

Задачи по Visual Prolog

1. Создайте предикаты, проверяющие, являются ли два человека:
 - a. сестрами, братьями;
 - b. дедушкой (бабушкой) и внуком (внучкой);
 - c. дядей (тетей) и племянником (племянницей);
 - d. супругами;
 - e. родственниками;
 - f. и т. д.

Построить граф — «родословное (генеалогическое) древо».

2. Создайте предикат, заменяющий в исходном списке (все) первое вхождение заданного значения другим.
3. Создайте предикат, порождающий по заданному натуральному числу N список, состоящий из натуральных чисел от 1 до N (от N до 1).
4. Создайте предикат, порождающий список, состоящий из случайного количества случайных чисел.
5. Создайте предикат, который увеличивает элементы исходного списка на единицу.
6. Создайте предикат, переводящий список чисел в список соответствующих им названий.
7. Создайте предикат, переводящий список арабских чисел в список соответствующих им римских чисел, и наоборот.
8. Создайте предикат, преобразующий исходный список в список позиций отрицательных элементов.
9. Создайте предикат, удаляющий из исходного списка элементы с четными номерами.
10. Создайте предикат, который разделит исходный список из целых чисел на два списка: список положительных чисел и список отрицательных чисел.
11. Создайте предикат, разделяющий исходный список на два подсписка. В первый из них должны попасть элементы с нечетными номерами, во второй — элементы с четными номерами.
12. Создайте предикат, вычисляющий по списку и числу, подсписок исходного списка, начинающийся с элемента с указанным номером.
13. Создайте предикат, осуществляющий удаление указанного количества последних элементов исходного списка.
14. Создайте предикат, осуществляющий разделение исходного списка на два подсписка. В первый из них должно попасть указанное количество элементов из начала списка, во второй — оставшиеся элементы.
15. Создайте предикат, осуществляющий разделение исходного списка на два подсписка. В первый из них должно попасть указанное количество элементов с конца списка, во второй — оставшиеся элементы.
16. Создайте предикат, находящий (удаляющий) предпоследний элемент списка.
17. Создайте предикат, заменяющий в исходном списке два подряд идущих одинаковых элемента одним.

18. Создайте предикат, удаляющий в исходном списке все повторные вхождения элементов.
19. Создайте предикат, осуществляющий перестановку двух элементов списка с заданными номерами.
20. Создайте предикат, генерирующий все перестановки элементов списка, указанного в качестве первого аргумента предиката.
21. Создайте предикат, осуществляющий циклический сдвиг элементов списка на заданное количество шагов влево (вправо).
22. Создайте предикат, осуществляющий поэлементное перемножение соответствующих элементов двух исходных списков.
23. Создайте предикат, вычисляющий скалярное произведение векторов, заданных списками целых чисел.
24. Создайте предикат, добавляющий элементы одного списка во второй список, начиная с заданной позиции.
25. Создайте предикат, возвращающий по списку и двум числам M и N подсписок исходного списка, состоящий из элементов с номерами от M до N .
26. Создайте предикат, формирующий список простых чисел, не превосходящих данного числа.
27. Создайте предикат, порождающий всевозможные перестановки исходного множества.
28. Создайте предикат, порождающий всевозможные подмножества исходного множества.
29. Реализовать генератор деревьев, чтобы выдаваемые им деревья имели количество вершин, точно соответствующее числу, указанному в его первом аргументе.
30. Создайте предикат, находящий максимальное из значений, находящихся в вершинах дерева.
31. Создайте предикат, проверяющий, что дерево является двоичным справочником.
32. Создайте предикат, переписывающий дерево в двоичный справочник.
33. Создайте предикат, который будет находить среднеарифметическое значений, находящихся в вершинах (листьях) дерева.
34. Напишите предикат, находящий сумму чисел, расположенных в вершинах дерева так, чтобы он суммировал только положительные числа.
35. Создайте предикат, подсчитывающий количество всех вершин данного дерева заданной высоты.
36. Создайте предикат, выводящий значения находящиеся в вершинах заданной высоты.
37. Создайте предикат, проверяющий, является ли одно дерево поддеревом второго.
38. Создайте предикат, выводящий дерево на экран, расположив дерево следующим образом: корень находится слева, а листья справа.
39. Создайте предикат, выводящий дерево на экран, расположив дерево обычным образом: корень находится сверху, а листья снизу.

