

Домашнє завдання з дисципліни «Функціональне та логічне програмування»

Виконав:

Студент групи СП-325

Щербина А.І.

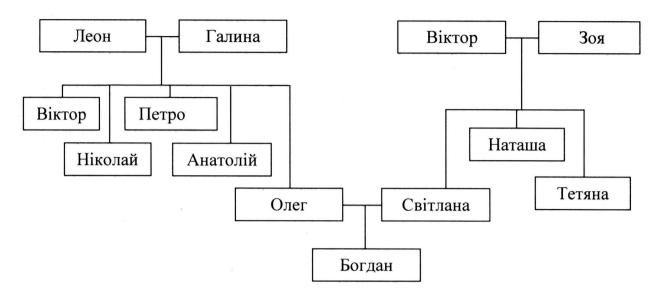
Перевірив:

Глазок О.М.

Завдання №1

Напишіть мовою Prolog програму з описом родинних відносин між членами заданої сім'ї. Додайте до програми правила, що визначають відношення "дядько", "тітка", "двоюрідний брат", "бабуся", "двоюрідна онука", тощо.

Знайдіть членів сім'ї, що задовольняють цим визначенням.



Код програми:

```
predicates
     male(symbol)
     female(symbol)
     parent(symbol, symbol)
     father(symbol,symbol)
     mother(symbol, symbol)
     daughter(symbol,symbol)
     son(symbol,symbol)
     sister(symbol,symbol)
     brother(symbol,symbol)
     aunt(symbol,symbol)
     uncle(symbol,symbol)
     cousin male(symbol,symbol)
                                   %двоюродный брат
     grandmother(symbol,symbol)
     cousin granddaughter(symbol,symbol)
                                           %двоюродная внучка
     oneparent(symbol,symbol)
                                   %имеют общего родителя
clauses
     male(leon). male(petro). male(anatoliy).
     male(viktor). male(nikolaj). male(oleg).
     male(bogdan).
     female(galina). female(svitlana). female(zoya).
     female(natasha). female(tetyana).
     parent(leon,petro). parent(leon,viktor1).
     parent(leon, nikolaj). parent(leon, anatoliy).
     parent(leon,oleg). parent(galina,viktor1).
     parent(galina,petro). parent(galina,nikolaj).
     parent(galina,anatoliy). parent(galina,oleg).
     parent(viktor2, natasha). parent(viktor2, tetyana).
```

```
parent(viktor2,svitlana). parent(zoya,natasha).
parent(zova,tetvana). parent(zova,svitlana).
parent(oleg,bogdan). parent(svitlana,bogdan).
oneparent(X,Y):-
                    %X и Y имеют общего родителя
     parent(Z,X),parent(Z,Y),X<>Y.
father(X,Y):-
     parent(X,Y), male(X).
mother(X,Y):-
     parent(X,Y), female(X).
son(X,Y):-
     parent(Y,X), male(X).
daughter(X,Y):-
     parent(Y,X), female(X).
sister(X,Y):-
     oneparent(X,Y), female(X).
brother(X,Y):-
     oneparent(X,Y), male(X).
aunt(X,Y):-
     parent(Z,Y), sister(X,Z).
uncle(X,Y):-
     parent(Z,Y), brother(X,Z).
cousin male(X,Y):-
     parent(Z,X), parent(K,Y), oneparent(Z,K), male(X).
grandmother(X,Y):-
     parent(Z,Y), mother(X,Z).
cousin granddaughter(X,Y):-
     daughter(X,Z), parent(K,Z), oneparent(Y,K).
```

Результати роботи програми:

```
Goal: cousin_male(X,Y)
No Solution
Goal: grandmother(X,Y)
                                                     Goal: sister(X,Y)
X=galina, Y=bogdan
                                                     X=natasha, Y=tetyana
X=zoya, Y=bogdan
                                                      K=natasha, Y=svitlana
2 Solutions
                                                     X=tetyana, Y=natasha
                                                     X=tetyana, Y=svitlana
Goal: cousin_granddaught Goal: aunt(X,Y)
                                                     X=svitlana, Y=natasha
X=svitlana, Y=tetyana
er(X,Y)
                           X=natasha, Y=bogdan
No Solution
                           X=tetyana, Y=bogdan
```

Рис. 1 – Пошук деяких родинних зв'зків.

