**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»**

**ХАБАРОВСКИЙ ИНСТИТУТ ИНФОКОММУНИКАЦИЙ (ФИЛИАЛ)**

**СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

по МДК 01.01 Системное программирование

на тему

Файлы. Работа с файлами.

Выполнил: А.В. Медведев

Проверил: Н.А. Насонова

Группа: ПКС-320

2023

Составление программ.

Для начала работы подключим библиотеки для работы с основными функциями. И подключим пространство имён std.

#include <iostream>

#include <fstream>

#include <algorithm>

#include <vector>

#include <iomanip>

using namespace std;

iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

fstream – заголовочный файл из стандартной библиотеки C++, включающий набор классов, методов и функций, которые предоставляют интерфейс для чтения/записи данных из/в файл.

algorithm – заголовочный файл в стандартной библиотеке языка программирования C++, включающий набор функций для выполнения алгоритмических операций.

vector – стандартный шаблон обобщённого программирования языка C++, реализующий динамический массив.

iomanip – библиотека, в которой содержится модификатор setw.

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

Дано имя файла и целое число N (> 1). Создать файл целых чисел с данным именем и записать в него N первых положительных четных чисел(2, 4, …).

#include <iostream>

#include <fstream>

using namespace std;

int main() {

int n;

string filename;

cout << "Enter the name of the file: ";

cin >> filename;

cout << "Enter a positive integer N: ";

cin >> n;

ofstream file;

file.open(filename);

for (int i = 2; i <= 2 \* n; i += 2) {

file << i << endl;

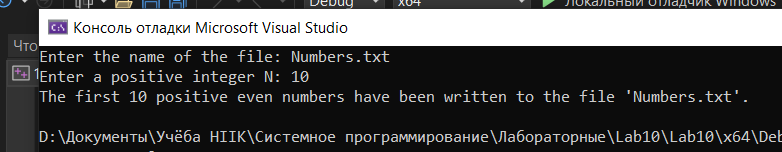
}

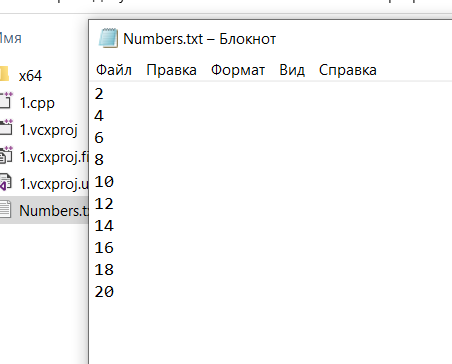
file.close();

cout << "The first " << n << " positive even numbers have been written to the file '" << filename << "'." << endl;

return 0;

}





Дано имя файла и вещественные числа A и D. Создать файл вещественных чисел с данным именем и записать в него 10 первых членов арифметической прогрессии с начальным членом A и разностью D : A, A + D, A + 2·D, A + 3·D, …

#include <iostream>

#include <fstream>

using namespace std;

int main() {

string filename;

float A, D;

cout << "Enter the name of the file: ";

cin >> filename;

cout << "Enter A: ";

cin >> A;

cout << "Enter D: ";

cin >> D;

ofstream file(filename, ios::out);

if (!file) {

cout << "Cannot open file" << endl;

return 1;

}

for (int i = 0; i < 10; i++) {

file << A + i \* D << endl;

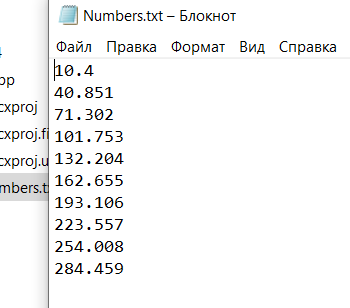
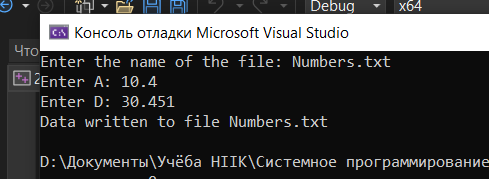
}

file.close();

cout << "Data written to file " << filename << endl;

return 0;

}



Дано имя файла целых чисел. Найти количество элементов, содержащихся в данном файле.Если файла с таким именем не существует, то вывести −1.

