First Prolog Assignment

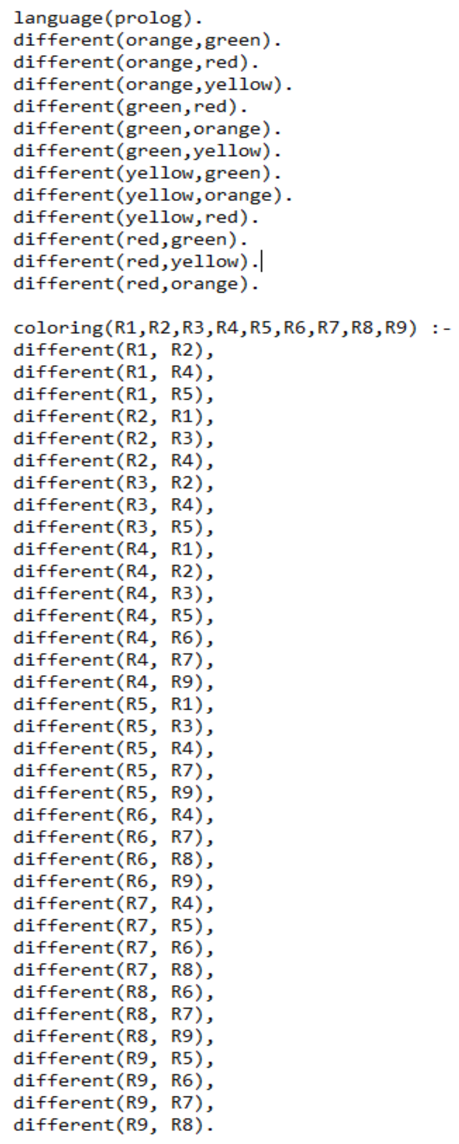
Justin Cesarini

Abstract

Overall I learned a lot in this assignment about Prolog. I did not have much knowledge about it coming into this class, but this assignment really helped me nail the basics as well as show me things a little bit more complex. The pokemon task was one of my favorites because it gave me insight on just how pieces of the actual pokemon game were formed. Overall I really enjoyed this assignment.

Task 1

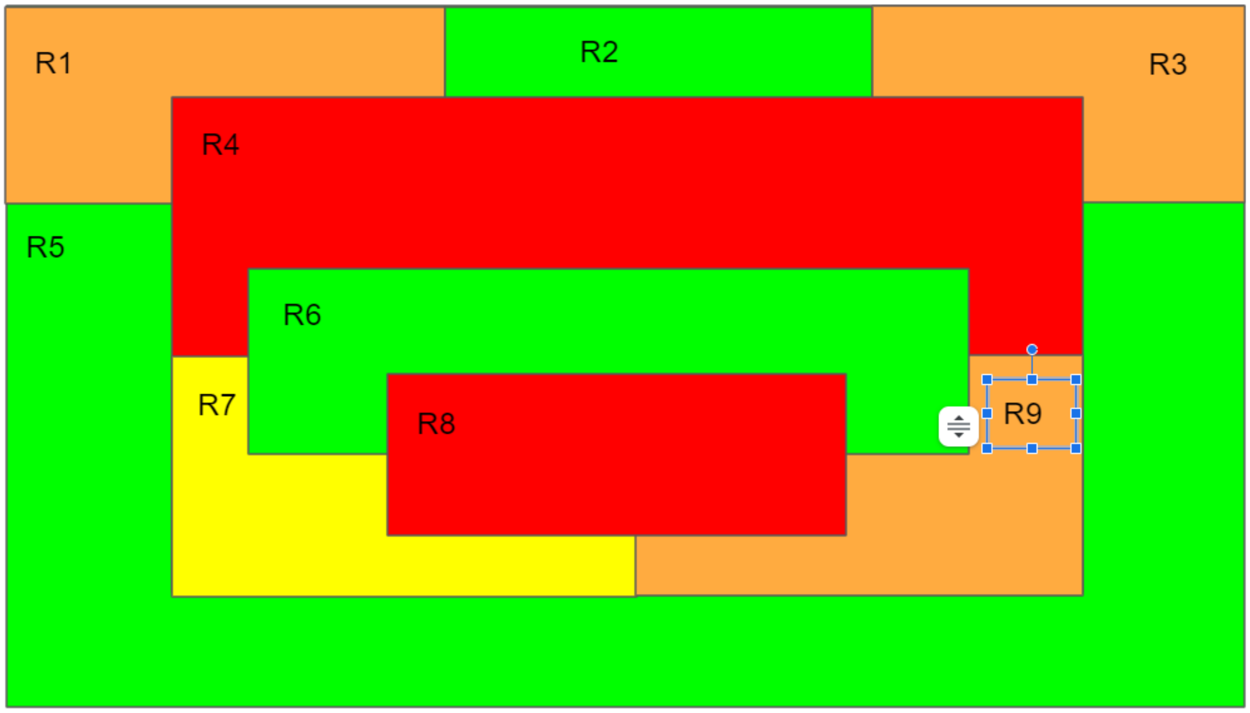
Code:



```
language(prolog).
different(orange,green).
different(orange,red).
different(orange,yellow).
different(green,red).
different(green,orange).
different(green,yellow).
different(yellow,green).
different(yellow,orange).
different(yellow,red).
different(red,green).
different(red,yellow).
different(red,orange).

coloring(R1,R2,R3,R4,R5,R6,R7,R8,R9) :-
different(R1, R2),
different(R1, R4),
different(R1, R5),
different(R2, R1),
different(R2, R3),
different(R2, R4),
different(R3, R2),
different(R3, R4),
different(R3, R5),
different(R4, R1),
different(R4, R2),
different(R4, R3),
different(R4, R5),
different(R4, R6),
different(R4, R7),
different(R4, R9),
different(R5, R1),
different(R5, R3),
different(R5, R4),
different(R5, R7),
different(R5, R9),
different(R6, R4),
different(R6, R7),
different(R6, R8),
different(R6, R9),
different(R7, R4),
different(R7, R5),
different(R7, R6),
different(R7, R8),
different(R8, R6),
different(R8, R7),
different(R8, R9),
different(R9, R5),
different(R9, R6),
different(R9, R7),
different(R9, R8).
```

