Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 16

По дисциплине “Основы алгоритмизации и программирования”

На тему “**Функции пользователя** ”

Выполнил:

Студент 1 курса 6 группы

Кравченко Сергей Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 8

1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов. Найти количество отрицательных, положительных и нулевых элементов массива.

2. Написать программу, которая вводит несколько слов текста и определяет общее количество слов. Слова разделены пробелами

1.

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

// Функция для заполнения двумерного массива случайными числами

void Random(int\*\* mass, int m, int n)

{

srand((unsigned)time(NULL));

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

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

mass[i][j] = rand() % 21 + (-10);

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

}

cout << endl;

}

}

int main() {

setlocale(LC\_CTYPE, "Russian");

int m, n;

cout << "Введите количество строк: ";

cin >> m;

cout << "Введите количество столбцов: ";

cin >> n;

int count = 0;

int negative = 0, positive = 0, null = 0;

int\*\* mass = new int\* [m]; // Выделяем память под указатели на строки двумерного массива

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

if (!(mass[i] = new int[n])) { // Выделяем память под каждую строку двумерного массива

cout << "Недостаточно памяти" << endl;

return 1;

}

}

Random(mass, m, n); // Вызываем функцию для заполнения массива случайными числами

// Подсчет количества положительных, отрицательных и нулевых элементов в массиве

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

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

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

positive++;

}

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

negative++;

}

if (mass[i][j] == 0) {

null++;

}

}

}

// Вывод результатов подсчета на экран

cout << "Количество положительных элементов: " << positive << endl;

cout << "Количество отрицательных элементов: " << negative << endl;

cout << "Количество элементов, равных нулю: " << null << endl;

// Освобождение памяти, выделенной под массив

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

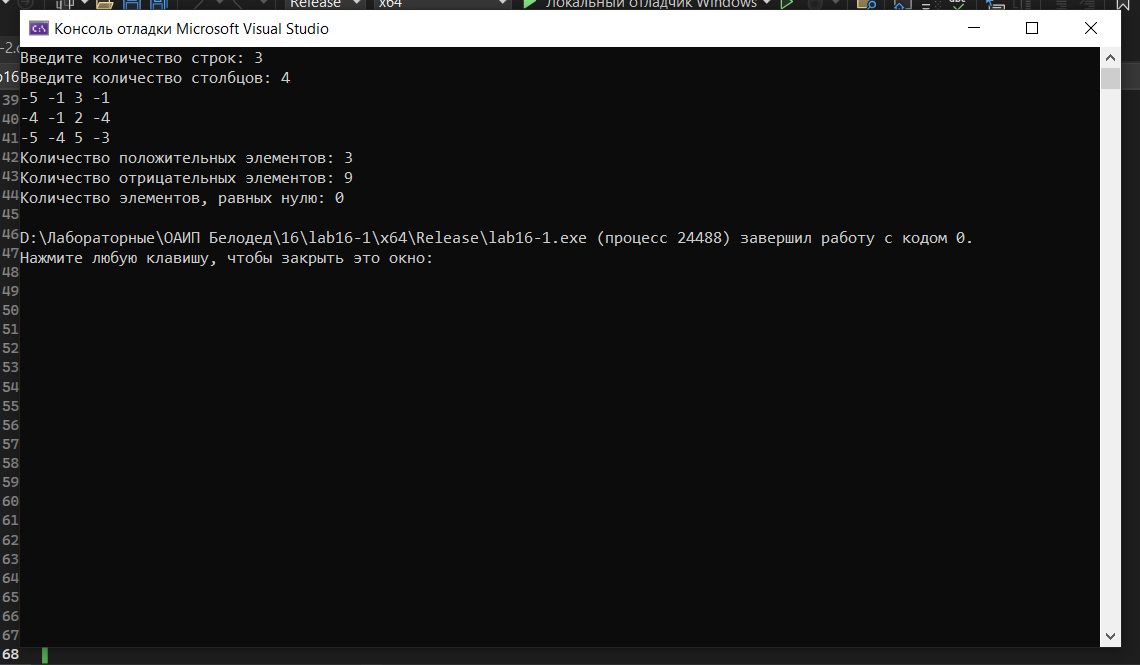
delete[] mass[i];

}

delete[] mass;

return 0;

}



2.

