

OBS: grupos de 4 pessoas

Versão usando *stl* das questões:

Questão 1. Desenvolva os procedimentos de **entrar e sair de uma fila** a partir dos procedimentos **empilhar, desempilhar** de uma pilha. **Utilize 2 (duas) pilhas.**

Questão 2. Desenvolva os procedimentos de **empilhar e desempilhar de uma pilha** a partir dos procedimentos **add_fila, remove_fila**, de uma fila. **Utilize 2 (duas) filas.**

Questão 3. Considerando **uma lista duplamente encadeada**, IMPLEMENTE AS FUNÇÕES QUE ESTÃO INCOMPLETAS NO CÓDIGO:

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

// Define a estrutura do nó
struct No {
    int dado;
    struct No* prox;
    struct No* prev;
};

// Função para criar um novo nó
struct No* novoNo(int dado) {
    struct No* no = (struct No*)malloc(sizeof(struct No));
    no->dado = dado;
    no->prox = NULL;
    no->prev = NULL;
    return no;
}

// Complete a Função para inserir um nó no início da lista
void inserirNoInicio(struct No** cabeca, int dado) {
    struct No* no = novoNo(dado);
}
```

```

// Complete a Função para exibir a lista do inicio até o final
void exibirLista(struct No* no) {

}

// Complete a Função para exibir a lista invertida(final até o
inicio)
void exibirLista_inv(struct No* no) {

}

// Função principal
int main() {
    struct No* cabeca = NULL;
    inserirNoInicio(&cabeca, 10);
    inserirNoInicio(&cabeca, 20);
    inserirNoInicio(&cabeca, 30);
    printf("Conteúdo da lista duplamente encadeada: ");
    exibirLista(cabeca);
    exibirLista_inv(cabeca);
    return 0;
}

```

Questão 4. Usando o modelo abaixo:

```

class No{
    public:
        int mat;
        char nome[23];
        No *prox;
        No *ant;
        No(int m,char n[23]){
            mat=m;
            strcpy(nome,n);
        }
}

```

```

        prox=NULL;

        ant=NULL;

    } }

```

4.1. Faça, usando uma LISTA DUPLAMENTE LIGADA, as seguintes funções:

Inserção ordenada;

Mostrar a lista do início até o final;

Remoção do último elemento;

Remoção do primeiro elemento;

4.2. Faça, usando uma LISTA DUPLAMENTE LIGADA, as seguintes funções:

Inserção no final da lista;

Inserção no início da lista;

Remoção do elemento procurado;

Mostrar a lista do início até o final;

Mostrar a lista do final até o início;

Questão 5.

Considerando uma lista circular, IMPLEMENTE AS FUNÇÕES QUE ESTÃO INCOMPLETAS NO CÓDIGO:

```

#include <stdio.h>

#include <stdlib.h>

// Define a estrutura do nó

struct No {

    int dado;

    struct No* prox;

};

// Função para criar um novo nó

struct No* novoNo(int dado) {

    struct No* no = (struct No*)malloc(sizeof(struct No));

    no->dado = dado;

    no->prox = NULL;

    return no;

```

```

}

// Completar Função para inserir um nó no final da lista
circular

void inserirNoFinal(struct No** ultimo, int dado) {
    struct No* no = novoNo(dado);

}

// Completar Função para exibir a lista circular

void exibirLista(struct No* ultimo) {

}

// Função principal

int main() {
    struct No* ultimo = NULL;
    inserirNoFinal(&ultimo, 10);
    inserirNoFinal(&ultimo, 20);
    inserirNoFinal(&ultimo, 30);
    printf("Conteúdo da lista circular: ");
    exibirLista(ultimo);
    return 0;
}

```

Questão 6.

Fazer a questão 10.2 do livro pag. 112 (cap10 – Listas Ordenadas)

Questao 7.

Fazer a questão 10.3 do livro pag. 112 (cap10 – Listas Ordenadas)

Questao 8.

Fazer a questão 10.4 do livro pag. 113 (cap10 – Listas Ordenadas) letras **(a) e (b)**

Questao 9.

Fazer a questão 10.4 do livro pag. 113 (cap10 – Listas Ordenadas) letras **(c) e (d)**

Questao 10.

Fazer a questão 10.6 do livro pag. 113 (cap10 – Listas Ordenadas)

Questao 11.

Fazer a questão 10.7 do livro pag. 113 (cap10 – Listas Ordenadas)