Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.10**

Дисциплина: «Основы теории алгоритмов и структуры данных»  
Тема: Сохранение данных в файле с использованием потоков

Вариант 14

Выполнил:

студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2021

**Цель работы**

1. Создание консольного приложения, состоящего из нескольких файлов.

2. Разработка программы, в которой данные сохраняются в файле, корректируются и выводятся из файла на печать. Работа с файлом осуществляется с использованием потоковых классов.

**Постановка задачи**

1. Создать пользовательский класс с минимальной функциональностью.
2. Написать функцию для создания объектов пользовательского класса и сохранения их в файле.
3. Написать функцию для чтения и просмотра объектов из файла.
4. Написать функцию для удаления объектов из файла.
5. Написать функцию для добавления объектов в файл.
6. Написать функцию для изменения объектов в файле.
7. Перегрузить необходимые операции.
8. Для вызова функций в основной программе предусмотреть меню.
9. Задание варианта: создать класс Pair. Реализовать операции сравнения и вычитания константы из пары. Удалить все записи из заданного пользователем интервала. Увеличить все записи с заданным значением в два раза. Добавить несколько записей в начало файла.

**Анализ задачи**

1. Описание классов:

class Para{

int a;

float b;

int \*u;

public:

Para();

Para(int c, float d);

Para(Para &p);

friend void Init(Para& p, int c, float d);

void Show();

void Input();

int RetA();

float RetB();

Para& operator =(Para &p);

friend istream& operator >>(istream &ist, Para &p);

friend ostream& operator <<(ostream &os, Para &p);

friend ifstream& operator >>(ifstream &ist, Para&p);

friend ofstream& operator <<(ofstream &ost, Para&p);

bool operator !=(Para &p);

bool operator ==(Para &p);

Para& operator -(int value);

Para& operator -(float value);

Para& operator \*(int i);

~Para();

};

2. Определение компонентных функций:  
Para::Para(){

a=0;

b=0.0F;

u=NULL;

}

Para::Para(int c, float d){

a=c;

b=d;

u=NULL;

}

Para::Para(Para &p){

a=p.RetA();

b=p.RetB();

u=NULL;

}

Para::~Para(){

delete u;

}

int Para::RetA() {return a;}

float Para::RetB() {return b;}

void Para::Input(){

cout << "a: "; cin >> a;

cout << "b: "; cin >> b;

}

void Para::Show(){cout << "Numbers: " << a << ":" << b << endl;}

Para& Para::operator =(Para &p){

a=p.RetA();

b=p.RetB();

return \*this;

}

bool Para::operator !=(Para &p){

if (a!=p.RetA()&&b!=p.RetB()) return 1;

else return 0;

}

bool Para::operator ==(Para &p){

if (a==p.RetA()&&b==p.RetB()) return 1;

else return 0;

}

Para& Para::operator \*(int i) {a\*=i;b\*=i;return \*this;}

Para& Para::operator -(int value) {a-=value;return \*this;}

Para& Para::operator -(float value) {b-=value;return \*this;}

3. Определение глобальных функций:  
void Init(Para &p, int c, float d){

p.a=c;

p.b=d;

}

ifstream& operator >>(ifstream &ist, Para &p){

int cel;

char sim;

float drob;

ist >> cel >> sim >> drob;

Init(p,cel,drob);

return ist;

}

ofstream& operator <<(ofstream &ost, Para &p){

ost << p.RetA() << ':' << p.RetB() << ' ';

}

void Infile(){

Para para;

para.Input();

ifstream ifile("file.txt");

char getter[100];

ifile.getline(getter,100);

ofstream ofile("file.txt", ios::ate);

ofile << getter;

ofile << para;

ofile.close();

}

void Fromfile(){

Para para;

bool esho=1;

ifstream ifile("file.txt");

while(esho){

ifile >> para;

para.Show();

cout << "Continue? (1/0) " << endl; cin >> esho;

}

ifile.close();

}

void Remove(){

ifstream ifile("file.txt");

ofstream ofile("file1.txt");

Para para;

char big[60];

int a,b;

cout << "From: "; cin >> a;

cout << "To: "; cin >> b;

b+=2;

int i=1;

while (i<a){

ifile >> para;

ofile << para;

i++;

