**Atividade de Ponteiros**

Executada por, Rafael Kazhito Vidal Beltrão ADS - Turma C

O código também esta disponível no meu repositório do GitHub (rkazuhito)

<https://github.com/rkazuhito/Faculdade/blob/main/2o%20semestre/estrutura%20de%20dados/atividade%20fila%2029-02/70c4bd141cd1e134665067b434fdde954a1e0e2521c2fd33a05609cbcbe4ecb2e68047f55c9674f23680399bdfd1c83beee96e7bf136ddb03c49aa8466397da9.cpp>

-> Código comentado e explicado <-

/\*

Implementação de fila simples

\*/

#include <iostream>

#define tamanho 5

using namespace std;

// Define a estrutura que será a fila

// A estrutura armazena a indicação do início e final da fila e um vetor com os itens (valores) da fila

typedef struct {

int ini = 0; // Indica o índice do primeiro elemento na fila

int fim = 0; // Indica o próximo índice disponível para adicionar um elemento

int item[tamanho]; // Array que armazena os elementos da fila

} FILA;

// Retorna verdadeiro se a fila estiver vazia, falso caso contrário

bool filaVazia(FILA p) {

if (p.ini == p.fim) { // Se o índice de início é igual ao índice de fim, a fila está vazia

return true;

} else {

return false;

}

}

// Retorna verdadeiro se a fila estiver cheia, falso caso contrário

bool filaCheia(FILA p) {

int tam = sizeof(p.item) / sizeof(int); // Determina o tamanho do vetor de itens da fila

if (p.fim < tam) { // Se o índice de fim for menor que o tamanho máximo, a fila não está cheia

return false;

} else {

return true;

}

}

// Adiciona um valor na fila

void enfilera(FILA &p, int x) {

p.item[p.fim++] = x; // Adiciona o valor na posição do índice de fim e incrementa o índice de fim

}

// Remove e retorna o valor da fila

int desenfilera(FILA &p) {

return (p.item[p.ini++]); // Retorna o valor na posição do índice de início e incrementa o índice de início

}

// Mostra os valores armazenados na fila

void mostraFila(FILA p) {

cout << "Valores da fila: ";

for (int i = p.ini; i < p.fim; i++) { // Itera sobre os elementos da fila do início até o fim

cout << p.item[i] << " "; // Imprime cada elemento da fila

}

cout << "\n";

}

// Função principal para testar a implementação

int main() {

FILA s; // Criar a fila

// Verificar se a fila está vazia

if (filaVazia(s)) {

cout << "A fila está vazia." << endl;

} else {

cout << "A fila não está vazia." << endl;

}

// Enfileira um valor e verifica se a fila está vazia

enfilera(s, 10);

if (filaVazia(s)) {

cout << "A fila está vazia." << endl;

} else {

cout << "A fila não está vazia." << endl;

}

// Insere 3 elementos na fila

enfilera(s, 20);

enfilera(s, 30);

enfilera(s, 40);

// Mostra os valores da fila

mostraFila(s);

// Verifica se a fila está cheia

if (filaCheia(s)) {

cout << "A fila está cheia." << endl;

} else {

cout << "A fila não está cheia." << endl;

}

// Enfileira um valor e verifica se a fila está cheia

enfilera(s, 50);

mostraFila(s);

if (filaCheia(s)) {

cout << "A fila está cheia." << endl;

} else {

cout << "A fila não está cheia." << endl;

}

// Desenfileira e mostra o valor desenfileirado

cout << "Valor removido da fila: " << desenfilera(s) << endl;

mostraFila(s);

// Verifica se a fila está cheia

if (filaCheia(s)) {

cout << "A fila está cheia." << endl;

} else {

cout << "A fila não está cheia." << endl;

}

return 0;

}

-> Execução print<-

Saída