#include <fstream>

#include <iostream>

using namespace std;

int main() {

int kont = 0;

int x;

string filename;

cout << "Enter the name of the file: ";

cin >> filename;

ifstream fin(filename);

if (!fin) {

cout << -1;

return 0;

}

while (fin >> x) {

kont++;

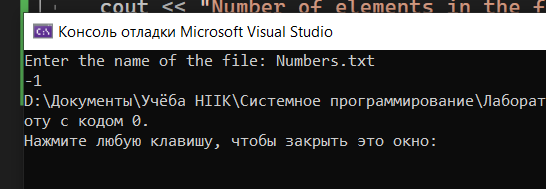
}

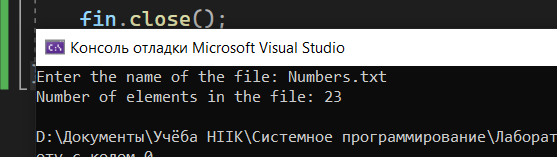
cout << "Number of elements in the file: " << kont << endl;

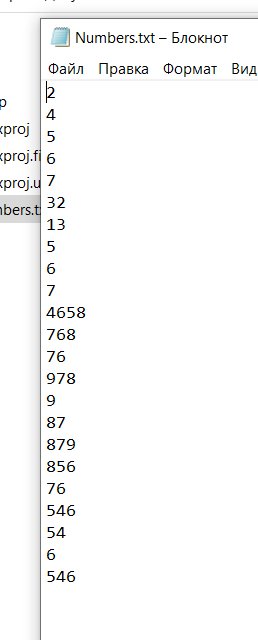
fin.close();

return 0;

}







Дано целое число K и файл, содержащий неотрицательные целые числа. Вывести K - й элемент файла.Если такой элемент отсутствует, то вывести −1.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

int k, num, cnt = 0;

cout << "Enter the value of k: ";

cin >> k;

string fileName;

cout << "Enter the name of the file: ";

cin >> fileName;

ifstream inFile(fileName);

if (!inFile.is\_open()) {

cout << -1 << endl;

return 0;

}

while (inFile >> num) {

cnt++;

if (cnt == k) {

cout << num << endl;

return 0;

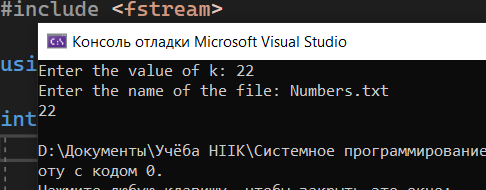
}

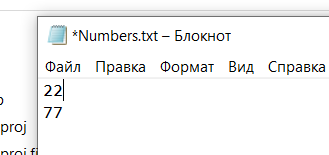
}

cout << -1 << endl;

return 0;

}





Дан файл целых чисел. Создать новый файл, содержащий те же элементы, что и исходный файл, но в обратном порядке.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

string in\_file\_name, out\_file\_name;

cout << "Enter the name of the input file: ";

cin >> in\_file\_name;

cout << "Enter the name of the output file: ";

cin >> out\_file\_name;

ifstream in\_file(in\_file\_name);

ofstream out\_file(out\_file\_name);

if (!in\_file.is\_open()) {

cout << "Error opening input file." << endl;

return 1;

}

vector<int> numbers;

int number;

while (in\_file >> number) {

numbers.push\_back(number);

}

if (!out\_file.is\_open()) {

cout << "Error opening output file." << endl;

return 1;

}

for (int i = numbers.size() - 1; i >= 0; i--) {

out\_file << numbers[i] << endl;

