Міністерство освіти і науки України

Дніпровський національний університет імені Олеся Гончара

Факультет прикладної математики і компʼютерних технологій

Кафедра компʼютерних технологій

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 2**

**з курсу «Моделі та методи штучного інтелекту»**

**на тему «Родинне дерево з фактами в динамічній базі даних»**

**Варіант №14**

Виконав:

студент гр. ПА-22-2

Овдієнко Андрій

Дніпро

2025

**Зміст**

[**1. Постановка задачі** 3](#_Toc53239595)

[**2. Опис розв’язку** 4](#_Toc53239596)

[**3. Вихідний текст програми розв’язку задачі** 5](#_Toc53239597)

[**(основні фрагменти з коментарями)** 5](#_Toc53239598)

[**4. Опис інтерфейсу програми (керівництво користувача)** 10](#_Toc53239599)

[**5. Опис тестових прикладів** 13](#_Toc53239600)

[**5.1 Приклад 1** 13](#_Toc53239601)

[**5.2 Приклад 2** 25](#_Toc53239602)

[**6. Усунення зауважень** 27](#_Toc53239603)

**1. Постановка задачі**

Вирішити задачі, аналогічні до тих, які дані попередньому завданні 1 (див. розділ “Об’єкти даних”), але родинне дерево з фактами повинне знаходитись в динамічній базі даних. Користувач має сам заповнити цю базу даних під час виконання програми, а потім записати її в файл. Можна потім зчитати її з файлу. Програма повинна мати меню, яке дозволяє виконувати команди користувача, наприклад:

1. заповнити базу даних;
2. записати базу в файл;
3. зчитати базу із файлу;
4. знайти прадіда;
5. вихід.

**2. Опис розв’язку**

Для роботи з базою даних потрібно її створити. Вона не матиме назви, але прийматиме 3 елементи: чоловік, жінка, опікун та дитина.

Для пошуку по меню створимо правила, які переберуть усі можливі варіанти введення рядка (readln(STRING StringVariable) - зчитує рядок символів з клавіатури), оскільки якщо ми зробимо це з числом, і людина щось введе (неправильно), виникне помилка.

Під час додавання використовуватимемо три функції. Кожну окремо для чоловіків (asserta(<домен\_фактів> Факт) - додає на початок для чоловіків), жінок (assertz(<домен\_фактів> Факт) - додає на кінець для жінок), зв'язків між батьками та дітьми (assert(<домен\_фактів> Факт) - додає на кінець для зв'язків).

Щоб видалити всі елементи одного типу, використовуватимемо: retractall(<домен\_фактів> Факт).

Щоб зберегти правила в текстовий файл, використовуватимемо: save(STRING OSFileName).

Для зчитування даних з бази даних використовуватимемо: consult(STRING OSFileName).

Вважатимемо умови if – це правила, які слід прописати.

**3. Вихідний текст програми розв’язку задачі**

**(основні фрагменти з коментарями)**

domains

person = string

database

man(person)

woman(person)

parent(person,person)

predicates

nondeterm equal(person,person)

nondeterm not\_equal(person,person)

nondeterm switch(string)

nondeterm menu

nondeterm add\_a\_new\_person(person,string)

nondeterm add\_a\_new\_node\_parent(person,person)

nondeterm save\_db(string)

nondeterm load\_db(string)

nondeterm sister(person,person)

nondeterm sisters(person,person)

nondeterm husband(person,person)

nondeterm brother\_in\_law(person,person)

clauses

menu:-

write("\n\nMain menu:\n\n"),

write("1. Add a new person;\n"),

write("2. Add a new logic: parent(dad/mom, son/daughter);\n"),

write("3. Push datas in the file;\n"),

write("4. Get datas from the file;\n"),

write("5. Seek the brother in law;\n"),

write("6. Show all facts;\n"),

write("7. Free db.\n"),

write("8. Exit.\n\n"),

write("Your choice:"),

readln(N),

switch(N).

switch("1") :-

write("\nThe person's name: "),

readln(Name),

write("\nThe sex (man/woman): "),

readln(Sex),

add\_a\_new\_person(Name,Sex).

switch("2"):-

write("\nThe parent's name: "),

readln(Parent),

write("\nThe child's name: "),

readln(Child),

add\_a\_new\_node\_parent(Parent,Child).

switch("3"):-write("\nThe db name: "),readln(Name),save\_db(Name).

switch("4"):-write("\nThe db name: "),readln(Name),load\_db(Name).

switch("5"):-brother\_in\_law(X,Y),write("\n",X ,"is brother in law of ", Y,".\n"),fail.

switch("5"):-menu.

switch("6"):-

man(X),write("man(",X,")\n"),

fail.

switch("6"):-

woman(X),write("woman(",X,")\n"),

fail.

switch("6"):-

parent(X,Y),write("parent(",X,"," ,Y,")\n"),

fail.

switch("6"):-

menu.

switch("7"):-retractall(man(\_)),retractall(woman(\_)),retractall(parent(\_,\_)),menu.

switch("8"):-retractall(man(\_)),retractall(woman(\_)),retractall(parent(\_,\_)),write("\nBye!\n\n").

switch(\_):-write("Do you speak English?\nNo?\nI didn't say this choice!\n"),menu.

add\_a\_new\_person(Name,"man"):-man(Name),write("\nman(",Name,") is in the db.\n"),menu.

add\_a\_new\_person(Name,"woman"):-woman(Name),write("\nwoman(",Name,") is in the db.\n"),menu.

add\_a\_new\_person(Name,"woman"):-man(Name),write("\nman(",Name,") is in the db.\n"),menu.

add\_a\_new\_person(Name,"man"):-woman(Name),write("\nwoman(",Name,") is in the db.\n"),menu.

add\_a\_new\_person(Name,"man"):-asserta(man(Name)),write("\nman(",Name,") has been successfully added.\n"),menu.

add\_a\_new\_person(Name,"woman"):-assertz(woman(Name)),write("\nwoman(",Name,") has been successfully added.\n"),menu.

add\_a\_new\_person(\_,\_):-write("\nDo you speak English? \nNo?\nI didn't say this sex !\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-parent(Parent,Child),write("\nThe node parent(",Parent,",",Child,") is in the db.\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-parent(Child,Parent),write("\nThe node parent(",Child,",",Parent,") is in the db.\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-not\_equal(Parent,Child),man(Parent),man(Child),assert(parent(Parent,Child)),write("\nparent(",Parent,", ",Child,") has been successfully added.\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-not\_equal(Parent,Child),man(Parent),woman(Child),assert(parent(Parent,Child)),write("\nparent(",Parent,", ",Child,") has been successfully added.\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-not\_equal(Parent,Child),woman(Parent),man(Child),assert(parent(Parent,Child)),write("\nparent(",Parent,", ",Child,") has been successfully added.\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-not\_equal(Parent,Child),woman(Parent),woman(Child),assert(parent(Parent,Child)),write("\nparent(",Parent,", ",Child,") has been successfully added.\n"),menu.

add\_a\_new\_node\_parent(Parent,Child):-write("\nI can't add a parent(",Parent,", ",Child,") because I don't know those people or one of them .\n"),menu.

save\_db(Name):-save(Name),write("\nThe db has been saved to ",Name,".\n"),menu.

save\_db(Name):-write("\nSomthing went wrong. I didn't save this db in the ", Name,".\n").

load\_db(Name):-retractall(man(\_)),retractall(woman(\_)),retractall(parent(\_,\_)),consult(Name),write("The db has been loaded from ",Name),menu.

load\_db(Name):-write("\nSomthing went wrong. I didn't load this db in the ", Name,".\n").

equal(X,Y):-X=Y.

not\_equal(X,Y):-X<>Y.

sister(X,Y):-woman(X),parent(Z,X),parent(Z,Y),woman(Z),not\_equal(X,Y),not\_equal(X,Z),not\_equal(Y,Z).

sisters(X,Y):-sister(X,Y),sister(Y,X).

husband(X,Y):-man(X),woman(Y),parent(X,Z),parent(Y,Z),not\_equal(X,Y),not\_equal(X,Z),not\_equal(Y,Z).

brother\_in\_law(X,Y):-sisters(Y,Z),husband(X,Z),not\_equal(X,Y),not\_equal(X,Z),not\_equal(Y,Z).

goal

menu.