}

for (int e=a;e<b;e++) ifile.getline(big,50,' ');

while(!ifile.eof()){

ifile >> para;

if(para.RetA()&&para.RetB())

ofile << para;

}

ifile.close();

ofile.close();

ifile.open("file1.txt");

ofile.open("file.txt");

while(!ifile.eof()){

ifile >> para;

if(para.RetA()&&para.RetB())

ofile << para;

}

ifile.close();

ofile.close();

}

void Redaktor(){

ifstream ifile("file.txt");

ofstream ofile("file1.txt");

Para para;

while (!ifile.eof()){

ifile >> para;

para=para\*2;

if(para.RetA()&&para.RetB())

ofile << para;

}

ifile.close();

ofile.close();

ifile.open("file1.txt");

ofile.open("file.txt");

while(!ifile.eof()){

ifile >> para;

if(para.RetA()&&para.RetB())

ofile << para;

}

ifile.close();

ofile.close();

}

4. Функция main():

main(){

ofstream o("file.txt");

o << '\r';

o.close();

int s=1;

bool full=0;

while(s){

cout << "1. Output in file; \n2. Input from file; \n3. Remove elements; \n4. Edit elements; \n5. Close the programm. " << endl;

cin >> s;

switch(s){

case 1: Infile(); full=1; break;

case 2: if(full) Fromfile(); else cout << "File is empty! "; break;

case 3: if(full) Remove(); else cout << "File is empty! "; break;

case 4: if(full) Redaktor(); else cout << "File is empty! "; break;

case 5: s=0;break;

default: cout << "Command if not detected. Repeat the attempt. " << endl; break;

