

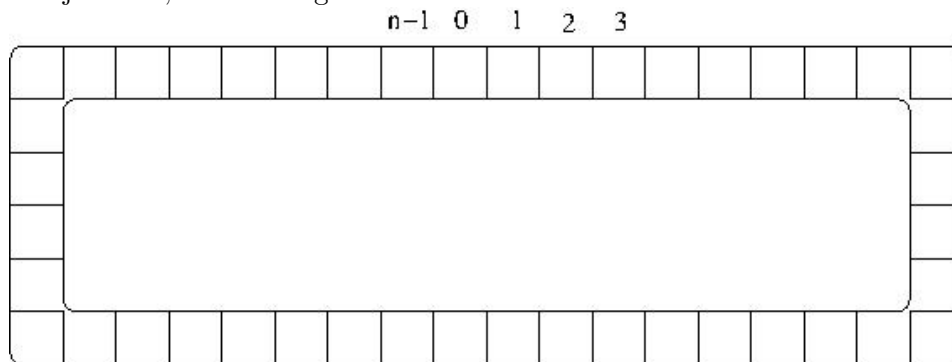
## MAC 121 – Algoritmos e Estruturas de Dados I

Segundo semestre de 2016

### Ordenação – Entrega: 17 de outubro de 2016

O objetivo deste exercício-programa é desenvolver algoritmos de ordenação e analisar sua complexidade.

Considere inicialmente um vetor “circular”, ou seja, em que as posições de índice 0 e  $n-1$  sejam considerados adjacentes, como na figura abaixo.



Agora, considere que a única operação disponível para modificar o conteúdo dos elementos do vetor seja uma *3-reversão*. Por exemplo, para o vetor:

0	1	2	3	4	5	6	7	8	9	10	11
23	12	-7	5	8	13	3	93	1	-12	41	81

Aplicando uma 3-reversão no vetor acima na posição de índice 5 obtemos:

0	1	2	3	4	5	6	7	8	9	10	11
23	12	-7	5	8	<b>93</b>	<b>3</b>	<b>13</b>	1	-12	41	81

Aplicando no vetor original uma 3-reversão na posição de índice 10 (lembrando que consideramos o vetor “circular”) obtemos:

0	1	2	3	4	5	6	7	8	9	10	11
<b>41</b>	12	-7	5	8	13	3	93	1	-12	<b>23</b>	<b>81</b>

Neste EP vocês deverão fazer um programa que leia  $n > 0$ , um vetor com  $n$  inteiros, e imprima uma série de 3-rotações que ordene este vetor em ordem crescente.

As perguntas que você deverá considerar em seu relatório são:

- é possível ordenar um vetor de qualquer tamanho com 3-rotações?
- dado um vetor de tamanho  $n$  é possível ordenar qualquer instância com 3-rotações?
- qual o número máximo de 3-rotações que seu algoritmo executa para ordenar um vetor?

## Exemplo de entrada

```
7
3 4 2 6 5 1 7
```

## Exemplo de saída

```
0
5
3
1
```

Ou seja, fazendo as 3-rotações nas posições de índices 0, 5, 3 e 1 obtemos o vetor ordenado.

## Observações

- Este programa é individual. Discuta com seus colegas, mas não compartilhe código.
- Seu programa deverá ser claro, ter boa endentação, bons comentários, variáveis com nomes mnemônicos, ter um laiaute consistente, etc. Tudo isso será levado em conta na nota.
- Faça vários testes, documente suas conclusões. Entregue um relatório junto com o EP, como feito nos anteriores, para análise do monitor.