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

Романов Игорь Вячеславович

Преподаватель: асс. Андронова М.В.

2023, Минск

Содержание

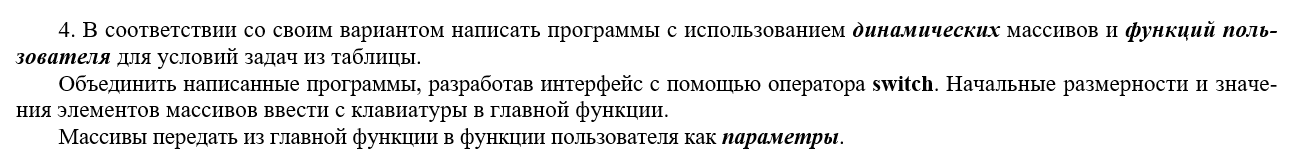
[Вариант 12 (основной) 3](#_Toc153140918)

[Вариант 7 7](#_Toc153140919)

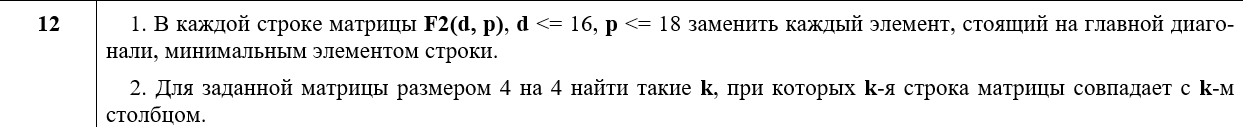
[Вариант 1 10](#_Toc153140920)

[Вариант 2 13](#_Toc153140921)

Задание 4



# Вариант 12 (основной)



#include <iostream>

using namespace std;

//функция для замены элементов в матрице

void replaceElements(int\*\* F2, int d, int p)

{

int i\_min = 99999, i ,j; //ввод счетчика минимального элемента строки

//определение минимального элемента в матрице

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

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

if (F2[i][j] < i\_min){

i\_min = F2[i][j];

}

}

}

//замена элементов по главной диагонали минимальным элементов

j = 0;

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

F2[i][j] = i\_min;

j++;

}

//вывод преобразованной матрицы

cout << "Преобразованная матрица: " << endl;

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

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

cout << F2[i][j] << "\t";

}

cout << endl;

}

}

//фукнция для нахождения одинаковых строк и столбцов

void findCoincidences(int\*\* A, int k, int rows, int cols)

{

int check, i, j; //ввод переменной, определяющий совпадение элементов строк; переменных итератора

//ввод числа k

cout << "Введите k от 0 до 3: "; cin >> k;

//если число не входит в диапазон, то пользователь вводит число еще раз

while (k < 0 || k > 3) {

cout << "Число не входит в заданный диапазон, введите число еще раз"; cin >> k;

}

//проверка совпадения

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

check = 0;

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

if (A[i][j] == A[j][i]) { //если находятся совпадения, то увеличить счетчик

check++;

}

}

if (check == 4) { //если строка и столбец совпали, то вывести соответствующий номер

k = i;

cout << k << "-ая строка матрицы совпадает с " << k << "-ым столбцом матрицы" << endl;

}

}

}

//главная функция

int main()

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do { //выбор вариантов работы программы

cout << "Выберите вариант работы" << endl;

cout << "1 - заменить элементы главной диагонали минимальным элементом матрицы" << endl;

cout << "2 - найти в матрице (4x4) совпадения k-ой строки и с k-ым столбцом" << endl;

cout << "3 - выход" << endl;

cin >> choice;

switch (choice)

{

case 1: {

int d, p, i, j; //ввод размеров матрицы, переменных итератора, числа k

cout << "Введите размер строки d(d <= 16): "; cin >> d; //ввод размера строк матрицы

while (d < 0 || d > 16) {

cout << "Число не входит в заданный диапазон(0, 16], введите число еще раз: "; cin >> d;

}

cout << "Введите размер строки d(d <= 16): "; cin >> p; //ввод размера столбцов матрицы

while (d < 0 || p > 18) {

cout << "Число не входит в заданный диапазон (0, 18], введите число еще раз: "; cin >> p;

}

//выделение динамической памяти для матрицы

int\*\* F2;

F2 = new int\* [d];

for (i = 0; i < d; i++)

