P3 de ICC

1. (2,5) Escreva, usando `do`, um procedimento `mostra` para imprimir os elementos de um dado vetor, um por linha, precedidos por seus índices.

2. (2,5) Escreva, usando `do`, um procedimento `para-todo` que receba um vetor v e uma função f, e aplique a função f sobre cada elemento do vetor, substituindo-o pelo resultado da função. Por exemplo, se v for `#(9 4 25 1)`, após a chamada `(para-todo v sqrt)` v será `#(3 2 5 1)`.

3. (a) (1,0) Escreva um procedimento `troca` que receba um vetor e dois índices, e troque os valores nas posições dos índices um pelo outro.

   (b) (1,5) Escreva um procedimento `inverte` que receba um vetor e inverta a ordem de seus elementos. Por exemplo, se v for `#(1 2 3 4)`, após a chamada `(inverte v)` v será `#(4 3 2 1)`. Você pode usar o procedimento `troca`, mesmo sem ter resolvido o ítem anterior.

4. (2,5) Considere um vetor onde cada elemento é uma lista com um símbolo e um número, como por exemplo:

```
(define v #((r 1) (o 2) (v 5) (p 0) (c 7) (a -1) (i 4) (c -1)))
```

   Considere que o número em cada posição do vetor é usado para indicar uma próxima posição, formando um *encadeamento*. Por exemplo, na posição 3 de `v` temos a lista `(p 0)`, indicando que a próxma posição no encadeamento é a posição 0. Na posição 0 temos `(r 1)`, indicando o 1 como a próxima posição. O valor -1 em uma posição indica o fim de uma cadeia.

   Escreva uma função `cadeia` que, dados um vetor como descrito acima, e uma posição inicial, retorne a lista com os símbolos contidos nas posições encadeadas. Por exemplo, com `v` declarado como acima, teriamos:

   `(cadeia v 3)` $\rightsquigarrow$ `(p r o v a)`

   `(cadeia v 6)` $\rightsquigarrow$ `(i c c)`