Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 01

Submitted by:

Name: Zannat Hossain Tamim

Roll No:1970

Lab Date:08.06.23

Submission Date: 12.06.23

Lab Report # Day 01

Example 1:

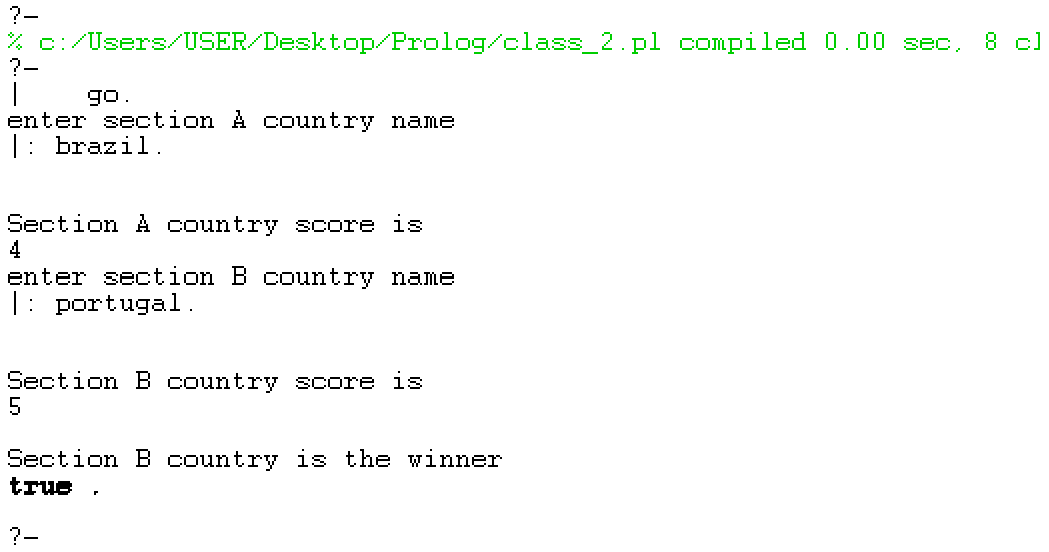
Clause:

```
goal(brazil,4).
goal(germany,3).
goal(france,1).
goal(argentina,2).
goal(portugal,5).
goal(japan,1).
go:-
write('enter section A country name'),nl,
read(X),nl,
goal(X,Y),nl,
write('Section A country score is '),nl,
write(Y),nl,
write('enter section B country name'),nl,
read(P),nl,
goal(P,Q),nl,
write('Section B country score is '),nl,
write(Q),nl,
compare(Y,Q).
compare(Y,Q):-
Y>Q,nl,
write('Section A country is the winner');
Y<Q,nl,
write('Section B country is the winner');
Y=:=Q,nl,
write('Draw in both section').
```

Queries:

1. ?- go.
2. brazil.
3. portugal.

Result :



```
?-
% c:/Users/USER/Desktop/Prolog/class_2.pl compiled 0.00 sec, 8 cl
?-
|    go.
enter section A country name
|: brazil.


Section A country score is
4
enter section B country name
|: portugal.


Section B country score is
5

Section B country is the winner
true .

?-
```

Example 2:

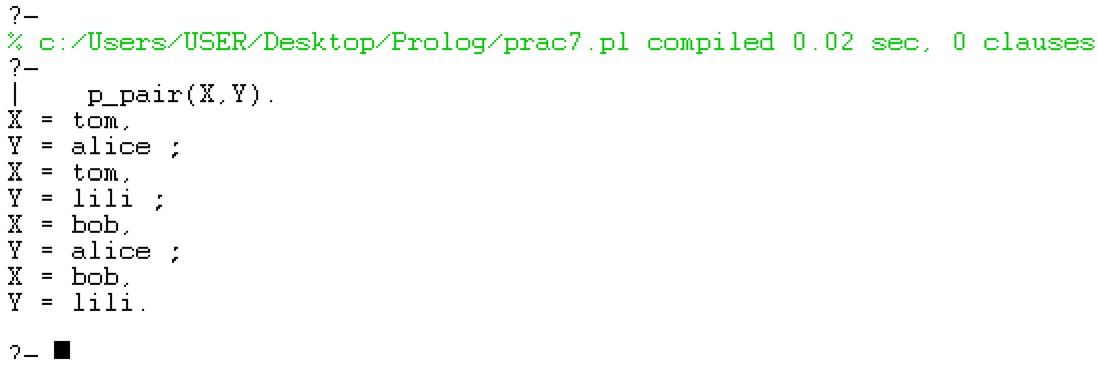
Clause:

```
boy(tom).
boy(bob).
girl(alice).
girl(lili).
p_pair(X,Y) :- boy(X),girl(Y).
```

