

**Искусственный интеллект. Лабораторная работа №1.  
Факты, правила, вопросы на языке ПРОЛОГ.**

Построить произвольное генеалогическое дерево, содержащее не менее 4-х уровней(поколений). Описать это дерево на языке ПРОЛОГ с помощью следующих отношений:

- родитель(X,Y).      */\* X является родителем Y \*/*
- мужчина(X).        */\* X – мужчина \*/*
- женщина(X).       */\* X – женщина \*/*

На основе этих трех отношений, написать правила для других различных родственных отношениях:

1. мать(X,Y)        */\* X является матерью для Y \*/*
2. отец(X,Y)
3. брат(X,Y)
4. сестра(X,Y)
5. ребенок(X,Y)   */\* X является ребенком для Y \*/*
6. дочь(X,Y)
7. сын(X,Y)
8. дедушка(X,Y)
9. бабушка(X,Y)
10. дядя(X,Y)
11. племянник(X,Y)
12. предок(X,Y)   */\* X является предком для Y – рекурсивное правило \*/*
13. отпрыск(X,Y)
14. двоюродный\_брат(X,Y)
15. двоюродная\_сестра(X,Y)
16. есть\_дети(X)   */\* X имеет детей \*/*
17. является\_дедушкой(X)
18. является\_бабушкой(X)
19. имеет\_братьев(X)
20. имеет\_двоюродных\_братьев(X)
21. др.

Сформулировать различные вопросы над программой.

**Для выполнения данной работы необходимо определить 7 различных отношений из вышеперечисленных.**

**Искусственный интеллект. Лабораторная работа №2.**  
**Рекурсивные функции.**

1. Определить предикат **fact(N,Rez)**, который вычисляет факториал натурального числа.
2. Определить предикаты **suma(A,B,Rez)** и **minus(A,B,Rez)**, которые находят с помощью рекурсии сумму и разность двух натуральных чисел.
3. Определить предикат **produs(A,B,Rez)**, который вычисляет с помощью рекурсии произведение двух натуральных чисел.
4. Определить предикат **nod(A,B,Rez)**, который рекурсивно находит наибольший общий делитель двух натуральных чисел, применяя алгоритм Евклида.
5. Определить предикат **expo(X,N,XlaN)**, который рекурсивно возводит произвольное число в натуральную степень.
6. Определить предикат **mod1(A,B,Rez)**, который рекурсивно находит остаток от деления одного натурального числа на другое.
7. Определить предикаты **succesor(A,B)** и **predecesor(A,B)**, которые рекурсивно определяют, если одно натуральное число является следующим или предыдущим для другого.
8. Определить предикат **div1(A,B,Rez)**, который рекурсивно находит целую часть от деления одного натурального числа на другое.
9. Определить рекурсивный предикат, который находит сумму первых **n** натуральных чисел.
10. Определить рекурсивный предикат, который находит **n**-ое число Фибоначи.
11. Определить предикат для функции Акермана:
$$A(n,m) = \begin{cases} m+1, & \text{дaк\ddot{a}} n=0 \\ A(n-1,1), & \text{дaк\ddot{a}} n>0, m=0 \\ A(n-1, A(n,m-1)), & \text{дaк\ddot{a}} n>0, m>0 \end{cases}$$
12. Определить рекурсивный предикат для подсчета длины натурального числа(кол. цифр).
13. Определить рекурсивный предикат для подсчета суммы всех делителей натурального числа.
14. Определить рекурсивный предикат проверяющий, если у двух натуральных чисел есть общие делители.
15. Определить рекурсивный предикат для подсчета суммы первых **n** нечетных нат. чисел.
16. Определить рекурсивный предикат проверяющий, если натуральное число является простым.
17. Определить рекурсивный предикат проверяющий, если натуральное число является совершенным (число равно сумме своих делителей, например  $6=3+2+1$ ).
18. Определить рекурсивный предикат, который для заданного натурального числа, находит сумму первых **n** натуральных чисел, для которых он является делителем.
19. Определить рекурсивный предикат, который находит сумму трех натуральных чисел.
20. Определить рекурсивный предикат, который преобразует два натуральных числа в два взаимно-простых числа (не имеют общих делителей).

