

## Exercícios para P2 – Parte 1- Turmas 33M e 33R

Ex1)

Escreva uma função, denominada `para_um_encontro`, que:

- receba uma lista com as 3 coisas favoritas de uma pessoa em busca de romance e uma lista de possíveis parceiros e suas respectivas 3 coisas favoritas,
- retorne 1 lista com os possíveis parceiros que tenham ao menos 2 coisas favoritas em comum.

É conveniente fazer uma função auxiliar que retorne quantas coisas comuns há em 2 listas de coisas favoritas

Escreva um programa completo para testar sua função.

```
lparc= [['lala', ['restaurante','bar','praia']], ['nena', ['cinema','leitura','praia']],  
        ['vivi', ['cinema','leitura','teatro']], ['lele', ['restaurante','bar','praia']],  
        ['nina', ['cinema','leitura','praia']], ['tita', ['cinema','leitura','teatro']],  
        ['leda', ['viagem','danca','esporte']], ['gege', ['praia','leitura','viagem']],  
        ['tati', ['cinema','leitura','dormir']], ['babi', ['viagem','cachorro','praia']],  
        ['tata', ['cachorro','leitura','praia']], ['zaza', ['cinema','gato','cachorro']]  
    ]
```

Por exemplo para `['praia','leitura','tv']` e a lista `lparc` acima seria retornada:  
`['nena', 'nina', 'gege', 'tata']`

Ex2):

Considere as listas `lHerois` e `lVotos` em que `lVotos[K]` é a quantidade de pessoas que escolheram `lHeroi[K]` como seu herói favorito. Assuma que as quantidades de votos são distintas.

Escreva uma função, denominada `cria_lista_ordenada_dec`, que receba duas listas como as descritas e construa e retorne uma nova lista, ordenada decrescentemente por `qtdVotos`, em que cada elemento é uma listinha `[heroi,qtdVotosDoHeroi]`.

As listas originais devem ficar vazias ao final da função.

A ideia de uma possível solução é determinar o máximo de votos da lista e o herói correspondente, retirando ambos de suas listas originais e adicionando um novo elemento no final da nova lista.

Repetir esse procedimento até que as listas originais fiquem vazias (basta testar uma delas)

Essa solução deve usar adequadamente:

- a função `len`, que recebe uma lista e retorna seu tamanho
- a função `max`, que recebe uma lista e retorna o maior valor da lista
- o método `index` de lista, que retorna a posição de um elemento na lista
- o método `pop` de lista ou o método `remove`, para fazer a retirada de elemento da lista
- o método `append`

Teste sua função com as listas abaixo:

```
lHeroi= ['batman','superman','wolverine','vampira','tempestade',  
        'xavier','hulk','flash','spyderman','venom']
```

```
lVotos=[ 2990, 2100, 3350, 1122, 1213, 2451, 2855, 1002, 2705, 1567]
```

Para as listas acima a lista retornada é:

```
 [['wolverine', 3350], ['batman', 2990], ['hulk', 2855], ['spyderman', 2705],  
  ['xavier', 2451], ['superman', 2100], ['venom', 1567], ['tempestade', 1213],  
  ['vampira', 1122], ['flash', 1002]]
```

Ex3)

Em um arquivo de nome apostas16junho.txt há o nome, a quantidade de apostas e as apostas de 6 números de jogadores da megasena, organizados da seguinte forma:

```
NOME_DO_JOGADOR_1,QTD_APOSTAS_JOG1
```

os seis números da primeira aposta do jogador1 separados por ,

os seis números da segunda aposta do jogador1 separados por ,

...

```
NOME_DO_JOGADOR_2,QTD_APOSTAS_JOG2
```

os seis números da primeira aposta do jogador2 separados por ,

...

Escreva um programa que simule um sorteio da megasena.

Para isso devem ser sorteados 6 numeros entre 1 e 60 sem repetição. Exiba os números sorteados.

Em seguida devem ser lidos os dados dos arquivos e informado, para cada jogador, cada aposta e quantos números ele acertou na aposta.

Por exemplo, caso o sorteio resulte em [56, 2, 14, 57, 47, 27] e com os dados do arquivo apostas16junho.txt, a saída seria:

Jogador: LALA

Aposta: ['12', '45', '32', '56', '44', '21'] -> 1

Aposta: ['13', '45', '42', '56', '44', '22'] -> 1

Aposta: ['13', '12', '42', '56', '44', '22'] -> 1

Jogador: BUBA

Aposta: ['32', '45', '32', '36', '44', '24'] -> 0

Aposta: ['14', '45', '52', '56', '44', '20'] -> 2

Aposta: ['32', '45', '32', '36', '15', '20'] -> 0

Aposta: ['14', '45', '52', '56', '15', '27'] -> 3

Jogador: ZEZE

Aposta: ['22', '45', '52', '56', '17', '18'] -> 1

Aposta: ['17', '45', '32', '35', '15', '13'] -> 0

Ex4)

Ex4A) Escreva uma função que receba um curso e um nome de arquivo com os dados dos alunos, um aluno por linha, e exiba nome e CR de todos os alunos do curso recebido. Os dados dos alunos em uma linha estão separados por , e são: o nome completo do aluno, o curso do aluno e o CR do aluno.

Ex4B) Escreva uma função que receba um valor numérico X e um nome de arquivo com os dados dos alunos, um aluno por linha, e exiba nome, curso e CR de todos os alunos com CR > X.

Os dados dos alunos em uma linha estão separados por , e são: o nome completo do aluno, o curso do aluno e o CR do aluno.

Ex4C) Escreva uma função que receba um curso e um nome de arquivo com os dados dos alunos, um aluno por linha, e grave em um arquivo de saída nomeado com o nome do curso +.txt os dados de todos os alunos do curso (um aluno por linha, somente nome e CR separados por , )

Ex4D) Escreva um programa completo para testar os itens A, B e C.

Obs: para facilitar será fornecido um arquivo de dados alunosecursos.txt para seus testes.