ФТФ, 2-ый курс, ИВТ-Б, Королёв Алексей.

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Тема:** Синтаксис языка C++. Работа с памятью в С++.

**Цель:** Научиться программировать на языке С++ базовые операции с данными разных типов, разветвляющиеся и циклические алгоритмы, научиться работать с динамической памятью в С++, осуществлять консольный ввод-вывод данных.

**Индивидуальные задания:**

1. Написать программу обработки одномерного массива с n элементами, в соответствии с вариантом (приложение А). Необходимые действия должны производиться в функции processArray(), возвращающей определенное значение и формирующей еще один (выходной) массив данных (см. свой вариант). Память под массивы выделять статически (объявлять массивы локально в функции main) и для доступа к элементам использовать индексы. Ввод-вывод данных организовать средствами cstdio.

Объявить массив из n=16 целых чисел, проинициализировать единицами. Функция processArray() должна заполнить элементы массива с четными индексами степенями двойки (1, 2, 4, 8, 16, …), с нечетными индексами – степенями тройки (3, 9, 27, …). Также подсчитать и вернуть count – количество двузначных чисел в массиве и сформировать выходной массив, содержащий только такие числа. Вывести на экран результирующие массивы.

1. Написать программу, которая преобразует одномерный массив (1D) в двумерный (2D) (или наоборот), в соответствии с вариантом. Необходимо оформить в отдельных функциях код следующих действий: 1) инициализация массива; 2) вывод массива; 3) преобразование массива (создание нового массива с другой структурой). Память под массивы выделять динамически и для доступа к элементам использовать указатели. Ввод-вывод данных организовать средствами iostream и iomanip.

Преобразование: 1D → 2D. Одномерный массив из 18 целых чисел необходимо отсортировать в порядке убывания и разложить по двумерной сетке 9х2 слева направо и сверху вниз. Инициализация: заполнить массив числами x[i] = i 2+1 и все элементы с четными индексами домножить на -1. Вывод на экран: на каждый элемент массива отвести 5 позиций.

1. Написать свой аналог стандартной функции обработки строк из файла cstring, в соответствии с вариантом. В функции main на тестовых данных продемонстрировать результат работы как стандартной функции, так и собственной версии. Ввод-вывод данных организовать средствами cstdio.

Функция strcpy. Формат char\* strcpy(char\* dest, const char\* src). Функция копирует строку src в строку dest.

**Ход работы:**

1. Код программы:

#include <cstdio>

#include <cmath>

const int n = 16;

int processArray(int arr[], int outArr[], int& count) {

count = 0;

int outIndex = 0;

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

// Заполнение степенями двойками

if (i % 2 == 0) {

arr[i] = pow(2, i / 2);

}

// Заполнение степенями тройками

else {

arr[i] = pow(3, i / 2);

}

// Если число двухзначное, то добавляется в выходной массив, увелечивая счётчик count

if (arr[i] >= 10 && arr[i] <= 99) {

count++;

outArr[outIndex++] = arr[i];

}

}

return count;

}

int main() {

int arr[n];

int outArr[n];

int count;

// Инициализация массива единицами

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

arr[i] = 1;

}

// Обработка массива

processArray(arr, outArr, count);

// Вывод исходного массива

printf("The source array:\n");

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

printf("%d ", arr[i]);

}

printf("\n\n");

// Вывод выходного массива

printf("Output array (TD numbers):\n");

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

printf("%d ", outArr[i]);

}

printf("\n\n");

