



Instruções para entrega do roteiro:

- Entregue o roteiro apenas no formato *.pdf* com o nome ***Y_roteiroX.pdf***, onde **X** é o número do roteiro e **Y** é o número da sua matrícula. Não serão aceitos outros formatos.
- Inclua nome e matrícula, e mantenha a resolução dos exercícios **ordenada** e **legível**.
- Códigos completos (com `int main`), compiláveis e executáveis, quando aplicável.
Para cada um, apresente uma imagem da tela de saída do seu programa.
- Após a data de entrega, a nota da entrega é 0.
- Em caso de dúvidas, procurem os monitores. Haverá um monitor após as aulas de laboratório para tirar dúvidas sobre a lista.

Roteiro 5

TAD: Filas e Pilhas

Data máxima de entrega: 06/10/2023
(Entrega: pelo SIGAA, na sua turma de laboratório.)

1 Fila

1.1 Reimplemente o **TAD: Fila Sequencial Estática** visto em aula e teste cada uma das suas operações. Crie uma *Main.c* com um menu (`do..while`) com as seguintes opções, chamando as operações:

- 1 - Criar fila;
- 2 - Enfileirar um item;
- 3 - Ver o início da fila.
- 4 - Desenfileirar um item;
- 5 - Imprimir a fila;
- 6 - Destruir a fila;
- 7 - Sair;

- 1.2 Considerando a struct abaixo, faça uma implementação de um **TAD: Fila Simplesmente Encadeada** com as mesmas operações e que use uma estrutura similar à uma *Lista Simplesmente Encadeada*:

```
1 #ifndef FSE_H
2 #define FSE_H
3 //Bibliotecas
4 typedef struct NO{
5     int info;
6     struct NO* prox;
7 }NO;
8 typedef struct{
9     int qtd;
10    struct NO* ini;
11    struct NO* fim;
12 }Fila;
13 #endif
```

Consulte o livro de Prof. André Backes (*Algoritmos e Estruturas de Dados em Linguagem C*), Capítulo 6, Página 206. Acesso online:

<http://www.biblioteca.ufsj.edu.br/pergamum/biblioteca/index.php>

2 Pilha

- 2.1 Reimplemente o **TAD: Pilha Sequencial Estática** visto em aula e teste cada uma das suas operações. Crie uma *Main.c* com um menu (do..while) com as seguintes opções, chamando as operações:

- 1 - Criar pilha;
- 2 - Empilhar um item;
- 3 - Ver o topo da pilha.
- 4 - Desempilhar um item;
- 5 - Imprimir a pilha;
- 6 - Destruir a pilha;
- 7 - Sair;

- 2.2 Considerando a struct abaixo, faça uma implementação de um **TAD: Pilha Simplesmente Encadeada** com as mesmas operações e que use uma estrutura similar à uma *Lista Simplesmente Encadeada*:

```
1 #ifndef PSE_H
2 #define PSE_H
3 //Bibliotecas
4 typedef struct NO{
5     int info;
6     struct NO* prox;
7 }NO;
8 typedef struct{
9     int qtd;
10    struct NO* topo;
11 }Pilha;
12 #endif
```

Consulte o livro de Prof. André Backes (*Algoritmos e Estruturas de Dados em Linguagem C*), Capítulo 9, Página 283. Acesso online:

<http://www.biblioteca.ufsj.edu.br/pergamum/biblioteca/index.php>