

Общая постановка задачи

- Для задания, соответствующего номеру вашего варианта, опишите на языке Prolog предикат, выводящий на экран результат для заданных аргументов.
- Опишите программу в текстовом файле с именем `task25-NN.pro`, где NN — номер вашего варианта. Полученный файл загрузите на портал в качестве выполненного задания.

Пример выполнения задания

ТЕКСТ ЗАДАЧИ: *Описать функцию, удаляющую подряд идущие дублирующие элементы в заданном списке.*

РЕШЕНИЕ: Содержимое файла `task26-NN.pro`:

```
1 delDuplicates([], []).
2 delDuplicates([X], [X]).
3 delDuplicates([X, Y | L], [X | R]) :- X \= Y, delDuplicates([Y | L], R).
4 delDuplicates([X, X | L], R) :- delDuplicates([X | L], R).
5
6 delDuplicates(X) :- delDuplicates(X, Res), write(Res).
7
8 ?- delDuplicates([ 111, 111, 111, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 1
9           , 1, 1, 2, 3, 3, 43, 4, 5, 6, 6, 6, 6, 4, 4, 4
10          , 4, 4, 4, 3, 3, 3, 2]).
```

Пример выполнения задания

ТЕКСТ ЗАДАЧИ: *Определите, является ли список множеством, т. е. входит ли каждый элемент списка в него лишь один раз.*

РЕШЕНИЕ: Содержимое файла `task26-NN.pro`:

```
1 delDuplicates ([], []).
2 delDuplicates ([X], [X]).
3 delDuplicates ([X, Y | L], [X | R]) :- X \= Y, delDuplicates ([Y | L], R).
4 delDuplicates ([X, X | L], R) :- delDuplicates ([X | L], R).
5
6 delDuplicates (X) :- delDuplicates (X, Res), write(Res).
7
8 ?- delDuplicates ([ 111, 111, 111, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 1
9      , 1, 1, 2, 3, 3, 43, 4, 5, 6, 6, 6, 6, 4, 4, 4
10     , 4, 4, 4, 3, 3, 3, 2]).
```

Варианты заданий

1. *Опишите функцию, определяющую является ли заданный список палиндромом.*
2. *Функция из исходного списка формирует список-результат: первый элемент — сумма всех элементов, второй — сумма элементов хвоста, третий — сумма элементов от третьего до последнего, и т. д..*
3. *Функция вычисляет скалярное произведение двух числовых списков. Если во входных списках задано разное количество элементов, то количество элементов в результате должно равняться количеству элементов в более коротком списке.*
4. *Реализовать функцию, включающую объект на заданную позицию в списке. Позиция в списке задается порядковым номером элемента, отсчитываемым от 0. Если номер заданной позиции больше чем количество элементов в данном списке, то последний элемент списка должен быть продублирован недостающее количество раз.*
5. *Реализовать функцию пересечения двух множеств, где каждое множество представляется списком неповторяющихся элементов.*
6. *Описать функцию, которая элемент списка с заданным номером заменяет на данный элемент.*

7. Описать функцию, которая заменяет все вхождения определенного элемента на данный.
8. Описать функцию, которая уменьшает все элементы числового списка на заданное значение.
9. Описать функцию, которая определяла бы положение заданного элемента в списке (возвращала бы порядковый номер элемента в списке).
10. Опишите функцию, которая определяет, является ли первое множество подмножеством второго. Каждое множество задается в виде списка неповторяющихся элементов.
11. Опишите функцию, принимающую в качестве аргументов два списка и возвращающую True, если все элементы, входящие в оба списка, расположены в этих списках в одном и том же порядке.
12. Описать функцию, которая, выдавала бы элемент списка по заданному номеру, отсчитывая элементы от конца списка.
13. Опишите функцию, удаляющую из списка каждый k -ый элемент. К примеру из [3, 1, 3, 2, 3, 5, 4, 4, 7, 9, 6, 6] при k равном 4 должен получиться список [3, 1, 3, 3, 5, 4, 7, 9, 6]
14. Опишите функцию, определяющую сколько раз в заданном списке встречается заданный объект.

- 15.** *Опишите функцию, которая в заданный список вставляла бы с данной позиции элементы второго списка*
- 16.** *Опишите функцию, принимающую в качестве аргументов два списка и возвращающую в качестве результата список, содержащий элементы первого списка, не принадлежащие второму списку.*