

<b>UNIFEI</b>	<b>Universidade Federal de Itajubá</b> Instituto de Engenharia de Sistemas e Tecnologias da Informação-IESTI
<b>Disciplina</b>	ECOP01 – Técnicas de Programação
<b>Professores</b>	Prof.Dr.Enzo Seraphim, Prof <sup>a</sup> .Dr <sup>a</sup> .Thatyana de Faria Piola Seraphim
<b>9ª Lista de exercícios</b>	

1) Faça uma função recursiva chamada **harmonico**, que recebe como parâmetro um número natural  $n$ . Esta função deve determinar e escrever o número harmônico  $H_n$  definido por:  $H_n = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$

2) Um matemático italiano da idade média conseguiu modelar o ritmo de crescimento da população de coelhos através de uma sequência de números naturais que passou a ser conhecida como Sequência de Fibonacci. O  $n$ -ésimo número da sequência de Fibonacci ( $F_n$ ) é dado pela seguinte formula:  $F_i = F_{i-1} + F_{i-2}$ , para  $i \geq 3$ . O resultado é a sequência {1, 1, 2, 3, 5, 8, 13, 21,...}. Escreva uma função recursiva chamada **fibonacci**, que recebe como parâmetro um número inteiro. Essa função deve determinar e retornar o valor do elemento de posição  $i$  na sequência de Fibonacci.

3) Escreva uma função recursiva chamada **potencia** que recebe dois parâmetros inteiros positivos: base e expoente. A função deve retornar a potência da base elevada ao expoente. Por exemplo: potencia(2, 3) é igual a 8.

4) Escreva uma função recursiva chamada **soma** que recebe como parâmetro um número inteiro positivo  $n$ . A função deve retornar a soma dos  $n$  primeiros números inteiros. Por exemplo, se a função receber  $n$  igual a 5, deve retornar 15, pois  $1 + 2 + 3 + 4 + 5 = 15$ .

5) Faça uma função recursiva chamada **divisores** que recebe como parâmetro um valor inteiro e positivo e imprime na tela os divisores desse valor.

6) Faça uma função recursiva chamada **primo** que recebe como parâmetro um inteiro positivo e, retorna verdadeiro se o inteiro é um número primo ou falso se contrário.

7) Faça uma função recursiva chamada **progSoma** que recebe como parâmetros: um número  $n$  (número de termos de uma progressão aritmética),  $a1$  (o primeiro termo da progressão) e  $r$  (a razão da progressão). A função deve retornar a soma dos elementos da progressão.

8) Faça uma função recursiva chamada **multiplicaMat** que faz a multiplicação de duas matrizes A e B de valores reais, sendo que o resultado da multiplicação deve estar em uma matriz C. A função deve imprimir na tela a matriz C.

9) Faça duas funções: uma chamada de **ordenacao** e outra de **buscaBinaria**. A função **ordenacao** não retorna nada e recebe como parâmetro um vetor de 1.000 inteiros. Para ordenar o vetor use o trecho de código abaixo. A função **buscaBinaria** recebe como parâmetro o vetor de inteiros ordenado pela função **ordenacao**, e um numero inteiro  $k$  qualquer. A função **buscaBinaria** deve retornar a posição no vetor onde está o elemento  $k$  ou -1 se o elemento não está no vetor. A função **buscaBinaria** deve-se ser recursiva e a procura do elemento é feita dividindo o vetor na metade a[n/2]. Se  $k$  for menor que a[n/2], o elemento deve ser procurado na metade da esquerda, senão na metade da direita. Esse processo deve continuar sucessivamente até encontrar ou não o elemento.

```
for(i=0; i<(n-1); i++){
    for(j=i+1; j<n; j++){
        if(vetor[i]>vetor[j]){
            aux=vetor[i];
            vetor[i]=vetor[j];
            vetor[j]=aux;
        }
    }
}
```

10) Um número  $a$  é dito permutação de um número  $b$  se os mesmos dígitos que formam  $a$ , também formam  $b$ , mesmo que em posições diferentes. Por exemplo, o número 5412434 é uma permutação de 4321445. Faça uma função recursiva chamada **permutação** que recebe um número inteiro e retorna todas as permutações do

<b>UNIFEI</b>	<b>Universidade Federal de Itajubá</b> Instituto de Engenharia de Sistemas e Tecnologias da Informação-IESTI
<b>Disciplina</b>	ECOP01 – Técnicas de Programação
<b>Professores</b>	Prof.Dr.Enzo Seraphim, Prof <sup>a</sup> .Dr <sup>a</sup> .Thatyana de Faria Piola Seraphim
<b>9ª Lista de exercícios</b>	

número inteiro. O conjunto das permutações dos inteiros de 1,...,n pode ser obtida através do conjunto das permutações dos inteiros de 1,...,n-1, inserindo **n** em cada possível posição de cada permutação.

11) Faça uma função recursiva chamada **palavraPalindromo** que recebe como parâmetro uma palavra de no máximo 20 caracteres. A função deve retornar verdadeiro se a palavra é um palíndromo ou não. Por exemplo, racificar é um palíndromo, pois, é a mesma palavra lida da esquerda para direita e da direita para esquerda. Os parâmetros da função devem ser definidos para garantir a execução recursiva. Para determinar o tamanho de uma palavra use a função `strlen()`;

12) Faça uma função recursiva chamada **contaRepetido** que recebe como parâmetros: um vetor de números inteiros não ordenados e três variáveis inteiras. A primeira e a segunda variáveis inteiras contém a primeira e a última posição do vetor. A terceira variável inteira contém um número que será contado. A função deve contar quantas vezes esse número se repete no vetor. Essa função retorna a quantidade de vezes que o número se repete no vetor. Não use laços de repetição, apenas chamadas recursivas.