Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 02

Submitted by:

Name: Md. Shakil Hossain

Roll No: 192340

Lab Date: 08/06/2023

Submission Date: 12/06/2023

Lab Report # Day 01

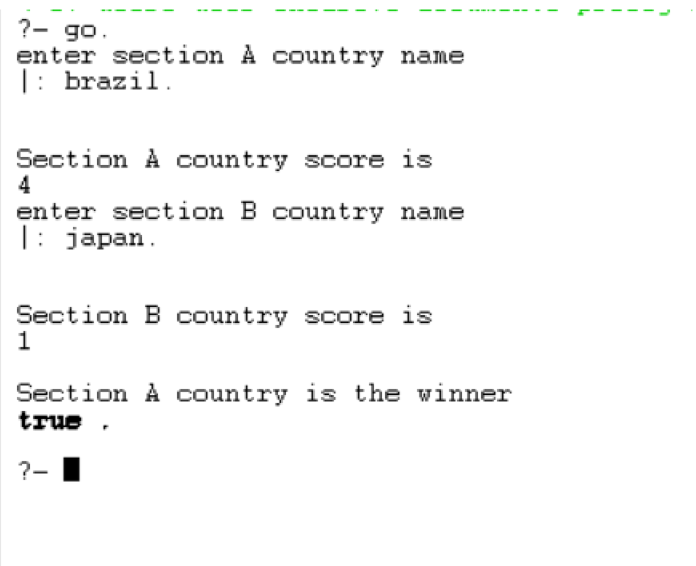
1. Problem name : Comparison-1:

Clause:

```
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:

Result:



```
?- go.
enter section A country name
|: brazil.


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


Section B country score is
1

Section A country is the winner
true .

?- █
```

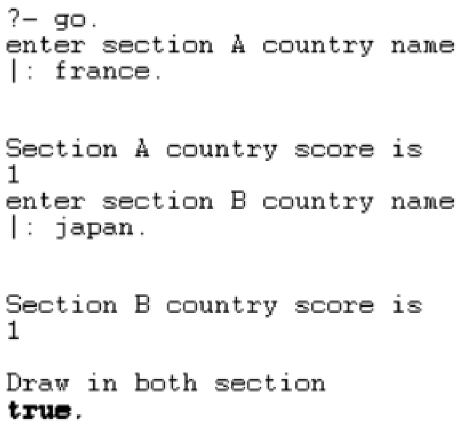
2. Problem name : Comparison-2:

Clause:

```
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:

Result:



```
?- go.
enter section A country name
|: france.


Section A country score is
1
enter section B country name
|: japan.


Section B country score is
1

Draw in both section
true.
```

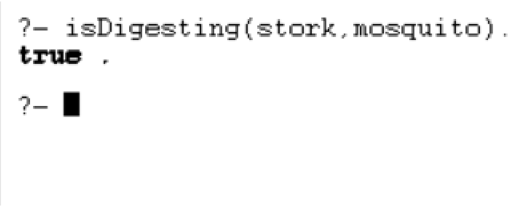
3. Problem Name: Recursion

Clause:

```
% Base case: Factorial of 0 is 1
factorial(0, 1).
% Recursive case: Calculate factorial of N as N multiplied by factorial of N-1
factorial(N, Result) :-
N > 0,
N1 is N - 1,
factorial(N1, SubResult),
Result is N * SubResult.
```

Queries:

Result:



```
?- isDigesting(stork,mosquito).
true .

?- █
```

4. Problem Name: Factorial

Clause:

woman(mia).
woman(jody).
woman(yolanda).
loves(vincent, mia).
loves(marsellus, mia).
loves(pumpkin, honey_bunny).
loves(honey_bunny, pumpkin).

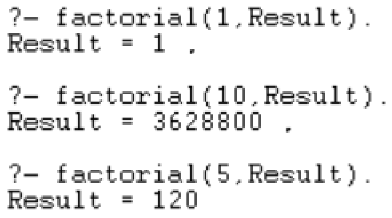
Queries:

?- woman(X).

?- loves(marsellus,X), woman(X).

?- loves(pumpkin,X), woman(X).

Result:



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

?- factorial(10,Result).
Result = 3628800 .