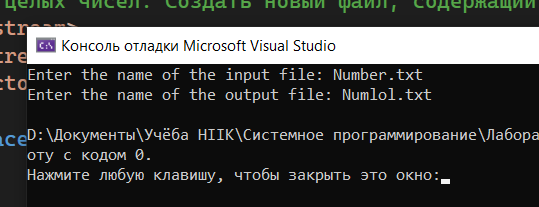
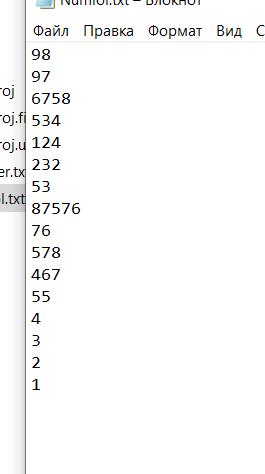
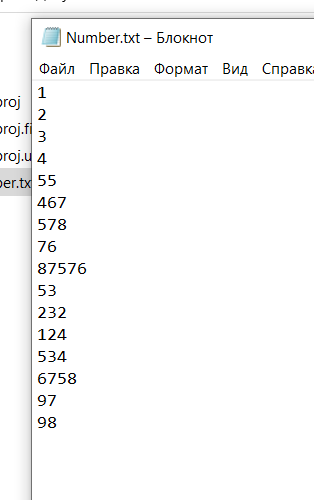
}

in\_file.close();

out\_file.close();

return 0;

}



Дан файл вещественных чисел. Создать два новых файла, первый из которых содержит элементы исходного файла с нечетными номерами(1, 3, …), а второй — с четными(2, 4, …).

#include <iostream>

#include <fstream>

using namespace std;

int main() {

string inputFileName, oddFileName, evenFileName;

cout << "Enter the name of the input file: ";

cin >> inputFileName;

cout << "Enter the name of the first output file (for odd numbers): ";

cin >> oddFileName;

cout << "Enter the name of the second output file (for even numbers): ";

cin >> evenFileName;

ifstream inputFile(inputFileName);

if (!inputFile) {

cout << "Error opening input file" << endl;

return 1;

}

ofstream oddFile(oddFileName);

ofstream evenFile(evenFileName);

int count = 0;

double number;

while (inputFile >> number) {

if (count % 2 == 0) {

evenFile << number << endl;

}

else {

oddFile << number << endl;

}

count++;

}

inputFile.close();

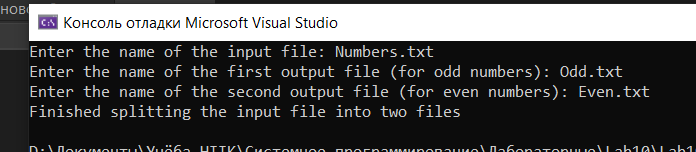
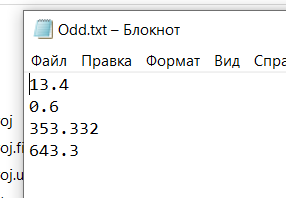
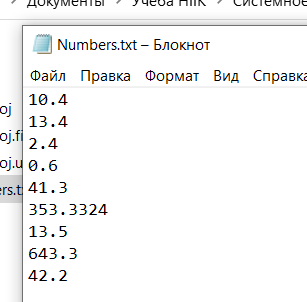
oddFile.close();

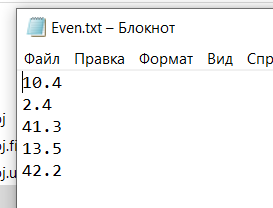
evenFile.close();

cout << "Finished splitting the input file into two files" << endl;

return 0;

}





Дан файл целых чисел. Создать два новых файла, первый из которых содержит положительные числа из исходного файла(в обратном порядке), а второй — отрицательные(также в обратном порядке). Если положительные или отрицательные числа в исходном файле отсутствуют, то соответствующий результирующий файл оставить пустым.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

string fileName;

cout << "Enter the name of the file: ";

cin >> fileName;

ifstream in(fileName);

if (!in.is\_open()) {

cout << "-1" << endl;

return 0;

