**ПРАКТИЧНЕ ЗАВДАННЯ № 10**

**Завдання №1 :**

Напишите программу для вывода предложения: "I hate C++" в поле шириной 30

символов с использованием двоеточия (:) в качестве символа заполнения.

**Код програми:**

#include <iostream>

#include <iomanip>

using namespace std;

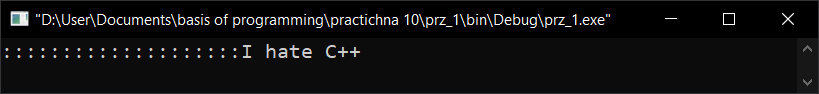
int main() {

cout << setw(30) << setfill(':') << "I hate C++" << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №2 :**

Напишите программу для вывода результата деления 10/3 с четырьмя знаками после десятичной точки. Для этого используйте функции-члены класса ios. Напишите вторую версию этой программы, используя манипуляторы ввода/вывода.

**Код програми:**

#include <iostream>

#include <iomanip>

using namespace std;

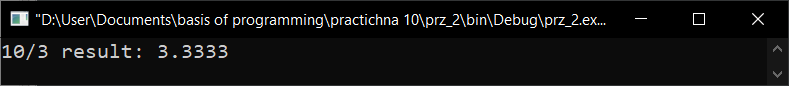
int main() {

cout << "10/3 result: " << setprecision(5) << 10/3. << endl;

return 0;

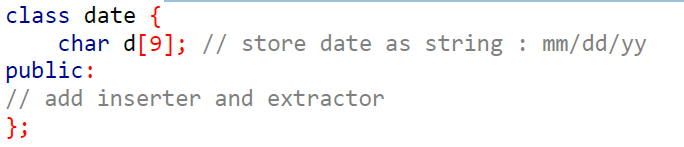
}

**Результати роботи програми (скриншоти)**



**Завдання №3 :**

Дан следующий класс. Создайте для него пользовательские функции ввода и вывода



**Код програми:**

#include <iostream>

using namespace std;

class Date {

char date[9];

public:

friend ostream &operator << (ostream &stream, Date &obj);

friend istream &operator >> (istream &stream, Date &obj);

};

istream &operator >> (istream &stream, Date &obj){

cout << "Enter date dd/mm/yy: ";

cin >> obj.date;

return stream;

}

ostream &operator << (ostream &stream, Date &obj){

stream << "Entered date: " << obj.date << endl;

return stream;

}

int main() {

Date a;

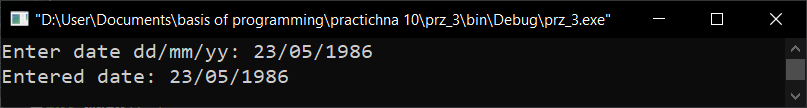
cin >> a;

cout << a;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №4 :**

Создайте свой манипулятор вывода для отображения текущего системного времени и даты. Назовите этот манипулятор tdm().

**Код програми:**

#include <iostream>

#include <ctime>

using namespace std;

ostream &timer(ostream &stream){

time\_t t = time(NULL);

stream << asctime(localtime(&t)) << endl;

return stream;

}

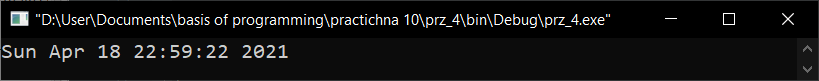
int main() {

cout << timer << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №5 :**

Создайте свой манипулятор вывода sethex(), который осуществляет вывод в шестнадцатиричной системе счисления и устанавливает флаги uppercase и showbase. Кроме того, создайте манипулятор вывода reset(), который отменяет изменения, сделанные манипулятором sethex().

**Код програми:**

#include <iostream>

using namespace std;

ostream &sethex(ostream &stream) {

stream.unsetf(ios:: dec);

stream.setf(ios::hex | ios::uppercase | ios::showbase);

return stream;

}

ostream &reset(ostream &stream) {

stream.unsetf(ios::hex | ios::uppercase | ios::showbase);

stream.setf(ios::dec);

return stream;

}

int main()

{

cout << 100 << endl;

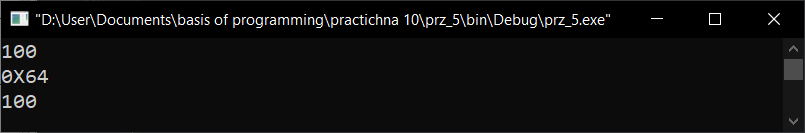
cout << sethex << 100 << endl;

cout << reset << 100 << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №6 :**

Создайте манипулятор ввода skipchar(), который поочередно то считывает, то пропускает каждые 5 символов потока ввода.

