

Escreva programas em Prolog para representar as seguintes relações:

1. Defina a relação `soma(X,S)` tal que `S` é a soma dos inteiros da lista de inteiros `X`.
2. `prefixo(X,Y)`: `X` é um prefixo (parte inicial) da lista `Y`.  
Por exemplo, `prefixo([1,2],[1,2,3])` é verdadeiro, mas `prefixo([1,3],[1,2,3])` é falso.
3. `sufixo(X,Y)`: `X` é um sufixo (parte final) da lista `Y`.  
Por exemplo, `sufixo([2,3],[1,2,3])`, mas `sufixo([2],[1,2,3])` é falso.
4. Verificar se uma lista `T` é sublista de outra `U`. Uma lista `T` é sublista de outra `U` se os elementos de `T` ocorrem em `U`, na mesma ordem. Por exemplo, `[a,b,c,d]` é uma sublista de `[p,a,b,t,c,e,d,w]`, mas não de `[d,c,b,a]`.
5. Verificar se uma lista `T` é uma subsequência de outra `U`. Uma lista é uma subsequência de outra `U` se `T` ocorre como uma sequência de elementos consecutivos em `U`. Por exemplo, `[a,b,c,d]` é uma subsequência of `[x,a,b,c,d,e]`, mas não de `[a,b,x,c,d]`.
6. `quebraNoMeio(X,X1,X2)`: `X1` e `X2` são a metade esquerda e a metade direita da lista `X`; se `X` tiver um número ímpar de elementos, `X1` tem um elemento a mais que `X2`.