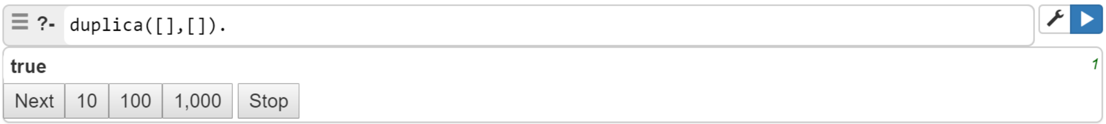
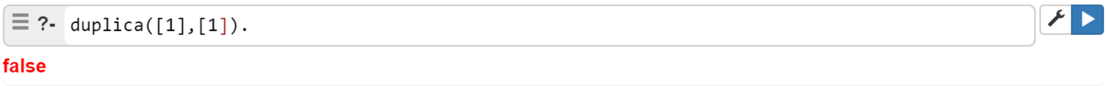
Práctica 3

Victoria Pelayo e Ignacio Rabuñal

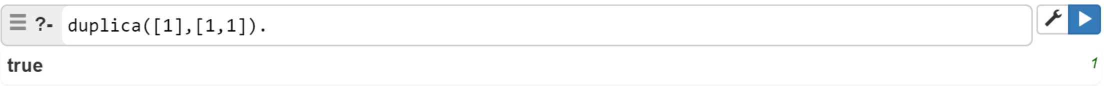
Ejercicio 1



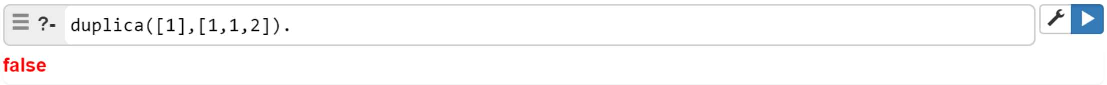
```
?- duplica([],[]).
true
```



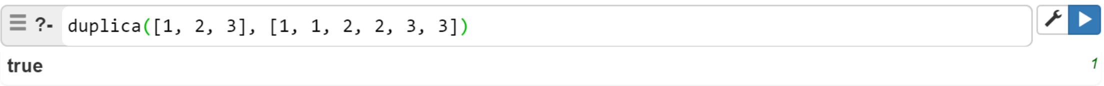
```
?- duplica([1],[1]).
false
```



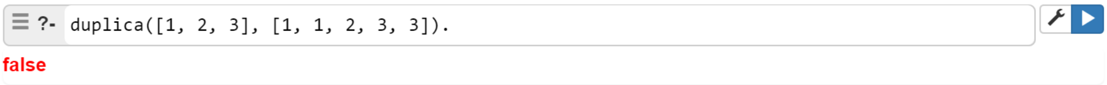
```
?- duplica([1],[1,1]).
true
```



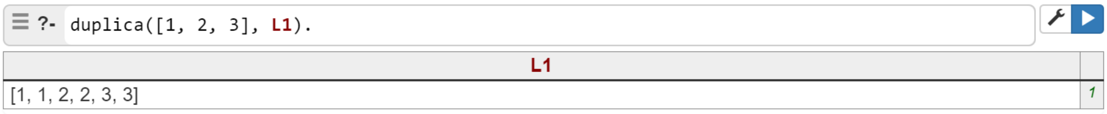
```
?- duplica([1],[1,1,2]).
false
```



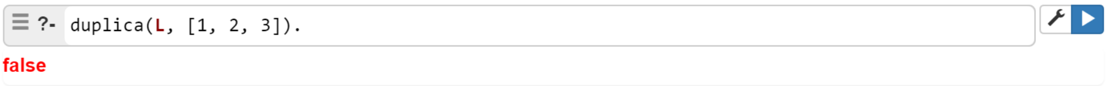
```
?- duplica([1, 2, 3], [1, 1, 2, 2, 3, 3])
true
```



```
?- duplica([1, 2, 3], [1, 1, 2, 3, 3]).
false
```

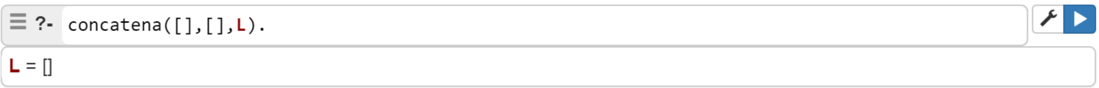


```
?- duplica([1, 2, 3], L1).
L1
[1, 1, 2, 2, 3, 3]
```

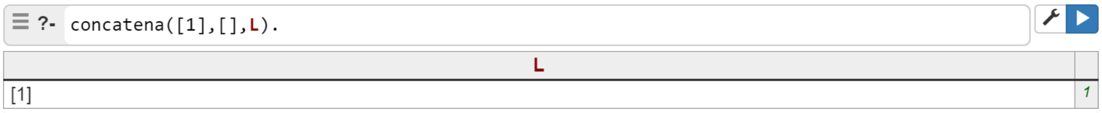


```
?- duplica(L, [1, 2, 3]).
false
```

