PIG-APX1-2020/2

NOME: ANDRÉ LUIS ALVES SILVEIRA MAT.: 17113050209

```
#!/usr/bin/env python
# coding: UTF-8
#
# @author: Andre Silveira
# @date: 25/09/2020
import sys
class Particao:
    # Define a partição de um todo em partes.
    def __init__(self,*p):
        # Define a partição de um todo (somatório dos elementos de uma
        # lista de números) em partes.
        self.__todo = 1
        self.__sum = 0
        self.__numbers = [ ]
        for number in range(len(p)):
            self.__sum += p[number]
        for number in range(len(p)):
            self.__numbers.append((p[number]/self.__sum)*self.__todo)
    @property
    def todo(self):
        # Retorna o valor do todo.
        return self.__todo

    @todo.setter
    def todo(self,t):
        # Passa a assumir que o tamanho do todo é t, do invés de 1.
        self.__todo = t
        for number in range(self.__len__()):
            self.__numbers[number] *= self.__todo

    def __len__(self):
        # Retorna o número de partes.
        return len(self.__numbers)
```

```
def __eq__(self, other):
    # Retorna se duas partições são iguais.
    if self.__numbers == other.__numbers:
        print(True)
        return True
    else:
        print(False)
        return False

def __getitem__(self, key):
    # Retorna o tamanho da key'ésima parte.
    return self.__numbers[key]

def __setitem__(self, key, val):
    # Modifica o tamanho da key'ésima parte.
    for number in range(self.__len__()):
        self.__numbers[number] /= self.__todo
    self.__numbers[key] = float(val)
    self.__sum = 0
    for number in range(self.__len__()):
        self.__sum += self.__numbers[number]
    for number in range(self.__len__()):
        self.__numbers[number] = (self.__numbers[number]/self.__sum)*self.__todo

def __repr__(self):
    # Retorna uma STRing com conteúdo detalhado da partição.
    st = ""
    for number in range(self.__len__()):
        if(st == ""):
            st += str(self.__numbers[number])
        else:
            st += " "
            st += str(self.__numbers[number])
    st += ": len = "
    st += str(self.__len__())
    st += ", todo = "
    st += str("%.6f" % self.__todo)
    return st
```

```
    def __str__(self):
        # Retorna uma string com o conteúdo da partição.
        st = ""
        for number in range(self.__len__()):
            if (st == ""):
                st += str(self.__numbers[number])
            else:
                st += " "
                st += str(self.__numbers[number])
        return st

def main():
    p = Particao(*[2,5,3])
    print("[%s]" % str(p).replace(' ', ', '))
    p.todo = 100
    print("%r" % p)
    p = Particao(5,6)
    print("%r" % p)
    p.todo = 11
    print(repr(p))
    q = Particao(5,6)
    print(q)
    print(p == q)
    print("({}, {})".format(p[0], p[1]))
    p[1] = 7
    print(f"({p[0]}, {p[1]})")

if __name__ == "__main__":
    sys.exit(main())
```