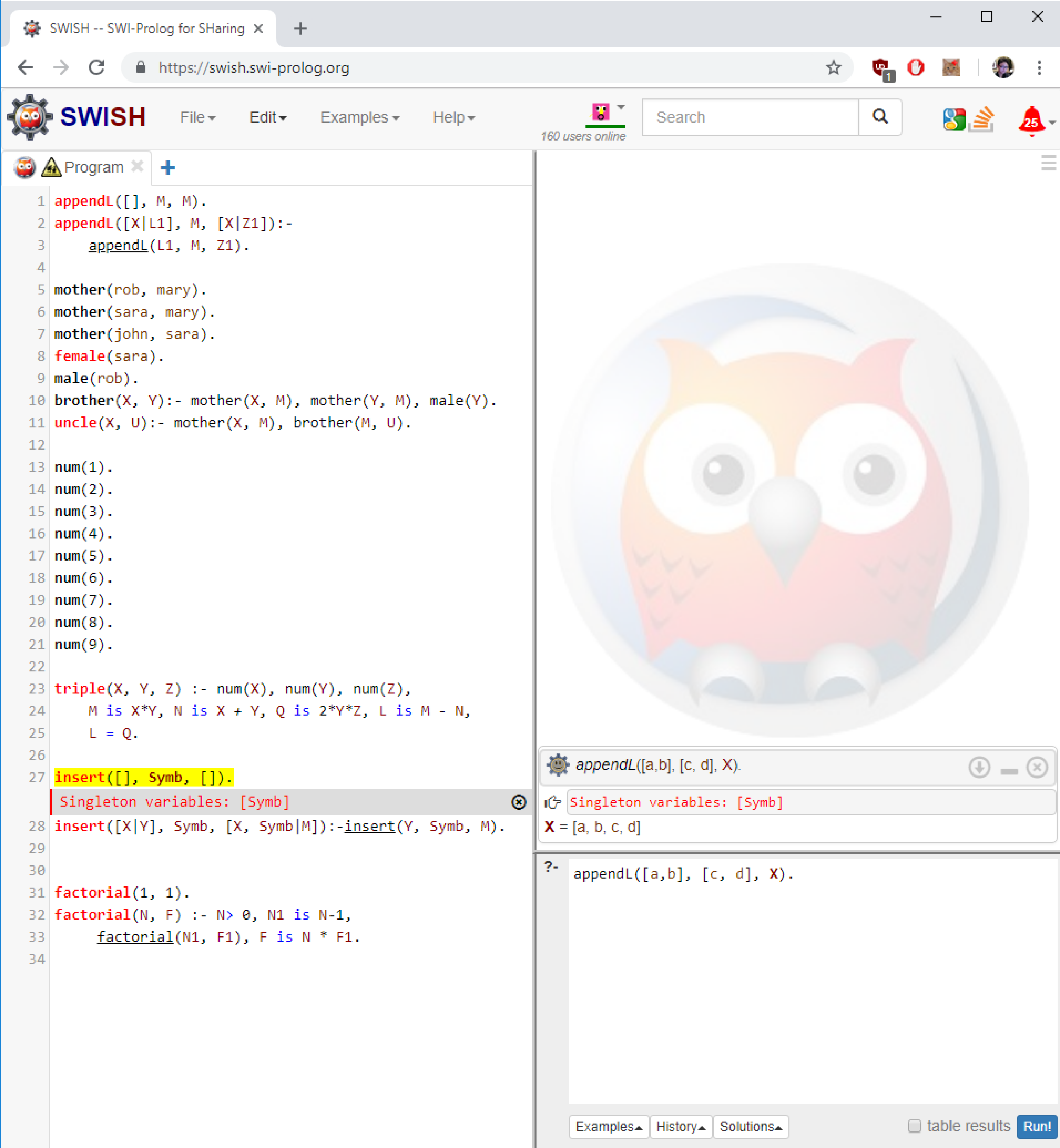
Beomjin Han

Professor Chidella

CSC135-2

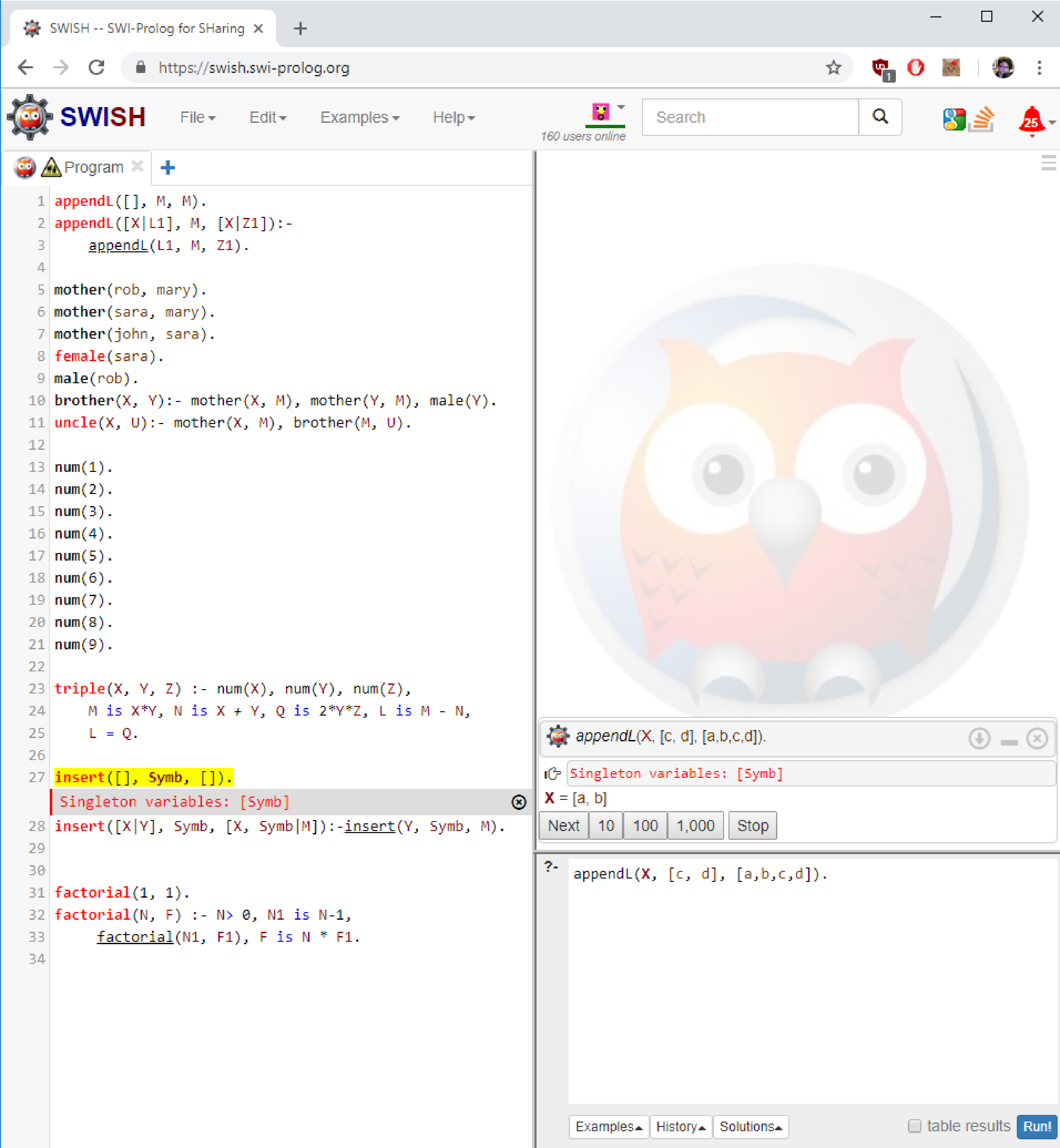
Assignment 3



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
appendL([a,b], [c, d], X).
Singleton variables: [Symb]
X = [a, b, c, d]
```

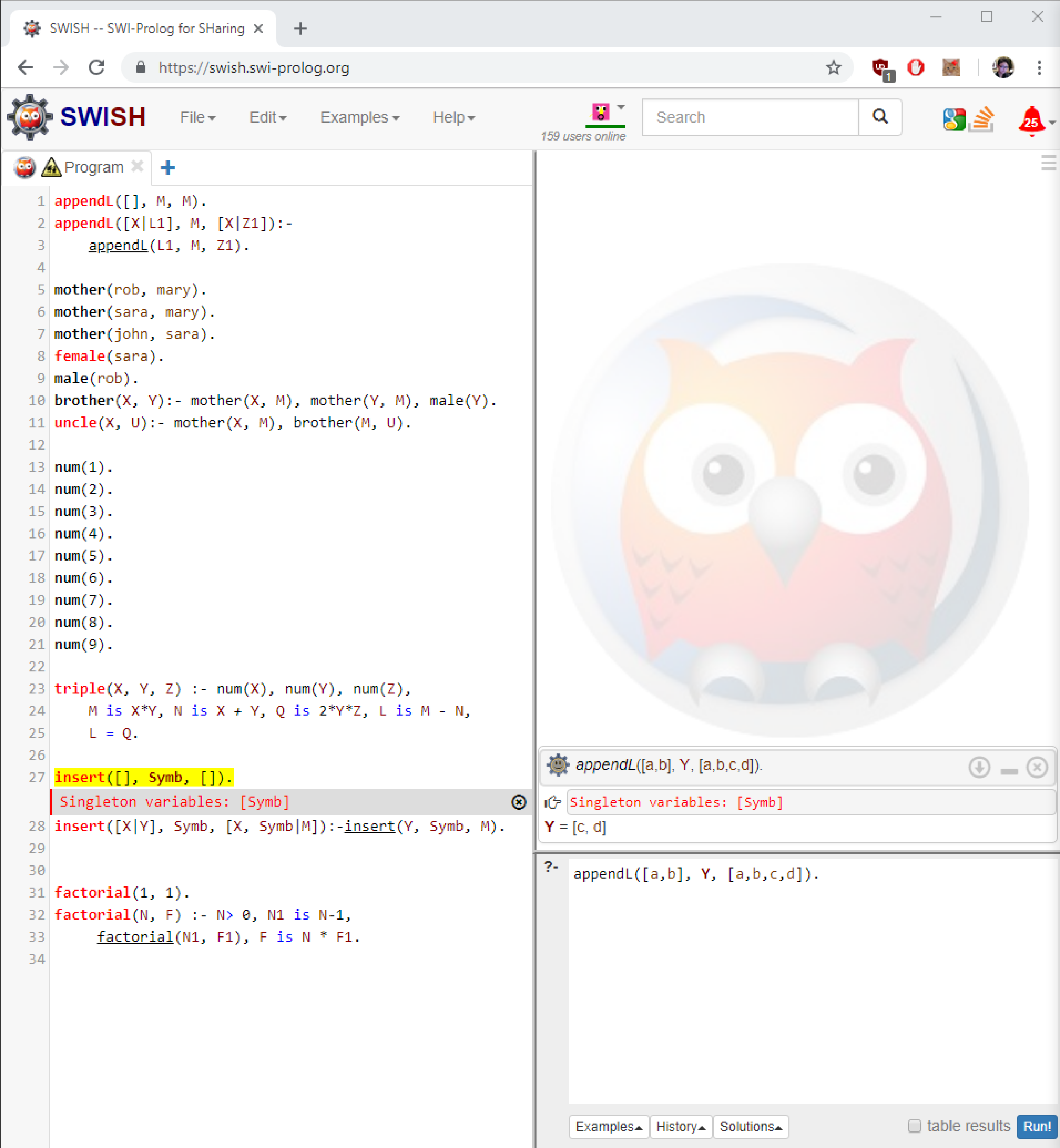
```
?- appendL([a,b], [c, d], X).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
appendL(X, [c, d], [a,b,c,d]).
Singleton variables: [Symb]
X = [a, b]
```

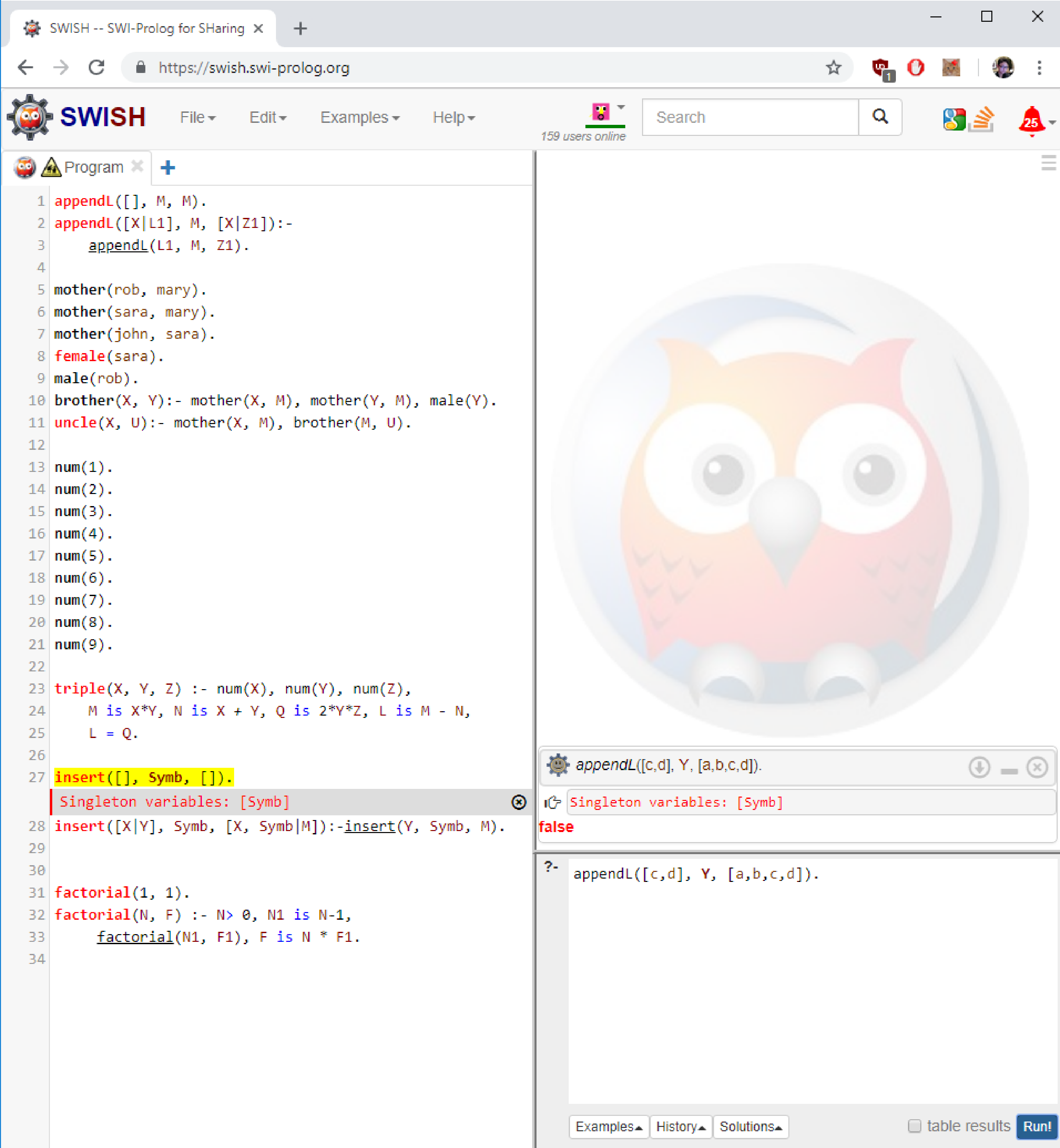
