Івано-Франківський національний технічний університет нафти і газу

Кафедра Комп’ютерних технологій ДНУ

Звіт до лабораторної роботи №2

«Робота зі списками»

з курсу «Алгоритми та структури даних»

Виконала:

ст.гр. ІП-22-4

Томин Є.В.

Перевірив:

Григорчук Л. І.

Івано-Франківськ

2023

**Тема.** Робота із списками (частина V).

**Мета:** Навчитися використовувати операції над списками в Пролог-програмах.

**Виконання роботи.**

1) Вивчити теоретичні відомості;

2) Вивчити практичну частину;

3) Виконати самостійну роботу;

4) Відповісти на контрольні питання.

**Теоретичні відомості**

**Список** - це впорядкований набір об'єктів, що випливають один за одним. Складові списку внутрішньо зв'язані між собою, тому з ними можна працювати і як із групою (списком у цілому), так і як з індивідуальними об'єктами (елементами списку). Ще одне з визначень список - послідовність із довільного числа елементів.

Список є основною структурою даних у Пролозі. Турбо-Пролог дозволяє виконувати зі списком цілий ряд операцій. Їхній перелік включає: доступ до об'єктів списку; перевірку на приналежність до списку; поділ списку на два; злиття двох списків; сортування елементів списку в порядку зростання або убування й т.д. Списки бувають корисні при створенні баз знань (баз даних), експертних систем, словників; перелік областей застосування можна продовжувати ще довго. У справжній главі розглядаються структура, організація й подання списків, демонструються деякі з методів, застосовуваних при програмуванні на Турбо-Пролозі.

Список є набором об'єктів того самого доменного типу. Об'єктами списку можуть бути цілі числа, дійсні числа, символи, символьні рядки й структури. Порядок розташування елементів є відмітною рисою списку; ті ж самі елементи, упорядковані іншим способом, являють вже зовсім інший список. Порядок відіграє важливу роль у процесі зіставлення. Турбо-Пролог допускає списки, елементами яких є структури. Якщо структури належать до альтернативного домену, елементи списку можуть мати різний тип. Такі списки використовуються для спеціальних цілей.

Для зручності обробки списків у Пролозі уведені два поняття: голова й хвіст. Перший елемент списку (або трохи перших елементів) є головою списку, а що залишилися - його хвостом. Елементи списку розділяються комами і беруться у квадратні дужки. Будь-який список являє собою:

- або порожній список (атом []);

- або непустий список - структуру, що складається із двох частин:

- перший елемент - голова (Head) списку;

- другий елемент - хвіст (Tail) списку.

**Самостійна робота.**

**1.** Створіть програму зчитування цілих чисел з термінала й занесення їх у список.

**2.** Створіть програму перетворення рядка в список символів.

**3.** Створіть програму перетворення рядка в список атомів.

**4.** На основі наявних даних створіть Пролог-Програму, що складається з елементів списку.

Дано базу даних у вигляді списку елементів - "Родина". Кожна родина складається із трьох об'єктів: чоловіка, дружин і дітей. Оскільки кількість дітей у різних родинах може бути різним, тому їх необхідно представити у вигляді списку, що складає з довільного числа елементів.

Кожного члена родини у свою чергу потрібно представити структурою, що складається із чотирьох компонентів: ім'я, прізвища, дати народження й роботи. Інформація про роботу - це або "не працює", або вказівки коду роботи й окладу. **Примітка.** Необхідно ввести структури, що відповідають відповідно даті народження члена родини (число, місяць, рік) і роботі.

**А)** На основі створеної програми знайти всі родини із трьома дітьми.

**Б)** Знайти імена й прізвища всіх жінок, що мають принаймні трьох дітей.

**5.4.5 Додати в створену вище програму (базу) наступні рядки.**

husband(X):-family(X,\_,\_). % X - чоловік

vife(X):-family(\_,X,\_). % X - дружина

child(X):-family(\_,\_,Children), % X - дитина

member(X,Children). exist(X):-husband(X);

vife(X); child(X). % X - будь-який член родини dohod(mem(\_,\_,\_,worker(\_,D)),D). % D - дохід працюючого dohod(mem(\_,\_,\_,notwork),0). % 0 - дохід непрацюючого

А) На основі відредагованої програми знайти імена й прізвища всіх людей з бази даних.

