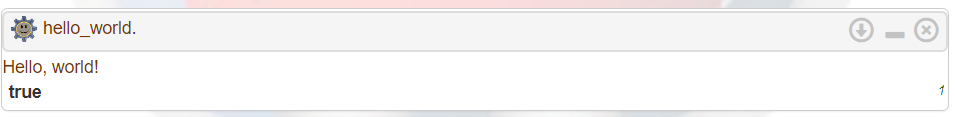
**Program 1: Hello World**

Code:

hello\_world :-

write('Hello, world!').

Output:



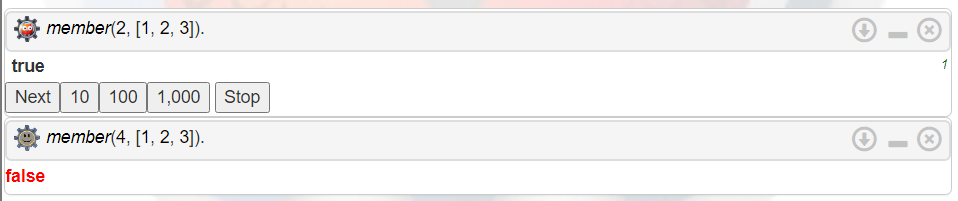
**Program 2: Check whether a number is a member of a list**

Code:

member(X, [X|\_]).

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

Output:



**Program 3: Program to append two lists**

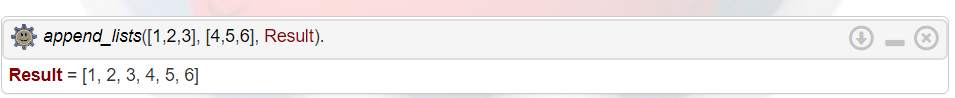
Code:

append\_lists([], L, L).

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

append\_lists(T, L, Result).

Output:



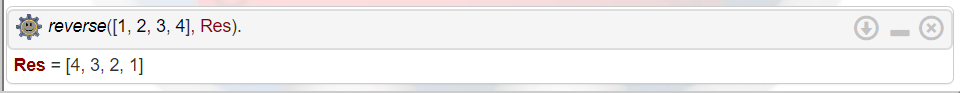
**Program 4: Reverse a list**

Code:

reverse([], []).

reverse([H|T], Res) :- reverse(T, TRes), append(TRes, [H], Res).

Output:

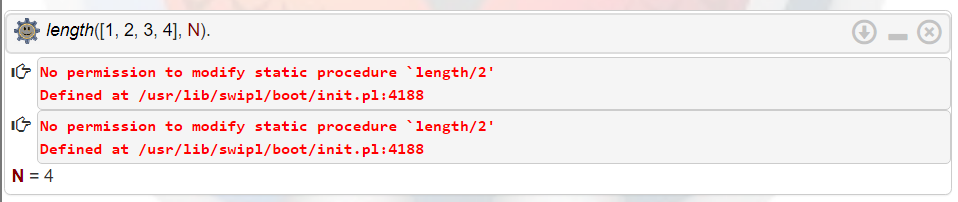


**Program 5: Find length of list**

Code:

length([], 0).

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

Output:  


**Program 6: Minimum and Maximum**

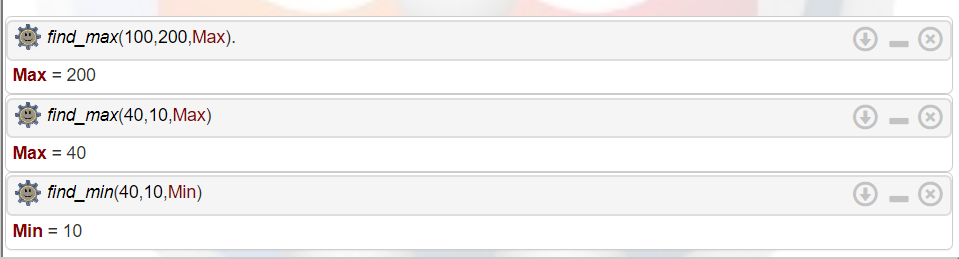
Code:

find\_max(X, Y, X) :- X >= Y, !.

find\_max(X, Y, Y) :- X < Y.

find\_min(X, Y, X) :- X =< Y, !.

find\_min(X, Y, Y) :- X > Y.

Output:  


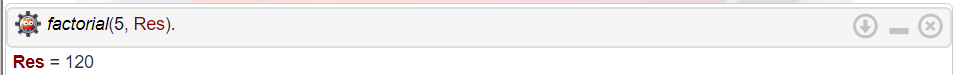
**Program 7: Factorial**

Code:

factorial(0, 1).

factorial(N, Res) :- N > 0, N1 is N - 1, factorial(N1, Res1), Res is N \* Res1.

Output:



**Program 8: Program to find nth number of fibonacci series**

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:



**Program 9: Program to find sum of elements of a list**

Code:

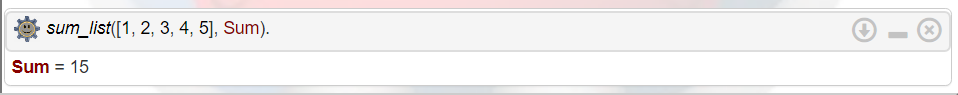
sum\_list([], 0).

sum\_list([H|T], Sum) :-

sum\_list(T, Rest),

Sum is H + Rest.

Output:



**Program 10: Program to find the smallest element of the list**

Code:

smallest([X], X).

smallest([H|T], X) :-

smallest(T, Y),

(H < Y -> X = H ; X = Y).

Output:

