DECLARATIVE PROGRAMMING – 2020 HOMEWORK - I

1\_

1. Construct a valid argument in English the validity of which can be captured in first-order predicate logic but not in propositional logic

All doctors are medical professionals

Canan Karatay is a doctor

Canan Karatay is a medical professionals

b) Translate the argument into propositional logic (L1).

All doctors are medical professionals -> q

Canan Karatay is a doctor -> p

----------------------------------------------------------

Canan Karatay is a medical professionals -> r

c) Re-translate the translation in (b) into Prolog.

doctor(canankaratay,doctors).

alldoctors(doctors,medicalprofessionals).

medical(X, Z):- doctor(X, Y), alldoctors(Y, Z). QUERY : ?- medical(X,Z).

d) Translate the argument into first-order predicate logic (L3).

∀x[alldoctors(x) ->medical professionals(x)]

doctor(Canan Karatay)

----------------------------------------------------------

medical professionals(Canan Karatay)

e) Re-translate the translation in (d) into Prolog.

medical(X) :- alldoctor(X).

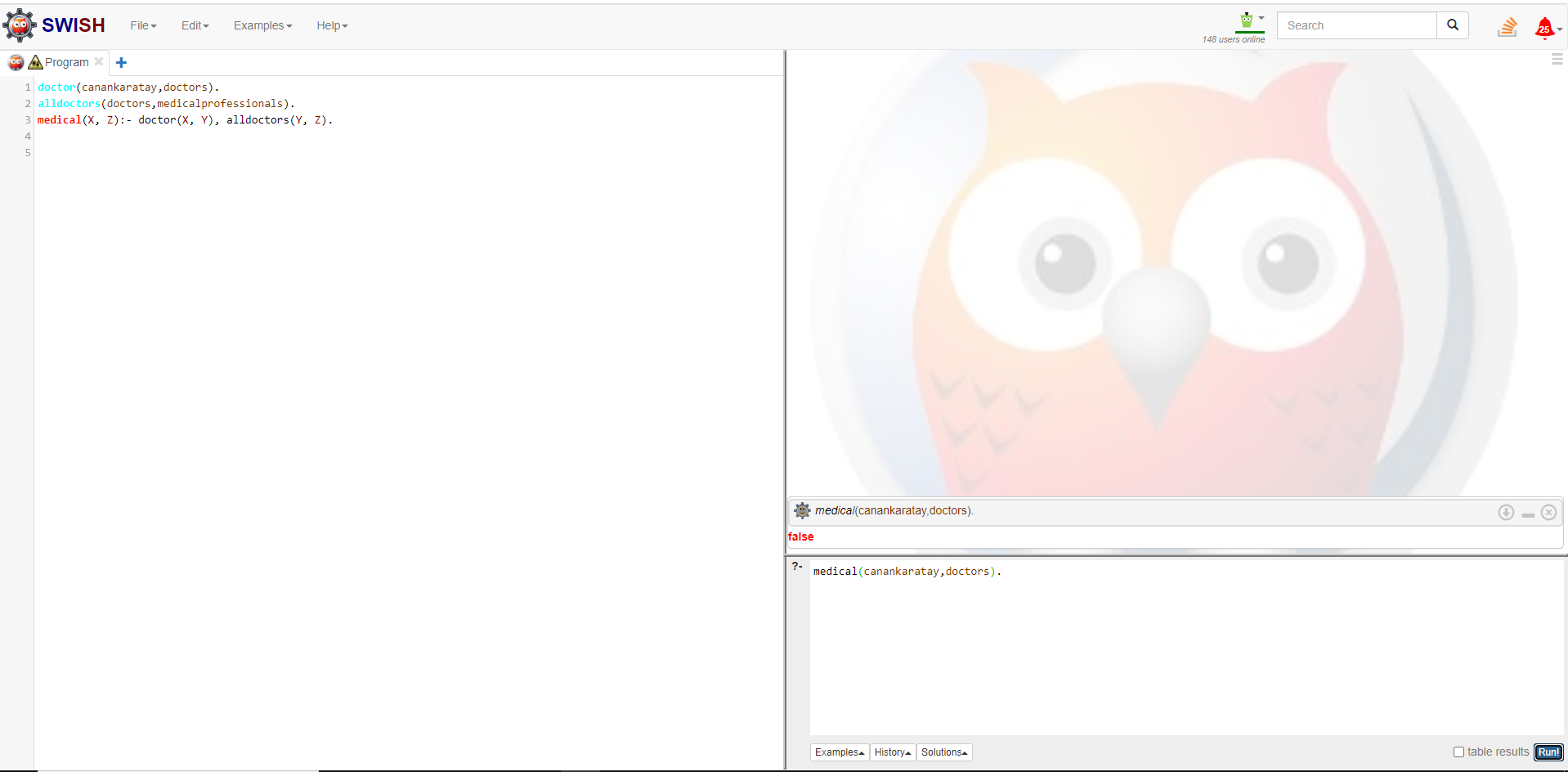
alldoctor(canankaratay).

f) Show that the argument loses its validity when translated into propositional logic with the appropriate Prolog query.

Query:

?- medical(canankaratay,doctors).

False

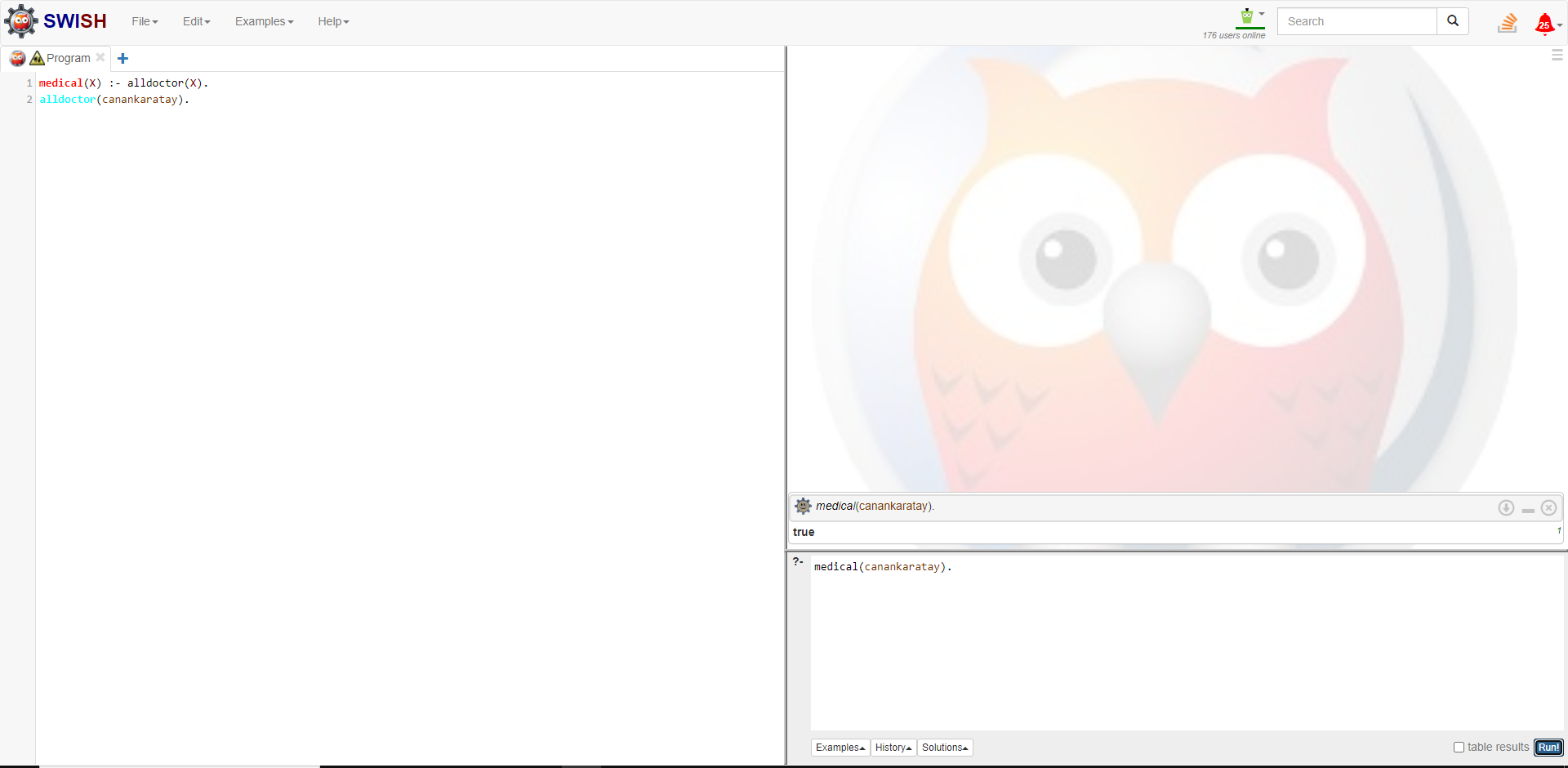


g) Show that the argument retains its validity when translated into first-order predicate logic with the appropriate Prolog query.

Query:

?- medical(canankaratay).