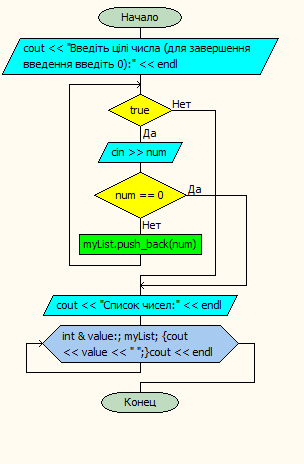
Б) Знайти всіх працюючих дружин.

В) Знайти прізвища людей, чий дохід менше ніж 1000.

**Виконання**

1. **Завдання**

**Блок-схема 1.**

****

#include <iostream>

#include <list>

using namespace std;

int main() {

list<int> myList; // Створюємо пустий список для зберігання цілих чисел

int num;

cout << "Введіть цілі числа (для завершення введення введіть 0):" << endl;

// Зчитування чисел з терміналу та додавання їх до списку

while (true) {

cin >> num;

if (num == 0) {

break; // Завершуємо введення, коли введено 0

}

myList.push\_back(num);

}

// Виведення списку чисел на екран

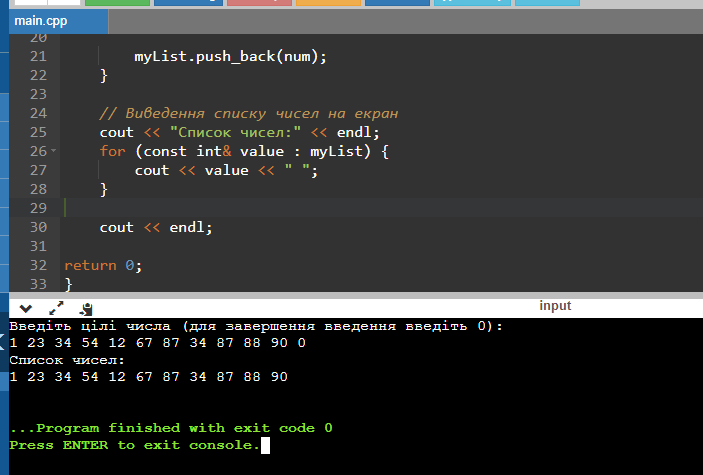
cout << "Список чисел:" << endl;

for (const int& value : myList) {

cout << value << " ";

}

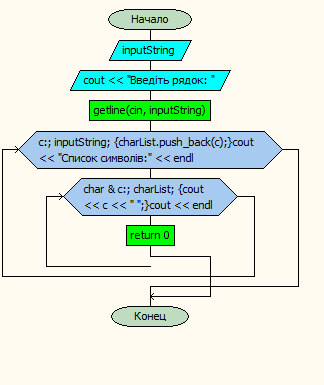
cout << endl;

return0;}

**Мал.1** Зчитування цілих чисел з термінала й занесення їх у список.

1. **Завдання**

**Блок-схема 2.**

****

#include <iostream>

#include <list>

#include <string>

using namespace std;

int main() {

string inputString;

cout << "Введіть рядок: ";

getline(cin, inputString); // Зчитуємо весь рядок, включаючи пробіли

list<char> charList;

for (char c : inputString) {

charList.push\_back(c);

}

cout << "Список символів:" << endl;

for (const char& c : charList) {

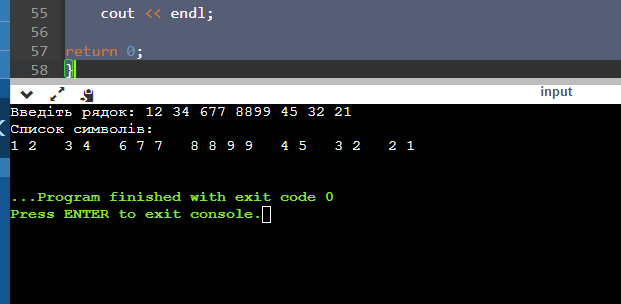
cout << c << " ";