**Код програми:**

#include <iostream>

#include <cstring>

#include <iomanip>

using namespace std;

istream &skipchar(istream &stream) {

char ch;

for(int i = 0; i < 3; ++i) {

cin.ignore(5);

for(int j = 0; j < 5; ++j) {

if (ch == ' ')

return stream;

stream >> ch;

cout << ch;

}

cout << endl;

}

return stream;

}

int main() {

string str;

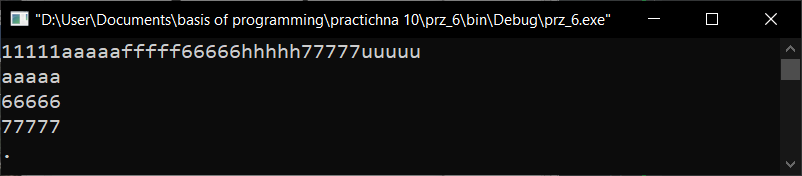
cin >> skipchar >> str;

cin >> str;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №7 :**

Напишите программу для копирования текстового файла. Эта программа должна подсчитывать число копируемых символов и выводить на экран полученный результат. Почему это число отличается от того, которое выводится при просмотре списка файлов каталога?

**Код програми:**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

int main(){

ifstream fin("copy");

char str[100];

cin >> str;

int lenght = strlen(str);

fin.close();

ofstream fout("copy");

cout << str << "-> number of symbols -> " << lenght << endl;

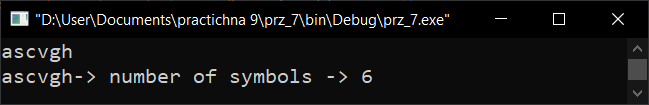
fout << str;

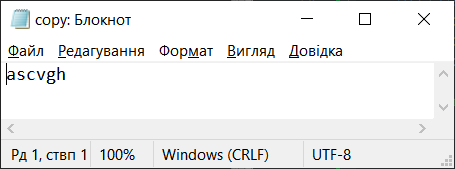
fout.close();

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №8 :**

Напишите программу для записи в файл phones.txt данных типа:

Ivan Tormozenko, +38067 415 5113

Petro Siplusenko, +38068 212 5332

Olena Kompailenko, +38050 202 54517

**Код програми:**

#include <iostream>

#include <fstream>

using namespace std;

int main() {

ifstream fin("phones");

ofstream fout("phones");

if(!fout)

return 1;

fout << "Ivan Tormozenko, +38067 415 5113" << endl;

fout << "Petro Siplusenko, +38068 212 5332" << endl;

fout << "Olena Kompailenko, +38050 202 54517" << endl;

fout.close();

if(!fin.good() && !fin.eof())

return 1;

string str;

while(!fin.eof()) {

getline(fin, str);

cout << str << endl;

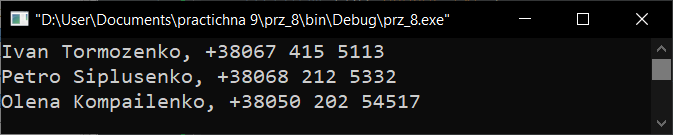
}

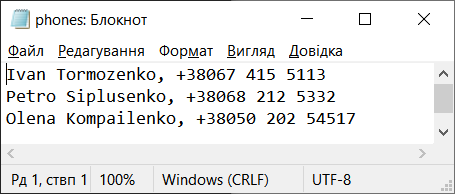
fin.close();

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №9 :**

Напишите программу для подсчета числа слов в файле. Для простоты считайте, что словом является все, имеющее с двух сторон пробелы.