```
?- appendL(X, [c, d], [a,b,c,d]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
appendL([a,b], Y, [a,b,c,d]).
Singleton variables: [Symb]
Y = [c, d]
```

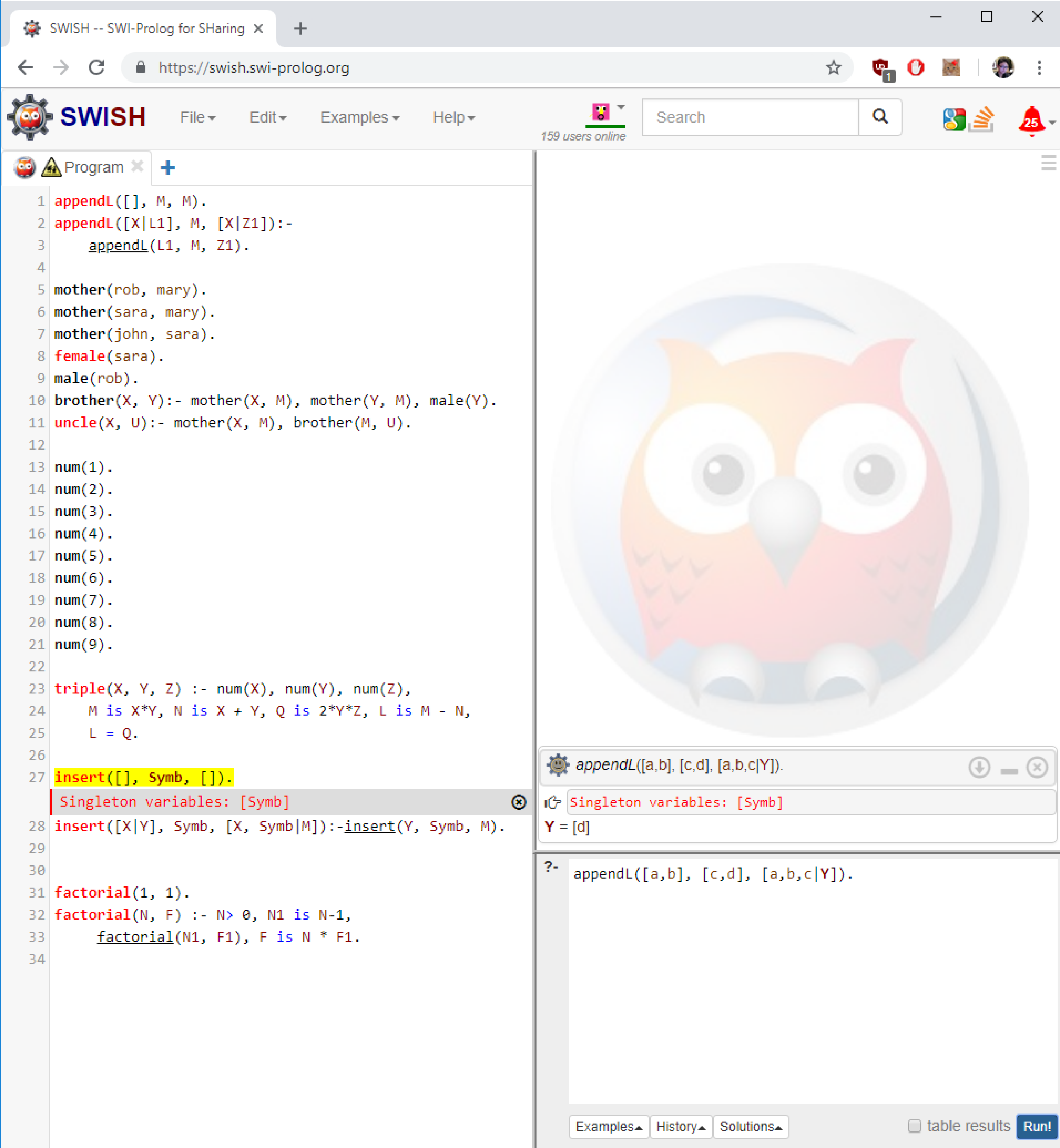
```
?- appendL([a,b], Y, [a,b,c,d]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
appendL([c,d], Y, [a,b,c,d]).
Singleton variables: [Symb]
false
```

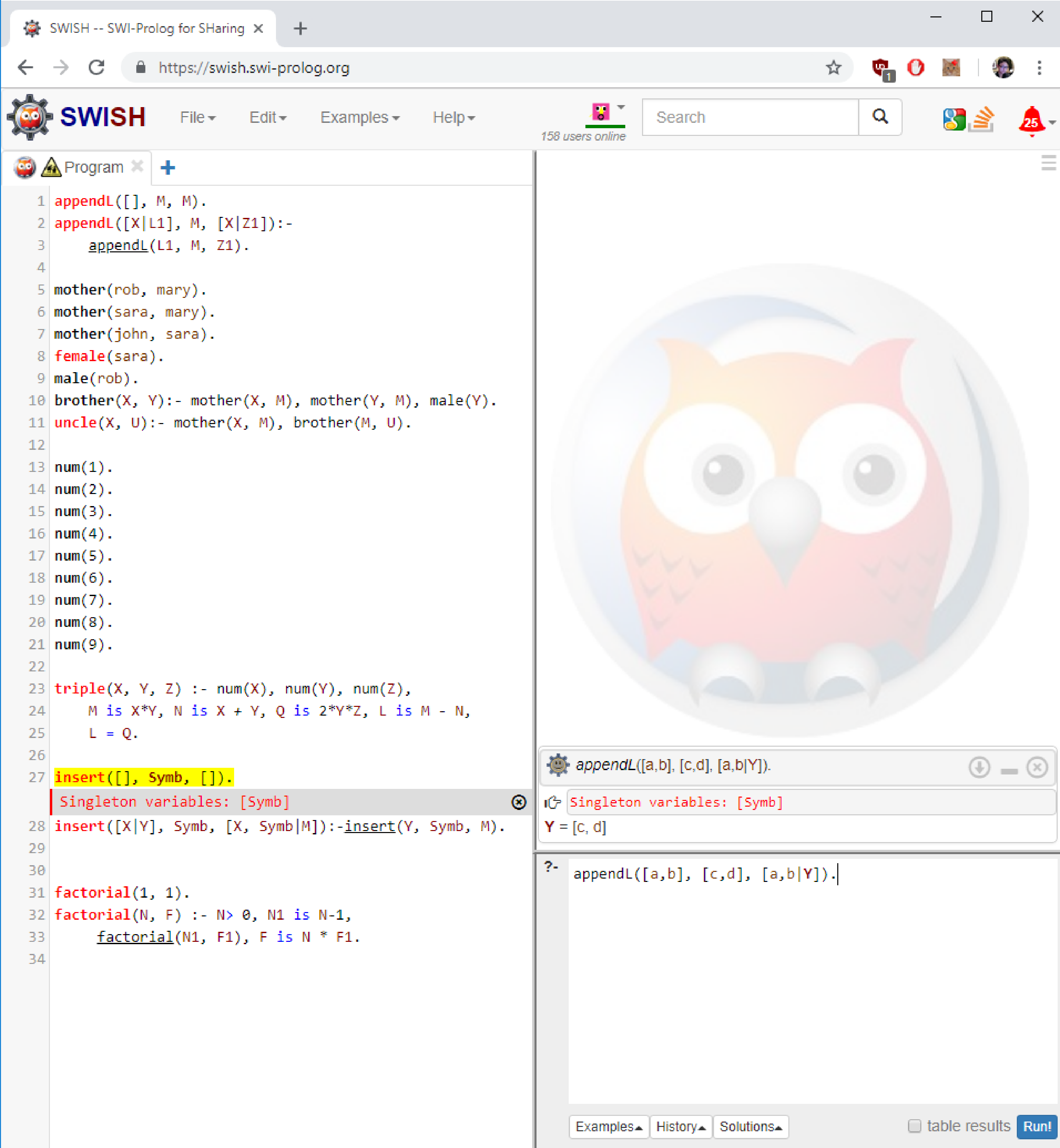
```
?- appendL([c,d], Y, [a,b,c,d]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