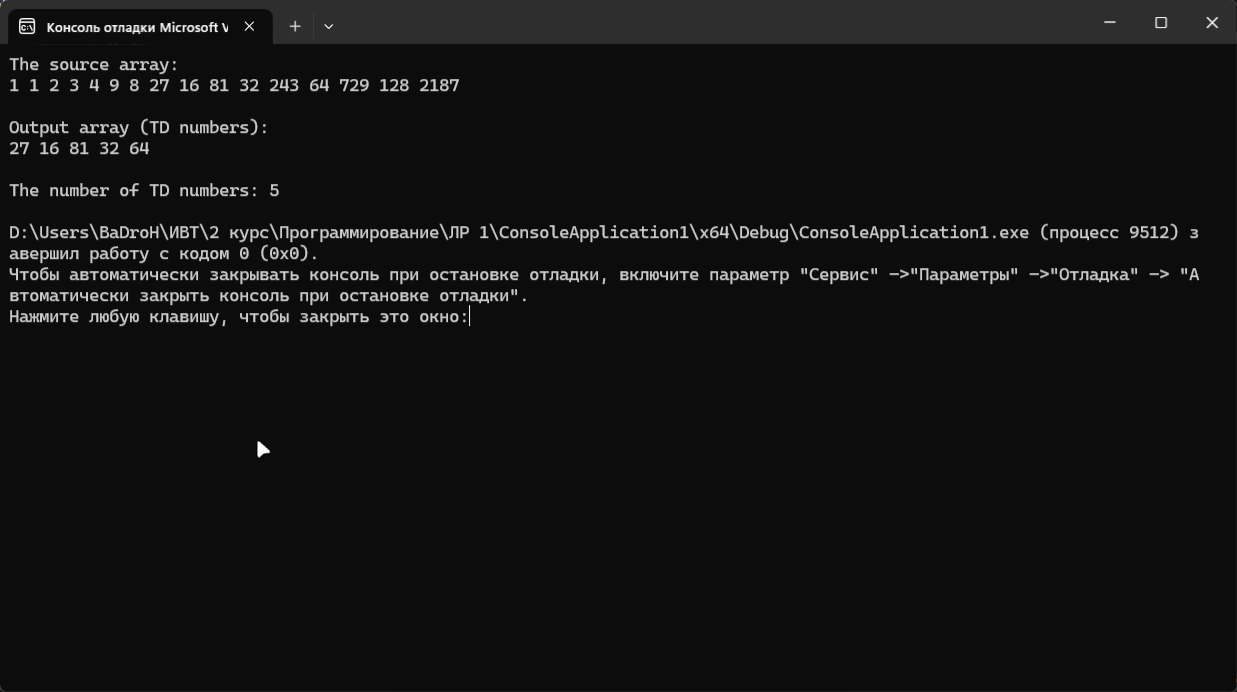
// Вывод количества двузначных чисел

printf("The number of TD numbers: %d\n", count);

return 0;

}

Результат программы:



1. Код программы:

#include <iostream>

#include <iomanip>

#include <algorithm>

void initializeArray(int\* arr, int size) {

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

arr[i] = i \* i + 1;

if (i % 2 == 0) {

arr[i] \*= -1;

}

}

}

void printArray(int\* arr, int size) {

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

std::cout << std::setw(5) << arr[i];

// Каждые 9 элементов происходит переход на новую строку

if ((i + 1) % 9 == 0) std::cout << std::endl;

}

std::cout << std::endl;

}

void transformArray(int\* arr1D, int size1D, int\*\*& arr2D, int& rows, int& cols) {

rows = 9;

cols = 2;

// Сортировка в порядке убывания

std::sort(arr1D, arr1D + size1D, std::greater<int>());

// Выделение памяти для 2D массива

arr2D = new int\* [rows];

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

arr2D[i] = new int[cols];

}

// Заполнение 2D массива

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

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

arr2D[i][j] = arr1D[i \* cols + j];

}

}

}

int main() {

const int size1D = 18;

int\* arr1D = new int[size1D];

initializeArray(arr1D, size1D);

std::cout << "1D array:" << std::endl;

printArray(arr1D, size1D);

int\*\* arr2D;

int rows, cols;

transformArray(arr1D, size1D, arr2D, rows, cols);

std::cout << "2D array:" << std::endl;

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

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

std::cout << std::setw(5) << arr2D[i][j];

}

std::cout << std::endl;

}

// Освобождение памяти

delete[] arr1D;

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

delete[] arr2D[i];

