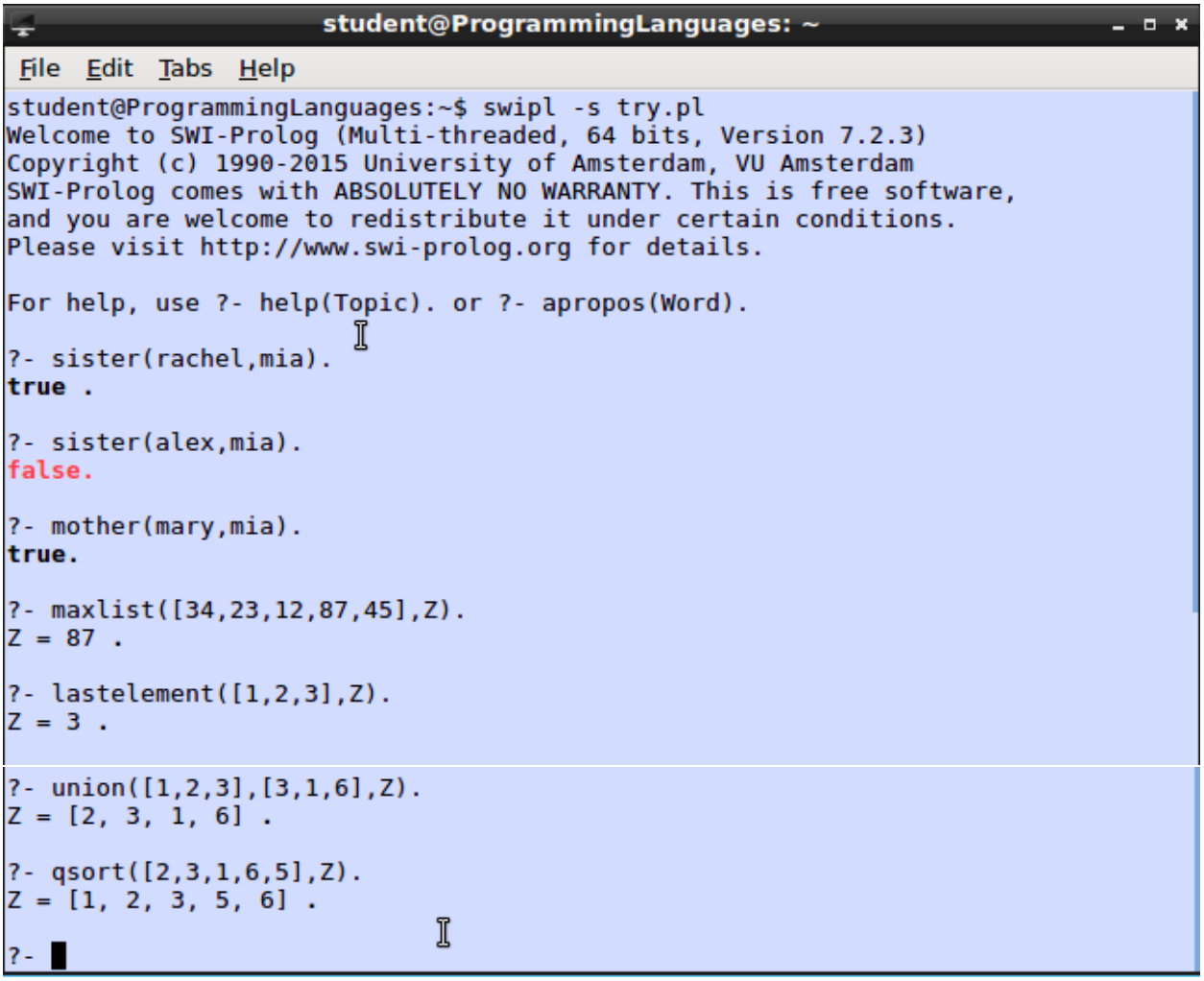
Programming Languages Assignment: Prolog

