

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

**Lista de exercícios 01 - Recursividade e conversão de bases**  
**16 de agosto de 2019**

**OBS: Para a parte 1, não será permitido utilizar atributos da classe em nenhuma das questões. Não devem ser usadas instruções de repetição for, while ou do-while.**

**Parte 1 - Recursividade (ESCOLHA 5 QUESTÕES PARA RESPONDER)**

1) Utilizando a recursividade, escreva métodos recursivos capazes de fazer as seguintes operações (utilize métodos auxiliares, se necessário):

- a) Somar todos os números primos de um intervalo fechado de números;
- b) Multiplicar todos os elementos pares para a n-ésima sequência de Fibonacci.

Exemplo: `public int multiFib(int n);`

Entrada	Saída
n = 1	0
n = 4	2
n = 6	16

2) Dado um número natural n na base decimal, escreva um método recursivo que retorna seu valor na base binária.

**Exemplos:**

`print(retornaBinario(4))`

Saída: 100

3) Considerando um número arbitrário máximo 1000, crie um método recursivo que *imprime* na tela uma sequência de números, primeiro em ordem crescente e depois decrescente, inicialmente de um número passado como argumento, e então multiplicando-o por 2 sem passar do número máximo.

**Ex.:**

`public void seq(int n);`

`seq(100) = 100, 200, 400, 800, 800, 400, 200, 100`

**ESCOLA POLITÉCNICA DE PERNAMBUCO – POLI**  
**ENGENHARIA DA COMPUTAÇÃO - ECOMP**  
**ESTRUTURAS DE DADOS - 2019.2**

4) Crie um método recursivo que verifica se todas as letras presentes em uma primeira String também estão contidas na segunda.

**Exemplos:**

Entrada: "oi" e "abacaxi" → Saída: false  
Entrada: "lua" e "australia" → Saída: true

5) O hiperfatorial de um número N, escrito  $H(n)$ , é definido por:

$$H(n) = \prod_{k=1}^n k^k = 1^1 \cdot 2^2 \cdot 3^3 \cdot \dots \cdot (n-1)^{n-1} \cdot n^n$$

Sendo assim, faça uma função recursiva que receba um número inteiro positivo N e retorne o hiperfatorial desse número.

6) Considere um sistema numérico que não tenha a operação de adição implementada e que você disponha somente dos operadores (funções) sucessor e predecessor. Então, pede-se para escrever uma função recursiva que calcule a soma de dois números x e y através desses dois operadores: sucessor e predecessor.

**Parte 2 - Conversão de bases (ESCOLHA 1 QUESTÃO PARA RESPONDER)**

7) Desenvolva um método que seja capaz de converter um número de 16 dígitos unsigned (sem sinal) de uma dada base x para outra y. Para isso, efetue os seguintes passos básicos:

1. Converter o número da base de entrada x para a base 10 e, por fim, converta-lo para a base y;
2. Certifique-se de que ambas as bases x e y estão definidas no intervalo [2, 16].
3. Para dígitos maiores que '9', crie um método que converta seus sucessores nas letras [A,B,C,D,E,F]. Com 'A' representando o valor 10 e assim por diante.

**Exemplo:** public String converterBase( String numero, int x, int y)

8) Desenvolva um método que recebe um número binário de 16 dígitos, na notação complemento de dois, e o converte para base decimal.  
Considere que os números de entrada são apenas inteiros, positivos ou negativos.