Министерство образование Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Информационно-технологический факультет

Кафедра технологий программирования

Лабораторная работа №3

По дисциплина «Основы алгоритмизации и программирования»

Выполнил: Крутилин С.Д.

Группа 19-ИТ-1.

Проверил: Адамовский Е.

Новополоцк

2019 г.

**Динамическая память**

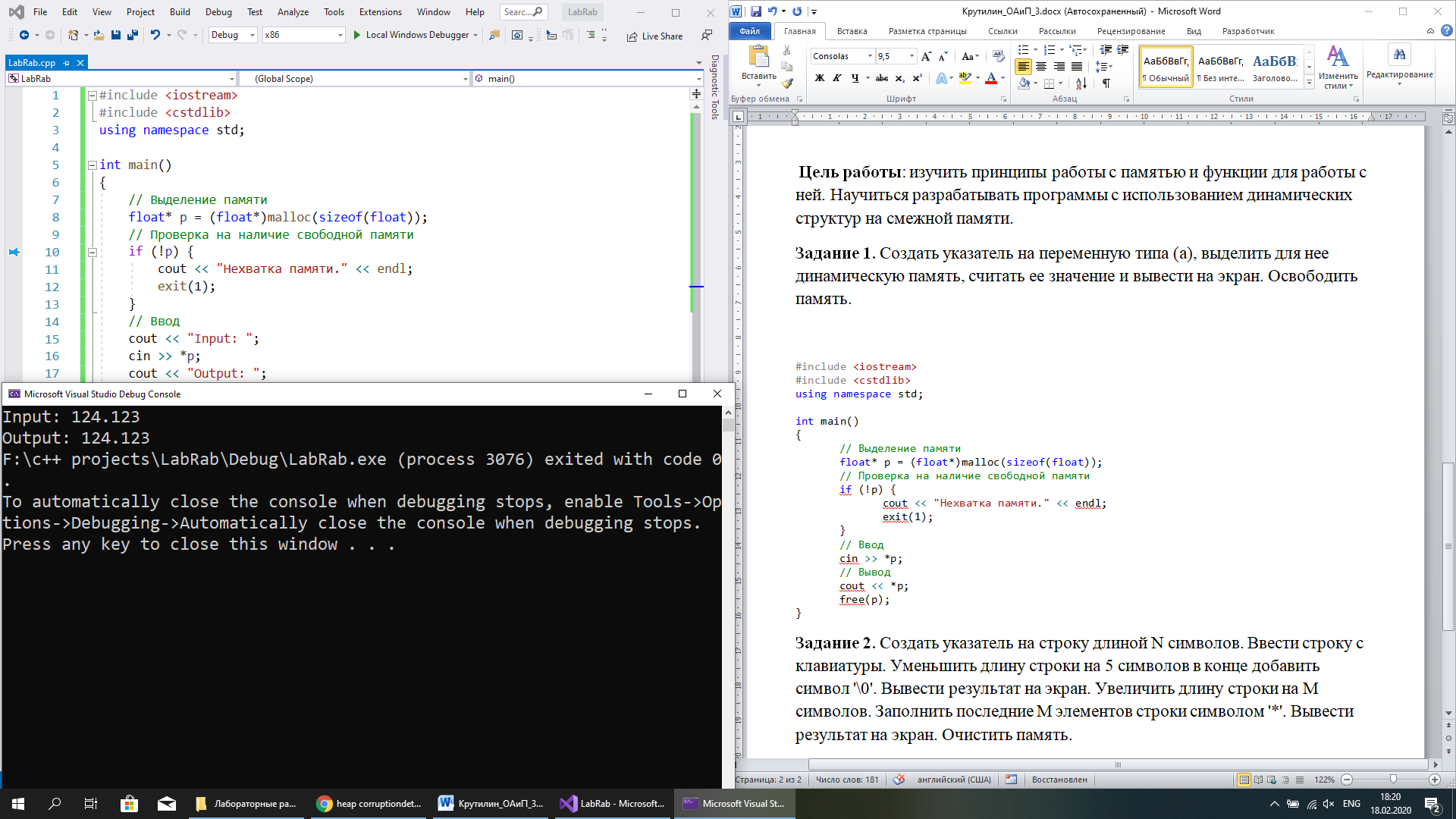
**Вариант 2**

**Цель работы**: изучить принципы работы с памятью и функции для работы с ней. Научиться разрабатывать программы с использованием динамических структур на смежной памяти.