?- factorial(5,Result).
Result = 120
```

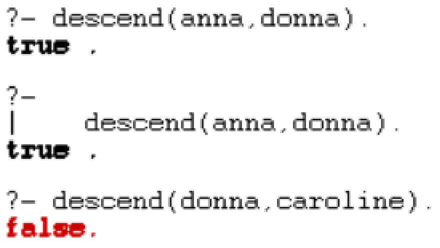
5. Problem Name: Descendent

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:

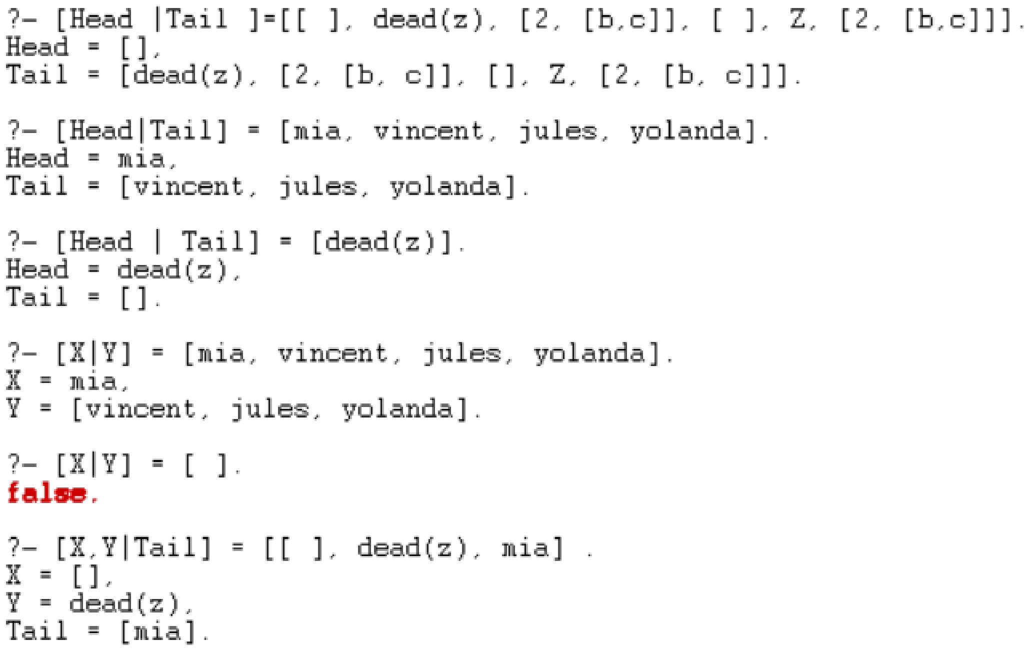
Result:



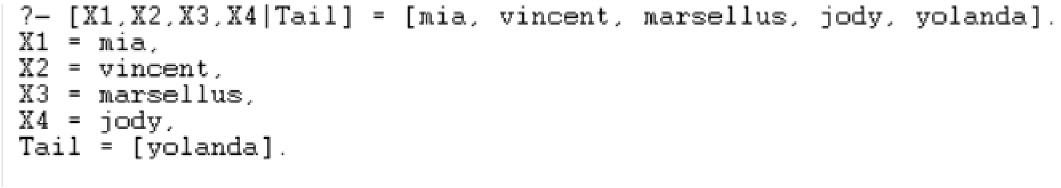
```
?- descend(anna,donna).
true .

?-
|    descend(anna,donna).
true .

?- descend(donna,caroline).
false.
```

6. Problem Name: List

Result:



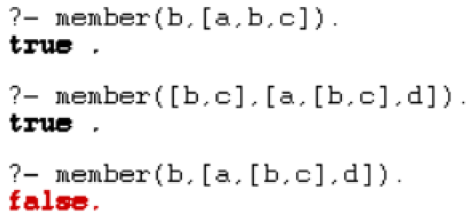
```
?- [Head |Tail ]=[[ ], dead(z), [2, [b,c]], [ ], Z, [2, [b,c]]].
Head = [],
Tail = [dead(z), [2, [b, c]], [], Z, [2, [b, c]]].

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

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

?- [X|Y] = [mia, vincent, jules, yolanda].
X = mia,
Y = [vincent, jules, yolanda].

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

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



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

7. Problem Name: Member of List

Result:



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

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

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

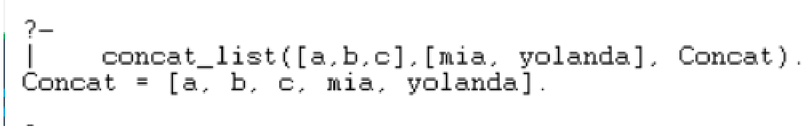
8. Problem Name: Concat List

Clause:

concat_lists(L1, L2, Concat) :- append(L1, L2, Concat)

Queries:

Result:



```
?-
|    concat_list([a,b,c],[mia, yolanda], Concat).
Concat = [a, b, c, mia, yolanda].
```

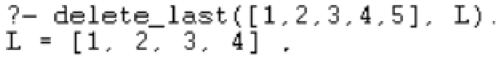
9. Problem name: Delete List

Clause:

delete_last([_], []). delete_last([Head|Tail], [Head|NewTail]) :- delete_last(Tail, NewTail).

