

Самостоятельная работа №5 (семинар 7)

Тема Работа со списками

1 Использовать программу delete (удаление) для вставки элементов в список.

2* Модернизировать программу delete для построения всех возможных перестановок списка(Для этого потребуется добавить новое отношение change с двумя аргументами.

При этом правило change перед рекурсивным вызовом самого себя должно использовать предикат delete).

Имеется в виду, что если у вас есть список [3,5,6],то в ответе будут все возможные перестановки:

L=[3,5,6] L=[3,6,5] L=[5,3,6] и так далее.Будьте осторожны и внимательны! Если поставить цель ?change(L,[3,5,6], произойдет заикливание. Постарайтесь это объяснить.

3 Сделать пословный перевод, используя списки слов, то есть выполнить отображение, преобразующее один список в другой путем применения функции translation к каждому элементу списка. Вместо списка английских слов получаете список русских и наоборот. Первоначально у вас есть некоторый словарь.

4 Используйте предикат append в обратном порядке для разбиения списка на две части. Покажите все возможные варианты разбиения.(см лекцию)

5 Создайте список студентов вашей группы(можно ограничиться 9-10человеками).Выдайте на экран фамилии студентов, предшествующие какому-то студенту,выбранному вами в качестве компаратора, и фамилии студентов, следующие за ним.(использовать базовый append)

6 Используя базовый предикат append удалить :

- 4 последних элемента из списка студентов вашей группы;
- 4 первых фамилии студентов из вашей группы;
- удалить 3 первых фамилии и 3 последних фамилии

7 Задать список целых чисел и найти максимальный (минимальный) элемент списка.

8 Из списка целых чисел исключить отрицательные элементы. (Предполагается, что исходный список содержит отрицательные и положительные элементы, включая ноль).

9 Определить процедуру discard-разбить (Числа, Положительные, Отрицательные), которая разбивает список на два списка: список, содержащий отрицательные числа и список, содержащий положительные числа и ноль.

10Определить последний элемент списка целых чисел с помощью предиката append.

11*Найти сумму чисел целого типа в списке, используя соотношение `sumlist(list,integer)`. В качестве образца используйте лекционный предикат `length_of`. Если поставить цель `?sumlist([1,2,3,4,5],Sum)` ответ должен получиться `Sum=15`.

11_1)Что произойдет, если поставить цель? `sumlist(List,15)?`.

11_2) Попробуйте организовать для отношений `sumlist` или `length_of` хвостовую рекурсию.

12* Введите два отношения `sublist(List1,List2)` (подсписок), где список `List1` содержится в списке `List2` в качестве подсписка(`sublist([2,3],[1,2,3,4,5])` –имеет место, `sublist([2,4],[1,2,3,4,5])` – не имеет места)

и `prefix(Prefix,List)`, где `Prefix` является начальным подсписком списка `List`, и тогда предикат `prefix` является истинным.

Используйте предикат `prefix` в качестве основы рекурсивного определения отношения `sublist`. Его базовое правило объявляет, что префикс списка является подсписком списка.

Рекурсивное правило объявляет, что подсписок хвоста списка является подсписком самого списка. Напишите на этой основе программу, которая позволит определять подсписки ваших списков, в том числе и префиксы.

13*Составить программу, выполняющую алгоритм быстрой сортировки списка (смотри лекцию)

14. Определите отношение для вычитания множеств

`set_difference(Set1, Set2,SetDifference),`

в котором все три множества представлены в виде списков вида:

`set_difference([a,b,c,d] , [b,d,e,f] , [a,c])`

При работе обратить особое внимание на принципиальную возможность работы некоторых предикатов в обоих направлениях: на основе исходных данных получать результат, и на основе результата вычислять исходные данные без дополнительного программирования. Такие отношения принято называть обратимыми. Готовясь к работе изучите соответствующий раздел учебного пособия, лекции, активно работать на семинаре.