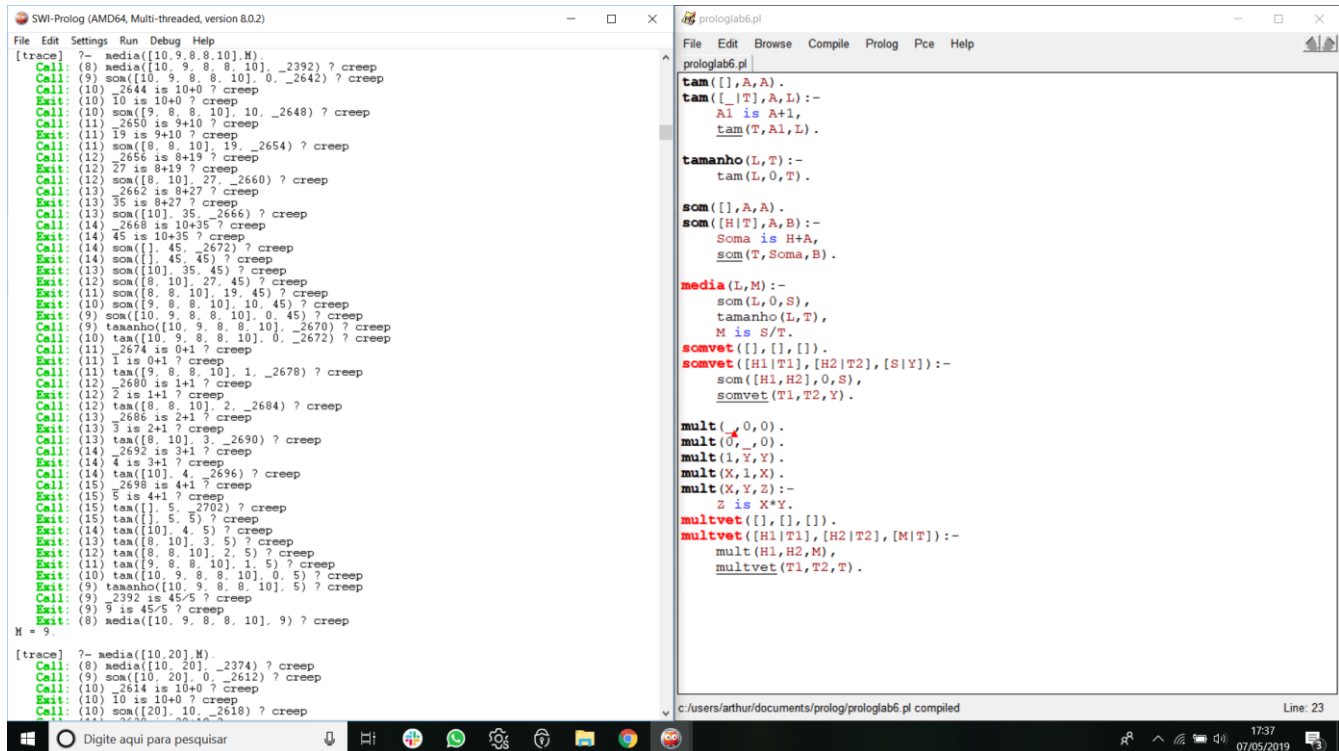


# Média



The screenshot shows two windows. The left window is SWI-Prolog (AMD64, Multi-threaded, version 8.0.2) displaying a Prolog trace. The right window is a text editor showing the source code of prologlab6.pl.

**SWI-Prolog Trace:**

```
[trace] ?- media([10,9,8,8,10],M).
