Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 16

По дисциплине «Основы алгоритмизации и программирования»

На тему «Функции пользователя»

 Выполнила:

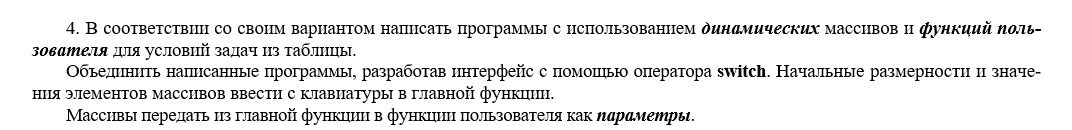
Студентка1 курса 6 группы

Альшевская Алина Михайловна

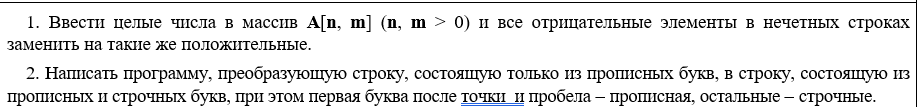
Преподаватель: асс. Андронова М.В.

2023, Минск

Задание 4



Вариант 1



#include <iostream>

#include <Windows.h>

using namespace std;

void replaceNegativesInOddRows(int\*\* arr, int n, int m) {

for (int i = 0; i < n; i += 2) { // Перебираем нечетные строки

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

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

// Если элемент отрицательный, заменяем на его абсолютное значение

arr[i][j] = -arr[i][j];

}

}

}

}

void transformString(char\* str) {

bool capitalizeNext = true; // Флаг для определения, следует ли делать следующую букву заглавной

while (\*str != '\0') {

if (\*str == '.' && \*(str + 1) == ' ') {

str++;

capitalizeNext = true; // Если встречается точка или пробел, следующая буква должна быть заглавной

}

else {

if (capitalizeNext) {

\*str = toupper(\*str); // Если флаг установлен, делаем букву заглавной

capitalizeNext = false; // Сбрасываем флаг только после обработки текущей буквы

}

else {

\*str = tolower(\*str); // Иначе делаем букву строчной

}

}

str++;

}

}

int main() {

SetConsoleOutputCP(1251);

char choice;

cout << "1 ? 2: "; cin >> choice;

int rmn = -20, rmx = 20;

switch (choice)

{

case '1': {

int n, m;

// Ввод размеров массива

cout << "Введите количество строк и столбцов (через пробел): ";

cin >> n >> m;

// Создание динамического массива

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

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

A[i] = new int[m];

}

srand((unsigned)time(NULL));

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

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

A[i][j] = ((int)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn;

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

}

cout << "\n";

}

// Вызов функции для замены отрицательных элементов в нечетных строках

replaceNegativesInOddRows(A, n, m);

// Вывод измененного массива

cout << "Измененный массив:\n";

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

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

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

}

cout << endl;

}

// Освобождение динамической памяти

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

delete[] A[i];

}

delete[] A;

break; }

case '2':

{

const int maxSize = 100; // Максимальный размер входной строки

char str[maxSize];

cout << "Введите строку из прописных букв: ";

cin.ignore(); // Игнорируем символ новой строки из предыдущего ввода

cin.getline(str, maxSize);

// Вызов функции для преобразования строки

transformString(str);

// Вывод измененной строки

cout << "Преобразованная строка: " << str << endl;

break;

}

default:

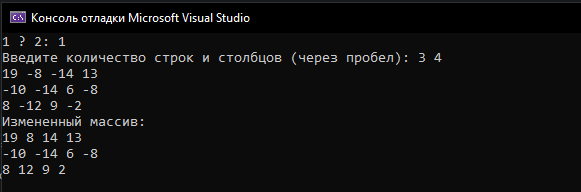
cout << "неправильный вариант";

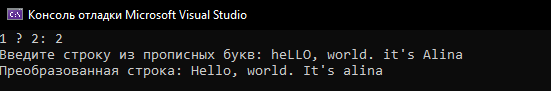
break;

}

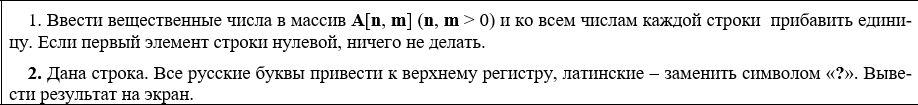
return 0;