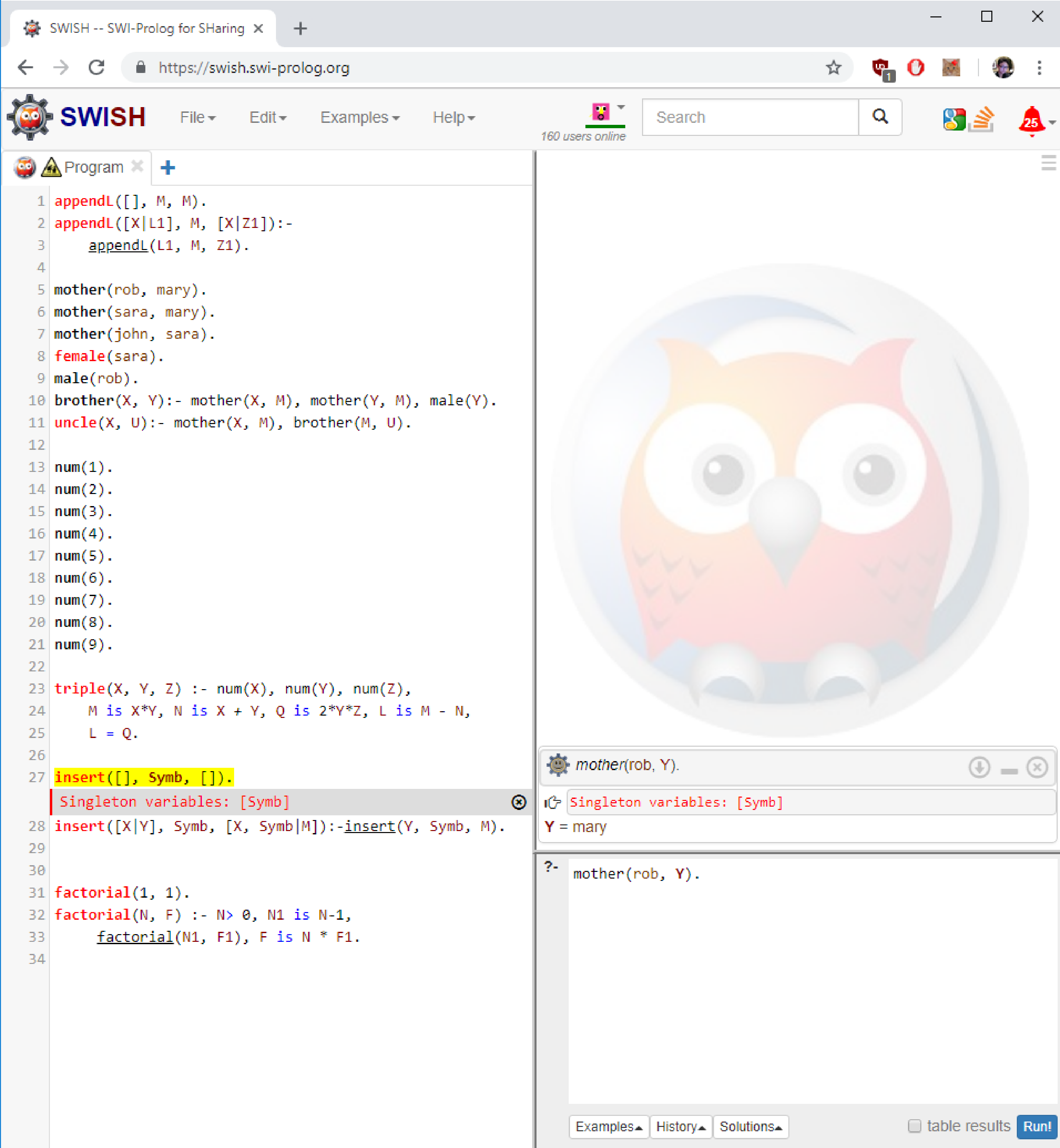
```
appendL([a,b], [c,d], [a,b,c|Y]).
Singleton variables: [Symb]
Y = [d]
```

```
?- appendL([a,b], [c,d], [a,b,c|Y]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

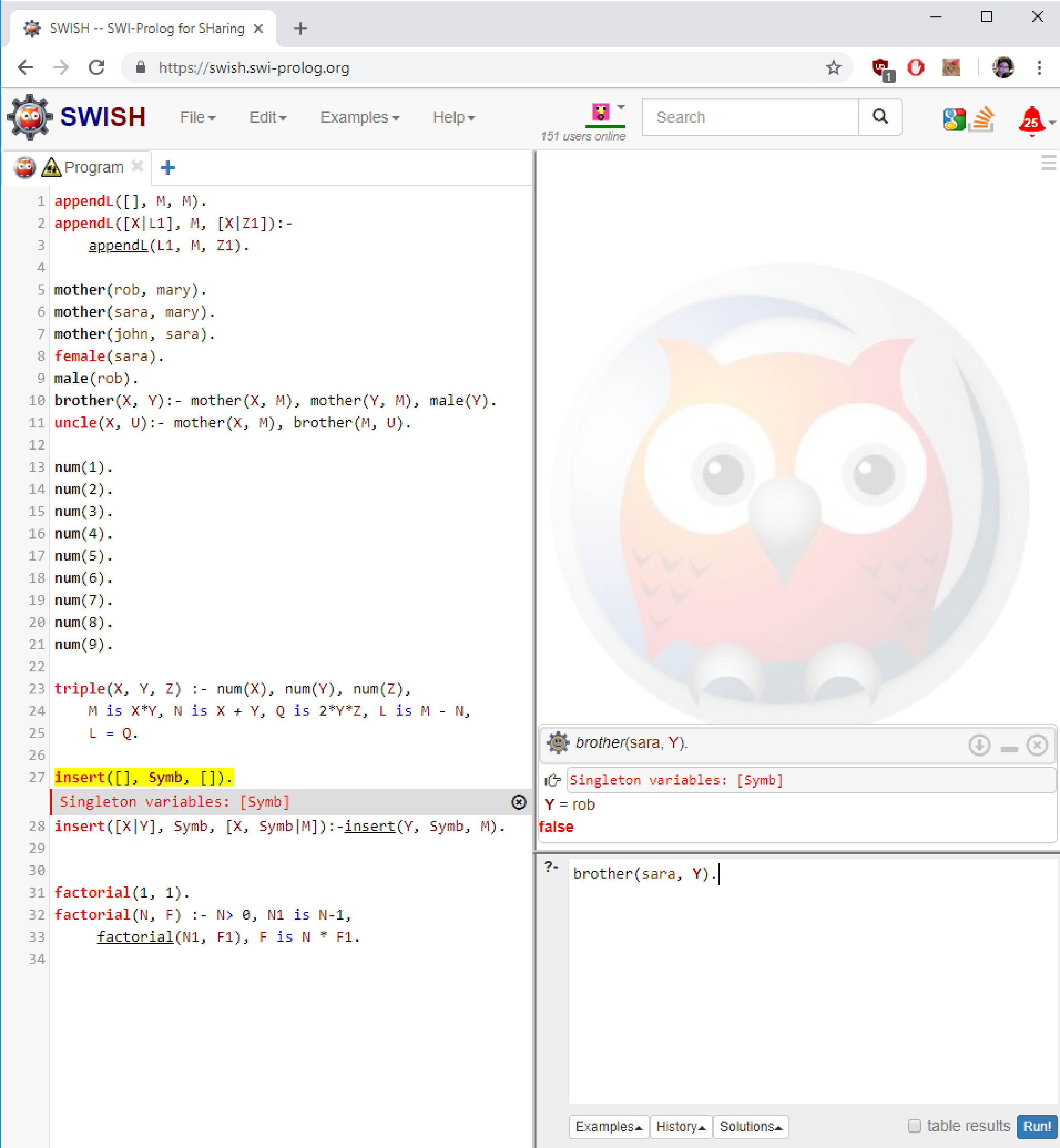
```
appendL([a,b], [c,d], [a,b|Y]).
Singleton variables: [Symb]
Y = [c, d]

