

DISCIPLINA DE ESTRUTURA DE DADOS

PROFESSORES BRUNO DE CASTRO HONORATO SILVA E WELLINGTON FRANCO

Entrega será no dia 25/01/2021

1. O que é e para que serve uma pilha?
2. Considere que existam duas pilhas vazias denominadas S1 e S2, preencha as ilustrações de cada uma das pilhas e desenvolva as instruções presentes no trecho de algoritmo abaixo. Considere uma rotina chamada Par, que recebe um valor e devolve o valor lógico Verdade se o número é par e falso se é ímpar. Preencha as ilustrações de cada pilha nos pontos definidos: A, B e C. Deixe as representações das pilhas como elas estão no ponto determinado.

```

A = 0
Para X = 12 ate 19 faça
    empilha(S1,X)
fim-para
imprima ('passo dois')
enquanto(não( pilha_vazia(S1)))
    faça
        x ← desempilha(S1)
        Se (Par( x))
            então empilha(S2,x)
        fim-se
    fim-enquanto
Imprima('final')
    
```

A		B		C	
S1	S2	S1	S2	S1	S2

3. Considere que existam duas filas vazias denominadas F1 e F2. Execute as instruções do trecho de algoritmo abaixo. Deixe as representações das filas como elas estão no ponto determinado. Preencha as ilustrações de cada fila nos pontos definidos: A, B e C. Considere que os dados a serem lidos no algoritmo são: { 3, 6, 8, 2, 5, 1, 7, 4, 0, 5}.

```

leia(x)
enquanto x <> 0 faça
    Insere(F1,x)
    leia(x)
fim-enquanto
leia(y)
enquanto(não( Fila_vazia(F1))) faça
    x ← Primeiro(F1)
    Remove(F1)
    Se x < y
        então Insere(F2,x)
    fim-se
fim-enquanto
imprima('final')
    
```

Preencher com as situações solicitadas:

A	F1								
	F2								
B	F1								
	F2								
C	F1								
	F2								

4. Cite as principais diferenças entre Pilha e Fila?

5. Em que situações uma fila pode ser utilizada?

Considere as seguintes estrutura de dados para as questões 5,6,7,8,9 e 10.

```
class Node {
public:
    TIPO x;
    Node *prox;
};

class Pilha {
private:
    Node *topo;

public:
    Pilha(); // Inicializa topo com NULL
    void push(TIPO); // empilhar
    TIPO pop(); // desempilhar
    TIPO get_top(); // obter elemento do topo da pilha
    bool is_empty(); // verifica se pilha está vazia
    void clear(); // limpar pilha
    int size(); // tamanho da pilha
    void inverter(); // inverte ordem dos elementos da pilha
    void print(); // imprimir elementos da pilha na tela

    // Outros métodos
};
```

Na classe Node acima, TIPO se refere ao tipo de dado que a pilha deverá armazenar (int, char, float). Todavia, NÃO é obrigatório o uso de template. Assim, se o aluno preferir, pode criar uma classe Node e uma classe Pilha para cada questão.

```
class Node {
public:
    TIPO x;
    Node *prox;
};

class Fila {
private:
    Node *inicio;
    Node *fim;

public:
    Fila(); // Inicializa inicio e fim com NULL
    void queue(TIPO); // enfileirar
    TIPO dequeue(); // desenfileirar
    TIPO first(); // elemento do início da fila
    TIPO last(); // elemento do fim da fila
    bool is_empty(); // verifica se fila está vazia
    void clear(); // limpar fila
    int size(); // tamanho da fila
    void inverter(); // inverte ordem dos elementos da fila
    void print(); // imprimir elementos da fila na tela

    // Outros métodos
};
```

Na classe Node acima, TIPO se refere ao tipo de dado que a fila deverá armazenar (int, char, float). Todavia, NÃO é obrigatório o uso de template. Assim,

se o aluno preferir, pode criar uma classe `Node` e uma classe `Fila` para cada questão.

6. Desenvolva uma rotina para inverter a posição dos elementos de uma pilha `P`.
7. Crie um MÉTODO chamado `is_empty` na classe `Pilha` que retorne `true` se a pilha estiver vazia ou `false` caso contrário.
8. Crie um MÉTODO chamado `size` na classe `Fila` que retorne a quantidade de elementos da fila.
9. Crie um MÉTODO chamado `print` na classe `Fila` para imprimir todos os elementos da fila em ordem de inserção.
10. Escreva uma função que, dado duas filas, concatene as duas filas. Retorne a fila concatenada em `F1`. `F2` deve ficar vazia.