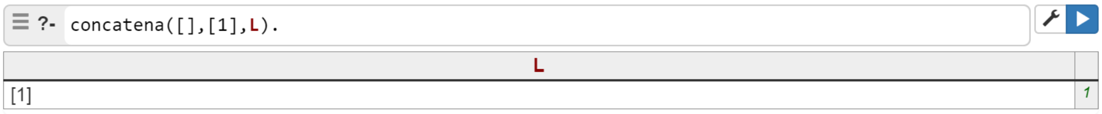
Ejercicio 2



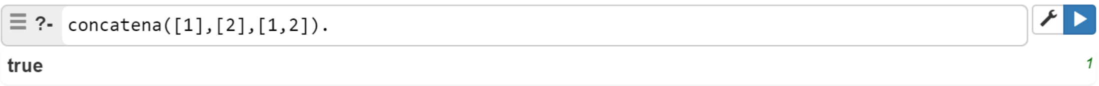
```
?- concatena([],[],L).
L = []
```



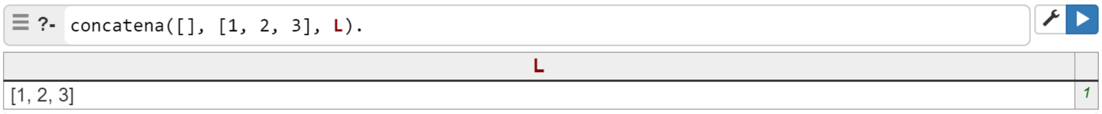
```
?- concatena([1],[],L).
L
[1]
```



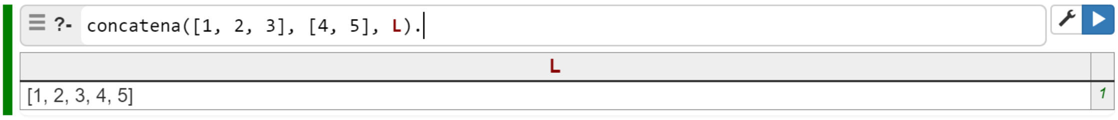
```
?- concatena([],[1],L).
L
[1]
```



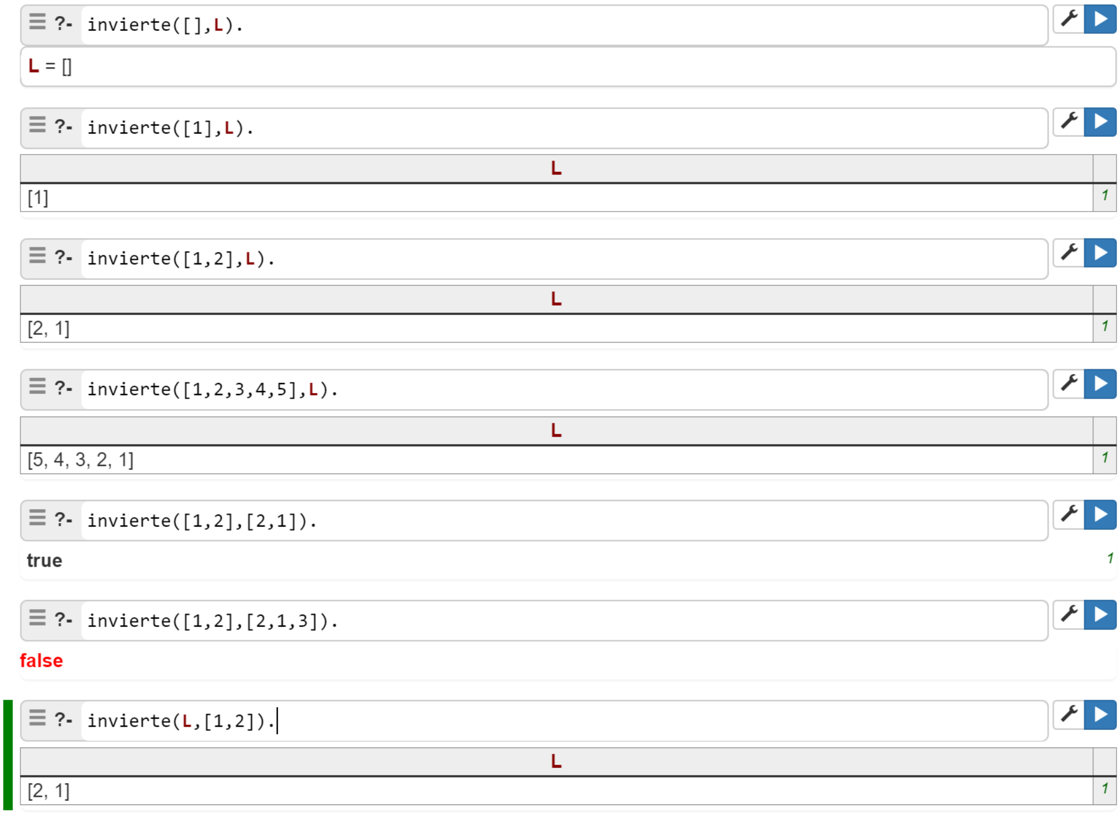
```
?- concatena([1],[2],[1,2]).
true
```



```
?- concatena([], [1, 2, 3], L).
L
[1, 2, 3]
```