Call: (8) media([10,9,8,8,10],_2392) ? creep
Call: (9) som([10,9,8,8,10],0,_2642) ? creep
Exit: (10) 10 is 10+0 ? creep
Call: (10) som([9,8,8,10],10,_2648) ? creep
Exit: (11) 2650 is 9+10 ? creep
Call: (11) som([8,8,10],19,_2654) ? creep
Exit: (12) 2656 is 8+19 ? creep
Call: (12) 27 is 8+19 ? creep
Exit: (12) som([8,10],27,_2660) ? creep
Call: (13) 2662 is 8+27 ? creep
Exit: (13) 35 is 8+27 ? creep
Call: (13) som([10],35,_2666) ? creep
Exit: (14) 2668 is 10+35 ? creep
Call: (14) 45 is 10+35 ? creep
Exit: (14) som([,45,2672) ? creep
Exit: (14) som([,45,45) ? creep
Exit: (13) som([10],35,45) ? creep
Exit: (12) som([8,10],27,45) ? creep
Exit: (11) som([8,8,10],19,45) ? creep
Exit: (10) som([9,8,8,10],10,45) ? creep
Call: (9) som([10,9,8,8,10],0,45) ? creep
Call: (9) tamanho([10,9,8,8,10],_2670) ? creep
Call: (10) tam([10,9,8,8,10],0,_2672) ? creep
Exit: (11) 1 is 0+1 ? creep
Exit: (11) tam([9,8,8,10],1,_2678) ? creep
Call: (12) 2680 is 1+1 ? creep
Exit: (12) 2 is 1+1 ? creep
Call: (12) tam([8,8,10],2,_2684) ? creep
Exit: (13) 2686 is 2+1 ? creep
Exit: (13) 3 is 2+1 ? creep
Call: (13) tam([8,10],3,_2690) ? creep
Exit: (14) 2692 is 3+1 ? creep
Exit: (14) 4 is 3+1 ? creep
Call: (14) tam([,4,2696) ? creep
Exit: (15) 2698 is 4+1 ? creep
Exit: (15) 5 is 4+1 ? creep
Call: (15) tam([,5,2702) ? creep
Exit: (15) tam([,5,5) ? creep
Exit: (14) tam([10,4,5) ? creep
Exit: (13) tam([8,10],3,5) ? creep
Exit: (12) tam([8,8,10],2,5) ? creep
Exit: (11) tam([9,8,8,10],1,5) ? creep
Exit: (10) tam([10,9,8,8,10],0,5) ? creep
Call: (9) tamanho([10,9,8,8,10],5) ? creep
Call: (8) 2392 is 45/5 ? creep
Exit: (8) media([10,9,8,8,10],9) ? creep
M = 9.

[trace] ?- media([10,20],M).
Call: (8) media([10,20],_2374) ? creep
Call: (9) som([10,20],0,_2612) ? creep
Exit: (10) 2614 is 10+0 ? creep
Exit: (10) 10 is 10+0 ? creep
Call: (10) som([10,20],10,_2618) ? creep
Exit: (11) 2620 is 10+10 ? creep
Exit: (11) tamanho([10,20],_2622) ? creep
Call: (12) tam([10,20],0,_2624) ? creep
Exit: (11) 2626 is 0+1 ? creep
Exit: (11) 1 is 0+1 ? creep
Call: (11) tam([20],1,_2630) ? creep
Exit: (12) 2632 is 1+1 ? creep
Exit: (12) 2 is 1+1 ? creep
Call: (12) tam([,2,2636) ? creep
Exit: (13) 2638 is 2+2 ? creep
Exit: (13) tam([20],1,2) ? creep
Exit: (10) tam([10,20],0,2) ? creep
Call: (9) tamanho([10,20],2) ? creep
Exit: (9) 2374 is 30/2 ? creep
Exit: (9) 15 is 30/2 ? creep
Exit: (8) media([10,20],15) ? creep
M = 15.