#include <iostream>

#include <string>

using namespace std;

// Функция для подсчета количества слов в тексте

int CountWords(const string& text) {

int wordCount = 0;

bool inside = false; // Для проверки находимся ли мы внутри слова

for (char c : text) {

if (c == ' ') {

inside = false;

}

else if (!inside) {

inside = true;

wordCount++;

}

}

return wordCount;

}

// Функция main

int main() {

setlocale(LC\_ALL, "RUS");

string text;

cout << "Введите текст: ";

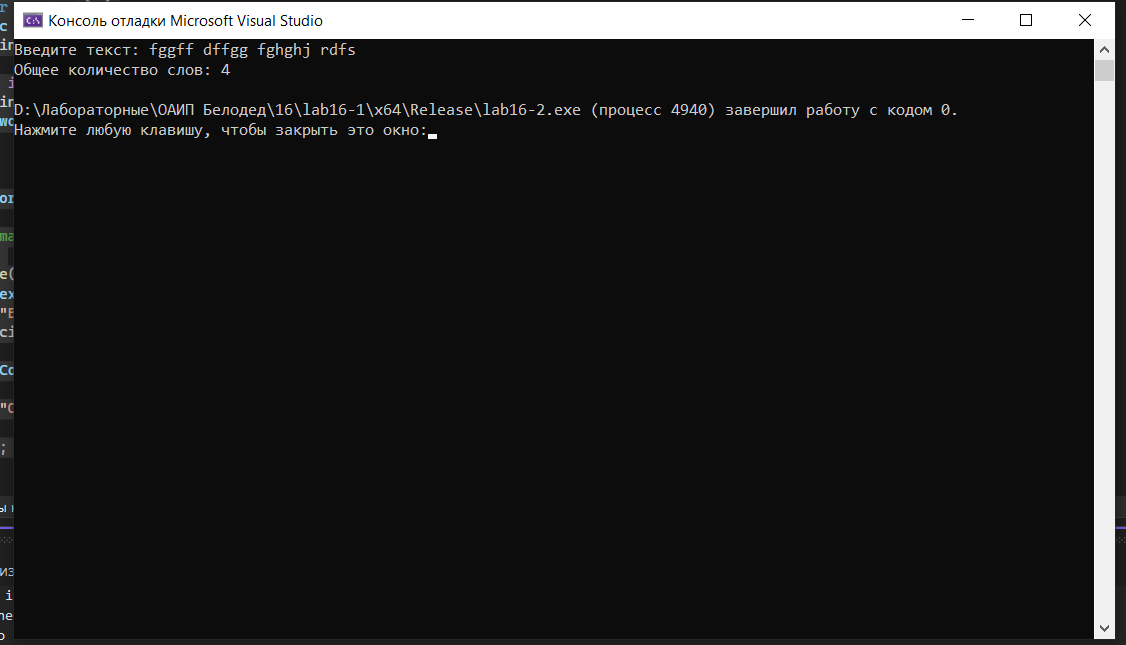
getline(cin, text); // Получаем длину строки

int wordCount = CountWords(text); // Вызов функции для подсчета количества слов

cout << "Общее количество слов: " << wordCount << endl;

return 0;

}



3. Объединение с использованием switch case

#include <iostream>

#include <Windows.h>

#include <string>

#include <ctime>

using namespace std;

void Random(int\*\* mass, int m, int n)

{

srand((unsigned)time(NULL));

for (int i = 0; i < m; i++) { // Заполняем массив рандомными числами

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

mass[i][j] = rand() % 21 + (-10);

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

}

cout << endl;

}

}

int CountWords(const string& text) {

int wordCount = 0;

bool inside = false; // Для проверки находимся ли мы внутри слова

for (char c : text) {

if (c == ' ') {

inside = false;

}

else if (!inside) {

inside = true;

wordCount++;

}

}

return wordCount;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choice;

do {

cout << "Выберите задачу" << endl;

cout << "1 - С матрицей " << endl;

cout << "2 - Со строкой " << endl;

cout << "3 - выход" << endl;

cin >> choice;

cin.ignore();// считываем выбор пользователя

switch (choice) { // создаём оператор выбора между задачами и опцию выхода

case 1: {

setlocale(LC\_CTYPE, "Russian");

int m, n;

cout << "Введите количество строк: ";

cin >> m;

cout << "Введите количество столбцов: ";

cin >> n;

int count = 0;

int negative = 0, positive = 0, null = 0;

int\*\* mass = new int\* [m]; // Выделяем память под указатели на строки двумерного массива