Queries:

1. ?- *p_pair(X,Y).*

Result :



```
?-
% c:/Users/USER/Desktop/Prolog/prac7.pl compiled 0.02 sec, 0 clauses
?-
|    p_pair(X,Y).
X = tom,
Y = alice ;
X = tom,
Y = lili ;
X = bob,
Y = alice ;
X = bob,
Y = lili.

?- ■
```

Example 3:

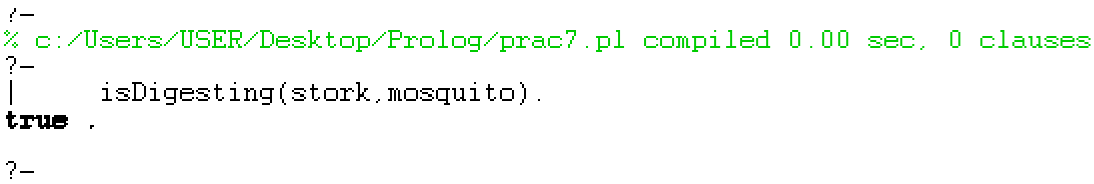
Clause:

```
isDigesting(X,Y):- justAte(X,Y).
isDigesting(X,Y):- justAte(X,Z), isDigesting(Z,Y).
justAte(mosquito,blood(john)).
justAte(frog,mosquito).
justAte(stork,frog).
```

Queries:

1. ?- isDigesting(stork,mosquito).

Result :



```
?-
% c:/Users/USER/Desktop/Prolog/prac7.pl compiled 0.00 sec, 0 clauses
?-
|       isDigesting(stork,mosquito).
true .

?-
```

Example 4:

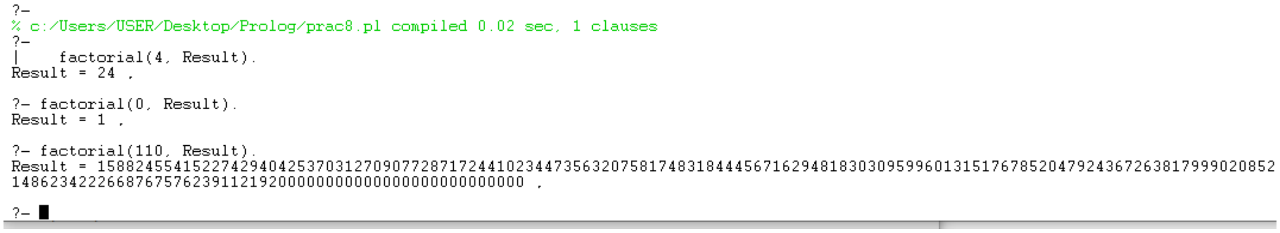
Clause:

```
factorial(0, 1).
factorial(N, Result) :-
 N > 0,
 N1 is N - 1,
 factorial(N1, SubResult),
 Result is N * SubResult.
```

Queries:

1. ?- factorial(4, Result).
2. ?- factorial(0, Result).
3. ?- factorial(110, Result).

Result :



```
?-
% c:/Users/USER/Desktop/Prolog/prac8.pl compiled 0.02 sec, 1 clauses
?-
|    factorial(4, Result).
Result = 24 ,

?- factorial(0, Result).
Result = 1 ,

?- factorial(110, Result).
Result = 1588245541522742940425370312709077287172441023447356320758174831844456716294818303095996013151767852047924367263817999020852
14862342226687675762391121920000000000000000000000000000 ,

?- █
```

Example 5:

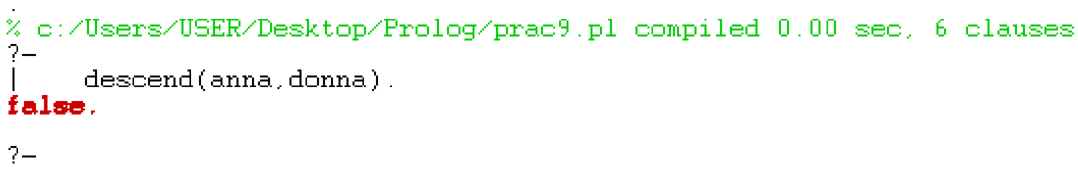
Clause:

child(anna,bridget).
child(bridget,caroline).
child(caroline,donna).
child(donna,emily).
descend(X,Y):- child(X,Y).
descend(X,Y):- child(X,Z), child(Z,Y).