}

cout << endl;

return 0;

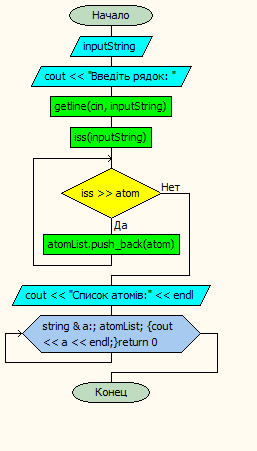
}



**Мал.2** Перетворення рядка в список символів.

**3.Завдання**

**Блок-схема 3.**

****

#include <iostream>

#include <list>

#include <string>

#include <sstream>

using namespace std;

int main() {

string inputString;

cout << "Введіть рядок: ";

getline(cin, inputString);

list<string> atomList;

istringstream iss(inputString);

string atom;

while (iss >> atom) {

atomList.push\_back(atom);

}

cout << "Список атомів:" << endl;

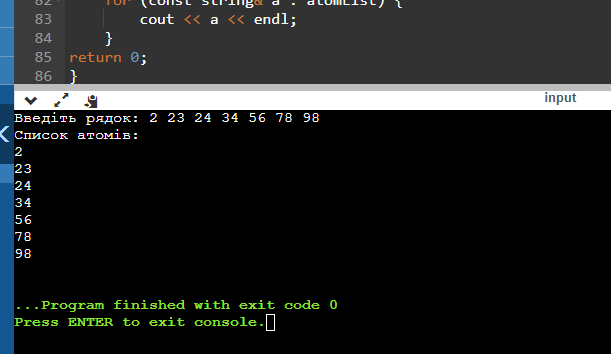
for (const string& a : atomList) {

cout << a << endl;

}

return 0;

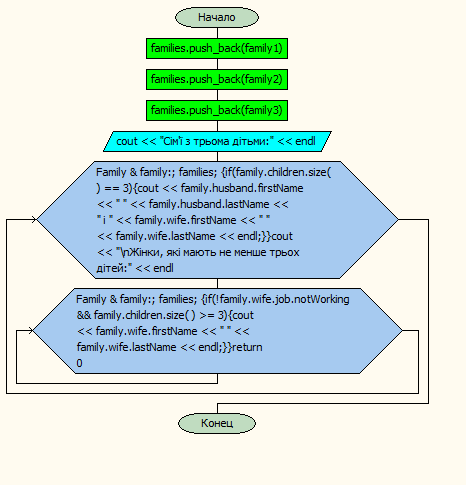
}



**Мал3.** Перетворення рядка в список атомів

**4.Завдання**

**Блок-схема 4.**

****

#include <iostream>

#include <string>

#include <vector>

using namespace std;

// Структура день народження

struct DateOfBirth {

int day;

int month;

int year;

};

// Структура інформація про роботу

struct JobInfo {

bool notWorking;

int jobCode;

double salary;

};

// Структура членів сім'ї

struct FamilyMember {

string firstName;

string lastName;

DateOfBirth dob;

JobInfo job;

};

// Страуктура про сім'ю

struct Family {

FamilyMember husband;

FamilyMember wife;

vector<FamilyMember> children;

};

int main() {

// Створення списку сімей

vector<Family> families;

// Заповнити базу даних зразками даних

Family family1{

{"Назар", "Будніков", {1, 1, 1980}, {false, 101, 50000}},

{"Євгенія", "Буднікова", {15, 5, 1982}, {false, 102, 45000}},

{

{"Петро", "Будніков", {10, 2, 2005}, {false, 0, 0}},

{"Діма", "Будніков", {20, 4, 2007}, {false, 0, 0}},

{"Ваня", "Будніков", {5, 9, 2010}, {false, 0, 0}}

}

};

families.push\_back(family1);

Family family2{

{"Дмитро", "Чорній", {2, 3, 1979}, {false, 104, 70000}},

{"Наталія", "Чорній", {19, 8, 1981}, {false, 115, 40000}},

{

{"Оля", "Чорній", {10, 2, 2003}, {false, 0, 0}},

{"Настя", "Чорній", {10, 2, 2003}, {false, 0, 0}}

}

};

families.push\_back(family2);

