива ==  " << t;

}

int is\_elem(int\* pA, int n, int summ) // вызываемая функция

{

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

if (pA[i] % 2 == 0 )

{

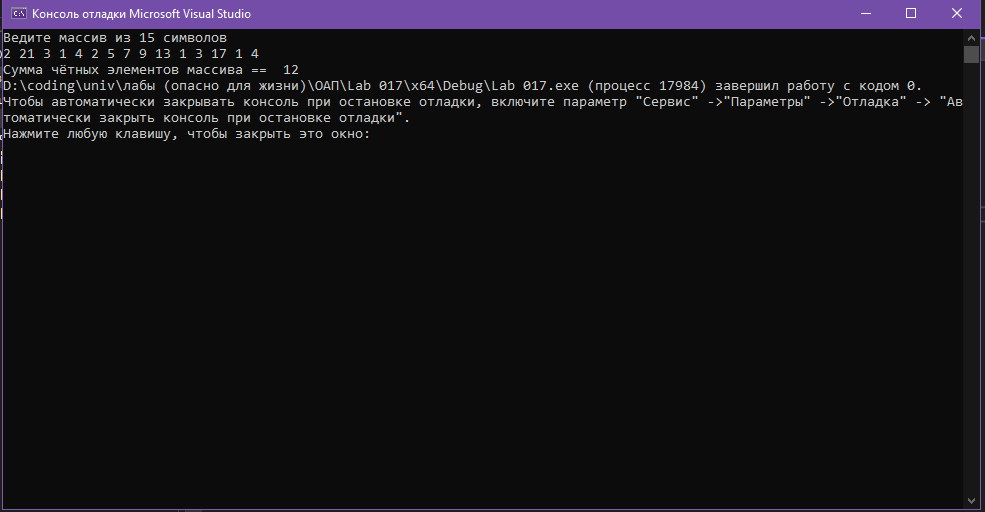
summ += pA[i];

}

return summ; // возвращаем значение суммы

}

Пример вывода консоли:



2. Даны две целочисленные матрицы размером **n** x **m** и **m** x **n**. Если в матрице отсутствуют нулевые элементы, то определить произведение этих матриц.

Исходный код:

#include <iostream>

using namespace std;

void sum(int\*\*, int\*\*, int, int);

int main()

{

    setlocale(LC\_CTYPE, "Russian");

    int m, n;

    cout << "введите m\n"; cin >> m;

    cout << "введите n\n"; cin >> n;

    int i, j;

    int\*\* matr\_1 = new int\* [n]; // выделяем память для матриц

    for (i = 0; i < n; i++)

        matr\_1[i] = new int[m];

    int\*\* matr\_2 = new int\* [m];

    for (i = 0; i < m; i++)

        matr\_2[i] = new int[n];

    cout << "Введите первую матрицу\n";

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < m; j++)

        {

            cin >> matr\_1[i][j];

        }

    }

    cout << "Введите вторую матрицу\n";

    for (i = 0; i < m; i++)

    {

        for (j = 0; j < n; j++)

        {

            cin >> matr\_2[i][j];

        }

    }

    cout << "Произведение матриц:\n";

    sum(matr\_1, matr\_2, n, m); // вызываем функцию

    for (int i = 0; i < n; i++) // очищаем память

        delete matr\_1[i];

    delete[] matr\_1;

    for (int i = 0; i < m; i++)

        delete matr\_2[i];

    delete[] matr\_2;

}

void sum(int\*\* x, int \*\* y, int n, int m)

{

    int res ;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) { //для каждого элемента матрицы n x n

            res = 0;

            for (int k = 0; k < m; k++) { // находим попарные суммы элементов строки и столбца

                res += x[i][k] \* y[k][j];

            }

            cout << res << "\t";

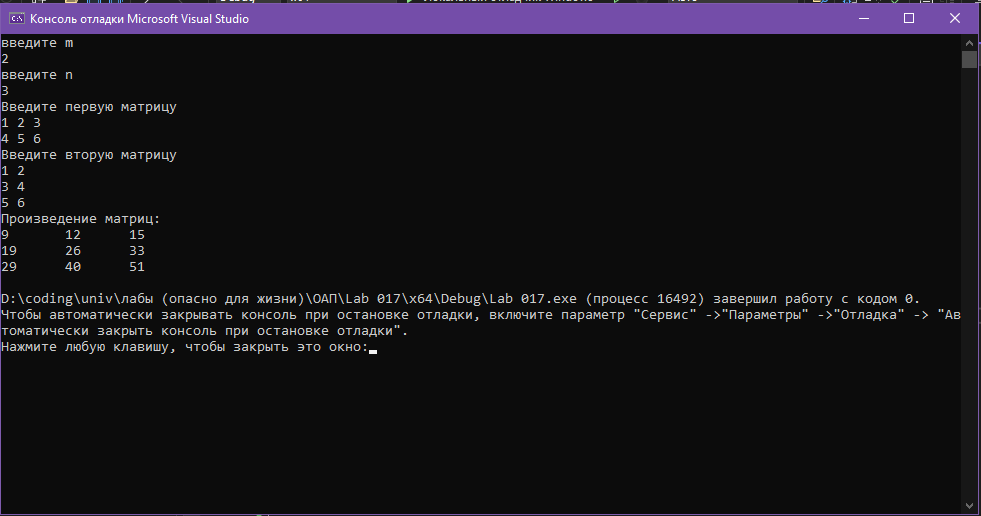
        }

        cout << endl;

    }

}

Пример вывода консоли:



5 вариант

1. В одномерном массиве, содержащем 15 элементов, посчитать и вывести сумму всех четных элементов массива.