**ЗАДАНИЕ 1.** Создать указатель на переменную типа (а), выделить для нее динамическую память, считать ее значение и вывести на экран. Освободить память.

**Алгоритм:** Выделяем память динамически с помощью malloc(также используем sizeof для определения необходимого кол-ва памяти под наши данные); делаем проверку на выделение памяти; считываем данные; выводим данные.

**Результат работы программы:**



#include <iostream>

#include <cstdlib>

using namespace std;

int main()

{

// Выделение памяти

float\* p = (float\*)malloc(sizeof(float));

// Проверка на наличие свободной памяти

if (!p) {

cout << "Нехватка памяти." << endl;

exit(1);

}

// Ввод

cin >> \*p;

// Вывод

cout << \*p;

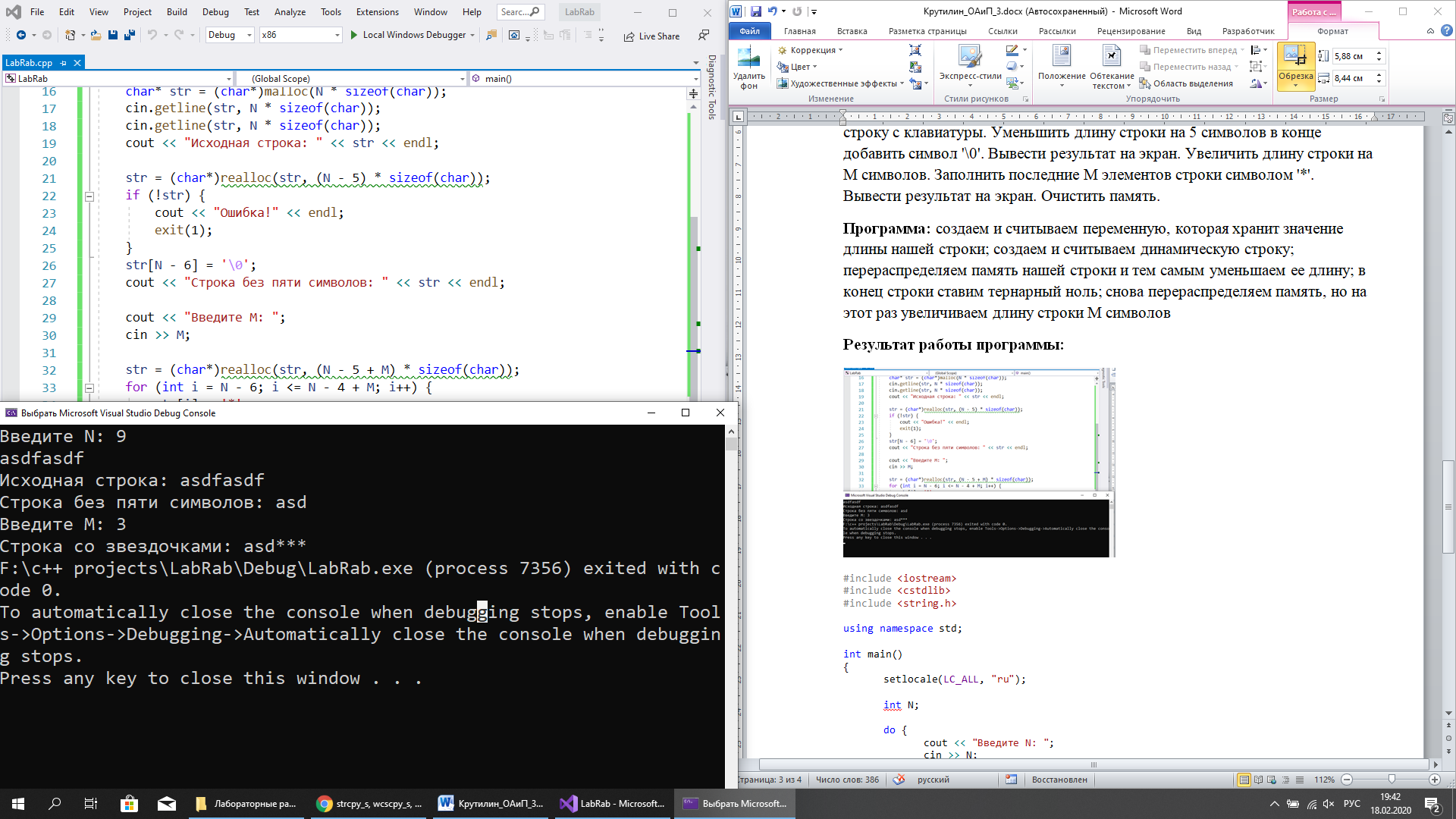
free(p);

