

Universidade Federal de São Carlos

Estrutura de Dados I - Lista 2

Os seguintes exercícios a seguir deverão ser implementados dentro dos respectivos arquivos presente no Lista2.zip disponibilizado no classrom. Exemplos de entrada e saída estarão presente nos diretórios. Utilize indentação e comente seu código em trechos não triviais. É estritamente proibido alterar o nome dos arquivos assim como realizar alterações nos arquivos header caso ele já exista.

Recursividade

1. Implemente as funções de imprimir normalmente e reversamente uma lista de maneira recursiva.
2. Inserir um elemento de maneira ordenada em uma lista ordenada.
3. Calcular o tamanho de uma lista.

Fila

1. Utilizando os conceitos de Fila circular estática. Implemente um programa que auxilie no gerenciamento da fila de espera de um lava a jato. Onde poderá a qualquer momento chegar mais veículos ou surgir oportunidade para lavar o prioritário. Caso chegue mais carros que o tamanho da fila exiba a mensagem: “Fila Cheia” e encerre o programa. Como identificação de cada veículo será utilizado a parte numérica das placas.
2. Seja N um número removido do começo da fila, faça um programa que possa enfileirar inteiros fornecidos pelo usuário ou realizar uma determinada função de acordo com a tabela de opções:

- | |
|--|
| <ol style="list-style-type: none">1. Enfileira um Inteiro natural maior que zero.2. Imprima os N primeiros termos da sequência Fibonática e enfileire o último termo.3. Joga o N no fim da fila.4. Imprime a fila.5. Libera memória (Se houver) e encerra. |
|--|

3. Implemente as funções pop e push (pseudo pilha) usando apenas 2 filas e suas respectivas funções.
4. Resolva o exercício [1110](#) do urionlinejudge.

Pilha

1. Utilizando apenas as funções disponibilizadas no header e pilhas auxiliares, faça um programa que ao receber um inteiro N, gere uma sequência entre 1 e 3N e imprima:
 - A sequência em sua forma original.
 - A sequência reversa.
 - A sequência alternada entre paridade. (n1, n3, n5, ..., n2, n4, n6,...)
 - A sequência original novamente.

NOTA: Não é permitido salvar em qualquer momento uma cópia da sequência original.

2. Implemente as funções enqueue e dequeue (pseudo fila) usando apenas 2 pilhas e suas respectivas funções.

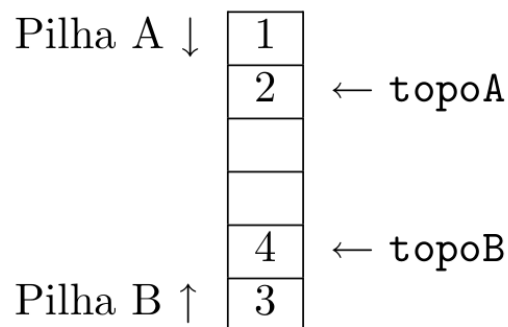
3. Duas pilhas A e B podem compartilhar o mesmo vetor, como esquematizado na figura a seguir. Faça as declarações de constantes e tipos necessários e escreva as seguintes rotinas:

(i) o procedimento criaPilhas que inicia os valores de topoA e topoB;

(ii) as funções vaziaA e vaziaB;

(iii) os procedimentos empilhaA, empilhaB, desempilhaA e desempilhaB.

Só deve ser emitida uma mensagem de pilha cheia se todas as posições do vetor estiverem ocupadas.



4. Resolva o exercício [1068](#) do urionlinejudge.

Observações Gerais

- Consulte sempre o arquivo de notas presente no diretório do exercício (se houver).
- Em headers vazios cabe a você implementar a estrutura de dados e funções, mas não se esqueça de respeitar o exercício proposto pelo enunciado.
- Caso haja alterações na lista, haverá um aviso informado no classroom.
- São boas práticas: diffar as entradas e saídas para verificar a integridade do aplicativo, utilizar o depurador para encontrar possíveis bugs e ir na monitoria tirar dúvidas.
- Você pode criar funções auxiliares, caso ache necessário, respeitando o exercício proposto. (Exemplo: não fazer uma função iterativa para resolver algo que deveria ser resolvido recursivamente).