**1.Quicksort\_dl**

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

quicksort(Lista,Resultado):-quicksort\_dl(Lista,Resultado\[]).

quicksort\_dl([],Resultado\Resultado).

quicksort\_dl([Car|Cdr],Menores\_ordenada\Mayores\_ordenada):-partir(Car,Cdr,Menores,Mayores),

quicksort\_dl(Menores,Menores\_ordenada\[Car|X]),

quicksort\_dl(Mayores,X\Mayores\_ordenada).

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).

**2.Analogía**

:-op(200,xfy,[dentro\_de]).

:-op(300,xfy,[es\_a]).

analogia(X es\_a Y, Z es\_a W, Relacion) :-

figura(X,FormaX), figura(Y,FormaY), X \== Y,

figura(Z,FormaZ), figura(W,FormaW), Z \== W,

verifican(FormaX,FormaY,Relacion),

verifican(FormaZ,FormaW,Relacion).

verifican(Figura\_3 dentro\_de Figura\_2 dentro\_de Figura\_1,

Figura\_2 dentro\_de Figura\_3 dentro\_de Figura\_1,

mismo\_contorno).

verifican(Figura\_1 dentro\_de Figura\_2 dentro\_de Figura\_3,

Figura\_1 dentro\_de Figura\_3 dentro\_de Figura\_2,

misma\_figura\_central).

verifican(Figura\_1 dentro\_de Figura\_2 dentro\_de Figura\_3,

Figura\_3 dentro\_de Figura\_2 dentro\_de Figura\_1,

misma\_figura\_intermedia).

verifican(Figura\_1 dentro\_de Figura\_2 dentro\_de Figura\_3,

Figura\_1 dentro\_de Figura\_2 dentro\_de Figura\_3,

igualdad).

verifican(Figura\_1 dentro\_de Figura\_2 dentro\_de Figura\_3,

Figura\_3 dentro\_de Figura\_2 dentro\_de Figura\_1,

inversion\_concentrica).

figura(a, triangulo dentro\_de circulo dentro\_de cuadrado).

figura(b, circulo dentro\_de triangulo dentro\_de cuadrado).

figura(c, triangulo dentro\_de cuadrado dentro\_de circulo).

figura(d, cuadrado dentro\_de triangulo dentro\_de circulo).

figura(e, circulo dentro\_de cuadrado dentro\_de triangulo).

figura(f, triangulo dentro\_de circulo dentro\_de cuadrado).

**3.Analizador sintactico**

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

% analiza(Arbol,['el','chico','sube','al','arbol']).

% analiza(Arbol,['Juan','sube','al','arbol']).

% analiza(Arbol,['Juan','esta','en','la','casa']).

% analiza(Arbol,['el','chico','esta','en','la','casa']).

analiza(Arbol,F) :- frase(Arbol,F\[]).

frase(f(FN,FV),P0\P2) :-

frase\_nominal(\_,Numero,FN,P0\P1),

frase\_verbal(Numero,FV,P1\P2),!.

frase\_nominal(\_,\_,fn(nil),P0\P0).

frase\_nominal(Genero,Numero,fn(n(N)),P0\P1) :-

es\_nombre(Genero,Numero,N,P0\P1).

frase\_nominal(Genero,Numero,fn(det(Det),n(N),rel(Rel)),P0\P3) :-

es\_determinante(Genero,Numero,Det,P0\P1),

es\_nombre(Genero,Numero,N,P1\P2),

es\_relativa(Rel,P2\P3).

s\_prep(sprep(prep(Prep),FN),P0\P2) :-

es\_preposicion(Prep,P0\P1),

frase\_nominal(\_,\_,FN,P1\P2).

frase\_verbal(\_,fv(nil),P0\P0).

frase\_verbal(Numero,fv(v(Verbo),objeto(FN)),P0\P2) :-

es\_verbo\_trans(Numero,Verbo,P0\P1),

frase\_nominal(\_,\_,FN,P1\P2).

frase\_verbal(Numero,fv(v(Verbo),cc(SP)),P0\P2) :-

es\_verbo\_intrans(Numero,Verbo,P0\P1),

s\_prep(SP,P1\P2).

es\_relativa(rel(nil),P0\P0).

es\_preposicion('a',['a'|X]\X).

es\_preposicion(a+el('al'),['al'|X]\X).

es\_preposicion('en',['en'|X]\X).

% es\_determinante(masculino,singular,'un',['un'|X]\X).

% es\_determinante(femenino,singular,'una',['una'|X]\X).

% es\_determinante(masculino,plural,'unos',['unos'|X]\X).

