mismo(X,X).

diferente(X,Y):- not(mismo(X,Y)).

madre(X,Y) :- progenitor(X,Y),mujer(X).

padre(X,Y) :- progenitor(X,Y),hombre(X).

hijos(X,Y) :- progenitor(Y,X).

hijo(X,Y) :- progenitor(Y,X),hombre(X).

hija(X,Y) :- progenitor(Y,X),mujer(X).

hombre(manuel).

hombre(epico).

hombre(hilbert).

hombre(fabricio).

hombre(cristofer).

hombre(octavio).

hombre(ernesto).

hombre(eladio).

hombre(cirilo).

hombre(degoberto).

mujer(zenaida).

mujer(mercedes).

mujer(alondra).

mujer(yvon).

mujer(eloidina).

mujer(aurelia).

nace(manuel,japon).

nace(epico,china).

nace(hilbert,peru).

nace(fabricio,francia).

nace(cristofer,rusia).

nace(octavio,alemania).

nace(ernesto,inglaterra).

nace(eladio,suecia).

nace(cirilo,brasil).

nace(degoberto,holanda).

nace(zenaida,brasil).

nace(mercedes,ecuador).

nace(alondra,espana).

nace(yvon,rusia).

nace(eloidina,suecia).

nace(aurelia,peru).

habla(peru,castellano).

habla(japon,japones).

habla(chino,mandarin).

habla(francia,frances).

habla(rusia,ruso).

habla(alemania,aleman).

habla(inglaterra,ingles).

habla(suecia,sueco).

habla(brasil,portugues).

habla(holanda,holandes).

habla(ecuador,castellano).

habla(espana,castellano).

idioma(X,N) :- nace(X,Y),habla(Y,N).

persona(X) :- hombre(X);mujer(X).

nro\_personas(N) :- findall(X,persona(X),L),length(L,G),N is G.

nro\_per\_idio(X,N) :- findall(Y,(persona(Y),idioma(Y,X)),L),

 length(L,G),N is G.

l\_per\_idio(X,L) :- findall(Y,(persona(Y),idioma(Y,X)),L).

por\_per\_idio(X,N) :- nro\_personas(P),nro\_per\_idio(X,I),N is (I/P\*100).