

Laboratório de Programação I (MATA57)

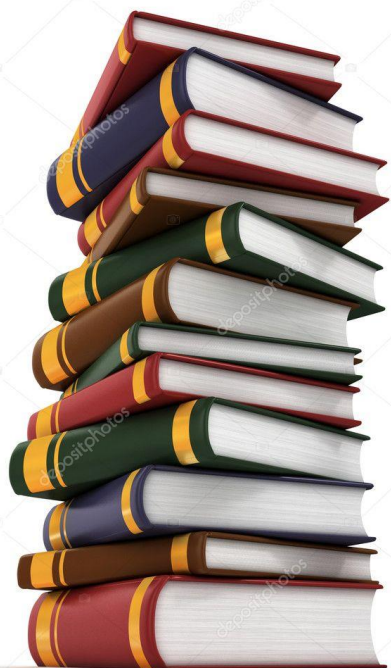


Prof.: Claudio Junior N. da Silva (claudiojns@ufba.br)

Pilhas, Filas e Listas

2023.1

PILHAS, FILAS E LISTAS



 **LISTA DE** _____ 

#	NOMBRE DEL ALUMNO/A				
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
16					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					

PILHAS, FILAS E LISTAS

- Pilhas, filas e listas são estruturas de dados, que nos permitem organizar e interagir com nossos dados e organizá-los de diferentes formas.
- Em C++ existem bibliotecas que nos permitem implementar esses tipos de estruturas.

Estrutura	Biblioteca
Pilhas	<code>#include <stack></code>
Filas	<code>#include <queue></code>
Listas	<code>#include <list></code>

Pilhas x Filas x Listas

	Pilhas	Filas	Listas
Tipo	Last-In First-Out	First-In First-Out	
Biblioteca	#include <stack>	#include <queue>	#include <list>
Declaração	stack<TIPO> S;	queue<TIPO> Q;	list <TIPO> L
Adiciona elemento	S.push(ELEMENTO); // topo	Q.push(ELEMENTO); // fim	L.push_front(ELEMENTO); L.push_back(ELEMENTO);
Retorna elemento	S.top(); // topo	Q.front(); // inicio	L.back(); L.front();
Remove elemento	S.pop(); // topo	Q.pop(); //início	L.pop_back(); L.pop_front(); L.remove(ELEMENTO); L.erase(list<TIPO>::iterator IT);
Verifica se vazio	S.empty();	Q.empty();	L.empty()
Reverte ordem			L.reverse();

Problema I – Como empilhar dados

- Faça um programa em C++ que leia n elementos e os armazene em uma estrutura de pilha.

```
#include <iostream>
#include <stack> // biblioteca para usar a pilha
using namespace std;

int main(){
    stack<int> pilha; // declaração da pilha com o tipo de dado
    int n, elemento;
    cout << "Qual o numero de elementos da Pilha? " << endl;
    cin >> n;
    cout << "Quais os elementos da Pilha? " << endl;
    for(int i=0; i<n; i++){
        cin >> elemento;
        pilha.push(elemento); // push permite empilhar os dados (FILO)
    }
    // retorna o elemento do topo (último) sem removê-lo
    cout << "O elemento do topo da Pilha e: " << pilha.top() << endl;
}
```

Problema I – Como listar elementos de uma pilha

- Altere o programa anterior para incluir uma função que imprima os elementos da pilha criada.

```
void listarElementosPilha(stack<int>& pilha) {
    stack<int> pilhaAuxiliar; // Pilha auxiliar

    // Transferir elementos da pilha original para a pilha auxiliar
    while (!pilha.empty()) {
        int elemento = pilha.top();
        pilha.pop();
        pilhaAuxiliar.push(elemento);
    }
    // Imprimir os elementos da pilha auxiliar
    cout << "Elementos da pilha:" << endl;
    while (!pilhaAuxiliar.empty()) {
        int elemento = pilhaAuxiliar.top();
        cout << elemento << endl;
        pilhaAuxiliar.pop();
    }
}

int main(){
    ...
    listarElementosPilha(pilha);
}
```

Problema II – Como enfileirar e listar elementos

- Faça um programa em C++ que leia n elementos e os armazene em uma estrutura de fila;
- Em seguida imprima os elementos da fila criada.

```
#include <iostream>
#include <queue> // biblioteca para usar a filas
using namespace std;
void imprimirElementosFila(queue<int> &fila) {
    cout << "Elementos da Fila:" << endl;
    while (!fila.empty()) {
        int elemento = fila.front();
        cout << elemento << endl;
        fila.pop();
    }
}

int main(){
    queue<int> fila;           // declaração da fila e seu tipo de dado
    int n = 6, elemento;
    cout << "Informe os elementos de da Fila: " << endl;
    for(int i=0;i<n;i++){
        cin >> elemento;
        fila.push(elemento); } // push permite enfileirar os dados (FIFO)
    cout << "Primeiro elemento da Fila: " << fila.front() << endl;
    cout << "Ultimo elemento da Fila : " << fila.back() << endl;

    imprimirElementosFila(fila);
}
```

Problema III – Inserir e imprimir elementos de uma lista

- Faça um programa em C++ que leia n elementos e os armazene em uma estrutura de lista;
- Em seguida imprima os elementos da lista criada.

```
#include <iostream>
#include <list>
using namespace std;
int main(){
    list<int> lista;
    list<int>::iterator it;
    int n = 6, elemento;
    cout << "Informe os elementos da Lista" << endl;

    for(int i=0;i<n;i++){
        cin >> elemento;
        lista.push_back(elemento); }

    cout << "Primeiro elemento da lista: " << lista.front() << endl;
    cout << "Último elemento da lista: " << lista.back() << endl;
    cout << "Tamanho da lista: " << lista.size() << endl;
    cout << "Os elementos da lista são: " << endl;
    for(it = lista.begin(); it!=lista.end();it++){
        cout << *it << endl;
    }
}
```