% es\_determinante(femenino,plural,'unas',['unas'|X]\X).

es\_determinante(masculino,singular,'el',['el'|X]\X).

es\_determinante(femenino,singular,'la',['la'|X]\X).

es\_determinante(masculino,plural,'los',['los'|X]\X).

es\_determinante(femenino,plural,'las',['las'|X]\X).

es\_nombre(masculino,singular,'chico',['chico'|X]\X).

es\_nombre(femenino,singular,'chica',['chica'|X]\X).

es\_nombre(masculino,plural,'chicos',['chicos'|X]\X).

es\_nombre(femenino,plural,'chicas',['chicas'|X]\X).

es\_nombre(masculino,singular,'Juan',['Juan'|X]\X).

es\_nombre(femenino,singular,'casa',['casa'|X]\X).

es\_nombre(femenino,plural,'casas',['casas'|X]\X).

es\_nombre(masculino,singular,'arbol',['arbol'|X]\X).

es\_nombre(masculino,plural,'arboles',['arboles'|X]\X).

es\_verbo\_trans(singular,'lee',['lee'|X]\X).

es\_verbo\_trans(plural,'leen',['leen'|X]\X).

es\_verbo\_trans(singular,'ama',['ama'|X]\X).

es\_verbo\_trans(plural,'aman',['aman'|X]\X).

es\_verbo\_intrans(singular,'esta',['esta'|X]\X).

es\_verbo\_intrans(plural,'estan',['estan'|X]\X).

es\_verbo\_intrans(singular,'sube',['sube'|X]\X).

es\_verbo\_intrans(plural,'suben',['suben'|X]\X).

**4.Criptoaritmetica**

%/ M A K E

% + S O M E

% ----------

% S O U N D /

mss([M,A,K,E,S,O,U,N,D]) :-

digitos(Digitos),

extraer(E,Digitos,D1),

sumar(E,E,0,D,C1),

extraer(D,D1,D2),

extraer(K,D2,D3),

extraer(M,D3,D4),

sumar(K,M,C1,N,C2),

extraer(N,D4,D5),

extraer(A,D5,D6),

extraer(O,D6,D7),

sumar(A,O,C2,U,C3),

extraer(U,D7,D8),

extraer(S,D8,\_),

sumar(M,S,C3,O,S),

S =\= 0.

digitos([0,1,2,3,4,5,6,7,8,9]).

extraer(A,[A|X],X).

extraer(A,[B|X],[B|Y]) :- extraer(A,X,Y).

sumar(A,B,C,Suma,Llevo) :- Suma is (A+B+C) mod 10,

Llevo is (A+B+C) // 10.

**5.Reordenamiento polinomios**

:-op(100,xfy,[^]).

ordena(0,x,0).

ordena(Cdr+Car,x,S) :- ordena(Cdr,x,R), inserta(Car,R,S).

inserta(X,0,0+X).

inserta(X\*Y^Z,Cdr+A\*Y^C,Cdr+A\*Y^C+X\*Y^Z) :- Z > C.

inserta(X\*Y^Z,Cdr+A\*Y^C,T+A\*Y^C) :- Z < C,

inserta(X\*Y^Z,Cdr,T).

inserta(X\*Y^Z,Cdr+A\*Y^C,Cdr+Res\*Y^Z) :- Z = C,Res is (A+X).

**6.Quicksort**

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).

**7.Multiplicacion**

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).

**8.Suma**

suma(0,X,X).

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

**9.Par**

par(0).

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

**10.Natural**

nat(0).

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

**11.Impar**

impar(sig(0)).

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

**12.Interseccion**

interseccion([],Conjunto,[]).

interseccion([Car|Cdr],Conjunto,[Car|Inter]):-miembro(Car,Conjunto),

interseccion(Cdr,Conjunto,Inter).

interseccion([\_|Cdr],Conjunto,Inter):-interseccion(Cdr,Conjunto,Inter).

**13.Union**

union([],Conjunto,Conjunto).

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

union(Cdr,Conjunto,R).

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

**14.Concatenar**

concat([],L,L).

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

**15.Inversa**

inversa([],[]).

inversa([Car|Cdr],R):inversa(Cdr,X),

concat(X,Car,R).

**16.Longitud**

longitud([],0).

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

**17.Menor**

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

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

**18. Menor o igual**

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

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

**19.Miembro**

miembro(Car,[Car|Cdr]).

miembro(Car,[X|Cdr]):-miembro(Car,Cdr).

**20.Inclusion**

incl([],\_).

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

incl(Cdr,Conjunto).

**21.Cartesiano**

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).

**22.Cartesiano\_dl**

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).

**23.Aplanar**

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).

**24.Aplanar\_dl**

:-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).

**25.Duplicar (una tras otra)**

duplicar(Lista,Dup) :- duplicar\_aux(Lista,Lista,Dup).

duplicar\_aux([],Lista,Lista).

duplicar\_aux([Car|Cdr],Lista,[Car|Duplicado\_Cdr]) :-

duplicar\_aux(Cdr,Lista,Duplicado\_Cdr), quicksort(Lista,[Car|Duplicado\_Cdr]).

**26.Exponencial**

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).

**27.Mortal**

hombre(adan).

hombre(cain).

mortal(P):-hombre(P).