**A)**

**Код програми:**

#include <iostream>

#include <fstream>

using namespace std;

int main() {

string str;

int count = 1;

ofstream out;

out.open("str");

getline(cin, str);

if (out.is\_open()) {

out << str << endl;

}

out.close();

ifstream in("str");

while (!in.eof()) {

getline(in, str);

for (int i = 0; i < str.length(); i++) {

if (str[i] == ' ')

count++;

}

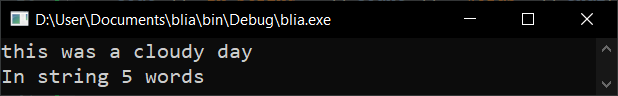
}

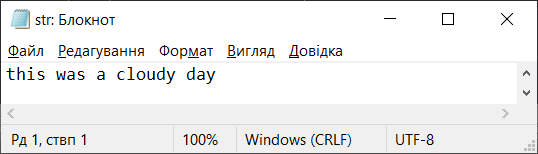
cout << "In string " << count << " words" << endl;

in.close();

return 0;

}





**B)**

#include <iostream>

#include <sstream>

#include <cstring>

using namespace std;

int main() {

string str = "Here is some text";

istringstream instr(str);

string word; int count = 0;

while(instr >> word){

cout << word << ' ';

count++;

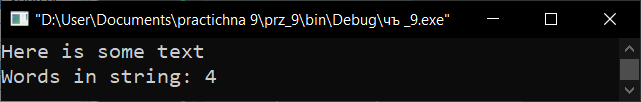
}

cout << endl << "Words in string: " << count << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №10 :**

Измените решения Задач 10.7 и 10.9 так, чтобы в них использовались функции get(), put(), read() и/или write(). Думайте и используйте эти функции там, где они, по вашему мнению, принесут наибольшую пользу.

**Код програми:**

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

int main(){

ifstream fin("copy");

char str[100]; int count = 0;

cin >> str;

int lenght = strlen(str);

fin.close();

ofstream fout("copy");

cout << str << "-> number of symbols -> " << lenght << endl;

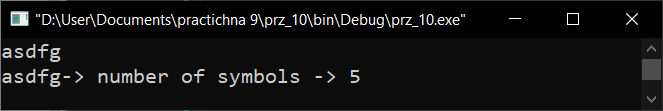
fout.write(str, lenght);

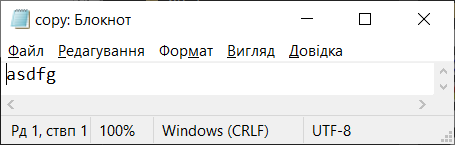
fout.close();

return 0;

}

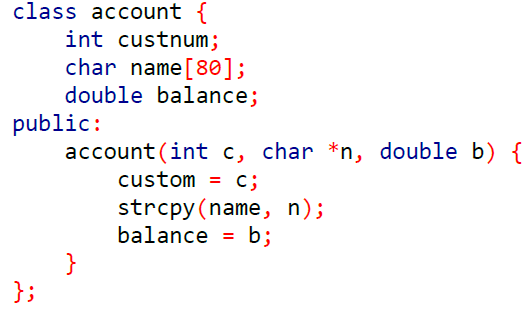
**Результати роботи програми (скриншоти)**





**Завдання №11 :**

Дан следующий класс. Напишите программу для вывода содержимого объекта данного класса в файл:



Подсказка: Для этой цели создайте пользовательскую функцию вывода.

**Код програми:**

#include <iostream>

#include <cstring>

#include <fstream>

using namespace std;

class Account {

int custom;

string name;

double balance;

public:

Account(int c, const string &n, double b) {

custom = c;

name = n;

balance = b;

}

friend ostream &operator <<(ostream &stream, Account &obj);

};

ostream &operator <<(ostream &stream, Account &obj){

stream << "Account:" << endl;

stream << "custom " << obj.custom << endl;

stream << "name " << obj.name << endl;

stream << "balance " << obj.balance << endl;

return stream;

}

int main() {

Account acc(234, "Alex", 87.97);

ofstream out("Account");

if(!out.good() && !out.eof())

exit(1);

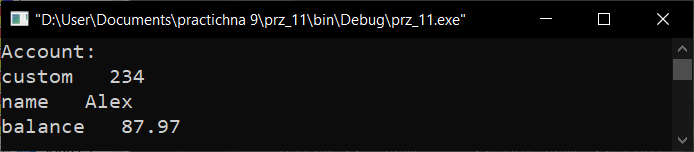
out << acc;