**Искусственный интеллект. Лабораторная работа №3.  
Обработка списков на языке ПРОЛОГ. Рекурсивные определения.**

1. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который проверяет принадлежность элемента списку.
  - b) предикат, который находит в списке элемент с заданной позицией.
  - c) предикат, который дублирует все элементы списка.
2. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который печатает элементы списка.
  - b) предикат, который печатает элементы списка в обратном порядке.
  - c) предикат, который дублирует все элементы списка.
3. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который склеивает два списка.
  - b) предикат, который обращает список.
  - c) предикат, проверяющий если один список содержится в другом(подмножество).
4. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который удаляет из списка заданный элемент.
  - b) предикат, который удаляет из списка элемент с заданной позицией.
  - c) предикат, который считает длину произвольного списка.
5. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, считающий сколько раз встречается заданный элемент в списке.
  - b) предикат, который считает сумму элементов числового списка.
  - c) предикат, считающий среднее арифметическое значение элементов списка.
6. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который находит максимальный элемент числового списка.
  - b) предикат, находящий позицию заданного элемента в списке.
  - c) предикат, находящий элемент списка с заданной позицией.
7. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, считающий произведение элементов числового списка.
  - b) предикат, который печатает элементы с нечетных позиций списка.
  - c) предикат, который печатает в обратном порядке элементы списка с четных позиций.
8. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который склеивает два списка.
  - b) предикат, который увеличивает на единицу все элементы числового списка.
  - c) предикат, удаляющий из списка последний элемент.
9. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который инверсирует список.
  - b) предикат, удаляющий из списка предпоследний элемент.
  - c) предикат, который в числовом списке уменьшает на 3 все элементы больше 3-х.
10. Определить на языке ПРОЛОГ следующие рекурсивные предикаты:
  - a) предикат, который строит подсписок нечетных элементов заданного списка.
  - b) предикат, удаляющий из списка третий элемент.
  - c) предикат, удаляющий из списка все элементы с четных позиций.

**Искусственный интеллект. Лабораторная работа №4.**  
**Обработка списков на языке ПРОЛОГ. Рекурсивные определения.**

1. Напишите правило для нахождения максимального элемента числового списка.
2. Напишите правило для склеивания двух списков следующим образом: первый список инверсируется и склеивается со вторым списком.
3. Напишите правило, которое в произвольном списке меняет местами элементы следующим образом: первый элемент со вторым, третий с четвертым, пятый с шестым и т.д. Если список содержит нечетное количество элементов, последний элемент списка останется на своей позиции.
4. Напишите правило, которое в произвольном списке удаляет все повторения элементов, кроме последнего повторения.
5. Напишите правило, которое проверяет, если заданный элемент встречается в заданном списке ровно два раза.
6. Напишите правило, которое удаляет из произвольного списка первые N элементов.
7. Напишите правило, которое проверяет, является ли некоторый список префиксом другого списка.
8. Напишите правило, которое проверяет, если заданный элемент встречается в некотором списке один единственный раз (является уникальным).
9. Напишите правило, которое вычисляет сумму последних двух элементов для заданного числового списка.
10. Напишите правило, которое считает сколько раз заданный элемент встречается в заданном списке.
11. Напишите правило, которое удаляет все появления заданного элемента с нечетных позиций из некоторого списка. Позицией первого элемента списка является 1.
12. Напишите правило, которое для произвольного списка строит новый список, состоящий из двух элементов: первый и последний элемент исходного списка.
13. Напишите правило, которое для заданного числа и заданного числового списка проверяет, есть ли в списке два подряд идущих элементов, сумма которых равна данному числу.
14. Для заданного натурального числа и заданного числового списка, определить содержится ли в списке два элемента, для которых заданное число является их средним значением.
15. Напишите правило, которое находит сумму положительных элементов числового списка.
16. Напишите правило, удаляющее из списка каждое второе появление заданного элемента.
17. Напишите правило для суммирования двух векторов, представленных в виде двух списков.
18. Напишите правило, которое добавляет заданный элемент в конец заданного списка.
19. Напишите правило для нахождения суммы индексов нулевых элементов списка. Индексом первого элемента списка является 1.
20. Напишите правило, которое каждый элемент списка умножает на свой индекс (первый на 1).

