

teste_Akaer

June 21, 2021

0.1 exercício 1

```
[ ]: n = int(input("Número de peças: "))
nv = [0]*(n+1)

ent = str(input("Id das peças: "))
ent = ent.replace(" ", "")

for i in ent:
    nv[int(i)] = int(i)

for i in range(1, len(nv)):
    if nv[i] == 0:
        print(i)
```

Número de peças: 4
Id das peças: 2 4 3
1

0.2 exercício 2

```
[ ]: N = int(input("Número de botas individuais entregues: "))

while N>0:
    cont = 0

    # D, E: arrays para controle de paridades
    D = [False]*61
    E = [False]*61

    while N:
        N = N - 1 # para index
        ML = str(input("TAM LADO (ex. 38 D): "))
        ML = ML.replace(" ", "")
        M = int(ML[0]+ML[1])
        L = ML[2]

        if L == 'E':
```

```

        E[M] = True

    else:
        D[M] = True

    if E[M] and D[M]:
        cont+=1
        E[M], D[M] = False, False

print("Pares que podem ser formados:", cont)
N = int(input("Número de botas individuais entregues: "))

```

```

Número de botas individuais entregues: 4
TAM LADO (ex. 38 D): 40 D
TAM LADO (ex. 38 D): 41 E
TAM LADO (ex. 38 D): 41 D
TAM LADO (ex. 38 D): 40 E
Pares que podem ser formados: 2
Número de botas individuais entregues: 6
TAM LADO (ex. 38 D): 38 E
TAM LADO (ex. 38 D): 38 E
TAM LADO (ex. 38 D): 40 D
TAM LADO (ex. 38 D): 38 D
TAM LADO (ex. 38 D): 40 D
TAM LADO (ex. 38 D): 37 E
Pares que podem ser formados: 1
Número de botas individuais entregues: 0

```

0.3 exercício 3

```

[ ]: # funcao de ordenação quicksort
# criada por mim na disciplina de estrutura de dados 2 (em 2020)

def quicksort(lista, inicio = 0, fim = None):
    if (fim == None):
        fim = len(lista)-1
    if (inicio < fim):
        p = partition(lista, inicio, fim)
        quicksort(lista, inicio, p-1)
        quicksort(lista, p+1, fim)

def partition(lista, inicio, fim):
    pivot = lista[fim]
    i = inicio
    for j in range(inicio, fim):
        if (lista[j] <= pivot):
            lista[j], lista[i] = lista[i], lista[j]

```

```

        i = i + 1
        lista[i], lista[fim] = lista[fim], lista[i]
        return i

t = int(input("Número de testes: "))
tv = [0]*t

for cont in range(t):
    values = input("\nEntre com valores de [n c p]: ")
    n, c, p = (int(num) for num in values.split(' '))
    values = input("Andares a serem visitados: ")
    andares = [0]*p

    i = 0
    for num in values.split(' '):
        andares[i] = int(num)
        i += 1

    quicksort(andares) # ordena para valores proximos ficarem lado a lado

    aux, res = 0, 0
    while True:
        if c > p:
            res = res + max(andares[aux:])*2
            break
        if c == p:
            res = res + max(andares[aux:c])*2
            break
        else:
            res = res + max(andares[aux:c])*2
            aux = c
            c += c

    print("Energia gasta:", res)

```

Número de testes: 3

Entre com valores de [n c p]: 10 1 3

Andares a serem visitados: 1 2 3

Energia gasta: 12

Entre com valores de [n c p]: 100 2 4

Andares a serem visitados: 10 10 10 3

Energia gasta: 40

Entre com valores de [n c p]: 100 2 5

Andares a serem visitados: 100 1 100 1 100
Energia gasta: 402

0.4 exercício 4

```
[24]: while True:
    val, n = map(str, input().split()) # outra forma de atribuir os valores

    if val == '0' and n == '0':
        break

    n = list(n)
    res, lista = [], []

    for i in range(len(n)):
        if val != n[i]:
            lista.append(int(n[i]))

    aux = 0

    for i in lista:
        if aux == 0:
            if i != 0:
                res.append(i)
                aux+=1
        else:
            res.append(i)
            aux+=1

    if sum(lista)==0:
        print('0')
    else:
        for i in res:
            print(i, end=" ".strip())
        print()
```

```
5 5000000
0
3 123456
12456
9 23454324543432
23454324543432
9 9999999999919999
1
7 777
0
0 0
```