

## INF 213 - Roteiro da Aula Prática

O objetivo desta aula é praticar o uso e implementação de pilhas, filas e filas de prioridade.

Arquivos fonte e diagramas utilizados nesta aula:

<https://drive.google.com/file/d/1EvZ238XqPK5Qmv9sMdbvRy0zDnAbSvaF/view?usp=sharing>

Considere o programa “driver.cpp”. Implemente cada função “etapaN()” adicionando a implementação da respectiva etapa (você pode usar includes na área de cada etapa para incluir classes ou dividir seu código em vários arquivos).

A entrada de cada etapa deve ser precedida de seu número. Por exemplo, para testar a entrada “5 3 2 5 4 1” na etapa 3 seu arquivo de entrada deverá conter: “3 5 3 2 5 4 1”.

### Etapa 1

Crie uma função que le 6 números da entrada padrão. Para cada número lido, ele deverá ser armazenado em uma fila e em uma pilha. Ao final, ele deve imprimir o conteúdo da pilha (com um espaço em branco após cada elemento), uma nova linha, o conteúdo da fila e uma nova linha.

**Qual a diferença entre a ordem dos números tirados da fila e da pilha?**

### Etapa 2

Desenvolva uma função que, dada uma entrada contendo vários caracteres e, entre eles, símbolos ( , ) , [ , ] , { e } , verifica se tais símbolos estão consistentes (balanceados). Sua função deverá imprimir “Consistente” caso a entrada esteja consistente e “Inconsistente” caso a entrada esteja inconsistente. Imprima uma nova linha (vazia) após o resultado.

Exemplos de entradas consistentes (nesse caso cada linha é uma entrada):

(3\*(5+2))  
[3+(2-1)\*5]  
[] {} () ([])  
( 5+ 1=500)  
( inf213 123)

Exemplos de entradas inconsistentes:

))( (  
)([  
[5+2. \*(4+2])  
[5+ 2\*(4+1]  
([])

Cada arquivo de teste terá apenas uma entrada (uma linha contendo a entrada).

### Etapa 3

Considere o arquivo MyQueue2.h : implemente as funções da classe MyQueue2 de forma com que tal classe represente uma fila. O desafio será fazer essa implementação utilizando duas pilhas como os únicos membros de dado da sua classe. Você não poderá declarar NENHUM membro de dados adicional e nem mesmo alocar variáveis locais em suas funções, exceto para 1 variável local “simples” (int, float, double, T, ...) se necessário! Em média, todas funções deverão ter tempo constante ( $O(1)$  amortizado) !

Propositamente nenhuma função da sua classe será constante (isso facilitará sua implementação).

Esta questão é tipicamente utilizada em entrevistas de emprego de grandes empresas (como Google e Facebook).

Sua classe será testada utilizando uma função que cria uma fila de strings e, a seguir, lê vários comandos de fila a partir da entrada, aplicando tais operações à sua fila (tal função já está pronta em etapa3(), mas precisa ser descomentada).

Por exemplo, considere a entrada abaixo:

Entrada	Comentario
p teste	Insere “teste” na fila
p abc	Insere “abc” na fila
s	Imprime o tamanho da fila (2)
e	Imprime se a fila está vazia (imprimira “cheia”)
f	Imprime a frente da fila (“teste”)
P	Remove o elemento da frente da fila (note que o P é maiúsculo)
f	Imprime a frente da fila (“abc”)

**Dica: como sempre, faça rascunhos em papel!.**

### Etapa 4

Considere o programa pilhaInteirosImpressao.cpp . A entrada dele consiste em um número N seguido por uma sequência contendo exatamente N caracteres do tipo I e N do tipo R (eles podem aparecer de forma embaralhada). Quando um caractere I é lido, um número é inserido em uma pilha (o primeiro número a ser inserido será sempre o 1, depois o 2, depois o 3 e

assim por diante). Quando o caractere R é lido, o topo da pilha é removido e impresso na tela (assuma que R nunca aparecerá quando a pilha estiver vazia).

Por exemplo, a entrada : 5 I I I R R I I R R R geraria a saída: 5 3 2 5 4 1 (o primeiro número 5 indica que a saída contém 5 números (além dele), ou seja, é o número N).

A entrada 3 I I I R R R geraria a saída: 3 3 2 1

Crie uma função que, dada uma saída no formato utilizado por “pilhaInteirosImpressao.cpp”, imprime SIM se for possível gerar tal saída utilizando alguma entrada para tal programa ou NAO se não for possível. Assuma que  $1 \leq N \leq 100000$  e que a saída será uma permutação dos números entre 1 e N.

Exemplos de entrada/saída:

Entrada	Saída
5 3 2 5 4 1	SIM
3 3 2 1	SIM
3 3 1 2	NAO

### **Submissao da aula pratica:**

A solucao deve ser submetida ate as 18 horas do dia 11/06 utilizando o sistema submittty (submittty.dpi.ufv.br). Envie todos arquivos (inclusive MyQueue, MyStack, ...). Atualmente a submissao so pode ser realizada dentro da rede da UFV.