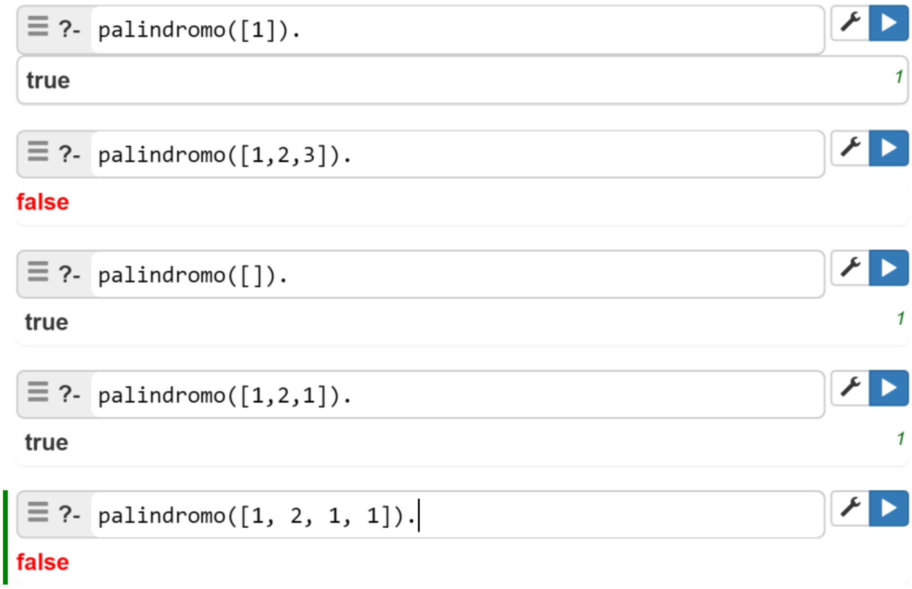


```
?- concatena([1, 2, 3], [4, 5], L).
L
[1, 2, 3, 4, 5]
```



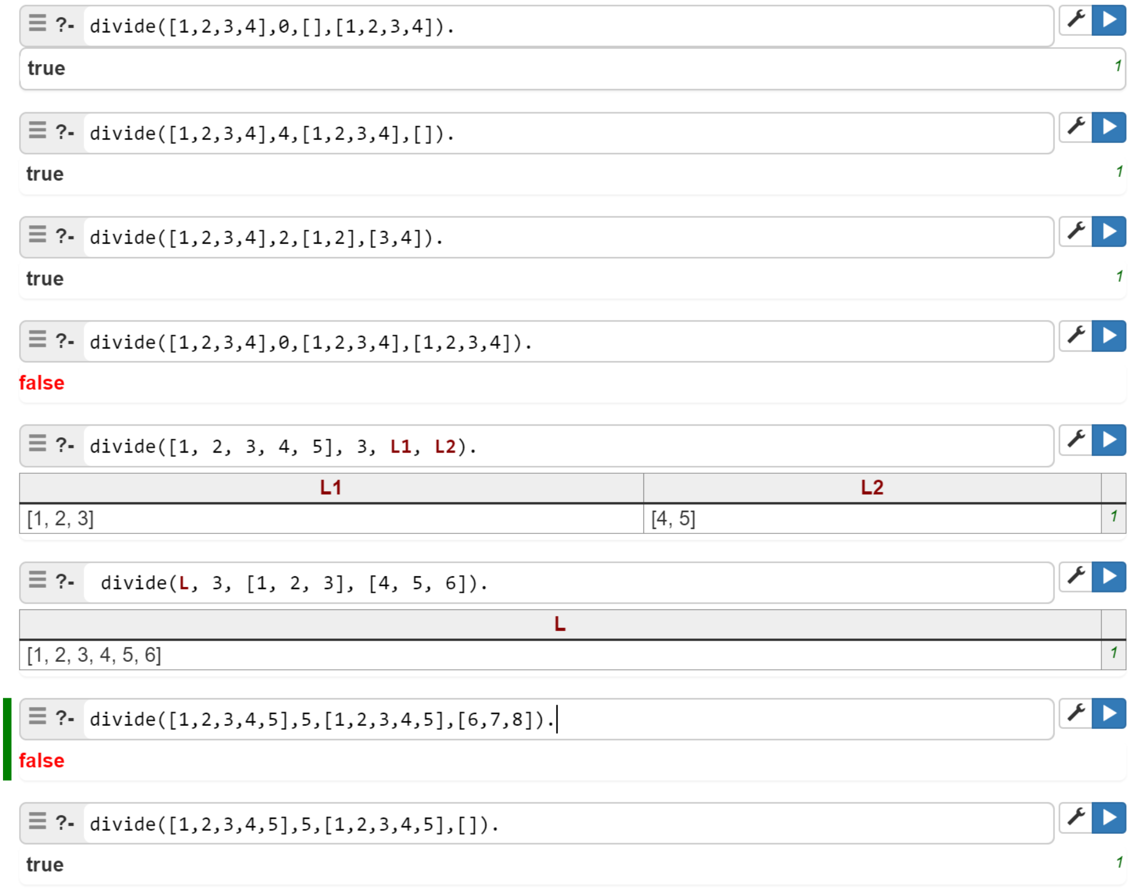
```
?- invierte([],L).
L = []

?- invierte([1],L).
                              L
[1]                                                   1

?- invierte([1,2],L).
                              L
[2, 1]                                                1

?- invierte([1,2,3,4,5],L).
                              L
[5, 4, 3, 2, 1]                                       1

?- invierte([1,2],[2,1]).
true                                                  1

?- invierte([1,2],[2,1,3]).
false

?- invierte(L,[1,2]).
                              L
[2, 1]                                                1
```