Задачи по Лиспу

1. Запишите последовательности вызовов CAR и CDR, выделяющие из приведенных ниже списков символ «цель». Упростите эти вызовы с помощью функций C...R.
 - a) (1 2 цель 3 4)
 - b) ((1) (2 цель) (3 (4)))
 - c) ((1 (2 (3 4 цель))))
2. Определите функцию, возвращающую последний элемент списка.
3. Определите функцию, заменяющую в исходном списке все вхождения заданного значения другим.
4. Определите функцию, порождающую по заданному натуральному числу N список, состоящий из натуральных чисел от 1 до N.
5. Определите функцию, которая увеличивает элементы исходного списка на единицу.
6. Определите функцию, переводящую список чисел в список соответствующих им названий.
7. Определите функцию, удаляющую из исходного списка элементы с четными номерами.
8. Определите функцию, которая разделит исходный список из целых чисел на два списка: список положительных чисел и список отрицательных чисел.
9. Определите функцию, разделяющую исходный список на два подсписка. В первый из них должны попасть элементы с нечетными номерами, во второй — элементы с четными номерами.
10. Определите функцию, осуществляющую удаление указанного количества последних элементов исходного списка.
11. Определите функцию, осуществляющую разделение исходного списка на два подсписка. В первый из них должно попасть указанное количество элементов с начала списка, во второй — оставшиеся элементы.
12. Определите функцию, заменяющую в исходном списке два подряд идущих одинаковых элемента одним.
13. Определите функцию, удаляющую в исходном списке все повторные вхождения элементов.
14. Определите функцию, осуществляющую перестановку двух элементов списка с заданными номерами.
15. Определите функцию, вычисляющую скалярное произведение векторов, заданных списками целых чисел.
16. Определите функцию, добавляющую элементы одного списка во второй список, начиная с заданной позиции.
17. Создайте предикат, порождающий всевозможные перестановки исходного множества.
18. Определите предикат, проверяющий, является ли аргумент одноуровневым списком.
19. Определите функцию (ЛУКОВИЦА n), строящую N-уровневый вложенный список, элементом которого на самом глубоком уровне является N.
20. Определите функцию ПЕРВЫЙ-АТОМ, результатом которой будет первый атом списка.
Пример:
> (ПЕРВЫЙ-АТОМ '(((a b)) c d))
A

21. Определите функцию, удаляющую из списка первое вхождение данного элемента на верхнем уровне.
22. Определите функцию, которая обращает список (a b c) и разбивает его на уровни (((c b) a)).
23. Определите функции, преобразующие список (a b c) к виду (a (b (c))) и наоборот.
24. Определите функции, осуществляющие преобразования между видами (a b c) и (((a) b) c).
25. Определите функцию, удаляющую из списка каждый второй элемент.
26. Определите функцию, разбивающую список (a b c d ...) на пары ((a b) (c d) ...).
27. Определите функцию, которая, чередуя элементы списков (a b ...) и (1 2 ...), образует новый список (a 1 b 2 ...).
28. Определите функцию, вычисляющую, сколько всего атомов в списке (списочной структуре).
29. Определите функцию, вычисляющую глубину списка (самой глубокой ветви).
30. Запрограммируйте интерпретатор ВЫЧИСЛИ, который преобразует инфиксную запись операций (например, +, -, * и /) выражения в префиксную и возвращает значение выражения. Пример:
> (ВЫЧИСЛИ '((-2 + 4) * 3))
6
31. Определите функцию (ПЕРВЫЙ-СОВПАДАЮЩИЙ x y), которая возвращает первый элемент, входящий в оба списка x и y, в противном случае NIL.
32. Определите предикат МНОЖЕСТВО-Р, который проверяет, является ли список множеством, т.е. входит ли каждый элемент в список лишь один раз.
33. Определите функцию МНОЖЕСТВО, преобразующую список в множество.
34. Определите предикат РАВЕНСТВО-МНОЖЕСТВ, проверяющий совпадение двух множеств (независимо от порядка следования элементов). Подсказка: используйте функцию REMOVE, удаляющую данный элемент из множества.
35. Определите функцию ПОДМНОЖЕСТВО, которая проверяет, является ли одно множество подмножеством другого. Определите также СОБСТВЕННОЕ — ПОДМНОЖЕСТВО.
36. Определите предикат НЕПЕРЕСЕКАЮЩИЕСЯ, проверяющий, что два множества не пересекаются, т.е. у них нет общих элементов.
37. Определите функцию ПЕРЕСЕЧЕНИЕ, формирующую пересечение двух множеств, т.е. множество из их общих элементов.
38. Определите функцию ОБЪЕДИНЕНИЕ, формирующую объединение двух множеств.
39. Определите функцию СИММЕТРИЧЕСКАЯ-РАЗНОСТЬ, формирующую множество из элементов не входящих в оба множества.
40. Определите функцию РАЗНОСТЬ, формирующую разность двух множеств, т.е. удаляющую из первого множества все общие со вторым множеством элементы.
41. Реализовать генератор деревьев, чтобы выдаваемые им деревья имели количество вершин, точно соответствующее числу, указанному в его первом аргументе.
42. Определите функцию, находящую максимальное из значений, находящихся в вершинах дерева.
43. Определите функцию, подсчитывающую количество всех вершин данного дерева заданной высоты.
44. Определите функцию, проверяющую, является ли одно дерево поддеревом второго.

45. Предположим, что у имени города есть свойства x и y , которые содержат координаты места нахождения города относительно некоторого начала координат. Напишите функцию

(РАССТОЯНИЕ $a\ b$)

вычисляющую расстояние между городами a и b , если значением функции ($\text{SQRT } x$) является квадратный корень числа x .

46. Предположим, что отец и мать некоторого лица, хранятся как значения соответствующих свойств у символа, обозначающего это лицо. Напишите функцию

(РОДИТЕЛИ x)

которая возвращает в качестве значения родителей, и предикат

(СЕСТРЫ-БРАТЬЯ $x_1\ x_2$)

который истинен в случае, если x_1 и x_2 — сестры или братья, родные или с одним общим родителем.

47. Определите функцию REMPROPS , которая удаляет все свойства символа.

48. Функция GET возвращает в качестве результата NIL в том случае, если у символа нет данного свойства, либо если значением этого свойства является NIL . Следовательно, функцией GET нельзя проверить, есть ли некоторое свойство в списке свойств. Напишите предикат

(HASPROP символ свойство)

который проверяет, обладает ли символ данным свойством.