

Några elementära rekursiva predikat.

tillhor(+Element, +Lista)

Elementet *Element* tillhör listan *Lista*.

langd(+Lista, -AntalElement)

AntalElement är antalet element i listan *Lista*.

antal_ett_elem(+Element, +Lista, -AntalElement)

AntalElement är antalet av elementet *Element* i listan *Lista*.

lagg_ihop1(+Lista1, +Lista2, -Lista1Lista2)

Lista1Lista2 är resultatet av att lägga ihop listorna *Lista1* och *Lista2* där elementen i *Lista1* placeras före elementen i *Lista2*.

lagg_ihop2(+Lista1, +Lista2, -Lista2Lista1)

Lista1Lista2 är resultatet av att lägga ihop listorna *Lista1* och *Lista2* där elementen i *Lista2* placeras före elementen i *Lista1*.

ta_bort(+Elem, +Lista1, -Lista2)

Lista2 är resultatet av att ta bort en ev. förekomst av elementet *Elem* ur listan *Lista1*.

ta_bort_alla(+Elem, +Lista1, -Lista2)

Lista2 är resultatet av att ta bort alla förekomster av elementet *Elem* ur listan *Lista1*.

ersatt_elem(+OutElem, +InElem, +Lista1, -Lista2)

Listan *Lista2* är resultatet av att ersätta en ev. förekomst av elementet *OutElem* i Listan *Lista1* med ett annat element *InElem*.

antal_udda(+Lista, -Antal)

Lista är en lista av heltal och *Antal* är antalet udda tal i listan.

summa(+Tal, -Summa)

Tal är ett positivt heltal och *Summa* är $1+2+3+ \dots +\text{Tal}$.

jamn_lista(+Lista)
Listan *Lista* är jämn.

sista(-Element, +Lista)
Elementet *Element* är det sista elementet i listan *Lista*.

omvand1(+Lista, -OmvandddLista)
OmvandddLista är resultatet av att vända om listan *Lista*.

omvand2(+Lista, +HjalpLista, -OmvandddLista)
OmvandddLista är resultatet av att vända om listan *Lista*. *HjalpLista* är en extra lista dit element från *Lista* flyttas ett i taget och samlas i omvänd ordning. Programmet anropas med en tom *HjalpLista*, t.ex.

?- omvand2([1,2,3,4], [], S).

placera_i_ordning(+Element, +Lista1, -Lista2)
Lista1 är en lista av heltal som är sorterade i stigande ordning. *Lista2* är resultatet av att placera elementet *Element* i ordning i *Lista1*

sortera(+Lista1, -Lista2)
Lista1 är en lista av heltal. *Lista2* är resultatet av att sortera *Lista1* i stigande ordning.

dela(+Tal, +Lista1, -Lista2, -Lista3)
Tal är ett heltal och *Lista1* är en lista av heltal. *Lista2* är lista av de talen från *Lista1* som är mindre än *Tal*, och *Lista3* är lista av de talen från *Lista1* som är större än eller lika med *Tal*.

quick_sort(+Lista1, -Lista2)
Lista1 är en lista av heltal. *Lista2* är resultatet av att sortera *Lista1* i stigande ordning.

sorted(+Lista)
Listan *Lista* är sorterad i stigande ordning.

snitt(+Lista1, +Lista2, -Snitt)
Snitt är snittet av listorna *Lista1* och *Lista2*.

platta_ut(+Lista1, -Lista2)

Lista2 är resultatet av att ”platta ut” listan *Lista1* så att 1) alla atomer i *List1* är med i *List2*, men 2) *Lista2* innehåller inga listor som element. T. ex.

?- platta_ut([1, [2, 3], [[[4]], [], 5]], Svar).
Svar= [1, 2, 3, 4, 5]

nast_storsta(-Number, +Lista)

Lista är en lista av heltal och *Number* är det näst största talet i *Lista*.

storsta(-Number, +Lista)

Lista är en lista av heltal och *Number* är det största talet i *Lista*.

inga_dubbletter(+List1, -List2)

List2 är resultatet av att ta bort alla dubbla element från *List1*.

?- inga-dubbletter([c,a,b,b,c,d,e], Svar).
Svar= [c,a,b,d,e]

dubbletter(+Lista, -Dubbletter)

Lista är en lista av heltal och *Dubbletter* är listan av de tal som förekommer *exakt två* gånger i *Lista*. T.ex.

?- dubbletter([3,1,5,2,5,3,7,3,2], Svar).
Svar = [5,2]