

# UNIRIO - TÉCNICAS DE PROGRAMAÇÃO II

## LISTA DE EXERCÍCIOS 2

1. Faça um programa que, a partir de uma string digitada pelo usuário, imprima:
  - a) O número de caracteres da string.
  - b) A string com todas suas letras em maiúsculo.
  - c) O número de vogais da string.
  - d) Se a string digitada começa com "UNI" (ignorando maiúsculas/minúsculas).
  - e) Se a string digitada termina com "RIO" (ignorando maiúsculas/minúsculas).
  - f) O número de dígitos (0 a 9) da string.
  - g) Se a string é um palíndromo ou não.
2. Escreva um programa que dado um valor numérico digitado pelo usuário (armazenado em uma variável inteira), imprima cada um dos seus dígitos por extenso.

Exemplo:

```
Entre o número: 4571
Resultado: quatro, cinco, sete, um
```

3. Escreva um programa que, a partir de um nome informado pelo usuário, exiba suas iniciais.

As iniciais são formadas pela primeira letra de cada nome, sendo que todas deverão aparecer em maiúsculas na saída do programa. Note que os conectores *e, do, da, dos, das, de, di, du* não são considerados nomes e, portanto, não devem ser considerados para a obtenção das iniciais. As iniciais devem ser impressas em maiúsculas, ainda que o nome seja entrado todo em minúsculas.

Exemplos:

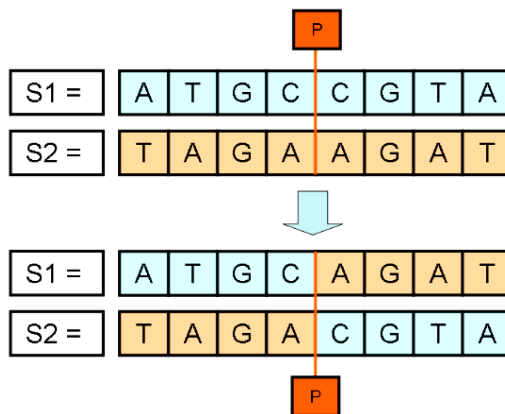
```
Maria das Graças Pimenta => MGP
João Carlos dos Santos => JCS
José da Silva => JS
Pedro Pereira Teixeira => PPT
```

4. Faça um programa que, a partir de um texto digitado pelo usuário, conte o número de caracteres total e o número de palavras (palavra é definida por qualquer sequência de caracteres delimitada por espaços em branco) e exiba o resultado.
5. Faça um programa que, a partir de um texto digitado pelo usuário, imprima o texto removendo todos os espaços em branco adicionais encontrados, de modo que haja, no máximo, um espaço em branco separando as palavras presentes nesse texto.

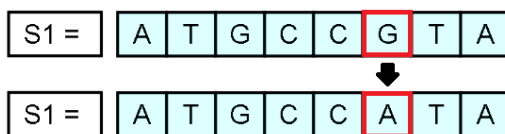
6. Um algoritmo genético é técnica de inteligência artificial que se inspira na forma como a natureza funciona e na teoria da evolução para resolver problemas complexos. Os algoritmos genéticos funcionando mantendo uma população de estruturas que evoluem de forma semelhante à evolução das espécies.

A recombinação e a mutação são os operadores básicos de um algoritmo genético.

Considerando que os indivíduos da população são codificados através de strings, o operador de recombinação recebe duas strings S1 e S2, sorteia aleatoriamente um ponto de corte em S1 e S2. Em seguida, realizada a troca de informações genéticas de S1 e S2, de forma a gerar dois novos indivíduos filhos. A figura abaixo ilustra o processo de recombinação.



O operador de mutação consiste em sortear aleatoriamente uma posição em um dos indivíduos filhos, e então modificar aleatoriamente o conteúdo daquela posição. A figura abaixo ilustra o processo de mutação.



Construa um programa em Java que implemente e utilize as funções `recombinacao` e `mutacao`:

- A função `recombinacao` recebe como parâmetro 4 Strings (S1, S2, S3 e S4), onde S1 e S2 são as Strings representado os indivíduos pais e S3 e S4 devem retornar os dois indivíduos filhos gerados com a recombinação de S1 e S2. Ambas as strings devem possuir o mesmo tamanho.
- A função `mutacao` recebe como parâmetro uma string S1 e modifica o conteúdo de S1 realizando o processo de mutação.

O seu programa deve ler duas strings S1 e S2 (de mesmo tamanho), fazer a recombinação dos dois indivíduos e aplicar uma mutação em um dos indivíduos resultantes. Por ultimo, o programa deve exibir os 4 indivíduos da população (S1, S2, S3 e S4).