Queries:

1. ?- descend(anna,donna).

Result :



```
.
% c:/Users/USER/Desktop/Prolog/prac9.pl compiled 0.00 sec, 6 clauses
?-
|    descend(anna,donna).
false.

?-
```

Example 6:

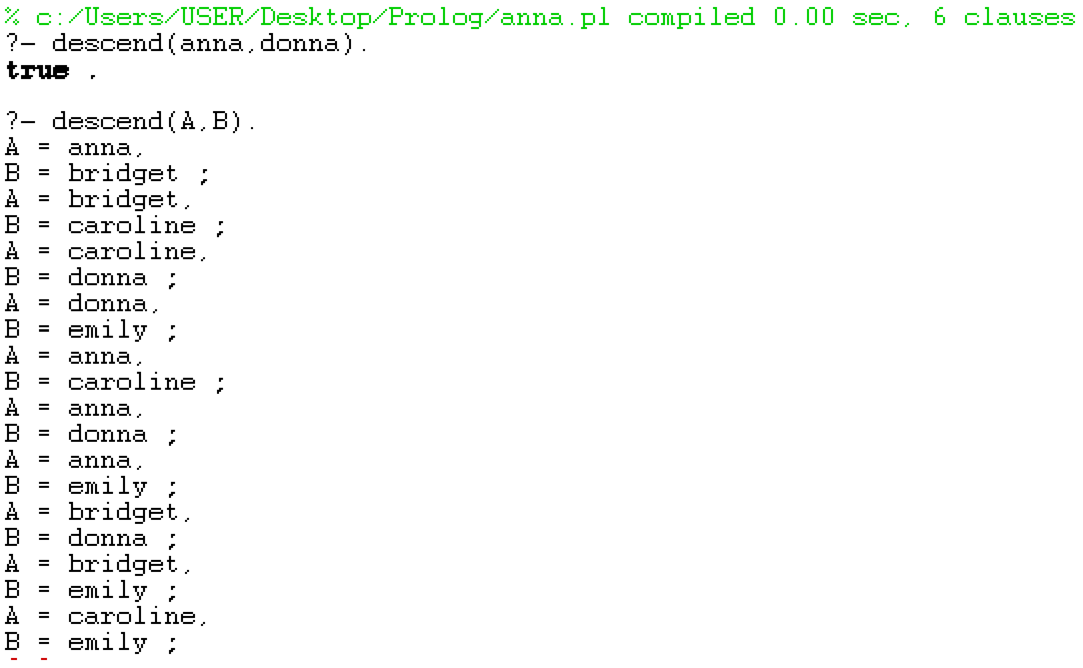
Clause:

child(anna,bridget).
child(bridget,caroline).
child(caroline,donna).
child(donna,emily).
descend(X,Y):- child(X,Y).
descend(X,Y):- child(X,Z), descend(Z,Y).

Queries:

1. `?-descend(anna,donna).
2. `?- descend(A,B).

Result:



```
% c:/Users/USER/Desktop/Prolog/anna.pl compiled 0.00 sec, 6 clauses
?- descend(anna,donna).
true .

?- descend(A,B).
A = anna,
B = bridget ;
A = bridget,
B = caroline ;
A = caroline,
B = donna ;
A = donna,
B = emily ;
A = anna,
B = caroline ;
A = anna,
B = donna ;
A = anna,
B = emily ;
A = bridget,
B = donna ;
A = bridget,
B = emily ;
A = caroline,
B = emily ;
```

Example 7:

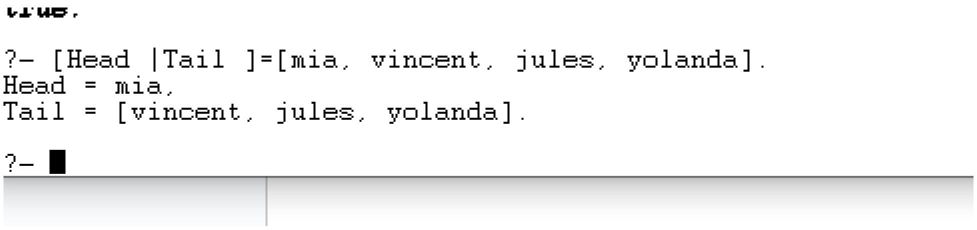
Clause:

[mia, vincent, jules, yolanda].