Family family3{

{"Володимир", "Назалевич", {10, 8, 1980}, {false, 131, 80000}},

{"Анна", "Назалевич", {23, 6, 1982}, {false, 182, 55000}},

{

{"Вікторія", "Назалевич", {10, 2, 2003}, {false, 0, 0}}

}

};

families.push\_back(family3);

// Завдання А: Знайти всі сім’ї з трьома дітьми

cout << "Сім'ї з трьома дітьми:" << endl;

for (const Family& family : families) {

if (family.children.size() == 3) {

cout << family.husband.firstName << " " << family.husband.lastName << " і " << family.wife.firstName << " " << family.wife.lastName << endl;

}

}

// Завдання Б: Знайдіть імена та прізвища всіх жінок, які мають не менше трьох дітей

cout << "\nЖінки, які мають не менше трьох дітей:" << endl;

for (const Family& family : families) {

if (!family.wife.job.notWorking && family.children.size() >= 3) {

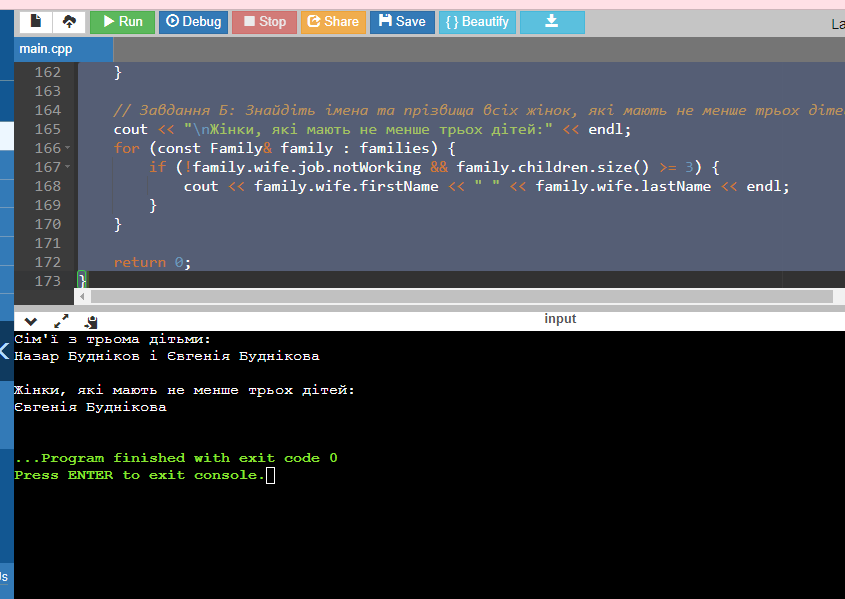
cout << family.wife.firstName << " " << family.wife.lastName << endl;

}

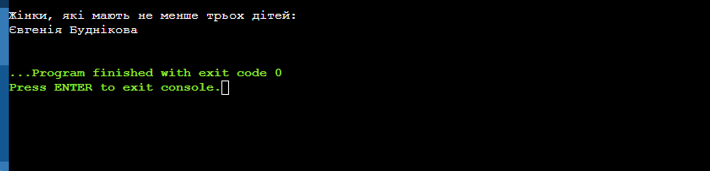
}

return 0;

}



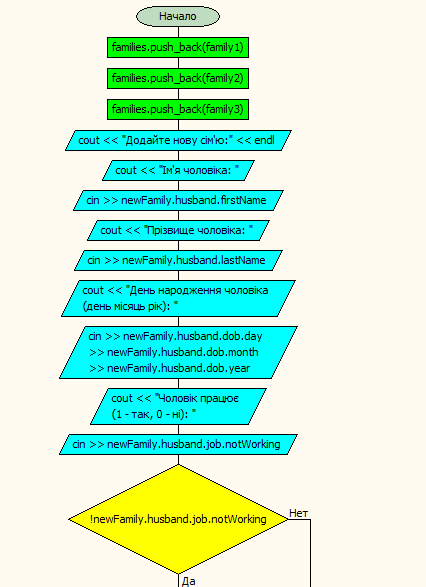
**Мал.4.1.** Створення програми, знаходження всіх родини із трьома дітьми.