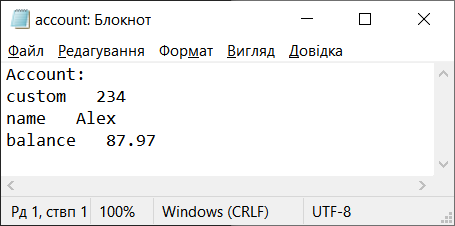
cout << acc;

return 0;

}

**Результати роботи програми (скриншоти):**





**Завдання №12 :**

Перепишите программу из Example 10.11 так, чтобы вместо функции getline() использовать функцию get(). Будет ли отличаться работа программы?

**Код програми:**

#include <iostream>

#include <cstring>

#include <fstream>

using namespace std;

class Account {

int custom;

string name;

double balance;

public:

Account(int c, const string &n, double b) {

custom = c;

name.assign(n);

balance = b;

}

friend ostream &operator <<(ostream &stream, Account &obj);

};

ostream &operator <<(ostream &stream, Account &obj) {

stream << "\nAccount:" << endl;

stream << "custom " << obj.custom << endl;

stream << "name " << obj.name << endl;

stream << "balance " << obj.balance << endl;

return stream;

}

int main() {

char str[80];

int i = 0;

ofstream out("accounts");

if(!out.good() && !out.eof())

exit(1);

ifstream in("accounts");

if(!in.good() && !in.eof())

exit(1);

while(cin.get(str[i])) {

if(str[i] == '\n') {

str[i] = '\0';

break;

}

++i;

}

Account acc(234, str, 87.97);

out << acc;

cout << acc;

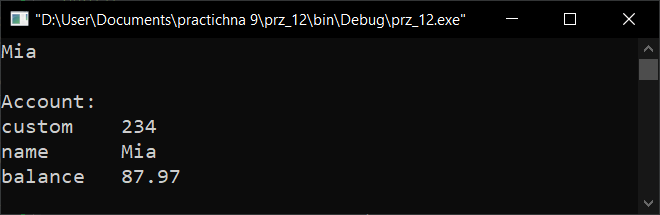
in.close();

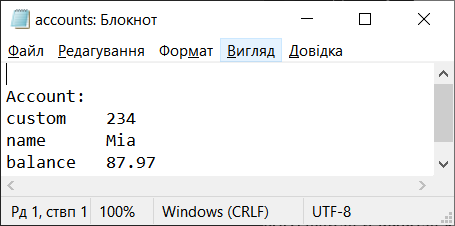
out.close();

return 0;

}

**Результати роботи програми (скриншоти):**





**Завдання №13 :**

Напишите программу для построчного считывания текстового файла и вывода каждой считанной строки на экран. Используйте функцию getline().

**Код програми:**

#include <iostream>

#include <fstream>

using namespace std;

int main(){

ifstream fin("prz\_13");

string str;

while(!fin.eof()){

getline(fin, str, '\n');

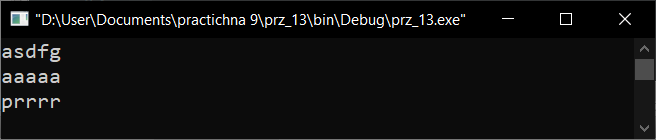
cout << str << endl;

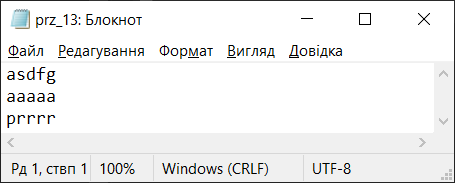
}

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №14 :**

Это задача для размышления. Вам уже известно, что при выполнении вывода данные не сразу записываются на связанное с потоком физическое устройство, а информация временно сохраняется во внутреннем буфере. Только после заполнения буфера его содержимое переписывается на устройство хранения (например, диск). Вызов специальной функции flush() вызывает физическую запись информации на диск до заполнения буфера (в Unit 10 эту функцию мы не рассматривали.) Прототип функции flush(), являющейся членом потоковых классов вывода имеет вид: ostream &flush();

Вызовы функции flush() оправданы при работе в неблагоприятной обстановке (например, в ситуациях, когда часто случаются сбои по питанию). Поразмышляйте о ситуациях, в которых может оказаться полезным вызов функции flush().