Queries:

1. `?- [Head |Tail]=[mia, vincent, jules, yolanda].

Result:



```
true.

?- [Head |Tail ]=[mia, vincent, jules, yolanda].
Head = mia,
Tail = [vincent, jules, yolanda].

?- 
```

Example 8:

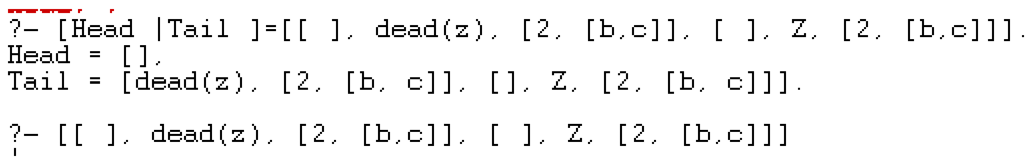
Clause:

[[ ], dead(z), [2, [b,c]], [ ], Z, [2, [b,c]]]

Queries:

1. ?- [Head |Tail]=[[ ], dead(z), [2, [b,c]], [ ], Z, [2, [b,c]]].

Result:



```
?- [Head |Tail ]=[[ ], dead(z), [2, [b,c]], [ ], Z, [2, [b,c]]].
Head = [],
Tail = [dead(z), [2, [b, c]], [], Z, [2, [b, c]]].

?- [[ ], dead(z), [2, [b,c]], [ ], Z, [2, [b,c]]]
```

Example 9:

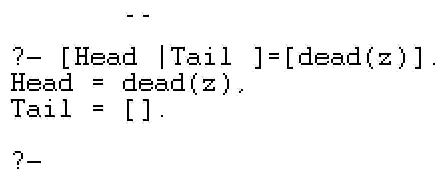
Clause:

[dead(z)]

Queries:

1. ?- [Head |Tail]=[dead(z)].

Result:



```
        - -
?- [Head |Tail ]=[dead(z)].
Head = dead(z),
Tail = [].

?-
```

Example 10:

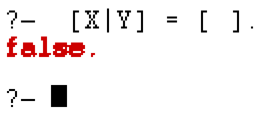
Clause:

[X|Y] = [ ]

Queries:

1. ?- [X|Y] = [ ].

Result:



```
?-  [X|Y] = [ ].
false.

?- █
```

Example 11:

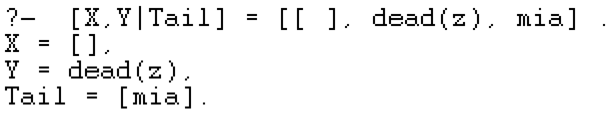
Clause:

[[ ], dead(z), mia] .

Queries:

1. ?- [X,Y|Tail] = [[ ], dead(z), mia] .

Result:



```
?-  [X,Y|Tail] = [[ ], dead(z), mia]  .
X = [],
Y = dead(z),
Tail = [mia].
```

Example 12:

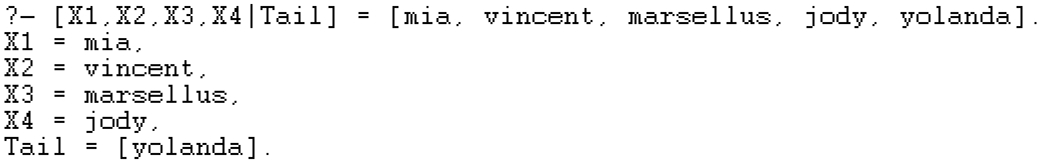
Clause:

[X1,X2,X3,X4|Tail] = [mia, vincent, marsellus, jody, yolanda].

Queries:

1. ?- [X1,X2,X3,X4|Tail] = [mia, vincent, marsellus, jody, yolanda].

