

MC-102 — Aula 18

Matrizes e Vetores Multidimensionais

Eduardo C. Xavier

Instituto de Computação – Unicamp

18 de Novembro de 2020

Roteiro

- 1 Matrizes de Caracteres
 - Exemplo

Palavras transpostas

- Criar aplicação que le um determinado número de strings, e as interpreta como uma matriz.

```
Quantidade de strings: 3
string 0: frutas
string 1: legumes
string 2: vegetais
```

```
Matriz
frutas##
legumes#
vegetais
```

- Inserimos o caracter especial # para deixar todas strings com mesmo tamanho.
- O programa deve então imprimir as palavras formadas pelas colunas desta matriz

```
flv
ree
ugg
tue
amt
sea
#si
##s
```

Palavras transpostas

- Criamos uma função que recebe uma lista de palavras e devolve uma matriz de caracteres (com # incluídos caso necessário).

```
def create_matrix(l):  
    '''Recebe lista com strings e devolve uma matriz com  
    caracteres como elementos da matriz. # são adicionados  
    no final de cada string para deixar strings com o mesmo tamanho.  
    '''  
    mx = largest_string(l)  
    l = [s + '#'*(mx-len(s)) for s in l] #todas strings terao o mesmo tamanho  
    m = [list(s) for s in l] #m é uma matriz de caracteres das strings  
    return m
```

- A função acima usa uma outra função para determinar o tamanho da maior string na lista.

```
def largest_string(l):  
    '''Devolve o tamanho da maior string  
    em uma lista de strings'''  
    mx = len(l[0])  
    for s in l:  
        if len(s) > mx:  
            mx = len(s)  
    return mx
```

Palavras transpostas

- Criamos uma função para imprimir a matriz.

```
def print_matrix(m):  
    for l in m:  
        print(''.join(l))
```

- Lembre-se que o método **join** de uma string recebe uma lista de caracteres como parâmetro e devolve uma string com a junção destes caracteres.

Palavras transpostas

- Para imprimir as palavras das colunas da matriz podemos computar a sua transposta e então imprimir a matriz.
- A função abaixo computa a transposta.

```
def transpose(m):  
    '''Devolve a matriz transposta de m'''  
    l = len(m) #num. linhas de m  
    c = len(m[0]) #num. colunas de m  
    #m tem dimensoes lxc e m2 terá cxl  
    m2 = [[' ' for j in range(l)] for i in range(c)]  
    for i in range(l):  
        for j in range(c):  
            m2[j][i] = m[i][j]  
    return m2
```

Palavras transpostas

- Com as funções anteriores podemos criar um programa que executa a função **main** como abaixo:

```
def main():
    n = int(input('Quantidade de strings: '))
    l = []
    for i in range(n):
        l.append(input('string {}:'.format(i)))

    m = create_matrix(l)
    print('\nMatriz original\n')
    print_matrix(m)

    m2 = transpose(m)
    print('\nTransposta\n')
    print_matrix(m2)

main()
```