}

**ЗАДАНИЕ 2.** Создать указатель на строку длиной N символов. Ввести строку с клавиатуры. Уменьшить длину строки на 5 символов в конце добавить символ '\0'. Вывести результат на экран. Увеличить длину строки на M символов. Заполнить последние M элементов строки символом '\*'. Вывести результат на экран. Очистить память.

**Программа:** создаем переменные N и M; считываем N; создаем и считываем динамическую строку; перераспределяем память нашей строки и тем самым уменьшаем ее длину; в конец строки ставим тернарный ноль; снова перераспределяем память, но на этот раз увеличиваем длину строки на M символов; заполняем пустые ячейки звездочками и закрываем строку тернарным нулем; освобождаем память.

**Результат работы программы:**



#include <iostream>

#include <cstdlib>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int N, M;

cout << "Введите N: ";

cin >> N;

char\* str = (char\*)malloc(N \* sizeof(char));

cin.getline(str, N \* sizeof(char));

cin.getline(str, N \* sizeof(char));

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

str = (char\*)realloc(str, (N - 5) \* sizeof(char));

if (!str) {

cout << "Ошибка!" << endl;

exit(1);

}

str[N - 6] = '\0';

cout << "Строка без пяти символов: " << str << endl;

cout << "Введите M: ";

cin >> M;

str = (char\*)realloc(str, (N - 5 + M) \* sizeof(char));

if (!str) {

cout << "Ошибка!" << endl;

exit(1);

}

for (int i = N - 6; i <= N - 4 + M; i++) {

str[i] = '\*';

}

str[N - 6 + M] = '\0';

cout << "Строка со звездочками: " << str;

free(str);

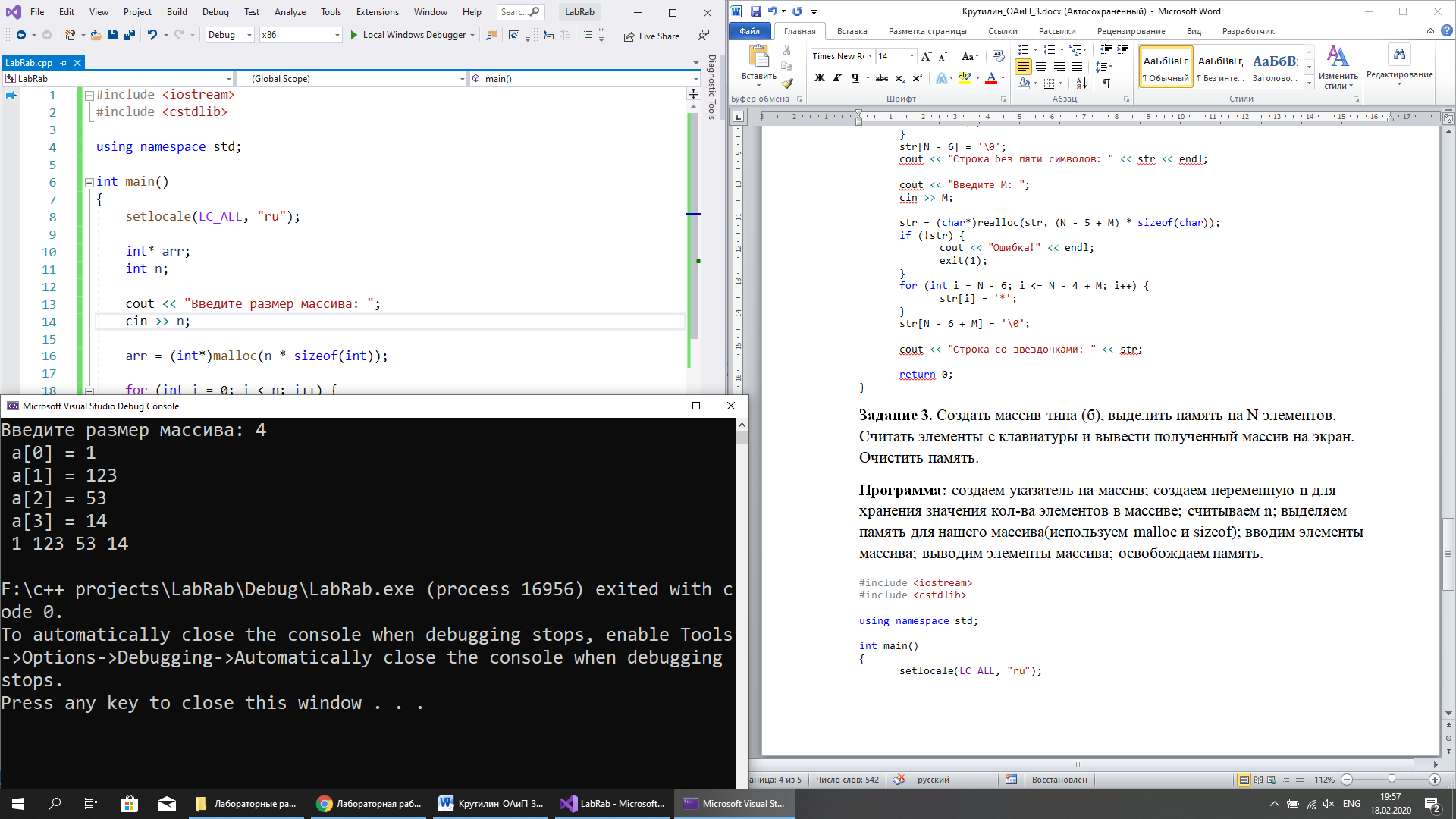
return 0;