}





Вариант 2



#include <iostream>

#include <Windows.h>

using namespace std;

//функция для увеличения элементов матрицы, отличных от 0, на 1

void decreaseMatrixValues(double\*\* ptr, int n, int m) {

int result = 0;

//перебираем элементы

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

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

if (ptr[i][j] != 0) { //если это не 0, то

ptr[i][j] = ptr[i][j] + 1; //увеличиваем на 1

cout << ptr[i][j] << ' '; //выводим

}

else {

cout << ptr[i][j] << ' '; //иначе мы просто выводим

}

}

cout << '\n';

}

}

//функция для изменения исходного предложения

void charsOccurrences(char\* ptr) {

int count = 0;

for (int i = 0; ptr[i] != '\0'; i++) {

if ('а' <= ptr[i] && ptr[i] <= 'я') { //если это маленькая русская буква

ptr[i] = ptr[i] - 32; //сделаем его большим, вычитая из яйца 32 (из кода будет вычтено 32, разница между маленьким и большим составляет ровно 32)

cout << ptr[i];

}

else if ('A' <= ptr[i] && ptr[i] <= 'z') { //для английских букв

ptr[i] = '?'; //изменяем на вопросительный знак

cout << ptr[i];

}

else {

cout << ptr[i];//иначе не меняем

}

}

}

int main()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

char choice;

cout << "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.\n";

cout << "2. Дана строка. Все русские буквы привести к верхнему регистру, латинские - заменить символом «?». Вывести результат на экран. \n";

cin >> choice;

switch (choice) {

case'1':

{

int n, m, rmn = -10, rmx = 10;

double\*\* ptr;

//Создаем массив размером n на m, используя new

cout << "Введите размер матрицы n x m: "; cin >> n >> m;

ptr = new double\* [n];

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

ptr[i] = new double[m];

srand((unsigned)time(NULL));

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

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

ptr[i][j] = ((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn;

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

}

cout << '\n';

}

cout << "Модифицированная матрица: \n";

decreaseMatrixValues(ptr, n, m);

break;

}

case'2':

{

char\* text;

cin.ignore(); //чтобы избежать проблем с вводом, когда может остаться символ \n и будет введена пустая строка

text = new char[100]; //создаем массив из 100 символов

cin.getline(text, 100); //вводим строку с помощью getline, чтобы учесть пробелы

charsOccurrences(text); //вызываем функцию и отображаем результат

break;

}

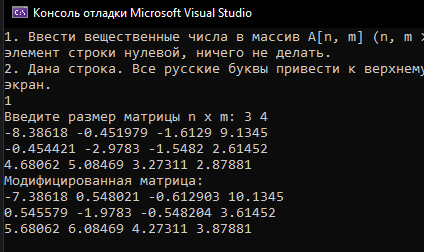
default:

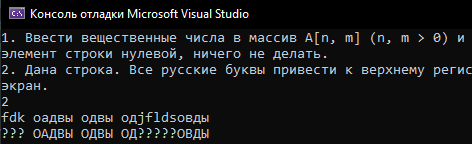
cout << "неправильный вариант";

return 0;

}

}





Вариант 3

#include <iostream>

#include <Windows.h>

using namespace std;

// Функция для нахождения суммы элементов под главной диагональю

int sumBelowMainDiagonal(int\*\* ptr, int n) {

int result = 0;

// Перебираем элементы под главной диагональю и складываем их

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

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

result += ptr[i][j];

}

}

// Возвращаем сумму

return result;

}

// Функция для подсчёта количества вхождений символа symbol

int charsOccurrences(char\*\* ptr, int numLines, char symbol) {

int count = 0;

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

for (int j = 0; ptr[i][j] != '\0'; ++j) { // Если '\0', то строка закончилась и дальше не идём

if (ptr[i][j] == symbol) { // Подсчитываем

count++;

}

}

}

// Возвращаем результат

return count;

}

int main() {

SetConsoleOutputCP(1251);

// Выбор, что делать

char choice;

// Для первого задания

int n, rmn = -10, rmx = 10;

int\*\* ptr;