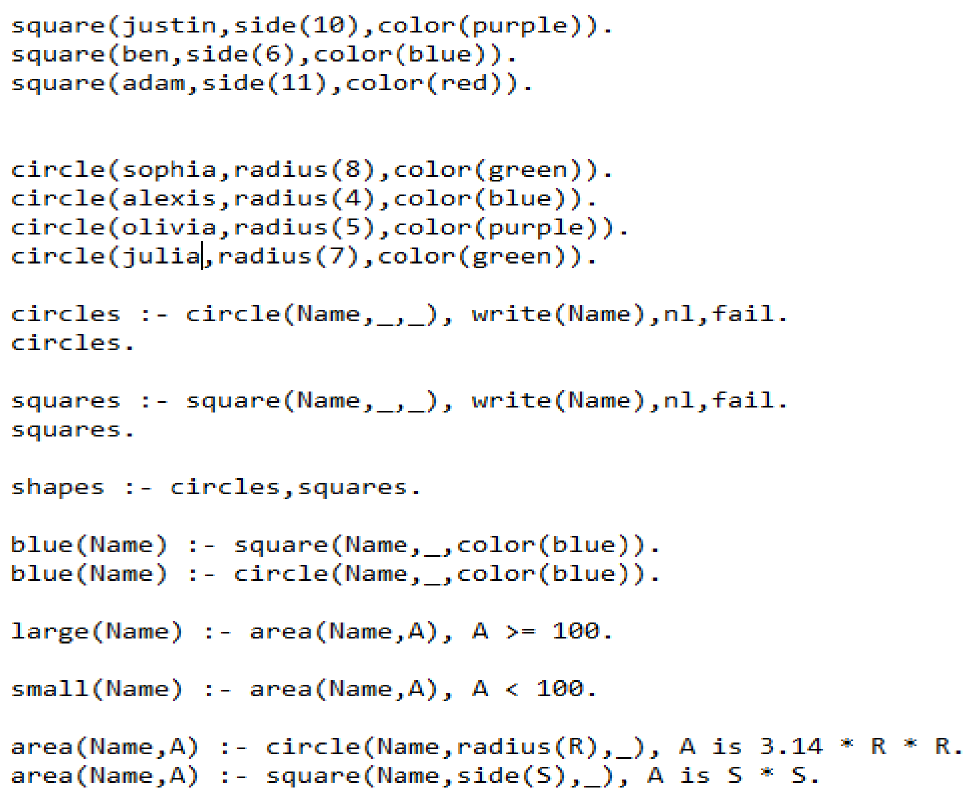
Demo:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('c:\\prolog\\prolog.pl').
true.

?- language(prolog).
true.

?- consult('c:\\prolog\\prolog.pl').
true.

?- coloring(R1,R2,R3,R4,R5,R6,R7,R8,R9).
R1 = R3, R3 = R9, R9 = orange,
R2 = R5, R5 = R6, R6 = green,
R4 = R8, R8 = red,
R7 = yellow
```



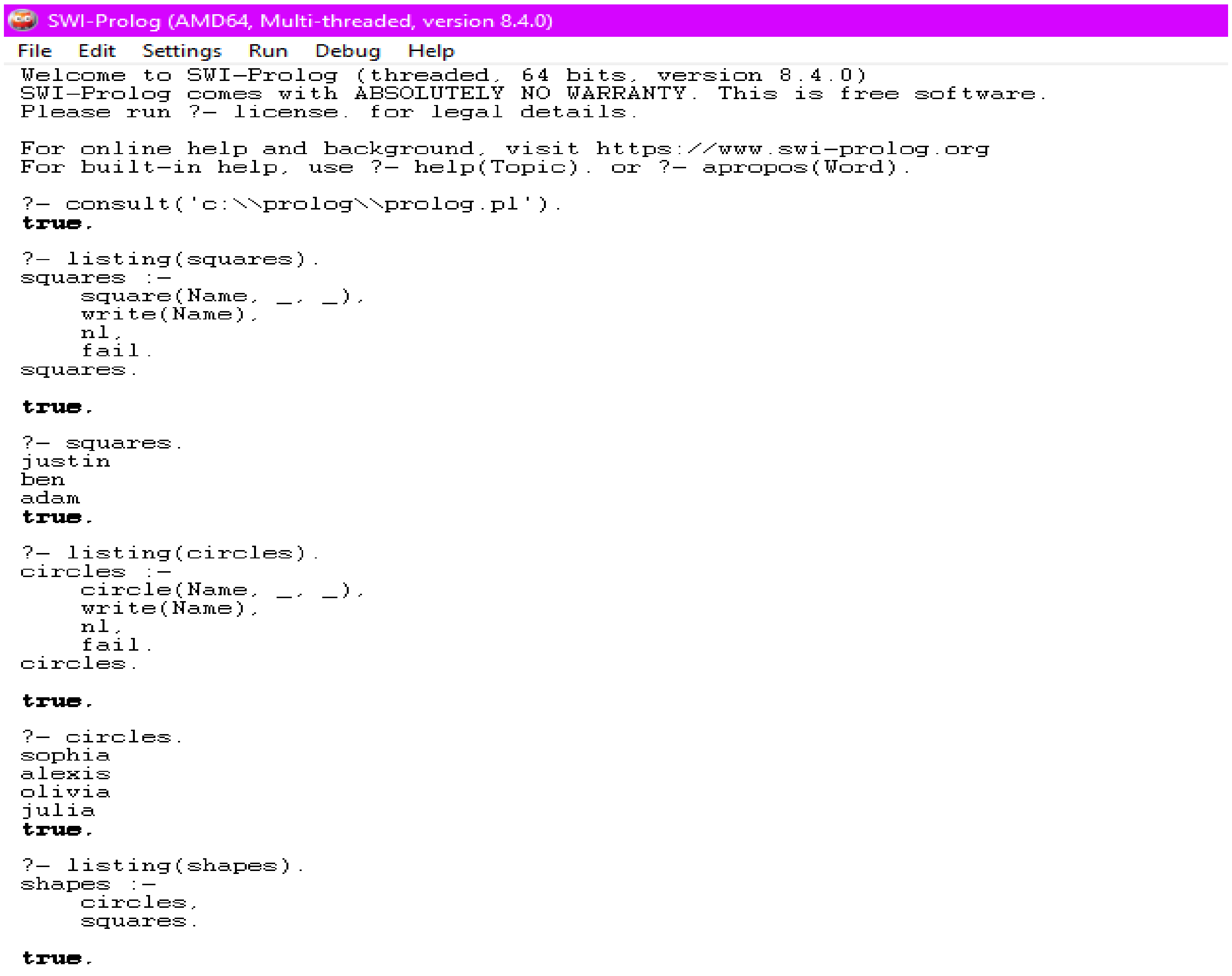
Task 2

Code:

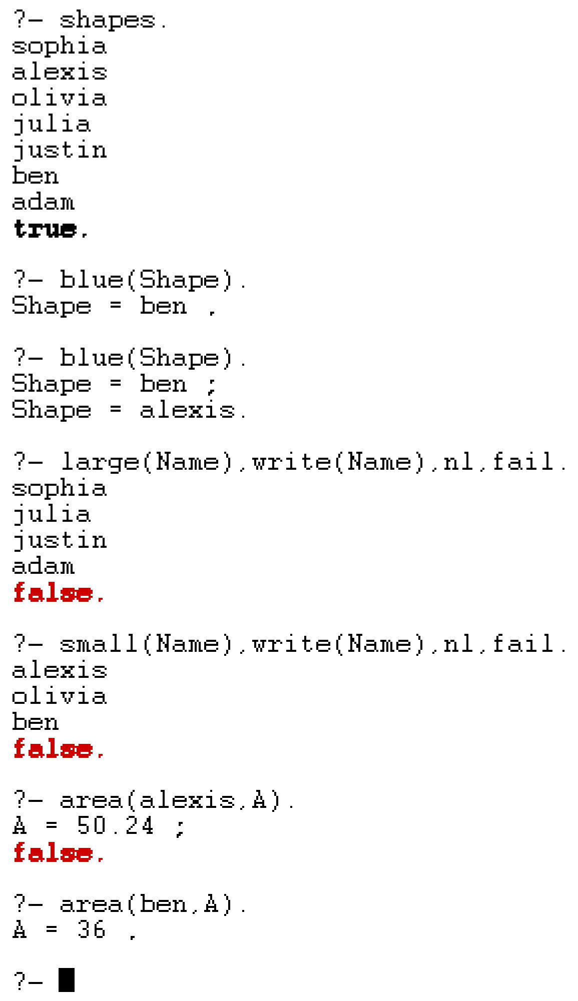


