**Лабораторна робота № 3**

**Тема:** Рекурсія та ітерація на мові Visual Prolog

**Мета:** набути практичних навичок в організації рекурсії та ітерації на мові Visual Prolog.

**Короткі теоретичні відомості**

Рекурсивна процедура – це процедура, що викликає сама себе до тих пір, доки не буде дотримана деяка умова, яка зупинить рекурсію. Таку умову називають граничною. Рекурсивне правило завжди складається принаймні з двох частин, одна з яких є нерекурсивною. Вона і визначає граничну умову. У рекурсивній процедурі немає проблеми запам’ятовування результатів її виконання, тому що будь-які обчислені значення можна передавати з одного виклику в інший як аргументи рекурсивно викликаного предикату. Рекурсія є33 ефективним способом розв’язування задач, що містять у собі підзадачу такого самого типу.

**Хід роботи:**

1. Обчислити суму 1+2+3+…+N.

2. Підрахувати суму ряду цілих парних чисел від 2 до N.

3. Обчислити суму ряду цілих непарних чисел від от 1 до n.

4. Знайти значення добутку: 2\*4\*6\*...\*26.

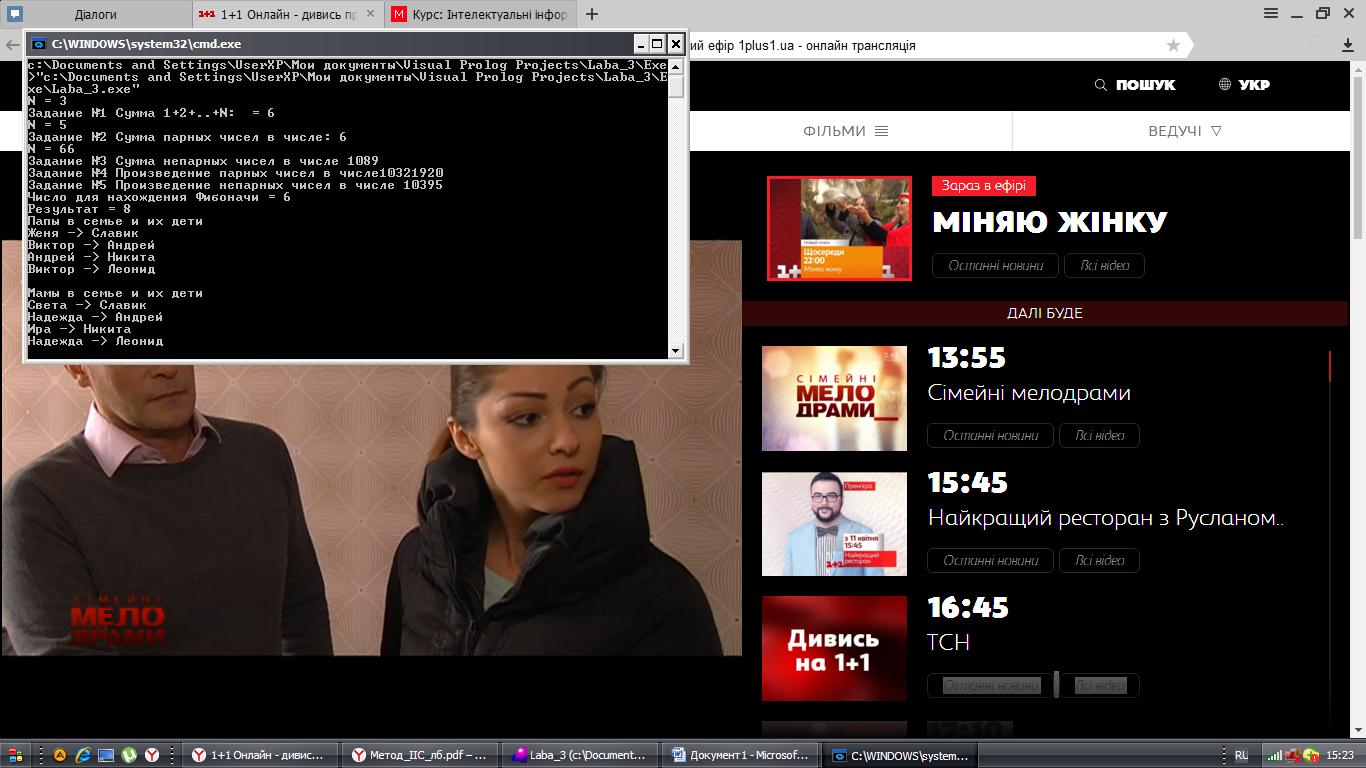
5. Знайти значення добутку: 1\*3\*5\*...\*11

6. Обчислити значення n-го члена ряду Фібоначчі: f(0)=0, f(1)=1, f(n)=f(n-1)+f(n-2).