?- appendL([a,b], [c,d], [a,b|Y]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
mother(rob, Y).
Singleton variables: [Symb]
Y = mary
```

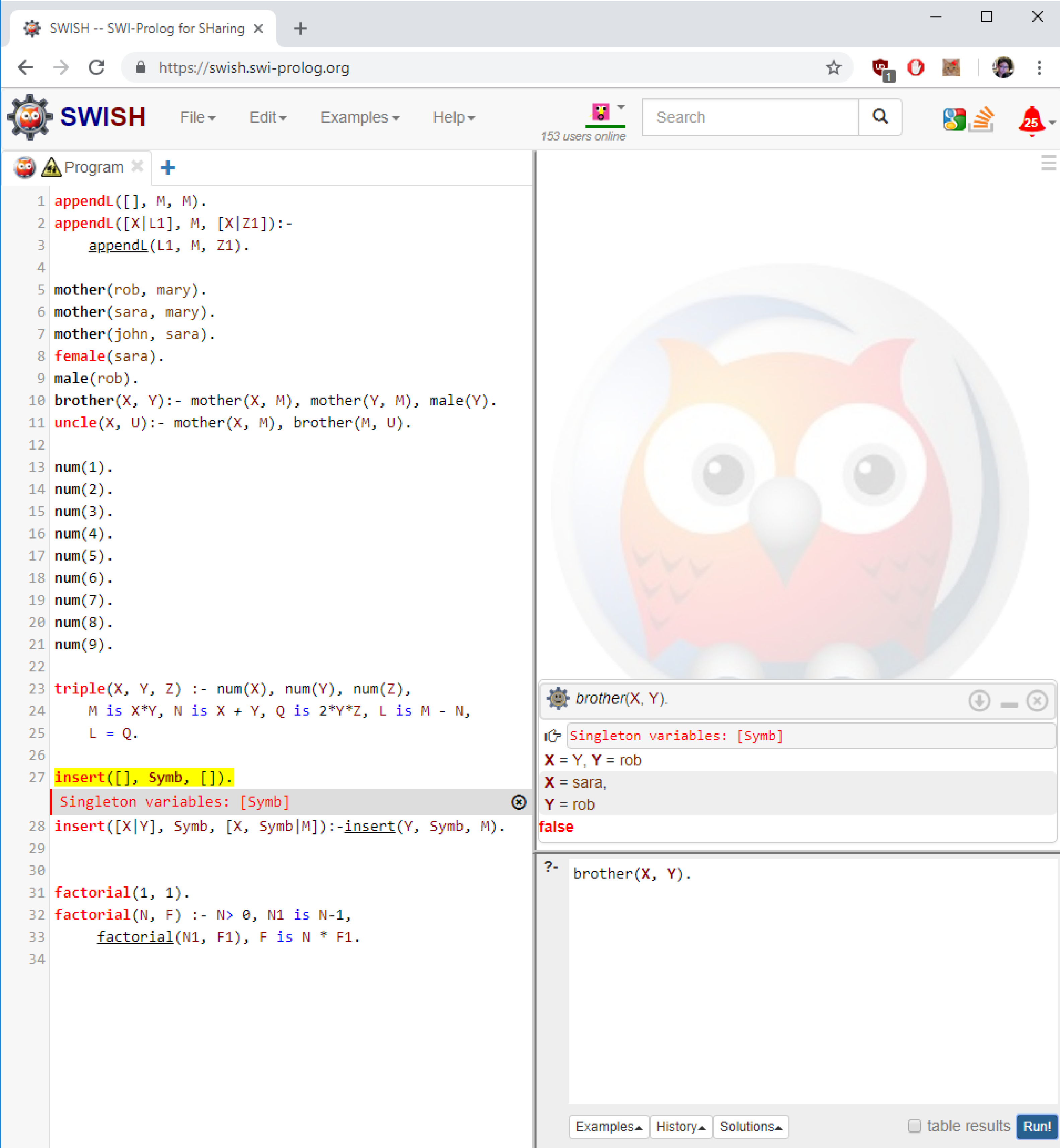
```
?- mother(rob, Y).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
brother(sara, Y).
Singleton variables: [Symb]
Y = rob
false
```

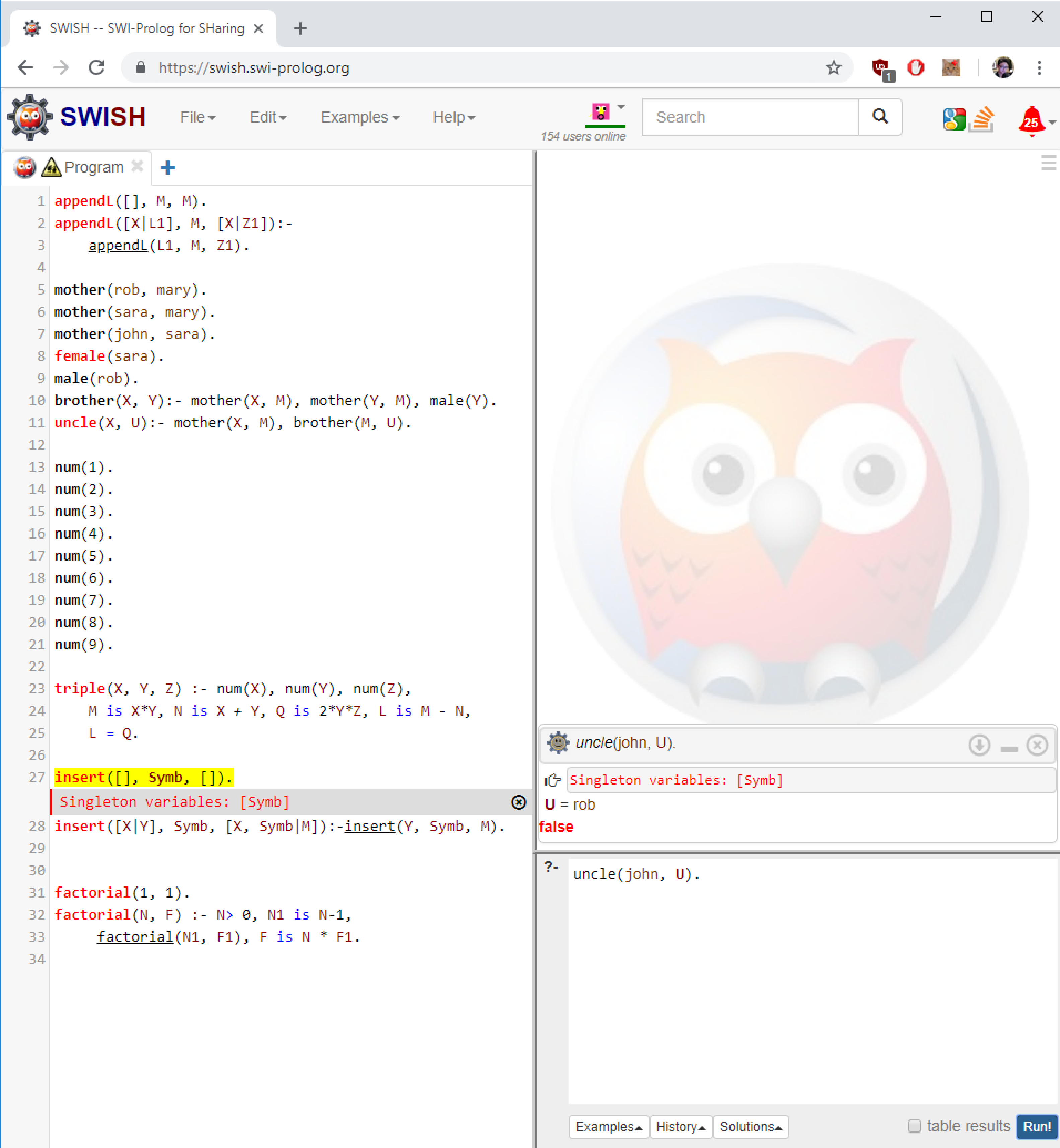
```
?- brother(sara, Y).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