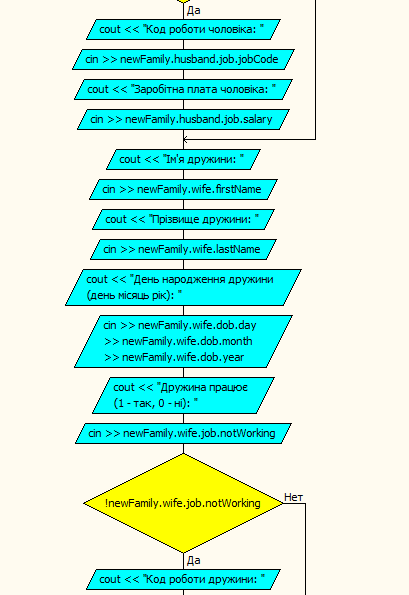


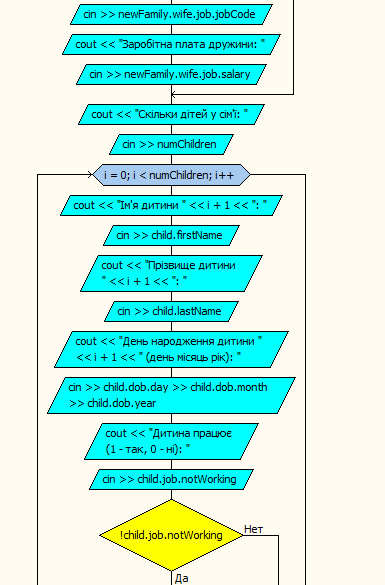
**Мал.4.2.** Знаходження імен і прізвищ всіх жінок, що мають принаймні трьох дітей.

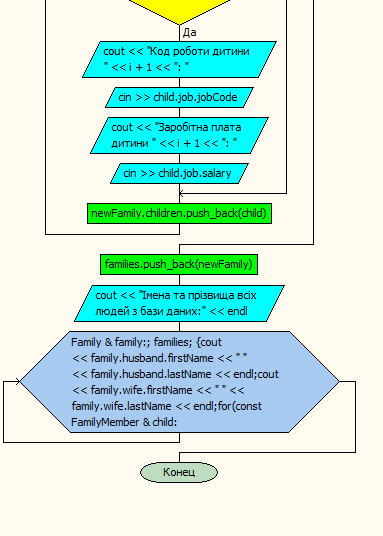
**5.Завдання**

**Блок-схема 5.**

****

****

****

****

#include <iostream>

#include <string>

#include <vector>

using namespace std;

// Структура день народження

struct DateOfBirth {

int day;

int month;

int year;

};

// Структура інформація про роботу

struct JobInfo {

bool notWorking;

int jobCode;

double salary;

};

// Структура членів сім'ї

struct FamilyMember {

string firstName;

string lastName;

DateOfBirth dob;

JobInfo job;

};

// Страуктура про сім'ю

struct Family {

FamilyMember husband;

FamilyMember wife;

vector<FamilyMember> children;

};

int main() {

// Створення списку сімей

vector<Family> families;

// Заповнити базу даних зразками даних

Family family1{

{"Назар", "Будніков", {1, 1, 1980}, {false, 101, 50000}},

{"Євгенія", "Буднікова", {15, 5, 1982}, {false, 102, 45000}},

{

{"Петро", "Будніков", {10, 2, 2005}, {false, 0, 0}},

{"Діма", "Будніков", {20, 4, 2007}, {false, 0, 0}},

{"Ваня", "Будніков", {5, 9, 2010}, {false, 0, 0}}

}

};

families.push\_back(family1);

Family family2{

{"Дмитро", "Чорній", {2, 3, 1979}, {false, 104, 70000}},

{"Наталія", "Чорній", {19, 8, 1981}, {false, 115, 40000}},

{

{"Оля", "Чорній", {10, 2, 2003}, {false, 0, 0}},

{"Настя", "Чорній", {10, 2, 2003}, {false, 0, 0}}

}

};

families.push\_back(family2);