Ejercicio 3



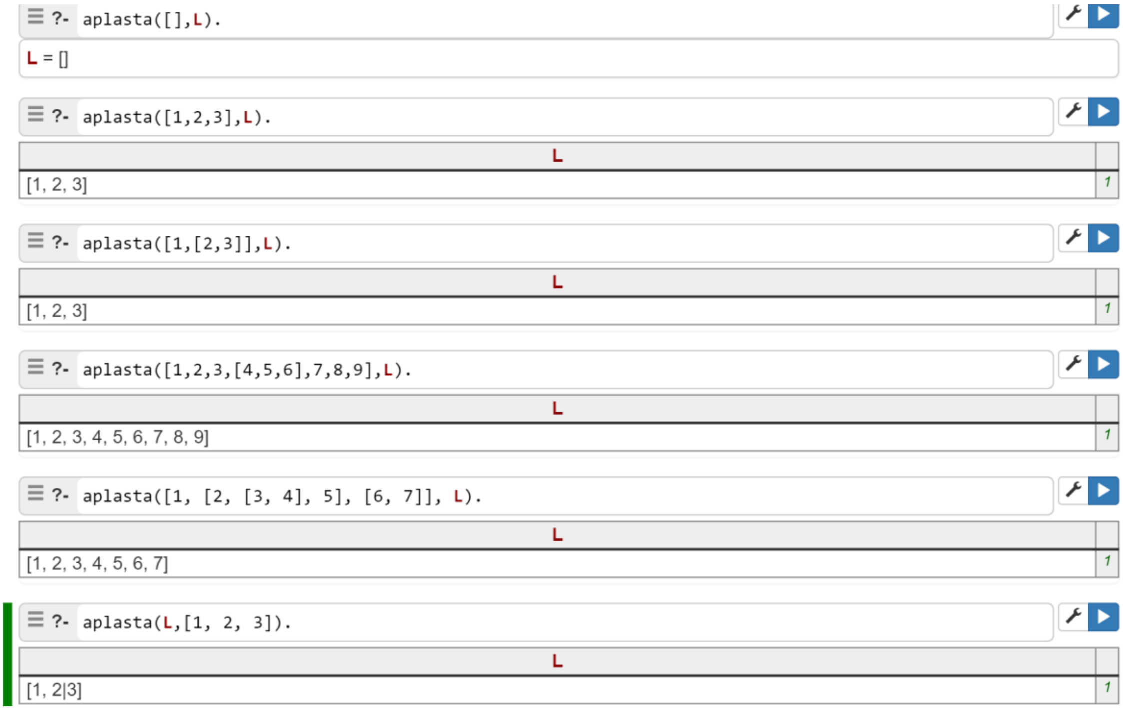
```
?- palindromo([1]).
true                                                  1

?- palindromo([1,2,3]).
false

?- palindromo([]).
true                                                  1

?- palindromo([1,2,1]).
true                                                  1

?- palindromo([1, 2, 1, 1]).
false
```

Ejercicio 4



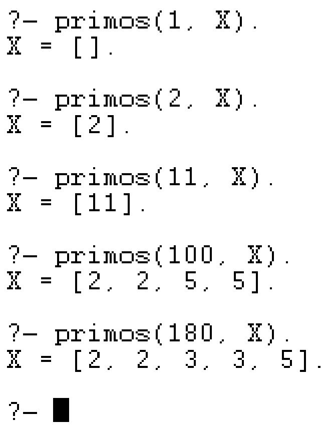
```
?- divide([1,2,3,4],0,[],[1,2,3,4]).
true                                                        1

?- divide([1,2,3,4],4,[1,2,3,4],[]).
true                                                        1

?- divide([1,2,3,4],2,[1,2],[3,4]).
true                                                        1

?- divide([1,2,3,4],0,[1,2,3,4],[1,2,3,4]).
false

?- divide([1, 2, 3, 4, 5], 3, L1, L2).
                L1                          L2
[1, 2, 3]                         [4, 5]                    1

?-  divide(L, 3, [1, 2, 3], [4, 5, 6]).
                              L
[1, 2, 3, 4, 5, 6]                                          1

?- divide([1,2,3,4,5],5,[1,2,3,4,5],[6,7,8]).
false

?- divide([1,2,3,4,5],5,[1,2,3,4,5],[]).
true                                                        1
```

Ejercicio 5



```
?- aplasta([],L).
L = []

?- aplasta([1,2,3],L).
                              L
[1, 2, 3]                                                   1

?- aplasta([1,[2,3]],L).
                              L
[1, 2, 3]                                                   1

?- aplasta([1,2,3,[4,5,6],7,8,9],L).
                              L
[1, 2, 3, 4, 5, 6, 7, 8, 9]                                 1

?- aplasta([1, [2, [3, 4], 5], [6, 7]], L).
                              L
[1, 2, 3, 4, 5, 6, 7]                                       1

?- aplasta(L,[1, 2, 3]).
                              L
[1, 2|3]                                                    1
```