Исходный код:

#include <iostream>

using namespace std;

int is\_elem(int\* , int , int);

void main()

{

setlocale(LC\_CTYPE, "Rus");

int t, even = 0;

int A[15];

cout << "Ведите массив из 15 символов\n";

for (int i = 0; i < 15; i++)

cin >> \*(A + i);

t = is\_elem(A, sizeof(A) / sizeof(int), even); // вызов функции

cout << "Сумма чётных элементов массива ==  " << t;

}

int is\_elem(int\* pA, int n, int summ) // вызываемая функция

{

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

if (pA[i] % 2 == 0 )

{

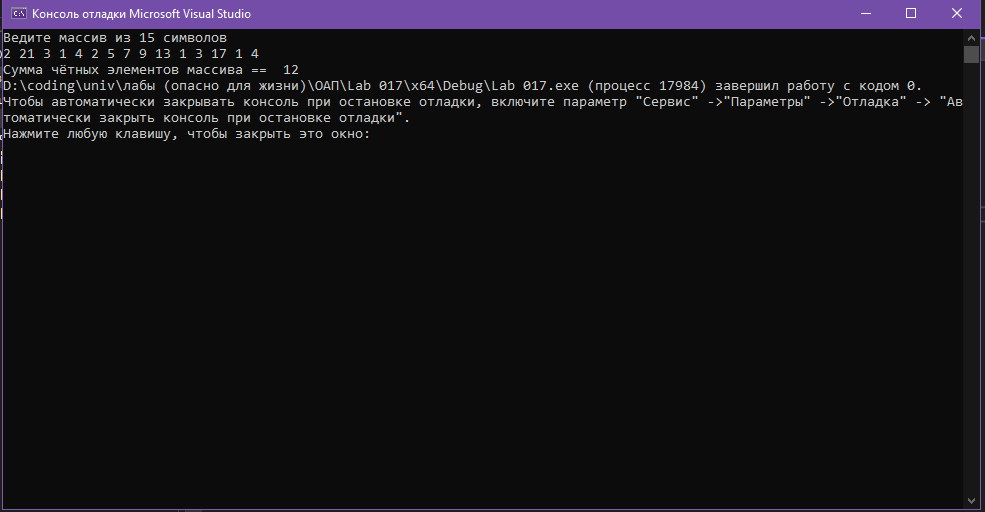
summ += pA[i];

}

return summ; // возвращаем значение суммы

}

Пример вывода консоли:



2. Даны две целочисленные матрицы размером **n** x **m** и **m** x **n**. Если в матрице отсутствуют нулевые элементы, то определить произведение этих матриц.

Исходный код:

#include <iostream>

using namespace std;

void sum(int\*\*, int\*\*, int, int);

int main()

{

    setlocale(LC\_CTYPE, "Russian");

    int m, n;

    cout << "введите m\n"; cin >> m;

    cout << "введите n\n"; cin >> n;

    int i, j;

    int\*\* matr\_1 = new int\* [n]; // выделяем память для матриц

    for (i = 0; i < n; i++)

        matr\_1[i] = new int[m];

    int\*\* matr\_2 = new int\* [m];

    for (i = 0; i < m; i++)

        matr\_2[i] = new int[n];

    cout << "Введите первую матрицу\n";

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < m; j++)

        {

            cin >> matr\_1[i][j];

        }

    }

    cout << "Введите вторую матрицу\n";

    for (i = 0; i < m; i++)

    {

        for (j = 0; j < n; j++)

        {

            cin >> matr\_2[i][j];

        }

    }

    cout << "Произведение матриц:\n";

    sum(matr\_1, matr\_2, n, m); // вызываем функцию

    for (int i = 0; i < n; i++) // очищаем память

        delete matr\_1[i];

    delete[] matr\_1;

    for (int i = 0; i < m; i++)

        delete matr\_2[i];

    delete[] matr\_2;

}

void sum(int\*\* x, int \*\* y, int n, int m)

{

    int res ;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) { //для каждого элемента матрицы n x n

            res = 0;

            for (int k = 0; k < m; k++) { // находим попарные суммы элементов строки и столбца

                res += x[i][k] \* y[k][j];

            }

            cout << res << "\t";

        }

        cout << endl;

    }

}

Пример вывода консоли:

