Name : M.Chanikya

Reg no : 192311193

Subject Code : CSA1793

Program 1: Counting Vowels

Code:

prolog

CollapseWrapCopy

```
vowel(X) :- member(X, [a, e, i, o, u]).

nr_vowel([], 0).

nr_vowel([X|T], N) :- vowel(X), nr_vowel(T, N1), N is N1 + 1, !.

nr_vowel([X|T], N) :- nr_vowel(T, N).
```

Input (Query):

text

CollapseWrapCopy

```
?- nr_vowel([a, r, e, d, i], X).
```

Output:

text

CollapseWrapCopy

```
X = 3
```

Explanation: The list [a, r, e, d, i] contains three vowels: a, e, and i. The predicate counts these and assigns X = 3.

Program 2: Sum of Integers from 1 to N

Code:

prolog

CollapseWrapCopy

```
sum_n(0, 0).

sum_n(N, Sum) :-
    N > 0,
    N1 is N - 1,
    sum_n(N1, Sum1),
    Sum is Sum1 + N.
```

Input (Query):

text

CollapseWrapCopy

?- sum_n(5, Sum).

Output:

text

CollapseWrapCopy

Sum = 15

Explanation: The sum of integers from 1 to 5 is 1 + 2 + 3 + 4 + 5 = 15.

Program 3: Database with Name and DOB

Code:

prolog

CollapseWrapCopy

```
person(john, date(1990, 5, 23)).
person(mary, date(1985, 12, 14)).
person(alex, date(2000, 7, 8)).
person(susan, date(1995, 10, 30)).
person(david, date(1988, 3, 15)).

find_dob(Name, DOB) :- person(Name, DOB).
find_name(DOB, Name) :- person(Name, DOB).
```

Input 1 (Query):

text

CollapseWrapCopy

?- find_dob(john, DOB).

Output 1:

text

CollapseWrapCopy

DOB = date(1990, 5, 23)

Input 2 (Query):

text

CollapseWrapCopy

?- find_name(date(2000, 7, 8), Name).

Output 2:

text

CollapseWrapCopy

Name = alex

Input 3 (Query):

text

CollapseWrapCopy

?- person(Name, DOB).

Output 3:

text

CollapseWrapCopy

Name = john, DOB = date(1990, 5, 23) ;

Name = mary, DOB = date(1985, 12, 14) ;

Name = alex, DOB = date(2000, 7, 8) ;

Name = susan, DOB = date(1995, 10, 30) ;

Name = david, DOB = date(1988, 3, 15).

Program 4: Medical Diagnosis

Code:

prolog

CollapseWrapCopy

```
disease(flu) :- has_symptom(fever), has_symptom(cough), has_symptom(sore_throat), has_symptom(runny_nose).

disease(common_cold) :- has_symptom(sneezing), has_symptom(runny_nose), has_symptom(sore_throat), has_symptom(mild_fever).

disease(covid19) :- has_symptom(fever), has_symptom(cough), has_symptom(shortness_of_breath), has_symptom(loss_of_taste_smell).

disease(malaria) :- has_symptom(fever), has_symptom(chills), has_symptom(sweating), has_symptom(headache).

disease(dengue) :- has_symptom(fever), has_symptom(severe_headache), has_symptom(joint_pain), has_symptom(skin_rash).


:- dynamic has_symptom/1.


diagnose(Disease) :- disease(Disease), !.


ask_symptom(Symptom) :-
    format("Do you have ~w? (yes/no): ", [Symptom]),
```

```
    read(Response),
    (Response == yes -> assertz(has_symptom(Symptom)) ; fail).

start_diagnosis :-
    retractall(has_symptom(_)),
    write("Medical Diagnosis System"), nl,
    write("Answer 'yes' or 'no' to the following symptoms."), nl,
    ask_symptom(fever),
    ask_symptom(cough),
    ask_symptom(sore_throat),
    ask_symptom(runny_nose),
    ask_symptom(sneezing),
    ask_symptom(mild_fever),
    ask_symptom(shortness_of_breath),
    ask_symptom(loss_of_taste_smell),
    ask_symptom(chills),
    ask_symptom(sweating),
    ask_symptom(headache),
    ask_symptom(severe_headache),
    ask_symptom(joint_pain),
    ask_symptom(skin_rash),
    (diagnose(Disease) -> format("You may have ~w. Please consult a doctor.", [Disease]), nl ;
     write("No matching disease found. Consult a doctor for further evaluation."), nl).
```