// Для второго задания

char\*\* text;

int numLines;

char symbol;

cout << "Выберите вариант:\n";

cout << "1. Найти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве A[n, n].\n";

cout << "2. Написать программу, которая вводит несколько строк текста и символ «m» и использует функцию, чтобы определить суммарное число вхождений символа в текст. \n";

cin >> choice;

switch (choice) {

case '1':

// Создаём массив n на n с помощью new

cout << "Введите размер матрицы: "; cin >> n;

ptr = new int\* [n];

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

ptr[i] = new int[n];

srand((unsigned)time(NULL)); // Инициализируем генератор случайных чисел

// Заполняем массив случайными числами и сразу выводим его

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

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

ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

cout << '\n';

}

// Вызов функции

cout << sumBelowMainDiagonal(ptr, n);

break;

case '2':

cout << "Введите количество строк: "; cin >> numLines;

cout << "Введите символ(регистр учитывается): "; cin >> symbol;

cin.ignore(); // Чтобы избежать проблем со вводом

text = new char\* [numLines]; // Создаём массив на numLines строк

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

text[i] = new char[100]; // 100 символов должно хватить на предложение

cout << "Введите строку " << (i + 1) << ": ";

cin.getline(text[i], 100);

}

cout << charsOccurrences(text, numLines, symbol);

break;

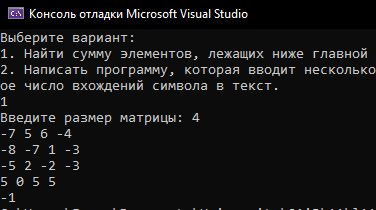
default:

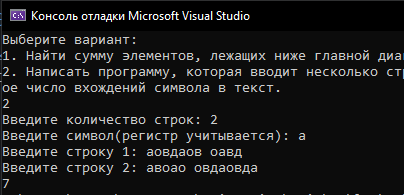
cout << "неправильный вариант";

return 0;

}

}





Вариант 7

#include <iostream>

#include <Windows.h>

using namespace std;

// Функция для поиска индекса строки, содержащей число b

void IndexOfTheRow(int\*\* ptr, int n, int m, int b) {

int minMaxIndex = 0;

// Перебираем элементы под главной диагональю и суммируем их

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

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

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

cout << "Индекс строки, где встречается " << b << ": " << ++i;

return; // Выводим индекс первого ряда, где встретилось число и выходим из функции

}

}

}

// Если число не было найдено, то будет выполнен этот код

cout << "Числа b нет в массиве";

}

// Функция для подсчета суммы кодов символов каждого слова

int countChars(char\* ptr) {

int count = 0;

for (int i = 0; ptr[i] != '\0'; i++) { // Если '\0', то строка закончилась и дальше не идем

if (ptr[i] != ' ') { // Считаем кроме пробелов

count+=int(ptr[i]);

}

}

// Возвращаем результат

return count;

}

int main() {

SetConsoleOutputCP(1251);

// Выбор, что делать

char choice;

// Для первого задания

int n, m, b, rmn = -10, rmx = 10;

int\*\* ptr;

// Для второго задания

char\* text;

cout << "Выберите вариант: \n";

cout << "1. Дана двумерная матрица, состоящая из N строк и М столбцов, а также число d. Найти строку, содержащую число d. \n";

cout << "2. В строке подсчитать сумму кодов символов каждого слова. Слова в строке разделены пробелами. \n";

cin >> choice;

switch (choice) {

case '1':

// Создаем массив n на n с помощью new

cout << "Введите размер матрицы n x m: "; cin >> n >> m;

cout << "Введите число, которое нужно найти: "; cin >> b;

ptr = new int\* [n];

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

ptr[i] = new int[m];

srand((unsigned)time(NULL));

// Заполняем массив случайными числами и сразу выводим его

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

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

ptr[i][j] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

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

}

cout << '\n';

}

// Вызов функции

IndexOfTheRow(ptr, n, m, b);

break;

case '2':

cin.ignore();

text = new char[256];

cin.getline(text, 256);

countChars(text);

// Вызов функции и вывод результата

cout << countChars(text);

break;

default:

cout << "Неверный вариант";

return 0;

}

}

