



# Instituto Tecnológico de Mexicali

## Practica 9

### **Carrera:**

Ingeniería en Sistemas Computacionales

### **Nombre del Alumno(a):**

Martínez Yebra Beatriz Andrea  
#13490929

### **Nombre del profesor(a):**

Tafoya Diaz Luis Aram

### **Materia:**

Programación Lógica y Funcional

### **Horario:**

5:00 p.m. – 6:00 p.m.

Mexicali Baja California, Miércoles 22 de Noviembre 2017.

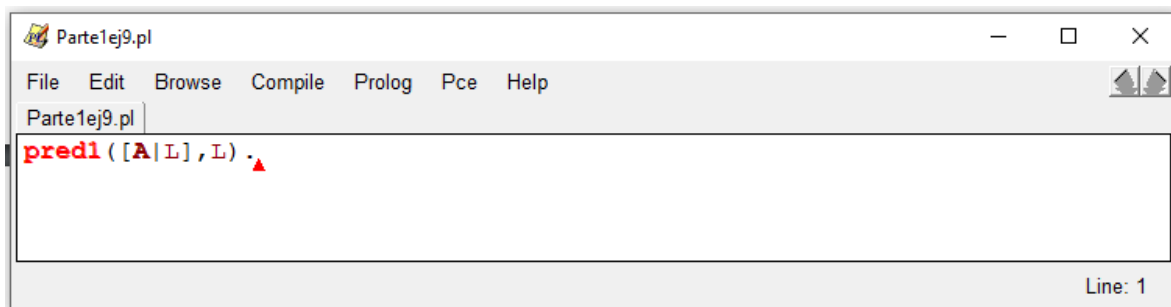
## Parte I

1. Define y prueba el predicado `pred1` e cual toma una lista en el primer argumento y regresa la cola(tail) de la lista en el segundo argumento.

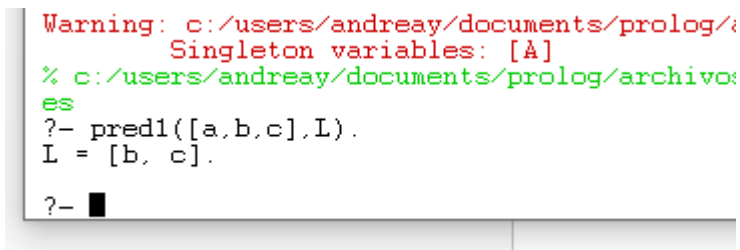
Por ejemplo:

?- `pred1([a,b,c],L).`

`L = [b, c].`



```
Parte1ej9.pl
File Edit Browse Compile Prolog Pce Help
Parte1ej9.pl
pred1 ([A|L], L) .
Line: 1
```



```
Warning: c:/users/andreay/documents/prolog/
Singleton variables: [A]
% c:/users/andreay/documents/prolog/archivos
es
?- pred1([a,b,c],L) .
L = [b, c].
?-
```

## Parte II

1. Define y prueba el predicado `inc` el cual toma una lista de números en el primer argumento y regresa una lista con los mismos números todos incrementados en uno en el segundo argumento.

**Por ejemplo:**

?- `inc`[1,2,-9,0,100,-54],L).

L = [2,3,-8,1,101,-53].

## Parte III

1. Define y prueba el predicado palíndromo el cual comprueba si la lista se puede leer de la misma forma de izquierda a derecha que de derecha a izquierda.