**Код програми:**

/\*Функція flush() – функція (маніпулятор), пов’язана з буферизацією даних.

При закриті файлів або перед вводом користувача, буфер неявно очищається, тобто відбувається збереження змін внесених до файлу, але після заповнення буфера. Функція flush() використовується у випадках, коли потрібно явно очистити буфер: коли необхідно вивести користувачу на консоль дані, але можлива втрата даних чи застрягання в буфері \*/

**Завдання №15 :**

Напишите программу для вывода на экран содержимого текстового файла в обратном порядке. Совет: Еще один раз обдумайте задание перед началом программирования. Решение проще, чем может показаться на первый взгляд.

**Код програми:**

#include<iostream>

#include<fstream>

#include<cstring>

using namespace std;

void reverse() {

fstream file\_in;

file\_in.open("reverse", ios::in);

char str[100];

while(!file\_in.eof()) {

file\_in.getline(str, 100);

for (int i = strlen(str) - 1; i >= 0; --i)

cout<<str[i];

cout << endl;

}

}

int main() {

fstream file\_in;

file\_in.open ("reverse", ios::app | ios::in);

char str[100];

cin.getline(str, 100);

file\_in << str << endl;

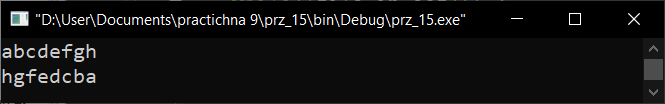
reverse();

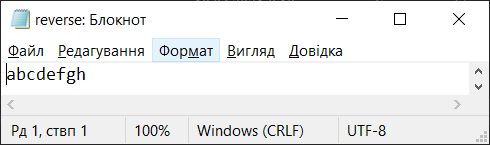
file\_in.close();

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №16 :**

Напишите программу, которая попарно меняет местами символы в текстовом файле. Например, если в файле содержится "1234", то после выполнения программы там должно содержаться "2143". (Для простоты считайте, что в файле содержится четное число символов.)

**Код програми:**

#include<iostream>

#include<fstream>

#include<string.h>

using namespace std;

void swap\_elemens() {

fstream file;

file.open("swap\_elemens", ios::in);

char str[100];

while(!file.eof()) {

file.getline(str, 100);

for (int i = 0; i < strlen(str) ; i += 2)

swap(str[i], str[i + 1]);

cout << str << endl;

cout << endl;

}

}

int main() {

fstream file;

file.open ("swap\_elemens", ios::app);//аргумент режима добавление к концу файла

char str[100];

cin.getline(str, 100);

file << str << endl;

cout << str << endl;

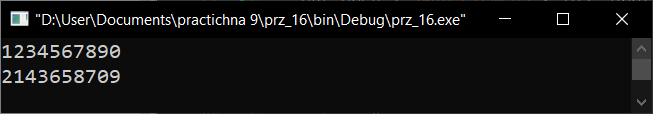
swap\_elemens();

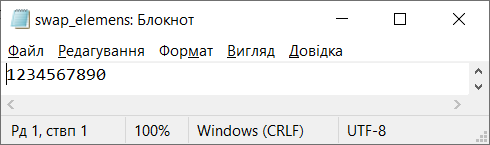
file.close();

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №17 :**

Создайте манипулятор для вывода трех символов табуляции и установки ширины поля равной 20. Продемонстрируйте работу манипулятора.

**Код програми:**

#include <iostream>

using namespace std;

ostream &manip(ostream &stream) {

stream << "\t\t\t";

stream.width(20);

return stream;

}

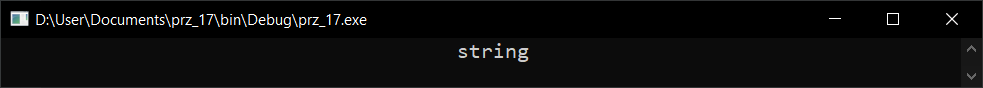
int main() {

cout << manip << "string" << endl;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №18 :**

Создайте манипулятор для ввода, который должен считывать и отбрасывать все неалфавитные символы. При считывании первого алфавитного символа, манипулятор должен возвратить его во входной поток и закончить работу. Назовите манипулятор findalpha.