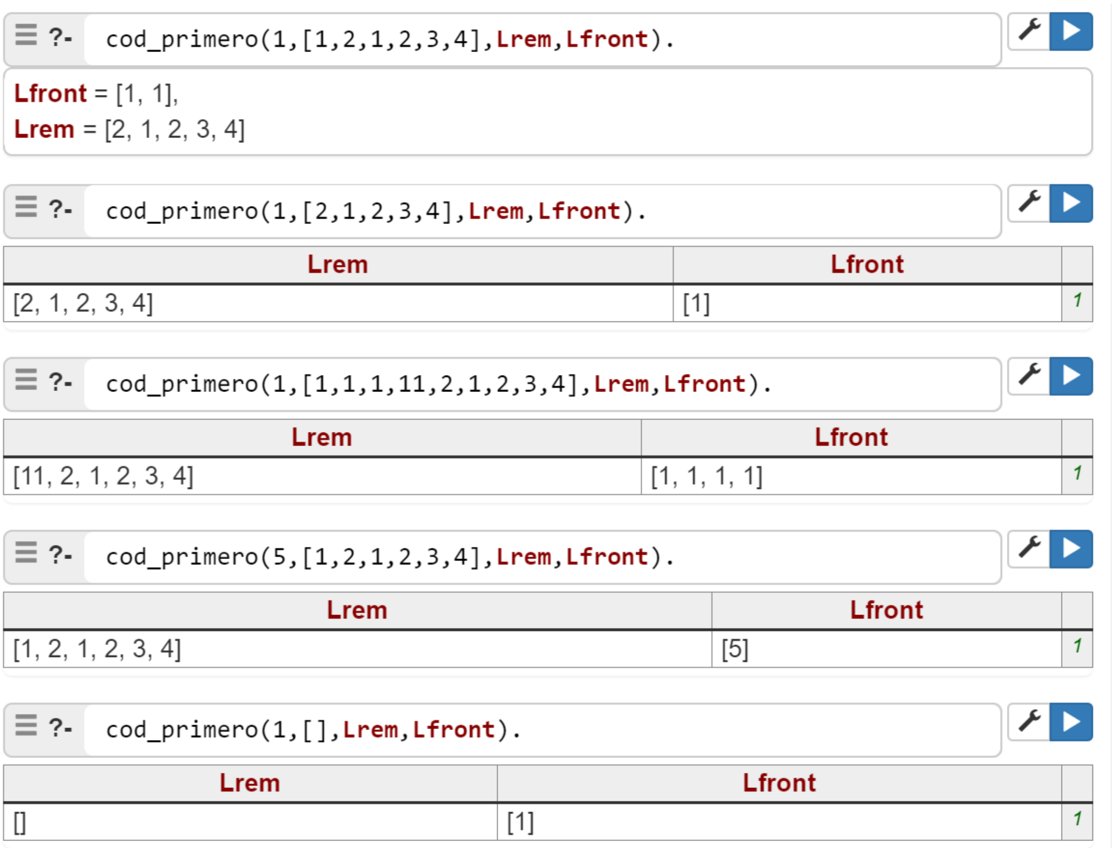
Ejercicio 6



```
?- primos(1, X).
X = [].

?- primos(2, X).
X = [2].

?- primos(11, X).
X = [11].

?- primos(100, X).
X = [2, 2, 5, 5].

?- primos(180, X).
X = [2, 2, 3, 3, 5].

?- █
```

Ejercicio 7

7.1



```
?- cod_primero(1,[1,2,1,2,3,4],Lrem,Lfront).
```

Lfront = [1, 1],
Lrem = [2, 1, 2, 3, 4]

```
?- cod_primero(1,[2,1,2,3,4],Lrem,Lfront).
```

Lrem	Lfront	
[2, 1, 2, 3, 4]	[1]	1

```
?- cod_primero(1,[1,1,1,11,2,1,2,3,4],Lrem,Lfront).
```