**4. Опис інтерфейсу програми (керівництво користувача)**

Після написання коду – можна запустити тест на виконання за допомогою кнопки, яка показано на рисунку (Рисунок 2).

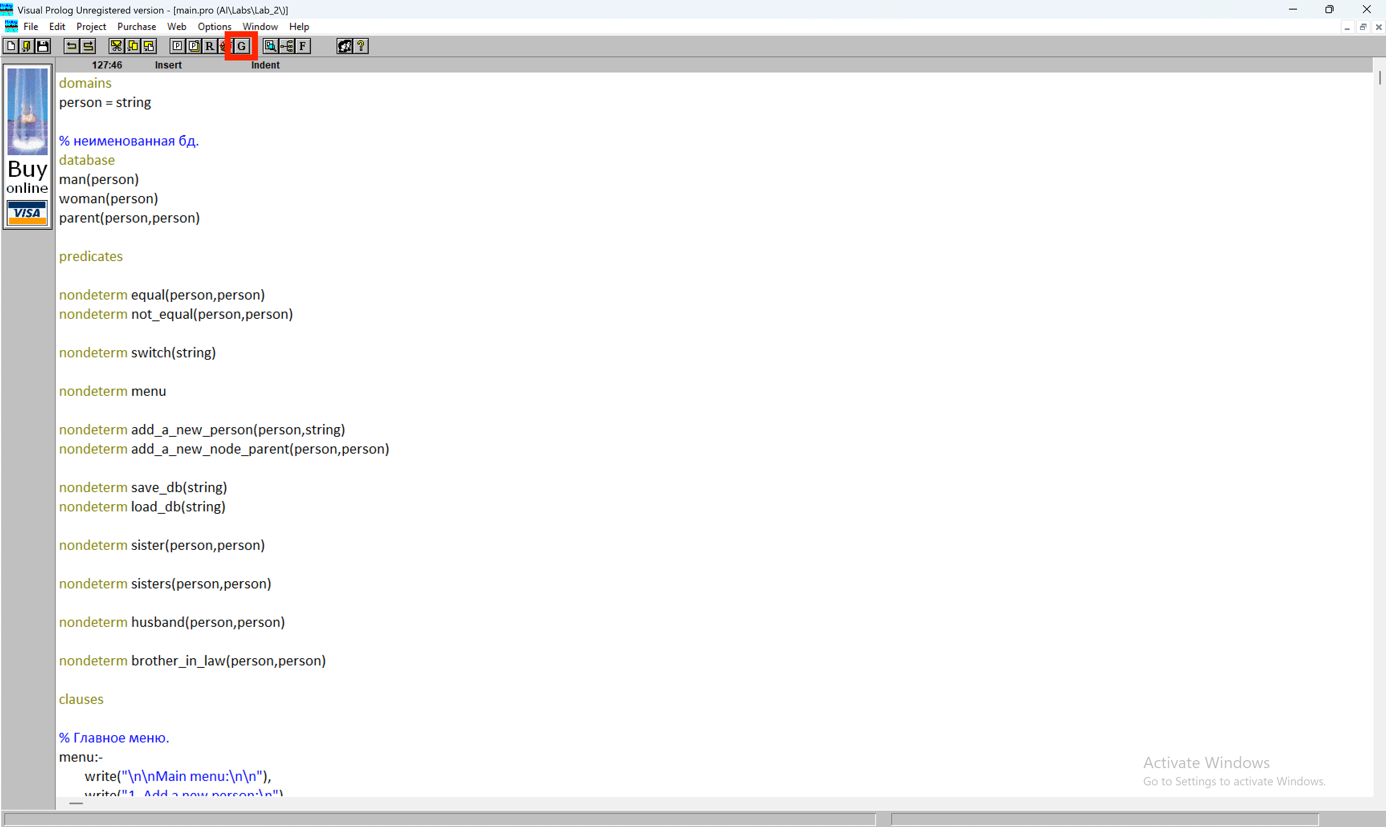


Рисунок 1 – Інтерфейс користувача та кнопка запуску теста.

Після запуску програми відкривається головне меню(Рисунок 2).

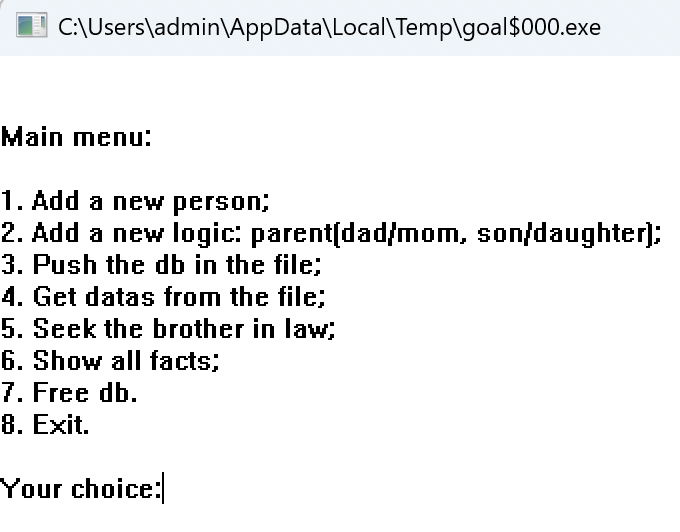


Рисунок 2 – Головне меню програми.

Можна побачити вісім пунктів меню:

1. Додається новий користувач. Пропонується ввести імʼя, а потім обрати стать (man/woman) – якщо стать обрана неправильно, то програма культурно скаже про це і недодасть людину. Також якщо ви спробуєте додати людину з іменем, яке вже є – програма засіче це і недодасть нову людину. Якщо все зроблено правильно – додається нова людина, і записується в неіменовану бд.
2. Додається звʼязок опікуна та дитини. Пропонується ввести імʼя опікуна а потім імʼя дитини. Якщо вже є такий звʼязок чи звʼязок де дитина - це опікун, а опікун – це дитина – скажемо про це і не створимо звʼязок. Також перевіряється, щоб обидві людини були в базі. Якщо все зроблено правильно – створюється новий звʼязок, і записується в неіменовану бд.
3. Дані з бази записуються у файл. Файл людина не може побачити напряму, тому не зможе йому зашкодити.
4. Дані у базу записуються із файла. Файл людина не може побачити напряму, тому не зможе йому зашкодити.
5. Пошук свояка. Видається повідомлення, що такий то чоловік є свояком для такоїто дружини.
6. Вивід правил. Виводить всі правила виключно в такому порядку, які записані в базі: чоловіки, жінки, опікуни.
7. База очищається. Всі правила онуляються.
8. Вихід. Скаже: “Bye!” – і очистить бд.

