1. A prolog program that define the following predicate

- Cities and countries / 1
  Cities in Africa
  Example Addis Ababa, Ethiopia.
- City Country/2
  A rule that displays list of Cities for a country name given by user.
- Add City with three arguments this should check if that City country pair exist and if the pair doesn't exist it should add it to the list.

Solution

```
% 1. Define the initial list of city-country pairs
cities_in_africa([
    city('Luanda', 'Angola'),
    city('Dakar', 'Senegal'),
    city('Abidjan', 'Ivory Coast'),
    city('Harare', 'Zimbabwe'),
    city('Tripoli', 'Libya'),
    city('Bamako', 'Mali'),
    city('Rabat', 'Morocco'),
    city('Windhoek', 'Namibia'),
    city('Kinshasa', 'Democratic Republic of Congo'),
    city('Maputo', 'Mozambique')
]).
% 2. cities_and_countries/1
% Displays all cities and their respective countries in the list
cities_and_countries(List) :-
    cities_in_africa(List),
    write('Cities in Africa:'), nl,
    print_cities(List).
% Helper predicate to print cities and countries
print_cities([]).
print_cities([city(City, Country) | Rest]) :-
    write(City), write(', '), write(Country), nl,
    print_cities(Rest).
% 3. city_country/2
% Displays the list of cities for a given country
city_country(Country, Cities) :-
    cities_in_africa(List),
    findall(City, member(city(City, Country), List), Cities),
    (   Cities = [] ->
        write('No cities found for '), write(Country), nl
    ;   write('Cities in '), write(Country), write(' are: '), write(Cities), nl
    ).
% 4. add_city/3
% Adds a new city-country pair if it doesn't already exist
add_city(City, Country, ResultList) :-
    cities_in_africa(List),
```

```
(   member(city(City, Country), List) ->
    write('The city-country pair already exists.'), nl,
    ResultList = List
;   append(List, [city(City, Country)], ResultList),
    write('City-country pair added successfully.'), nl
).
```

2. Write a prolog program that store countries with their federal language used in a list form using thepredicate lang_country. Define the right rule to for the following:
    - To display countries that uses some specific language such as English or Amharic as their official or federal language. It should count the number of countries that use the language in addition to the list.
    - To display the language used in a country taking the name of the country as input.

- Solution

```
% Facts: countries and their languages
languages_country(canada, english).
languages _country(brazil, portuguese).
languages _country(india, hindi).
languages _country(china, chinese).
languages _country(mexico, spanish).
languages _country(russia, russian).
languages _country(south_africa, afrikaans).
languages _country(saudi_arabia, arabic).
languages _country(nigeria, english).
languages _country(south_korea, korean).
% Rule 1: Find countries by language
countries_with_language(Language) :-
   findall(Country, languages _country(Country, Language), Countries),
   write('Countries that use '), write(Language), write(' as their official language:'), nl,
   write_list(Countries),
   length(Countries, Count),
   write('Number of countries: '), write(Count), nl.
% Rule 2: Find language by country
language_of_country(Country) :-
   languages _country(Country, Language),
   write('The official language of '), write(Country), write(' is '), write(Language), nl.
language_of_country(Country) :-
   \+ lang_country(Country, _),
   write('No data available for the country: '), write(Country), nl.
% Helper Print list
write_list([]).
write_list([H|T]) :-
   write('- '), write(H), nl,
   write_list(T).
```

3. Create three predicates in a knowledge base that add list element at the beginning, at the middle and at the end of a list.

Solution

```
% 1. Predicate to add an element at the beginning of a list
add_at_beginning(Element, List, [Element|List]).
% 2. Predicate to add an element in the middle of a list
add_at_middle(Element, List, Result) :-
   length(List, Length),
   MiddleIndex is Length // 2,        % Find the middle position
   split_at(List, MiddleIndex, Front, Back), % Split the list at the middle
   append(Front, [Element|Back], Result).
% 3. Predicate to add an element at the end of a list
add_at_end(Element, List, Result) :-
   append(List, [Element], Result).
% Helper predicate to split a list at a specific index
split_at(List, 0, [], List).
split_at([H|T], N, [H|Front], Back) :-
   N > 0,
   N1 is N - 1,
   split_at(T, N1, Front, Back).
```

4. List any five built in predicates used in prolog related to list. Use example to explain each.

1. length/2 :- This predicate determines the length of a list or can generate a list of a specified length.

Example ?- length([a, b, c, d], L). L = 4.

Example ?- length(List, 3). List = [_, _, _].

2. reverse/2: This predicate reverses the order of elements in a list.

Example: ?- reverse([1, 2, 3, 4], Result). Result = [4, 3, 2, 1].

3. nth0/3: This predicate retrieves an element from a list based on its 0-based index. **Example:** ?- nth0(2, [a, b, c, d], Element). Element = c.

4. nth1/3: This is similar to nth0/3, but it uses 1-based indexing

Example: ?- nth1(3, [a, b, c, d], Element). Element = c.

5. select/3: Removes an element from a list and returns the rest of the elements.

Example: ?- select(2, [1, 2, 3, 4], Result). Result = [1, 3, 4].