brother(X, Y).
Singleton variables: [Symb]
X = Y, Y = rob
X = sara,
Y = rob
false
```

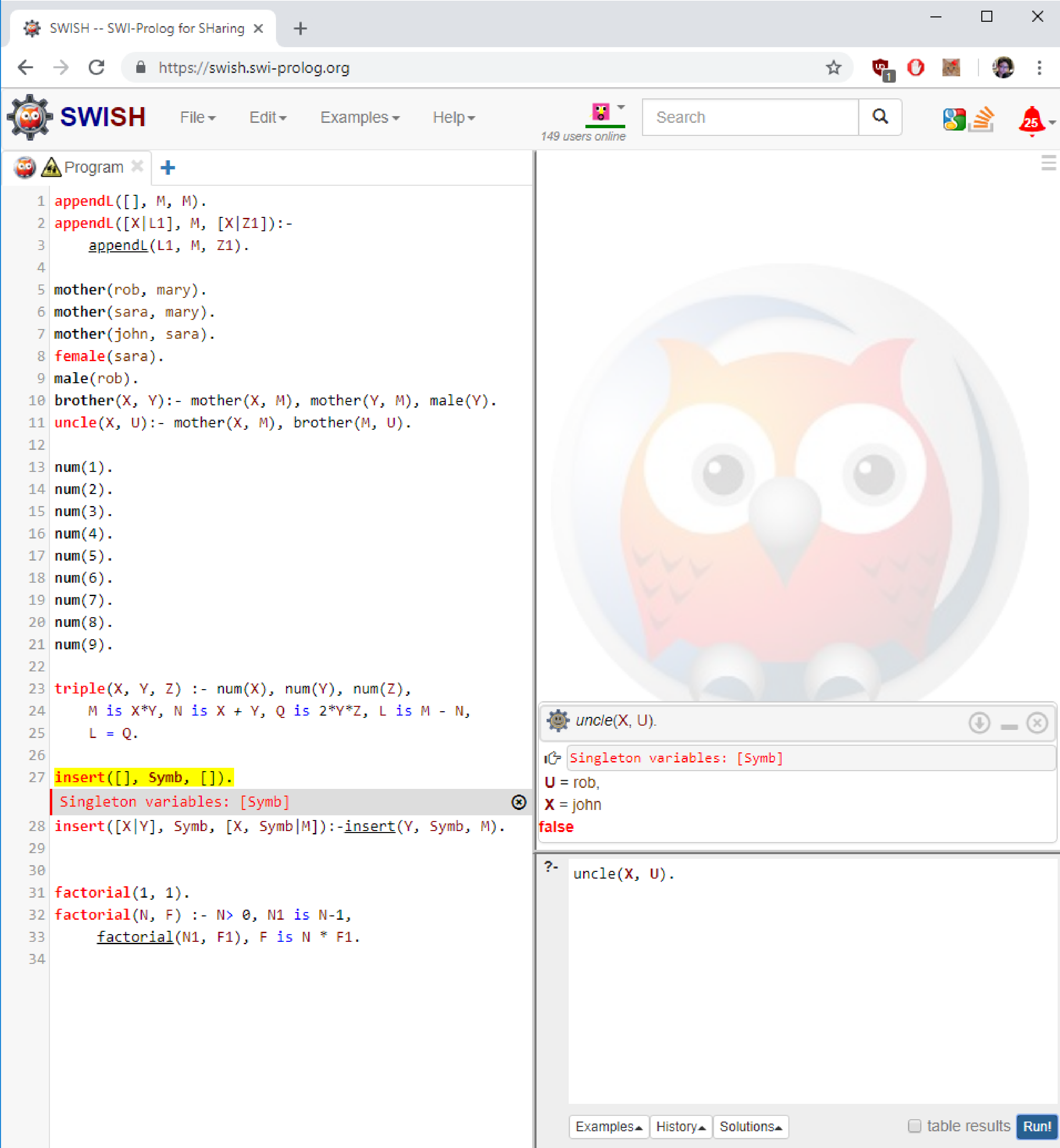
```
?- brother(X, Y).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
uncle(john, U).
Singleton variables: [Symb]
U = rob
false
```

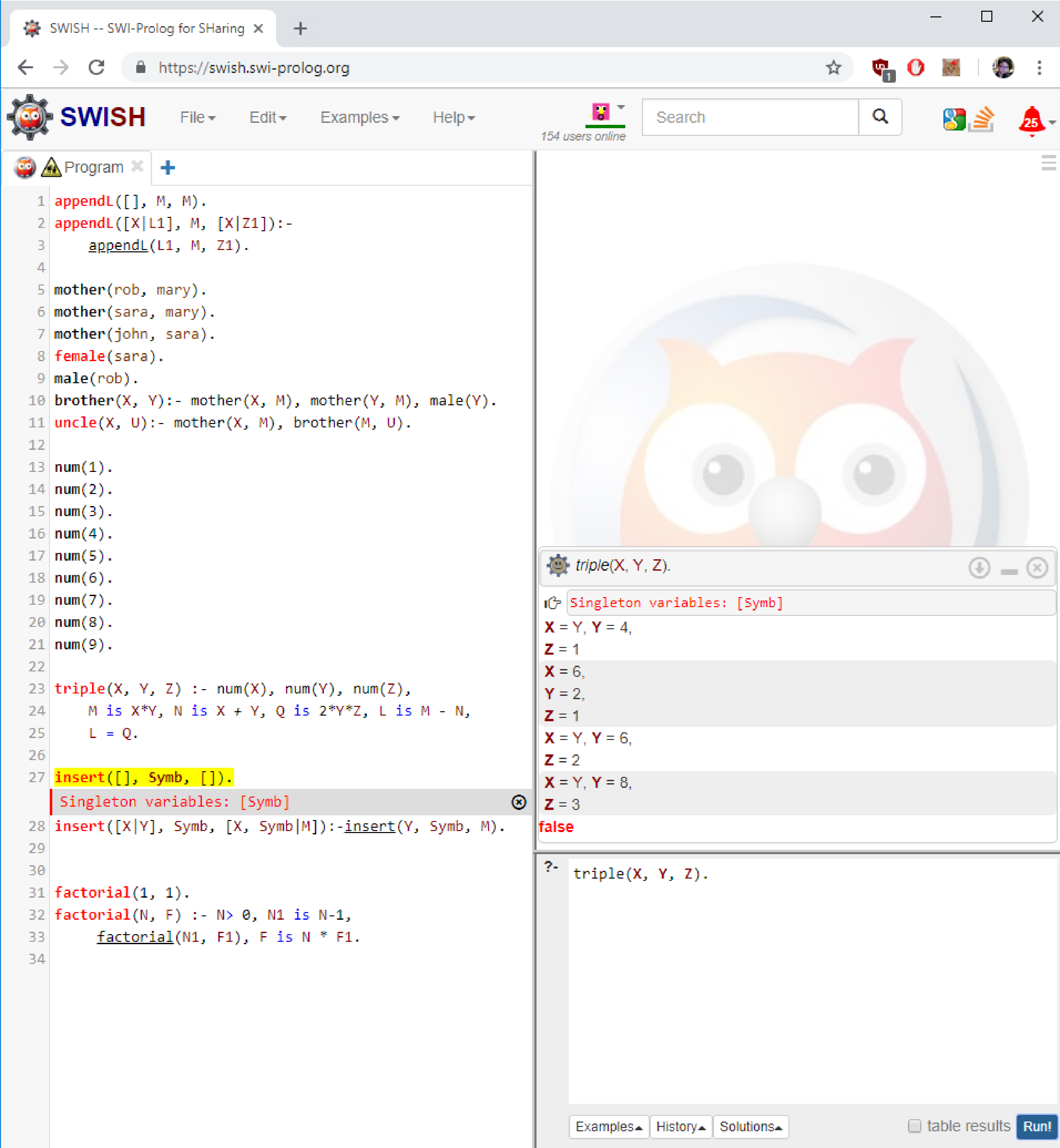
```
?- uncle(john, U).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
uncle(X, U).
Singleton variables: [Symb]
U = rob,
X = john
false
```

