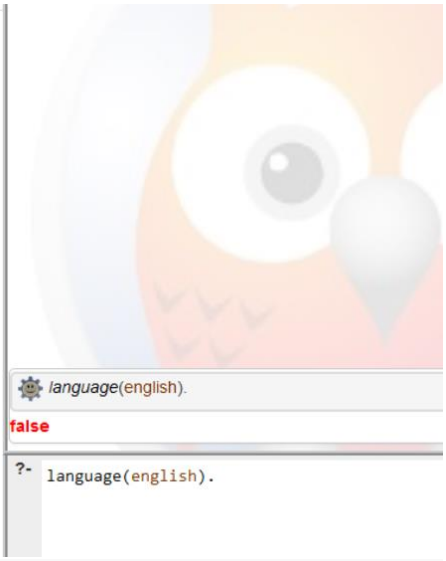
```
language(prolog).
language(java).
language(program).
```



Program

```
language(prolog).
language(java).
language(program).
```

Program

```
female(pam).
female(liz).
female(pat).
female(ann).
male(jim).
male(bob).
male(tom).
male(peter).
parent(pam,bob).
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pat).
parent(pat,jim).
parent(bob,peter).
parent(peter,jim).
mother(X,Y):- parent(X,Y),female(X).
father(X,Y):- parent(X,Y),male(X).
haschild(X):- parent(X,_).
sister(X,Y):- parent(Z,X),parent(Z,Y),female(X),X\==Y.
brother(X,Y):-parent(Z,X),parent(Z,Y),male(X),X\==Y.
```

mother(X,Y).

X = pam,
Y = bob
X = pat,
Y = jim
false

?- mother(X,Y).

Program

```
food(burger).
food(sandwich).
food(pizza).
lunch(sandwich).
dinner(pizza).

meal(X) :- food(X).
```

predecessor(bob,X).

X = ann
X = pat
X = peter
X = jim
X = jim
false

meal(x,lunch).

procedure `meal(A,B)' does not ex

meal(X).,*lunch*(X).