Family family3{

{"Володимир", "Назалевич", {10, 8, 1980}, {false, 131, 80000}},

{"Анна", "Назалевич", {23, 6, 1982}, {false, 182, 55000}},

{

{"Вікторія", "Назалевич", {10, 2, 2003}, {false, 0, 0}}

}

};

families.push\_back(family3);

// Додати нову сім'ю з клавіатури

Family newFamily;

cout << "Додайте нову сім'ю:" << endl;

cout << "Ім'я чоловіка: ";

cin >> newFamily.husband.firstName;

cout << "Прізвище чоловіка: ";

cin >> newFamily.husband.lastName;

cout << "День народження чоловіка (день місяць рік): ";

cin >> newFamily.husband.dob.day >> newFamily.husband.dob.month >> newFamily.husband.dob.year;

cout << "Чоловік працює (1 - так, 0 - ні): ";

cin >> newFamily.husband.job.notWorking;

if (!newFamily.husband.job.notWorking) {

cout << "Код роботи чоловіка: ";

cin >> newFamily.husband.job.jobCode;

cout << "Заробітна плата чоловіка: ";

cin >> newFamily.husband.job.salary;

}

cout << "Ім'я дружини: ";

cin >> newFamily.wife.firstName;

cout << "Прізвище дружини: ";

cin >> newFamily.wife.lastName;

cout << "День народження дружини (день місяць рік): ";

cin >> newFamily.wife.dob.day >> newFamily.wife.dob.month >> newFamily.wife.dob.year;

cout << "Дружина працює (1 - так, 0 - ні): ";

cin >> newFamily.wife.job.notWorking;

if (!newFamily.wife.job.notWorking) {

cout << "Код роботи дружини: ";

cin >> newFamily.wife.job.jobCode;

cout << "Заробітна плата дружини: ";

cin >> newFamily.wife.job.salary;

}

int numChildren;

cout << "Скільки дітей у сім'ї: ";

cin >> numChildren;

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

FamilyMember child;

cout << "Ім'я дитини " << i + 1 << ": ";

cin >> child.firstName;

cout << "Прізвище дитини " << i + 1 << ": ";

cin >> child.lastName;

cout << "День народження дитини " << i + 1 << " (день місяць рік): ";

cin >> child.dob.day >> child.dob.month >> child.dob.year;

cout << "Дитина працює (1 - так, 0 - ні): ";

cin >> child.job.notWorking;

if (!child.job.notWorking) {

cout << "Код роботи дитини " << i + 1 << ": ";

cin >> child.job.jobCode;

cout << "Заробітна плата дитини " << i + 1 << ": ";

cin >> child.job.salary;

}

newFamily.children.push\_back(child);

}

// Додати нову сім'ю до списку сімей

families.push\_back(newFamily);

// Завдання А: Знайти імена та прізвища всіх людей з бази даних

cout << "Імена та прізвища всіх людей з бази даних:" << endl;

for (const Family& family : families) {

cout << family.husband.firstName << " " << family.husband.lastName << endl;

cout << family.wife.firstName << " " << family.wife.lastName << endl;

for (const FamilyMember& child : family.children) {

cout << child.firstName << " " << child.lastName << endl;

}

}

// Завдання Б: Знайти всіх працюючих дружин

cout << "Працюючі дружини:" << endl;

for (const Family& family : families) {

if (!family.wife.job.notWorking) {

cout << family.wife.firstName << " " << family.wife.lastName << endl;

}

}

// Завдання В: Знайти прізвища людей, чий дохід менше ніж 100000

cout << "Прізвища людей з доходом менше 100000:" << endl;

for (const Family& family : families) {

if (family.husband.job.salary < 100000) {

cout << family.husband.lastName << endl;

}

if (!family.wife.job.notWorking && family.wife.job.salary < 100000) {

cout << family.wife.lastName << endl;

}

for (const FamilyMember& child : family.children) {

if (!child.job.notWorking && child.job.salary < 100000) {

cout << child.lastName << endl;