```
square(justin,side(10),color(purple)).
square(ben,side(6),color(blue)).
square(adam,side(11),color(red)).


circle(sophia,radius(8),color(green)).
circle(alexis,radius(4),color(blue)).
circle(olivia,radius(5),color(purple)).
circle(julia,radius(7),color(green)).

circles :- circle(Name,_,_), write(Name),nl,fail.
circles.

squares :- square(Name,_,_), write(Name),nl,fail.
squares.

shapes :- circles,squares.

blue(Name) :- square(Name,_,color(blue)).
blue(Name) :- circle(Name,_,color(blue)).

large(Name) :- area(Name,A), A >= 100.

small(Name) :- area(Name,A), A < 100.

area(Name,A) :- circle(Name,radius(R),_), A is 3.14 * R * R.
area(Name,A) :- square(Name,side(S),_), A is S * S.
```

Demo:


SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('c:\\prolog\\prolog.pl').
true.

?- listing(squares).
squares :-
        square(Name, _, _),
        write(Name),
        nl,
        fail.
squares.

true.

?- squares.
justin
ben
adam
true.

?- listing(circles).
circles :-
        circle(Name, _, _),
        write(Name),
        nl,
        fail.
circles.

true.

?- circles.
sophia
alexis
olivia
julia
true.

?- listing(shapes).
shapes :-
        circles,
        squares.

true.
```



```
?- shapes.
sophia
alexis
olivia
julia
justin
ben
adam
true.

?- blue(Shape).
Shape = ben ,

?- blue(Shape).
Shape = ben ;
Shape = alexis.

?- large(Name),write(Name),nl,fail.
sophia
julia
justin
adam
false.

?- small(Name),write(Name),nl,fail.
alexis
olivia
ben
false.

?- area(alexis,A).
A = 50.24 ;
false.

?- area(ben,A).
A = 36 ,

?- 
```

Task 3

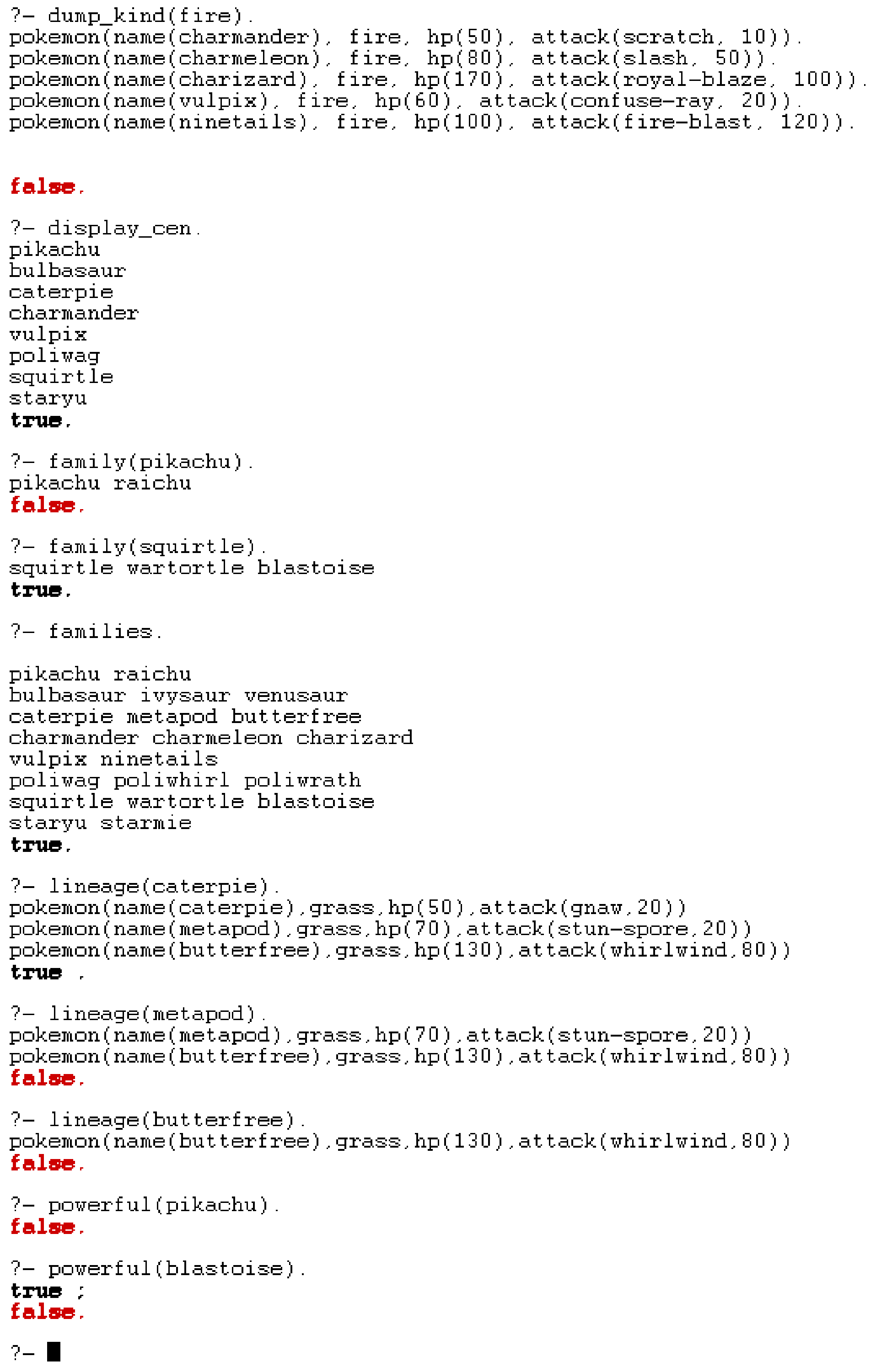
Code:

```
cen(pikachu).
cen(bulbasaur).
cen(caterpie).
cen(charmander).
cen(vulpix).
cen(poliwag).
cen(squirtle).
cen(staryu).

evolves(pikachu,raichu).
evolves(bulbasaur,ivysaur).
evolves(ivysaur,venusaur).
evolves(caterpie,metapod).
evolves(metapod,butterfree).
evolves(charmander,charmeleon).
evolves(charmeleon,charizard).
evolves(vulpix,ninetails).
evolves(poliwag,poliwhirl).
evolves(poliwhirl,poliwrath).
evolves(squirtle,wartortle).
evolves(wartortle,blastoise).
evolves(staryu,starmie).

pokemon(name(bulbasaur), grass, hp(40), attack(leech-seed, 20)).
pokemon(name(ivysaur), grass, hp(60), attack(vine-whip, 30)).
pokemon(name(venusaur), grass, hp(140), attack(poison-powder, 70)).
pokemon(name(caterpie), grass, hp(50), attack(gnaw, 20)).
pokemon(name(metapod), grass, hp(70), attack(stun-spore, 20)).
pokemon(name(butterfree), grass, hp(130), attack(whirlwind, 80)).
pokemon(name(charmander), fire, hp(50), attack(scratch, 10)).
pokemon(name(charmeleon), fire, hp(80), attack(slash, 50)).
pokemon(name(charizard), fire, hp(170), attack(royal-blaze, 100)).
pokemon(name(vulpix), fire, hp(60), attack(confuse-ray, 20)).
pokemon(name(ninetails), fire, hp(100), attack(fire-blast, 120)).
pokemon(name(poliwag), water, hp(60), attack(water-gun, 30)).
pokemon(name(poliwhirl), water, hp(80), attack(amnesia, 30)).
pokemon(name(poliwrath), water, hp(140), attack(dashing-punch, 50)).
pokemon(name(squirtle), water, hp(40), attack(bubble, 10)).
pokemon(name(wartortle), water, hp(80), attack(waterfall, 60)).
pokemon(name(blastoise), water, hp(140), attack(hydro-pump, 60)).
pokemon(name(staryu), water, hp(40), attack(slap, 20)).
pokemon(name(starmie), water, hp(60), attack(star-freeze, 20)).

display_names :- pokemon(name(Name),_,_,_),write(Name),nl,fail.
display_names.

display_attacks :-
pokemon(_,_,_,attack(Attack,_)),write(Attack),nl,fail.
display_attacks.
```

