**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 №11

по МДК 01.01 Системное программирование

на тему

Нетипизированные файлы. Работа с нетипизированными 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>

using namespace std;

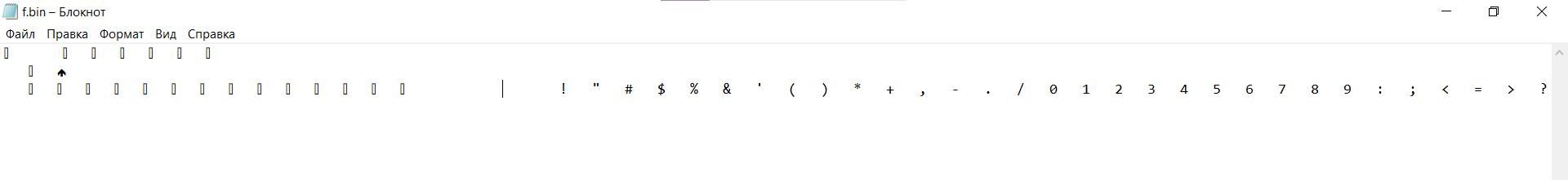
iostream — заголовочный файл с классами, функциями и переменными для организации ввода-вывода.

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

В методе main опишем использования русского алфавита.

setlocale(LC\_ALL, "Russian");

В ходе работы нет смысла скриншотить выходные файлы, так как там бинарная информация:



Составить программу, которая создает нетипизированный файл из 100 чисел и выводит на экран k-ый элемент.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

ofstream file("f", ios::out | ios::binary);

for (int i = 1; i <= 100; i++) {

file.write(reinterpret\_cast<const char\*>(&i), sizeof(i));

}

file.close();

int k;

cout << "Введите к:\t";

cin >> k;

ifstream infile("f", ios::in | ios::binary);

infile.seekg((k - 1) \* sizeof(int));

int num;

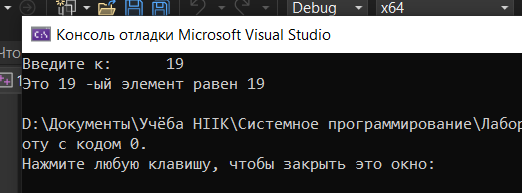
infile.read(reinterpret\_cast<char\*>(&num), sizeof(num));

infile.close();

cout << "Это " << k << " -ый элемент равен " << num << endl;

return 0;

}



Составить программу, которая создает копию элементов нетипизированного файла f и помещает в файл g.

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

ifstream infile("f", ios::in | ios::binary);

if (!infile) {

cerr << "Не могу найти файл f" << endl;

return 1;

}

ofstream outfile("g", ios::out | ios::binary);

if (!outfile) {

cerr << "Не могу открыть файл g" << endl;

return 1;

}

char buffer[1024];

while (infile.read(buffer, sizeof(buffer))) {

outfile.write(buffer, infile.gcount());

}

outfile.write(buffer, infile.gcount());

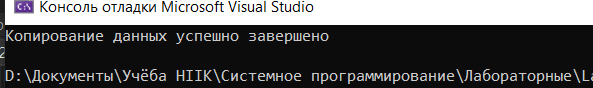
infile.close();

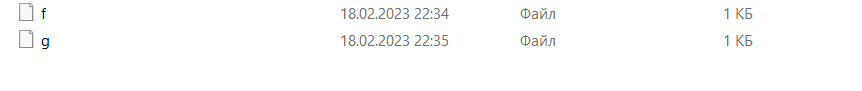
outfile.close();

cout << "Копирование данных успешно завершено" << endl;

return 0;

}





Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет среднее арифметическое элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int sum = 0, count = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

sum += num;

count++;

}

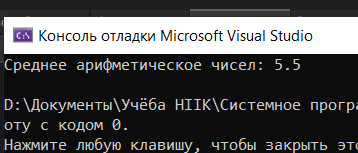
infile.close();

double average = static\_cast<double>(sum) / count;

cout << "Среднее арифметическое чисел: " << average << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет сумму элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int sum = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

sum += num;

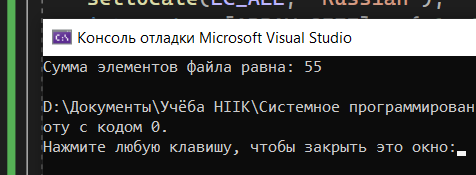
}

infile.close();

cout << "Сумма элементов файла равна: " << sum << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет произведение элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int product = 1;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

product \*= num;

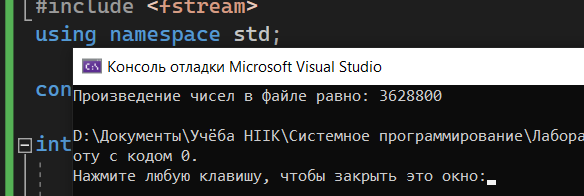
}

infile.close();

cout << "Произведение чисел в файле равно: " << product << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет количество чётных элементов файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int count = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

if (num % 2 == 0) {

count++;

}

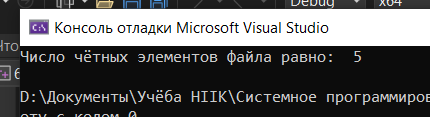
}

infile.close();

cout << "Число чётных элементов файла равно: " << count << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет количество элементов меньше числа К файла.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int count = 0;

int k;

cout << "Введите k:\t";

cin >> k;

cout << endl;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

if (num < k) {

count++;

}

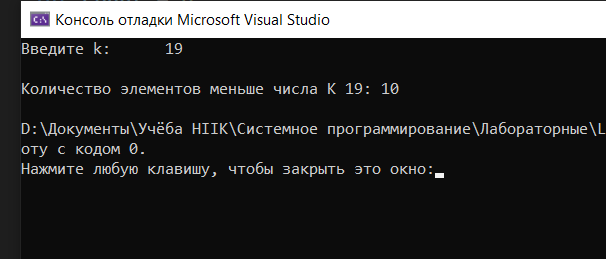
}

infile.close();

cout << "Количество элементов меньше числа K " << k << ": " << count << endl;

return 0;

}



Составить программу, которая создает массив целых чисел и записывает его в нетипизированный файл, а также вычисляет среднее арифметическое элементов файла и сумму элементов меньше среднего арифметического.

#include <iostream>

#include <fstream>

using namespace std;

const int ARRAY\_SIZE = 10;

int main() {

setlocale(LC\_ALL, "Russian");

int numbers[ARRAY\_SIZE] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

ofstream outfile("numbers", ios::out | ios::binary);

outfile.write(reinterpret\_cast<const char\*>(&numbers), sizeof(numbers));

outfile.close();

ifstream infile("numbers", ios::in | ios::binary);

int sum = 0;

int num;

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

sum += num;

}

infile.close();

double average = static\_cast<double>(sum) / ARRAY\_SIZE;

int sum\_less\_than\_average = 0;

infile.open("numbers", ios::in | ios::binary);

while (infile.read(reinterpret\_cast<char\*>(&num), sizeof(num))) {

if (num < average) {

sum\_less\_than\_average += num;

}

}

infile.close();

cout << "Среднее арифметическое чисел в файле : " << average << endl;

cout << "Сумма чисел меньше среднего арифметического: " << sum\_less\_than\_average << endl;

return 0;

}