## Искусственный интеллект. Лабораторная работа №5.

### Обработка списков на языке ПРОЛОГ. Применение списков для представления различных структур данных.

1. Написать программу для сортировки списка методом перебора: генерируются разные перестановки списка до тех пор, пока не получим упорядоченную перестановку элементов.
2. Написать программу для сортировки списка методом вставки: голова списка удаляется, сортируется хвост списка, затем голова вставляется в упорядоченный хвост на соответствующую позицию.
3. Написать программу для сортировки списка пузырьковым методом.
4. Написать программу для сортировки списка быстрым методом: для заданного списка  $[Голова|Хвост]$ , *Хвост* разделяется на два списка  $L$  и  $M$ , таким образом, что  $L$  содержит все элементы, которые меньше чем *Голова*, а  $M$  содержит все элементы, которые больше чем *Голова*. Сортированный список будет получен через склеивание упорядоченного списка  $L$  с элементом *Голова* и с упорядоченным списком  $M$ . В свою очередь, списки  $L$  и  $M$  сортируются рекурсивно тем же быстрым методом.
5. Написать программу для сортировки списка методом селекции: из списка выбирается максимальный элемент и переносится в голову результирующего списка, изначально пустого. Процесс повторяется, пока не будут перенесены все элементы списка.
6. Написать программу, которая для натурального числа  $N$  и двух списков *Список* и *Комб* выполняет:
  - а) проверяет, является ли список *Комб* комбинацией из  $N$  элементов списка *Список*;
  - б) для заданного списка *Список* натурального числа  $N$ , строит все списки *Комб*, которые являются комбинациями из  $N$  элементов списка *Список*.
7. Написать программу, которая для заданного списка (множества чисел) и заданного числа, находит подмножество элементов, сумма которых равна заданному числу.
8. Написать программу, которая для двух произвольных списков проверяет, встречаются ли элементы первого списка во втором списке в том же порядке (удалив лишние элементы из второго списка, получаем первый список).  
Например: Список1=[a, b, d, x, z], Список2=[s, b, a, c, b, b, d, e, x, r, z].
9. Большие целые числа могут быть представлены в виде:  $[n_1, c_1, n_2, c_2, \dots, n_k, c_k]$ , где  $n_i$  – количество повторений цифры  $c_i$ . Написать программу, которая преобразует число, представленное в виде списка цифр, в список вышеописанных пар.  
Например:  $[1, 1, 1, 1, 2, 3, 3, 5, 5, 0, 0, 0] \rightarrow [4, 1, 1, 2, 2, 3, 2, 5, 3, 0]$ .
10. Большие целые числа могут быть представлены в виде:  $[n_1, c_1, n_2, c_2, \dots, n_k, c_k]$ , где  $n_i$  – количество повторений цифры  $c_i$ . Написать программу, которая преобразует число, заданное в виде списка вышеописанных пар, в список цифр, из которых состоит число.  
Например:  $[4, 1, 1, 2, 2, 3, 2, 5, 3, 0] \rightarrow [1, 1, 1, 1, 2, 3, 3, 5, 5, 0, 0, 0]$ .
11. Квадратные матрицы  $n \times n$  будем представлять в виде списков из  $n^2$  элементов, в которых первые  $n$  элементов соответствуют первой строке матрицы, следующие  $n$  элементов – второй строке, и.т.д. Написать программу, которая для двух чисел  $i, j$  и списка представляющего матрицу, находит элемент с позиции  $(i, j)$  в матрице.
12. Написать программу, которая в произвольном числовом списке считает, сколько элементов содержатся, удовлетворяющих условию: являются больше любого элемента справа.  
Например:  $[1, 14, 3, 12, 4, 7, 3, 6, 2, 1, 2] \rightarrow 5$ .

13. Даны два списка: первый – произвольный список чисел, второй – список чисел отсортированный в порядке возрастания. Написать программу, которая добавляет все элементы первого списка во второй, таким образом, чтобы второй список оставался сортированным.

14. Дан список, содержащий все натуральные числа от 1 до N, расположенные в произвольном порядке. Написать программу, которая переставляет элементы этого списка таким образом, чтобы на любых двух соседних позициях не находились два соседних натуральных числа (строит новый список). Например: 1 и 2, 4 и 5, 5 и 4, 8 и 7 и др.

15. Дан список, содержащий произвольные подсписки чисел. Написать программу, сортирующую этот список в порядке возрастания суммы элементов подсписков.

Например:  $[[5, 3, 1], [], [-4, 2], [0, 1, 2, 3]] \rightarrow [[-4, 2], [], [0, 1, 2, 3], [5, 3, 1]]$ .

16. Дан список, состоящий из символьных подсписков. Написать программу, которая преобразует этот список в множество – список символов (без подсписков) в котором элементы не повторяются (программа строит новый список).

Например:  $[[x, a, b, a], [b, c, d, e], [], [b, c, f]] \rightarrow [x, a, b, c, d, e, f]$ .

17. Написать программу, которая удаляет из произвольного списка первое и последнее повторение каждого элемента списка.

Например:  $[a, b, a, c, d, c, a, b, f, d, b, a, b] \rightarrow [a, a, b, b]$ .

18. Написать программу, которая удаляет из произвольного списка второе и последнее повторение каждого элемента списка.

Например:  $[u, v, x, u, y, u, z, x, a, x, z, y, u] \rightarrow [u, x, y, u, z]$ .

19. Большие целые числа могут быть представлены в виде:  $[[n_1, c_1], [n_2, c_2], \dots, [n_k, c_k]]$ , где  $n_i$  – количество повторений цифры  $c_i$ . Написать программу, которая преобразует число, представленное в виде списка цифр, в список вышеописанных пар (список подсписков каждый из которого состоит из двух элементов).

Например:  $[1, 1, 1, 1, 2, 3, 3, 5, 5, 0, 0, 0] \rightarrow [[4, 1], [1, 2], [2, 3], [2, 5], [3, 0]]$ .

20. Большие целые числа могут быть представлены в виде:  $[[n_1, c_1], [n_2, c_2], \dots, [n_k, c_k]]$ , где  $n_i$  – количество повторений цифры  $c_i$ . Написать программу, которая преобразует число, заданное в виде списка вышеописанных пар (списка подсписков из двух элементов), в список цифр, из которых состоит число.

Например:  $[[4, 1], [1, 2], [2, 3], [2, 5], [3, 0]] \rightarrow [1, 1, 1, 1, 2, 3, 3, 5, 5, 0, 0, 0]$ .