**Код програми:**

#include <iostream>

#include <cctype>

using namespace std;

istream &findalpha(istream &stream){

char a;

for(int i = 0; !isalpha(a); ++i){

stream >> a;

if(isalpha(a)){

cout << a;

return stream;

}

}

return stream;

}

int main() {

char str[100];

cin >> findalpha >> str;

return 0;

}

**Результати роботи програми (скриншоти)**



**Завдання №19 :**

Напишите программу копирования текстового файла. При копировании измените регистр всех букв.

**Код програми:**

#include <iostream>

#include <fstream>

#include <cctype>

using namespace std;

int main() {

ofstream fileOut("in");

fileOut << "abcHKMNPmn24Tt";

fileOut.close();

ifstream file1("in");

ofstream file2("out");

char ch;

while(file1 >> ch) {

ch = islower(ch) ? toupper(ch) : tolower(ch);

file2.put(ch);

cout << ch;

}

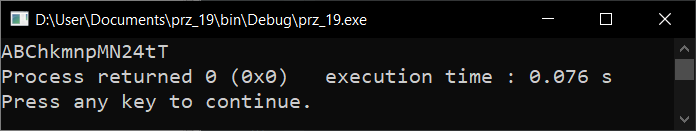
file1.close();

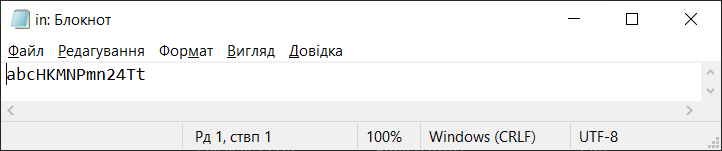
file2.close();

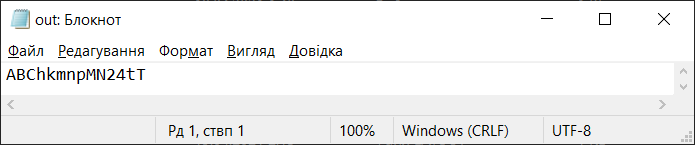
return 0;

}

**Результати роботи програми (скриншоти)**







**Завдання №20:**

Напишите программу, которая считывает текстовый файл, а затем сообщает, сколько раз каждая буква алфавита появляется в файле.

**Код програми:**

#include <iostream>

#include <fstream>

#include <cctype>

using namespace std;

int main() {

string str = "aaabbbc788821";

ofstream("prz\_10") << str;

cout << str << endl;

freopen("prz\_10", "r", stdin);

int cnt[100];

for (int i = 0; i < 100; ++i)

cnt[i] = 0;

for (int i = getchar(); i != EOF; i = getchar())

++cnt[i];

for (int i = 0; i < 100; ++i) {

if (cnt[i] != 0)

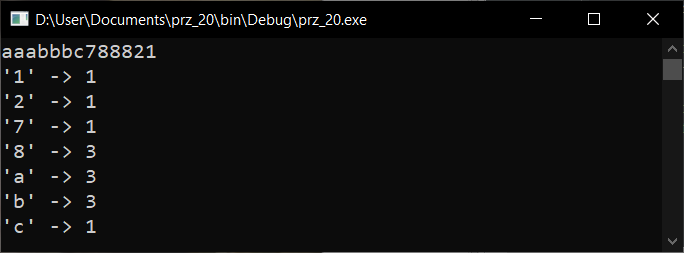
cout << "'" << char(i) << "' -> " << cnt[i] << "\n";

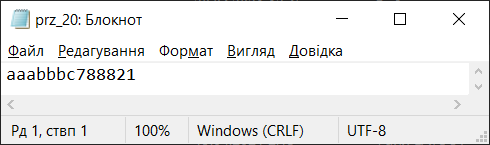
}

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №21 :**

Ниже приведена переработанная версия класса inventory из предыдущей Практической работы. Добавьте функции store() и retrieve(). Затем создайте небольшой файл, содержащий несколько инвентарных записей. Далее, используя произвольный доступ, по номеру записи отобразите на экране информацию об одном из элементов.