**5. Опис тестових прикладів**

**5.1 Приклад 1**

Для початку покажемо, що ми працюємо з порожньою базою.

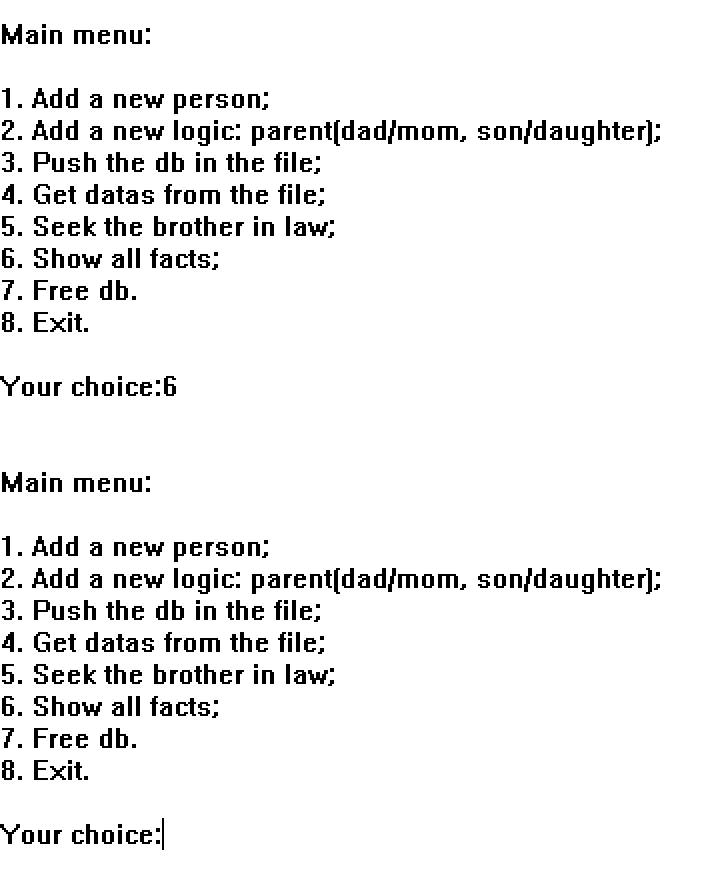


Рисунок 3 – Порожня динамічна база.

Додамо туди Андрія та виведемо факти.

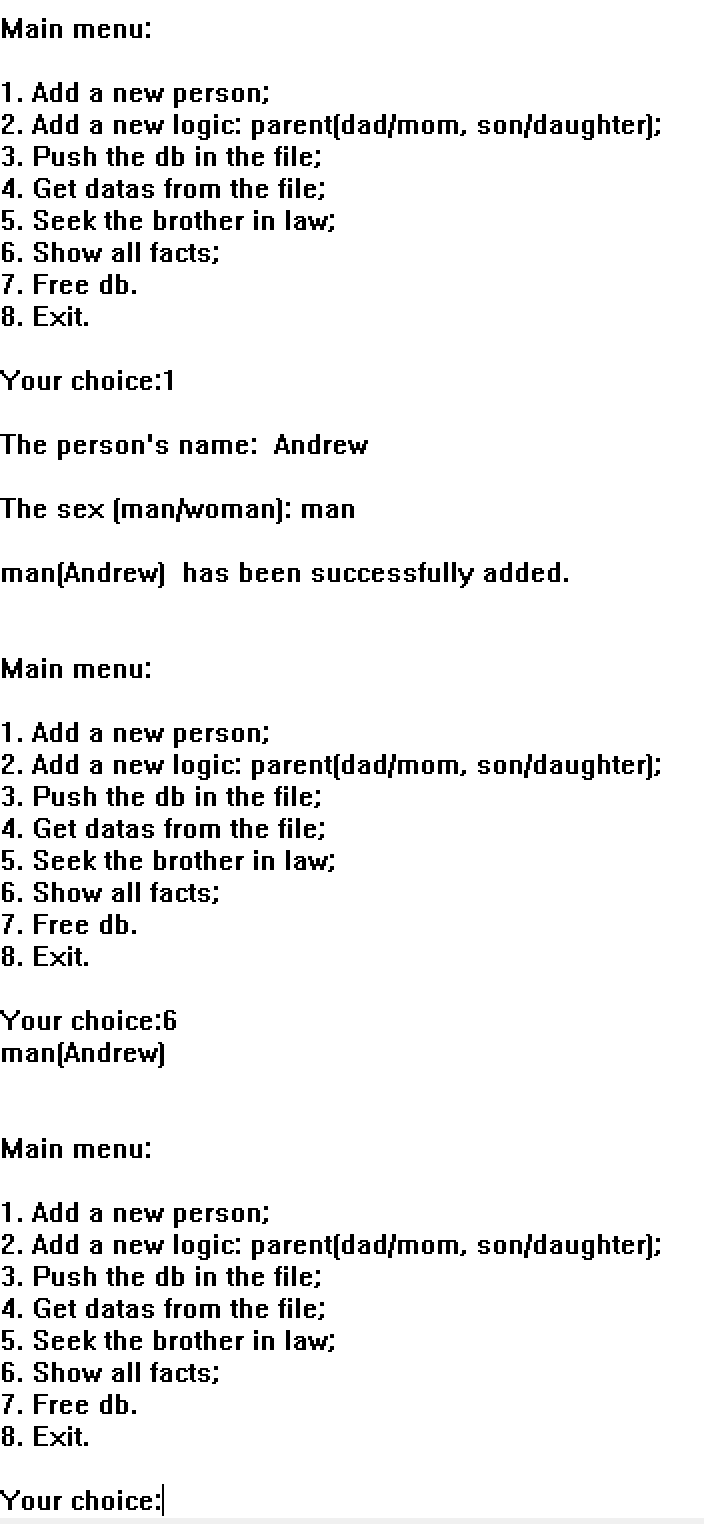


Рисунок 4 – Додавання чоловіка і вивід фактів.

Спробуємо додати Катерину, але стать вкажемо хибну.