True



2)

a) Write a Prolog program to determine whether or not a given list of characters is a palindrome

conc([],L,L).

conc([X|Y],T,[X|Z]):-

conc(Y,T,Z).

reverse([],[]).

reverse([H|Y],ReversedList):-

reverse(Y,Reverse),

conc(Reverse,[H],ReversedList).

palindrome(L):-

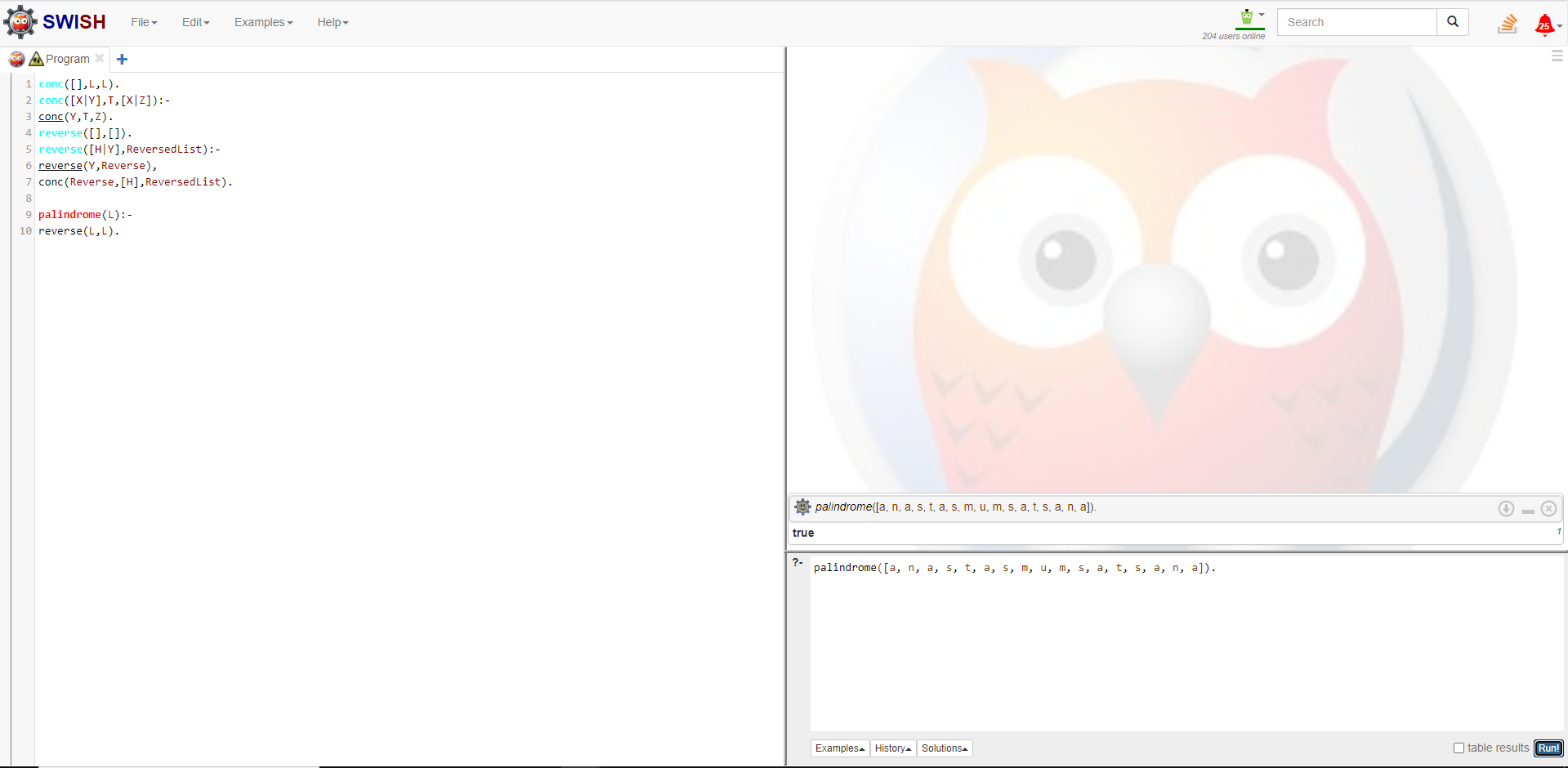
reverse(L,L).

1. Demonstrate with two queries that your program can distinguish between palindromes and non-palindromes.

Query:

?-palindrome([a, n, a, s, t, a, s, m, u, m, s, a, t, s, a, n, a]).

True



Query :

?-palindrome([k,a,r,p,u,z]).

False