Result:



```
?- [X1,X2,X3,X4|Tail] = [mia, vincent, marsellus, jody, yolanda].
X1 = mia,
X2 = vincent,
X3 = marsellus,
X4 = jody,
Tail = [yolanda].
```

Example 13:

Clause:

member(b,[a,b,c]).
member(b,[a,[b,c]]).
member([b,c],[a,[b,c]]).

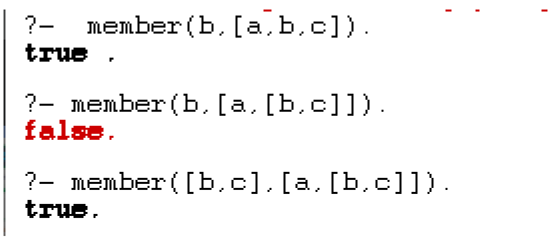
Queries:

1.?- member(b,[a,b,c]).

2. ?- member(b,[a,[b,c]]).

3. ?- member([b,c],[a,[b,c]]).

Result:



```
?-  member(b,[a,b,c]).
true .

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

?- member([b,c],[a,[b,c]]).
true.
```

Example 14:

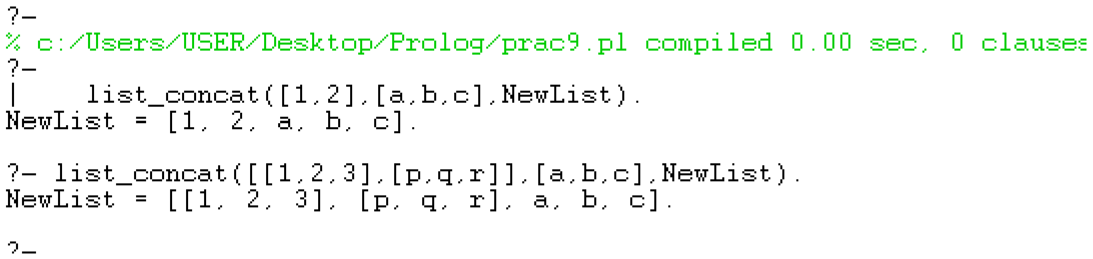
Clause:

list_concat([],L,L).
list_concat([X1|L1],L2,[X1|L3]) :- list_concat(L1,L2,L3).

Queries:

1. ?- list_concat([1,2],[a,b,c],NewList).
2. ?- list_concat([[1,2,3],[p,q,r]],[a,b,c],NewList).

Result:



```
?-
% c:/Users/USER/Desktop/Prolog/prac9.pl compiled 0.00 sec, 0 clauses
?-
|    list_concat([1,2],[a,b,c],NewList).
NewList = [1, 2, a, b, c].

?- list_concat([[1,2,3],[p,q,r]],[a,b,c],NewList).
NewList = [[1, 2, 3], [p, q, r], a, b, c].

?-
```

Example 15:

Clause:

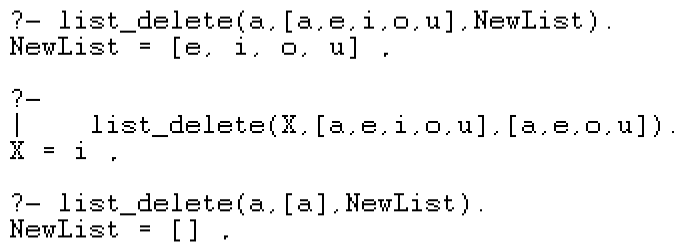
list_delete(X, [X], []).
list_delete(X,[X|L1], L1).
list_delete(X, [Y|L2], [Y|L1]) :- list_delete(X,L2,L1).

Queries:

1. ?-list_delete(a,[a,e,i,o,u],NewList).
2. list_delete(X,[a,e,i,o,u],[a,e,o,u]).