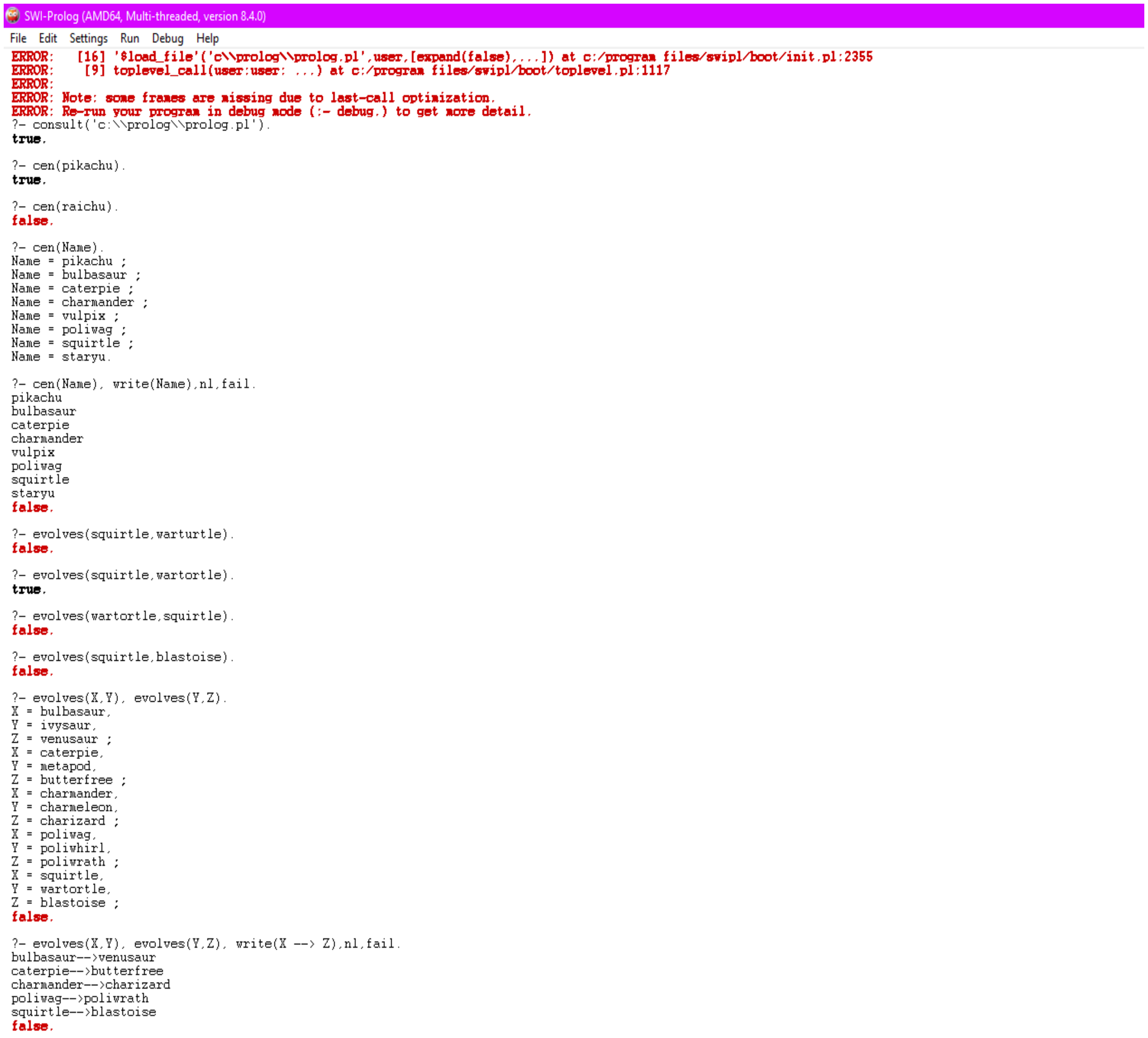
```
powerful(Name) :- pokemon(name(Name),_,_,attack(_,Damage)), Damage >
55.

tough(Name) :- pokemon(name(Name),_,hp(HP),_), HP > 99.

type(Name,Type) :- pokemon(name(Name),T,_,_), Type = T.

dump_kind(Type) :-
listing(pokemon(_,Type,_,_)),nl,fail.

display_cen :- cen(Name),write(Name),nl,fail.
display_cen.

family(Cen) :- evolves(Cen,Y), write(Cen), write(' '), write(Y),
evolves(Y,Z), write(' '), write(Z).

families :- cen(Cen), evolves(Cen,Y), nl, write(Cen), write(' '),
write(Y), evolves(Y,Z), write(' '), write(Z),fail.
families.

lineage(Name) :-
pokemon(name(Name),Type,hp(HP),attack(Attack,Damage)),
write(pokemon(name(Name),Type,hp(HP),attack(Attack,Damage))),nl,
evolves(Name,Y),
pokemon(name(Y),Type2,hp(HP2),attack(Attack2,Damage2)),
write(pokemon(name(Y),Type2,hp(HP2),attack(Attack2,Damage2))),nl,
evolves(Y,Z),
pokemon(name(Z),Type3,hp(HP3),attack(Attack3,Damage3)),
write(pokemon(name(Z),Type3,hp(HP3),attack(Attack3,Damage3))).
```

