

## SCC0502 – Algoritmos e Estruturas de Dados I

Prof. Dr. Renato Moraes Silva

## Lista de Exercícios 1

**Exerc. 1.** Considere que  $S$  seja uma pilha estática com tamanho 4. Considerando as operações mostradas abaixo, apresente a sequência de saídas, os valores em cada posição do *array* e a posição e valor do elemento que está no topo da pilha.

- Se ocorrer erro de *overflow* ou *underflow*, considere que a estrutura exibirá uma mensagem de erro, mas não sofrerá nenhuma alteração. As operações seguintes irão ocorrer normalmente considerando o estado da pilha antes do erro.

```
1 push(S, 5)
2 push(S, 3)
3 pop(S)
4 push(S, 2)
5 push(S, 8)
6 pop(S)
7 pop(S)
8 push(S, 9)
9 push(S, 1)
10 pop(S)
11 push(S, 7)
12 push(S, 6)
13 pop(S)
14 pop(S)
15 push(S, 4)
16 pop(S)
17 pop(S)
```

**Exerc. 2.** Considere que  $Q$  seja uma fila estática com tamanho 4. Considerando as operações mostradas abaixo, apresente a sequência de saídas, os valores finais da fila em sequência, os valores em cada posição do *array* que guarda a fila, além da posição e valor dos elementos *front* e *rear*.

- Se ocorrer erro de *overflow* ou *underflow*, considere que a estrutura exibirá uma mensagem de erro, mas não sofrerá nenhuma alteração. As operações seguintes irão ocorrer normalmente considerando o estado da fila antes do erro.

```
1 enqueue(Q, 5)
2 enqueue(Q, 3)
3 dequeue(Q)
4 enqueue(Q, 2)
5 enqueue(Q, 8)
```

```

6 dequeue(Q)
7 dequeue(Q)
8 enqueue(Q, 9)
9 enqueue(Q, 1)
10 dequeue(Q)
11 enqueue(Q, 7)
12 enqueue(Q, 6)
13 dequeue(Q)
14 dequeue(Q)
15 enqueue(Q, 4)
16 dequeue(Q)
17 dequeue(Q)

```

**Exerc. 3.** Faça a mesma coisa do exercício anterior, considerando a mesma sequência de operações. Porém, considere que a fila é estática e circular.

**Exerc. 4.** Considere que Q seja um deque estático circular com tamanho 4. Considerando as operações mostradas abaixo, apresente a sequência de saídas, os valores finais do deque em sequência, os valores em cada posição do *array* que guarda o deque, além da posição e valor dos elementos *front* e *rear*.

- Caso ocorra erro de *overflow* ou *underflow*, considere que a estrutura exibirá uma mensagem de erro, mas não sofrerá nenhuma alteração. As operações seguintes irão ocorrer normalmente considerando o estado do deque antes do erro.

```

1 enqueueFront(Q, 5)
2 enqueueFront(Q, 3)
3 dequeueRear(Q)
4 enqueueRear(Q, 2)
5 enqueueRear(Q, 8)
6 dequeueFront(Q)
7 dequeueFront(Q)
8 enqueueRear(Q, 9)
9 enqueueRear(Q, 1)
10 dequeueFront(Q)
11 enqueueFront(Q, 7)
12 enqueueRear(Q, 6)
13 dequeueFront(Q)
14 enqueueRear(Q, 4)
15 dequeueFront(Q)

```

**Exerc. 5.** Considere o pseudocódigo da função F apresentada abaixo. Considere que a função receba como entrada o número 6 e uma pilha vazia. Qual será a pilha final retornada por essa função?

---

```

1: função F(numero, pilha)
2:   se numero ≤ 2 então
3:     retorna pilha
4:   fim se
5:   se numero % 3 = 0 and isEmpty(pilha) = false então
6:     aux ← pop(pilha)
7:   senão
8:     se isEmpty(pilha) = true então:
9:       aux ← 0
10:      push(pilha, aux)
11:     senão
12:       aux ← numero - top(pilha)
13:       push(pilha, aux)
14:     fim se
15:   fim se
16:   retorna F(numero - 1, pilha)
17: fim função

```

---

**Exerc. 6.** Considere uma pilha P vazia e uma fila F não vazia. Utilizando apenas as funções da fila e da pilha, escreva o pseudocódigo de uma função que inverta a ordem dos elementos da fila.

**Exerc. 7.** Escreva um programa em C contendo os itens descritos abaixo.

- Crie uma estrutura chamada **Livro** que contenha as seguintes informações:
  - **titulo**: uma string (array de char) de tamanho 100 que armazena o título do livro.
  - **autor**: uma string (array de char) de tamanho 50 que armazena o nome do autor.
  - **anoPublicacao**: um inteiro que armazena o ano de publicação do livro.
  - **numeroPaginas**: um inteiro que armazena o número de páginas do livro.
- Implemente uma função **criarLivro** que cria e retorna uma nova instância de **Livro** com os valores fornecidos para **titulo**, **autor**, **anoPublicacao** e **numeroPaginas**.
- Implemente uma função **atualizarNumeroPaginas** que atualiza o número de páginas do livro para um novo valor fornecido.
- Implemente uma função **adicionarSubtitulo** que adiciona um subtítulo ao título do livro. O subtítulo deve ser concatenado ao título original, separado por dois-pontos (":").
- Implemente uma função **imprimirLivro** que imprime todas as informações do livro, incluindo título (com subtítulo, se houver), autor, ano de publicação e número de páginas.
- Escreva um programa em C que utilize a estrutura **Livro** e as funções associadas para:
  - Criar um vetor dinâmico de **Livro** e permitir ao usuário adicionar múltiplos livros a esse vetor.
  - Atualizar o número de páginas de um livro específico no vetor.
  - Adicionar um subtítulo a um livro específico no vetor.
  - Imprimir as informações completas de todos os livros no vetor, incluindo títulos (com subtítulos, se houver), autores, anos de publicação e número de páginas.

**Exerc. 8.** Escreva um programa em C que implemente um pilha estática contendo as operações tradicionais: push, pop, isEmpty, isFull, top, size e clear.