3.?- list_delete(a,[a],NewList)

Result:



```
?- list_delete(a,[a,e,i,o,u],NewList).
NewList = [e, i, o, u] .

?-
|    list_delete(X,[a,e,i,o,u],[a,e,o,u]).
X = i .

?- list_delete(a,[a],NewList).
NewList = [] .
```

Example 16:

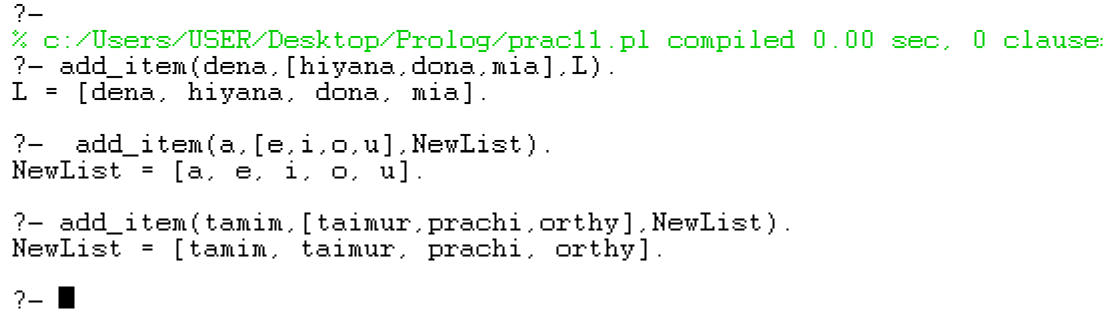
Clause:

```
list_member(X,[X|_]).
list_member(X,[_|TAIL]) :- list_member(X,TAIL).
add_item(A,T,T) :- list_member(A,T),!.
add_item(A,T,[A|T]).
```

Queries:

1. ?-add_item(dena,[hiyana,dona,mia],L).
2. ?- add_item(a,[e,i,o,u],NewList).
3. ?- add_item(tamim,[taimur,prachi,orthy],NewList).

Result:



```
?-
% c:/Users/USER/Desktop/Prolog/prac11.pl compiled 0.00 sec, 0 clause
?- add_item(dena,[hiyana,dona,mia],L).
L = [dena, hiyana, dona, mia].

?-  add_item(a,[e,i,o,u],NewList).
NewList = [a, e, i, o, u].

?- add_item(tamim,[taimur,prachi,orthy],NewList).
NewList = [tamim, taimur, prachi, orthy].

?- █
```

Example 17:

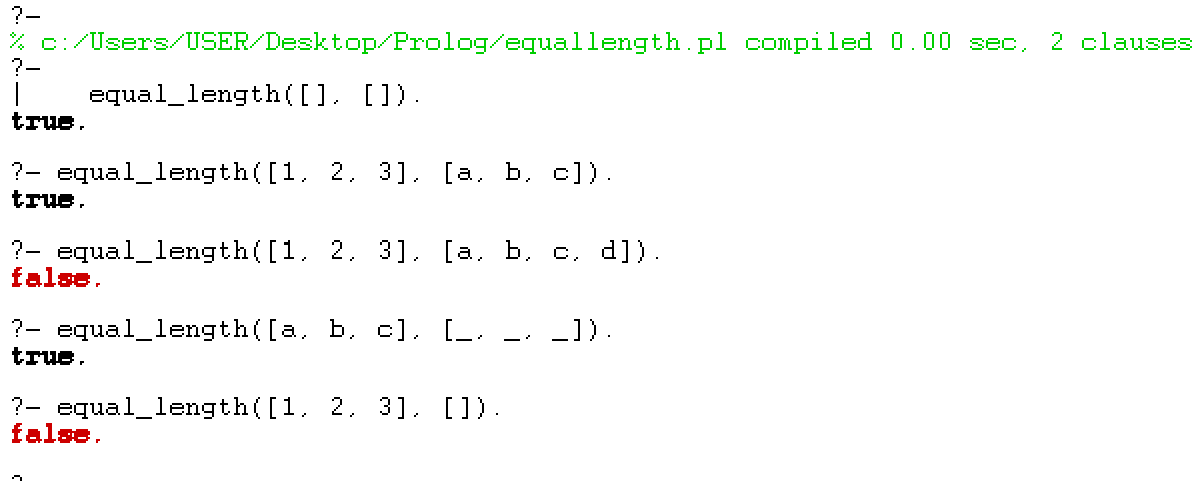
Clause:

equal_length([], []).
equal_length([_|T1], [_|T2]) :- equal_length(T1, T2).

Queries:

1. ?-equal_length([], []).
2. ?- equal_length([1, 2, 3], [a, b, c]).
3. ?- equal_length([1, 2, 3], [a, b, c, d]).
4. ?-equal_length([a, b, c], [_, _, _]).
5. ?- equal_length([1, 2, 3], []).