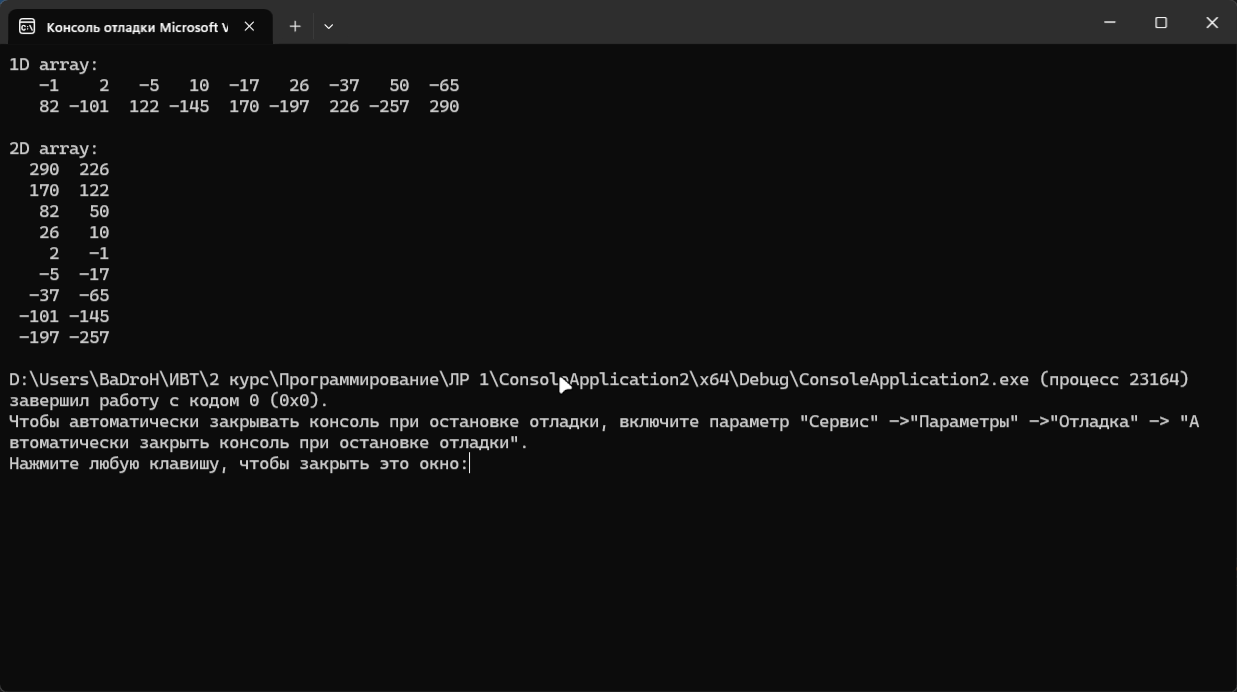
}

delete[] arr2D;

return 0;

}

Результат программы:



1. Код программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <cstdio>

#include <cstring>

// Пользовательская реализация strcpy

char\* my\_strcpy(char\* dest, const char\* src) {

char\* original\_dest = dest;

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

\*dest = \*src;

dest++;

src++;

}

\*dest = '\0';

return original\_dest;

}

int main() {

const char\* source = "Hello, World!";

char dest1[20];

char dest2[20];

// Использование безопасной функции strcpy\_s

strcpy\_s(dest1, sizeof(dest1), source);

printf("Standard function strcpy\_s:\n");

printf("String: %s\n", source);

printf("The copied string: %s\n\n", dest1);

// Использование пользовательской функции my\_strcpy

my\_strcpy(dest2, source);

printf("The user function my\_strcpy:\n");