Lrem	Lfront	
[11, 2, 1, 2, 3, 4]	[1, 1, 1, 1]	1

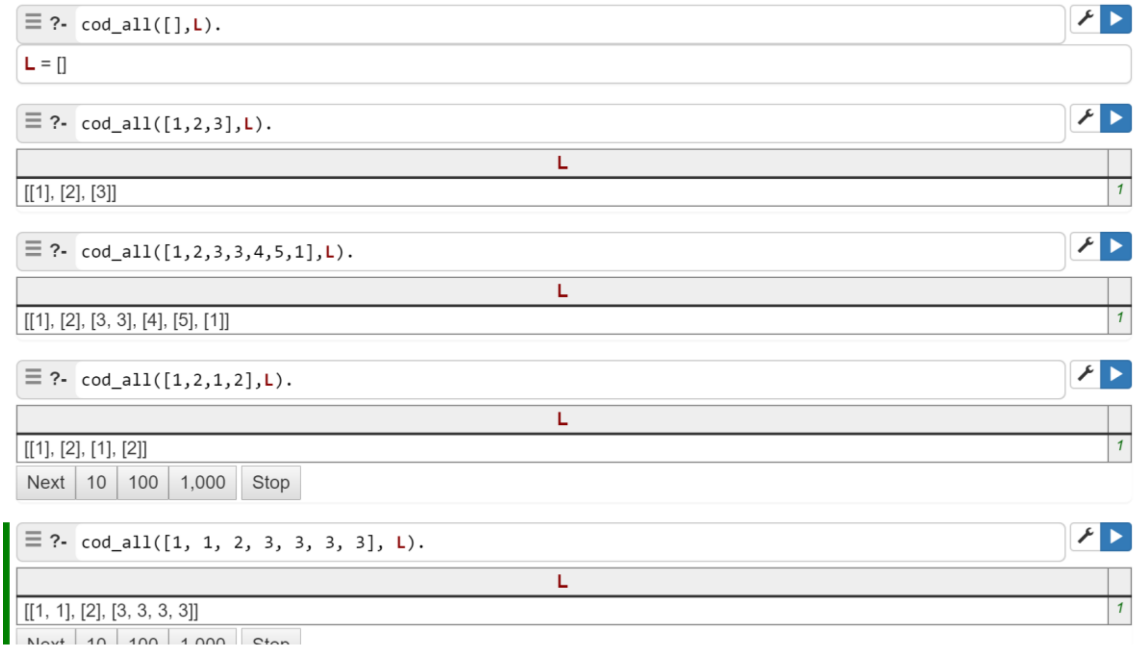
```
?- cod_primero(5,[1,2,1,2,3,4],Lrem,Lfront).
```

Lrem	Lfront	
[1, 2, 1, 2, 3, 4]	[5]	1

```
?- cod_primero(1,[],Lrem,Lfront).
```

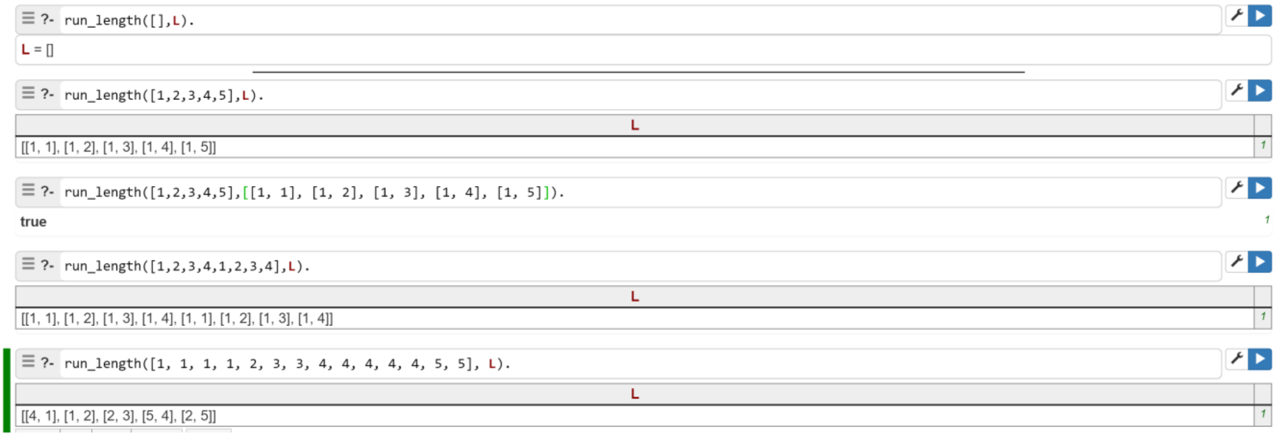
Lrem	Lfront	
[]	[1]	1

7.2



```
?- cod_all([],L).
L = []

?- cod_all([1,2,3],L).
L
[[1], [2], [3]]

?- cod_all([1,2,3,3,4,5,1],L).
L
[[1], [2], [3, 3], [4], [5], [1]]

?- cod_all([1,2,1,2],L).
L
[[1], [2], [1], [2]]

?- cod_all([1, 1, 2, 3, 3, 3, 3], L).
L
[[1, 1], [2], [3, 3, 3, 3]]
```

