

## Lista de exercícios 6 - Strings

\* O termo “uma função que receba como entrada dois valores inteiros” quer dizer que a função possui como parâmetro dois valores inteiros. Da mesma forma “a função retorna” indica a informação que deve ser retornada pelo comando *return*. Quando for requisitado que a função interaja com o usuário, será usado “a função recebe do usuário” (*scanf*, *fgets*) e “a função imprime na tela” (*printf*).

\* Cuidado para não testar suas funções utilizando como argumento strings literais. Ao invés de `funcao(“strings teste”);` utilize `str[] = “strings teste”;` `funcao(str);` para funções que modificam a string. Veremos a razão disso na aula sobre ponteiros.

1. A função `strcmp` definida em `string.h` considera que caracteres maiúsculos e minúsculos são diferentes. Por exemplo, a string “Casa” vem antes de “casa” no alfabeto, pois o código ASCII de ‘A’ é menor que o de ‘a’.  
Faça uma função que receba como entrada duas strings e retorne o resultado da função `strcmp` aplicada a essas duas strings, mas ignorando caracteres maiúsculos. Utilize a função `tolower` para transformar os caracteres em minúsculo.
2. Faça uma função, chamada `palindromo`, que receba como entrada uma palavra (string) e identifique se ela é um palíndromo, isto é, se a palavra lida ao contrário é igual à palavra original. A função retorna 1 em caso positivo e 0 caso contrário. Essa função foi feita na lista 3 sobre strings em Python
3. Faça uma função que receba como entrada um valor inteiro `n`. A função recebe do usuário `n` palavras e imprime na tela a maior palavra digitada.  
Dica: Crie um vetor `palavra_atual` para receber novas palavras do usuário, e um vetor `maior_palavra` para armazenar a maior palavra. Use uma variável do tipo `int` para armazenar o respectivo tamanho da maior palavra. Use a função `strcpy` para copiar palavras entre os vetores e `strlen` para calcular o tamanho das palavras.
4. Escreva uma função que receba como entrada uma string e verifique se para todo caractere ‘(’ (abre parênteses) há um caractere ‘)’ (fecha parênteses) respectivo. A função retorna 1 se os parênteses estiverem ok e 0 caso contrário.  
Dica: defina um contador e some +1 sempre que encontrar ‘(’ e -1 sempre que encontrar ‘)’. Se em algum momento o contador for negativo, a função retorna 0, pois há mais caracteres ‘)’ do que ‘(’. Se ao final da string o contador for positivo, também é retornado 0, pois um ou mais parênteses não foram fechados.
5. Faça uma função que receba como entrada um valor inteiro `n`. A função recebe do usuário os nomes completos e respectivas idades de `n` pessoas. A função então imprime na tela o nome da pessoa mais idosa. Não é necessário salvar os dados em vetores.
6. Escreva uma função que receba como entrada uma string e retorne o tamanho da maior palavra presente na string. Considere que a string possui somente palavras e espaços (sem números e pontuação).

Dica: Sempre que encontrar o caractere ' ' (espaço), reinicie um contador e faça contador++ até encontrar o próximo espaço. Armazene o maior valor do contador encontrado.

7. Faça uma função que receba como entrada quatro parâmetros

**texto**: string contendo um texto

**busca**: string contendo uma palavra

**subs**: string contendo outra palavra

**res**: vetor char no qual o resultado será inserido

A função copia o conteúdo de **texto** para **res**, mas substitui a primeira ocorrência da palavra **busca** por **subs**.

Essa função é similar com a que foi feita em aula, com a diferença que as palavras **busca** e **subs** podem ter tamanhos distintos.

Dica: Crie um índice para acessar os elementos da string **texto** e outro índice para acessar os elementos da string **res**, incrementando-os conforme necessário.