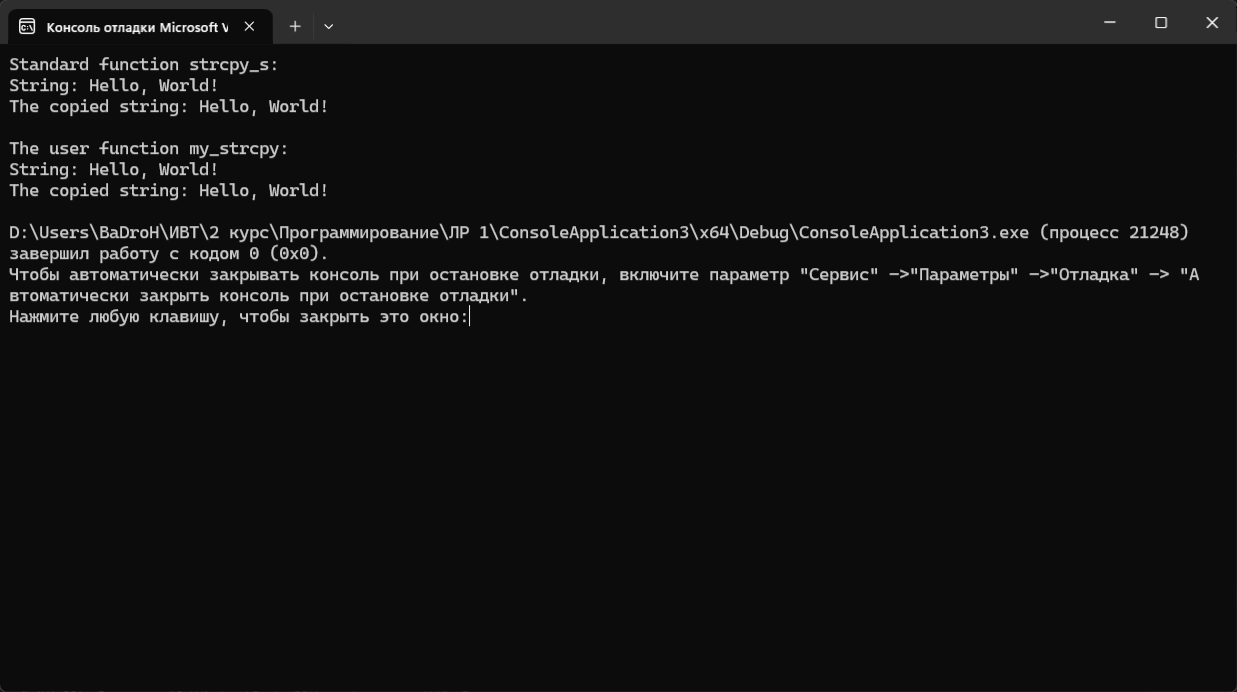
printf("String: %s\n", source);

printf("The copied string: %s\n", dest2);

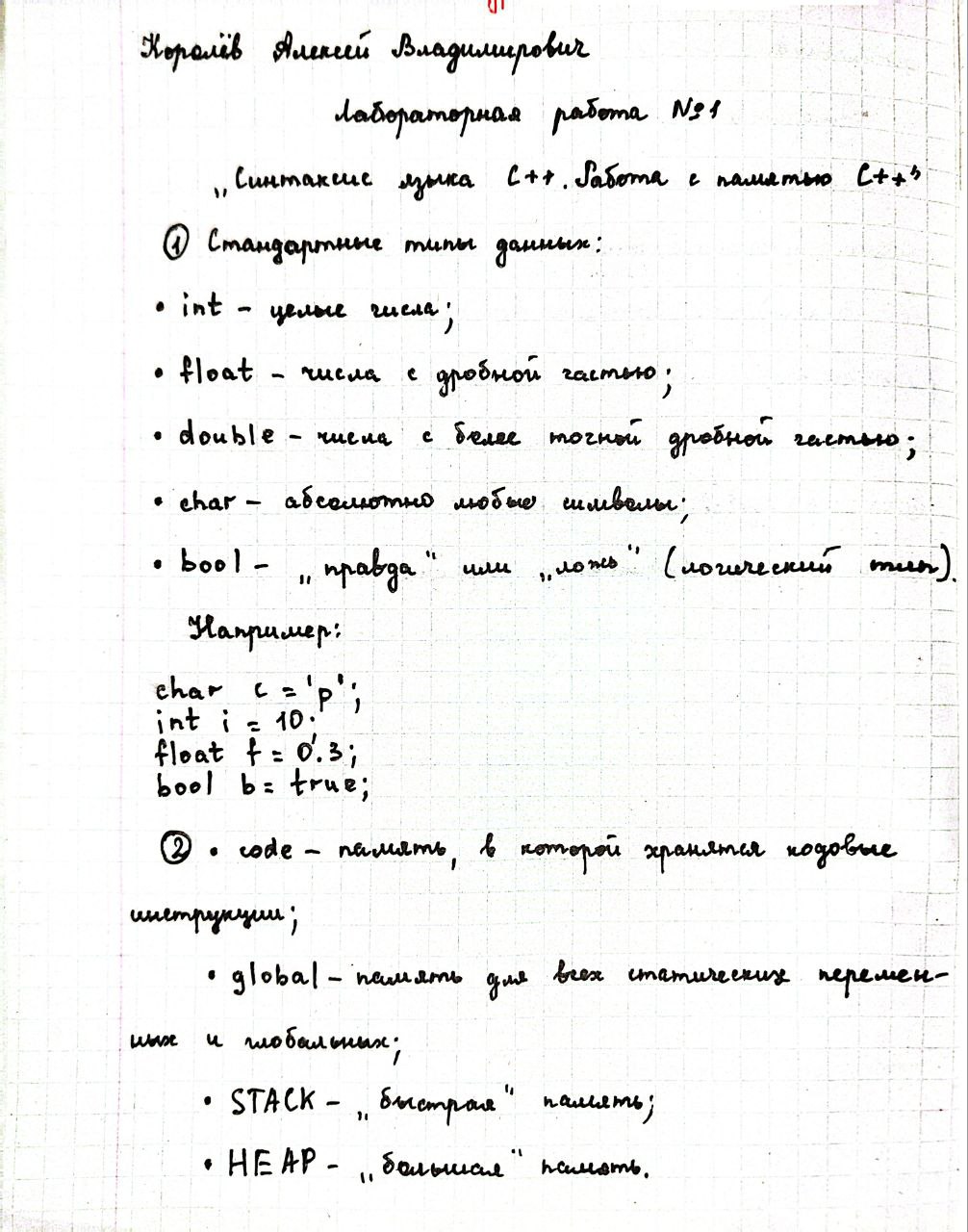
return 0;