}

**Задание 3.** Создать массив типа (б), выделить память на N элементов. Считать элементы с клавиатуры и вывести полученный массив на экран. Очистить память.

**Программа:** создаем указатель на массив; создаем переменную n для хранения значения кол-ва элементов в массиве; считываем n; выделяем память для нашего массива(используем malloc и sizeof); вводим элементы массива; выводим элементы массива; освобождаем память.

**Результат работы программы:**



#include <iostream>

#include <cstdlib>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

int\* arr;

int n;

cout << "Введите размер массива: ";

cin >> n;

arr = (int\*)malloc(n \* sizeof(int));

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

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

cin >> arr[i];

}

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

cout << " " << arr[i];

}

cout << endl;

free(arr);

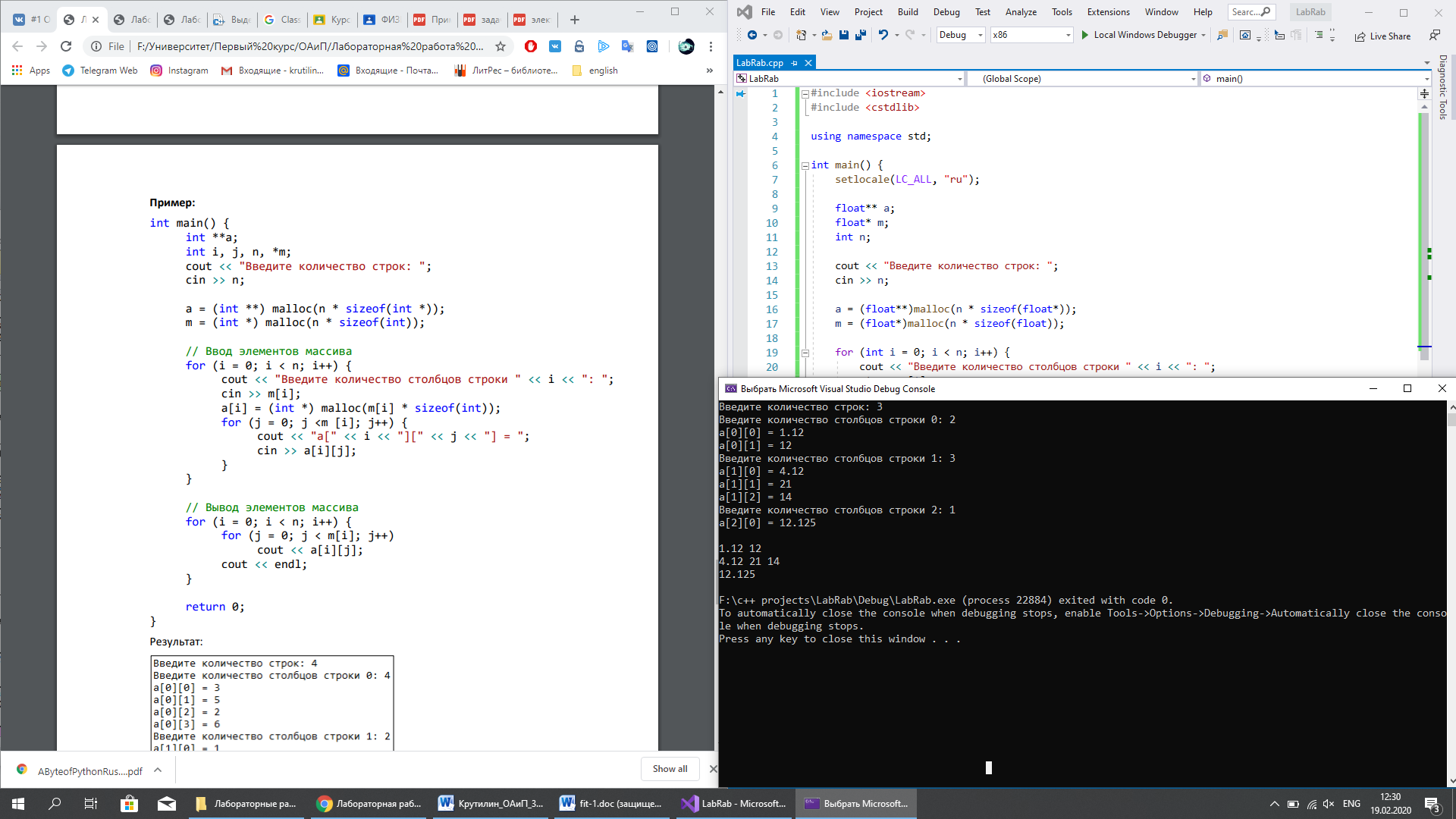
return 0;

}

**ЗАДАНИЕ 4.** Создать свободный массив типа (а). Считать количество элементов в строке и значения элементов с клавиатуры. Вывести результат на экран. Очистить память.

**Программа:** создаем указатель на указатель на строку; создаем указатель на массив, который хранит длину нашей строки ; создаем целочисленную переменную для хранения кол-ва строк; выделяем динамическую память для хранения нашего свободного массива (используем malloc и sizeof); в цикле по строкам считываем значения для длины строк, затем выделяем память и заполняем эти строки; выводим массив; освобождаем память.

**Результат работы программы:**



#include <iostream>

#include <cstdlib>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

float\*\* a;

int\* m;

int n;

cout << "Введите количество строк: ";

cin >> n;

a = (float\*\*)malloc(n \* sizeof(float\*));

m = (int\*)malloc(n \* sizeof(int));

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

cout << "Введите количество столбцов строки " << i << ": ";

cin >> m[i];

a[i] = (float\*)malloc(m[i] \* sizeof(float));

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

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

cin >> a[i][j];

}

}

cout << endl;

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

for (int j = 0; j < m[i]; j++)

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

cout << endl;

free(a[i]);

}

free(m);

free(a);

