**Лабораторна робота № 4**

**Використання динамічної БД Prolog. Функціональне програмування з використанням вбудованих предикатів Prolog**

**Вимоги до роботи:**

* перше завдання необхідно виконати з використанням БД Prolog без залучення предикатів генерації списків;
* у дру­гому завданні можна вико­рис­товувати дина­міч­ну БД, предикати генерації списків та інші засоби Prolog за винятком предикатів бібліотеки aggregate;
* третє завдання потрібно виконувати із залученням засобів бібліотеки aggregate.

**Варіант № 1**

У програмі визначені відношення parent(X, Y) (X — ім’я батька або матері, Y — ім’я дитини), male(X) (X — особа чоловічої статі), female(X) (X — особа жіночої статі), age(X, Y) (вік людини Х рівний Y).

1. Написати предикат ancestors(Person, Ancestors) для знаходження спис­ку предків заданої особи.
2. Написати правило для відношення grand\_sons(Name, List) для зна­ход­жен­ня спис­ку онуків (чо­ло­вічої статі) людини із заданим ім’ям, вік яких більший за се­редній вік усіх онуків цієї людини.
3. Написати правило для знаходження списку осіб, найбільший вік нащадків жіночої статі яких є мінімальним (враховувати лише тих осіб, у яких є нащадки жіночої статі).

**Варіант № 2**

Відношення department(Department\_name, Employees) містить дані про назви та списки працівників цих відділів (працівник може працювати у кількох від­ділах), відношення factory(Factory\_name, Department\_name) — дані про назви підприємств та їх відділів (на підприємстві може бути кілька відділів).

1. Написати правило для відношення fact\_emp(Factory, Employees) таким чи­ном, щоб список Employees містив усіх працівників фабрики без пов­то­рень.
2. Написати предикат most\_employed(Employees) для зна­ходження списку пра­ців­ни­ків, які працюють у найбільшій кількості відділів.
3. Написати правило для знаходження підприємств, середня кількість працівників у одному відділі яких є найменшою.

**Варіант № 3**

У програмі задані факти для відношень likes(Person, Object) (Person — осо­ба, якій подобається Object та creature(Object) — жива істота.

1. Написати правило для відношення likes\_things(Person, Things) для зна­ходження списку неживих об’єктів, які подобаються відповідній особі.
2. Визначити предикат likes\_exactly(Count, Persons) для знаходження спис­ку усіх осіб, яким подобається рівно Count живих істот.
3. Написати правило для знаходження осіб, відсоток унікальних неживих уподобань яких є найбільшим (унікальне уподобання — це об’єкт, який подобається рівно одній особі).

**Варіант № 4**

У програмі задані факти для відношення element(Point, Class), яке доз­во­ляє встановити належність точок до одного із класів (точки задаються за допомогою струк­тури point(X, Y)).

1. Написати предикат size(Class, Count) для знаходження кількості еле­мен­тів кожного класу.
2. Написати предикат compactest(List) для знаход­жен­ня списку класів най­мен­шого діаметру (діаметр класу — відстань між двома найбільш від­даленими точ­ками класу).

**Варіант № 5**

У програмі задані факти для відношень employee(Name, Job, Salary) (Name — прізвище, Job — посада, Salary — зарплата) та sex(Name, Gender) (Gender може при­й­мати значення male або female), які містять дані про пра­цівників під­при­ємства.

1. Написати предикат average\_salary(Job, Avg) для знаходження середньої зар­плати усіх працівників із заданою посадою
2. Написати предикат max\_salary(Gender, Max, Names) для знаходження спи­с­ку прізвищ працівників із найбільшою зарплатою серед працівників заданої статі.

**Варіант № 6**

У Пролог-програмі задані факти для відношення family\_info(Family, Members) про членів родин (Family — назва родини, Members — список імен членів родини).

1. Написати предикат count(Name, Count) для обчислення того, скільки разів зу­стрічається задане ім’я у списках членів родин.
2. Написати предикат largest\_families(List) для знаходження списку назв най­більш численних родин.

**Варіант № 7**

У програмі є факти для відношень "подобатися" — likes(Person, Object) (Person — особа, якій подобається Object та "предмет" — thing(Object).

