

Lista de exercícios 5

Strings

1. Faça uma função que recebe o nome, idade e endereço de uma pessoa. A função usa essas informações para criar uma nova string, estruturada da seguinte forma:
“<<nome>> possui <<idade>> anos e mora na rua <<endereço>>”
A função retorna a string criada.

Por exemplo, para o nome “Pedro”, idade “36” e endereço “Carlos Botelho, 1532”, a nova string será :

“Pedro possui 36 anos e mora na rua Carlos Botelho, 1532”

*Dica: Lembrem-se que podemos concatenar duas strings através do operador soma (string3 = string1 + string2)

2. Faça uma função que recebe como entrada uma string estruturada na forma “nome, nota”, a função retorna dois valores: o nome e a nota convertida para o tipo float.

Por exemplo, para a string “Eduardo Silva, 8.5” a função retorna “Eduardo Silva” e também o valor 8.5.

*Dica: Use o método .split()

3. Faça uma função que recebe como entrada uma string estruturada na forma “nome1, nota1; nome2, nota2; nome3, nota3; ...”. A função retorna duas listas: uma contendo os nomes e a outra contendo as notas dos alunos contidos na string.

Por exemplo, para a string “Eduardo Silva,8.5;Aline Cardoso,5.4;Pedro Silveira,6.2;Ana Clara, 7.3” a função retorna duas listas:

nomes: [“Eduardo Silva”, “Aline Cardoso”, “Pedro Silveira”, “Ana Clara”]

notas: [8.5, 5.4, 6.2, 7.3]

4. (Autolab) Faça uma função, chamada **ehPalindromo**, que recebe como entrada uma palavra (string) e identifica se ela é um palíndromo, isto é, se a palavra lida ao contrário é igual à palavra original. A função retorna **True** em caso positivo e **False** caso contrário.

Exemplos de palíndromos:

ele, erre, osso, reviver

Arquivos

5. Faça um programa que execute as seguintes tarefas:
 1. O programa requisita que o usuário digite o nome e as três notas que um aluno obteve durante o semestre
 2. A informação é salva em um arquivo do computador
 3. O programa continua requisitando nomes e notas, até que o usuário digite somente o caractere 's'.

*Salve o nome e as três notas na mesma linha do arquivo.

6. Faça um programa que inverte as linhas de um arquivo de texto. Isto é, a primeira linha se torna a última, a segunda se torna a penúltima, e assim por diante.
7. (Autolab) Faça uma função, chamada **infoArquivo**, que recebe como parâmetro o nome de um arquivo de texto. A função retorna valores inteiros indicando o número de linhas, número de vogais sem acento e o número de espaços em branco no arquivo, nessa ordem.

* Ao criar o arquivo de texto para testes, tome cuidado com espaços em branco “invisíveis”. Por exemplo, espaços no final de uma linha ou no final do arquivo.

8. Faça uma função que recebe como entrada dois valores inteiros **nl** e **nc**, um caractere **c**, e uma string **nomeArquivo**. A função escreve o caractere **c** em **nl** linhas e **nc** colunas de um arquivo chamado nomeArquivo. Por exemplo, para **nl=4**, **nc=6**, **c='@'** e nomeArquivo='arrobas.txt', a função deve escrever no arquivo arrobas.txt:

```
@ @ @ @ @ @
@ @ @ @ @ @
@ @ @ @ @ @
@ @ @ @ @ @
```

} nl

{ nc

9. Faça uma função que recebe como entrada um valor inteiro **nl**, um caractere **c**, e uma string **nomeArquivo**. A função escreve o caractere **c** em **nl** linhas do arquivo. O número de caracteres escritos em cada linha é igual ao índice da linha. Por exemplo, para **nl=5** e **c='@'**, o programa deve escrever:

```
@
@@
@@@
@@@@
@@@@@
```

10. Faça um programa que cria uma “bomba de arquivos”, isto é, dado um valor inteiro **n**, o programa cria **n** arquivos no computador.

11. (Autolab) Faça uma função, chamada **comparaArq**, que recebe como entrada o nome de dois arquivos de texto. A função retorna **True** caso os dois arquivos possuam ao menos uma linha idêntica, e **False** caso contrário.