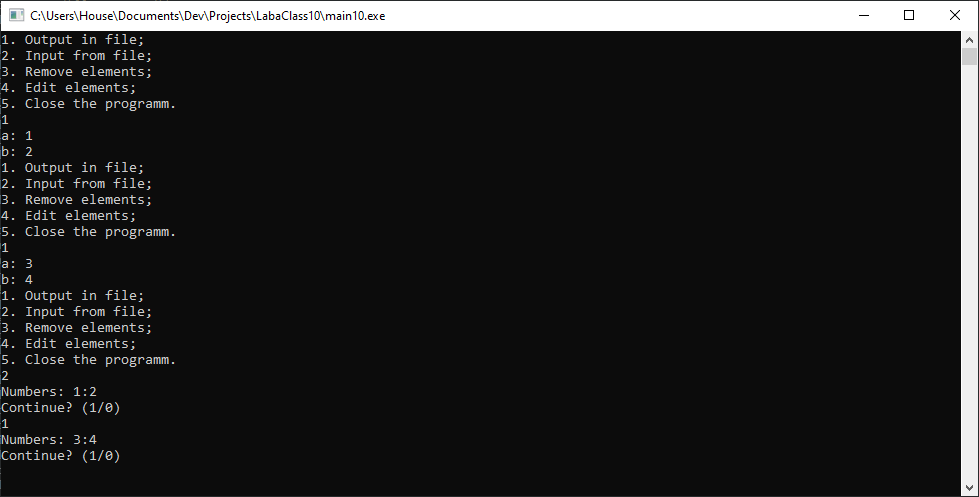
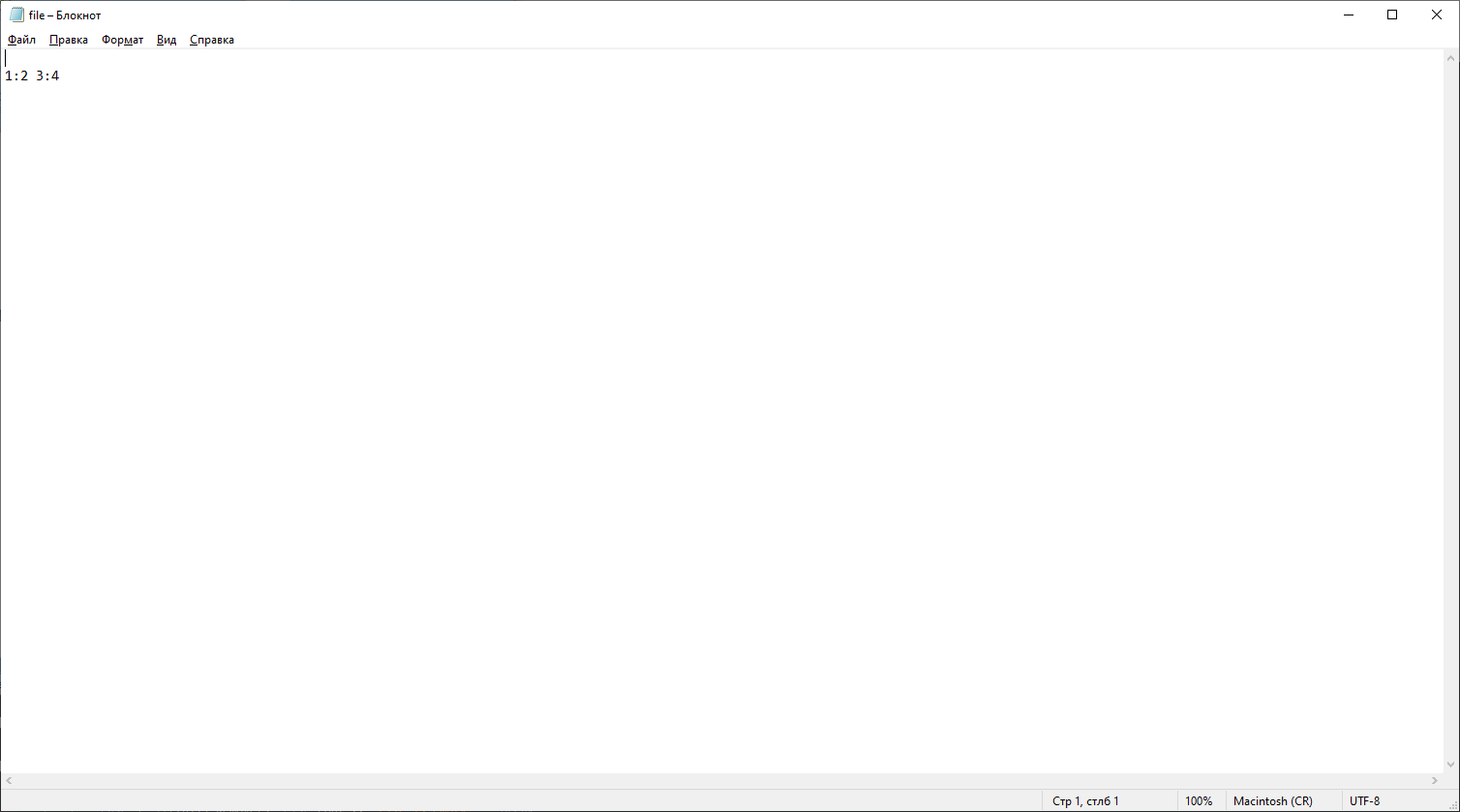
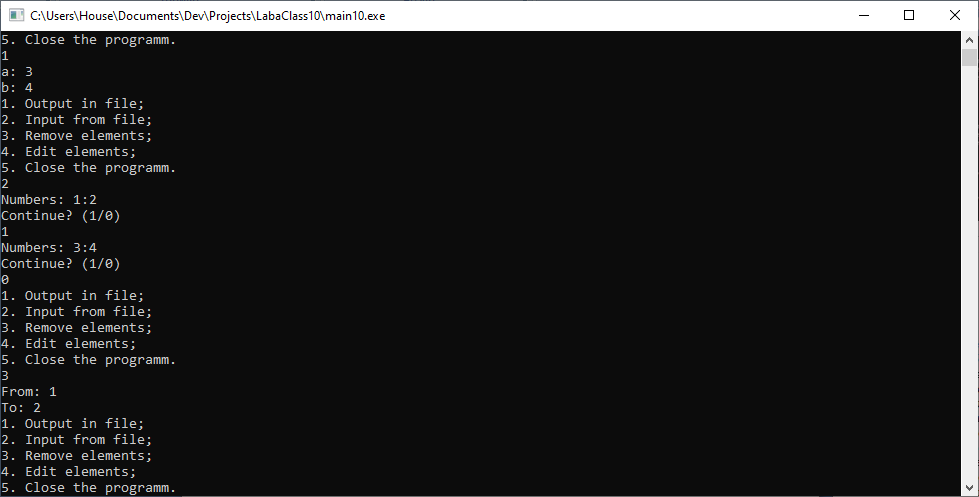
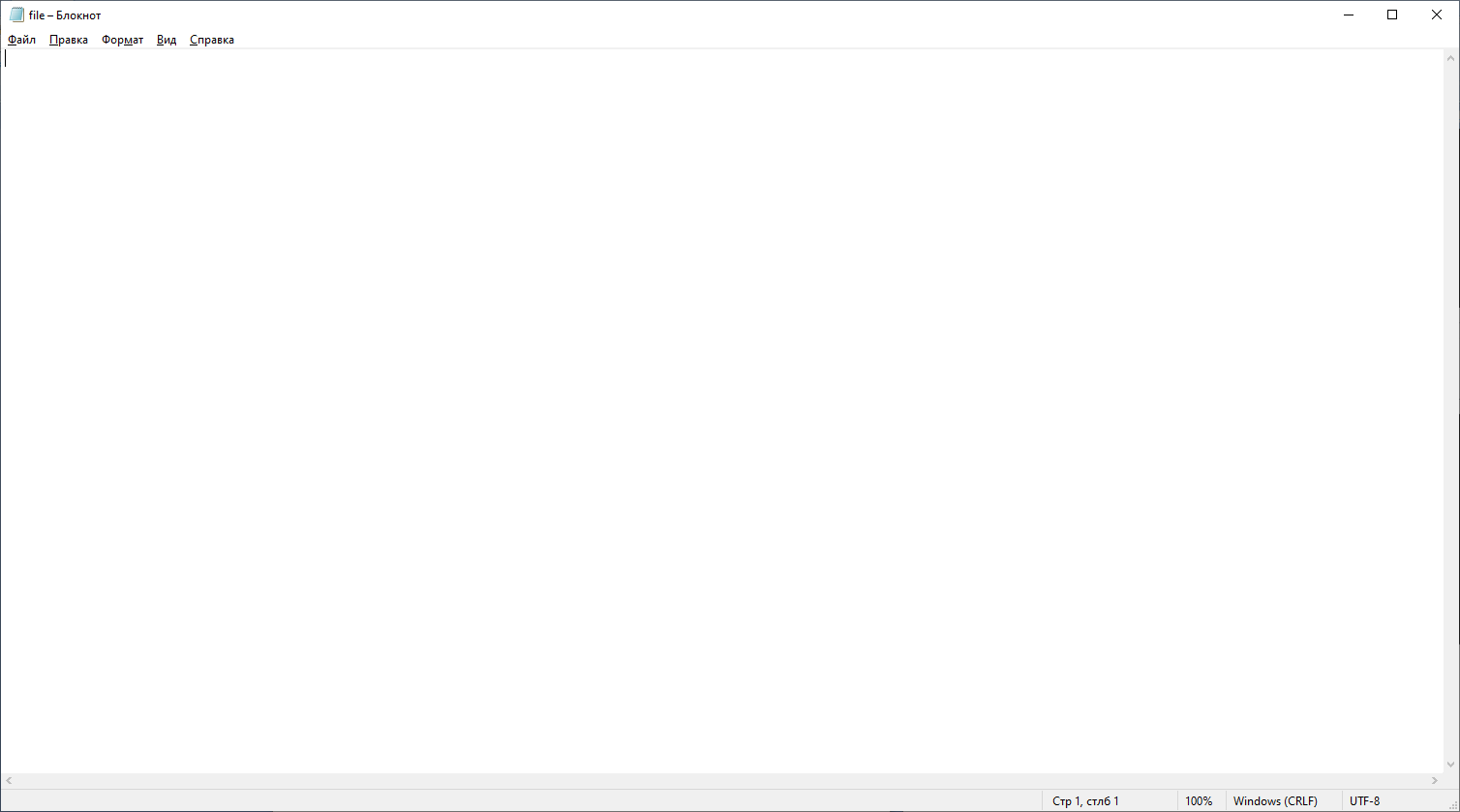
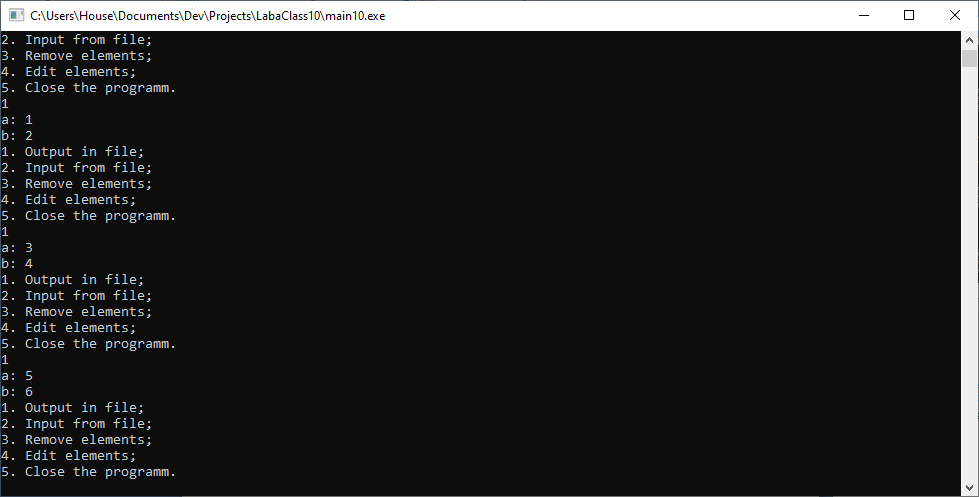
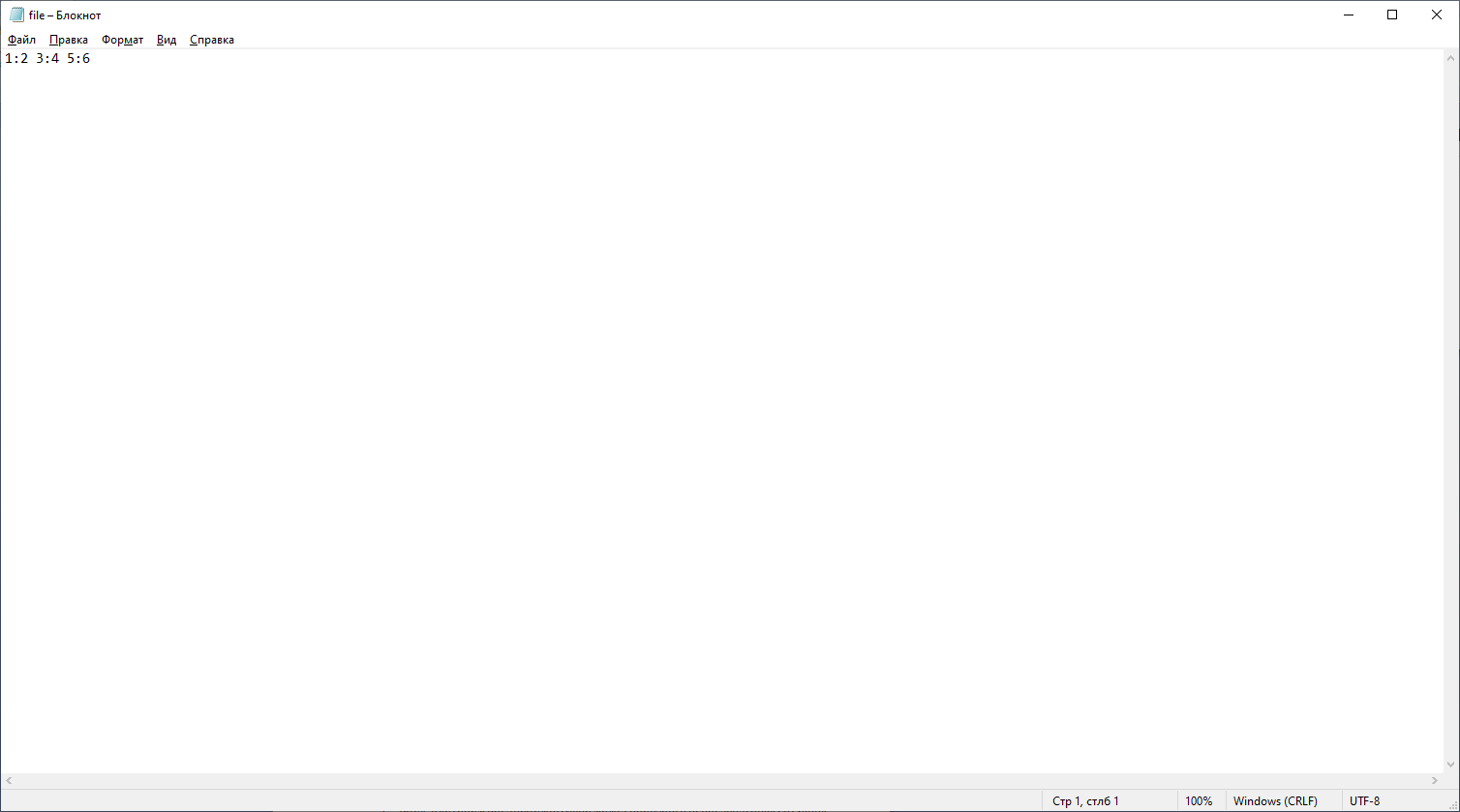
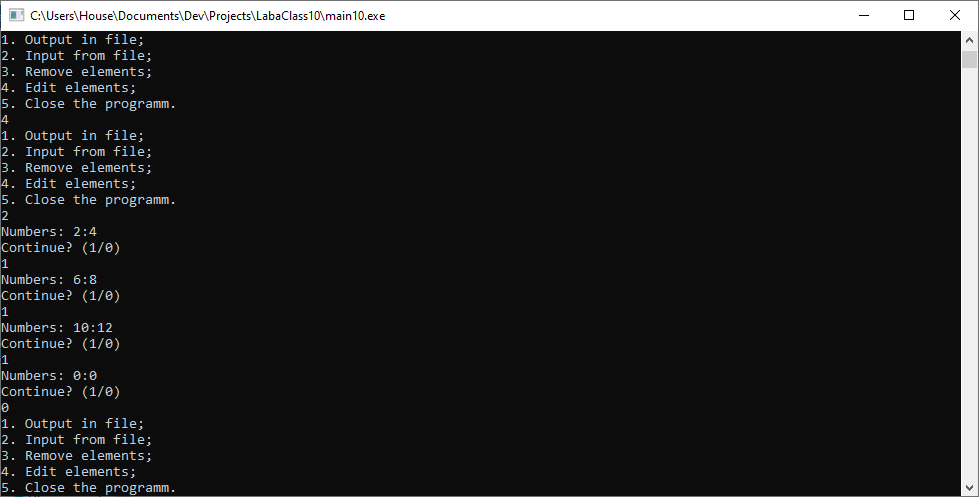
