The user guide for lp.pro

Introduction

This is the user guide for list processing program ‘lp.pro’. This Prolog code has varies programs that work with lists, numerical or lists that associate with atoms. All programs in the code can be worked individually and the program doesn’t need anything to be imported.

Table of contents – programs guide

programs guide

1. **first**
   Use: first(L,H)
   Description: This program returns the head of the given list
   Inputs:
   L: A list that user inputs
   H: The output variable that first element of the list will be assigned to
   Demo:
   ?- first([1,2,3],H).
   H = 1.
2. **rest**
   Use: rest(L,T)
   Description: This program returns the Tail of the given list
   Inputs:
   L: A user inputted list
   T: The output of the program which the tail of the list will be assigned to
   Demo:
   ?- rest([1,2,3],T).
   T = [2, 3].
3. **last_element**
   Use: last_element(L,Result)
   Description: This program takes a list then return the last element of that list
   Inputs:
   L: User inputted list
   Result: the last element of the list
   Demo:
   ?- last_element([1,2,3],Result).
   Result = 3 .
4. **write_list**
   Use: write_list(L).
   Description:
   This program takes a inputted list then prints all the elements out one line by one line
   Inputs:
   L: A prolog list
   Demo:
   ?- write_list([a,b,c]).
   a
   b
   c
   true.

5. **write_list_reversed**

   Use: write_list_reversed(L)

   Description:

   This program outputs all the elemetns in the list but unlike write_list, its in reversed order

   Inputs:

   L: A inputted list

   Demo:

   ?- write_list_reverse([a,b,c]).

   c

   b

   a

   true.

6. **size**

   Use: size(L,Size)

   Description:

   This program takes a inputted list then return the number that indicates how many elements existed in the list

   Inputs:

   L: A inputted list

   Size: The size of the inputted list

   Demo:

   ?- size([one,two,three],S).

   S = 3.

7. **count**

   Use: count(Element,List,Count)

   Description: This program counts how many the given element exist in a given list

   Inputs:

   Element: the element that need to be counted

   List: A inputted list

   Count: returns the result that how many given elements are a found in the list

   Demo:

   ?- count(a,[a,b,c,a,b],Count).

   Count = 2 .

8. **element_of**

   Use: element_of(Element,List)

   Description: this program returns true if the given element exists in the given list

   Inputs:

   Element: the given element

   List: inputted list

   Demo:

   ?- element_of(a,[a,b,c]).

   true .

9. **contains**

   Use: contains(List,Element)

   Description: this program returns true if the given element exists in the given list

   Inputs:

   Element: the given element

   List: inputted list

   Demo:

   ?- contains([a,b,c],a).

   true .

10. **nth**

    Use: nth(Index,List,Element)

    Description: this program returns the nth place of the given list which n starts at 0

    Inputs:

    Index: a given number

    List: inputted list

    Element: variable that takes the oupout

    Demo:

    ?- nth(0,[a,b,c],E).

    E = a .

11. **pick**

    Use: pick(List,Element)

    Description: this program picks out the given element from the given list

    Inputs:

    List: the given list

    Element: output

    Demo:

    ?- pick([a,b,c],Item).

    Item = c .

12. **sum**

    Use: sum(List,Sum)

    Description: This program takes a numeric list and add all the elements togather

    Inputs:

    List: inputted list

    Sum: the sum of all the elements in the list

    Demo:

    ?- sum([1,2,3],Sum).

    Sum = 6.

13. make_list

Use: make_list(Length,Element,Output)

Description:

This program takes one element then make is to the list with a certain length

Inputs:

Length: the length of the output list

Element: given element

Output: the final result list

Demo:

?- make_list(4,a,List).

List = [a, a, a, a] .

14. iota

Use: iota(Length,Iota)

Description:

This program counts numbers to the certain length then output the list, starts from 1

Inputs:

Length: the length of the list

Iota: final list

Demo:

?- iota(9,Iota).

Iota = [1, 2, 3, 4, 5, 6, 7, 8, 9] .

15. add_first

Use: add_first(Element,List,Result)

Description: this program adds the given element to the first place of the given list

Inputs:

Element: the given element

List: the given list

Result: the final list with the given element being at the first place

Demo:

?- add_first(a,[b,c],A).

A = [a, b, c].

16. add_last

Use: add_last(Element,List,Result)

Description: This program adds the given element to the last position of the given list

Inputs:

Element: the given element

List: the given list

Result: the Final list that which the given element positioned at the end of the list

Demo:

?- add_last(a,[b,c,d],List).

List = [b, c, d, a] .

17. esrever

Use: esrever(List,Result)

Description: This program returns the reversed version of the given list

Inputs:

List: the inputted list

Result: the final result that put the inputted list reversed

Demo:

?- esrever([a,b,c],Reversed).

Reversed = [c, b, a] .

