

1. Declare uma função que retorne a união entre duas listas.
2. Declare uma função que retorne a interseção entre duas listas
3. Declare uma função que retorne os n últimos elementos de uma lista.
4. Declare uma função que retorne a exponenciação de dois números naturais.
5. Declare uma função que converta um número binário (representado como uma *String*) em inteiro.
6. Declare uma função que retorne o menor valor de uma lista.
7. Usando a declaração anterior, declare uma função que ordene os elementos de uma lista.
8. Declare uma função que receba um elemento e uma lista ordenada insira este elemento na lista colocando-o na posição correta, ou seja, a lista resultante deve estar ordenada. Se o elemento já pertencer à lista, ele não deve ser incluído.
9. Declare uma função que receba um inteiro e retorne a *True* se o inteiro for par e *False* se for ímpar (considere zero como um inteiro par).
10. Declare uma função que receba como parâmetro um predicado e uma lista, a função deve retornar uma lista contendo apenas os elementos que satisfazem ao predicado. Um predicado é uma função que recebe apenas um argumento e retorna a um valor booleano. Por exemplo, a função declarada na questão anterior é um predicado.

O tipo dessa função deve ser: $(\alpha \rightarrow Bool) \rightarrow [\alpha] \rightarrow [\alpha]$

11. Usando as duas declarações anteriores declare uma função que receba uma lista e retorne apenas seus elementos ímpares.
12. Declare uma função que receba como argumento uma lista de duplas e retorne uma lista contendo os primeiros elementos dessas duplas (use a função *map*).
13. Declare uma função que receba uma lista de valores booleanos e retorne a *True* se todos os elementos forem *True*, a função deve retornar a *False* em caso contrário (use uma das versões da função *fold*).
14. Declare uma função que receba uma lista de números e retorne o somatório dos seus elementos (use uma das versões da função *fold*).

15. Declare uma função que receba duas listas e retorne uma lista de duplas, onde o primeiro elemento da dupla pertence a primeira lista e o segundo elemento da dupla pertence a segunda.
16. Declare uma lista infinita de números primos, use o crivo de Eratóstenes.
17. Declare uma função que retorne os n primeiros números primos.
18. Declare uma função que retorne uma lista infinita com a sequência de Fibonacci.
19. Declare uma função que retorne os n primeiros números da sequência de Fibonacci.
20. Declare uma função para a inserção de elementos em uma árvore binária de pesquisa.
21. Declare uma função que remova um elemento de uma árvore binária.
22. Declare uma função que converta uma árvore binária em uma lista, a lista deve ser criada considerando o percurso na árvore *in-order*.
23. Dada a declaração do tipo de dado *Nat*, que representa os números naturais declare uma função para adição, multiplicação e exponenciação de dois números naturais representados por este tipo de dado.

data Nat = Zero | Suc Nat