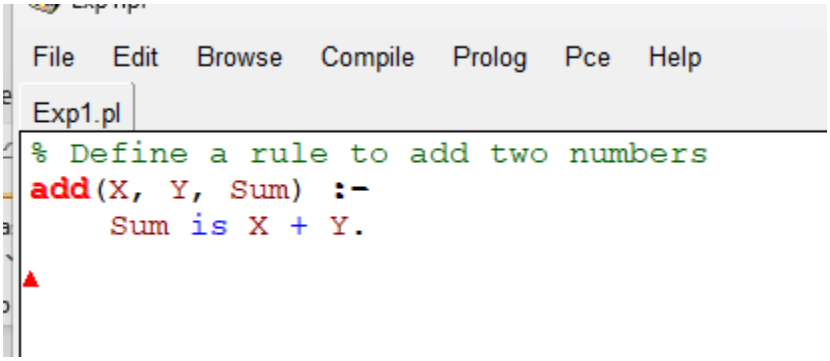
---

1. Write a Prolog in Prolog calculate addition of two no:




```
?-
% c:/users/utkarsh/documents/prolog/exp1 compiled 0.00 sec, -1 clauses
?- add(5, 7, Result).
Result = 12.
```

2. Write a Prolog in Prolog to find Maximum of two no:

```
% Define a rule to find the maximum of two numbers
max(X, Y, Max) :-
    (X >= Y -> Max = X; Max = Y).
```

```
% c:/users/utkarsh/documents/prolog/exp1 compiled 0.00 sec, 0 clauses
?- max(10, 20, Result).
Result = 20.
```

3. Write a Prolog in Prolog that take number N from the user and count from N to 10:

```
% Define a rule to count from N to 10
count_to_ten(N) :-
    N =< 10,
    write(N), nl,
    N1 is N + 1,
    count_to_ten(N1).


count_to_ten(N) :-
    N > 10.
```

```
?- count_to_ten(7).
7
8
9
10
true
```

4. Write a Prolog in Prolog that take number N from the user and count from N to 1.

```
% Define a rule to count from N down to 1
count_down(N) :-
    N >= 1,
    write(N), nl,
    N1 is N - 1,
    count_down(N1).


count_down(N) :-
    N < 1.
```

```
% c:/users/utkarsh/documents/prolog/exp1 compiled 0.00 sec, 0 clauses
?-
|    count_down(5).
5
4
3
2
1
true
```

5. Write a Prolog in Prolog that take number N from the user calculate factorial of no.

```
% Define a rule to calculate the factorial of N
factorial(0, 1).                    % Base case: factorial of 0 is 1
factorial(N, Result) :-
    N > 0,                          % Ensure N is positive
    N1 is N - 1,                    % Decrement N by 1
    factorial(N1, TempResult),      % Recursively calculate the facto
    Result is N * TempResult.       % Multiply N by the factorial of N
```

```
% c:/users/utkarsh/documents/prolog/exp1 compiled 0.00 sec, 0 clauses
|    factorial(5, Result).
Result = 120
```

6. Write a Prolog in Prolog that take number N from the user calculate square of no from N to 20 and display it:

```
% Define a rule to calculate and display the square of numbers from N to 20
square_to_twenty(N) :-
    N =< 20,
    Square is N * N,
    write(N),
    write(' squared is '),
    write(Square), nl,
    N1 is N + 1,
    square_to_twenty(N1).

square_to_twenty(N) :-
    N > 20.
```

```
?- square_to_twenty(17)
17 squared is 289
18 squared is 324
19 squared is 361
20 squared is 400
true
```