}

vector<int> positiveNumbers, negativeNumbers;

int number;

while (in >> number) {

if (number > 0) {

positiveNumbers.push\_back(number);

}

else if (number < 0) {

negativeNumbers.push\_back(number);

}

}

in.close();

ofstream outPositive(fileName + "\_positive.txt");

if (!positiveNumbers.empty()) {

for (int i = positiveNumbers.size() - 1; i >= 0; i--) {

outPositive << positiveNumbers[i] << endl;

}

}

outPositive.close();

ofstream outNegative(fileName + "\_negative.txt");

if (!negativeNumbers.empty()) {

for (int i = negativeNumbers.size() - 1; i >= 0; i--) {

outNegative << negativeNumbers[i] << endl;

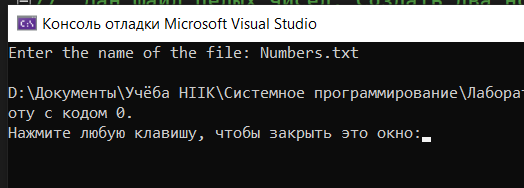
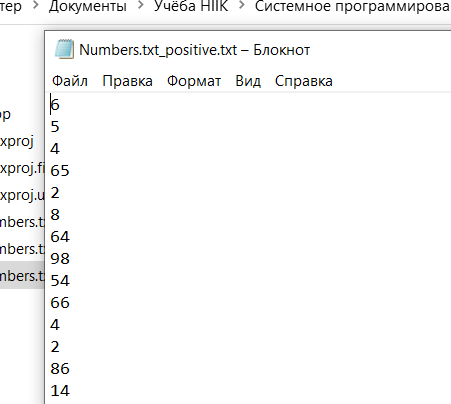
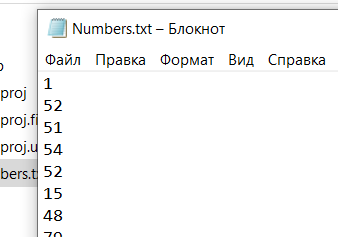
}

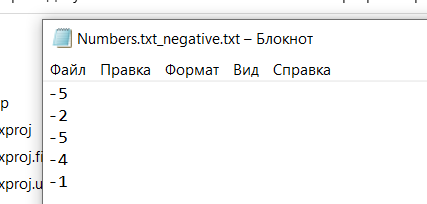
}

outNegative.close();

return 0;

}





Дан файл вещественных чисел. Найти среднее арифметическое его элементов.

#include <fstream>

#include <iostream>

using namespace std;

int main() {

double sum = 0, number, count = 0;

ifstream inFile("Numbers.txt"); // открытие файла для чтения

if (!inFile) { // проверка на успешное открытие файла

cout << "Could not open the file" << endl;

return 1;

}

while (inFile >> number) { // чтение всех чисел из файла

sum += number;

count++;

}

inFile.close(); // закрытие файла

if (count == 0) { // проверка на пустой файл

cout << "The file is empty" << endl;

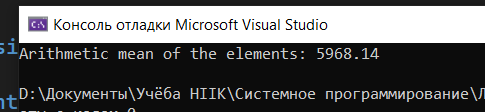
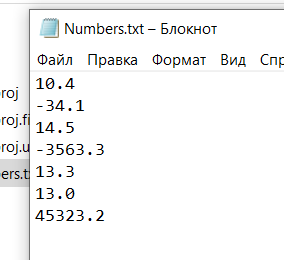
return 1;

}

cout << "Arithmetic mean of the elements: " << sum / count << endl;

return 0;

}



Дан файл вещественных чисел. Найти сумму его элементов с четными номерами.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ifstream inputFile("Numbers.txt");

double sum = 0.0, num;

int counter = 0;

while (inputFile >> num) {

if (counter % 2 == 0) {

sum += num;

}

counter++;