18. join_lists

Use: join_lists(L1,L2,Result)

Description: this program takes two inputted lists then join them together to one single list

Inputs:

L1: the first inputted list

L2: the second inputted list

Result: joined list

Demo:

?- join_lists([one,two],[1,2],[a,b],Result).

Result = [one, two, 1, 2, a, b].

19. product

Use: product(List,Product)

Description:

this program takes a numerical list then multiply all the elements together

Inputs:

List: the inputted numerical list

Product: the product of all the elements

Demo:

?- product([1,2,3],Product).

Product = 6.

20. factorial

Use: factorial(N,Fac)

Description: this program calculate the factorial of the given numerical list

Inputs:

N: number of the factorial to be calculated

Fac: the result

Demo:

?- factorial(8,Fac).

Fac = 40320 .

21. make_set

Use: make_set(List,Set)

Description: this program takes the given list then remove the duplicates.

Inputs:

List: the inputted list

Set: A list that contains no duplicates

Demo:

?- make_set([a,b,c,c],Set).

Set = [a, b, c] .

22. replace

Use: replace(Index,Element,List0,List)

Description:

This program replaces the element at the given index with the given element then output the replaced list

Inputs:

Index: indicates which element is going to be replaced

Element: the replacement element

List0: inputted list

List: outputted list

Demo:

?- replace(0,1,[a,b,c,d],List).

List = [1, b, c, d] .

23. remove

Use: remove(Element,List,Result)

Description: this program remove the certain element from the given list

Inputs:

Element: the element needs to be removed

List: inputted list

Result: the outputted list, without the given element.

Demo:

?- remove(a,[a,b,c],Result).

Result = [b, c] .

24. take

Use: take(List,E,R)

Description: This program takes a random element from the given list then return the rest of the list

Inputs:

List: inputted list

E: element to be taken

R: The list that without the taken element

Demo:

?- take([a,b,c,d],E,R).

E = d,

R = [a, b, c] .

25. split

Use: split(L,L1,L2)

Description: This program takes one list then returns two lists that each one takes half of the inputted list

Inputs:

L: inputted list that has small lists as elements

L1: outputted list 1

L2: outputted list 2

Demo:

?- split([[a,b],[c,d],[e,f]],L1,L2).

L1 = [a, c, e],

L2 = [b, d, f].

26. min_pair

Use: (N,N1,Min)

Description: this program returns the smaller number amount two inputted numbers

Inputs:

N: First number

N2: second number

Min: the result which is the smaller number

Demo:

?- min_pair(2,9,Min).

Min = 2 .

27. max_pair

Use: max_pair(N,N1,Max)

Description: this program returns the larger number among two inputted numbers

Inputs:

N: first number

N2: second number

Max: the larger number among two inputted numbers

Demo:

?- max_pair(9,2,Max).

Max = 9 .

28. min

Use: min(List,Min)

Description: this program returns the minimum number of the given numerical list

Inputs:

List: inputted numerical list

Min: the minimum number within the given list

Demo:

?- min([4,5,6,7],Min).

Min = 4.

29. max

Use: max(List,Max)

Description: this pogrom takes the given numerical list then returns the maximum number within the list

Inputs:

List: the given list

Max: the maximum number of the given list

Demo:

?- max([2,7,2,67,1,9],Max).

Max = 67 .

30. sort_inc

Use: sort_inc(List,Sorted)

Description: this program takes the given numerical list then sort the list to an increase order

Inputs:

List: the given numerical list

Sorted: the sorted numerical list that has an increasing order

Demo:

?- sort_inc([1,5,2,6,2,8,8,34,14],Sorted).

Sorted = [1, 2, 2, 5, 6, 8, 8, 14, 34] .

31. sort_dec

Use: sort_dec(List,Sorted)

Description: this program takes the given numerical list then sort the list to a decreasing order

Inputs:

List: the given numerical list

Sorted: the sorted numerical list that has a decreasing order

Demo:

?- sort_dec([2,5,8,2,6,1,7,26,35],Sorted).

Sorted = [35, 26, 8, 7, 6, 5, 2, 2, 1] .

32. a_list

Use: a_list(List1,List2,AssociationList)

Description: this program takes two inputted lists then put them into a list with an association relation

Inputs:

List1: the first inputted list

List2: the second inputted list

AssociationList: the list that paired elements in the two inputted lists

Demo:

?- a_list([3,2,1],[a,b,c],AssociationList).

AssociationList = [pair(3, a), pair(2, b), pair(1, c)] .

33. assoc

Use: assoc(List,Key,Value)

Description: this program takes a paired list from a_list then returns the Value according to the given Key

Inputs:

List: the given list

Key: the Key that exists in the given list

Value: value to be found from the given list

Demo:

?- assoc([pair(hi,hello),pair(weather,sunny),pair(dog,woof)],dog,Value).

Value = woof .