Queries Demo:

```
?- display_attacks.
leech-seed
vine-whip
poison-powder
gnaw
stun-spore
whirlwind
scratch
slash
royal-blaze
confuse-ray
fire-blast
water-gun
amnesia
dashing-punch
bubble
waterfall
hydro-pump
slap
star-freeze
true.

?- tough(raichu).
false.

?-  tough(venusaur).
true.

?- tough(Name), write(Name), nl, fail.
venusaur
butterfree
charizard
ninetails
poliwrath
blastoise
false.

?- type(caterpie,grass).
true.

?- type(charmander,water).
false.

?- type(N,electric).
false.

?- type(N,water), write(N), nl, fail.
poliwag
poliwhirl
poliwrath
squirtle
wartortle
blastoise
staryu
starmie
false.

?- dump_kind(water).
pokemon(name(poliwag), water, hp(60), attack(water-gun, 30)).
pokemon(name(poliwhirl), water, hp(80), attack(amnesia, 30)).
pokemon(name(poliwrath), water, hp(140), attack(dashing-punch, 50)).
pokemon(name(squirtle), water, hp(40), attack(bubble, 10)).
pokemon(name(wartortle), water, hp(80), attack(waterfall, 60)).
pokemon(name(blastoise), water, hp(140), attack(hydro-pump, 60)).
pokemon(name(staryu), water, hp(40), attack(slap, 20)).
pokemon(name(starmie), water, hp(60), attack(star-freeze, 20)).


false.
```

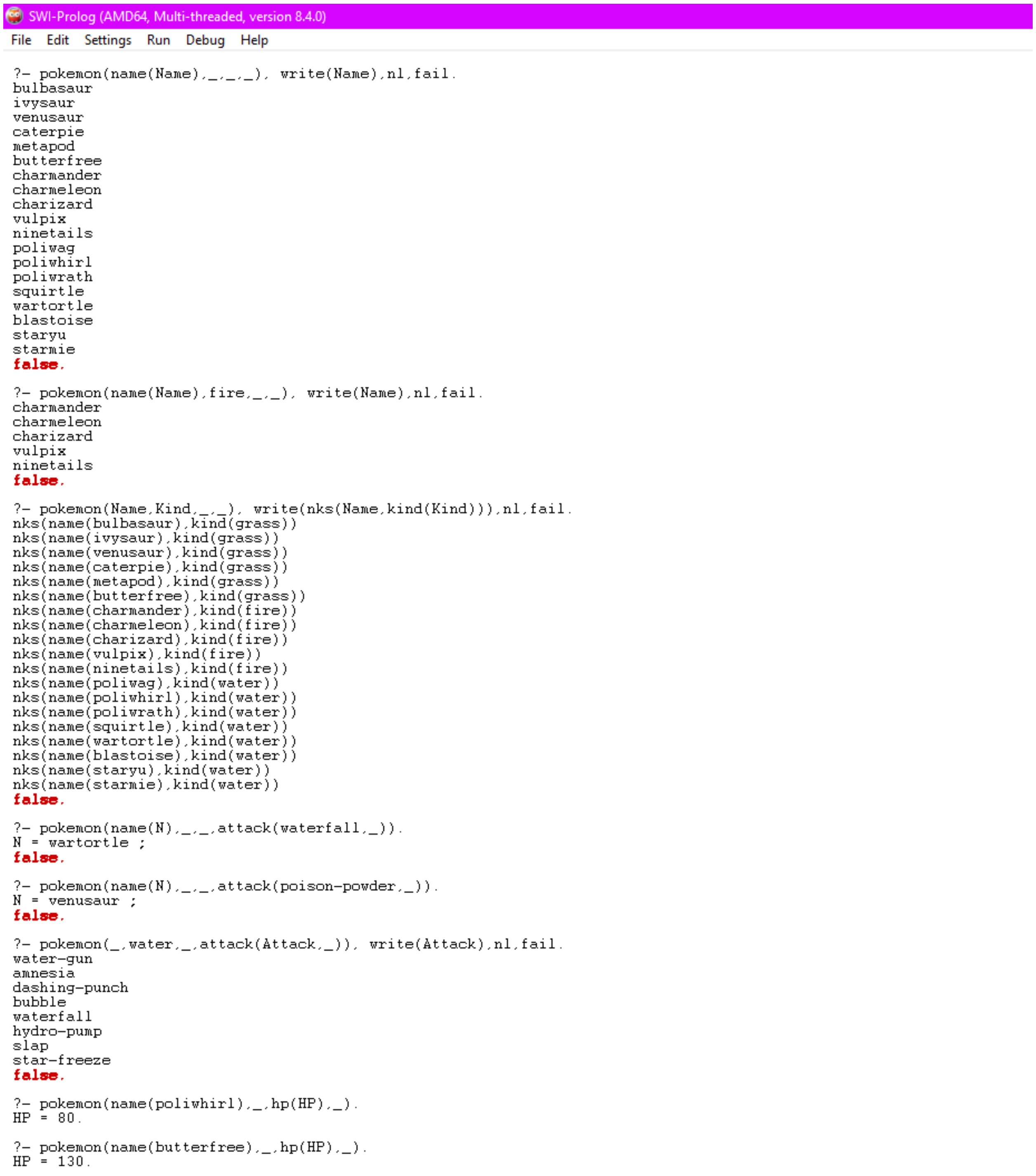


```
?- dump_kind(fire).
pokemon(name(charmander), fire, hp(50), attack(scratch, 10)).
pokemon(name(charmeleon), fire, hp(80), attack(slash, 50)).
pokemon(name(charizard), fire, hp(170), attack(royal-blaze, 100)).
pokemon(name(vulpix), fire, hp(60), attack(confuse-ray, 20)).
pokemon(name(ninetails), fire, hp(100), attack(fire-blast, 120)).


false.

?- display_cen.
pikachu
bulbasaur
caterpie
charmander
vulpix
poliwag
squirtle
staryu
true.

?- family(pikachu).
pikachu raichu
false.

?- family(squirtle).
squirtle wartortle blastoise
true.

?- families.

pikachu raichu
bulbasaur ivysaur venusaur
caterpie metapod butterfree
charmander charmeleon charizard
vulpix ninetails
poliwag poliwhirl poliwrath
squirtle wartortle blastoise
staryu starmie
true.

?- lineage(caterpie).
pokemon(name(caterpie),grass,hp(50),attack(gnaw,20))
pokemon(name(metapod),grass,hp(70),attack(stun-spore,20))
pokemon(name(butterfree),grass,hp(130),attack(whirlwind,80))
true .

?- lineage(metapod).
pokemon(name(metapod),grass,hp(70),attack(stun-spore,20))
pokemon(name(butterfree),grass,hp(130),attack(whirlwind,80))
false.

?- lineage(butterfree).
pokemon(name(butterfree),grass,hp(130),attack(whirlwind,80))
false.

?- powerful(pikachu).
false.

?- powerful(blastoise).
true ;
false.

?-
```