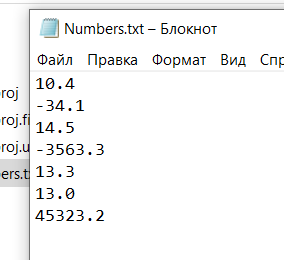
}

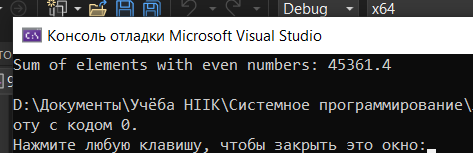
inputFile.close();

cout << "Sum of elements with even numbers: " << sum << endl;

return 0;

}





Дан файл вещественных чисел. Найти его первый локальный минимум (локальным минимумом называется элемент, который меньше своих соседей).

#include <fstream>

#include <iostream>

int main() {

std::ifstream input\_file("Numbers.txt");

double prev, curr, next;

if (input\_file >> prev) {

while (input\_file >> curr) {

if (input\_file >> next) {

if (prev > curr && curr < next) {

std::cout << "The first local minimum is " << curr << std::endl;

return 0;

}

prev = curr;

}

else {

break;

}

}

std::cout << "No local minimum found." << std::endl;

}

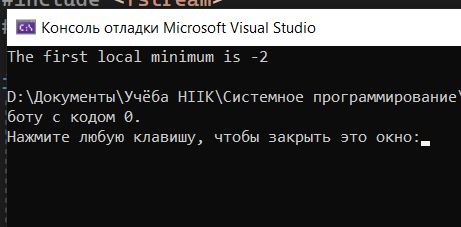
else {

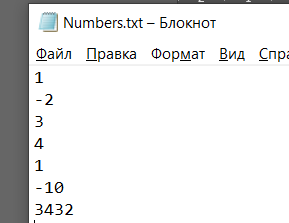
std::cout << "Input file is empty." << std::endl;

}

return 0;

}





Дан файл вещественных чисел. Поменять в нем местами минимальный и максимальный элементы.

#include <iostream>

#include <fstream>

#include <algorithm>

using namespace std;

int main() {

ifstream input\_file("Numbers.txt");

ofstream output\_file("Num.txt");

double num, min\_val, max\_val;

int min\_pos, max\_pos;

input\_file >> num;

min\_val = max\_val = num;

min\_pos = max\_pos = 0;

int pos = 0;

while (input\_file >> num) {

++pos;

if (num < min\_val) {

min\_val = num;

min\_pos = pos;

}

if (num > max\_val) {

max\_val = num;

max\_pos = pos;

}

}

input\_file.close();

input\_file.open("Numbers.txt");

pos = 0;

while (input\_file >> num) {

if (pos == min\_pos) {

output\_file << max\_val << endl;

}

else if (pos == max\_pos) {

output\_file << min\_val << endl;

}

else {

output\_file << num << endl;

}

++pos;

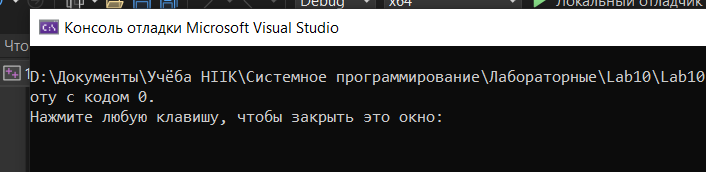
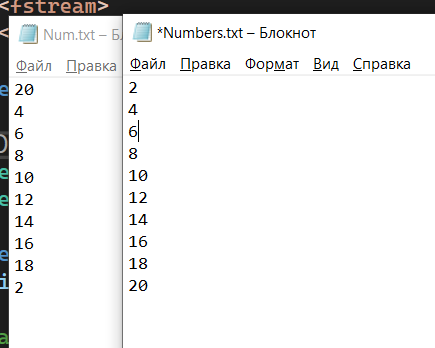
}

input\_file.close();

output\_file.close();

return 0;

}



Дан файл целых чисел с элементами A1, A2, …, AN (N — количество элементов в файле).Заменить исходное расположение его элементов на следующее : A1, AN, A2, AN−1, A3, … .