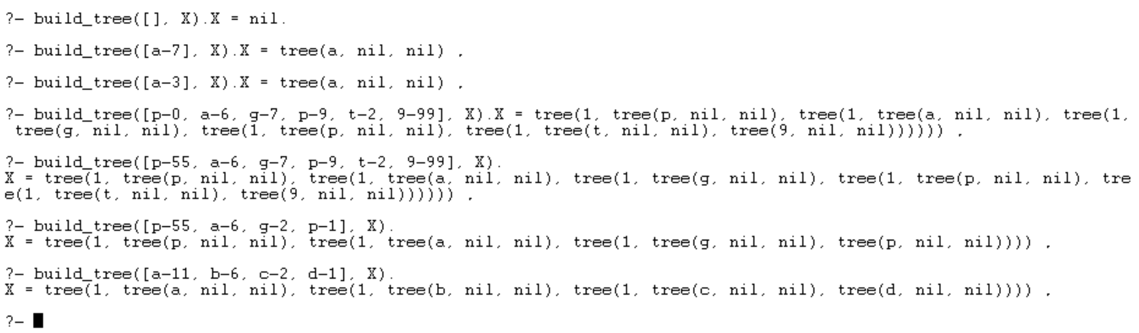
7.3



```
?- run_length([],L).
L = []

?- run_length([1,2,3,4,5],L).
L
[[1, 1], [1, 2], [1, 3], [1, 4], [1, 5]]

?- run_length([1,2,3,4,5],[[1, 1], [1, 2], [1, 3], [1, 4], [1, 5]]).
true

?- run_length([1,2,3,4,1,2,3,4],L).
L
[[1, 1], [1, 2], [1, 3], [1, 4], [1, 1], [1, 2], [1, 3], [1, 4]]

?- run_length([1, 1, 1, 1, 2, 3, 3, 4, 4, 4, 4, 4, 5, 5], L).
L
[[4, 1], [1, 2], [2, 3], [5, 4], [2, 5]]
```

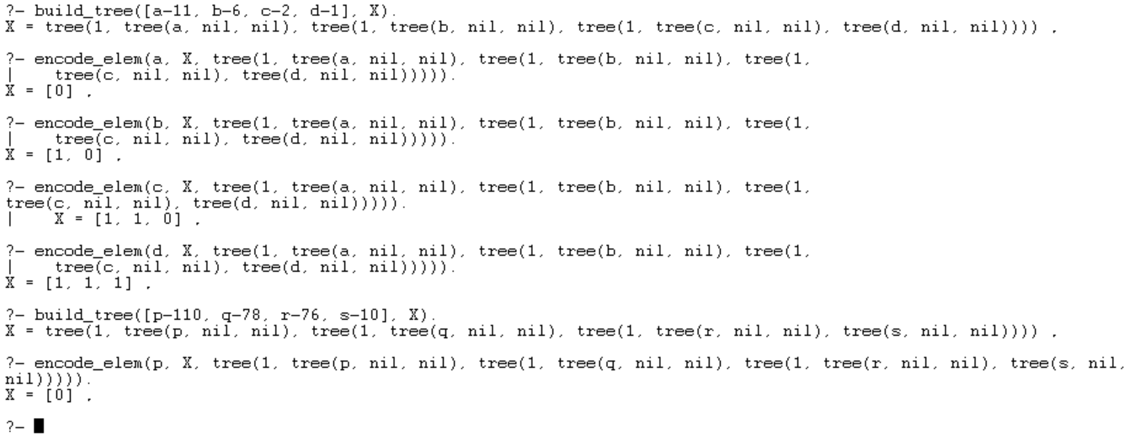
Ejercicio 8

8.0



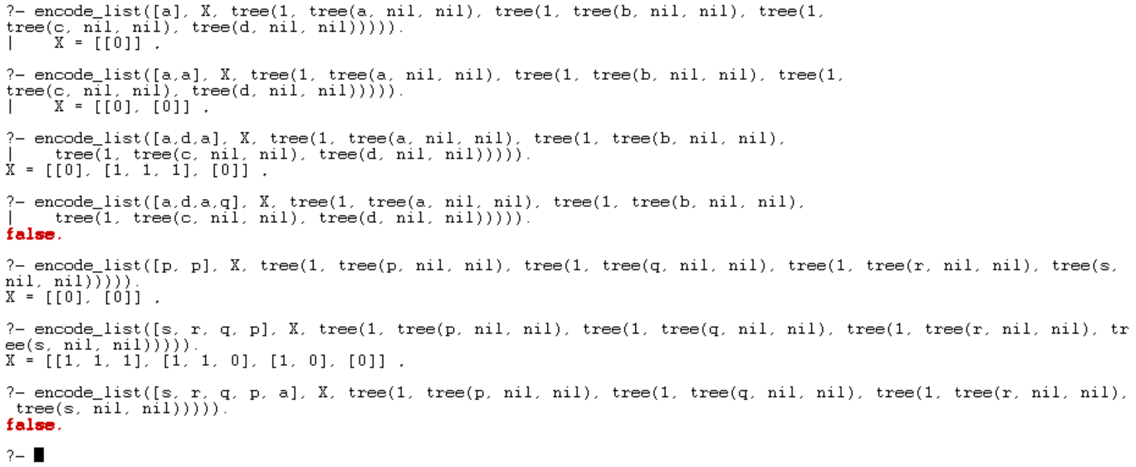
```
?- build_tree([], X).X = nil.

?- build_tree([a-7], X).X = tree(a, nil, nil) .

?- build_tree([a-3], X).X = tree(a, nil, nil) .

?- build_tree([p-0, a-6, g-7, p-9, t-2, 9-99], X).X = tree(1, tree(p, nil, nil), tree(1, tree(a, nil, nil), tree(1,
 tree(g, nil, nil), tree(1, tree(p, nil, nil), tree(1, tree(t, nil, nil), tree(9, nil, nil)))))) .

?- build_tree([p-55, a-6, g-7, p-9, t-2, 9-99], X).
X = tree(1, tree(p, nil, nil), tree(1, tree(a, nil, nil), tree(1, tree(g, nil, nil), tree(1, tree(p, nil, nil), tre
e(1, tree(t, nil, nil), tree(9, nil, nil)))))) .

?- build_tree([p-55, a-6, g-2, p-1], X).
X = tree(1, tree(p, nil, nil), tree(1, tree(a, nil, nil), tree(1, tree(g, nil, nil), tree(p, nil, nil)))) .

?- build_tree([a-11, b-6, c-2, d-1], X).
X = tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1, tree(c, nil, nil), tree(d, nil, nil)))) .

?- █
```