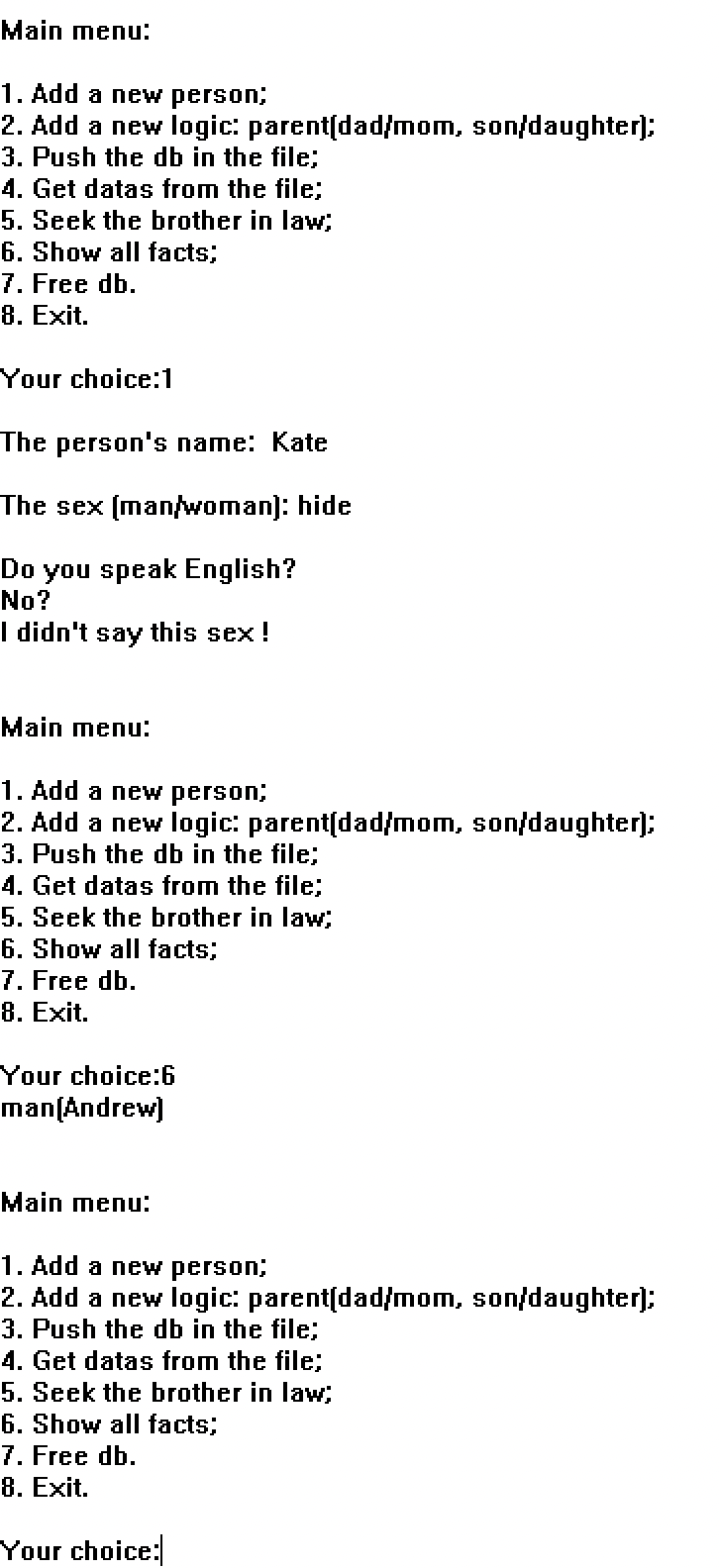


Рисунок 5 – Хибне додавання жінки. Вивід фактів.

Додамо Катерину та виведемо факти.

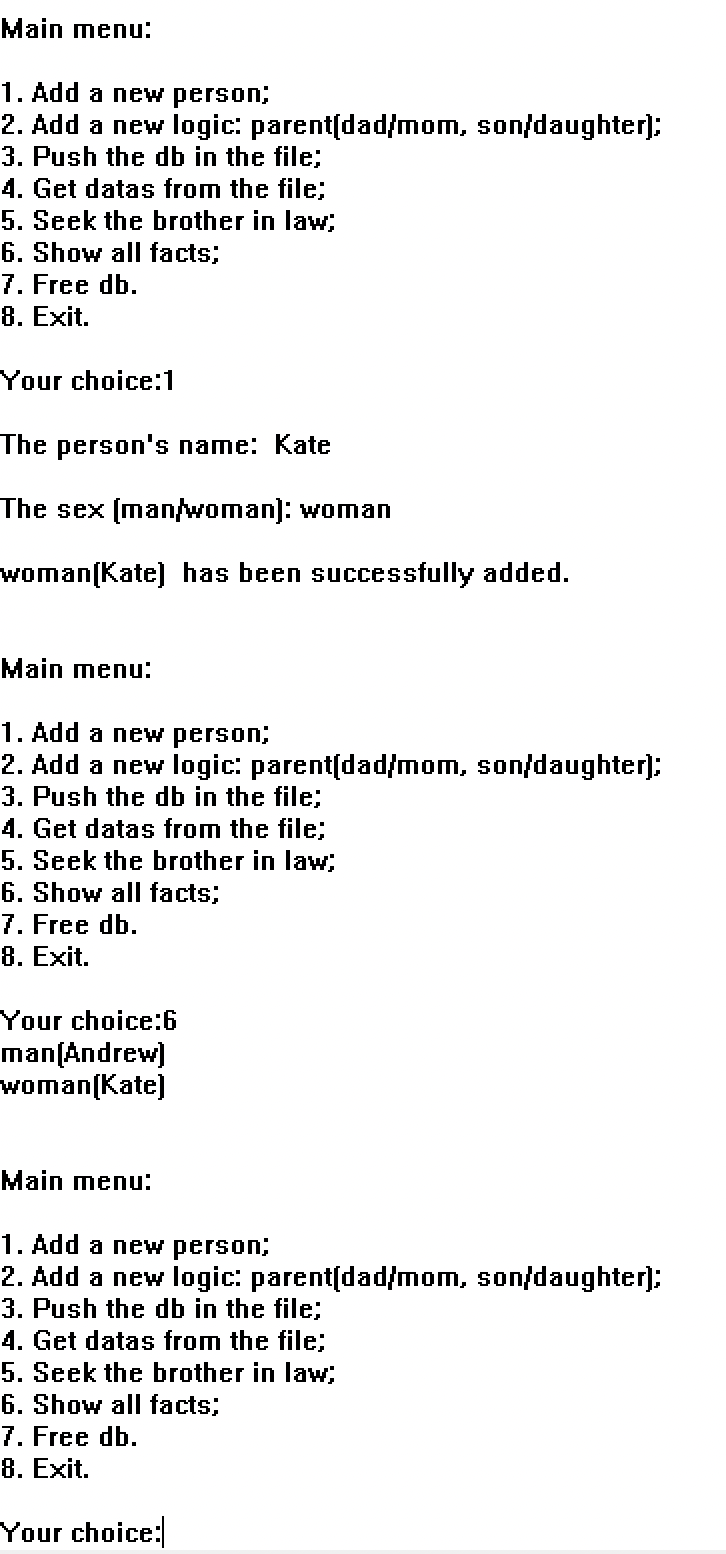


Рисунок 6 – Додавання жінки і вивід фактів.

Спробуємо занести Андрія вдруге.

