

1) O que faz a seguinte função?

```
(defun xxx (x)
  (+ 1 x))
```

2) Suponha que tenha sido feita a definição de função acima.  
Qual o valor das seguintes expressões?

- a. (xxx 2)
- b. (xxx (+ (xxx 5) 3))
- c. (+ 4 xxx)

3) Suponha que além da definição de função acima façamos ainda

```
(setf xxx 5)
```

Qual o valor da expressão abaixo?

```
(xxx xxx)
```

- 4) Escreva em lisp uma função fatorial **recursiva**, utilizando apenas os operadores aritméticos básicos (+,-,\*,/) , comandos condicionais e predicados vistos em sala de aula.
- 5) Até ao momento, a linguagem nanoLisp apenas trabalha com números inteiros positivos. Admitindo que as operações 1+, 1- e zerop também funcionam com números negativos, defina a função `negativo` que recebe um número inteiro positivo e retorna o seu simétrico. Assim, pretendemos obter: `(negativo 3) => -3`.
- 6) É possível definir a soma de dois números inteiros positivos em nanoLisp , i.e., apenas recorrendo às funções 1+ e 1- que somam e subtraem uma unidade, respectivamente. Defina a operação soma.
- 7) Do mesmo modo que a soma pode ser definida exclusivamente em termos de sucessor 1+ e predecessor 1-, a multiplicação pode ser definida exclusivamente em termos da soma. Defina a função `mult` que recebe dois número e os multiplica usando a função `soma`.