F2[i] = new int[p];

//ввод элементов матрицы с клавиатуры

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

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

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

cout << "Элемент [" << i << "," << j << "]: ";

cin >> F2[i][j];

}

}

//вывод полученной матрицы на экран

cout << "Полученная матрица: " << endl;

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

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

cout << F2[i][j] << "\t";

}

cout << endl;

}

//переход к функции

replaceElements(F2, d, p);

//освобождение динамической памяти

for (i = 0; i < d; i++)

delete F2[i];

delete[] F2;

}

break;

case 2: {

int rows = 4, cols = 4, i, j, k = 0; //ввод размеров матрицы, переменных итератора, числа k

//выделение динамической памяти для матрицы

int\*\* A;

A = new int\* [rows];

for (i = 0; i < rows; i++)

A[i] = new int[cols];

//ввод элементов матрицы с клавиатуры

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

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

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

cout << "Элемент [" << i << "," << j << "]: ";

cin >> A[i][j];

}

}

//вывод полученной матрицы на экран

cout << "Полученная матрица: " << endl;

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

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

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

}

cout << endl;

}

//переход к функции

findCoincidences(A, k, rows, cols);

//освобождение динамической памяти

for (i = 0; i < rows; i++)

delete A[i];

delete[] A;

}

break;

case 3: { //выход из программы

break;

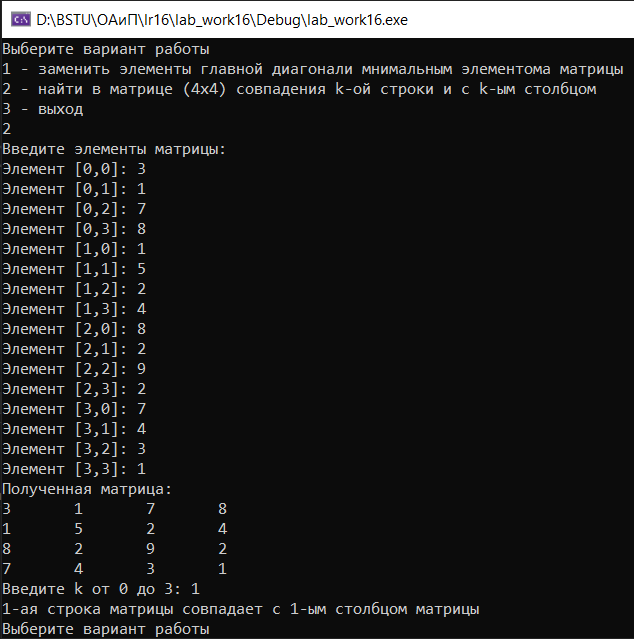
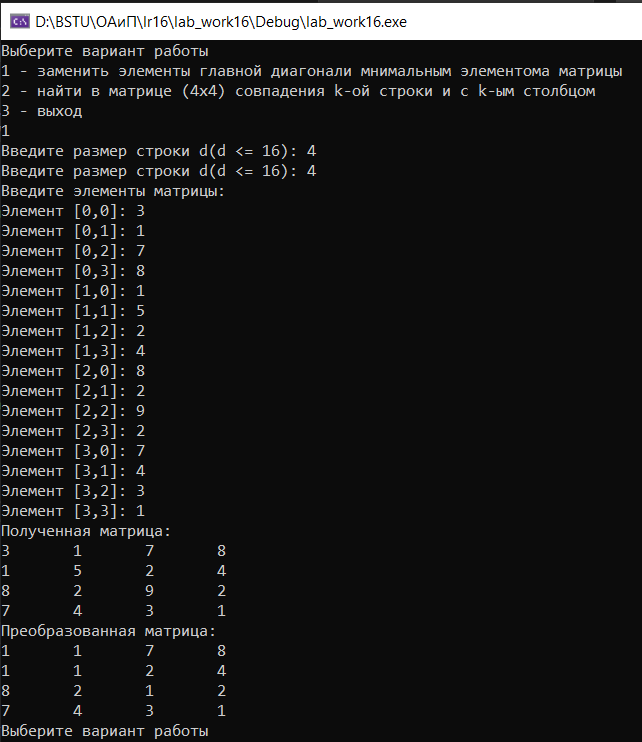
}