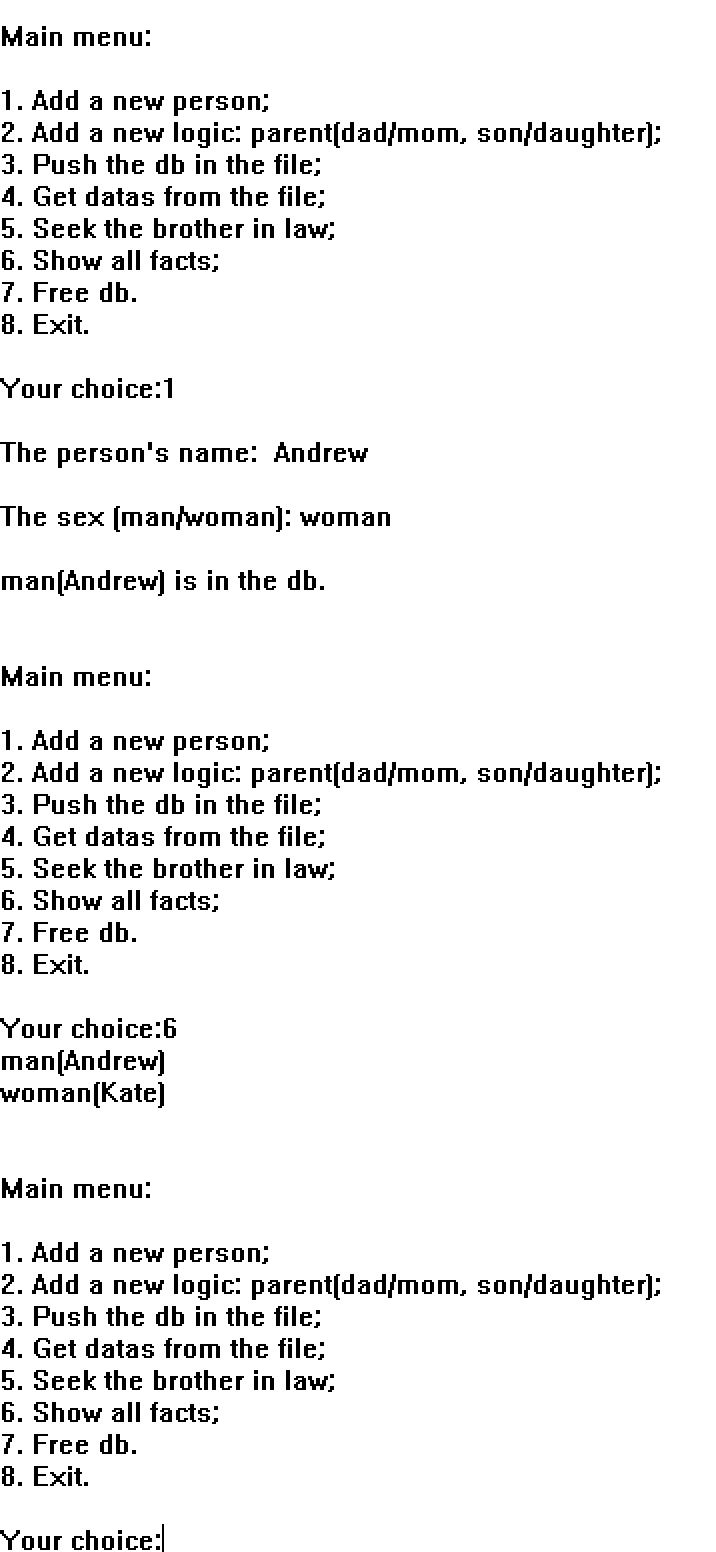


Рисунок 7 – Спроба занести Андрія вдруге та вивід фактів.

Спробуємо створити звʼязок опікуна і дитини, якої немає в бд, як людини.

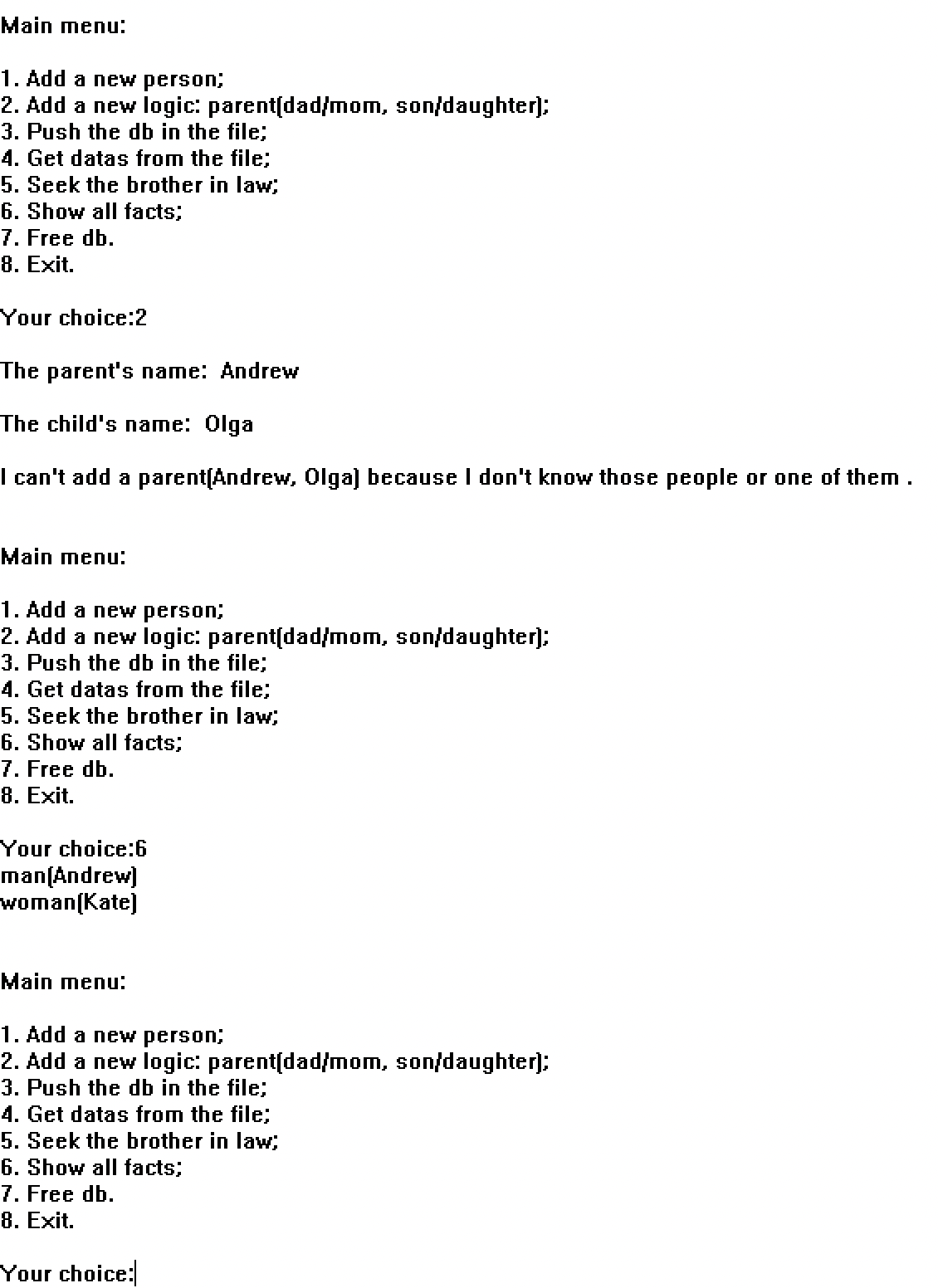


Рисунок 8 – спроба створити звʼязок опікуна та дитини, в той час, як дитини немає в базі, вивід фактів.

Додамо до бази Тома.

