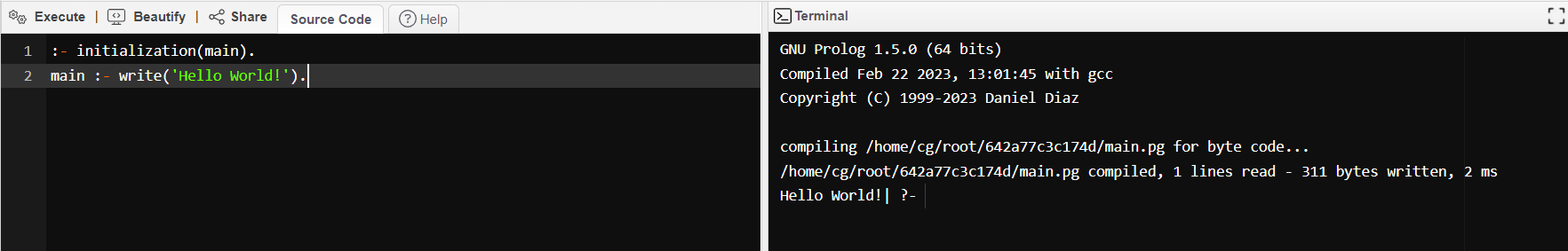
1. Hello world program

:- initialization(main).

main :- write('Hello World!').



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

is\_member(X, [X | \_]) :- !. % If the head of the list is X

is\_member(X, [\_ | Rest]) :- % else recur for the rest of the list

is\_member(X, Rest).

A screenshot of a computer screen

Description automatically generated with medium confidence

1. Program to append two lists

% If L1 is empty, resultant list will be equal to L2 (base case)

append\_list([], L2, L2).

append\_list([X | L1], L2, [X | L3]) :-

append\_list(L1, L2, L3).

Graphical user interface

Description automatically generated with low confidence

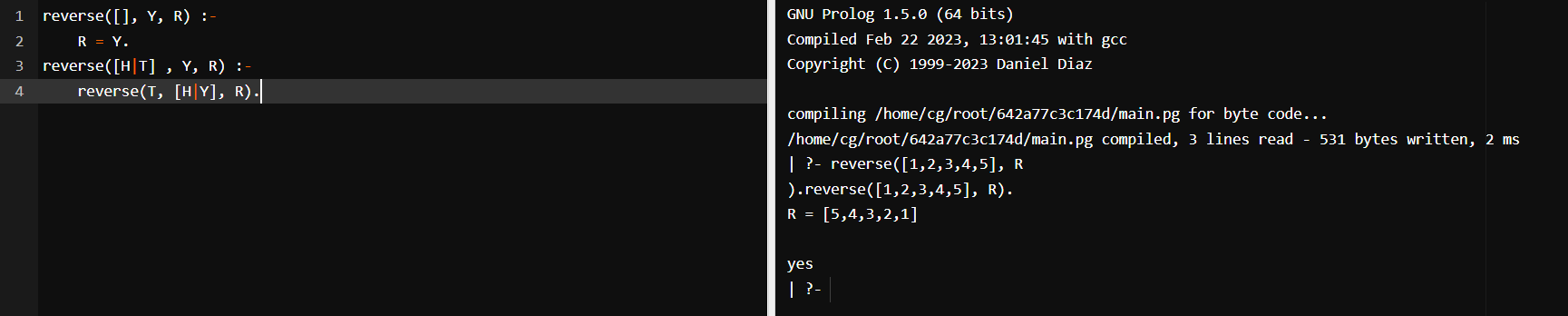
1. Program to reverse a list

reverse([], Y, R) :-

R = Y.

