LISTA1 - N1 - ESTRUTURA DE DADOS 2

Estruturas de dados básicas, Alocação Sequencial/Dinâmica e Recursão

> Prof. Ed

Instruções:

- Os programas **NÃO DEVEM SER COMPACTADOS**. O código-fonte deve ser enviado via upload diretamente na resposta do exercício (arquivo por aquivo)
- Cada arquivo/classe deve ter o seguinte formato: `ED2Lista1-questaoXX` onde `XX` é o número da questão correspondente.
- Os programas podem ser em C ou Java (e SOMENTE em uma das duas). É permitido usar ambas as linguagens no exercício (fazer umas questões em C e outras em Java, caso se queira)
- O trabalho é em **INDIVIDUAL**
- **IMPORTANTE**: NÃO SERÃO ACEITOS TRABALHOS QUE NÃO ESTIVEREM NO FORMATO ACIMA
- OBSERVAÇÃO: TODOS os programas entregues devem ter o seguinte cabeçalho(ou comentário do javadoc):

```
/**
**    Função :
**    Autor :
**    Data  :
**    Observações:
*/
```

Onde deverá estar escrito o que o programa faz, o autor (nome, turma, a data e as observações que forem pertinentes. Os trabalhos **não serão aceitos** após a data SOB HIPÓTESE ALGUMA.

Recursão, Alocação e Estruturas de Dados Básicas

1. Escreva uma função recursiva (ou classe com método recursivo) para resolver a seguinte série:
   - $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \ldots$
2. Usando recursividade faça uma função (ou uma classe com método recursivo) que INVERTA a ordem de um array de inteiros (ou de objetos/structs)
3. Escreva uma função recursiva (ou uma classe com método recursivo) que determine quantas vezes um dígito K ocorre em um número natural N. Por exemplo, o dígito 2 ocorre 3 vezes em 762021192
4. Sabendo que o resto da divisão, operação também conhecida como MOD na Matemática, de x por y (sendo x e y inteiros positivos), pode ser descrita por:
   - $mod(x, y) = mod(x - y, y)$ se $x > y$
   - $mod(x, y) = x$ se $x < y$
   - $mod(x, y) = 0$ se $x = y$

   Faça uma função (ou método recursivo) que implemente o resto da divisão como definido acima.
5. Escreva um método recursivo EM JAVA que pega uma string s e exibe seu inverso. Assim, por exemplo, o inverso de "*Estruturas de Dados*" é "*sodaD ed saruturtsE*".
6. Faça um programa em Java recursivo que conte o número de nós (células) de uma lista encadeada (não é permitido usar Collections neste exercício)
7. Faça um programa (em C ou Java) para mostrar todas as permutações de uma string -- todas as strings diferentes que podem ser criadas pela reorganização dos caracteres da string original (as palavras criadas a partir dessas strings são conhecidas como *anagramas*), exibindo ao final, a quantidade de anagramas desta string específica que deve ser inserida pelo usuário. Como exemplo, se a entrada fosse a string `abc` a saída seria:
8. ```
   abc
   acb
   bac
   bca
   cab
   cba
   Quantidade de permutações: 6
   ```
   ```

9. Escreva um programa em Java ou C que possa executar a Cifra de César para mensagens em Português que incluam tanto caracteres minúsculos como maiúsculos. Seu programa deve criptografar e descriptografar a mensagem.

10.

11. Use recursão para escrever um método em Java para determinar se uma string s tem mais vogais que consoantes.

12. ESTRUTURAS ANINHADAS (OU COMPOSTAS). Imagine que, em um sistema de reserva de passagens aéreas, as informações são armazenadas em uma Lista de Voos. Cada nó da lista de voos armazena o número do voo, o destino do voo, o total de assentos, o número de assentos ainda disponíveis, a lista de passageiros daquele voo (outra lista, obviamente) e a fila de espera daquele voo. Implemente uma lista encadeada que armazene a Lista de Passageiros, usando estruturas (classes) para armazenar os passageiros (apenas informações básicas como nome, cpf e telefone), voos (número, destino, aeronave, assentos totais e sobrando). Em seguida faça uma Lista de Voos considerando tanto a lista de passageiros como a fila de prioridades (1 a 5 sendo a menor a mais alta). Use TAD's (ou classes) para manipular essas estruturas exclusivamente através das operações de cada TAD (insere, eliminar e assim por diante). Pode ser em C ou Java

13. Usando recursividade faça uma função (ou uma classe com método recursivo) para calcular a soma dos quadrados dos N primeiros números inteiros positivos.

14. Um palíndromo é uma string (palavra) que se escreve exatamente igual lida em um sentido ou em outro. Palavras assim como ana, $radar$, ala, etc. são exemplos de palíndromos. Escreva uma função recursiva para testar se uma função é ou não um palíndromo.

15. Ler um númer inteiro positivo $n < 10$ e, então calcular o desenvolvimento do polinômio $(x + 1)^n$. Imprimir cada potencia x^2 na forma x**i.

    - Sugestão:
    - $(x + 1)^n$ = $C_{n,n}x^n + C_{n,n-1}x^{n-1} + C_{n,n-2}x^{n-2} + \cdots + C_{n,2}x^2 + C_{n-1}x^1 + C_{10}x^0$
    - onde $C_{n,n}$ e $C_{n,0}$ são 1 para qualquer valor de n.

    - A relação de recorrência dos coeficientes binomiais é:
        - $C(n, 0) = 1$
        - $C(n, n) = 1$

- $C(n, k) = C(n-1, k-1) + C(n-1, k)$

- Esses coeficientes constituem o famoso Triângulo de Pascal e será preciso definir a função que gera o triângulo...
- 1
  1 1
  1 2 1
  1 3 3 1
  1 4 6 4 1
  1 5 10 10 5 1
- FELIZMENTE todos fizeram este programa (que gera o Triângulo de Pascal) no período passado.

16. Escrever uma função recursiva para calcular o valor aproximado do número e, somando a série: $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$ até que os termos adicionais para somar sejam menores que $1.0e^{-8}$.

"Todos querem ganhar medalhas de ouro, mas poucos querem treinar na intensidade necessária para conquista-las" – Mark Spitz, ganhador de 11 medalhas olímpicas.

"Não é o mais forte que sobrevive, nem o mais inteligente. Quem sobrevive é o mais disposto à mudança” – Charles Darwin, biólogo e naturalista.

Portanto, MUDE!

BOM EXERCÍCIO! @edkallenn