```

**prologlab6.pl Source Code:**

```
prologlab6.pl
File Edit Browse Compile Prolog Pce Help

tam([],A,A).
tam([_|T],A,L):-
    A1 is A+1,
    tam(T,A1,L).

tamanho(L,T):-
    tam(L,0,T).

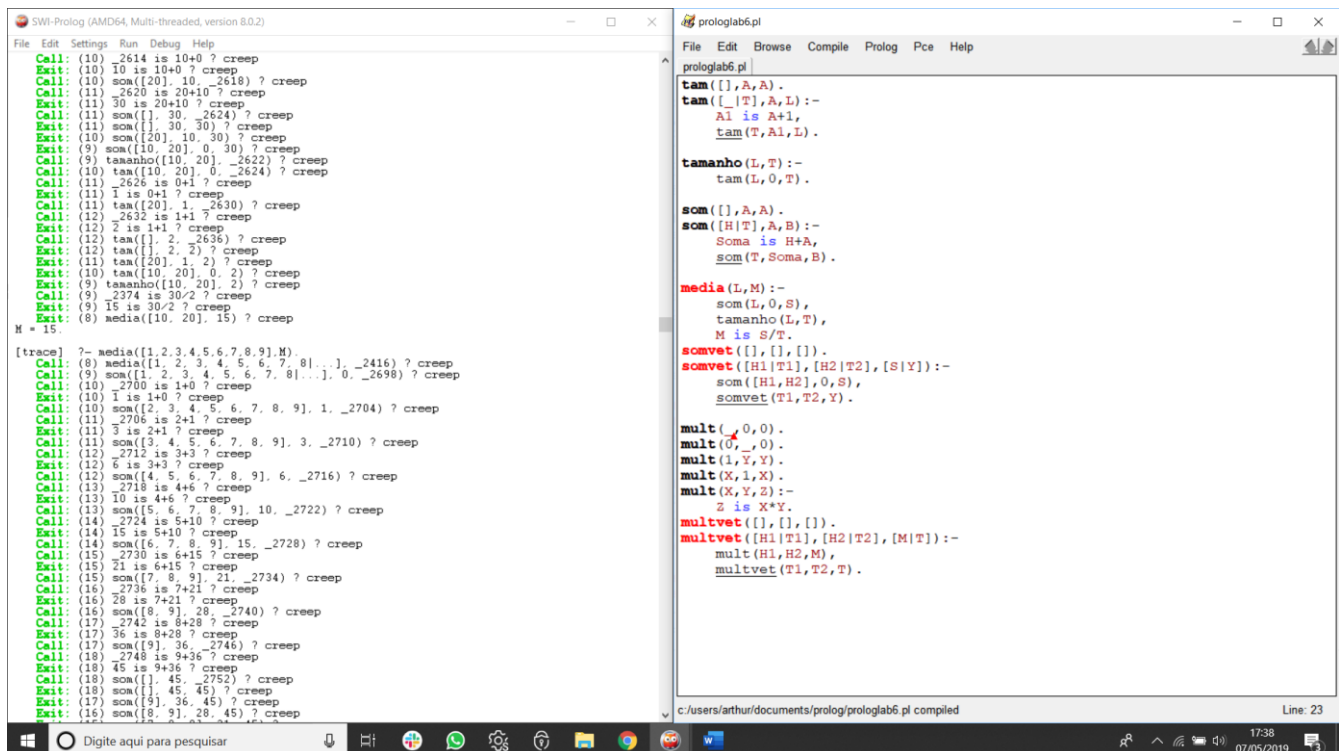
som([],A,A).
som([H|T],A,B):-
    Soma is H+A,
    som(T,Soma,B).

media(L,M):-
    som(L,0,S),
    tamanho(L,T),
    M is S/T.

somvet([],[],[]).
somvet([H1|T1],[H2|T2],[S|Y]):-
    som([H1,H2],0,S),
    somvet(T1,T2,Y).

mult(0,0,0).
mult(0,_,0).
mult(1,Y,Y).
mult(X,1,X).
mult(X,Y,Z):-
    Z is X*Y.

multvet([],[],[]).
multvet([H1|T1],[H2|T2],[M|T]):-
    mult(H1,H2,M),
    multvet(T1,T2,T).
```



The screenshot shows two windows. The left window is SWI-Prolog (AMD64, Multi-threaded, version 8.0.2) displaying a Prolog trace. The right window is a text editor showing the source code of prologlab6.pl.

**SWI-Prolog Trace:**

```
[trace] ?- media([1,2,3,4,5,6,7,8,9],M).
Call: (8) media([1,2,3,4,5,6,7,8,9],_2416) ? creep
Call: (9) som([1,2,3,4,5,6,7,8,9],0,_2698) ? creep
Exit: (10) 2700 is 1+0 ? creep
Exit: (10) 1 is 1+0 ? creep
Call: (10) som([2,3,4,5,6,7,8,9],1,_2704) ? creep
Exit: (11) 2706 is 2+1 ? creep
Exit: (11) 3 is 2+1 ? creep
Call: (11) som([3,4,5,6,7,8,9],3,_2710) ? creep
Exit: (12) 2712 is 3+3 ? creep
Exit: (12) 6 is 3+3 ? creep
Call: (12) som([4,5,6,7,8,9],6,_2716) ? creep
Exit: (13) 2718 is 4+6 ? creep
Exit: (13) 10 is 4+6 ? creep
Call: (13) som([5,6,7,8,9],10,_2722) ? creep
Exit: (14) 2724 is 5+10 ? creep
Exit: (14) 15 is 5+10 ? creep
Call: (14) som([6,7,8,9],15,_2728) ? creep
Exit: (15) 2730 is 6+15 ? creep
Exit: (15) 21 is 6+15 ? creep
Call: (15) som([7,8,9],21,_2734) ? creep
Exit: (16) 2736 is 7+21 ? creep
Exit: (16) 28 is 7+21 ? creep
Call: (16) som([8,9],28,_2740) ? creep
Exit: (17) 2742 is 8+28 ? creep
Exit: (17) 36 is 8+28 ? creep
Call: (17) som([9],36,_2746) ? creep
Exit: (18) 2748 is 9+36 ? creep
Exit: (18) 45 is 9+36 ? creep
Call: (18) som([,45,2752) ? creep
Exit: (18) som([,45,45) ? creep
Exit: (17) som([9],36,45) ? creep
Exit: (16) som([8,9],28,45) ? creep
Exit: (15) som([7,8,9],21,45) ? creep
Exit: (14) som([6,7,8,9],15,45) ? creep
Exit: (13) som([5,6,7,8,9],10,45) ? creep
Exit: (12) som([4,5,6,7,8,9],6,45) ? creep
Exit: (11) som([3,4,5,6,7,8,9],3,45) ? creep
Exit: (10) som([2,3,4,5,6,7,8,9],1,45) ? creep
Exit: (9) som([1,2,3,4,5,6,7,8,9],0,45) ? creep
Exit: (8) media([1,2,3,4,5,6,7,8,9],45) ? creep
M = 45.
```

**prologlab6.pl Source Code:**

```
prologlab6.pl
File Edit Browse Compile Prolog Pce Help

tam([],A,A).
tam([_|T],A,L):-
    A1 is A+1,
    tam(T,A1,L).

tamanho(L,T):-
    tam(L,0,T).

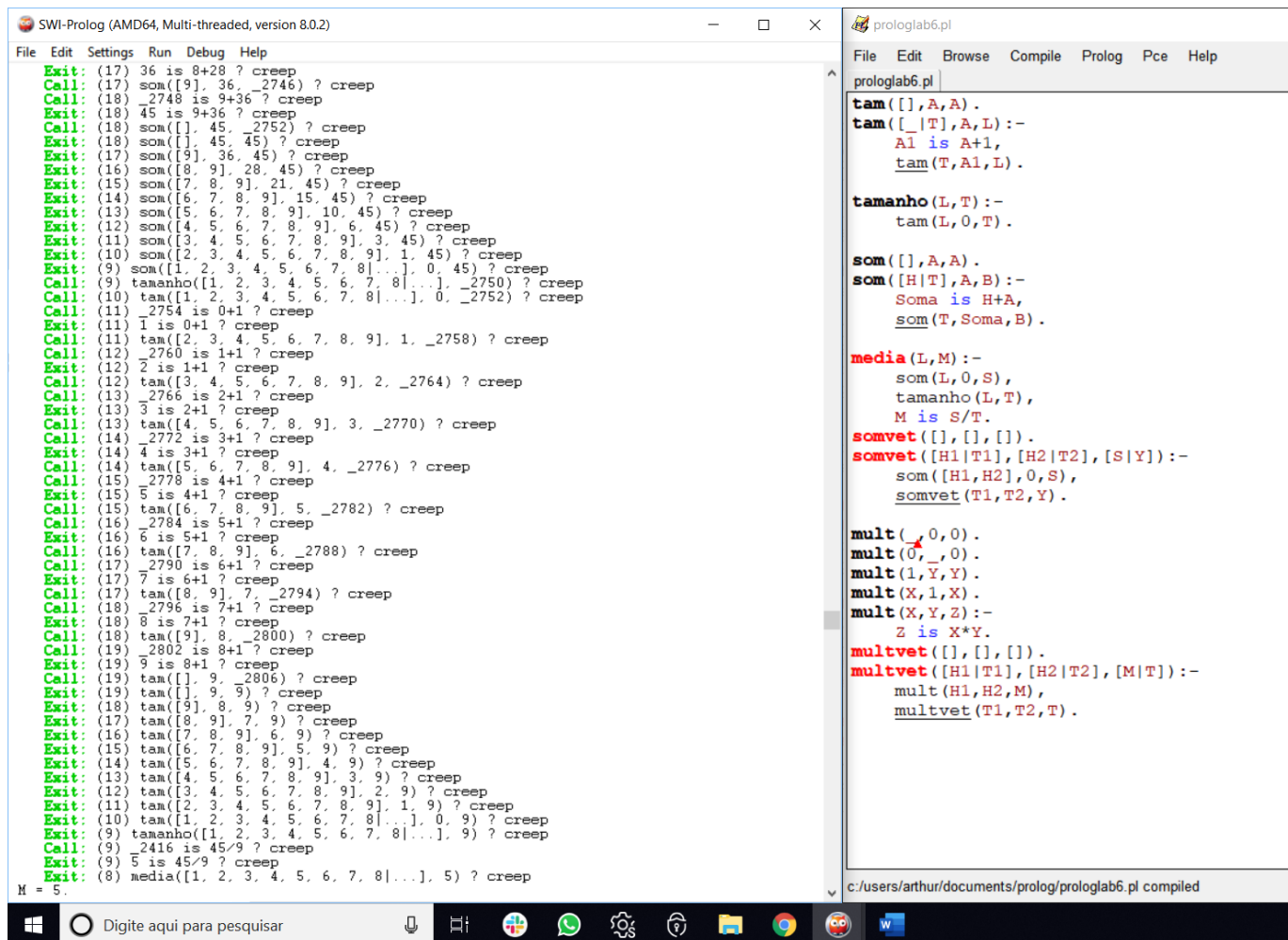
som([],A,A).
som([H|T],A,B):-
    Soma is H+A,
    som(T,Soma,B).