Input (Query):

text

CollapseWrapCopy

```
?- start_diagnosis.
```

Interactive Input Example:

text

CollapseWrapCopy

```
Do you have fever? (yes/no): yes
Do you have cough? (yes/no): yes
Do you have sore_throat? (yes/no): yes
Do you have runny_nose? (yes/no): yes
```

Do you have sneezing? (yes/no): no

Do you have mild_fever? (yes/no): no

Do you have shortness_of_breath? (yes/no): no

Do you have loss_of_taste_smell? (yes/no): no

Do you have chills? (yes/no): no

Do you have sweating? (yes/no): no

Do you have headache? (yes/no): no

Do you have severe_headache? (yes/no): no

Do you have joint_pain? (yes/no): no

Do you have skin_rash? (yes/no): no

Output:

text

CollapseWrapCopy

Medical Diagnosis System

Answer 'yes' or 'no' to the following symptoms.

You may have flu. Please consult a doctor.

Program 5: Sum of Integers (Alternative Version)

Code:

prolog

CollapseWrapCopy

```
sum(1, 1).
sum(N, S) :-
   N > 1,
   N1 is N - 1,
   sum(N1, S1),
   S is S1 + N.
```

Input (Query):

text

CollapseWrapCopy

?- sum(5, S).

Output:

text

CollapseWrapCopy

S = 15

Program 6: Database with Name and DOB (Alternative Version)

Code:

prolog

CollapseWrapCopy

```
person('John Doe', '1990-05-15').
person('Alice Smith', '1985-11-23').
person('Bob Johnson', '1993-07-08').
```

Input (Query):

text

CollapseWrapCopy

```
?- person(Name, DOB).
```

Output:

text

CollapseWrapCopy

```
Name = 'John Doe', DOB = '1990-05-15' ;
Name = 'Alice Smith', DOB = '1985-11-23' ;
Name = 'Bob Johnson', DOB = '1993-07-08'.
```

Program 7: Student-Teacher-Subject-Code

Code:

prolog

CollapseWrapCopy

```
teaches('Dr. Smith', 'Math', 'M101').
teaches('Prof. Johnson', 'Physics', 'P202').
studies('Alice', 'Math', 'M101').
studies('Bob', 'Physics', 'P202').
```

Input (Query):

text

CollapseWrapCopy

```
?- teaches(Teacher, Subject, Code).
```

Output:

text

CollapseWrapCopy

Teacher = 'Dr. Smith', Subject = 'Math', Code = 'M101' ;

Teacher = 'Prof. Johnson', Subject = 'Physics', Code = 'P202'.

Program 8: Planets Database

Code:

prolog

CollapseWrapCopy

planet(mercury, small, 0.39).

planet(venus, medium, 0.72).

planet(earth, medium, 1.00).

planet(mars, small, 1.52).

Input (Query):

text

CollapseWrapCopy

?- planet(Name, Size, Distance).

Output:

text

CollapseWrapCopy

Name = mercury, Size = small, Distance = 0.39 ;

Name = venus, Size = medium, Distance = 0.72 ;

Name = earth, Size = medium, Distance = 1.0 ;

Name = mars, Size = small, Distance = 1.52.

Program 9: Towers of Hanoi

Code:

prolog

CollapseWrapCopy

```
toh(1, Source, Dest, _) :-
    write('Move disk from '), write(Source), write(' to '), write(Dest), nl.
toh(N, Source, Dest, Aux) :-
    N > 1,
    N1 is N - 1,
    toh(N1, Source, Aux, Dest),
```

toh(1, Source, Dest, _),

toh(N1, Aux, Dest, Source).

Input (Query):

text

CollapseWrapCopy

?- toh(3, left, right, middle).

Output:

text

CollapseWrapCopy

Move disk from left to right

Move disk from left to middle

Move disk from right to middle

Move disk from left to right

Move disk from middle to left

Move disk from middle to right

Move disk from left to right

true

Program 10: Bird Can Fly

Code:

prolog

CollapseWrapCopy

bird(sparrow).

bird(eagle).

not_fly(ostrich).

can_fly(Bird) :- bird(Bird), \+ not_fly(Bird).

Input 1 (Query):

text

CollapseWrapCopy

?- can_fly(sparrow).

Output 1:

text

CollapseWrapCopy

true

Input 2 (Query):

text

CollapseWrapCopy

?- can_fly(ostrich).

Output 2:

text

CollapseWrapCopy

false

Program 11: Family Tree

Code:

prolog

CollapseWrapCopy

parent(john, alice).

parent(alice, bob).

grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

Input (Query):

text

CollapseWrapCopy

?- grandparent(john, bob).