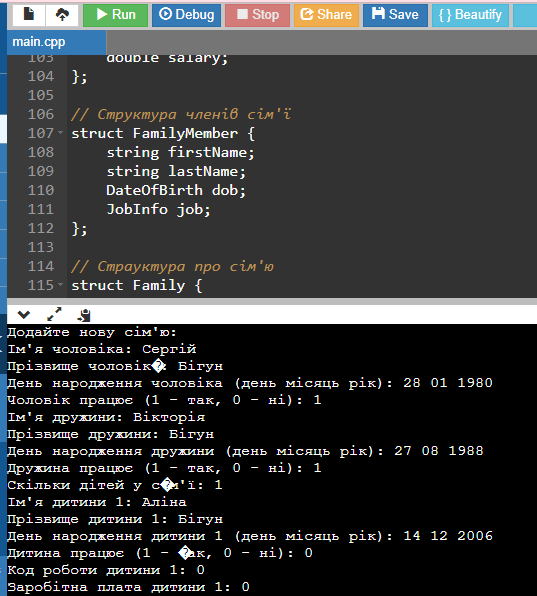
}

}

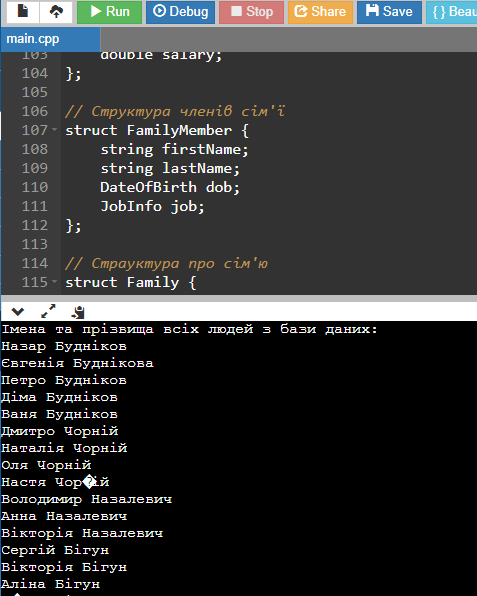
}

return 0;

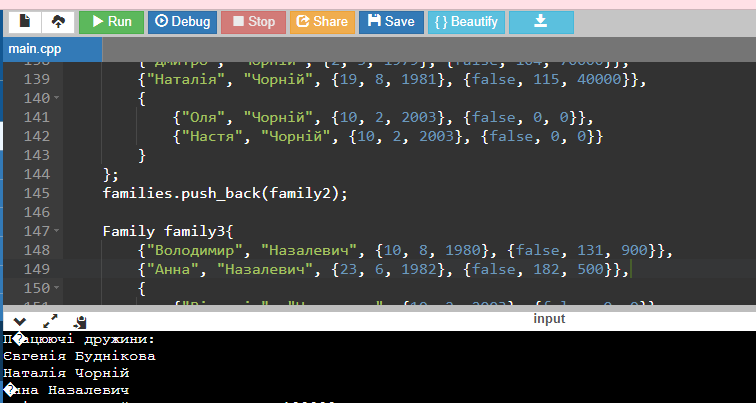
}



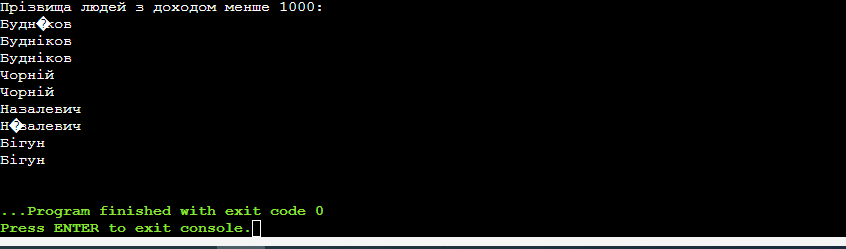
**Мал.5.1 Додавання з клавіатури нової сімї**

****

**Мал.5.2 Виведення всіх членів бази даних**

****

**Мал.5.3** Виведення всіх працюючих дружин.

****

**Мал.5.4** Виведенняпрізвищ людей, чий дохід менше ніж 10000.

**Висновок**. Виконуючи лабораторну роботу, я навчилася працювати зі списками та навчилася використовувати операції над списками в Пролог-програмах.