media(L,M):-
    som(L,0,S),
    tamanho(L,T),
    M is S/T.

somvet([],[],[]).
somvet([H1|T1],[H2|T2],[S|Y]):-
    som([H1,H2],0,S),
    somvet(T1,T2,Y).

mult(0,0,0).
mult(0,_,0).
mult(1,Y,Y).
mult(X,1,X).
mult(X,Y,Z):-
    Z is X*Y.

multvet([],[],[]).
multvet([H1|T1],[H2|T2],[M|T]):-
    mult(H1,H2,M),
    multvet(T1,T2,T).
```



## Soma dos Vetores

```
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
[trace] 7- sonavet([1,2,3],[3,4,5],L).
Call: (8) sonavet([1,2,3],[3,4,5],_3982) ? creep
Exit: (9) son([1,3],0,_4232) ? creep
Call: (10) _4270 is 1+0 ? creep
Exit: (10) 1 is 1+0 ? creep
Call: (10) son([3],1,_4232) ? creep
Exit: (11) _4276 is 3+1 ? creep
Exit: (11) 4 is 3+1 ? creep
Call: (11) son([1],4,_4232) ? creep
Exit: (11) son([1],4,6) ? creep
Exit: (10) son([3],1,4) ? creep
Exit: (9) son([1,3],0,4) ? creep
Call: (9) sonavet([2,3],[4,5],_4234) ? creep
Call: (10) son([2,4],0,_4262) ? creep
Call: (11) _4300 is 2+0 ? creep
Exit: (11) 2 is 2+0 ? creep
Call: (11) son([4],2,_4262) ? creep
Call: (12) _4306 is 4+2 ? creep
Exit: (12) 6 is 4+2 ? creep
Call: (12) son([1],6,_4262) ? creep
Exit: (12) son([1],6,6) ? creep
Exit: (11) son([4],2,6) ? creep
Exit: (10) son([2,4],0,6) ? creep
Call: (10) sonavet([2],[5],_4264) ? creep
Call: (11) son([5],0,_4292) ? creep
Exit: (12) 3 is 3+0 ? creep
Exit: (12) 3 is 3+0 ? creep
Call: (12) son([5],3,_4292) ? creep
Call: (13) _4336 is 5+3 ? creep
Exit: (13) 8 is 5+3 ? creep
Call: (13) son([1],8,_4292) ? creep
Exit: (13) son([1],8,8) ? creep
Exit: (12) son([5],3,8) ? creep
Exit: (11) son([5],0,8) ? creep
Call: (11) sonavet([1],_4294) ? creep
Exit: (11) sonavet([1],[1]) ? creep
Exit: (10) sonavet([5],[9]) ? creep
Exit: (9) sonavet([2,3],[4,5],[6,8]) ? creep
Exit: (8) sonavet([1,2,3],[3,4,5],[4,6,8]) ? creep
I = [4,6,8]
[trace] 7- sonavet([2,2,2],[1,0,4],L).
Call: (8) sonavet([2,2,2],[1,0,4],_3982) ? creep
Call: (9) son([2,1],0,_4232) ? creep
Call: (10) _4270 is 2+0 ? creep
Exit: (10) 2 is 2+0 ? creep
Call: (11) _4276 is 1+2 ? creep
Exit: (11) 3 is 1+2 ? creep
Call: (11) son([1],3,_4232) ? creep
Exit: (11) son([1],3,3) ? creep
Exit: (10) son([2,1],0,3) ? creep
Exit: (9) son([2,1],0,3) ? creep
Call: (9) sonavet([2,2],[0,4],_4234) ? creep
Call: (10) son([2,0],0,_4262) ? creep
Call: (11) _4300 is 2+0 ? creep
Exit: (11) 2 is 2+0 ? creep
Call: (11) son([0],2,_4262) ? creep
Call: (12) _4306 is 0+2 ? creep
Exit: (12) 2 is 0+2 ? creep
Call: (12) son([1],2,_4262) ? creep
Exit: (12) son([1],2,2) ? creep
Exit: (11) son([0],2,2) ? creep
Exit: (10) son([2,0],0,2) ? creep
Call: (10) sonavet([2],[4],_4264) ? creep
Call: (11) son([2,4],0,_4292) ? creep
Exit: (12) 2 is 2+0 ? creep
Call: (12) son([4],2,_4292) ? creep
Exit: (13) _4336 is 4+2 ? creep
Exit: (13) 6 is 4+2 ? creep
Call: (13) son([1],6,_4292) ? creep
Exit: (13) son([1],6,6) ? creep
Exit: (12) son([4],2,6) ? creep
Exit: (11) son([2,4],0,6) ? creep
Call: (11) sonavet([1],_4294) ? creep
Exit: (11) sonavet([1],[1]) ? creep
Exit: (10) sonavet([2],[4],[6]) ? creep
Exit: (9) sonavet([2,2],[0,4],[2,6]) ? creep
Exit: (8) sonavet([2,2,2],[1,0,4],[3,2,6]) ? creep
I = [3,2,6]
[trace] 7- sonavet([4,4,4,4],[1,2,3,5,4],L).
