

```
''' Proposta de Resolucao: Troca de Cartas '''

def Troca(dicionario):
    meninal, menina2 = dict.keys(dicionario)
    repetidas = {meninal:[], menina2:[]}
    trocaveis = {meninal:[], menina2:[]}

    for menina, lista_cartas in dicionario.items(): # percorre o dicionario
original
        for carta in lista_cartas: # percorre lista associada a chave
correspondente a uma menina
            if lista_cartas.count(carta) > 1 and carta not in repetidas[menina]: #
verifica se a carta eh repetida e ainda nao foi adicionada
                repetidas[menina].append(carta)#adiciona carta repetida no
dicionario 'repetidas',na chave correspondente a uma das meninas

            for chave, cartas in repetidas.items(): # percorre o dicionario de repetidas
                for c in cartas:
                    if chave == meninal:
                        if c not in dicionario[menina2]: # verifica se carta repetida da
menina 1 nao existe nas cartas da menina 2
                            trocaveis[meninal].append(c)
                        elif chave == menina2:
                            if c not in dicionario[meninal]: # verifica se carta repetida da
menina 2 nao existe nas cartas da menina 1
                                trocaveis[menina2].append(c)

    return min(len(trocaveis[meninal]),len(trocaveis[menina2])) #retorna o menor
valor dentre as cartas trocaveis de ambas as meninas

''' Proposta de Resolucao: Troca de Cartas '''

def Troca2(dicionario):
    meninal, menina2 = dict.values(dicionario)

    repetidas_m1 = []
    repetidas_m2 = []
    trocaveis_m1 = []
    trocaveis_m2 = []

    for carta in meninal: # percorre lista associada a chave correspondente a
primeira menina
        if meninal.count(carta) > 1 and carta not in repetidas_m1: # verifica se a
carta eh repetida e ainda nao foi add a lista de repetidas
            repetidas_m1.append(carta)#adiciona carta repetida na lista de
'repetidas' correspondente a primeira menina

    for carta in menina2: # percorre lista associada a chave correspondente a
segunda menina
        if menina2.count(carta) > 1 and carta not in repetidas_m2: # verifica se a
carta eh repetida e ainda nao foi add a lista de repetidas
            repetidas_m2.append(carta)#adiciona carta repetida na lista de
'repetidas' correspondente a segunda menina

    for carta in repetidas_m1: # percorre a lista de repetidas da primeira menina
        if carta not in menina2: # verifica se carta repetida da menina 1 nao
existe nas cartas da menina 2
            trocaveis_m1.append(carta)

    for carta in repetidas_m2: # percorre a lista de repetidas da segunda menina
        if carta not in meninal: # verifica se carta repetida da menina 2 nao
existe nas cartas da menina 1
            trocaveis_m2.append(carta)

    return min(len(trocaveis_m1), len(trocaveis_m2)) #retorna o menor valor dentre
```

as cartas trocaveis de ambas as meninas

```
''' Proposta de Resolucao: Ordena Notas Escolares '''
def selectionSort(lista):

    for i in range(len(lista)-1): # percorre do primeiro elemento (de indice 0) ate
o penultimo elemento (de indice len-1)
        for k in range(i+1, len(lista)): # percorre do elemento seguinte ao elemento
atual ate o ultimo elemento (de indice len)
            if lista[i] > lista[k]: # compara se o elemento sucessor de indice k eh
menor do que o elemento atual
                temp = lista[i]
                lista[i] = lista[k]
                lista[k] = temp
            print 'iteracao(', i, '-', k, '):', lista

    return lista

''' Outra Proposta de Resolucao: Ordena Notas Escolares '''
def selectionSort2(lista):

    for i in range(len(lista)-1): # percorre do primeiro elemento (de indice 0) ate
o penultimo elemento (de indice len-1)

        menor = i # guarda o indice do elemento atualmente sendo verificado

        for k in range(i+1, len(lista)): # percorre do elemento seguinte ao elemento
atual ate o ultimo elemento (de indice len)
            if lista[k] < lista[menor]: # compara se o elemento sucessor de indice k
eh menor do que o elemento atual
                menor = k # se o if for True, menor passa a guardar o indice do
elemento sucessor, de indice k

        # as 3 linhas seguintes fazem a troca entre o elemento atual (i) e o seu
menor elemento sucessor (menor), se houver
        temp = lista[menor]
        lista[menor] = lista[i]
        lista[i] = temp
        print 'iteracao(', i, '):', lista

    return lista

def main():
    meninas = {"Alice": [1, 1, 2, 3, 5, 7, 8, 8, 9, 15], "Beatriz": [2, 2, 2, 3, 4,
6, 10, 11, 11]}
    print "Alice =", meninas["Alice"]
    print "Beatriz =", meninas["Beatriz"]
    print "\n"

    print "Quantidade trocas a ser feitas (funcao Troca):", Troca(meninas)
    print "Quantidade trocas a ser feitas (funcao Troca2):", Troca2(meninas)
    print "\n"

    lista = [20, 12, 10, 15, 2]
    print "Lista a ser ordenada:", lista
    print 'resultado ordenado (funcao SelectionSort):', '\n', selectionSort
(lista), '\n'
    print '-----', '\n'

    lista = [20, 12, 10, 15, 2]
    print "Lista a ser ordenada:", lista
    print 'resultado ordenado (funcao SelectionSort2):', '\n', selectionSort2
(lista), '\n'
```

```
    print '.....', '\n'
```

```
if __name__ == "__main__":  
    main()
```