

```
interseccion([], Conjunto, []).
interseccion([Car|Cdr], Conjunto, [Car|Inter]):-miembro(Car, Conjunto),!,
                                                    interseccion(Cdr, Conjunto, Inter).
interseccion(_|Cdr, Conjunto, Inter):-interseccion(Cdr, Conjunto, Inter).
```

```
union([], Conjunto, Conjunto).
union([Car|Cdr], Conjunto, R):-miembro(Car, R), union(Cdr, Conjunto, R).
union(_|Cdr, Conjunto, [Car|R]):-union(Cdr, Conjunto, R).
```

```
miembro(Elem, [Elem|Cdr]).
miembro(Elem, _|Cdr):-miembro(Elem, Cdr).
```

```
inversa([], []).
inversa([Car|Cdr], R) :- inversa(Cdr, X), concat(X, Car, R).
```

```
concat([], L, L).
concat([Car|Cdr], L, [Car|R]):-concat(Cdr, L, R).
```

```
quicksort([], []).
quicksort([Car|Cdr], R):-partir(Car, Cdr, Menores, Mayores),
                        quicksort(Menores, Menores_ordenada),
                        quicksort(Mayores, Mayores_ordenada),
                        concat(Menores_ordenada, [Car|Mayores_ordenada], R).

partir(_, [], [], []).
partir(Pivote, [Car|Cdr], [Car|Menores], Mayores):- Car <= Pivote, !,
                                                    partir(Pivote, Cdr, Menores, Mayores).
partir(Pivote, [Car|Cdr], Menores, [Car|Mayores]):- partir(Pivote, Cdr, Menores, Mayores).
```

```
:-op(600,xfy,[\]).
quicksort(Lista,Resultado):-quicksort_dl(Lista,Resultado\[]).
quicksort_dl([],Resultado\Resultado).
quicksort_dl([Car|Cdr],Menoresordenada\Mayoresordenada):-partir(Car,Cdr,Menores,Mayores),
                                                             quicksort_dl(Menores,Menoresordenada\[Car|X]),
                                                             quicksort_dl(Mayores,X\Mayoresordenada).
```

```
partir(_,[],[],[]).
partir(Pivote,[Car|Cdr],[Car|Menores],Mayores):- Car <= Pivote, !,
                                                    partir(Pivote,Cdr,Menores,Mayores).
partir(Pivote,[Car|Cdr],Menores,[Car|Mayores]):- partir(Pivote,Cdr,Menores,Mayores).
```

```
linea(_,[],[]).
linea(X,[Car|Cdr],[X,Car|R]):-linea(X,Cdr,R).

cartesiano([],_,[]).
cartesiano([Car|Cdr],C,Result):-linea(Car,C,R),cartesiano(Cdr,C,Z),concat(R,Z,Result).
```

```
linea(_,[],[]).
linea(X,[Car|Cdr],[X,Car|R]):-linea(X,Cdr,R).
```

```
cartesiano(Lista,Lista2,Resultado):-cartesiano_dl(Lista,Lista2,Resultado\[]).
```

```
cartesiano_dl([],_,Lista\Lista).
cartesiano_dl([Car|Cdr],C,Cabeza\Dif):-linea(Car,C,Cabeza\Dif_car),
                                         cartesiano_dl(Cdr,C,Dif_car\Dif).
```

```
concat([],L,L).
concat([Car|Cdr],L,[Car|R]):-concat(Cdr,L,R).
```

```
aplanar([],[]).
aplanar(Atomo,[Atomo]):-atomic(Atomo),Atomo \== [].
aplanar([Car|Cdr],Result):-aplanar(Car,R),aplanar(Cdr,Z),concat(R,Z,Result).
```

`:-op(600,xfy,[\]).`

`aplanar(Lista,Resultado):-aplanar_dl(Lista,Resultado\[]).`

`aplanar_dl([],Lista\Lista).`

`aplanar_dl(Atomo,[Atomo|Cdr]\Cdr):-atomic(Atomo),Atomo \== [].`

`aplanar_dl([Car|Cdr],Cabeza\Dif):-aplanar_dl(Car,Cabeza\Dif_car),
aplanar_dl(Cdr,Dif_car\Dif).`

`suma(0,X,X).`

`suma(sig(X),Y,sig(Z)):-suma(X,Y,Z).`

`mult(0,X,0):-nat(X).`

`mult(sig(0),X,X):-nat(X).`

`mult(sig(X),Y,Z):-mult(X,Y,V), suma(V,Y,Z).`

`par(0).`

`par(sig(sig(X))):-par(X).`

`nat(0).`

`nat(sig(X)):-nat(X).`

`hombre(adan).`

`hombre(cain).`

`mortal(P):-hombre(P).`

`menorigual(0,X):-nat(X).`

`menorigual(sig(X),sig(Y)):-menorigual(X,Y).`

menor(0,sig(X)):-nat(X).
menor(sig(X),sig(Y)):-menor(X,Y).

longitud([],0).
longitud([_|Cdr],sig(X)):-longitud(Cdr,X).

incl([],_).
incl([Car|Cdr],Conjunto):-miembro(Car,Conjunto),
incl(Cdr,Conjunto).

impar(sig(0)).
impar(sig(sig(X))):-impar(X).

exp(X,0,sig(0)):-nat(X).
exp(X,sig(0),X):-nat(X).
exp(X,sig(Y),Res):-exp(X,Y,Par), mult(Par,X,Res).