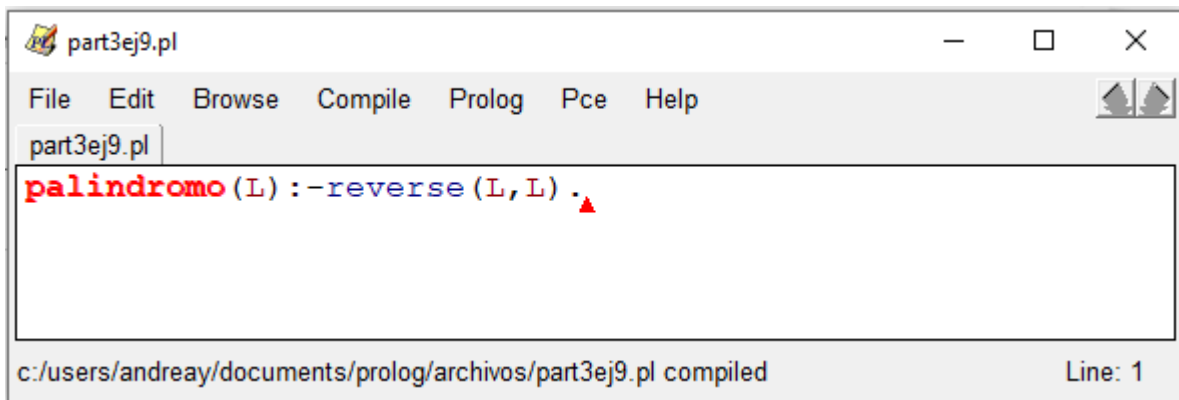
Por ejemplo:

?- palindromo([a,b,c,b,a]).

true.

?- palidromo([a,b,c,d,e]).

false.



```
part3ej9.pl
File Edit Browse Compile Prolog Pce Help
palindromo(L) :- reverse(L, L) .
c:/users/andreay/documents/prolog/archivos/part3ej9.pl compiled Line: 1
```

```
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is f
Please run ?- license. for legal details.
```

```
For online help and background, visit http://www.swi-pr
For built-in help, use ?- help(Topic). or ?- apropos(Wc
```

```
?-
% c:/users/andreay/documents/prolog/archivos/part3ej9 c
es
?- palindromo([a,b,c,b,a]).
true.
?- palindromo([a,b,c,d,e]).
false.
?- ■
```

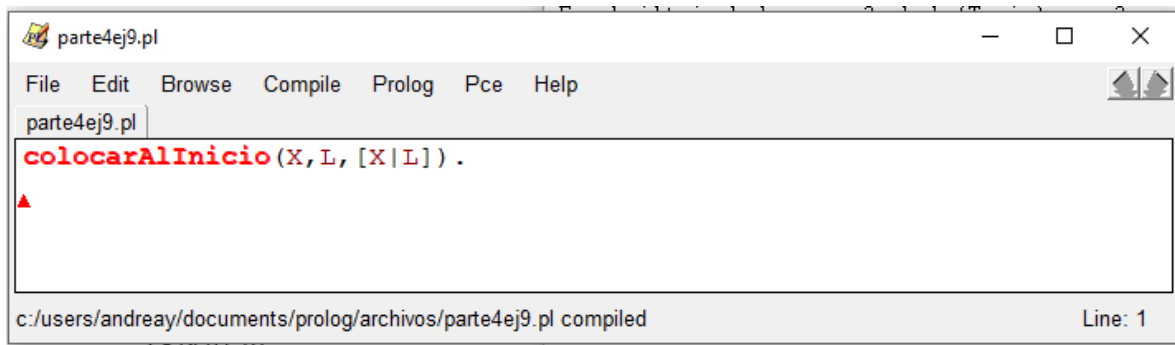
## Parte IV

1. Define y prueba el predicado `colocarAlInicio` el cual agrega un término específico al inicio de la lista.

Por ejemplo:

?- colocarAlInicio(a,[b,c,d],L).

