

* list Module PREDICATION

A list module is a set of the library module of the standard included in the system.
Include is done before using.

? <include list>;

When calling, it describes after ::list.

::list <append #out #in1 #in2>

#In1 is connected with # in2 list, and it sets it to # out.

::list <reverse #out #in>

List #in is converted in reverse the order and it sets it to #out.

::list <member #mem #list>

Whether #mem is included in #list is judged.

::list <last #out #list>

The last element of list #list is set to #out.

::list <flatten #out #in>

The list of the nest in list #in is converted into a smooth list and it sets it to #out.

::list <difference #out #in1 #in2>

An element in the element of list #in1 and #in2 lists not common is set the list of the element that doesn't exist in #in2 is set to #out though it is in #in1 as a difference.

::list <intersect #out #in1 #in2>

Elements of list #in1 and #in2 are compared, and the list of a common element is set to #out. The universal component is extracted regardless of the order of the element.

::list <union #out #in1 #in2>

The element of list #in1 and #in2 is unified, and a common element makes to one so as not to overlap, and sets all elements to #out.

::list <subset #subset #set>

When #subset is a subset of #set, it becomes true.

::list <join #out #in1 #in2>

The list that annexes #in1 to #in2 is set to #out.
Annexation is done in each element of the order of #in1
and #in2 corresponding. The first #in1 and the first #in2
are added to #out, and in a word, the following queue up
the second element, and set this to #out repeatedly even
by the last element of the list.

::list <occures #n #e #list>

Element #e sets the frequency that appears in #list to #n.