Завдання №2

Поясніть поняття «Інтелектуальні задачі». Наведіть приклади інтелектуальних задач. Поясніть, чому декларативний підхід до програмування вважається придатним для розв'язання інтелектуальних задач.

Суть інтелектуальних задач полягає у виявленні знань в базах даних, необхідних для прийняття рішень у різних сферах. Важливим призначенням методів рішення таких задач ϵ

наочне представлення результату вирішення без поглиблення в реалізацію його алгоритму. Для таких задач підходить декларативний метод програмування, в якому користувач описує який результат він хоче отримати, а не як досягти бажаного результату.

В імперативному програмуванні для отриманя результату, мені потрібно написати:

- постав сковорідку на вогонь;
- візьми два яйця (курячі);
- нанеси удар ножем по кожному;
- вилий вміст на сковорідку;
- викинь шкаралупу;

А в декларативному:

- приготуй яєчню

А далі програма сама шукає шляхи розв'язку задачі на основі тих фактів, що ϵ в її базі знань.

Завдання №3

Мовою Prolog напишіть програму на основі фактів та правил у відповідності з індивідуальним варіантом предметної області.

Предметна область «Житло».

Описати об'єкти предметної області: будинок; квартира; сім'я.

Описати запити:

- підрахувати кількість приватних будинків;
- вивести відомості про 2-кімнатні квартири (адреса, власник, площа квартири, кількість зареєстрованих осіб, тощо), в порядку зменшення житлової площі на одну особу, що проживає у даній квартирі;
- знайти середню вартість 1 кв. м. житла, зареєстрованого в базі знань програми.

Код програми:

```
domains
      area, persons, rooms, number, price = integer
      address, owner = symbol
predicates
      family(symbol,symbol,symbol)
      house(number,address,owner,area,persons,rooms,price)
       apartment(number,address,owner,area,persons,rooms,price)
      houses num(integer)
      midprice(integer,integer,integer)
      h price(integer,integer)
      a price(integer,integer)
      max(integer,integer,integer,integer,integer)
      maxtwo(integer,integer,integer)
      tworooms(integer,integer,integer)
      lifearea
      areaforperson(integer)
clauses
      family(tom, alice, bob, leon).
      family(adam,abby," "," ").
      family(jason,lisa,derek," ").
```

```
family(philip, holly, otto, emily).
       family(robert,ann," "," ").
       house(1,"117 Groove Street",tom,130,4,5,910).
       house(2,"53 Sullivan Street",adam,125,2,4,850).
       house(3,"13 Baker Street", rose, 110, 4, 4, 835).
       house(4,"38 King's Road Street, Apt 25", jason, 140, 3, 4, 946).
       house(5, "8 Carnaby Street, Apt 104", holly, 115, 4, 4, 890).
       house(6,"74 Downing Street, Apt 4", robert, 95, 2, 3, 789).
       apartment(1, "38 King's Road Street, Apt 25", jason, 140, 3, 2, 946).
       apartment(2,"8 Carnaby Street, Apt 104",holly,115,4,4,890).
       apartment(3,"74 Downing Street, Apt 4", robert, 130, 2, 2, 789).
       apartment(4,"117 Groove Street",tom,130,4,5,910).
       apartment(5, "53 Sullivan Street", adam, 125, 2, 2, 850).
       apartment(6, "13 Baker Street", rose, 110, 4, 4, 835).
       tworooms(1,3,5).
       houses_num(N):- not(house(N,_,_,_,_,_)).
       houses num(N):-
              house(N,_,_,_,_),N1 = N + 1,houses_num(N1),
              write("Kolichestvo domov = ",N),nl,fail.
       lifearea:-
              tworooms(N1,N2,N3),apartment(N1,_,_,Area1,Persons1,_,_),
              apartment(N2,_,_,Area2,Persons2,_,_),
              apartment(N3,_,_,Area3,Persons3,_,_),
              K1 = Area1/Persons1, K2 = Area2/Persons2, K3 =
Area3/Persons3,
              \max(K1, K2, K3, N1, N2, N3).
       \max(X,Y,Z,N1,N2,N3) :-
              X>=Y,X>=Z,apartment(N1,A,O,_,_,R,_),
              write("Address: ",A),nl,
              write("Owner: ",0),nl,
              write("Rooms: ",R),nl,
              write("Area for person: ",X),nl,
              maxtwo(Y,Z,N2,N3).
       max(X,Y,Z,N1,N2,N3) :-
              Y>=X,Y>=Z,apartment(N2,A,O,_,_,R,_),
              write("Address: ",A),nl,
              write("Owner: ",0),nl,
              write("Rooms: ",R),nl,
              write("Area for person: ",Y),nl,
              maxtwo(X,Z,N1,N3).
       max(X,Y,Z,N1,N2,N3) :-
              Z>=X,Z>=Y,apartment(N3,A,O,_,_,R,_),
              write("Address: ",A),nl,
              write("Owner: ",0),nl,
              write("Rooms: ",R),nl,
              write("Area for person: ",Z),nl,
              maxtwo(X,Y,N1,N2).
       maxtwo(X,Y,N2,N3) :-
```

