P4 de ICC

1. (2.5) Escreva uma função `cria-lista` que receba n, f, e x_0, e retorne a lista

$$x_0, f(x_0), f(f(x_0)), \ldots$$

com n elementos. Por exemplo,

`(cria-lista 4 (lambda (x) (* x x)) 2)` $\leadsto$ `(2 4 16 256)`

`(cria-lista 6 (lambda (x) (+ x 1)) 0)` $\leadsto$ `(0 1 2 3 4 5)`

2. (2.5) Escreva, usando `do`, um procedimento `substitui` que substitua todas as ocorrências de um dado símbolo em um vetor por outro símbolo; por exemplo, se v é o vetor `'#(um rato maior que outro rato)`, após a chamada `(substitui 'rato 'gato v)` v será `'#(um gato maior que outro gato)`.

3. (2.5) Escreva uma função que receba uma árvore binária, representada na forma usual (pré-fixada), e retorne uma lista com todos os elementos da árvore (em qualquer ordem); por exemplo:

`(lista '(5 (3 4 ()) (10 13 12)))` $\leadsto$ `(4 3 5 13 10 12)`

(DICA: se necessário, use a função pré-definida `append` para concatenar duas listas).

4. Considere uma data representada por uma lista de três elementos, `(dia mes ano)`. Por exemplo, o dia de hoje seria representado pela lista `(28 6 1997)`.

   (a) (0.5) Escreva três funções, `dia`, `mes` e `ano`, que recebam uma data e retornem, respectivamente, o dia, o mês e o ano correspondentes; por exemplo:

   `(mes '(4 5 1996))` $\leadsto$ `5`

   (b) (1.0) Escreva um predicado que receba duas datas e verifique se uma é anterior a outra. Por exemplo:

   `(ant '(29 6 1996) '(28 6 1997))` $\leadsto$ `#t`

   `(ant '(29 6 1997) '(28 6 1997))` $\leadsto$ `#f`

   `(ant '(29 6 1997) '(28 8 1997))` $\leadsto$ `#t`

   (c) (1.0) Assumindo a existência de um vetor com os nomes de todos os meses, conforme abaixo:

   `(define meses '#("janeiro" "fevereiro" ... "dezembro"))`

   escreva um procedimento que receba uma data e a imprima no formato usual, conforme o exemplo abaixo:

   `(mostra '(3 2 1990))` $\xrightarrow{mostra}$ $\boxed{\texttt{3 de fevereiro de 1997}}$