}

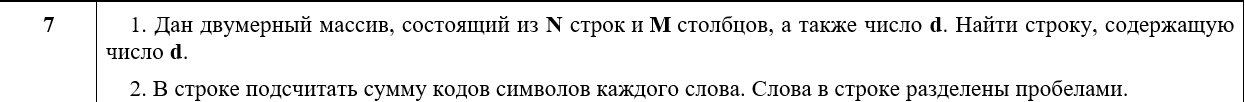
} while (choice != 3);

}

Результат



# Вариант 7



#include <iostream>

using namespace std;

//функция для замены элементов в матрице

void findElement(int\*\* array, int N, int M)

{

int d, i, j; //ввод переменной, вводимой с клавиатуры; переменных итератора

//ввод d с клавиатуры

cout << "Введите число d: "; cin >> d;

//поиск числа d в матрице

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

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

if (array[i][j] == d) {

cout << "Цифра встречена в строке под номером " << i << endl;

}

}

}

}

//фукнция для расчета суммы кодов символов в строке

void countCode(char\* str, int size) {

int sum = 0, i;

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

if (str[i] != ' ') { //пропускать пробел

sum += static\_cast<int>(str[i]); //вычисление суммы кодов символов

}

}

cout << "Сумма кодов символов в строке: " << sum << endl;

}

//главная функция

int main()

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do { //выбор вариантов работы программы

cout << "Выберите вариант работы" << endl;

cout << "1 - найти строку, содержащую число d в матрице MxN" << endl;

cout << "2 - посчитать сумму кодов символов в строке" << endl;

cout << "3 - выход" << endl;

cin >> choice;

switch (choice)

{

case 1: {

int N, M, i, j; //ввод размеров матрицы, переменных итератора, числа k

cout << "Введите размер строк N: "; cin >> N; //ввод размера строк матрицы

cout << "Введите размер столбцов M: "; cin >> M; //ввод размера столбцов матрицы

//выделение динамической памяти для матрицы

int\*\* array;

array = new int\* [N];

for (i = 0; i < N; i++)

array[i] = new int[M];

//ввод элементов матрицы с клавиатуры

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

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

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

cout << "Элемент [" << i << "," << j << "]: ";

cin >> array[i][j];

}

}

//вывод полученной матрицы на экран

cout << "Полученная матрица: " << endl;

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

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

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

}

cout << endl;

}

//переход к функции

findElement(array, N, M);

//освобождение динамической памяти

for (i = 0; i < N; i++)

delete array[i];

delete[] array;

}

break;

case 2: {

int size = 0; //ввод размера строки

cout << "Введите размер строки: "; cin >> size;

char\* str = new char[size]; //выделение памяти для строки

cout << "Введите строку: "; //ввод строки с клавиатуры

cin >> str;

countCode(str, size); //переход к функции

delete[] str; //освобождение памяти

}

break;

case 3: { //выход из программы

break;

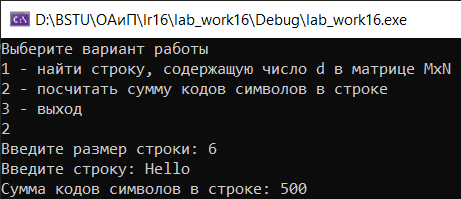
}

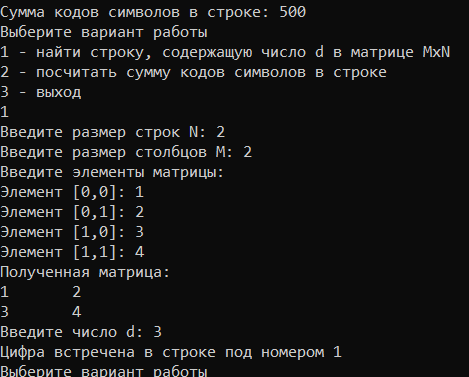
}

} while (choice != 3);