```
?- uncle(X, U).
```

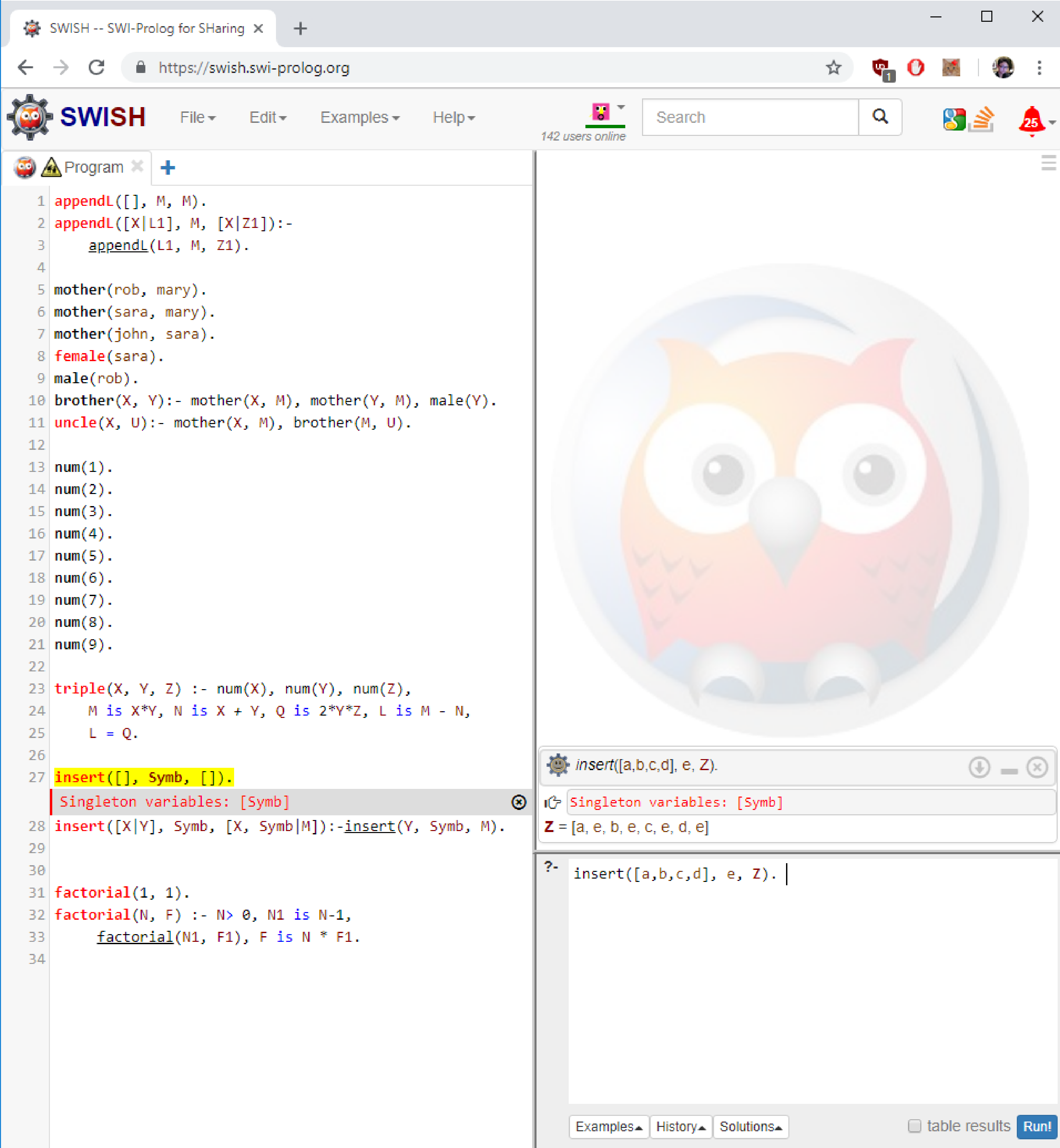


```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
triple(X, Y, Z).
Singleton variables: [Symb]
X = Y, Y = 4,
Z = 1
X = 6,
Y = 2,
Z = 1
X = Y, Y = 6,
Z = 2
X = Y, Y = 8,
Z = 3
false
```