reverse([H|T] , Y, R) :-

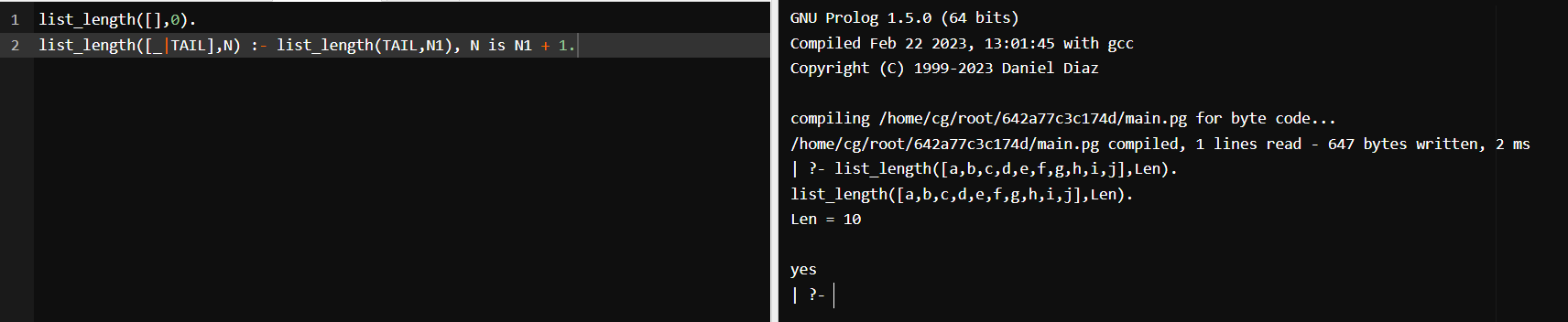
reverse(T, [H|Y], R).



1. Program to find the length of a list

list\_length([],0).

list\_length([\_|TAIL],N) :- list\_length(TAIL,N1), N is N1 + 1.



1. Program to find the maximum of two numbers

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.

Graphical user interface

Description automatically generated

1. Program to find the factorial of a number

fact(0,1).

fact(N,F):-

(

% The below is for +ve factorial.

N>0 ->

(

N1 is N-1,

fact(N1,F1),

F is N\*F1

)

;

% The below is for -ve factorial.

N<0 ->

(

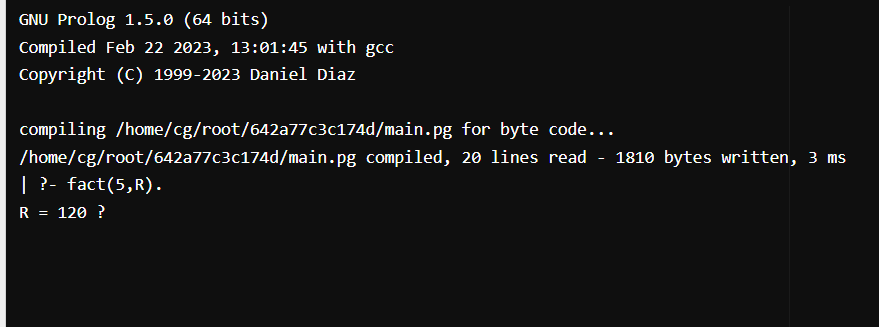
N1 is N+1,

fact(N1,F1),

F is N\*F1

)

).



1. Program to find the nth Fibonacci number

fib(0, 1) :- !.

fib(1, 1) :- !.

fib(N, F) :-

N > 1,

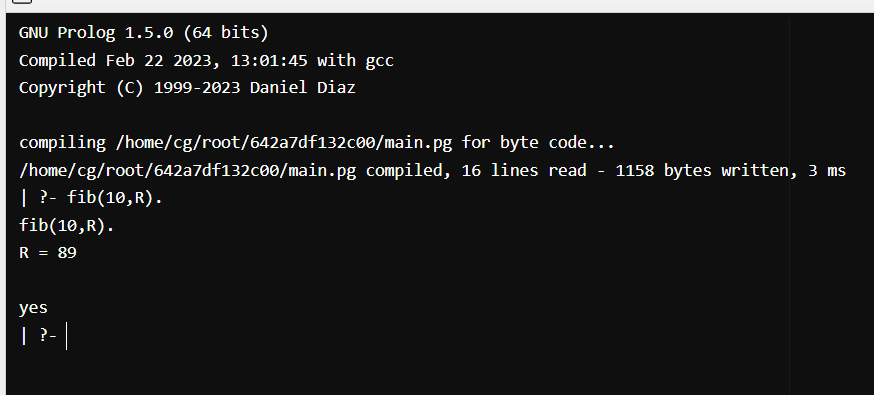
N1 is N-1,

N2 is N-2,

fib(N1, F1),

fib(N2, F2),

F is F1+F2.



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

sumlist([],0).

sumlist([H|T],R):-

sumlist(T,R1),

R is H+R1.

Text

Description automatically generated

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

list\_min([L|Ls], Min) :- list\_min(Ls, L, Min).

list\_min([], Min, Min).

list\_min([L|Ls], Min0, Min) :-

Min1 is min(L, Min0),

list\_min(Ls, Min1, Min).

