

UENF

Universidade Estadual do Norte Fluminense Darcy Ribeiro

Curso: Ciência de Computação

Data: 15/04/2021

Atividade: Módulo 2 – LE2

Período: 3º

Disciplina: Estrutura de Dados I

Professor: Fermín Alfredo Tang

Turno: Diurno

Nome do aluno:

Matrícula:

Atividades para LE2

Considerando os conceitos sobre Pilhas:

1. Escreva um programa que converta um número decimal em hexadecimal, **usando** o método de divisões sucessivas e **uma pilha**. Obs. Lembrar que caso o resto da divisão for 10, 11, 12, 13, 14 ou 15, o dígito correspondente deverá ser A, B, C, D, E ou F.
2. Um palíndromo é uma *string* que pode ser lida para frente e para trás com o mesmo resultado. Por exemplo, as seguintes frases são palíndromos:

A base do teto desaba.

A diva em Argel alegre-me a vida.

Adias a data da saída.

Socorram-me, subi no ônibus em Marrocos.

Observe que nos palíndromos se deve ignorar o espaçamento, a pontuação, e as diferenças entre letras maiúsculas e minúsculas.

Escreva uma função para testar se uma *string* é um palíndromo **usando uma pilha**. A ideia é armazenar os caracteres da *string* na pilha e recuperar eles formando uma nova *string*. Depois comparar a nova *string* com a *string* original. Teste a sua função com os exemplos mostrados acima.

3. Modifique o programa de casamento de parênteses para considerar o casamento de { }, [] e (). O programa deverá verificar a correspondência entre o símbolo de abertura e seu correspondente de fechamento. Identifique e reporte os tipos de erro possíveis caso existam, indicando a linha, ou uma mensagem indicando que o casamento é perfeito. Teste o seu programa com arquivos de texto, com erros e sem erros.
4. Escreva um programa para encontrar todas as soluções para o problema das 8 rainhas. Observe que não basta apenas uma configuração viável, queremos descobrir todas elas. O programa deve ser capaz de continuar a busca após a primeira configuração válida. Contar o número de soluções encontradas e imprimir essas configurações na tela.

Considerando os conceitos sobre Filas:

5. Escreva um programa em C para avaliar o resultado de uma expressão prefixa. O programa deverá utilizar uma fila para armazenar os elementos da expressão (operadores e operandos). O processo de avaliação consiste em fazer a leitura dos

elementos armazenados (tokens) na fila, substituindo-se, sempre que encontrada, uma sequência de um operador seguida de dois operandos pelo resultado da operação indicada. Caso contrário, os elementos lidos são reinseridos na fila. O processo é repetido até a fila conter um único elemento que é o resultado da expressão.

Por exemplo, a seguinte expressão é uma expressão prefixa cujo resultado é 159.

- + * 9 + 2 8 * + 4 8 6 3

Por exemplo, quando um operador é seguido de dois operandos, tal como + 2 8, colocamos o resultado, 10, na fila.

Após a primeira avaliação da expressão temos:

- + * 9 10 * 12 6 3

Após a segunda avaliação, temos:

- + 90 72 3

Após a terceira avaliação, temos:

- 162 3

Após a quarta avaliação, temos:

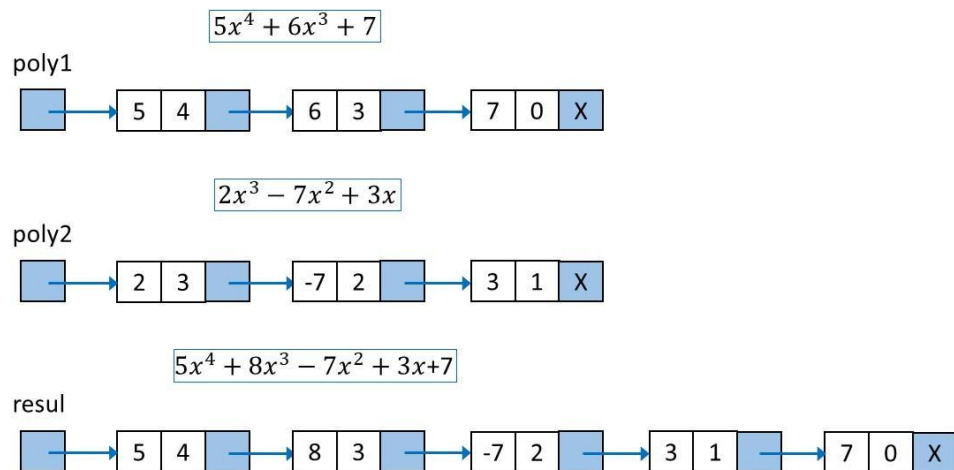
159

- 6.- Usando somente algoritmos no TAD fila, escreva uma função em C chamada *catQueue()* que concatene duas filas. A segunda fila deve ser colocada no final da primeira fila. Considere os casos em que: i) a segunda fila precisa ser preservada; ii) a segunda fila não deve ser preservada; iii) posso adicionar operações extras ao TAD para ganhar eficiência na concatenação em ii).
- 7.- Escreva um programa em C que classifique as palavras em um texto, disponível na forma de arquivo, em categorias de acordo com a letra inicial de cada palavra. As palavras iniciadas com a mesma letra deverão ser colocadas em uma mesma fila. Após a classificação, exiba o número de palavras existentes em cada categoria.

Considerando os conceitos sobre Listas:

8. Escreva um programa que **some** polinômios. Cada polinômio deve ser representado como uma lista encadeada. O primeiro nó na lista representa o termo de maior grau, enquanto que o último representa o termo de menor grau. Cada nó contém três campos: i) coeficiente do termo; ii) expoente do termo; iii) o ponteiro ao seguinte termo.

Os polinômios podem ser adicionados utilizando um processo de intercalação. As duas listas são combinadas por intercalação, além de realizar uma operação de **soma** quando os expoentes forem iguais. Observe o exemplo da Figura.



Regras para a adição:

- Se os expoentes são iguais, os coeficientes são somados e o resultado inserido;
- Se os expoentes são diferentes, o termo com maior expoente deverá ser inserido, enquanto o menor aguarda para ser comparado com outro termo.
- Se o resultado da adição é 0, o termo deve ser eliminado do polinômio soma.
- Se um polinômio termina antes o outro, os termos restantes do polinômio maior serão adicionados no polinômio soma.

Os dados do polinômio (coeficientes e potências) podem ser lidos de um arquivo ou ingressados do teclado. Imprima os dois polinômios de entrada e o polinômio soma.

- Modifique a TAD Lista de maneira a implementar uma **lista duplamente encadeada**. Inclua uma função para percorrer a lista no sentido inverso (backward). Teste a TAD duplamente encadeada usando os dados de censo populacional das cidades dos EUA. Crie uma lista duplamente encadeada com esses dados, ordenada com base no campo população. Imprima os dados da lista em ordem crescente (forward) e decrescente (backward).

Metropolitan area	2000		
New York–No. NJ	21,199,865	Atlanta area	4,112,198
Los Angeles area	16,373,645	Miami–Fort Lauderdale	3,876,380
Chicago area	9,157,540	Seattle area	3,554,760
Washington–Baltimore	7,608,070	Phoenix area	3,251,876
San Francisco area	7,039,362	Minneapolis–St. Paul	2,968,806
Philadelphia–Atlantic City area	6,188,463	Cleveland area	2,945,831
Boston area	5,819,100	San Diego area	2,813,833
Detroit area	5,456,428	St. Louis area	2,603,607
Dallas–Fort Worth	5,221,801	Denver area	2,581,506
Houston–Galveston area	4,669,571	San Juan, PR, area	2,450,292

- Defina o conceito de lista multi-encadeada. Ilustre graficamente. Dar um exemplo de aplicação.