```
?- triple(X, Y, Z).
```

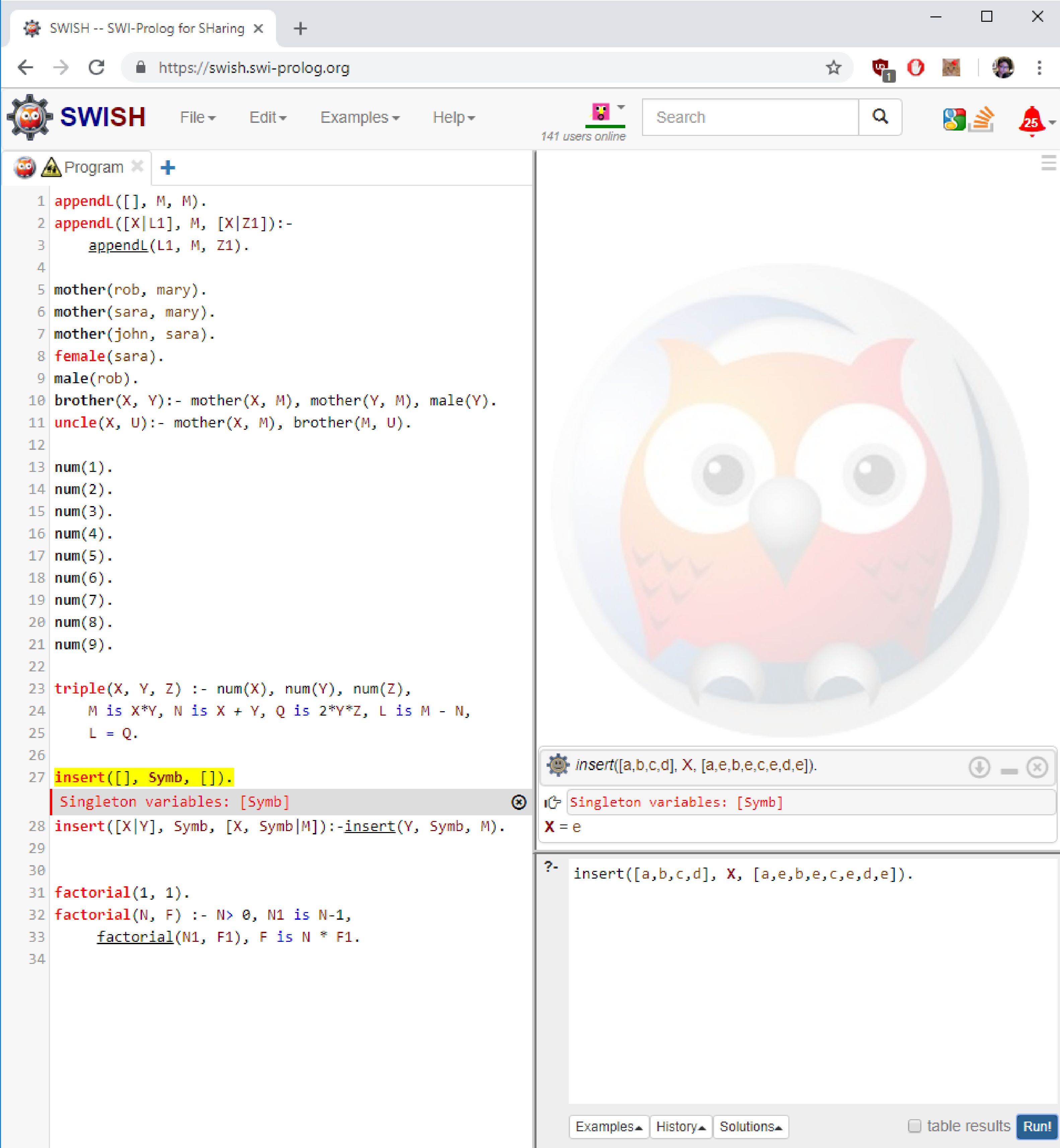
]



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
insert([a,b,c,d], e, Z).
Singleton variables: [Symb]
Z = [a, e, b, e, c, e, d, e]
```

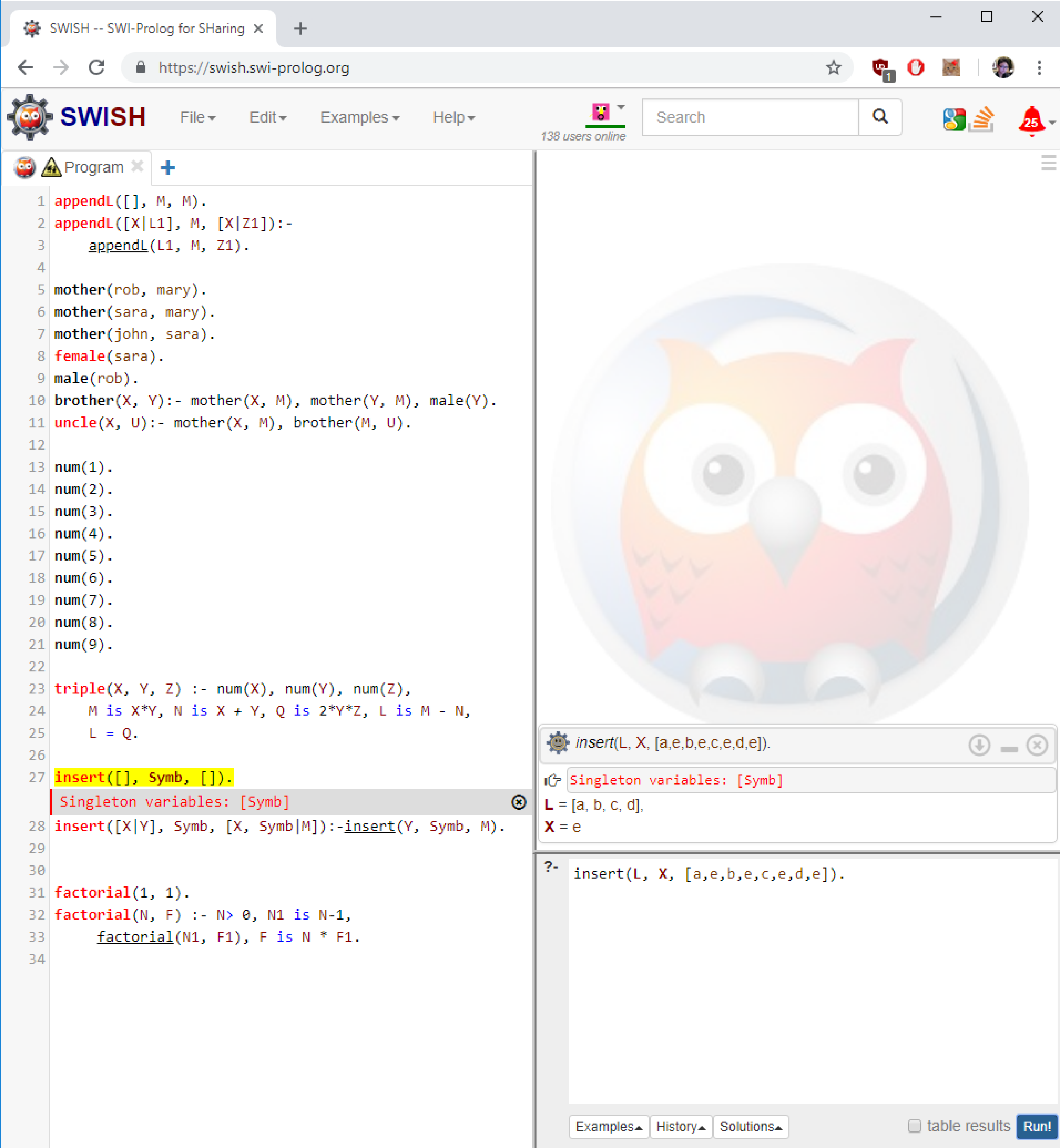
```
?- insert([a,b,c,d], e, Z).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
insert([a,b,c,d], X, [a,e,b,e,c,e,d,e]).
Singleton variables: [Symb]
X = e
```