L- [a,b,c,d].



```
?- colocarAlInicio(a,[b,c,d],L).  
L = [a, b, c, d].  
?-
```

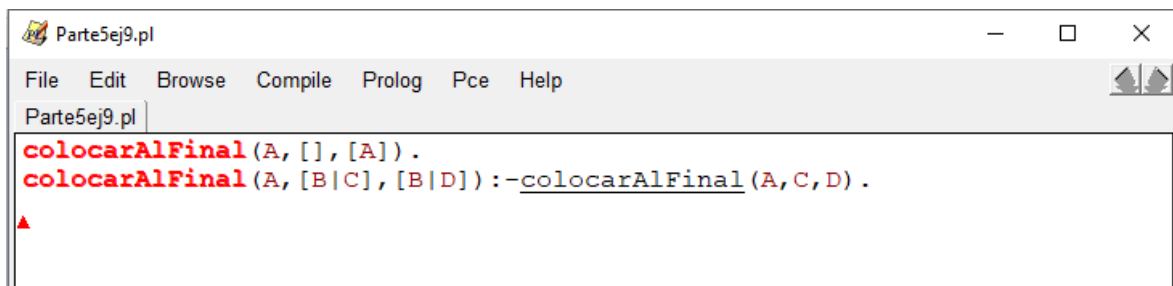
## Parte V

1. Define y prueba el predicado `colocarAlFinal` el cual agrega un término específico al final de la lista.

Por ejemplo:

?- colocarAlInicio(e,[a,b,c,d],L).

L=[a,b,c,d,e].



```
colocarAlFinal(A, [], [A]).  
colocarAlFinal(A, [B|C], [B|D]) :- colocarAlFinal(A, C, D).
```

```
?- colocarAlFinal(e,[a,b,c,d],L).  
L = [a, b, c, d, e]
```

## Parte VI

1. Utilizado `findall` define y prueba los predicados `pred2/2`, `pred3/2` y `pred4/2` que modifican una lista.

Por ejemplo:

?- `pred2([a,b,c],L).`

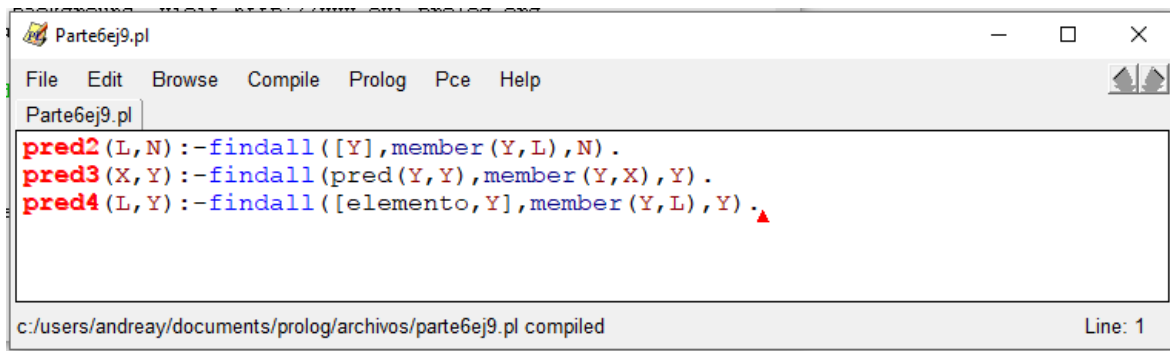
L- `[[a],[b],[c]].`

? `pred3([a,b,c],L).`


L- `[pred(a,a),pred(b,b),pred(c,c)]-`

?- `pred4([a,b,c],L).`

L- `[[elemento,a],[elemento,b],[elemento,c]].`



```
File Edit Browse Compile Prolog Pce Help
Parte6ej9.pl
pred2(L,N):-findall([Y],member(Y,L),N).
pred3(X,Y):-findall(pred(Y,Y),member(Y,X),Y).
pred4(L,Y):-findall([elemento,Y],member(Y,L),Y).
c:/users/andreay/documents/prolog/archivos/parte6ej9.pl compiled Line: 1
```



```
SWI-Prolog (AMD64, Multi-threaded, version 7.6.0)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/users/andreay/documents/prolog/archivos/parte6ej9 compiled 0.00 sec, 0 clauses
?- pred2([a,b,c],L).
L = [[a], [b], [c]].

?- pred3([a,b,c],L).
L = [pred(a, a), pred(b, b), pred(c, c)].

?- pred4([a,b,c],L).
L = [[elemento, a], [elemento, b], [elemento, c]].

?-
```