Output:

text

CollapseWrapCopy

true

Program 12: Dieting System

Code:

prolog

CollapseWrapCopy

recommend_diet(diabetes, 'Low sugar, high fiber').

recommend_diet(hypertension, 'Low salt, high potassium').

Input (Query):

text

CollapseWrapCopy

?- recommend_diet(diabetes, Diet).

Output:

text

CollapseWrapCopy

Diet = 'Low sugar, high fiber'

Program 13: Monkey-Banana Problem

Code:

prolog

CollapseWrapCopy

monkey_at(left).

banana_at(right).

move(left, right) :- write('Monkey moves to right').

Input (Query):

text

CollapseWrapCopy

?- move(left, right).

Output:

text

CollapseWrapCopy

Monkey moves to right

true

Program 14: Fruit and Color using Backtracking

Code:

prolog

CollapseWrapCopy

fruit(apple, red).

fruit(banana, yellow).

fruit(grape, purple).

find_color(Fruit, Color) :- fruit(Fruit, Color).

Input (Query):

text

CollapseWrapCopy

?- find_color(apple, Color).

Output:

text

CollapseWrapCopy

Color = red

Program 15: Best First Search Implementation

Code:

prolog

CollapseWrapCopy

```
edge(a, b, 4).
edge(a, c, 2).
edge(b, d, 5).
edge(b, e, 12).
edge(c, f, 8).
edge(d, g, 3).
edge(e, h, 7).
edge(f, i, 6).

heuristic(a, 7).
heuristic(b, 6).
heuristic(c, 5).
heuristic(d, 4).
heuristic(e, 10).
heuristic(f, 3).
heuristic(g, 2).
heuristic(h, 8).
heuristic(i, 1).

best_first_search(Start, Goal) :-
    best_first_search_helper([(Start, 0)], Goal, []).

best_first_search_helper([(Goal, _)|_], Goal, _) :-
    write('Reached Goal: '), write(Goal), nl.
```

```
best_first_search_helper([(Node, Cost)|Rest], Goal, Visited) :-
  findall((Next, NewCost),
      (edge(Node, Next, _), heuristic(Next, H), NewCost is H, \+ member(Next, Visited)),
      Neighbors),
  append(Rest, Neighbors, NewQueue),
  sort(2, @=<, NewQueue, SortedQueue),
  write('Expanding: '), write(Node), nl,
  best_first_search_helper(SortedQueue, Goal, [Node|Visited]).
```

Input (Query):

text

CollapseWrapCopy

```
?- best_first_search(a, i).
```

Output:

text

CollapseWrapCopy

```
Expanding: a
Expanding: c
Expanding: f
Reached Goal: i
true
```

Program 16: Simple Medical Diagnosis

Code:

prolog

CollapseWrapCopy

```
diagnose(fever, 'Take rest and stay hydrated').
diagnose(cough, 'Drink warm fluids and rest').
diagnose(flu, 'Consult a doctor and take medications').
```

Input (Query):

text

CollapseWrapCopy

```
?- diagnose(flu, Advice).
```

Output:

text

CollapseWrapCopy

Advice = 'Consult a doctor and take medications'

Program 17: Forward Chaining

Code:

prolog

CollapseWrapCopy

fact(sunny).

rule(umbrella_needed) :- fact(raining).

rule(sunglasses_needed) :- fact(sunny).

Input (Query):

text

CollapseWrapCopy

?- rule(sunglasses_needed).

Output:

text

CollapseWrapCopy

true

Program 18: Backward Chaining

Code:

prolog

CollapseWrapCopy

prove(Goal) :- fact(Goal).

prove(Goal) :- rule(Goal), prove(Condition).

fact(sunny).

rule(sunglasses_needed) :- fact(sunny).

Input (Query):

text

CollapseWrapCopy

?- prove(sunglasses_needed).

Output:

text

CollapseWrapCopy

true

Program 19: Web Blog (HTML, Not Prolog)

Code:

html

PreviewCollapseWrapCopy

```
<!DOCTYPE html>
<html>
<head>
  <title>My WordPress Blog</title>
</head>
<body>
  <h1>Welcome to My Blog</h1>
  <p>This is a demonstration of HTML elements in WordPress.</p>
  <a href="https://www.example.com">Click Here</a> to visit an external website.
</body>
</html>
```

Input: N/A (This is static HTML, not a query-based system.) **Output:** When opened in a browser, it displays a webpage with a title, heading, paragraph, and clickable link.