

# Aula 02

## Algoritmos e Programação II

Agosto / 2017



Prof. Mario Liziér  
lizier@ufscar.br

# Strings

- Uma *string* é uma sequência ou cadeia de caracteres
- Em C, uma *string* é armazenada em um vetor com elementos do tipo *char*, mantendo a propriedade de que o caracter '\0' existe (ou é atribuído) logo após o último caracter da sequência
  - Temos sempre que garantir que o vetor possui espaço alocado para pelo menos um elemento a mais do que a maior sequência de caracteres que será armazenada!
  - Uma string sempre começa na posição 0 do vetor, e continuar a até o primeiro elemento '\0'
  - Podemos utilizar as funções auxiliares str\*
  - O tipo char armazena uma tabela pequena (ASCII)

# Exercícios

- Vamos fazer os exercícios propostos nos slides do prof. Tiago

- ✓ 1. Implemente as seguintes funções sem usar `string.h`:
  - `int myStrLen(str[ ])` - calcula o tamanho de uma *string*
  - `int myStrLenAlNum(str[ ])` - calcula a quantidade de caracteres alfanuméricos de uma *string*
  - `void myStrCat(str[ ], str1[ ], str2[ ])` - concatena `str1` e `str2` e coloca em `str`, separados por um espaço em branco.
  - `void str2maiusculo(str[ ])` - converte `str` para maiúsculo
  - `void str2minusculo(str[ ])` - converte `str` para minúsculo

# Exercício – Frequência de letras

- Construa um algoritmo que informe a quantidade de cada letra presente na *string*.

String: 

S	o	r	o	c	a	b	a	s	!	\0
---	---	---	---	---	---	---	---	---	---	----

Resposta: 

s	2
o	2
r	1
c	1
a	2
b	1

# Exercício – Busca em textos

- Considere uma *string* **P** de comprimento **m**
  - Uma *substring* **P**[i..j] de **P** é a subsequência de **P** formada pelo caracteres nas posições **i** até **j**, inclusive
  - Um prefixo de **P** é a *substring* do tipo **P**[0..i],  $i \leq m-1$
  - Um sufixo de **P** é uma *substring* do tipo **P**[i..m-1],  $i \geq 0$
- Sejam as *strings* **T** (texto de tamanho **n**) e **P** (padrão de tamanho **m**), o problema do reconhecimento de padrões em *string* consiste em encontrar as subsequências em **T**, que sejam iguais a **P**

T: 

C	T	C	C	A	T	T	C	G	A	G	A	T	C	G	C	\0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

P: 

T	C	G	\0								
---	---	---	----	--	--	--	--	--	--	--	--

# Exercício – Busca em textos

- Construa um algoritmo para resolver este problema!
  - Versão 1: faça uma função que retorne a posição do início do primeiro padrão presente no texto, ou -1
  - Versão 2: imprima na tela a posição do início de todos os padrões presentes no texto (ou uma mensagem informando que não foram encontrados).
  - Versão 3: Não faça chamadas printf de dentro de subprogramas.

# Exercício – Busca em textos

- Vamos contar quantas operações são realizadas quando executamos nosso programa?
  - Acesso ao vetor / comparações
- Em casa, procure sobre os algoritmos:
  - Boyer-Moore
  - Knuth–Morris–Pratt
- Tarefa:
  - Exercícios da Lista01