Correct to: "sonavet([4,4,4,4],[1,2,3,5,4],L)"? yes
Call: (8) sonavet([4,4,4,4],[1,2,3,5,4],_4006) ? creep
Call: (9) son([4,1],0,_4820) ? creep
Exit: (10) _4866 is 4+0 ? creep
Exit: (10) 4 is 4+0 ? creep
Call: (10) son([1],4,_4820) ? creep
Exit: (11) _4872 is 1+4 ? creep
Exit: (11) 5 is 1+4 ? creep
Call: (11) son([1],5,_4820) ? creep
Exit: (11) son([1],5,5) ? creep
Exit: (10) son([4,1],0,5) ? creep
Exit: (9) son([4,1],0,5) ? creep
Call: (9) sonavet([4,4,4],[2,3,5,4],_4830) ? creep
Call: (10) son([4,2],0,_4858) ? creep
Exit: (11) _4896 is 4+0 ? creep
Exit: (11) 4 is 4+0 ? creep
Call: (11) son([2],4,_4858) ? creep
Call: (12) _4902 is 2+4 ? creep
Exit: (12) 6 is 2+4 ? creep
Call: (12) son([1],6,_4858) ? creep
Exit: (12) son([1],6,6) ? creep
Exit: (11) son([2],4,6) ? creep
```

```
prologlab6.pl
File Edit Browse Compile Prolog Pce Help
prologlab6.pl
tam([1,A,A]).
tam([1,T],A,L):-
    A is A+1,
    tam(T,A1,L).

tamanho(L,T):-
    tam(L,0,T).

som([1,A,A]).
som([H|T],A,B):-
    Soma is H+A,
    som(T,Soma,B).

media(L,M):-
    som(L,0,S),
    tamanho(L,T),
    M is S/T.

sonavet([1],[1],[1]).
sonavet([H1|T1],[H2|T2],[S|Y]):-
    som([H1,H2],0,8),
    sonavet(T1,T2,Y).

mult(0,0,0).
mult(0,_,0).
mult(1,_,Y).
mult(X,1,X).
mult(X,Y,2):-
    2 is X*Y.
multvet([1],[1],[1]).
multvet([H1|T1],[H2|T2],[M|T]):-
    mult(H1,H2,M),
    multvet(T1,T2,T).
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
[trace] 7- sonavet([1,2,3],[3,4,5],L).
Call: (8) sonavet([1,2,3],[3,4,5],_3982) ? creep
Call: (9) son([1,3],0,_4232) ? creep
Exit: (10) _4270 is 1+0 ? creep
Exit: (10) 1 is 1+0 ? creep
Call: (10) son([3],1,_4232) ? creep
Exit: (11) _4276 is 3+1 ? creep
Exit: (11) 4 is 3+1 ? creep
Call: (11) son([1],4,_4232) ? creep
Exit: (11) son([1],4,6) ? creep
Exit: (10) son([3],1,4) ? creep
Exit: (9) son([1,3],0,4) ? creep
Call: (9) sonavet([2,3],[4,5],_4234) ? creep
Call: (10) son([2,4],0,_4262) ? creep
Call: (11) _4300 is 2+0 ? creep
Exit: (11) 2 is 2+0 ? creep
Call: (11) son([4],2,_4262) ? creep
Call: (12) _4306 is 4+2 ? creep
Exit: (12) 6 is 4+2 ? creep
Call: (12) son([1],6,_4262) ? creep
Exit: (12) son([1],6,6) ? creep
Exit: (11) son([4],2,6) ? creep
Exit: (10) son([2,4],0,6) ? creep
Call: (10) sonavet([2],[5],_4264) ? creep
Call: (11) son([5],0,_4292) ? creep
Exit: (12) 3 is 3+0 ? creep
Exit: (12) 3 is 3+0 ? creep
Call: (12) son([5],3,_4292) ? creep
Call: (13) _4336 is 5+3 ? creep
Exit: (13) 8 is 5+3 ? creep
Call: (13) son([1],8,_4292) ? creep
Exit: (13) son([1],8,8) ? creep
Exit: (12) son([5],3,8) ? creep
Exit: (11) son([5],0,8) ? creep
Call: (11) sonavet([1],_4294) ? creep
Exit: (11) sonavet([1],[1]) ? creep
Exit: (10) sonavet([5],[9]) ? creep
Exit: (9) sonavet([2,3],[4,5],[6,8]) ? creep
Exit: (8) sonavet([1,2,3],[3,4,5],[4,6,8]) ? creep
I = [4,6,8]
[trace] 7- sonavet([2,2,2],[1,0,4],L).