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

if (!(mass[i] = new int[n])) { // Выделяем память под каждую строку двумерного массива

cout << "Недостаточно памяти" << endl;

return 1;

}

}

Random(mass, m, n); // Вызываем функцию для заполнения массива случайными числами

// Подсчет количества положительных, отрицательных и нулевых элементов в массиве

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

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

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

positive++;

}

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

negative++;

}

if (mass[i][j] == 0) {

null++;

}

}

}

// Вывод результатов подсчета на экран

cout << "Количество положительных элементов: " << positive << endl;

cout << "Количество отрицательных элементов: " << negative << endl;

cout << "Количество элементов, равных нулю: " << null << endl;

// Освобождение памяти, выделенной под массив

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

delete[] mass[i];

}

delete[] mass;

break;

}

case 2: {

setlocale(LC\_ALL, "RUS");

string text;

cout << "Введите текст: ";

getline(cin, text); // Получаем длину строки

int wordCount = CountWords(text); // Вызов функции для подсчета количества слов

cout << "Общее количество слов: " << wordCount << endl;

break;

}

case 3: { // если пользователь выбрал выход, то прерываем программу

break;

}

}

} while (choice != 3);

}



Доп варианты

|  |  |
| --- | --- |
| **2** | 1. Ввести вещественные числа в массив **А**[**n**, **m**] (**n**, **m** > 0) и ко всем числам каждой строки прибавить единицу. Если первый элемент строки нулевой, ничего не делать.  **2.** Дана строка. Все русские буквы привести к верхнему регистру, латинские − заменить символом «**?**». Вывести результат на экран. |

Объединение с использованием switch case

#include <iostream>

#include <Windows.h>

using namespace std;

double\*\* changes(double\*\* a, int n, int m) { // функция для добавления 1 к элементам матрицы

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

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

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

continue;

}

else {

a[i][j] += 1;

}

}

}

return a;

}

void conclusion(double\*\* a, int n, int m) { // функция вывода матрицы

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

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

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

}

cout << endl;

}

}

char\* change\_simbol(char\* a, int n) { // функция замены символов в строке

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

if (a[i] >= 'а' and a[i] <= 'я') {

a[i] = a[i] - 'а' + 'А';

}

else if ((a[i] >= 'A' and a[i] <= 'Z') or (a[i] >= 'a' and a[i] <= 'z')) {

a[i] = '?';

}

}

return a;

}

int main() { // функция main

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choice;

do {

cout << "Выберите задачу" << endl;

cout << "1 - Добавление 1 к элементу в массивве " << endl;

cout << "2 - Преобразование строки " << endl;

cout << "3 - Выход" << endl;

cin >> choice; // считываем выбор пользователя

switch (choice) { // создаём оператор выбора между задачами и опцию выхода

case 1: {

int m, n;

cout << "Введите количество строк: ";

cin >> n;

cout << "Введите количество столбцов: ";

cin >> m;

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

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];

}

}

a = changes(a, n, m);

conclusion(a, n, m);

delete[] a; // освобождаем память

break;

}

case 2: {

char\* a; // создаём указатель на массив символов

cout << "Введите размер строки ";

int n;

cin >> n; // считываем размер строки

a = new char[n + 1];

cout << "Введите строку" << endl;

cin >> a; // считываем строку

change\_simbol(a, n); // выполняем замену символов в соответствии с заданием

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

cout << a[i];

}

cout << endl;

delete[] a; // освобождаем память

break;