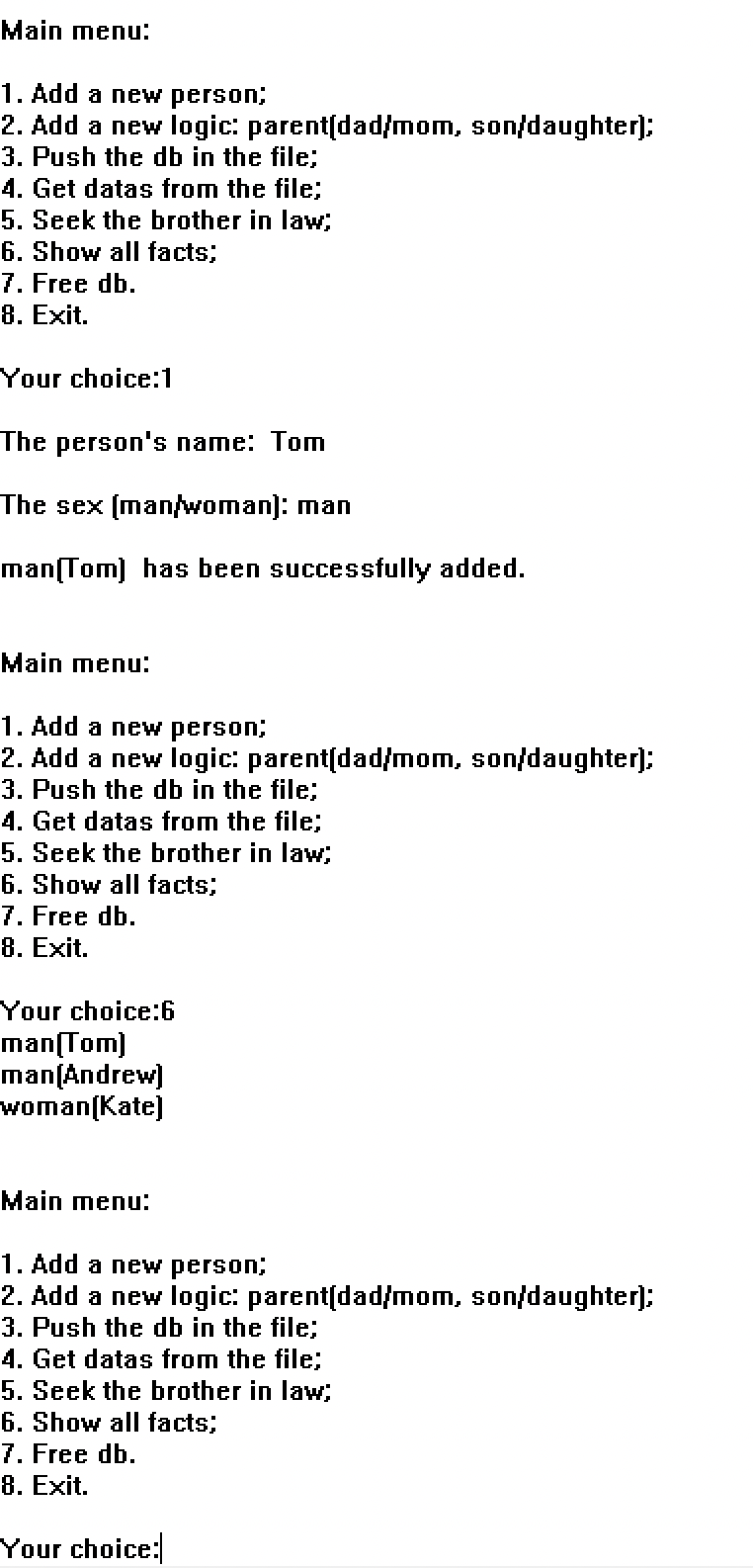


Рисунок 9 – Додавання до бази Тома, вивід фактів.

Створимо звʼязок, що Андрій є опікуном Тома.

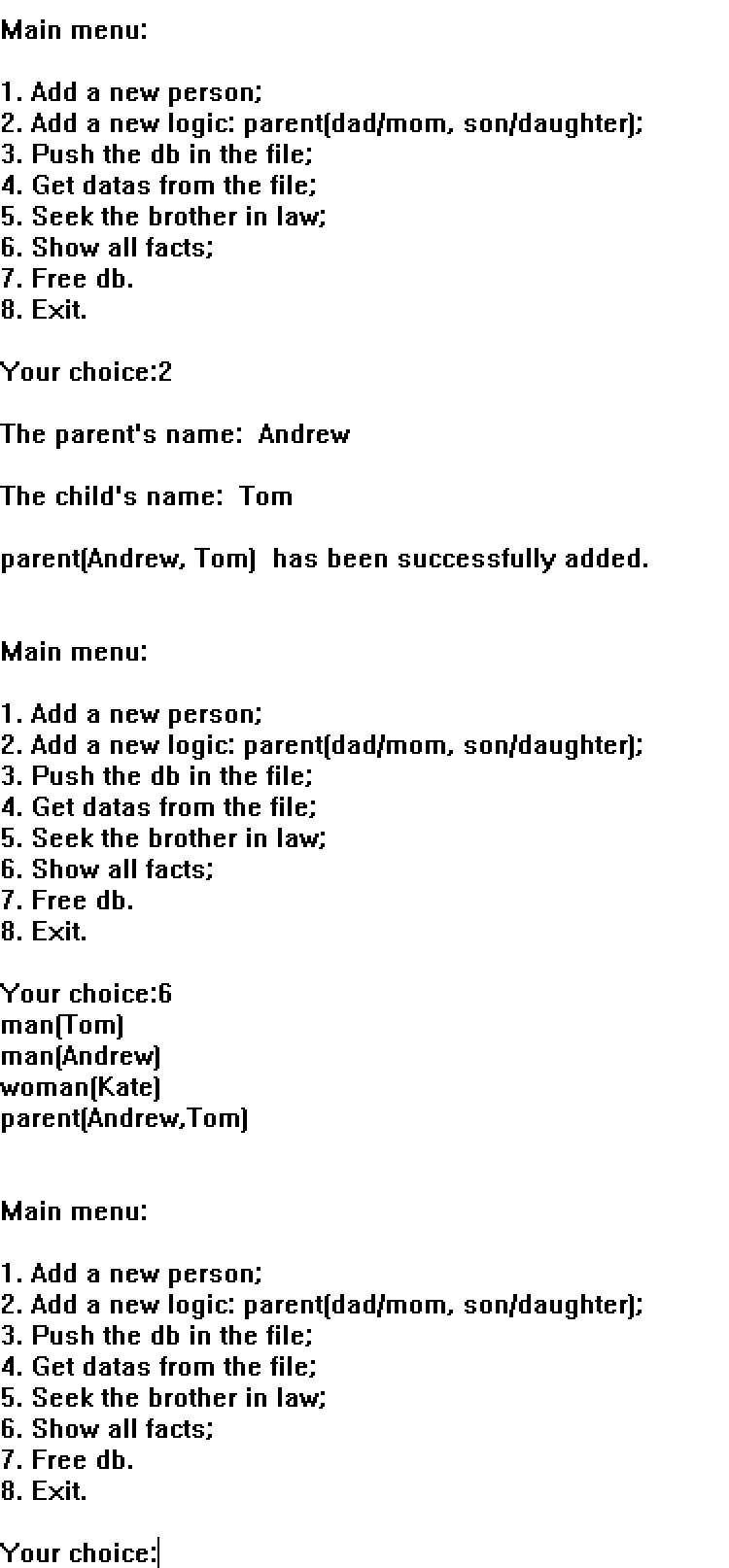


Рисунок 10 – Створення звʼязка, що Андрій опікун Тома, вивід фактів.

Спроба додати, що Том опікун Андрія.

