**German University in Cairo**
**Media Engineering and Technology**
**Dr. Nourhan Ehab**

# CSEN 403: Concepts of Programming Languages, Spring Term 2024
## Practice Assignment 2

### Exercise 2-1 Matching

Specify whether the query submitted to Prolog succeeds or fails; and if it succeeds, specify what is assigned to the variables by the matching; if it fails, explain why.

a) `| ?- X = fred.`

b) `| ?- jane = fred.`

c) `| ?- X = fred, X = Y.`

d) `| ?- X = happy(jim).`

e) `| ?- X is Y.`

f) `| ?- 2 + 1 = 3.`

g) `| ?- f(X,a) = f(a,X).`

h) `| ?- likes(jane, X) = likes(X, jim).`

i) `| ?- A = b(c).`

j) `| ?- a = b(c).`

### Exercise 2-2 Recursion

Consider a predicate `edge/2`. `edge(A,B)` is true if there is a connecting edge between the two nodes `A` and `B`. Implement rules for the predicate `path/2`. `path(A,B)` is true if there is a path connecting the two nodes `A` and `B`.

### Exercise 2-3 Recursion

In this exercise you are required to create a database about ingredients and recipes. It should contain two predicates: `ingredient(X,Y)` which holds if `X` is an ingredient of `Y`, and `partOfRecipe(X,Y)` which holds if `X` is used in the recipe of `Y`.

a) Add five facts to express the following information:

1. flour is an ingredient for dough.
2. egg is an ingredient for dough.
3. dough is an ingredient for a cake.
4. sugar is an ingredient for frosting.
5. chocolate is an ingredient for cake.

[0] The exercises are due to Prof. Dr. Slim Abdennadher and Dr. Nada Sharaf.

b) Add a rule called `partOfRecipe(X,Y)` that uses the facts in part a to tell us the constituents of `Y`. `X` is considered part of the recipe of `Y` if it's an ingredient or sub-ingredient of it.

Sample Queries:

```
?- partOfRecipe(flour,cake).
true.

?- partOfRecipe(X,cake).
X=dough;
X=chocolate;
X=flour;
X=egg;
false.

?- partOfRecipe(sugar,cake).
false
```

**Exercise 2-4**

In this exercise you are required to create a database about checmical reactions. It should contain two predicates: `reaction(X,Y,Z)` which holds if `Z` is the outcome of a chemical reaction between `X` and `Y`, and `checkPartOf(X,Y)` which holds if `X` contributed in a series of chemical reactions to form `Y`.

a) Add five facts to express the following information:

1. h2 is the result of a chemical reaction between h and h.
2. h2o is the result of a chemical reaction between h2 and o.
3. oh is the result of a chemical reaction between o and h.
4. h3 is the result of a chemical reactions between h2 and h.
5. naoh is the result of a chemical reactions between h and o.

b) Add a rule called `checkPartOf(X,Y)` that uses the facts in part a to tell us whether `X` was used in the chemical reactions leading to `Y` or not.

Sample Queries:

```
?- checkPartOf(h,h2o).
true ;
true ;
false.

11 ?- checkPartOf(X,h2o).
X = h2 ;
X = o ;
X = h ;
X = h ;
false.

14 ?- checkPartOf(h2,naoh).
false.
```

**Exercise 2-5**

In this exercise, you are required to create a database about the order you must follow when watching some specific movies. It should contain two predicates: `before(M1, M2)` which holds when movie `M1` has to be watched directly before movie `M2`, and `mustWatchBefore(M1, M2)` that succeeds when movie `M1` has to be watched before movie `M2`.

a) `"the_hunger_games"` movie must be watched directly before the movie `"the_hunger_games_catching_fire"`.

b) `"the_hunger_games_catching_fire"` movie must be watched directly before the movie `"the_hunger_games_mockingjay_part1"`.

c) `"the_hunger_games_mockingjay_part1"` movie must be watched directly before the movie `"the_hunger_games_mockingjay_part2"`.

Add the rule `mustWatchBefore(M1, M2)`.

Sample Queries:

```
?- mustWatchBefore(X, the_hunger_games_mockingjay_part2).
X = the_hunger_games_mockingjay_part1 ;
X = the_hunger_games ;
X = the_hunger_games_Catching_fire ;
false.

?- mustWatchBefore(the_hunger_games_mockingjay_part2, the_hunger_games).
false.
```

### **Exercise 2-6** Recursion

Consider a predicate `parent/2`. `parent(X,Y)` is true if person `X` is the parent of person `Y`. Implement rules for the predicate `descendant/2`. `descendant(X,Y)` is true if person `X` is the descendant of person `Y`.

### **Exercise 2-7** Recursion

In this exercise, you are required to create a database about building material. It should contain two predicates: `composedOf(X, Y)` which holds when material `X` is composed of material `Y`, and `partOf(X, Y)` which succeeds if `X` is part of `Y`.

a) Add eight relation facts to express the following information:

1. aggregate is composed of sand.
2. aggregate is composed of rock.
3. aggregate is composed of gravel.
4. concrete is composed of aggregate.
5. concrete is composed of cement.
6. concrete is composed of water.
7. wall is composed of concrete.
8. wall is composed of brick.

b) Add a rule called `partOf(X, Y)`.

Sample Queries:

```
?- partOf(sand, concrete).
true ;
false.

?- partOf(sand, wall).
true ;
false.

?- partOf(water, cement).
false.
```