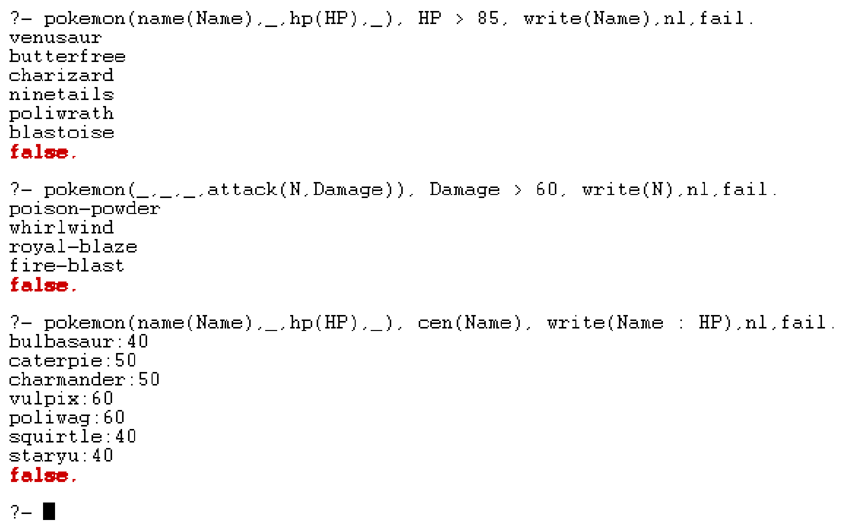

SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help

```
ERROR:   [16] '$load_file'('c\\prolog\\prolog.pl',user,[expand(false),...]) at c:/program files/swipl/boot/init.pl:2355
ERROR:    [9] toplevel_call(user:user: ...) at c:/program files/swipl/boot/toplevel.pl:1117
ERROR:
ERROR: Note: some frames are missing due to last-call optimization.
ERROR: Re-run your program in debug mode (:- debug.) to get more detail.
?- consult('c:\\prolog\\prolog.pl').
true.

?- cen(pikachu).
true.

?- cen(raichu).
false.

?- cen(Name).
Name = pikachu ;
Name = bulbasaur ;
Name = caterpie ;
Name = charmander ;
Name = vulpix ;
Name = poliwag ;
Name = squirtle ;
Name = staryu.

?- cen(Name), write(Name),nl,fail.
pikachu
bulbasaur
caterpie
charmander
vulpix
poliwag
squirtle
staryu
false.

?- evolves(squirtle,warturtle).
false.

?- evolves(squirtle,wartortle).
true.

?- evolves(wartortle,squirtle).
false.

?- evolves(squirtle,blastoise).
false.

?- evolves(X,Y), evolves(Y,Z).
X = bulbasaur,
Y = ivysaur,
Z = venusaur ;
X = caterpie,
Y = metapod,
Z = butterfree ;
X = charmander,
Y = charmeleon,
Z = charizard ;
X = poliwag,
Y = poliwhirl,
Z = poliwrath ;
X = squirtle,
Y = wartortle,
Z = blastoise ;
false.

?- evolves(X,Y), evolves(Y,Z), write(X --> Z),nl,fail.
bulbasaur-->venusaur
caterpie-->butterfree
charmander-->charizard
poliwag-->poliwrath
squirtle-->blastoise
false.
```

Personal additions Demo


SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help

```
?- pokemon(name(Name),_,_,_), write(Name),nl,fail.
bulbasaur
ivysaur
venusaur
caterpie
metapod
butterfree
charmander
charmeleon
charizard
vulpix
ninetails
poliwag
poliwhirl
poliwrath
squirtle
wartortle
blastoise
staryu
starmie
false.

?- pokemon(name(Name),fire,_,_), write(Name),nl,fail.
charmander
charmeleon
charizard
vulpix
ninetails
false.

?- pokemon(Name,Kind,_,_), write(nks(Name,kind(Kind))),nl,fail.
nks(name(bulbasaur),kind(grass))
nks(name(ivysaur),kind(grass))
nks(name(venusaur),kind(grass))
nks(name(caterpie),kind(grass))
nks(name(metapod),kind(grass))
nks(name(butterfree),kind(grass))
nks(name(charmander),kind(fire))
nks(name(charmeleon),kind(fire))
nks(name(charizard),kind(fire))
nks(name(vulpix),kind(fire))
nks(name(ninetails),kind(fire))
nks(name(poliwag),kind(water))
nks(name(poliwhirl),kind(water))
nks(name(poliwrath),kind(water))
nks(name(squirtle),kind(water))
nks(name(wartortle),kind(water))
nks(name(blastoise),kind(water))
nks(name(staryu),kind(water))
nks(name(starmie),kind(water))
false.

?- pokemon(name(N),_,_,attack(waterfall,_)).
N = wartortle ;
false.

?- pokemon(name(N),_,_,attack(poison-powder,_)).
N = venusaur ;
false.

?- pokemon(_,water,_,attack(Attack,_)), write(Attack),nl,fail.
water-gun
amnesia
dashing-punch
bubble
waterfall
hydro-pump
slap
star-freeze
false.

?- pokemon(name(poliwhirl),_,hp(HP),_).
HP = 80.

?- pokemon(name(butterfree),_,hp(HP),_).
HP = 130.
```



```
?- pokemon(name(Name),_,hp(HP),_), HP > 85, write(Name),nl,fail.
venusaur
butterfree
charizard
ninetails
poliwrath
blastoise
false.

?- pokemon(_,_,_,attack(N,Damage)), Damage > 60, write(N),nl,fail.
poison-powder
whirlwind
royal-blaze
fire-blast
false.

?- pokemon(name(Name),_,hp(HP),_), cen(Name), write(Name : HP),nl,fail.
bulbasaur:40
caterpie:50
charmander:50
vulpix:60
poliwag:60
squirtle:40
staryu:40
false.

?- 
```