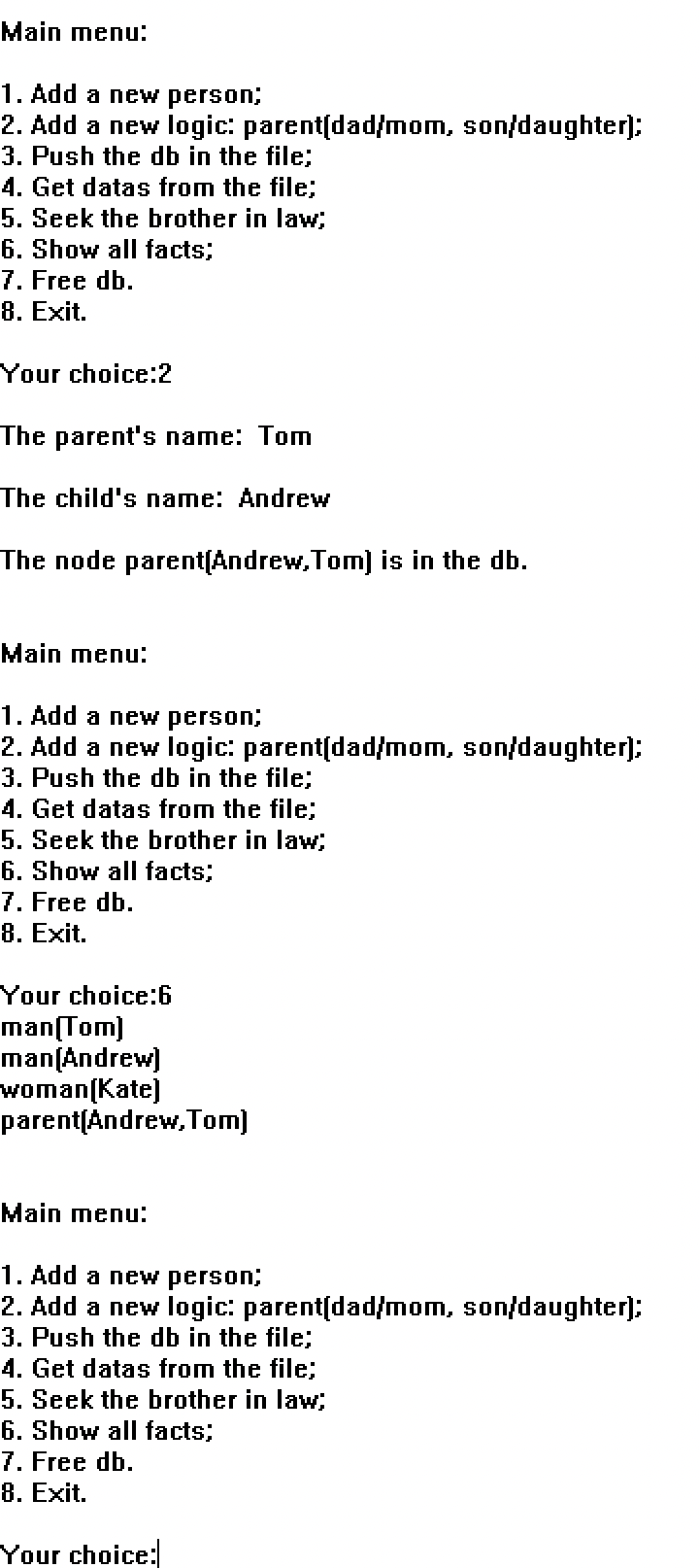


Рисунок 11 – Хибна спроба додати, що опікун є дитиною, а дитина є опікуном, вивід фактів.

Завантажимо дані у файл: “file.txt”.

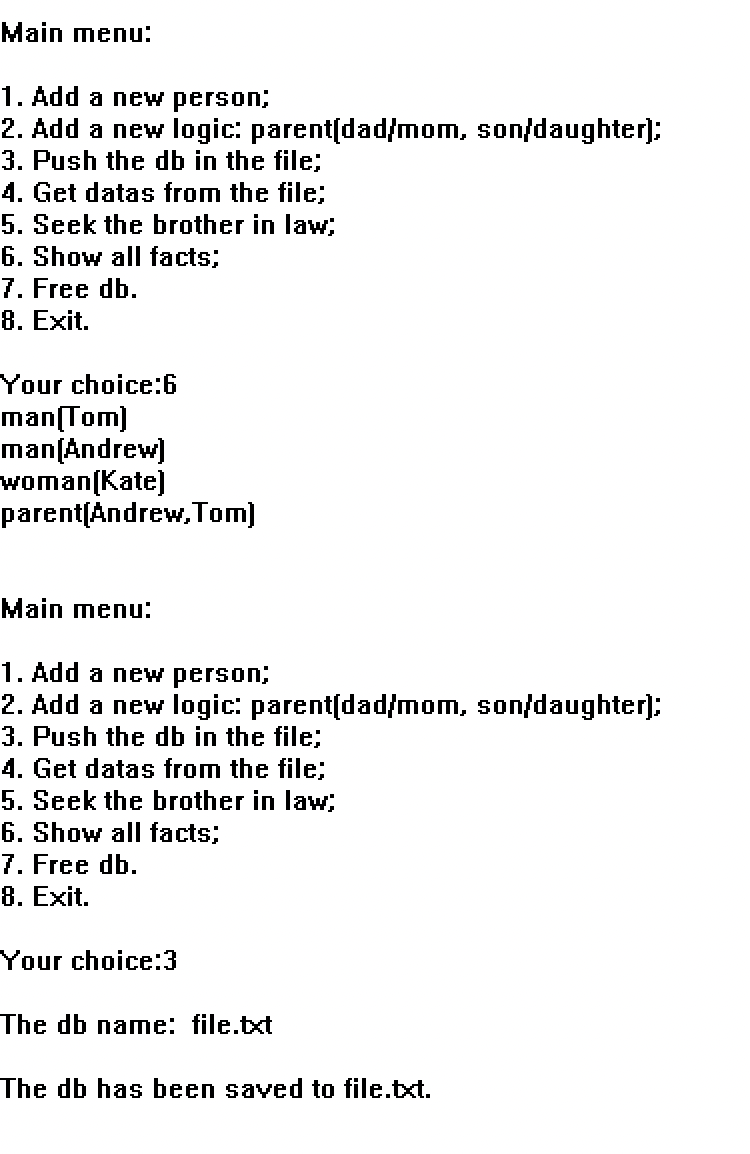


Рисунок 12 – вивід фактів, завантаження їх до file.txt.

Виведемо дані, очистимо базу та виведемо факти ще раз.