1. Написати правило для відношення average\_likes(Persons, Avg) для зна­ходження середньої кількості живих істот (не предметів), які подобаються особам, які належать до списку Persons.
2. Визначити предикат prefered(Things) для знаходження списку усіх пред­ме­тів, які подобаються найбільшій кількості осіб.

**Варіант № 8**

У програмі визначені відношення parent(X, Y) (X — ім’я батька або матері, Y — ім’я дитини), male(X) (X — особа чоловічої статі), female(X) (X — особа жіночої статі), age(Person, Age).

1. Написати відношення big\_families(List) для знаходження списку усіх родин, у яких є принаймні три дитини. Елементи списку мають мати вигляд family(Father, Mother, Children), де Children — список дітей.
2. Написати предикат eldest\_grandchildren(Parent, Children) для зна­ход­ження списку найстарших онуків відповідної особи.

**Варіант № 9**

У Пролог-програмі задані факти для відношення employee(Name, Job, Age, Gender, Workshop) (Name — прізвище, Job — посада, Age — вік, Workshop — цех, Gender — стать може при­й­мати два значення: male або female), які містять дані про працівників під­при­ємства.

1. Написати предикат female\_jobs(Job, Females) для знаходження списку жі­нок, які обіймають відповідну посаду.
2. Визначити предикат most\_aged(Workshops) для знаходження цехів, у яких працює найбільша кількість працівників пенсійного віку.

**Варіант № 10**

У Пролог-програмі задані факти для відношень likes(Person, Thing) (Per­son — особа, якій подобається річ Thing) та price(Thing, Price) (Price — ціна).

1. Написати предикат most\_valuable(Person, Things) для знаходження списку найдорожчих речей, які подобаються заданій особі.
2. Визначити предикат most\_ressembling(Person1, Person2) для знаход­жен­ня осіб, які мають найбільше спільних уподобань.

**Варіант № 11**

У Пролог-програмі задані факти для відношення class(Name, Points), де Na­me — назва класу, Points — список точок, які належать до класу (точки задаються за допомогою струк­тури point(X, Y)).

1. Написати предикат classes(Point, Classes) для знаходження списку кла­сів, у які потрапляє задана точка.
2. Написати предикат distantest(Point, Names) для знаход­жен­ня списку назв класів, найбільш віддалених від заданої точки (відстань від точки *А* до класу — відстань між точкою *А* та найближчою до неї точкою класу).

**Варіант № 12**

У програмі задані факти для відношень department(Dep\_name, Employees) про назви відділів та список їх працівників та employee(Person, Age, Salary).

1. Написати предикат для відношення dep\_info(Dep\_name, Avg\_age, Total­\_salary) для знаходження середнього віку та сумарної зарплати працівників відділу.
2. Написати предикат single\_job\_holder(List) для зна­ходження списку пра­ців­ників, які працюють лише у одному відділі.

**Варіант № 13**

У програмі задані факти про рейси у вигляді trip(Departure, Arrival, Kind) (Departure — пункт відправлення, Arrival — пункт призначення, Kind — вид транспорту).

1. Написати предикат direct\_flies(Departure, Arrivals) для знаходження спис­ку пунктів призначення, у які напряму можна потрапити літаком із заданого пун­кту відправлення.
2. Визначити предикати minimal\_changes(Departure, Arrival) для зна­ход­ження мінімальної кількості пересадок, необхідних для того, щоб потрапити із пункту відправлення у пункт призначення одним видом транспорту.

**Варіант № 14**

У Пролог-програмі задані факти для відношень circle(Name, Members) про наз­ви і учасників гуртків (Name — назва гуртка, Members — список учасників гур­тка) та gender(Person, Sex).

1. Написати предикат female\_average(Value) для знаход­жен­ня середньої кількості гуртків, у яких беруть участь учасники-жінки.
2. Написати предикат activest(Members) для знаходження списку тих, хто бе­ре участь у найбільшій кількості гуртків.

**Варіант № 15**

У Пролог-програмі визначені відношення parent(X, Y) (X — ім’я батька або ма­тері, Y — ім’я дитини), male(X) (X — особа чоловічої статі), female(X) (X — особа жіночої статі).

1. Написати правило для відношення family(Father, Mother, ChildrenList).
2. Визначити відношення max\_descendants(Persons) для знаходження спис­ку осіб, які мають найбільшу кількість нащадків.