

Experiment No: 6

Title: Prolog Programming Set 1

Objective: To get acquainted with logical programming
Implement

1. Hello, World! program

Code:

main:-

 process,
 halt.

process:-

 write('Hello World').

:- main.

Output:

```
Hello World

...Program finished with exit code 0
Press ENTER to exit console.
```


2. Program to check if an element is a member of a list


Code:

member(X, [X|_]).

member(X, [_|T]) :- member(X, T).

Output:

 member(11, [7, 8, 11, 56]).
true

 member(5, [7, 8, 11, 56]).
false


3. Program to append two lists

Code:

append([], L, L).

append([H|T], L, [H|Result]) :- append(T, L, Result).

Output:

 append([a, b], [c, d, e], Result).
Result = [a, b, c, d, e]

4. Program to reverse a list

Code:

```
reverse([], []).
```

```
reverse([H|T], Reversed) :- reverse(T, Result), append(Result, [H], Reversed).
```

Output:

 `reverse([1, 2, 3], Reversed).`

Reversed = [3, 2, 1]


5. Program to find the length of a list

Code:

```
len([], 0).
```

```
len([_|T], N) :- len(T, N1), N is N1 + 1.
```

Output:

 `len([1, 2, 3, 4, 5], Length).`

Length = 5


6. Program to find the maximum of two numbers

Code:

```
max(X, Y, X) :- X >= Y.
```

```
max(X, Y, Y) :- X < Y.
```

Output:

 `max(10, 7, Max).`

Max = 10

7. Program to find the factorial of a number

Code:

```
factorial(0, 1).
```

```
factorial(N, Result) :-
```

```
    N > 0,
```

```
    N1 is N - 1,
```

```
    factorial(N1, Result1),
```

```
    Result is N * Result1.
```

Output:

 `factorial(7, Result)`

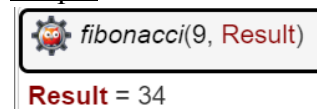
Result = 5040

8. Program to find the nth Fibonacci number

Code:

```
fibonacci(0, 0).  
fibonacci(1, 1).  
fibonacci(N, Result) :-  
    N > 1,  
    N1 is N - 1,  
    N2 is N - 2,  
    fibonacci(N1, Result1),  
    fibonacci(N2, Result2),  
    Result is Result1 + Result2.
```

Output:

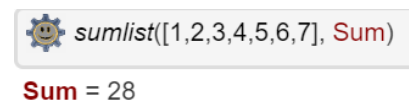


The image shows a Prolog execution window with a gear icon and the text `fibonacci(9, Result)`. Below the window, the text `Result = 34` is displayed.

9. Program to find the sum of a list of numbers

Code:

```
sumlist([], 0).  
sumlist([H|T], Sum) :-  
    sumlist(T, Sum1),  
    Sum is H + Sum1.
```



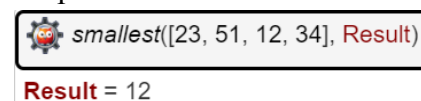
The image shows a Prolog execution window with a gear icon and the text `sumlist([1,2,3,4,5,6,7], Sum)`. Below the window, the text `Sum = 28` is displayed.

10. Program to find the smallest element in a list.

Code:

```
smallest([X], X).  
smallest([H|T], X) :-  
    smallest(T, X1),  
    (H < X1 -> X = H ; X = X1).
```

Output:



The image shows a Prolog execution window with a gear icon and the text `smallest([23, 51, 12, 34], Result)`. Below the window, the text `Result = 12` is displayed.