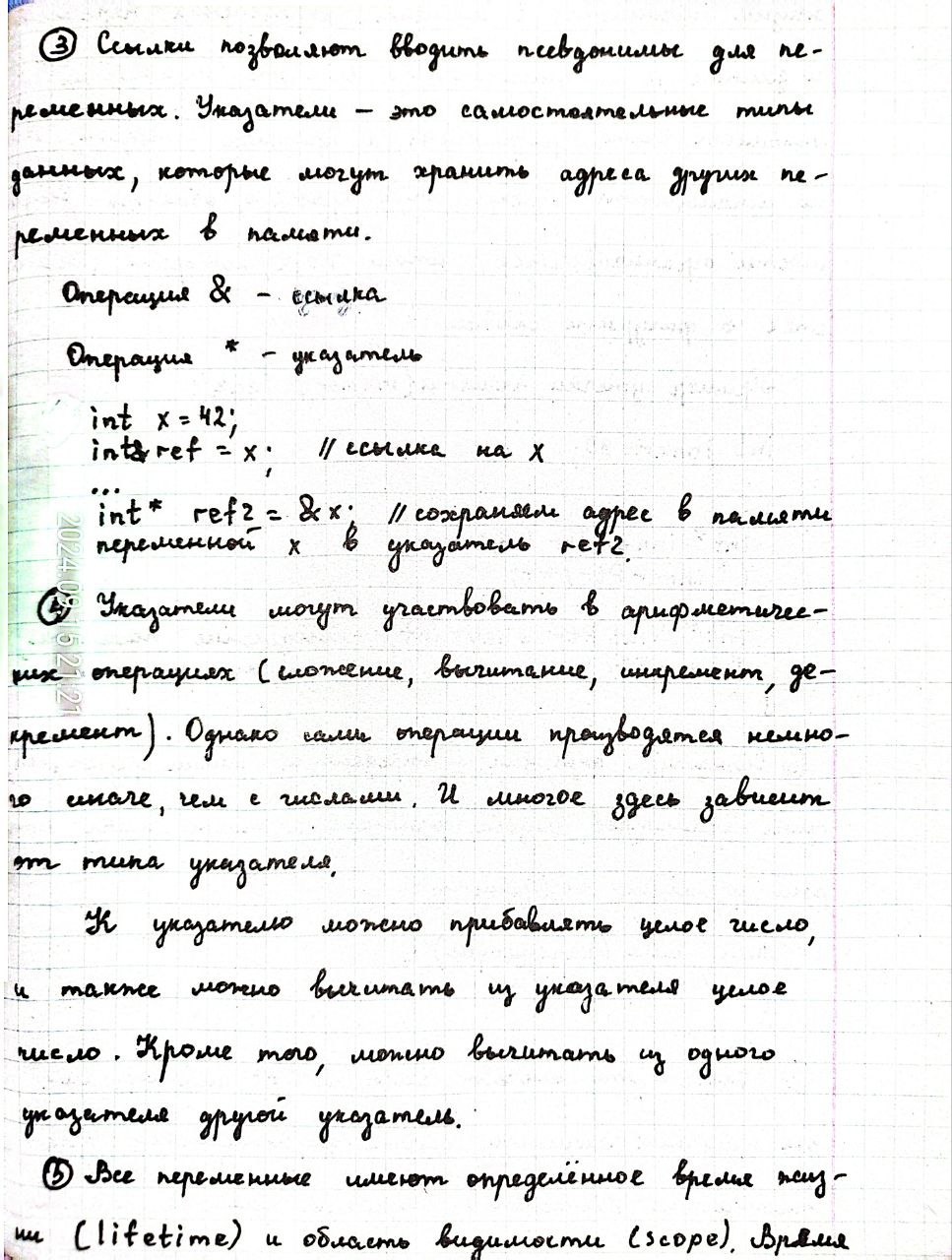
}