8.1



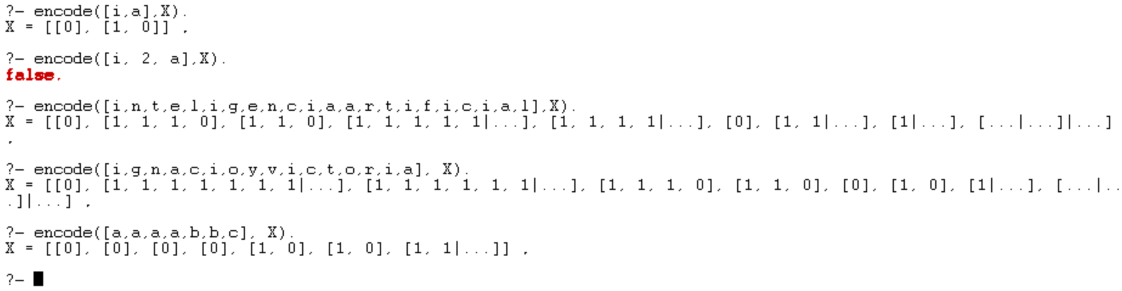
```
?- build_tree([a-11, b-6, c-2, d-1], X).
X = tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1, tree(c, nil, nil), tree(d, nil, nil)))) .

?- encode_elem(a, X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1,
|    tree(c, nil, nil), tree(d, nil, nil))))).
X = [0] .

?- encode_elem(b, X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1,
|    tree(c, nil, nil), tree(d, nil, nil))))).
X = [1, 0] .

?- encode_elem(c, X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1,
tree(c, nil, nil), tree(d, nil, nil))))).
|    X = [1, 1, 0] .

?- encode_elem(d, X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1,
|    tree(c, nil, nil), tree(d, nil, nil))))).
X = [1, 1, 1] .

?- build_tree([p-110, q-78, r-76, s-10], X).
X = tree(1, tree(p, nil, nil), tree(1, tree(q, nil, nil), tree(1, tree(r, nil, nil), tree(s, nil, nil)))) .

?- encode_elem(p, X, tree(1, tree(p, nil, nil), tree(1, tree(q, nil, nil), tree(1, tree(r, nil, nil), tree(s, nil,
nil))))).
X = [0] .

?- █
```

8.2



```
?- encode_list([a], X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1,
tree(c, nil, nil), tree(d, nil, nil))))).
|    X = [[0]] .

?- encode_list([a,a], X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil), tree(1,
tree(c, nil, nil), tree(d, nil, nil))))).
|    X = [[0], [0]] .

?- encode_list([a,d,a], X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil),
|    tree(1, tree(c, nil, nil), tree(d, nil, nil))))).
X = [[0], [1, 1, 1], [0]] .

?- encode_list([a,d,a,q], X, tree(1, tree(a, nil, nil), tree(1, tree(b, nil, nil),
|    tree(1, tree(c, nil, nil), tree(d, nil, nil))))).
false.

?- encode_list([p, p], X, tree(1, tree(p, nil, nil), tree(1, tree(q, nil, nil), tree(1, tree(r, nil, nil), tree(s,
nil, nil))))).
X = [[0], [0]] .

?- encode_list([s, r, q, p], X, tree(1, tree(p, nil, nil), tree(1, tree(q, nil, nil), tree(1, tree(r, nil, nil), tr
ee(s, nil, nil))))).
X = [[1, 1, 1], [1, 1, 0], [1, 0], [0]] .

?- encode_list([s, r, q, p, a], X, tree(1, tree(p, nil, nil), tree(1, tree(q, nil, nil), tree(1, tree(r, nil, nil),
 tree(s, nil, nil))))).
false.

?- █
```

8.3



```
?- encode([i,a],X).
X = [[0], [1, 0]] .

?- encode([i, 2, a],X).
false.

?- encode([i,n,t,e,l,i,g,e,n,c,i,a,a,r,t,i,f,i,c,i,a,l],X).
X = [[0], [1, 1, 1, 0], [1, 1, 0], [1, 1, 1, 1, 1|...], [1, 1, 1, 1|...], [0], [1, 1|...], [1|...], [...|...]|...]
.

?- encode([i,g,n,a,c,i,o,y,v,i,c,t,o,r,i,a], X).
X = [[0], [1, 1, 1, 1, 1, 1, 1|...], [1, 1, 1, 1, 1, 1|...], [1, 1, 1, 0], [1, 1, 0], [0], [1, 0], [1|...], [...|..
.]|...] .

?- encode([a,a,a,a,b,b,c], X).
X = [[0], [0], [0], [0], [1, 0], [1, 0], [1, 1|...]] .

?- █
```