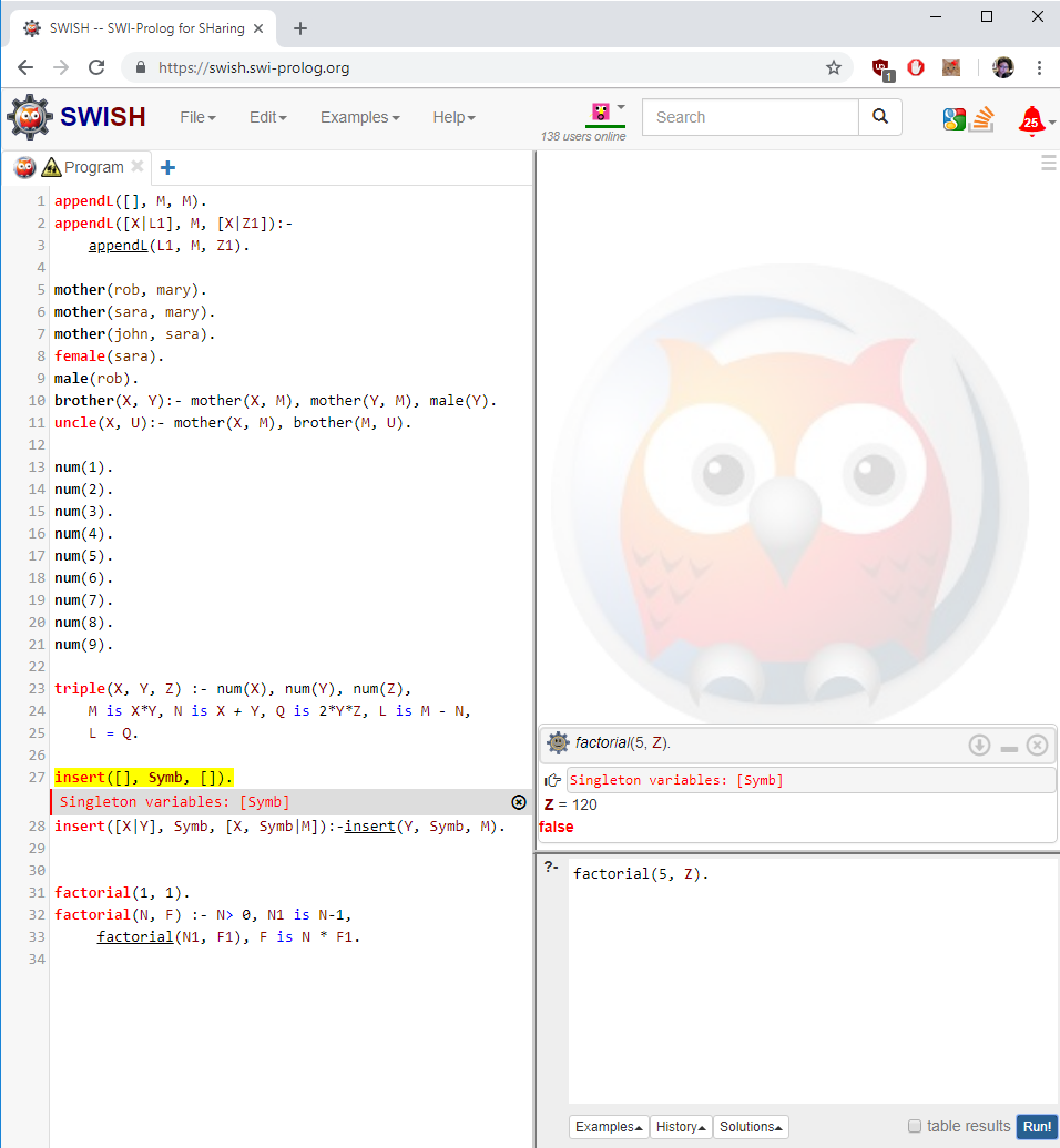
```
?- insert([a,b,c,d], X, [a,e,b,e,c,e,d,e]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
insert(L, X, [a,e,b,e,c,e,d,e]).
Singleton variables: [Symb]
L = [a, b, c, d],
X = e
```

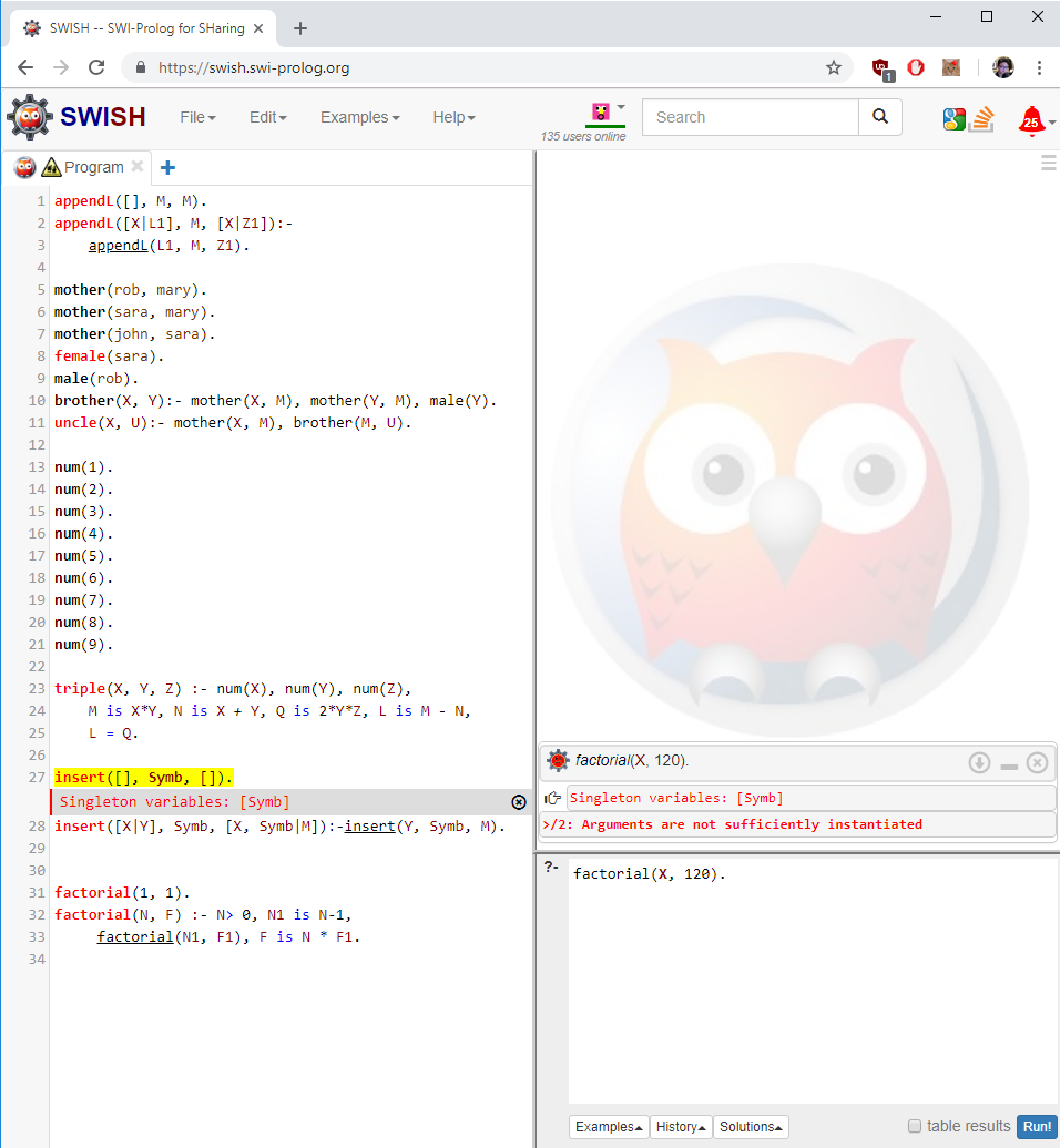
```
?- insert(L, X, [a,e,b,e,c,e,d,e]).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
factorial(5, Z).
Singleton variables: [Symb]
Z = 120
false
```

```
?- factorial(5, Z).
```



```
appendL([], M, M).
appendL([X|L1], M, [X|Z1]):-
    appendL(L1, M, Z1).

mother(rob, mary).
mother(sara, mary).
mother(john, sara).
female(sara).
male(rob).
brother(X, Y):- mother(X, M), mother(Y, M), male(Y).
uncle(X, U):- mother(X, M), brother(M, U).

num(1).
num(2).
num(3).
num(4).
num(5).
num(6).
num(7).
num(8).
num(9).

triple(X, Y, Z) :- num(X), num(Y), num(Z),
    M is X*Y, N is X + Y, Q is 2*Y*Z, L is M - N,
    L = Q.

insert([], Symb, []).
 Singleton variables: [Symb]
insert([X|Y], Symb, [X, Symb|M]):-insert(Y, Symb, M).


factorial(1, 1).
factorial(N, F) :- N> 0, N1 is N-1,
     factorial(N1, F1), F is N * F1.
```

```
factorial(X, 120).
Singleton variables: [Symb]
>/2: Arguments are not sufficiently instantiated
```

```
?- factorial(X, 120).
```

2. In the “triple” predicate, there are 4 solutions because there are 4 possibilities for M, N, Q, L, X, Y, and Z to all work together.

3. In factorial(X, 120), an error message occurs saying “Arguments are not sufficiently instantiated”. The program cannot backtrack because it does not know how many recursions are needed to reach 120.

4. In the “triple” query, the program tries to run the program with an integer. If that integer makes the program fail, the program backtracks and tries another integer.

5. The “insert” predicate takes the head (car) of the first parameter and puts it in the result, and it is followed by an “e” and then rest of the phrase(tail, cdr), which also has the “insert” predicate applied to it. This happens until the phrase has no tail.

6. The “brother” predicate works with a series of relationship that is predetermined between rob, mary, sara, and john. The program searches for two members with the same mother, where the second member is a male. Therefore, in brother(X, Y), X is sara, because Y needs to be rob, who is the only male in the sibling.