family(john, rose, gina, ben).

```
X>=Y,apartment(N2,A1,O1,_,_,R1,_),apartment(N3,A2,O2,_,_,R2,_),
              write("Address: ",A1),n1,
              write("Owner: ",01),n1,
              write("Rooms: ",R1),n1,
              write("Area for person: ",X),nl,
              write("Address: ",A2),n1,
              write("Owner: ",02),n1,
              write("Rooms: ",R2),n1,
              write("Area for person: ",Y).
       maxtwo(X,Y,N2,N3) :-
              apartment(N3,A1,O1,_,_,R1,_),apartment(N2,A2,O2,_,_,R2,_),
              write("Address: ",A1),nl,
              write("Owner: ",01),nl,
              write("Rooms: ",R1),n1,
              write("Area for person: ",Y),nl,
              write("Address: ",A2),n1,
              write("Owner: ",02),n1,
              write("Rooms: ",R2),n1,
              write("Area for person: ",X),nl.
       h price(0,0).
       h price(N,Sum):-
              N1 = N - 1, h_{price}(N1,Sum1), house(N,_,_,Kv_M,_,_,P),
              Sum = Sum1 + P/Kv M.
       a price(0,0).
       a price(N,Sum):-
              N1 = N - 1, a_price(N1,Sum1),apartment(N,_,_,Kv_M,_,_,P),
              Sum = Sum1 + P/Kv M.
       midprice(N h,N a,Mid):-
              h_price(N_h,Sum_h),a_price(N_a,Sum a),Mid = (Sum h +
Sum a)/12.
       areaforperson(N):-
              apartment(N1, , ,A,P,2, ),N1>N, areaforperson(N1),K = A/P,
write(K), nl, fail.
       areaforperson(_):- !.
Код програми з використанням списків, динамічної бази даних, експертної
системи, інтерфейсом користувача (у псевдографічному вигляді):
domains
       rlist = real*
       slist = symbol*
database - zhitlo
       family(slist) %семья
       house(rlist,slist)
                           %дом
       apartment(rlist,slist)
                                   %квартира
predicates
              %вызов меню
       process(integer)
                            %вызов операции базы данных
```

```
process2(integer)
                             %вызов операции
       houses num(integer)
                                    %Задание 1:найти количество домов,
запрос: houses num(1)
       midprice(integer,integer) %Задание 3: найти среднюю цену
жилья за кв.м., запрос: midprice(6,6,N)
       h price(integer,integer)
                                    %поиск суммарной цены домов
       a price(integer,integer)
                                    %поиск суммарной цены квартир
       areaforperson(integer,rlist,rlist) %поиск площади на 1 человека
                     %Задание 2: общая ф-я (поиск, сортировка, вывод) для
       afp
2-комнатных квартир
       writeList(rlist,rlist)
                                    %вывод данных из списка
       append(rlist,rlist,rlist)
                                    %добавить элемент в список
       appends(slist,slist,slist) %добавить элемент в список
       tworooms(rlist)
                                                  %номера 2-комнатных
квартир
       reverse(rlist,rlist) %инвертировать список
       add(rlist,real,rlist,rlist,real,rlist) %функции для сортировки,
       sort(rlist,rlist,rlist,rlist)
                                                          %работающиее
вместе
       addfamily(integer,slist)
                                                  %формирование списка
имен семьи для добавления/удаления
       expertsvs
       addobj(integer)
       delobj(integer)
       findobj(integer)
clauses
       family([tom,alice,bob,leon]).
       family([adam,abby]).
       family([jason,lisa,derek]).
       family([john,rose,gina,ben]).
       family([philip,holly,otto,emily]).
       family([robert,ann]).
       house([1,130,4,5,910],["117 Groove St",tom]).
       house([2,125,2,4,850],["52 Sullivan St",adam]).
house([3,110,4,4,835],["13 Baker St",rose]).
       house([4,140,3,4,946],["33 King's Road St",jason]).
       house([5,115,4,4,890],["9 Carnaby St",holly]).
       house([6,95,2,3,789],["74 Downing St",robert]).
       apartment([1,140,3,2,946],["38 King's Road St",jason]).
       apartment([2,115,4,4,890],["8 Carnaby St",holly]).
       apartment([3,130,2,2,789],["74 Downing St",robert]).
       apartment([4,130,4,5,910],["207 Groove St",tom]).
       apartment([5,125,2,2,850],["53 Sullivan St",adam]).