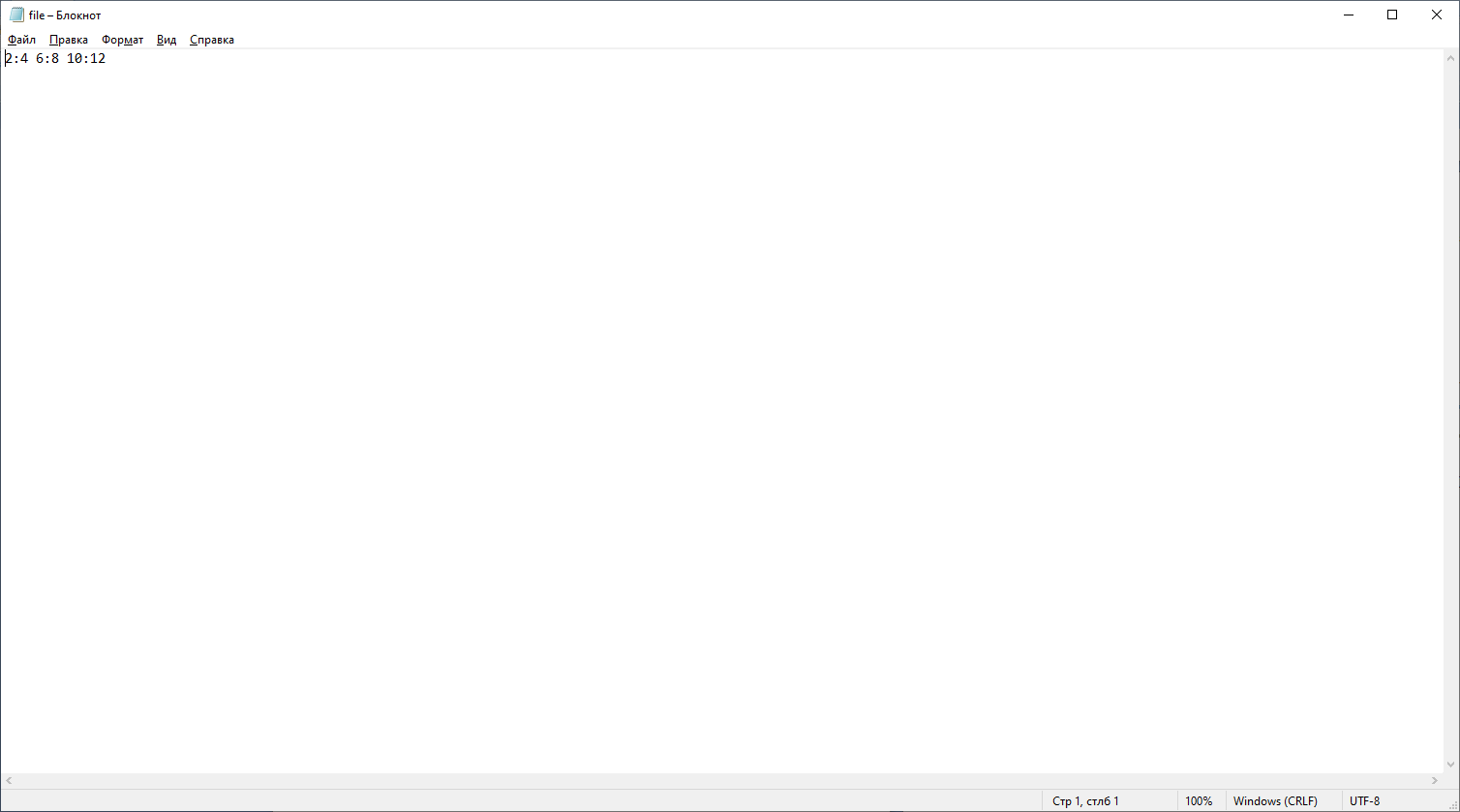
}

}

return 0;

}

**Результаты работы программы**

**       **

**Ответы на вопросы**

1. Последовательность байтов, не зависящая от конкретного устройства.
2. Стандартные, строковые, файловые.
3. Iostream
4. Fstream
5. Stringstream
6. Операция сдвига влево.
7. Операция сдвига вправо.
8. Put, write.
9. Read, get, getline.
10. In, out, trunk, app, ate, binary.
11. App.
12. In.
13. In|out.
14. Out.
15. Для добавления элементов в поток (запись).
16. Для удаления старого содержимого потока (запись).
17. Для удаления старого содержимого потока (чтение и запись).
18. ifstream file(“f.txt”).
19. ofstream file(“f.txt”).
20. fstream file(“f.txt”,ios::in|ios::out);
21. file.getline(s,10);
22. file.put(simvol);
23. Скопировать в другой файл (или в оперативную память) все элементы, которые не подлежат удалению. По желанию заменить содержимое первого файла содержимым второго файла.
24. Скопировать в другой файл все элементы, стоящие до добавляемого, вставить добавляемый элемент, скопировать из первого файла все оставшиеся элементы.
25. Скопировать все элементы из файла в оперативную память, изменить необходимые, очистить файл и вставить в него элементы из оперативной памяти.