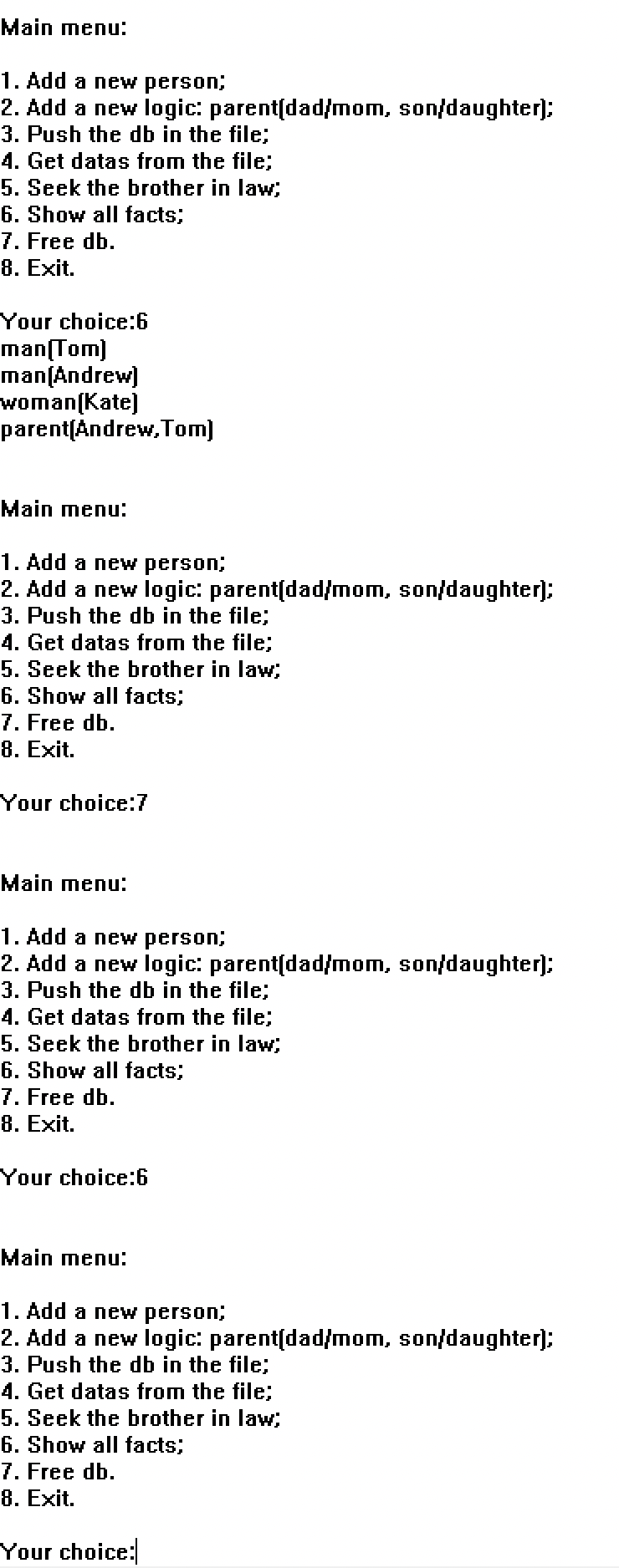


Рисунок 13 – Вивід фактів, очистка бази, вивід фактів.

Завантажимо дані з файла: “file.txt”.



Рисунок 14 – Вивід порожніх фактів, завантаження файла file.txt фактів, вивід фактів.

Закінчимо роботу програми.

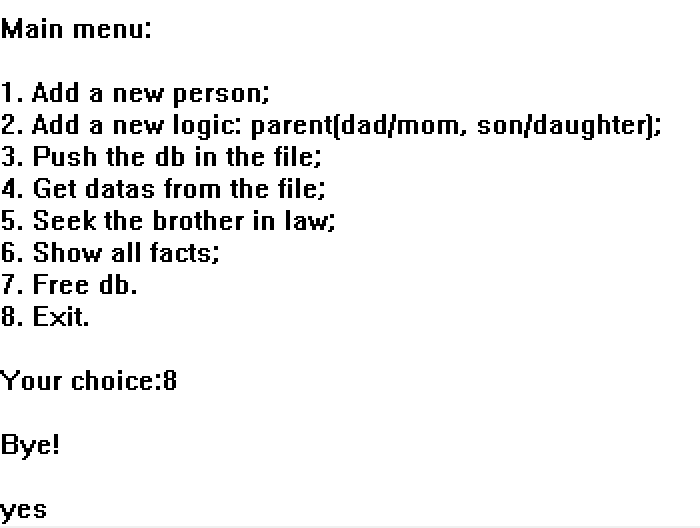


Рисунок 15 – Закриття програми, вихід.

**5.2 Приклад 2**

Для другого прикладу використаємо ту базу данних, яка була в першій лабі. Заповнемо її і продемоструємо вже фінальні факти і роботу пошука свояка.

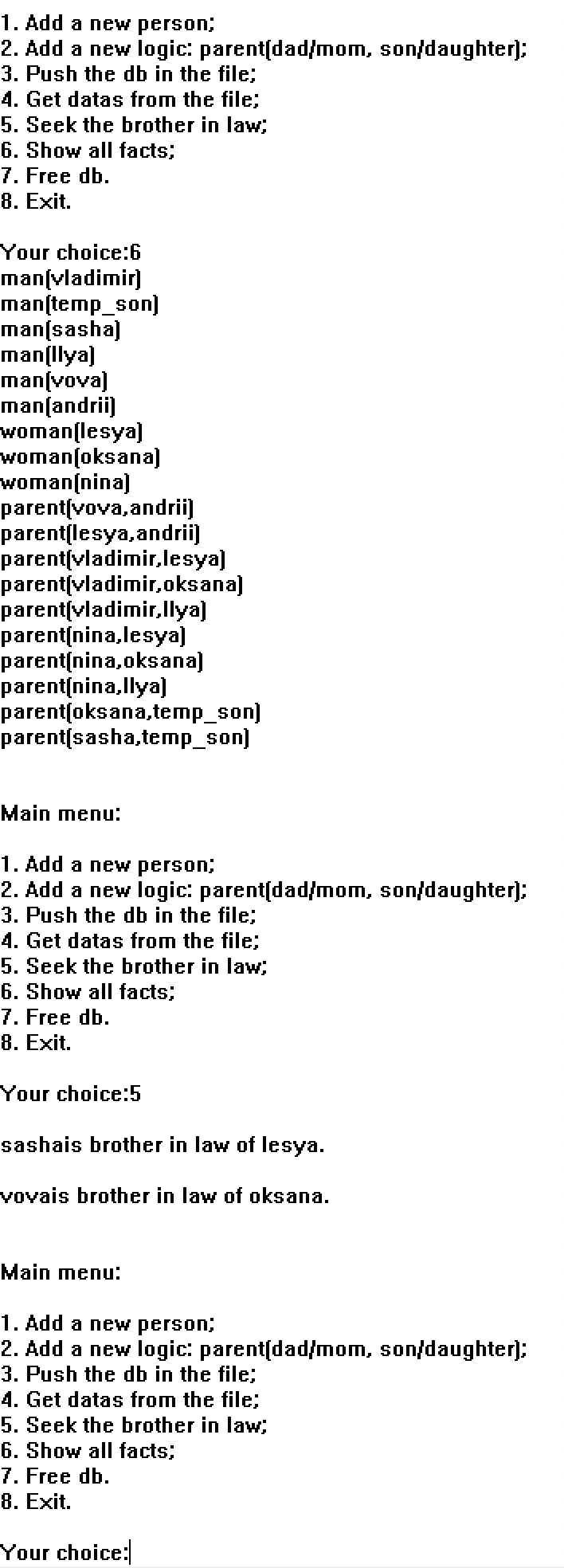


Рисунок 16 - Факти, індивідуальне завдання – пошук свояка.

**6. Усунення зауважень**

При виводі індивідуального завдання перед is додан пробіл.

switch("5"):-brother\_in\_law(X,Y),write("\n",X," is brother in law of ", Y,".\n"),fail.