       apartment([6,110,4,4,835],["16 Baker St",rose]).
       addfamily(0,[]).
       addfamily(N,NameList):-
              N1 = N - 1, addfamily(N1, NewNameList),
              write("\179 Name: "),readln(Name),
              appends(NewNameList, [Name], NameList).
```

```
tworooms ([1,3,5]).
      houses num(N):- not(house([N| ], )).
      houses num(N):-
             house([N|], ), N1 = N + 1, houses_num(N1),
             write("\179 Houses number = ",N),write(" \179"),nl,
             write("\
96\196\196\217"),nl,fail.
      h price(0,0).
      h price(N,Sum):-
             N1 = N - 1, h price(N1,Sum1),house([N,KV, , ,Price], ),
             Sum = Sum1 + Price/KV.
      a price(0,0).
      a price(N,Sum):-
             N1 = N - 1, a price(N1,Sum1),apartment([N,KV, , ,Price], ),
             Sum = Sum1 + Price/KV.
      midprice(N h,N a,Mid):-
             h price(N h,Sum h),a price(N a,Sum a),Mid = (Sum h +
Sum a)/12.
       reverse([],[]).
       reverse([X|L1], L2):-reverse(L1, L3),
      append( L3, [X], L2).
      add([],X,[X],[],Y,[Y]).
       add([H|[]],X,L,[NH|[]],NX,NL):-X>=H,L=[X|[H]],NL = [NX|[NH]];
    X<H, L=[H|[X]], NL=[NH|[NX]],!
       add([H|T],X,L,[NH|NT],NX,NL):-X>=H,L=[X|[H|T]],NL=[NX|[NH|NT]],!;
    X < H, add(T, X, L2, NT, NX, NL2), L=[H|L2], NL=[NH|NL2], !.
       sort([H|[]],L,[NH|[]],NL):-add([],H,L,[],NH,NL).
       sort([H|T],L,[NH|NT],NL):-sort(T,S,NT,NS),add(S,H,L,NS,NH,NL).
       append([], L, L).
       append([X|L1], L2, [X|L3]):- append(L1, L2, L3).
       appends([], L, L).
       appends([X|L1], L2, [X|L3]):- appends(L1, L2, L3).
      writeList([],[]):-write(' ').
      writeList([H|T],[H2|T2]):-apartment([H|],[DH,DT]),
      write("\179 Ad: ",DH),nl,
      write("\179 Owner: ",DT),nl,
      write("\179 AFP: ",H2),nl,writeList(T,T2).
      afp:-
tworooms(RNlist), areaforperson(3, TRarea, RNlist), reverse(RNlist, Nlist),
sort(TRarea, Sortlist, Nlist, SortNlist), writeList(SortNlist, Sortlist).
```

```
areaforperson(0,[],[]).
areaforperson(N,List,[H|T]):-
       N1 = N - 1, areaforperson(N1, Newlist, T),
       apartment([H,A,P,2,], ), K = A/P, append(Newlist, [K], List).
addobj(1):-
       write("\179 Count of persons: "),
       readint(N), N>0,
       addfamily(N,NameList),
       assert(family(NameList)),
       write("\179 Family added!"),nl.
addobj(2):-
       write("\179 Index: "),readint(I),
      write("\179 Area: "),readint(A),
       write("\179 Persons: "),readint(P),
       write("\179 Rooms: "),readint(R),
       write("\179 Price: "),readint(Pr),
       write("\179 Address: "),readln(Ad),
       write("\179 Owner: "),readln(Ow),
       assert(house([I,A,P,R,Pr],[Ad,Ow])),
       write("House added!"),nl.
addobj(3):-
       write("\179 Index: "),readint(I),
       write("\179 Area: "),readint(A),
       write("\179 Persons: "),readint(P),
       write("\179 Rooms: "),readint(R),
       write("\179 Price: "),readint(Pr),
      write("\179 Address: "),readln(Ad),
       write("\179 Owner: "),readln(Ow),
       assert(apartment([I,A,P,R,Pr],[Ad,Ow])),
       write("Apartment added!"),nl.
delobj(1) :-
       write("\179 Count of persons: "),
       readint(N),
       addfamily(N,NameList),
       retract(family(NameList)),
       write("\179 Family deleted!"),nl.
delobj(2) :-
       write("\179 Index of house: "),readint(I),
       retract(house([I,_,_,_,],[_,_])),
       write("\179 House deleted!"),nl.
delobj(3) :-
       write("\179 Index of apt.: "),readint(I),
       retract(apartment([I,_,_,_,],[_,_])),
       write("\179 Apartment deleted!"),nl.
findobj(1) :-
       write(" Count of persons: "),
       readint(N),
       addfamily(N,NameList),
       family(NameList),
       write("\179 Family found!"),nl.
findobj(2) :-
       write(" Address: "),readln(Ad),
```

```
house([I,A,P,_,_],[Ad,Ow]),
         write("\179 Index: ",I),nl,
         write("\179 Area: ",A),nl,
         write("\179 Persons: ",P),nl,
         write("\179 Owner: ",Ow),nl,
         write("\179 House found!"),nl.
    findobj(3) :-
         write(" Address: "),readln(Ad),
         apartment([I,A,P,_,_],[Ad,Ow]),
         write("\179 Index: '
                       ',I),nl,
         write("\179 Area: ",A),nl,
