

```

predicates
wr_list(list)
write_list(list)
new_list(list, list)
result
clauses
/*Вивід елементів списку. Після кожного елемента виводиться
кома*/
wr_list([ ]).
wr_list([H|T]):-write(H), write(','), wr_list(T).
/* Вивід списку в стандартній формі */
write_list(L):- write('['), wr_list(L),
write(']').
/* Перетворення вхідного списку */
new_list([ ],[ ]).
new_list([H1|T1],[H2|T2]):-H2 is H1*2,new_list(T1, T2).
result:- L=[9,6,0,-5,9], new_list(L,L1),
write('Вхідний список L='), write_list(L),nl,nl,
write('Новий список L1='), write_list(L1).
goal
result.

```

#### Зміст завдань по лабораторній роботі.

1. Приклад програми, яка реалізує суму двох чисел.

predicates

suma(integer, integer, integer)

clauses

suma(X, Y, Z):-Z is X+Y.

Написати програму в якій будуть реалізовані всі арифметичні операції для цілих і дійсних чисел.

2. Написати програму для знаходження більшого із 4 чисел.

3. Написати програму, яка виводить n-й член ряду Фібоначчі (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...).

4. Написати процедуру для обчислення  $x^n$  за допомогою одного тільки множення.

5. Написати процедуру для обчислення суми  $f(x,n) = 1 + x + x^2 + \dots + x^n$ .

#### ВАРІАНТИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

##### Варіант 1

1. Просумувати цілі додатні числа, які слідують одне за одним з кроком d і закінчуються числом n. Значення d і n вводяться по запиті з екрана дисплея. (Наприклад, n=11, d=3, сумуються числа 11+8+5+2=26). В випадку d>n сума дорівнює n.

2. Видалити початок списку до заданого елемента X (включно). Видалити кінець списку, починаючи з заданого елемента X.

3. Написати програму сортування списку бульбашковим методом, результатом роботи програми повинен бути вхідний і відсортований список.

```

maximum(real,real,real)
CLAUSES
add(X,Y):-Z is X+Y,write('Sum= '), write(Z), nl.
fadd(X,Y):-Z is X+Y,write('FSum= '),write(Z),nl.
maximum(X,X,X).
maximum(X,Y,X):- X>Y.
maximum(X,Y,Y):- X<Y.

```

**ПРИКЛАД.** Написати програму на мові Пролог, яка друкує суму всіх цифр введенного з клавіатури числа.

Для вирішення даної задачі скористаємося двома предикатами. Предикат `сума/2` має своїм першим аргументом число, сума цифр якого, є його другим аргументом. Другий предикат - `друк_суми/0` - запитує число, викликає предикат `сума/2` і друкує отриманий результат.

```

сума(X,Y):- integer(X), X<10, Y is X.
сума(X,Y):- integer(X), X1 is X//10, сума(X1,Y1),
               Z is X mod 10, Y is Y1+Z.
друк_суми:- write('Введіть число (в кінці крапка): '),
               read(X), nl, сума(X,Y),
               write('Сума цифр числа '), write(X),
               write(' рівна '), write(Y), nl.

```

#### МЕТОД РЕКУРСІЇ

Потужним засобом програмування в Пролозі є рекурсія. Рекурсія – це визначення деякого відношення через самого себе. Так як в Пролозі відсутні оператори цикла, то рекурсія служить основним засобом програмування циклічних процесів.

Класичим прикладом рекурсії є обчислення факторіала  $F=N! = 1*2*3*...*(N-1)*N$ . Цей вираз можна записати так:  $N!=(N-1)!*N$ . Останній вираз – приклад рекурсивних обчислень, тобто кожний наступний результат отримується шляхом використання попереднього результату. Для того, щоб рекурсивний метод був результативним, він повинен в кінці кінців привести до задачі, яка розв'язується безпосередньо. Розв'язати її дозволяють твердження, які називають граничними умовами(в нашому прикладі це  $0!=1$ ).

/\* Обчислення факторіала  $F=N!$  \*/

domains

n,f=real

predicates

factorial(n,f)

result

clauses

factorial(1,1).

factorial(N,F):-N>0,N1=N-1, factorial(N1,F1), F is F1\*N.

result:- write('Введіть число N'), nl,

write('N= '),read(N),N>0, factorial(N,F),

write(N,'!=',F), result.

goal

write('Вихід із програми при  $N<=0$ '),nl, nl, result.