

EXERCÍCIOS SOBRE RECURSÃO (LISTA 2)

Instruções:

- Crie uma classe chamada Exercicios.
- Para todos os exercícios, crie os métodos público e privado. O nome do método público deve ser o nome do exercício (e.g. exercicio1, exercicio2, exercicio3 etc.).
- Crie uma classe de teste e coloque o método main. No main, teste todos os exercícios criados, imprimindo na tela os resultados.
- Em todos os exercícios, utilize tratamento de exceções conforme visto em aula.

Exercício 1. Crie um método recursivo que recebe um array de inteiros e um inteiro N e retorna verdadeiro caso o inteiro N esteja presente no array, ou falso caso contrário.

Exercício 2. Crie um método recursivo que recebe um inteiro N por parâmetro e retorna verdadeiro caso ele seja um número primo e falso caso contrário.

Exercício 3. Crie um método recursivo que recebe uma String e retorna uma nova String, que é a String de entrada invertida.

Exercício 4. Crie um método recursivo que recebe um inteiro positivo (e maior que 0) N e calcula o seguinte: $1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/N$.

Exercício 5. Crie um método recursivo que retorna verdadeiro caso uma String recebida por parâmetro seja um palíndromo e falso caso contrário. Um palíndromo é um texto que, desconsiderando espaços, pontuação e acentuação, pode ser lido da mesma forma tanto de frente pra trás quanto de trás para frente. Por exemplo:

- Ana

- Renner
- O teu dueto
- Eva, asse essa ave
- Seco de raiva, coloco no colo caviar e doces
- Fica facil escrever palindromos com nomes proprios, como fazem Mezaf Omocso, Irporp Semon, Mocs Omord, Nila Prevercs e Lica Facif

No seu método público, antes de chamar o método privado, você deve remover pontuações (exclamações, pontos de interrogação, pontos finais e vírgulas) e espaços, para então chamar o método privado. Você pode assumir que o texto de entrada não possuirá acentos.

Exercício 6. Crie um método recursivo que recebe um array de String e retorna a quantidade de posições que contém palíndromos neste array. Utilize o método desenvolvido no exercício 5 para auxiliar.

Exercício 7. Crie um método recursivo que conta quantas vezes um determinado dígito positivo D aparece em um inteiro positivo N. Por exemplo, se $N = 161451811$ e $D = 1$, o retorno do método deve ser 5, pois o dígito 1 aparece 5 vezes em N. Você deve utilizar apenas operações matemáticas para isto, sem utilização de Strings.

Exercício 8. Crie um método recursivo que recebe um inteiro positivo (maior do que 0) e retorna a média dos seus dígitos. Por exemplo, para a entrada 6845, o resultado do método deve ser 5.75. Você deve utilizar apenas operações matemáticas para isto, sem utilização de Strings.

Exercício 9. Crie um método recursivo que recebe um número inteiro (maior do que 0) e retorna verdadeiro caso o número seja um número perfeito e falso caso contrário. Um número perfeito é um número que a soma de todos os seus divisores naturais próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo, 6 é um número perfeito, pois $6 = 1+2+3$. Outro número perfeito é o 28, pois $28 = 1+2+4+7+14$. Atualmente, são poucos os números perfeitos conhecidos (cerca de 49). Para testar seu método, os números abaixo são os 10 menores números perfeitos:

- 6
- 28
- 496
- 8.128
- 33.550.336
- 8.589.869.056
- 137.438.691.328
- 2.305.843.008.139.952.128
- 2.658.455.991.569.831.744.645.692.615.953.842.176
- 191.561.942.608.236.107.294.793.378.084.303.638.130.997.321.548.169.216

Os 54 dígitos do último número listado não são nada perto do número perfeito encontrado em 2016, que possui mais de 44 milhões de dígitos.

Naturalmente, você poderá testar apenas os 4 primeiros números perfeitos no seu método, devido ao alto poder de processamento necessário para a verificação dos demais.

Você deve utilizar apenas operações matemáticas para isto, sem utilização de Strings.