

Lista de ejercicios del 2º. Parcial

1. Definir la relación `trasladar(L1, L2)` que permita trasladar una lista de números `L1` a una lista de sus correspondientes nombres `L2`.
`?- trasladar([1,2,3,7],[uno,dos,tres,siete]).`
`True`
2. Definir dos predicados: `par(Lista)`, `impar(Lista)` que serán verdaderos si `Lista` tiene un número par e impar de elementos, respectivamente.
`?- par([1,2,3,7]).`
`True`
3. Definir la relación `aplanar(L, Lplana)` en donde `L`, que puede ser una lista de listas, pase a ser el conjunto de todos los elementos de `L` pero en una lista plana en `Lplana`.
`?- aplanar([a,b,[c,d],[],[[e]]],f),L).`
`L=[a,b,c,d,e,f]`
4. Defina la relación `eliminaRep(Lista, ListaR)` que elimine los elementos repetidos consecutivos de `Lista`.
`?- eliminaRep([a,a,a,a,b,c,c,a,a,d,e,e,e,e], LR).`
`LR=[a,b,c,a,d,e]`
5. Defina una relación que clasifique los elementos duplicados consecutivos de una lista, en una lista de duplas `[Cantidad,Elem]`.
`?- clasificaRep([a,a,a,a,b,c,c,a,a,d,e,e,e,e], LR).`
`LR=[[4,a],[1,b],[2,c],[2,a],[1,d],[4,e]]`
6. Definir el predicado `ordenada(L)` que se verifique si la lista de números `L` está ordenada de manera decreciente.
7. Definir la relación `multiplicada(L, N, LR)` que se verifica si `LR` es la lista obtenida al repetir `N` veces los elementos de la lista `L`.
`?- multiplicada([a,b,c],3,LR).`
`LR=[a,a,a,b,b,b,c,c,c]`
8. Defina una relación que borre todos los `i`-esimos elementos de una lista.
`?- borraiesimo([a,b,c,d,e,f,g,h,i,k],3,LR).`
`LR=[a,b,d,e,g,h,k]`
9. Defina la relación `sustituye(L1, E1, E2, LR)` que sustituya todas las ocurrencias de `E1` en `L1`, por elemento `E2`.

```
?- sustituye([a,b,c,a,e,a,g,a,i],a,f,LR) .  
LR = [f,b,c,f,e,f,g,f,i]
```

10. Defina la relación `cuentavocales(L1,NumV)` que cuenta el número de vocales que existen en `L1`.

```
?-cuentavocales([a,b,c,a,e,a,g,a,i],N) .  
N = 6
```

11. Defina la relación `factorial(N,NumF)` que genere el factorial del número `N`.

```
?-factorial(3,N) .  
N = 6
```

12. Defina la relación `crecimiento(L1,L2)` que verifique si `L2` es la lista correspondiente a los crecimientos de la lista `L1`. Es decir, entre cada par de elementos consecutivos `X` e `Y` de `L1`, la relación colocará el signo `+`, si $X < Y$, y signo `-`, en caso contrario.

```
?-crecimiento([1,3,2,2,5,3],L2) .  
L2 =[1,+,3,-,2,-,2,+,5,-,3]
```

13. Defina la relación `sumadigitos(Num,Sum)` que sume los dígitos del número `Num`.

```
?-sumadigitos(237,Sum) .  
Sum =12
```

14. Defina la relación `suma_posiciones(Lista,N,Suma)` que verifique si `Suma` es la suma de los elementos de `Lista` que ocupan las posiciones que son múltiplos de `N`.

```
?-suma_posiciones([3,5,7,9,1,2],2,Suma) .  
Suma =16
```

15. Defina la relación `decodificada(L1,L2)` que encuentre una lista `L2` cuya codificación reducida por longitud es `L1`.

```
?-decodificada([a,2-b,3-a,c,3-b],L2) .  
L2 =[a,b,b,a,a,a,c,b,b,b]
```

16. Defina la relación `cota_superior(L,N)` que verifique si `N` es una cota superior para `L` (es decir, todos los elementos en `L` son menores o iguales que `N`).

```
?-cota_superior([1,5,3],7) .  
True
```