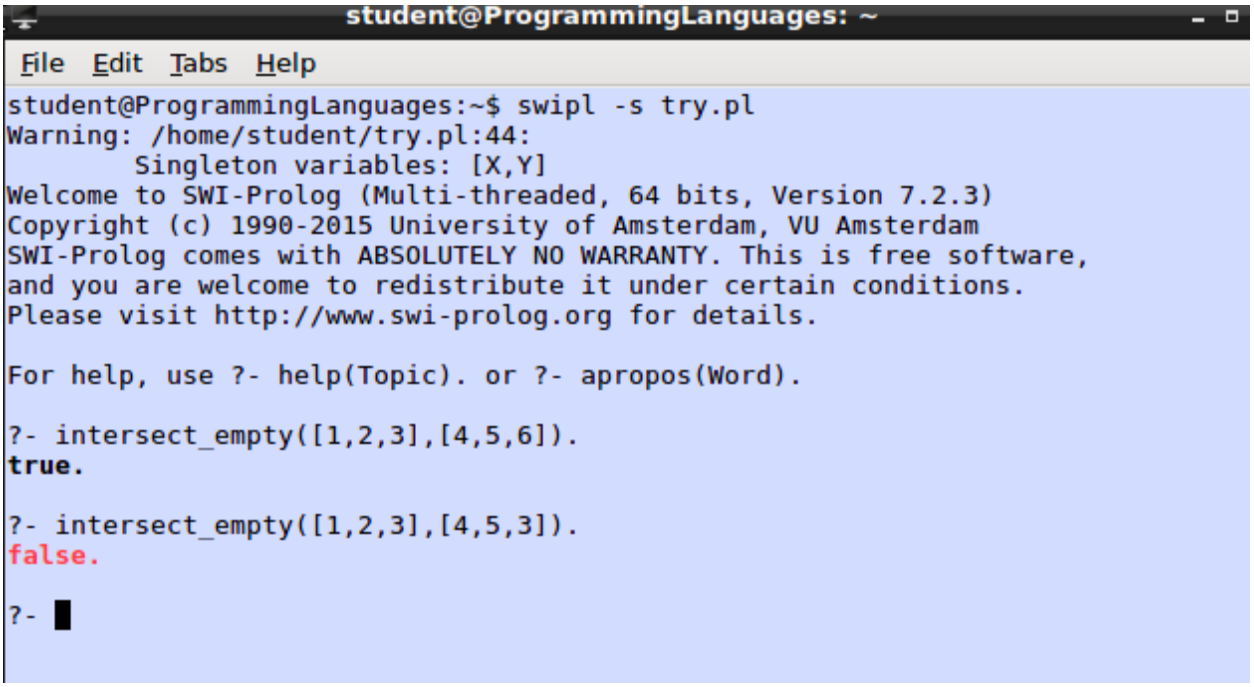
CODE:

```
male(jake).
male(john).
male(steve).
male(alex).

female(mary).
female(martha).
female(rachel).
female(mia).

parent(jake,rachel).
parent(jake,mia).
parent(jake,steve).
parent(mary,rachel).
parent(mary,mia).
parent(mary,steve).
parent(john,alex).
parent(martha,alex).

sibling(X,Y) :- parent(M,X),parent(M,Y),not(X = Y).
sister(X,Y) :- female(X),female(Y),sibling(X,Y),not(X = Y).
mother(X,Y):- parent(X,Y),female(X),not(X = Y).

maxlist([Max],Max).
maxlist([Head|Tail],Max) :- maxlist(Tail,TailMax), (Head > TailMax -> Max = Head ; Max =
TailMax).

lastelement([Y],Y).
lastelement([X|Xs],Y):- lastelement(Xs,Z), (Xs==[] -> Y = X; Y is Z).

append([A | B], C, [A | D]) :- append(B, C, D).
append([], A, A).

member(A, [A | _]).
member(A, [_ | B]) :- member(A, B).

union([],Z,Z).
union([A|B], C, D) :- ((member(A,C)) -> union(B,C,D)).
union([A|B], C, [A|D]) :- union(B,C,D).
```

```
intersect([],_,[]).
intersect([X|Xs],Y,[X|Z]):- (member(X,Y)-> intersect(Xs,Y,Z)).
intersect([X|Xs],Y,Z):-not(member(X,Y)),intersect(Xs,Y,Z).

intersect_empty([X],[Y]).
intersect_empty(X,Y):- intersect(X,Y,[]).

qsort([],[]).
qsort([H|T],S):- split(H,T,L,R), qsort(L,LS), qsort(R,Rs), append(LS,[H|Rs],S).

split(_,[],[],[]).
split(H,[A|X],[A|Y],Z):- A =< H,!,split(H,X,Y,Z).
split(H,[A|X],Y,[A|Z]):- A > H,!,split(H,X,Y,Z).
```

OUTPUT:




student@ProgrammingLanguages: ~
File Edit Tabs Help
student@ProgrammingLanguages:~$ swipl -s try.pl
Warning: /home/student/try.pl:44:
Singleton variables: [X,Y]
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
For help, use ?- help(Topic). or ?- apropos(Word).
?- intersect_empty([1,2,3],[4,5,6]).
true.
?- intersect_empty([1,2,3],[4,5,3]).
false.
?-