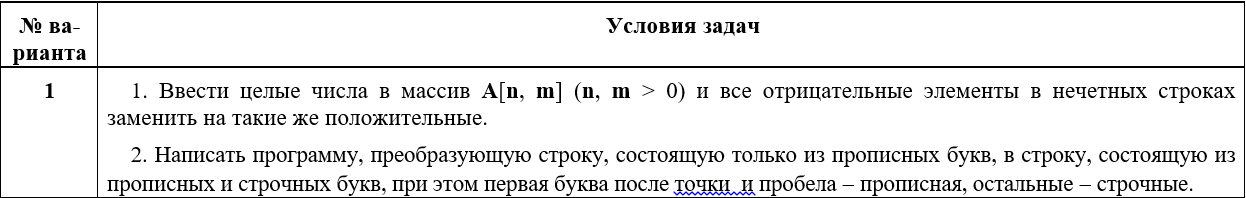
}

Результат





# Вариант 1



#include <iostream>

using namespace std;

//функция для замены отрицательных элементов в матрице

void invertNegative(int\*\* A, int N, int M)

{

int i, j; //ввод переменных итератора

//замена отрицательных элементов, стоящих в нечетной строке

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

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

if (i % 2 != 0 && A[i][j] < 0) {

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

}

}

}

//вывод преобразованной матрицы на экран

cout << "Преобразованная матрица: " << endl;

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

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

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

}

cout << endl;

}

}

//фукнция для преобразования букв строки

void transformString(char\* str, int size) {

int i;

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

if (str[i] == '.' || str[i] == ' ') { //если встречается точка или пробел

str[i + 1] = toupper(str[i + 1]); // то преобразовать букву после пробела или точки

}

}

cout << "Преобразованная строка: \n" << str << endl; //вывод преобразованной строки

}

//главная функция

int main()

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do { //выбор вариантов работы программы

cout << "Выберите вариант работы" << endl;

cout << "1 - заменить отрицательные числа в нечетных строках на обратные им" << endl;

cout << "2 - перевести первые буквы слов в верхний регистр" << endl;

cout << "3 - выход" << endl;

cin >> choice;

switch (choice)

{

case 1: {

int N, M, i, j; //ввод размеров матрицы, переменных итератора, числа k

cout << "Введите размер строк N: "; cin >> N; //ввод размера строк матрицы

cout << "Введите размер столбцов M: "; cin >> M; //ввод размера столбцов матрицы

//выделение динамической памяти для матрицы

int\*\* A;

A = new int\* [N];

for (i = 0; i < N; i++)

A[i] = new int[M];

//ввод элементов матрицы с клавиатуры

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

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

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

cout << "Элемент [" << i << "," << j << "]: ";

cin >> A[i][j];

}

}

//вывод полученной матрицы на экран

cout << "Полученная матрица: " << endl;

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

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

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

}

cout << endl;

}

//переход к функии

invertNegative(A, N, M);

//освобождение динамической памяти

for (i = 0; i < N; i++)

delete A[i];

delete[] A;

}

break;

case 2: {

int size = 0; //ввод размера строки

cout << "Введите размер строки: "; cin >> size;

cin.ignore(); // очистка символа новой строки из буфера

char\* str = new char[size + 1]; //выделение памяти для строки

cout << "Введите строку: "; //ввод строки с клавиатуры

cin.getline(str, size + 1); // считываем строку целиком

transformString(str, size + 1); //переход к функции

delete[] str; //освобождение памяти

}

break;

case 3: { //выход из программы

break;