Queries:

Result:



```
?- delete_last([1,2,3,4,5], L).
L = [1, 2, 3, 4] .
```

10.Problem Name: Deleting from an item

Clause:

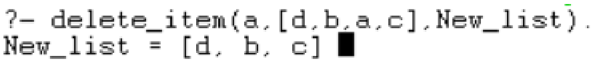
delete_item(X,[X|Tail],Tail).
delete_item(X,[Y|Tail],[Y|Tail1]):- delete_item(X,Tail,Tail1).

Queries :

?- delete_item(a,[d,b,a,c],New_list).

New_list = [d, b, c].

Results:



```
?- delete_item(a,[d,b,a,c],New_list).
New_list = [d, b, c]
```

11. Problem Name: Adding an item

Clause:

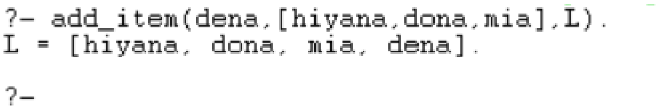
```
add_item(Item, List, NewList) :- append(List, [Item], NewList).
```

Queries:

?- add_item(dena,[hiyana,dona,mia],L).

L = [hiyana, dona, mia, dena].

Result:



```
?- add_item(dena,[hiyana,dona,mia],L).
L = [hiyana, dona, mia, dena].

?-
```

12. Problem Name: Implement a Prolog predicate ‘equal_length/2’ that takes two lists as input and succeeds if both lists have the same length .Give some example queries and their expected outputs.

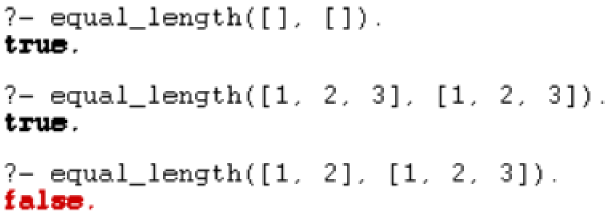
Clause:

```
equal_length([], []).
equal_length([H|T], [H1|T1]) :-
length([H|T]) = length([H1|T1]),
equal_length(T, T1).
```

Queries:

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

Result:



```
?- equal_length([], []).
true.

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

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

13. Prolog predicate ‘maximum/3’ that takes three integers as input and returns the maximum of the three.

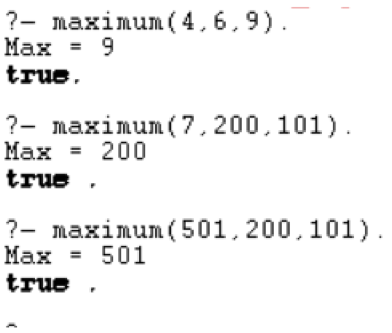
Clause:

```
maximum(A, B, C) :-
  A >= B,
  A >= C,
   write('Max = '),write(A).
maximum(A, B, C) :-
  B >= A,
  B >= C,
  write('Max = '),write(B).
maximum(A, B, C) :-
  C >= A,
  C >= B,
  write('Max = '),write(C).
```

Queries:

```
?- maximum(4,6,9).
?- maximum(7,200,101).
?- maximum(501,200,101).
```

Result:



```
?- maximum(4,6,9).
Max = 9
true.

?- maximum(7,200,101).
Max = 200
true .

?- maximum(501,200,101).
Max = 501
true .
```

14. Write a Prolog predicate to find the length of a list

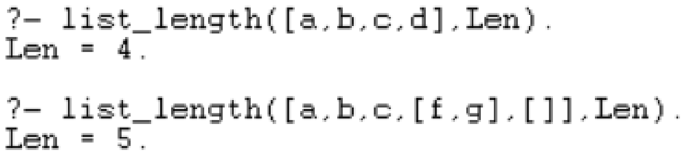
Clause:

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

Queries:

?- list_length([a,b,c,d],Len)
?- list_length([a,b,c,[f,g],[]],Len).

Result:



```
?- list_length([a,b,c,d],Len).
Len = 4.

?- list_length([a,b,c,[f,g],[]],Len).
Len = 5.
```