         write("\179 Persons: ",P),nl,
         write("\179 Owner: ",Ow),nl,
         write("\179 Apartment found!"),nl.
    expertsys :-
         write("\
96\196\196\191"),nl,
         write("\179
                    Expert System \179"),nl,
         write("\179What you want to do?\179"),nl,
         write("\
96\196\196\180"),nl,
         write("\179 N \179 process \179"),nl,
         write("\
96\196\196\180"),nl,
         write("\179 1 \179 add smth
                                 \179"),nl,
         write("\179 2 \179 del smth
                                 179, nl,
         write("\179 3 \179 find smth
                                 \179"),nl,
         write("\179 4 \179 save database \179"),nl,
         write("\179 5 \179 download db
                                 \179"),nl,
         write("\179 6 \179 exit
                                 \179"),nl,
         write("\
96\196\196\217"),nl,
         readint(X),process(X).
    process(1) :-
         write("\
96\196\196\191"),nl,
         write("\179 Add something
                              \179"),nl,
         write("\179 What to add?
                              \179"),nl,
         write("\
96\196\196\180"),nl,
         write("\179 N \179 object \179"),nl,
         write("\
96\196\196\180"),nl,
         write("\179 1 \179 family
                                 \179"),nl,
         write("\179 2 \179 house
                                 179"),nl,
```

```
write("\179 3 \179 apartment \179"),nl,
       write("\
96\196\196\217"),nl,
       write("Enter: "),readint(N),
       addobj(N).
    process(2) :-
       write("\
96\196\196\191"),nl,
       write("\179 Delete something
                         \179"),nl,
       write("\179 What to delete?
                         179, nl,
       write("\
96\196\196\180"),nl,
       write("\179 N \179 object \179"),nl,
       write("\
96\196\196\180"),nl,
       write("\179 1 \179 family
                           \179"),nl,
       write("\179 2 \179 house
                           \179"),nl,
       write("\179 3 \179 apartment
                           \179"),nl,
       write("\
96\196\196\217"),nl,
       write("Enter: "),readint(N),
       delobj(N).
    process(3) :-
       write("\
96\196\196\191"),nl,
                         \179"),nl,
       write("\179 Find something
       write("\179 What to find?
                         \179"),nl,
       write("\
96\196\196\180"),nl,
       write("\179 N \179 object
                          \179"),nl,
       write("\
96\196\196\180"),nl,
       write("\179 1 \179 family
                           \179"),nl,
       write("\179 2 \179 house
                           \179"),nl,
       write("\179 3 \179 apartment
                           179, nl,
       write("\
96\196\196\217"),nl,
       write("Enter: "),readint(N),
       findobj(N).
    process(4) :-
       write("Save database"),nl,
       write("\
96\196\196\191"),nl,
```

```
save(zhitlo,zhitlo),
        write("\179Database saved
                          \179"),nl,
        write("\179 to file: ZHITLO
                          \179"),nl,
        write("\
96\196\196\217"),nl.
    process(5) :-
        write("Download database").nl.
        write("\
96\196\196\191"),nl,
        consult(zhitlo,zhitlo),
        write("\179 Database downloaded\179"),nl,
        write("\179 from file: ZHITLO \179"),nl,
        write("\
96\196\196\217"),nl.
    process(6) :- !.
    menu :-
        write("Choose a process: "),nl,
        write("\
96\196\196\191"),nl,
        write("\179 N \179 process \179"),nl,
        write("\
96\196\196\180"),nl,
        write("\179 1 \179 houses number \179"),nl,
        write("\179 2 \179 mid price
                            179"),nl,
        write("\179 3 \179 sort two-roomed\179"),nl,
        write("\179 \179 flats by AFP \179"),nl,
        write("\
96\196\196\217"),nl,
        readint(X),process2(X).
    process2(1) :-
        write("\
96\196\196\191"),nl,
        houses_num(1),nl.
    process2(2) :-
        write("\
96\196\196\191"),nl,
        midprice(6,6,N),
        write("\179 Mid price = ",N),write(" \179"),nl,
        write("\
96\196\196\217"),nl.
    process2(3) :-
        write("Sorted 2-roomed flats"),nl,
```