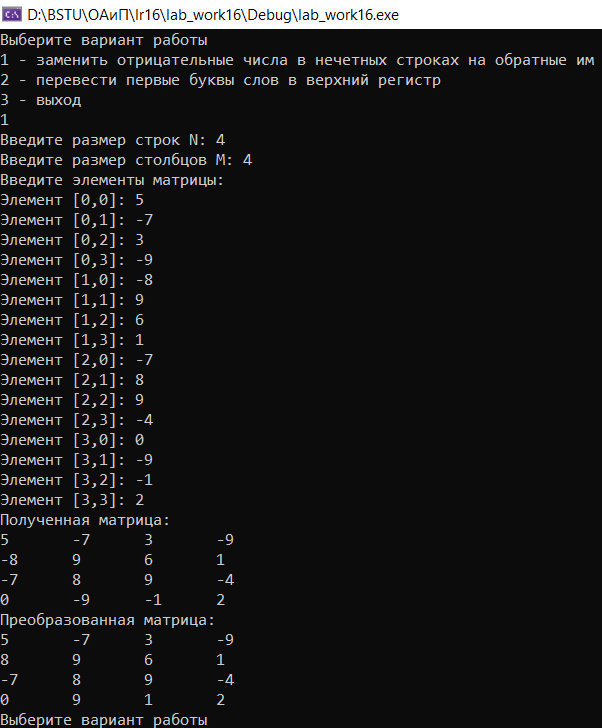
}

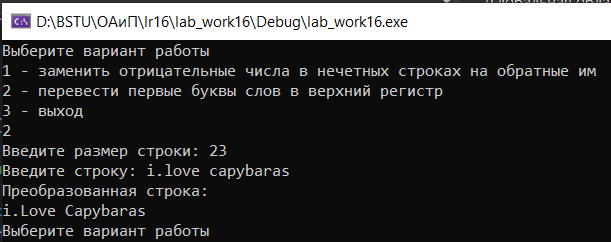
}

} while (choice != 3);

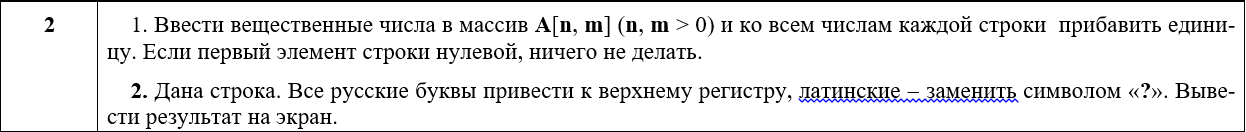
}

Результат





# Вариант 2



#include <iostream>

using namespace std;

//функция для замены отрицательных элементов в матрице

void addElement(int\*\* A, int N, int M)

{

int i, j; //ввод переменных итератора

//прибавить ко всем элементам 1, если первый элемент не равен 0

if (A[0][0] != 0) {

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

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

A[i][j]++;

}

}

//вывод преобразованной матрицы на экран

cout << "Преобразованная матрица: " << endl;

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

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

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

}

cout << endl;

}

}

else { //если первый элемент 0, то вывести соответствующее сообщение

cout << "Первым элементом является 0" << endl;

}

}

//фукнция для преобразования букв строки

void transformString(char\* str, int size) {

int i;

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

if (str[i] >= 'А' && str[i] <= 'я') { //привести русские буквы к верхнему регистру

str[i] = toupper(str[i]);

}

else if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z')) { //привести русские буквы к верхнему регистру

str[i] = '?';

}

}

cout << "Преобразованная строка: \n" << str << endl; //вывод преообразованной строки

}

//главная функция

int main()

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do { //выбор вариантов работы программы

cout << "Выберите вариант работы" << endl;

cout << "1 - ко всем числам массива прибавить 1" << endl;

cout << "2 - перевести русскмие буквы слов в верхний регистр, а латинские заменить на вопросительный знак" << endl;

cout << "3 - выход" << endl;

cin >> choice;

switch (choice)

{

case 1: {

int N, M, i, j; //ввод размеров матрицы, переменных итератора, числа k

cout << "Введите размер строк N: "; cin >> N; //ввод размера строк матрицы

cout << "Введите размер столбцов M: "; cin >> M; //ввод размера столбцов матрицы

while (M <= 0) {

cout << "Введите число больше 0: "; cin >> M;

}

//выделение динамической памяти для матрицы

int\*\* A;

A = new int\* [N];

for (i = 0; i < N; i++)

A[i] = new int[M];

//ввод элементов матрицы с клавиатуры

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

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

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

cout << "Элемент [" << i << "," << j << "]: ";

cin >> A[i][j];

}

}

//вывод полученной матрицы на экран

cout << "Полученная матрица: " << endl;

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

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

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

}

cout << endl;

}

//переход к функии

addElement(A, N, M);

//освобождение динамической памяти

for (i = 0; i < N; i++)

delete A[i];

delete[] A;

}

break;

case 2: {

int size = 0; //ввод размера строки

cout << "Введите размер строки: "; cin >> size;

cin.ignore(); // очистка символа новой строки из буфера

char\* str = new char[size + 1]; //выделение памяти для строки

cout << "Введите строку: "; //ввод строки с клавиатуры

cin.getline(str, size + 1); // считываем строку целиком

transformString(str, size + 1); //переход к функции

delete[] str; //освобождение памяти

}

break;

case 3: { //выход из программы

break;

}

}

} while (choice != 3);

}

