



INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE LABORATÓRIO DE ESTRUTURAS DE DADOS
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Lab. Estruturas de Dados

Atividade Prática 9 - Filas Sequenciais Circulares

Instruções

Responda às questões abaixo, desenvolvendo cada uma em um novo arquivo .cpp ou .c. Temos 3 questões, sendo que a questão 1 vale 3 pontos e as questões 2 e 3, 2 pontos cada, totalizando 7 pontos.

Questões

- Usando o código do material como base, faça:
 - implemente uma função chamada de `getFront` que retorna o elemento do início da Fila sem removê-lo
 - implemente uma função chamada de `elementoNaFila` que retorna 1 caso o elemento passado como argumento esteja presente na Fila, e 0 caso contrário
 - implemente uma função chamada de `indiceNaFila` que retorna 1 caso o índice passado como argumento contenha algum elemento da Fila, e 0 caso contrário.
- A forma como implementamos nossa Fila Sequencial Circular no material traz uma grande limitação: o nosso programa só pode manipular uma Fila por vez, e isso não é bom. Podemos resolver isso criando um registro `FILA`, que contém as variáveis particulares necessárias para o controle de cada Fila. Desse modo, basta adicionarmos um novo parâmetro às funções para que elas operem em cima da Fila passada como argumento. A partir do código abaixo, implemente uma versão melhorada de uma Fila Sequencial Circular.

```
#include <stdio.h>
#include <stdlib.h>

struct sfila{
    int *arr, ini, fim;
    int MAX;
};
```

```
typedef struct sFILA FILA;

void criar(FILA *fi, int tam_MAX);
void apagar(FILA *fi);

void enqueue(FILA *fi, int dado);
int dequeue(FILA *fi);

int tamanho(FILA *fi);
void imprimir(FILA *fi);
```

Note que, da forma como criamos o registro, cada registro FILA terá campos para o controle individual da Fila. Veja um exemplo de como criar e usar a Fila:

```
int main() {
    FILA fila1;
    criar(&fila1);

    enqueue(&fila1, 100);
    enqueue(&fila1, 50);
    enqueue(&fila1, 200);

    printf("Dequeue em Fila1 -> %d", dequeue(&fila1));

    imprimir(&fila1);

    apagar(&fila1);

    return 0;
}
```

Considere:

- a função **criar** tem a função de inicializar as variáveis da Fila. A função recebe tam_MAX que por sua vez, deve ser usado para alocar o array arr.
- a função **apagar**, por sua vez, deverá desalocar o array arr e reajustar os campos do registro FILA.
- as demais funções farão a mesma coisa conforme visto no material. Desta vez, no entanto, considerando o parâmetro **FILA *fi**, que espera receber o ponteiro para a Fila a ser manipulada.

Em essência, o código está praticamente pronto no material, à exceção da função criar. Você fará apenas as adequações necessárias para atender às novas especificações.

3. Deque (ou Fila de duas cabeças) é um tipo de Estrutura de Dados que suporta quatro operações:
- `inserir_ini`: insere um elemento no início
 - `remover_ini`: remove/obtem um elemento do início
 - `inserir_fim`: insere um elemento no fim
 - `remover_fim`: remove/obtem um elemento do fim

Usando como base o código da Fila Sequencial Circular, implemente uma Deque de inteiros.