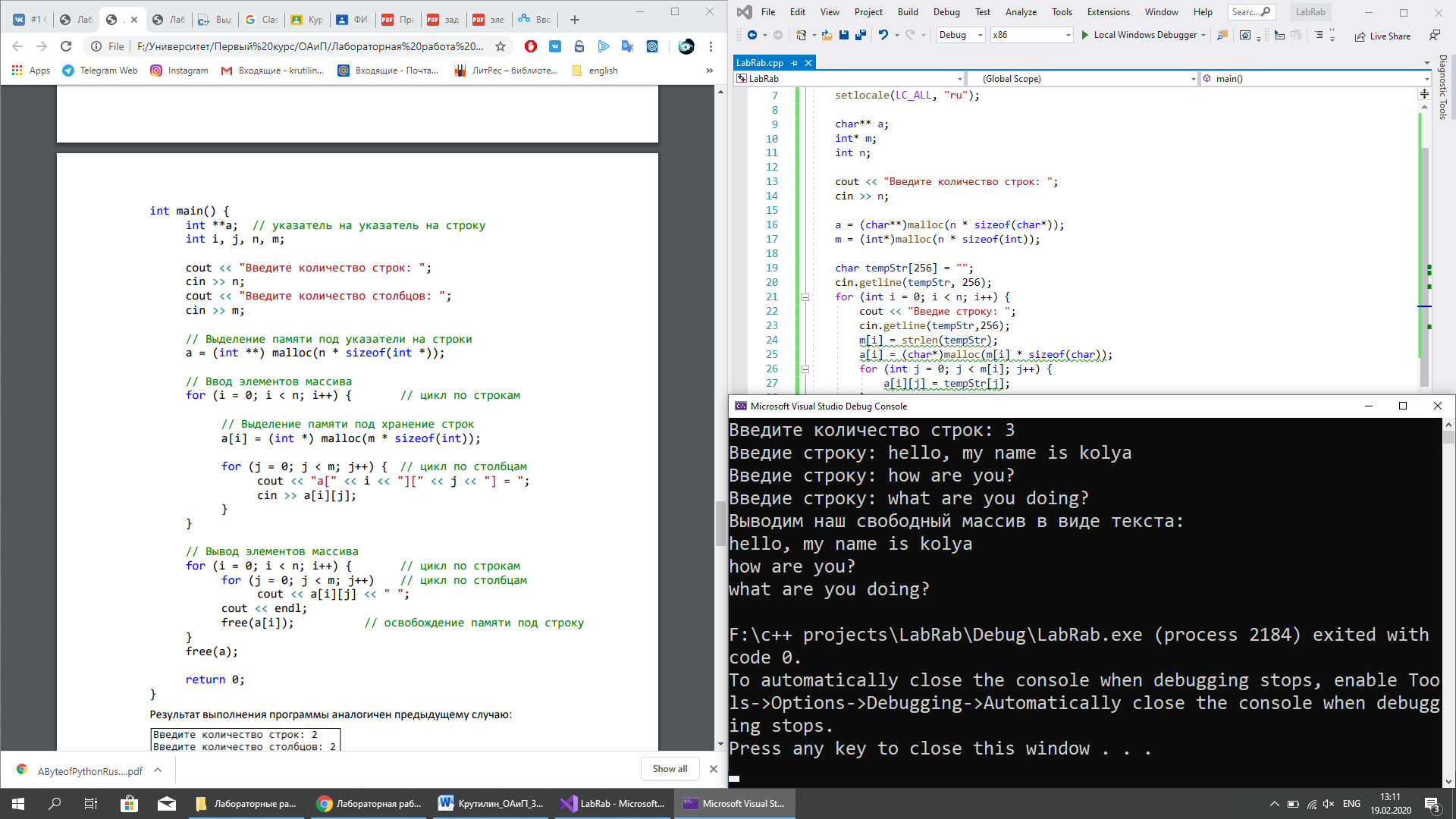
return 0;

}

**Задание 5.** В динамический свободный массив записывать строки, заранее считанные с клавиатуры. То есть, сначала считывается строка, затем определяется ее длина, после чего она заносится в строку динамического массива. Вывести содержимое массива в виде текста. Очистить память.

**Программа:** принцип работы программы такой же как и в 4 задаче, кардинально меняется только часть со считыванием данных и способ записи в наш массив;

**Результат работы программы:**



#include <iostream>

#include <cstdlib>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

char\*\* a;

int\* m;

int n;

cout << "Введите количество строк: ";

cin >> n;

a = (char\*\*)malloc(n \* sizeof(char\*));

m = (int\*)malloc(n \* sizeof(int));

char tempStr[256] = "";

cin.getline(tempStr, 256);

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

cout << "Введие строку: ";

cin.getline(tempStr,256);

m[i] = strlen(tempStr);

a[i] = (char\*)malloc(m[i] \* sizeof(char));

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

a[i][j] = tempStr[j];

}

}

cout << "Выводим наш свободный массив в виде текста: " << endl;

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

for (int j = 0; j < m[i]; j++)

cout << a[i][j];

cout << endl;

free(a[i]);

}

free(m);

free(a);

return 0;

}