Результати роботи програми:

96\196\196\217"),nl.

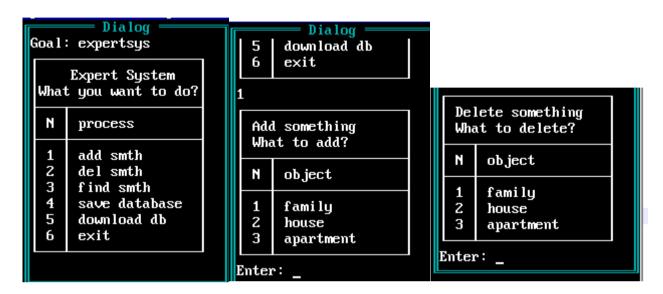


Рис. 2 – Головне та побічні меню експертної системи.

```
Enter: 2
| Index: 7
| Area: 113
| Persons: 3
| Rooms: 5
| Price: 1202
| Name: artem
| Name: roman
| Family added!
| Enter: 2
| Index: 7
| Area: 113
| Persons: 3
| Rooms: 5
| Price: 1202
| Address: 4 Baker St
| Owner: artem
| House added!
```

Рис. 3 – Інтерфейс додавання об'єктів.

```
Enter: 1
| Count of persons: 2
| Name: adam
| Name: abby
| Family deleted!
| Yes
```

Рис. 4 – Інтерфейс видалення об'єктів (видалення квартири за індексом).

```
Count of persons: 2
| Name: robert | Address: 74 Downing St | Index: 3 | Area: 130 | Persons: 2 | Owner: robert | Apartment found! | Yes | Yes | Yes | Address: 74 Downing St | Index: 3 | Area: 130 | Persons: 2 | Owner: robert | Apartment found! | Yes |
```

Рис. 5 – Інтерфейс пошуку об'єктів (пошук квартири за адресою).

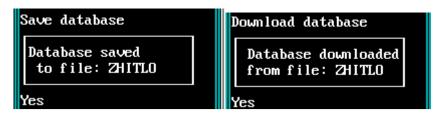


Рис. 6 – Збереження та завантаження бази даних.

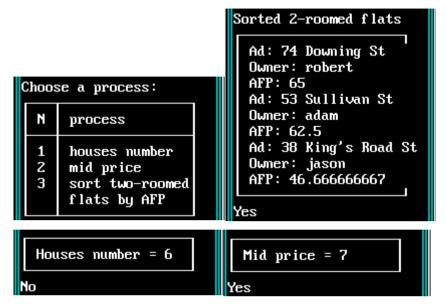


Рис. 7 – Меню запитів, що були поставлені у завданні, та резулат їх виконання.