SCC-502 Algoritmos e Estruturas de Dados 1 **M. Cristina/Jorge**

Lista de Exercícios 2 - Filas

A lista de exercícios deverá ser feita individualmente e enviada para o e-mail "jorgehpopae@gmail.com" com o assunto "LISTA_2 - <PRIMEIRONOME>_<NºUSP>", por exemplo, "LISTA_2 – Jorge_9999999". O e-mail deve conter no anexo apenas 1 arquivo compactado, "LISTA2_<NOME>_<NºUSP>.zip", por exemplo, "LISTA2_Jorge_9999999.zip".

Atenção: Entregar **apenas** os exercícios **4** e **8** no arquivo compactado.

Neste arquivo, devem estar os exercícios resolvidos em arquivos separados, com nomes "exercicio4.c", "exercicio8.c". A data limite para entrega é **30/09**.

1) Dada uma fila de inteiros, escreva um programa que exclua todos os números negativos sem alterar a posição dos outros elementos da fila. Utilize o protótipo abaixo:
void remove_negativo(Fila *F);

2) Faça uma função que receba três filas, duas já preenchidas em ordem crescente e preencha a última com os valores das duas primeiras em ordem crescente. Utilize o protótipo abaixo:
void preenche (Fila *F1, Fila *F2, Fila *Retorno);

3) Considere uma pilha P vazia e uma fila F não vazia. Utilizando apenas os testes de fila e pilha vazias, as operações Enfileira, Desenfileira, Empilha, Desempilha, e uma variável auxdo TipoItem, escreva uma função que inverta a ordem dos elementos da fila.

4) Considere a implementação de filas usando **vetores "circulares".** Escreva uma função FuraFila(TipoFila* pFila, TipoItem x) que insere um item na primeira posição da fila. O detalhe é que seu procedimento deve ser O(1), ou seja, não pode movimentar os outros itens da fila. **(observe que este neste caso, estaremos desrepeitando o conceito de FILA – primeiro a entrar é o primeiro a sair)**. **Declare também a estrutura de fila com vetor circular.**

5) Existem partes de sistemas operacionais que cuidam da ordem em que os programas devem ser executados. Por exemplo, em um sistema de computação de tempo-compartilhadao ("time-shared") existe a necessidade de manter um conjunto de processos em uma fila, esperando para serem executados. Escreva três funções, uma para cada operação abaixo:

a) Incluir novos processos na fila de processo;
b) Retirar da fila o processo com o maior tempo de espera;
c) Imprimir o conteúdo da lista de processo em determinado momento.
Assuma que cada processo é representado por um registro composto por um número identificador do processo e seu nome (faça uma struct).

6) Como você implementaria uma fila de pilhas? Escreva rotinas para implementar as operações corretas de inserção e remoção de números inteiros na pilha e de inserção e remoção de pilhas na fila.

7) Escreva o TAD Fila (estrutura + operações de inserção, remoção, cheia, vazia) que pode ser de 4 tipos diferentes: **int, char, float e double**. Utilize a declaração de union do C.

8) Escreva um procedimento que, dado duas filas, Fila *F1 e Fila *F2, concatene as duas filas e retorne:
a) Fila concatenada em *F1;
a) Fila vazia em *F2.
Utilize o protótipo
```
void concatena(Fila *F1, Fila *F2);
```

9) Dada as operações de fila Enqueue e Dequeue (insere e remove da fila), escreva a configuração final da fila após as seguintes operações:

Siga o exemplo: Fila = 1 -> 9 -> 3 -> 7 -> 9

Operações
Enqueue (10)
Enqueue (20)
Dequeue ()
Enqueue (40)
Enqueue (15)
Enqueue (25)
Dequeue (19)
Enqueue (1)
Dequeue ()

10) Dada as operações de fila Enqueue(Fila *F, int valor) e Dequeue() (insere e remove da fila) e Push (Fila *F) e Pop() (insere e remove da pilha), escreva a configuração final da pilha de filas.
Considere a struct abaixo.

```
typedef struct {
        int inicio, fim, total;
        int itens[TamFila];
} Fila;


typedef struct {
        int topo;
        Fila itens[TamPilha];
} Pilha_de_Fila;
```

Operações
Enqueue (F1,53)
Enqueue (F1,14)
Push(F1)
Enqueue (F2,40)
Enqueue (F2,15)
Enqueue (F2,25)
Push(F2)
Enqueue (F3,40)
Push (F3)

Siga o exemplo abaixo:

2
3 -> 7
4 -> 2
Início Pilha: 1 -> 2 -> 3