Call: (8) sonavet([2,2,2],[1,0,4],_3982) ? creep
Call: (9) son([2,1],0,_4232) ? creep
Call: (10) _4270 is 2+0 ? creep
Exit: (10) 2 is 2+0 ? creep
Call: (11) _4276 is 1+2 ? creep
Exit: (11) 3 is 1+2 ? creep
Call: (11) son([1],3,_4232) ? creep
Exit: (11) son([1],3,3) ? creep
Exit: (10) son([2,1],0,3) ? creep
Exit: (9) son([2,1],0,3) ? creep
Call: (9) sonavet([2,2],[0,4],_4234) ? creep
Call: (10) son([2,0],0,_4262) ? creep
Call: (11) _4300 is 2+0 ? creep
Exit: (11) 2 is 2+0 ? creep
Call: (11) son([0],2,_4262) ? creep
Call: (12) _4306 is 0+2 ? creep
Exit: (12) 2 is 0+2 ? creep
Call: (12) son([1],2,_4262) ? creep
Exit: (12) son([1],2,2) ? creep
Exit: (11) son([0],2,2) ? creep
Exit: (10) son([2,0],0,2) ? creep
Call: (10) sonavet([2],[4],_4264) ? creep
Call: (11) son([2,4],0,_4292) ? creep
Exit: (12) 2 is 2+0 ? creep
Call: (12) son([4],2,_4292) ? creep
Exit: (13) _4336 is 4+2 ? creep
Exit: (13) 6 is 4+2 ? creep
Call: (13) son([1],6,_4292) ? creep
Exit: (13) son([1],6,6) ? creep
Exit: (12) son([4],2,6) ? creep
Exit: (11) son([2,4],0,6) ? creep
Call: (11) sonavet([1],_4294) ? creep
Exit: (11) sonavet([1],[1]) ? creep
Exit: (10) sonavet([2],[4],[6]) ? creep
Exit: (9) sonavet([2,2],[0,4],[2,6]) ? creep
Exit: (8) sonavet([2,2,2],[1,0,4],[3,2,6]) ? creep
I = [3,2,6]
[trace] 7- sonavet([4,4,4,4],[1,2,3,5,4],L).
Correct to: "sonavet([4,4,4,4],[1,2,3,5,4],L)"? yes
Call: (8) sonavet([4,4,4,4],[1,2,3,5,4],_4006) ? creep
Call: (9) son([4,1],0,_4820) ? creep
Exit: (10) _4866 is 4+0 ? creep
Exit: (10) 4 is 4+0 ? creep
Call: (10) son([1],4,_4820) ? creep
Exit: (11) _4872 is 1+4 ? creep
Exit: (11) 5 is 1+4 ? creep
Call: (11) son([1],5,_4820) ? creep
Exit: (11) son([1],5,5) ? creep
Exit: (10) son([4,1],0,5) ? creep
Exit: (9) son([4,1],0,5) ? creep
Call: (9) sonavet([4,4,4],[2,3,5,4],_4830) ? creep
Call: (10) son([4,2],0,_4858) ? creep
Exit: (11) _4896 is 4+0 ? creep
Exit: (11) 4 is 4+0 ? creep
Call: (11) son([2],4,_4858) ? creep
Call: (12) _4902 is 2+4 ? creep
Exit: (12) 6 is 2+4 ? creep
Call: (12) son([1],6,_4858) ? creep
Exit: (12) son([1],6,6) ? creep
Exit: (11) son([2],4,6) ? creep
```

```
prologlab6.pl
File Edit Browse Compile Prolog Pce Help
prologlab6.pl
tam([1,A,A]).
tam([1,T],A,L):-
    A is A+1,
    tam(T,A1,L).