}

case 3: {

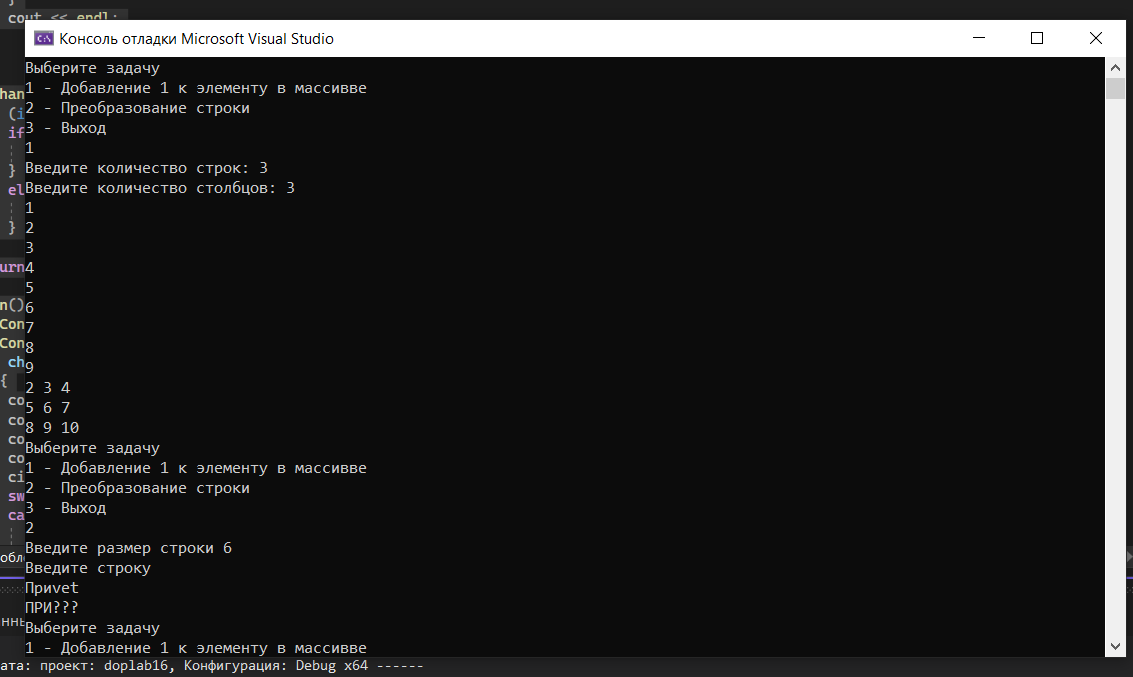
break;

}

}

} while (choice != 3);

}



|  |  |
| --- | --- |
| **4** | 1. Ввести целочисленный массив **A**[**n**, **n**] и вычесть из каждой строки предыдущую. Из первой строки вычесть последнюю строку. Получившийся массив вывести.  2. Дана строка. Слова в предложении разделены одним или несколькими пробелами. Слова могут состоять только из цифр или букв. Найти сумму чисел, входящих в строку. |

Объединение с использованием switch case

#include <iostream>

#include <cstdlib>

using namespace std;

// Функция для вычитания предыдущего элемента из каждого элемента в массиве

void stroki(int\*\* A, int n, int m) {

// Перебор каждого элемента в массиве

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

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

// Вычитание предыдущего элемента из текущего

A[i][j] -= A[i - 1][j];

}

}

// Вычитание последнего элемента из первого

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

A[0][j] -= A[n - 1][j];

}

}

// Функция для суммирования чисел в строке

int my\_sum(const char\* src) {

int sum = 0;

// Перебор каждого символа в строке

while (\*src) {

// Пропуск символов, не являющихся числами или минусом

while (\*src && !((\*src >= '0' && \*src <= '9') || \*src == '-'))

++src;

// Если текущий символ является числом, добавляем его к сумме

if (\*src >= '0' && \*src <= '9') {

sum += atoi(src);

}

// Пропуск чисел

while (\*src && ((\*src >= '0' && \*src <= '9') || \*src == '-'))

++src;

}

return sum;

}

int main() {

setlocale(LC\_ALL, "RUS");

int c;

do {

cout << endl;

cout << "Введите" << endl;

cout << "1 - 1 задание" << endl;

cout << "2 - 2 задание" << endl;

cout << "3 - выход" << endl;

cin >> c;

switch (c) {

case 1: {

int n, m;

cout << "Введите количество строк: ";

cin >> n;

cout << "Введите количество столбцов: ";

cin >> m;

// Создание двумерного массива

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

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

A[i] = new int[m];

}

cout << "Введите элементы массива A:\n";

// Заполнение массива

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

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

cin >> A[i][j];

}

}

cout << "Ваш массив: " << endl;

// Вывод массива

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

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

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

}

cout << endl;

}

// Применение функции stroki к массиву

stroki(A, n, m);

cout << "Получившийся массив:\n";

// Вывод получившегося массива

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

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

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

}

cout << endl;

}

// Освобождение памяти, выделенной под массив

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

delete[] A[i];

}

delete[] A;

break;

}

case 2: {

cout << "Введите строку: ";

cin.ignore();

char str[1000];

cin.getline(str, sizeof(str));

// Применение функции my\_sum к строке

int sum = my\_sum(str);

cout << "Сумма чисел в строке: " << sum << endl;

break;

}

case 3:

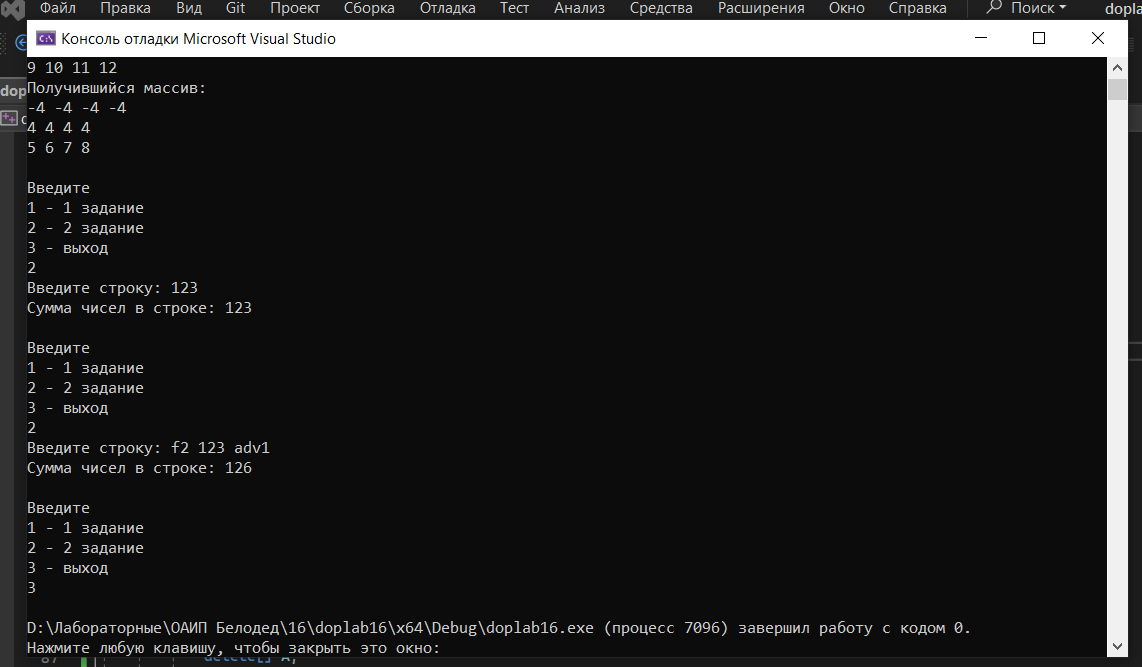
break;

}

} while (c != 3);

return 0;

}



|  |  |
| --- | --- |
| **3** | 1. Найти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве **A**[**n**, **n**].  2. Написать программу, которая вводит несколько строк текста и символ «**m**» и использует функцию, чтобы определить суммарное число вхождений символа в текст. |

Объединение с использованием switch case

#include <iostream>

#include <Windows.h>

using namespace std;

int summ(int\*\* a, int n, int& sum) { // функция для подсчёта суммы элементов матрицы, которые стоят ниже главной диагонали

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

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

sum += a[i][j];

}

}

return sum;

}

int coutstr(char\*\* a, int n, char b, int& amount) { // функция ввода строк и подсчёта количества вхождений введённого символа

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

int x;

cout << "Введите длину строки ";

cin >> x;

a[i] = new char[x];

cin >> a[i];

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

if (a[i][j] == b) {

amount += 1;

}

}

}

return amount;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choice;

do {

cout << "Выберите задачу: " << endl;

cout << "1 - Нахождение суммы элементов " << endl;

cout << "2 - Подсчёт вхождений символа " << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice) {

case 1: {

int n, sum = 0;

cout << "Введите размер матрицы ";

cin >> n;

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

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];

}

}

sum = summ(a, n, sum); // выполняем функцию

cout << "Сумма элементов, которые стоят ниже главной диагонали = " << sum << endl;

delete[] a; // освобождаем память

break;

}

case 2: {

char\*\* a; // создаём указатель на массив массивов символов

char b;

cout << "Введите количество строк ";

int n, kol = 0;

cin >> n;

a = new char\* [n + 1];

cout << "Введите символ ";

cin >> b;

coutstr(a, n, b, kol); // вызываем функцию

cout << "Количество вхождение символа " << b << " равно " << kol << endl;

delete[] a; // освобождаем память

break;

}

case 3: { // если пользователь выбрал выход, то прерываем программу

break;

}

}

} while (choice != 3);

}