Result:



```
?-
% c:/Users/USER/Desktop/Prolog/equallength.pl compiled 0.00 sec, 2 clauses
?-
|    equal_length([], []).
true.

?- equal_length([1, 2, 3], [a, b, c]).
true.

?- equal_length([1, 2, 3], [a, b, c, d]).
false.

?- equal_length([a, b, c], [_, _, _]).
true.

?- equal_length([1, 2, 3], []).
false.

?
```

Example 18:

Clause:

```
max(A,B,C):-
            A>B,
            A>C,
            write(A).
max(A,B,C):-
            A>B,
            write(C).
max(_,B,C):-
            B>C,
            write(B).
max(_,_,C):-
            write(C).

get_maxnumber(A,B,C):-
            max(A,B,C).
```

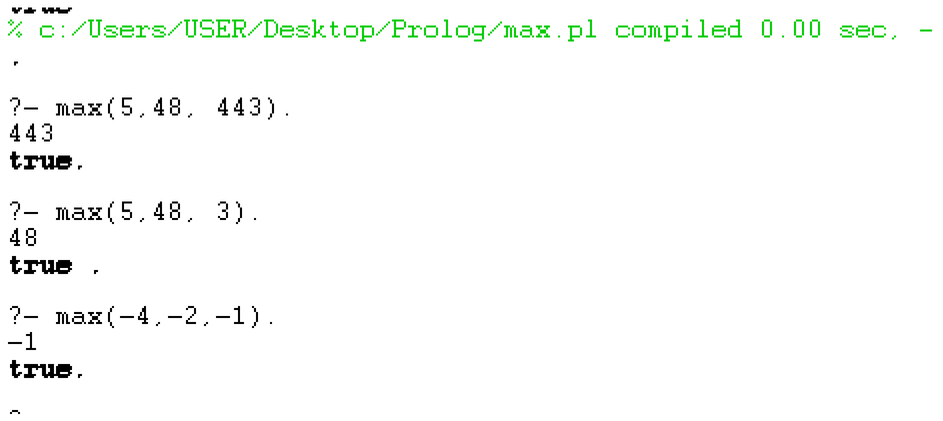
Queries:

1. ?-max(5,48, 443).

2. ?-   max(5,48, 3).

3.?- max(-4,-2,-1).

Result:



```
% c:/Users/USER/Desktop/Prolog/max.pl compiled 0.00 sec, -
.

?- max(5,48, 443).
443
true.

?- max(5,48, 3).
48
true .

?- max(-4,-2,-1).
-1
true.
```

Example 19:

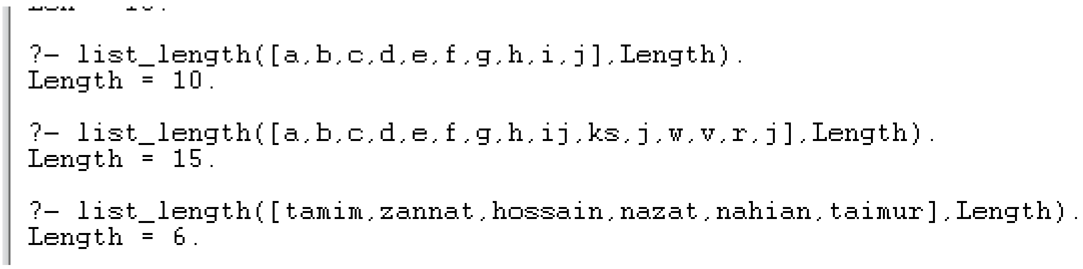
Clause:

```
list_length([],0).
list_length([_|TAIL],N) :- list_length(TAIL,N1), N is N1 + 1.
```

Queries:

1. ?-list_length([a,b,c,d,e,f,g,h,i,j],Length).
2. ?- list_length([a,b,c,d,e,f,g,h,ij,ks,j,w,v,r,j],Length).
3. ?- list_length([tamim,zannat,hossain,nazat,nahian,taimur],Length).

Result:



```
Len = 10.

?- list_length([a,b,c,d,e,f,g,h,i,j],Length).
Length = 10.

?- list_length([a,b,c,d,e,f,g,h,ij,ks,j,w,v,r,j],Length).
Length = 15.

?- list_length([tamim,zannat,hossain,nazat,nahian,taimur],Length).
Length = 6.
```