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

#define SIZE 40

class inventory {

char item[SIZE]; // name of item

int onhand; // number on hand

double cost; // cost of item

public:

inventory(char\* i, int o, double c) {

strcpy(item, i);

onhand = o;

cost = c;

}

void store(fstream& stream);

void retrieve(fstream& stream);

friend ostream& operator <<(ostream& stream, inventory obj);

friend istream& operator >>(istream& stream, inventory& obj);

};

ostream& operator <<(ostream& stream, inventory obj) {

stream << obj.item << ": " << obj.onhand;

stream << " on hand at $" << obj.cost << '\n';

return stream;

}

istream& operator >>(istream& stream, inventory& obj) {

cout << "Enter item name: ";

stream >> obj.item;

cout << "Enter number on hand: ";

stream >> obj.onhand;

cout << " Enter cost: ";

stream >> obj.cost;

return stream;

}

**Код програми:**

#include <iostream>

#include <fstream>

#include <cstring>

#define SIZE 40

using namespace std;

class Inventory {

string item; // name of item

int onhand; // number on hand

double cost; // cost of item

public:

Inventory(string i, int o, double c) {

item = i;

onhand = o;

cost = c;

}

Inventory() {

item = "";

onhand = 0;

cost = 0;

}

void store(fstream &stream);

void retrieve(fstream &stream);

friend ostream &operator <<(ostream &stream, Inventory &obj);

friend istream &operator >>(istream &stream, Inventory &obj);

};

ostream &operator <<(ostream &stream, Inventory &obj) {

cout << endl;

stream << obj.item << ": " << obj.onhand;

stream << " on hand at $" << obj.cost << endl;

return stream;

}

istream &operator >>(istream &stream, Inventory &obj) {

cout << "Enter item name: ";

stream >> obj.item;

cout << "Enter number on hand: ";

stream >> obj.onhand;

cout << "Enter cost: ";

stream >> obj.cost;

return stream ;

}

void Inventory::store(fstream &stream){

stream << item << endl;

stream << onhand << endl;

stream << cost << endl;

}

void Inventory::retrieve(fstream &stream){

stream >> item;

stream >> onhand;

stream >> cost;

}

int main(){

fstream file("prz\_21", ios::out | ios::binary);

Inventory obj1("Books", 2 , 35);

Inventory obj2;

cout << "Enter second item. " << endl;

cin >> obj2;

if(!file)

return 1;

obj1.store(file);

obj2.store(file);

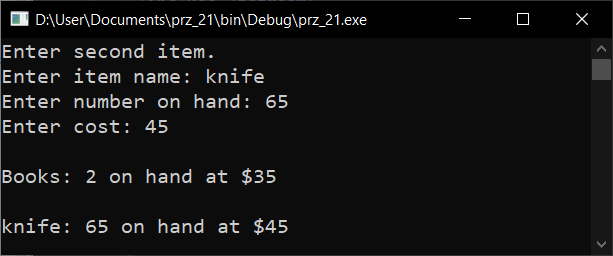
cout << obj1 << obj2;

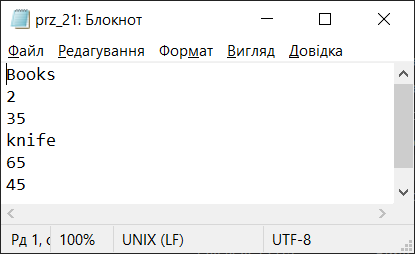
file.close();

return 0;

}

**Результати роботи програми (скриншоти)**





**Завдання №22 :**

Создайте класс stack для хранения символов в файле, а не в массиве.

**Код програми:**

#include <iostream>

#include <fstream>

#define SIZE 5

using namespace std;

class Stack {

private:

FILE\* stack;

fpos\_t position;

public:

Stack(const char\* filename): stack(fopen(filename, "w+")) { }

void push(int ch) {

fputc(ch, stack);

}

int seek() {

return fseek(stack, -1, SEEK\_CUR) + fgetc(stack);

}

int pop() {

return seek() + fseek(stack, -1, SEEK\_CUR);

}

void create() {

char a;

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

cin >> a;

push(a);

}

}

void display(){

for(int i = 0; i < SIZE; ++i)

cout << (char)pop() << ' ';

}

~Stack() {

fclose(stack);

}

};

int main() {

Stack obj = Stack("prz\_22");

obj.create();

obj.display();

return 0;

}

**Результати роботи програми (скриншоти)**