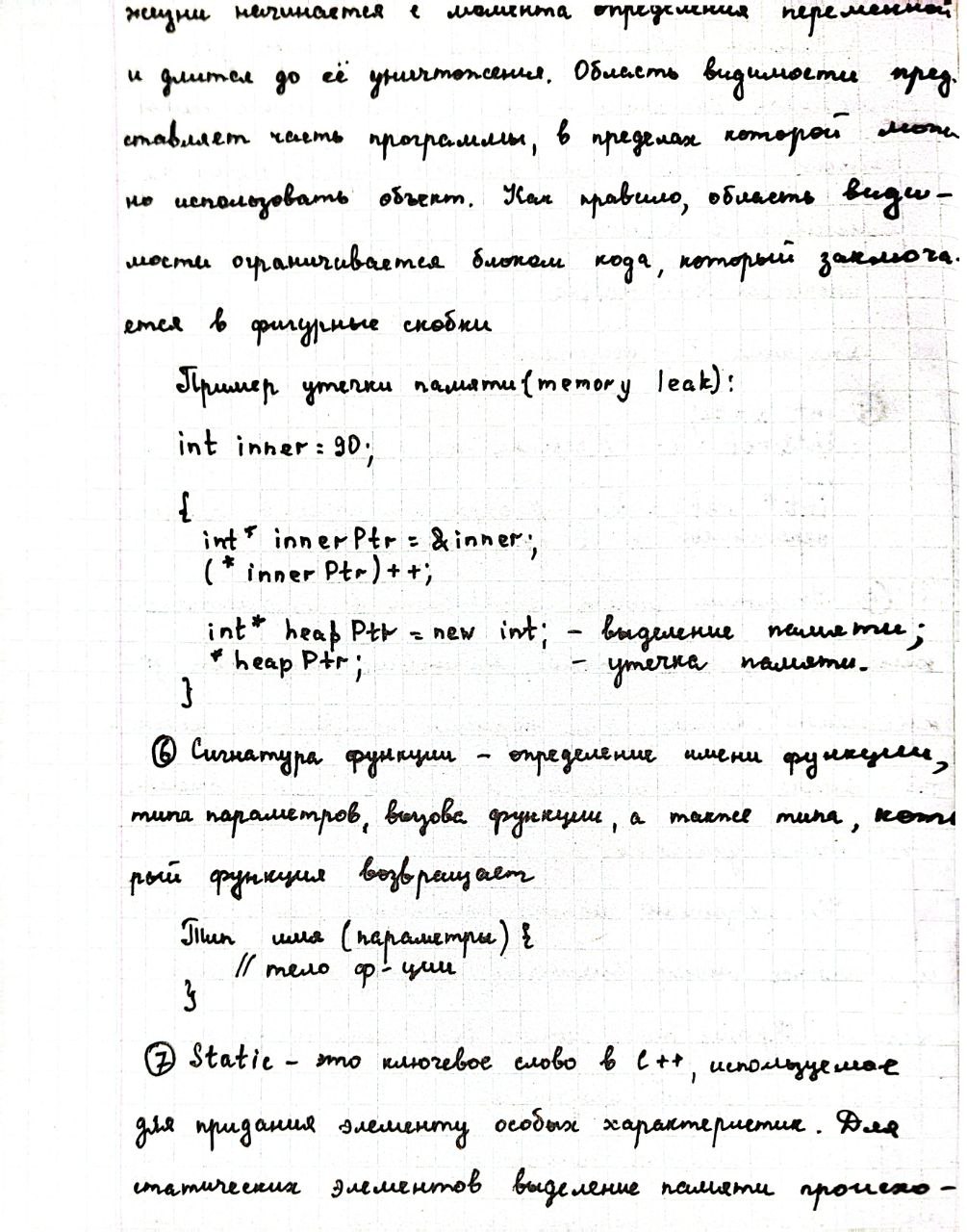
Результат программы:

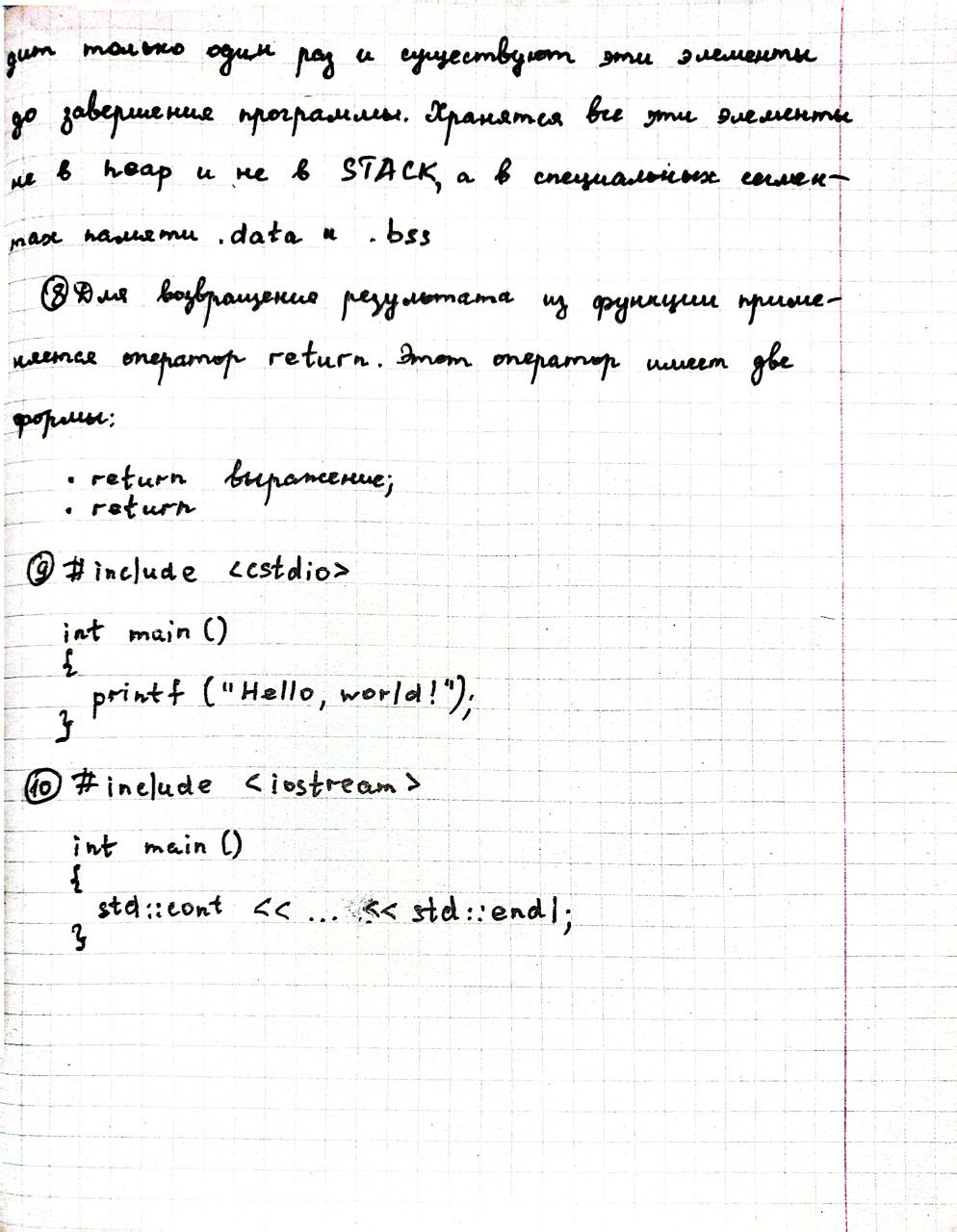


Контрольные вопросы:









Вывод: Научился программировать на языке С++ базовые операции с данными разных типов, разветвляющиеся и циклические алгоритмы, научился работать с динамической памятью в С++, осуществлять консольный ввод-вывод данных.