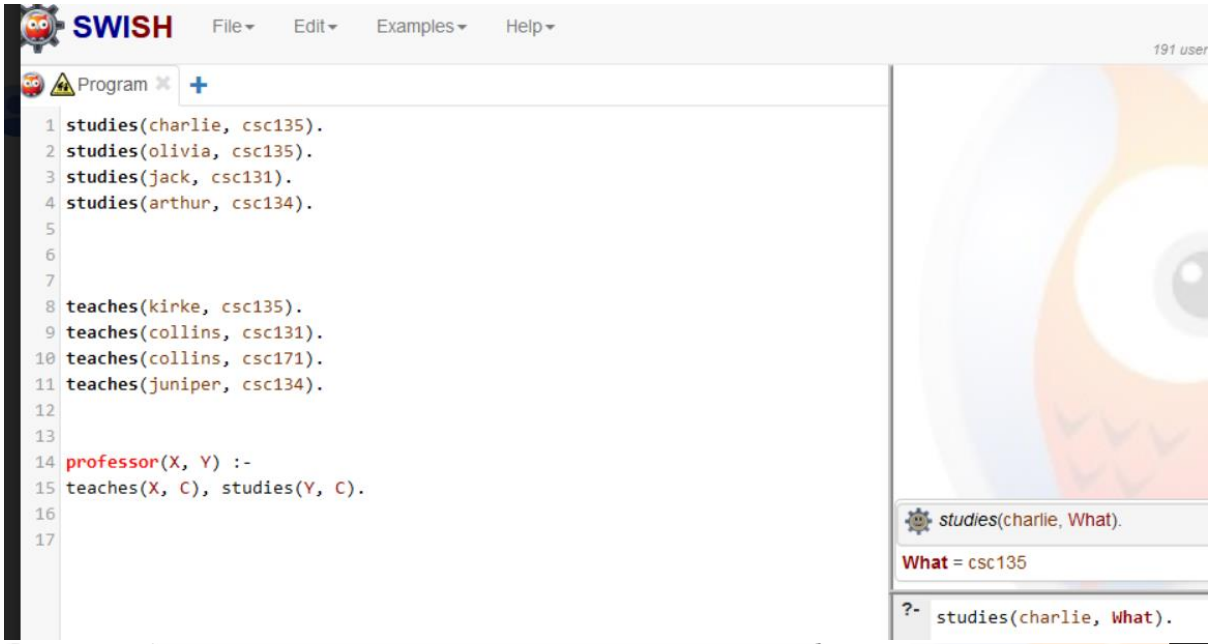
Syntax error: Operand expected, u

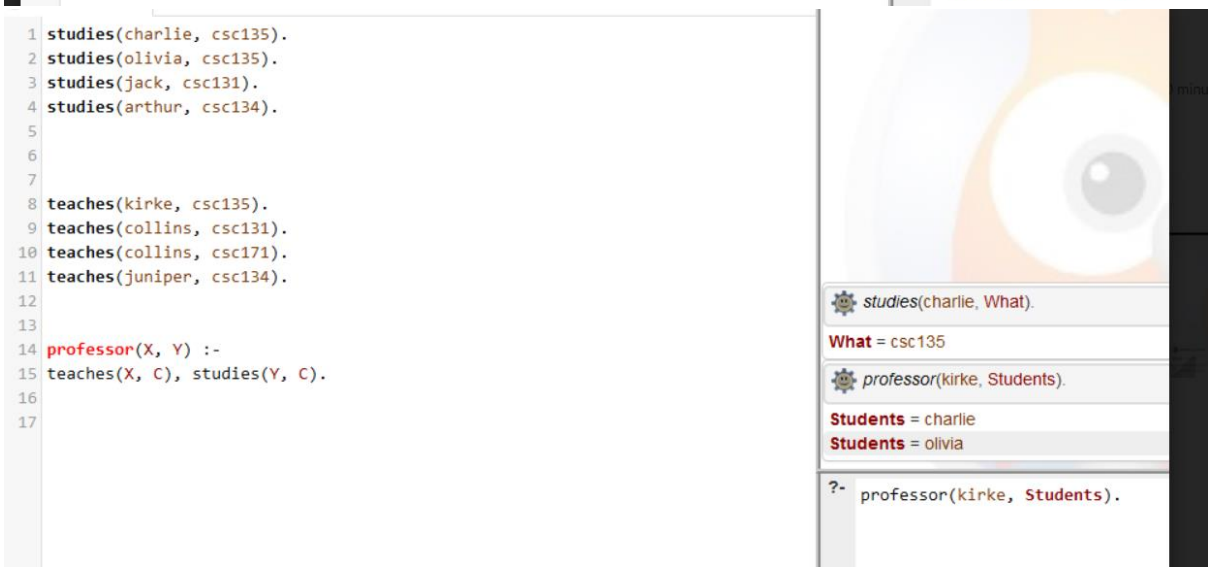
meal(X) , *lunch*(X).

X = sandwich

Next | 10 | 100 | 1,000 | Stop

?- meal(X) , lunch(X).


SWISH
File
Edit
Examples
Help
191 user
Program
1 studies(charlie, csc135).
2 studies(olivia, csc135).
3 studies(jack, csc131).
4 studies(arthur, csc134).
8 teaches(kirke, csc135).
9 teaches(collins, csc131).
10 teaches(collins, csc171).
11 teaches(juniper, csc134).
14 professor(X, Y) :-
15 teaches(X, C), studies(Y, C).
studies(charlie, What).
What = csc135
?- studies(charlie, What).


1 studies(charlie, csc135).
2 studies(olivia, csc135).
3 studies(jack, csc131).
4 studies(arthur, csc134).
8 teaches(kirke, csc135).
9 teaches(collins, csc131).
10 teaches(collins, csc171).
11 teaches(juniper, csc134).
14 professor(X, Y) :-
15 teaches(X, C), studies(Y, C).
studies(charlie, What).
What = csc135
professor(kirke, Students).
Students = charlie
Students = olivia
?- professor(kirke, Students).

SWISH File Edit Examples Help 168 users online Search

Program

```
move(1,X,Y,_) :-
   write('Move top disk from '), write(X), write(' to '), write(Y), nl.
move(N,X,Y,Z) :-
   N>1,
   M is N-1,
   move(M,X,Z,Y),
   move(1,X,Y,_),
   move(M,Z,Y,X).
```

move(4,source,target,auxiliary).

```
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
Move top disk from source to auxiliary
Move top disk from target to source
Move top disk from target to auxiliary
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
Move top disk from auxiliary to source
Move top disk from target to source
Move top disk from auxiliary to target
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
true
```

?- move(4,source,target,auxiliary).

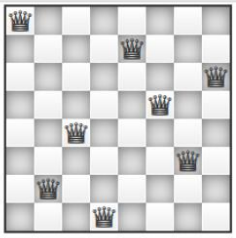
SWISH File Edit Examples Help 253 users online Search

queens

```
% render solutions nicely.
:- use_rendering(chess).
%%    queens(+N, -Queens) is nondet.
%   @param  Queens is a list of column numbers for placing the queens.
%   @author Richard A. O'Keefe (The Craft of Prolog)
queens(N, Queens) :-
    length(Queens, N),
    board(Queens, Board, 0, N, _, _),
    queens(Board, 0, Queens).
board([], [], N, N, _, _).
board([_|Queens], [Col-Vars|Board], Col0, N, [_|VR], VC) :-
    Col is Col0+1,
    functor(Vars, f, N),
    constraints(N, Vars, VR, VC),
    board(Queens, Board, Col, N, VR, [_|VC]).
constraints(0, _, _, _) :- !.
constraints(N, Row, [R|Rs], [C|Cs]) :-
    arg(N, Row, R-C),
    M is N-1,
    constraints(M, Row, Rs, Cs).
queens([], _, []).
queens([C|Cs], Row0, [Col|Solution]) :-
    Row is Row0+1,
    select(Col-Vars, [C|Cs], Board),
    arg(Row, Vars, Row-Row),
    queens(Board, Row, Solution).
/** <examples>
```

queens(8, Queens).

Queens =



Queens =

?- queens(8, Queens).

Examples History Solutions