#include <iostream>

#include <fstream>

#include <vector>

int main() {

std::ifstream in("Number.txt");

std::ofstream out("Num.txt");

std::vector<int> arr;

int n;

in >> n;

int x;

while (in >> x) {

arr.push\_back(x);

}

for (int i = 0; i < arr.size() / 2; i++) {

out << arr[i] << " " << arr[arr.size() - i - 1] << " ";

}

if (arr.size() % 2 == 1) {

out << arr[arr.size() / 2];

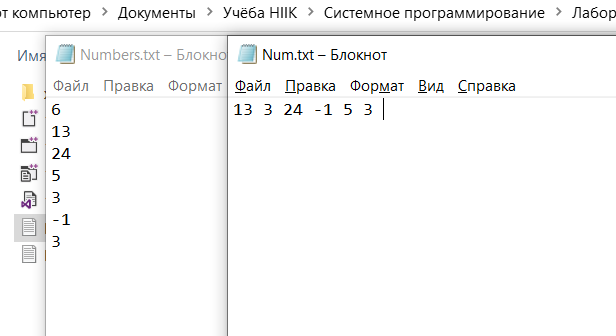
}

in.close();

out.close();

return 0;

}



Дан файл вещественных чисел. Заменить в файле каждый элемент, кроме начального и конечного, на его среднее арифметическое с предыдущим и последующим элементом.

#include <fstream>

#include <iostream>

using namespace std;

int main() {

ifstream inputFile("Number.txt");

ofstream outputFile("Num.txt");

if (!inputFile.is\_open() || !outputFile.is\_open()) {

cout << "Error opening file" << endl;

return 1;

}

double prev, current, next;

inputFile >> prev;

outputFile << prev << endl;

while (inputFile >> current) {

inputFile >> next;

outputFile << (prev + current + next) / 2 << endl;

prev = current;

}

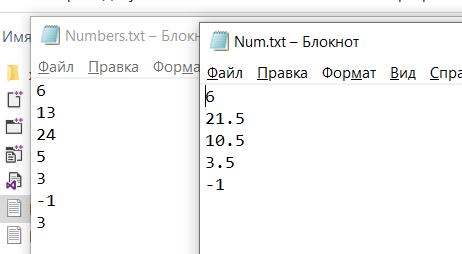
outputFile << current << endl;

inputFile.close();

outputFile.close();

return 0;

}



Дан файл целых чисел, содержащий более 5 элементов. Уменьшить его размер до 5 элементов, удалив из файла необходимое количество конечных элементов.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

int num, count = 0;

int arr[100];

ifstream fin;

fin.open("Numbers.txt");

while (fin >> num) {

arr[count] = num;

count++;

}

fin.close();

ofstream fout;

fout.open("Num.txt");

int n = count > 5 ? 5 : count;

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

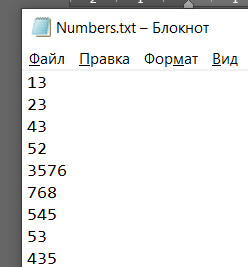
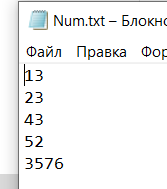
fout << arr[i] << endl;

}

fout.close();

return 0;

}

Дан файл целых чисел. Удалить из него все элементы с четными номерами.

#include <iostream>

#include <fstream>

#include <vector>

int main() {

std::ifstream input("Numbers.txt");

std::ofstream output("Num.txt");

std::vector<int> numbers;

int x;

while (input >> x) {

numbers.push\_back(x);

}

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

if (i % 2 != 0) {

output << numbers[i] << ' ';

}

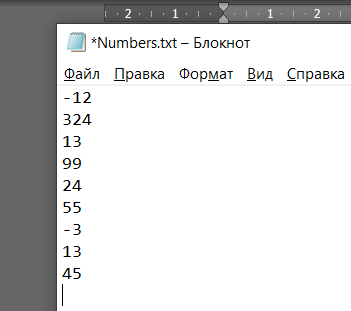
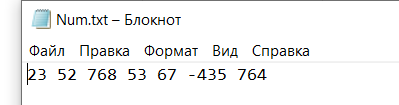
}

input.close();

output.close();

return 0;

}

Дан файл целых чисел. Удвоить его размер, записав в конец файла все его исходные элементы(в том же порядке).

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main()

{

vector<int> numbers;

int x;

ifstream inputFile("Numbers.txt");

if (!inputFile)

{

cout << "File not found" << endl;

return 1;

}

while (inputFile >> x)

{

numbers.push\_back(x);

}

inputFile.close();

ofstream outputFile("Num.txt", ios::app);

for (int i = 0; i < numbers.size(); i++)

{

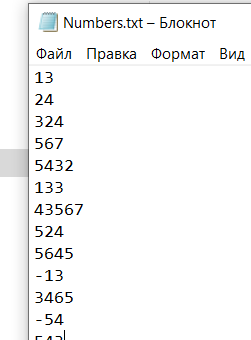
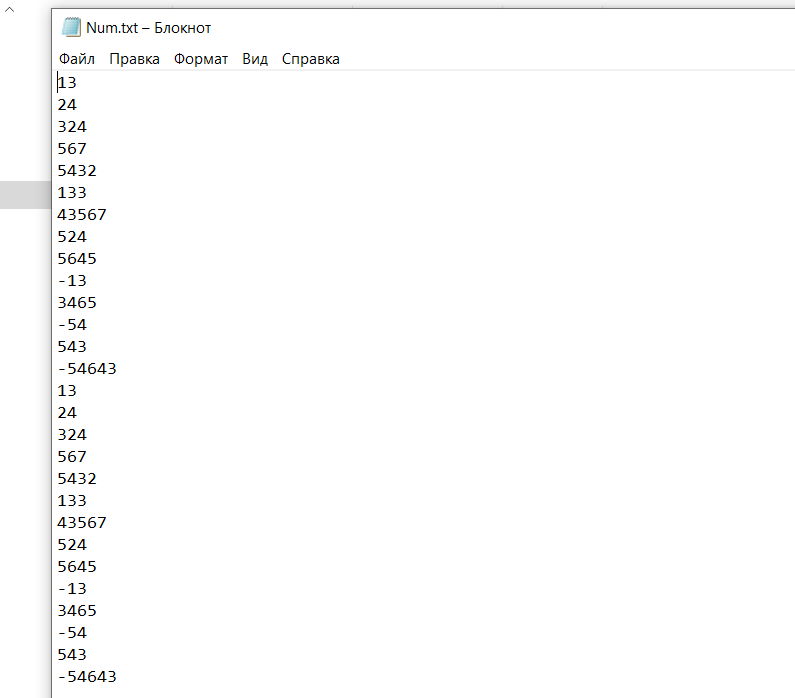
outputFile << numbers[i] << endl;

}

outputFile.close();

return 0;

}

Дан файл целых чисел. Продублировать в нем все элементы с нечетными номерами.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

vector<int> numbers;

int x;

ifstream input("Number.txt");

if (!input.is\_open()) {

cerr << "Error: Unable to open the file." << endl;

return 1;

}

while (input >> x) {

numbers.push\_back(x);

}

input.close();

ofstream output("Num.txt");

if (!output.is\_open()) {

cerr << "Error: Unable to open the file." << endl;

return 1;

}

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

output << numbers[i] << endl;

if (i % 2 != 0) {

output << numbers[i] << endl;