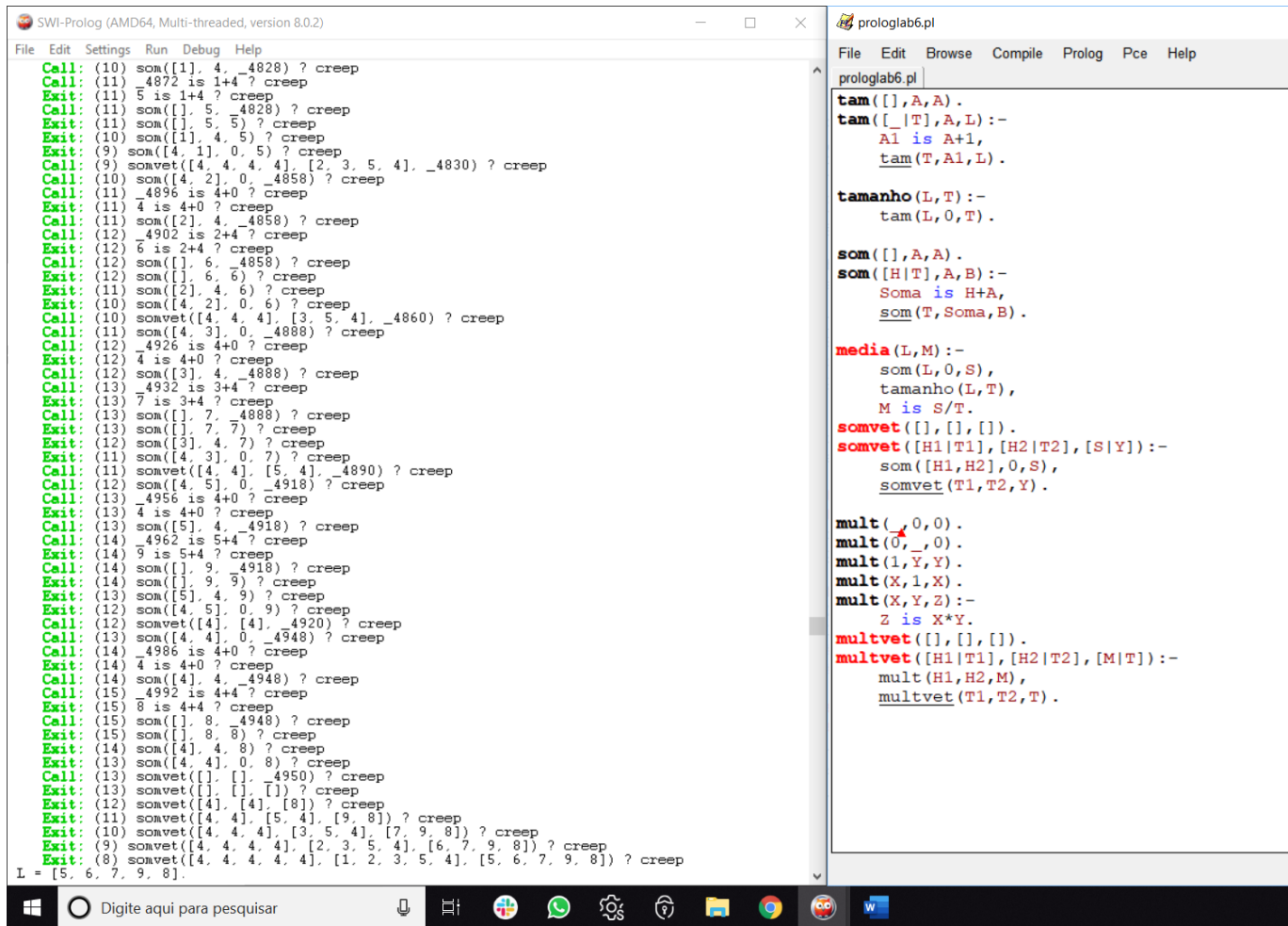
tamanho(L,T):-
    tam(L,0,T).

som([1,A,A]).
som([H|T],A,B):-
    Soma is H+A,
    som(T,Soma,B).

media(L,M):-
    som(L,0,S),
    tamanho(L,T),
    M is S/T.

sonavet([1],[1],[1]).
sonavet([H1|T1],[H2|T2],[S|Y]):-
    som([H1,H2],0,8),
    sonavet(T1,T2,Y).

mult(0,0,0).
mult(0,_,0).
mult(1,_,Y).
mult(X,1,X).
mult(X,Y,2):-
    2 is X*Y.
multvet([1],[1],[1]).
multvet([H1|T1],[H2|T2],[M|T]):-
    mult(H1,H2,M),
    multvet(T1,T2,T).
```



## Produto escalar dos vetores