7. Використовуючи базу даних і правило предок з прикладу 2, скласти правило для визначення всіх нащадків-чоловіків.

8. Використовуючи базу даних і правило предок з прикладу 2, скласти правило для визначення всіх нащадків-чоловіків.

implement main  
    open core, console  
  
class facts -relatives  
    parent: (string, string).  
    man: (string).  
    woman: (string).  
  
class predicates  
    suma: (integer, integer) nondeterm anyflow.  
    suma1: (integer, integer) nondeterm anyflow.  
    suma2: (integer, integer) nondeterm anyflow.  
    dob1: (integer, integer) nondeterm anyflow.  
    dob2: (integer, integer) nondeterm anyflow.  
    fib:(integer, integer) nondeterm anyflow.  
    parent\_man: (string, string) nondeterm anyflow.  
    parent\_woman: (string, string) nondeterm anyflow.  
  
clauses  
    parent\_man(X, Y):- parent(X, Y), man(X).  
    parent\_woman(X, Y):-parent(X, Y), woman(X).  
    suma(0, 0):-!.  
    suma(N, S):-N1 = N - 1, suma(N1, S1), S = N + S1.  
    suma1(2, 2):-!.  
    suma1(N, S):-N mod 2 = 1,  suma1(N - 1, S), !.  
    suma1(N, S + N):-suma1(N - 2, S).  
    suma2(1, 1):-!.  
    suma2(N, S):-N mod 2 = 0, suma2(N - 1, S),!.  
    suma2(N, S + N):-suma2(N - 2, S).  
    dob1(2, 2):-!.  
    dob1(N, M):-N mod 2 = 1, dob1(N - 1, M), !.  
    dob1(N, M \* N):-dob1(N - 2, M).  
    dob2(1, 1):-!.  
    dob2(N, M):-N mod 2 = 0, dob2(N - 1, M), !.  
    dob2(N, M \* N):- dob2(N - 2, M).  
    fib(0,0):-!.  
    fib(1,1):-!.  
    fib(X,Y):-fib(X - 1, Y1), fib(X - 2, Y2), Y = Y1 + Y2.  
    run() :-  
        init(),  
file::consult("famil.txt", relatives),  
        write("N = "),  
            N = read(),  
            suma(N, S),  
            write("Задание №1 Сумма 1+2+..+N:  = ", S),  
            fail;  
  
        nl,  
        write("N = "),  
            N = read(),  
         suma1(N, S1),  
            write("Задание №2 Сумма парных чисел в числе: ", S1),  
            fail;  
  
        nl,  
        write("N = "),  
            N = read(),  
            suma2(N, S1),  
            write("Задание №3 Сумма непарных чисел в числе ", S1),  
            fail;  
  
        nl,  
            dob1(16, M),  
            write("Задание №4 Произведение парных чисел в числе", M),  
            fail;  
  
        nl,  
            dob2(11, M),  
            write("Задание №5 Произведение непарных чисел в числе ", M),  
            fail;  
        nl,  
            write("Число для нахождения Фибоначи = "),  
            X = read(),  
            fib(X, Y),  
            write("Результат = ", Y),  
            fail;  nl,  
             write("Папы в семье и их дети"),nl,  
            parent\_man(X, Y),  
                write(X, ' -> ', Y), nl,  
            fail;  
        nl,  
        write("Мамы в семье и их дети"),nl,  
            parent\_woman(X, Y),  
                write(X, ' -> ', Y), nl,  
            fail;  
  
        \_ = readLine(),  
        \_ = readLine().  
  
end implement main  
  
goal  
    console::run(main::run).

****