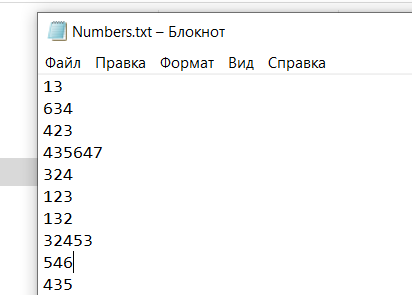
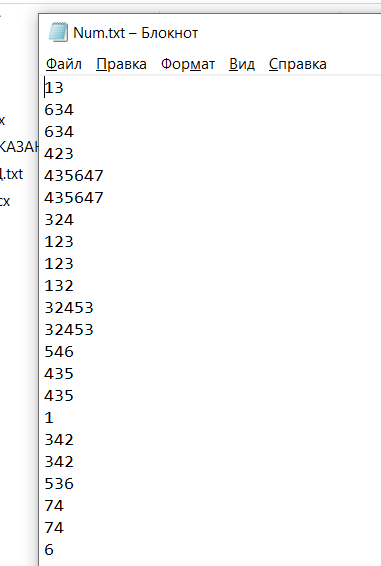
}

}

output.close();

return 0;

}

Дан файл целых чисел. Продублировать в нем все числа, принадлежащие диапазону 5 – 10.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

int main() {

ifstream input\_file("Numbers.txt");

ofstream output\_file("Num.txt");

vector<int> numbers;

int number;

while (input\_file >> number) {

numbers.push\_back(number);

}

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

if (numbers[i] >= 5 && numbers[i] <= 10) {

output\_file << numbers[i] << endl;

output\_file << numbers[i] << endl;

}

else {

output\_file << numbers[i] << endl;

}

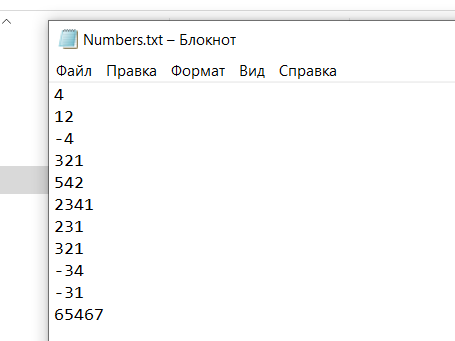
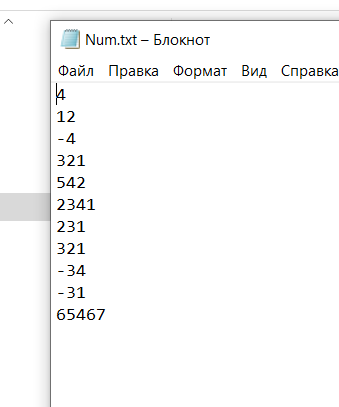
}

input\_file.close();

output\_file.close();

return 0;

}

Дан файл целых чисел. Заменить в нем каждый элемент с четным номером на два нуля.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ifstream inputFile("Numbers.txt");

ofstream outputFile("Num.txt");

int number;

int count = 0;

while (inputFile >> number) {

count++;

if (count % 2 == 0) {

outputFile << "0 0" << endl;

}

else {

outputFile << number << endl;

}

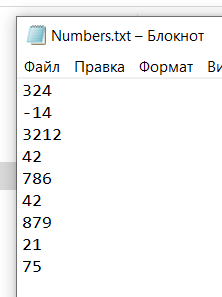
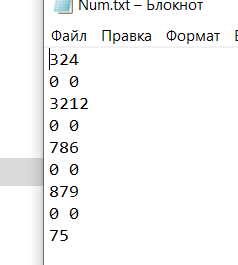
}

inputFile.close();

outputFile.close();

return 0;

}

Дан файл целых чисел. Заменить в нем каждое положительное число на три нуля.

#include <fstream>

#include <iostream>

int main() {

std::ifstream inputFile("Numbers.txt");

std::ofstream outputFile("Num.txt");

int number;

while (inputFile >> number) {

if (number > 0) {

outputFile << "0 0 0 ";

}

else {

outputFile << number << " ";

}

}

inputFile.close();

outputFile.close();

return 0;

}