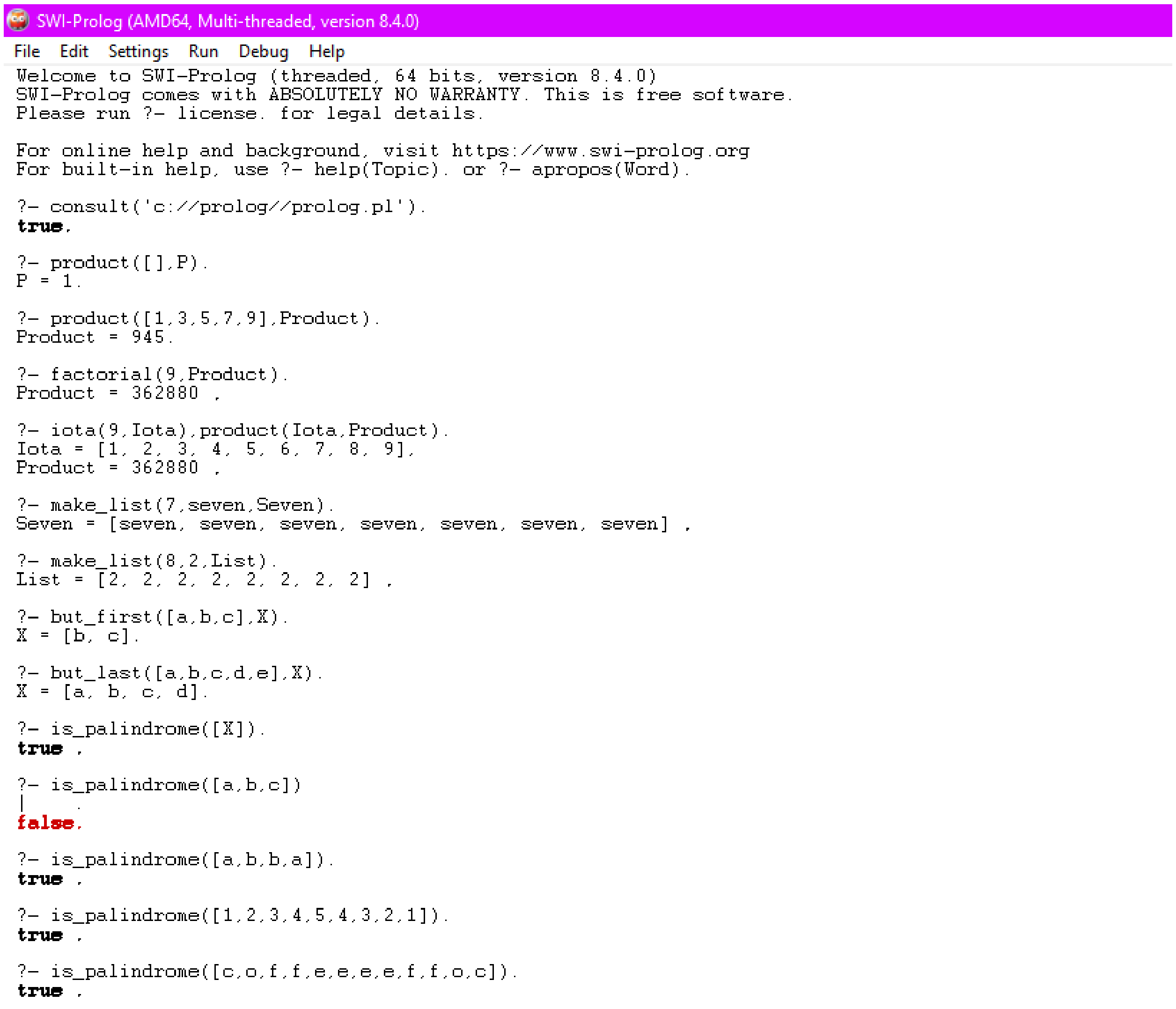
Task 4

Code:

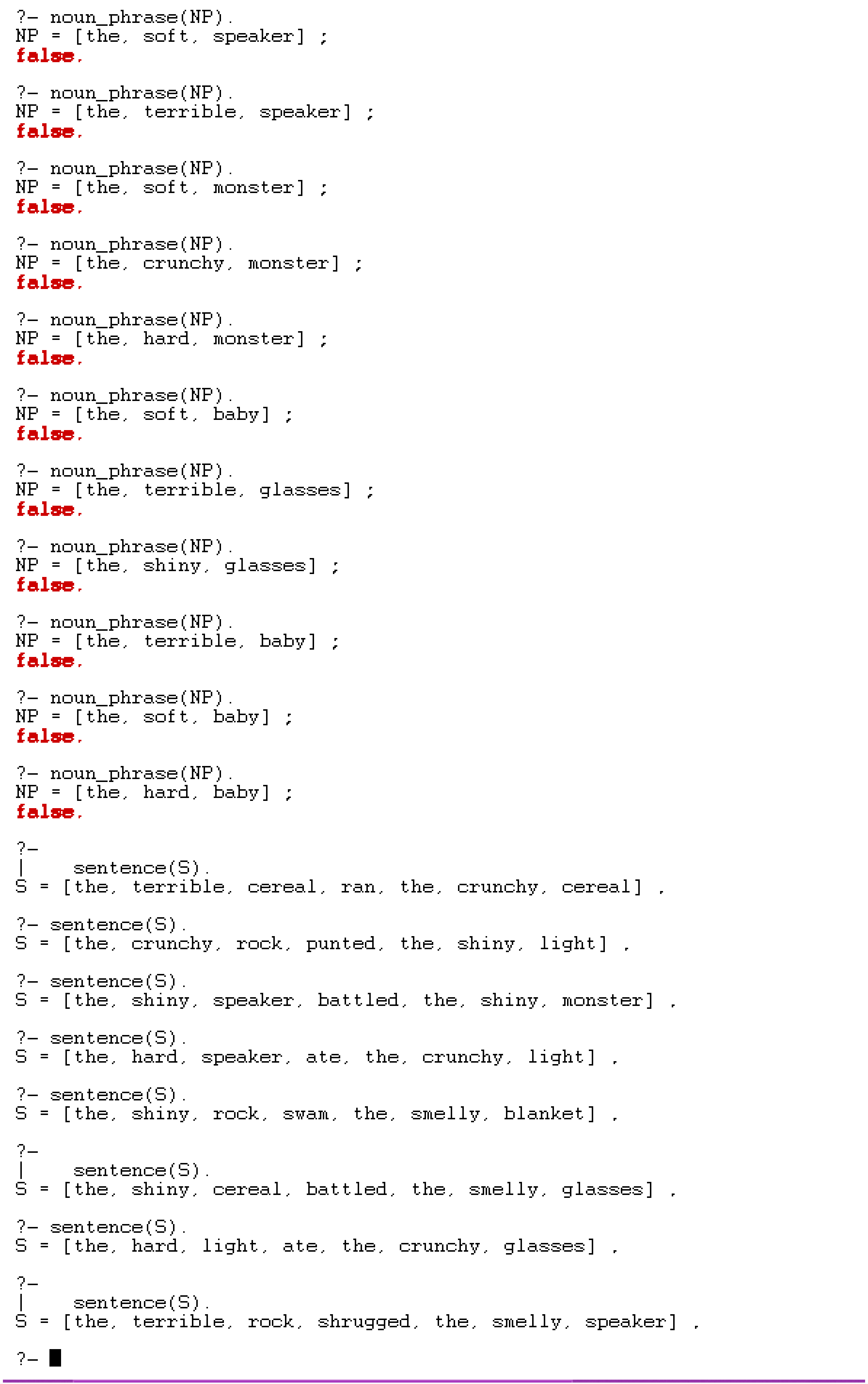
```
first([H|_], H).

rest([_|T], T).

last([H|[]], H).
last([_|T], Result) :- last(T, Result).

nth(0,[H|_],H).
nth(N,[_|T],E) :- K is N - 1, nth(K,T,E).

writelist([]).
writelist([H|T]) :- write(H), nl, writelist(T).

sum([],0).
sum([Head|Tail],Sum) :-
sum(Tail,SumOfTail),
Sum is Head + SumOfTail.

add_first(X,L,[X|L]).

add_last(X,[],[X]).
add_last(X,[H|T],[H|TX]) :- add_last(X,T,TX).

iota(0,[]).
iota(N,IotaN) :-
K is N - 1,
iota(K,IotaK),
add_last(N,IotaK,IotaN).

pick(L,Item) :-
length(L,Length),
random(0,Length,RN),
nth(RN,L,Item).

make_set([],[]).
make_set([H|T],TS) :-
member(H,T),
make_set(T,TS).
make_set([H|T],[H|TS]) :-
make_set(T,TS).
```

```
product([],1).
product([Head|Tail],Product) :-
 product(Tail, ProductOfTail),
 Product is Head * ProductOfTail.

factorial(0,0).
factorial(Num,Name) :- iota(Num,Iota), product(Iota,Product), Name is
Product.

make_list(0,_,[]).
make_list(Num,Element,Name) :-
 K is Num - 1,
 make_list(K,Element,NameK),
 add_last(Element,NameK,Name).

but_first([],[]).
but_first([_],[]).
but_first([_|N],N).

but_last([],[]).
but_last([_],[]).
but_last([H|T], Name) :-
 reverse(T, [_|B]), reverse(B, RDC), add_first(H,RDC,Name).

is_palindrome([]).
is_palindrome([_]).
is_palindrome(List) :-
 first(List,FirstChar), last(List,LastChar),
 FirstChar = LastChar,
 but_first(List,A), but_last(A,B),
 is_palindrome(B).

noun_phrase(Name) :-
 pick([crunchy, shiny, smelly, hard, soft,
terrible],Adj),
 pick([speaker, light, rock, blanket, glasses, cereal, monster,
baby],Noun),
 add_last(Adj,[the],Start), add_last(Noun,Start,Name).


sentence(Name) :-
 noun_phrase(N), noun_phrase(M),
 pick([ran, battled, shrugged, punted, ate, swam,
threw],Verb),
 add_last(Verb,N,T),
 append(T,M,Name).
```

Demo:


SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('c://prolog//prolog.pl').
true.

?- product([],P).
P = 1.

?- product([1,3,5,7,9],Product).
Product = 945.

?- factorial(9,Product).
Product = 362880 .

?- iota(9,Iota),product(Iota,Product).
Iota = [1, 2, 3, 4, 5, 6, 7, 8, 9],
Product = 362880 .

?- make_list(7,seven,Seven).
Seven = [seven, seven, seven, seven, seven, seven, seven] .

?- make_list(8,2,List).
List = [2, 2, 2, 2, 2, 2, 2, 2] .

?- but_first([a,b,c],X).
X = [b, c].

?- but_last([a,b,c,d,e],X).
X = [a, b, c, d].

?- is_palindrome([X]).
true .

?- is_palindrome([a,b,c])
|    .
false.

?- is_palindrome([a,b,b,a]).
true .

?- is_palindrome([1,2,3,4,5,4,3,2,1]).
true .

?- is_palindrome([c,o,f,f,e,e,e,e,f,f,o,c]).
true .
```



```
?- noun_phrase(NP).
NP = [the, soft, speaker] ;
false.

?- noun_phrase(NP).
NP = [the, terrible, speaker] ;
false.

?- noun_phrase(NP).
NP = [the, soft, monster] ;
false.

?- noun_phrase(NP).
NP = [the, crunchy, monster] ;
false.

?- noun_phrase(NP).
NP = [the, hard, monster] ;
false.

?- noun_phrase(NP).
NP = [the, soft, baby] ;
false.

?- noun_phrase(NP).
NP = [the, terrible, glasses] ;
false.

?- noun_phrase(NP).
NP = [the, shiny, glasses] ;
false.

?- noun_phrase(NP).
NP = [the, terrible, baby] ;
false.

?- noun_phrase(NP).
NP = [the, soft, baby] ;
false.

?- noun_phrase(NP).
NP = [the, hard, baby] ;
false.

?-
|    sentence(S).
S = [the, terrible, cereal, ran, the, crunchy, cereal] .

?- sentence(S).
S = [the, crunchy, rock, punted, the, shiny, light] .

?- sentence(S).
S = [the, shiny, speaker, battled, the, shiny, monster] .

?- sentence(S).
S = [the, hard, speaker, ate, the, crunchy, light] .

?- sentence(S).
S = [the, shiny, rock, swam, the, smelly, blanket] .

?-
|    sentence(S).
S = [the, shiny, cereal, battled, the, smelly, glasses] .

?- sentence(S).
S = [the, hard, light, ate, the, crunchy, glasses] .

?-
|    sentence(S).
S = [the, terrible, rock, shrugged, the, smelly, speaker] .

?-
```