Universidade Federal de Goiás
Escola de Engenharia Elétrica, Mecânica e de Computação
Aluno: Edson Júnior Frota Silva Matrícula: 201515412
Engenharia de Computação

Laboratório de Inovação e Automação 1 Prof.: Cássio Vinhal

Exercícios referentes ao capítulo 12:

**Exercicio_12.01:**

```
fin = open('/root/words.txt')
def most_frequent(s):
    aux = dict()
    for key in up:
        if key.isalpha():
            aux[key] = aux.get(key, 0) + 1
    kO = []
    for letter, count in aux.items():
        kO.append((count, letter))
    kO.sort(reverse = True)
    return  kO
up = fin.read()
x = most_frequent(up)
for y in x:
    print(y[1])
```

**Exercicio_12.02:**

```
fin = open('/root/words.txt')
def load_dict():
    aux = dict()    for line in fin:
        word = line.strip()
       kO = ''.join(sorted(word))
        aux[word] = kO
    return aux
def anagrams(aux):
    anag_dict = dict()
    l = []
    for join, x in aux.items():
        if x in anag_dict:
            anag_dict[x].append(join)
        else:
            anag_dict[x] = [join]
    y.sort(key = len, reverse=True)
    return y
aux = load_dict()
up = anagrams(aux)
for z in up:
    print(z)
```

**Exercicio_12.03:**

```
fin = open('/root/words.txt')
```

```
def load_dict():
    d = dict()
    for line in fin:
        word = line.strip()
        t = ''.join(sorted(word))
        d[word] = t
    return d
def anagrams(d):
    anag_dict = dict()
    l = []
        for k, v in d.items():
        if v in anag_dict:
            anag_dict[v].append(k)
        else:
            anag_dict[v] = [k]
    l.sort(key = len, reverse = True)
    return l
def get_diference(w1, w2):
    i = 0
    for c1, c2 in zip(w1, w2):
        if c1 != c2:
            i += 1
    return i
def get_metateses(anagrams):
    l = []
    for words in anagrams:
        for word1 in words:
            for word2 in words:
                d = get_diference(word1, word2)
                if d == 2 and word1 < word2:
                    l.append((word1, word2))
    return l
d = load_dict()
a = anagrams(d)
m = get_metateses(a)
for e in m:
    print(e)
```

**Exercicio_12.04:**

```
fin = open('/root/words.txt')
def load_dict():
    d = dict()     for line in fin:
        word = line.strip()
        d.setdefault(word)
    return d
def children(word, d):
    b = False
    t = []
    for e in word[1:]:
        w = list(word[1:])
        w.remove(e)
        w.insert(0, word[0])
        w = ''.join(w)
        if w in d:
            t.append(w)
            b = True
```

```
    if b:
        t.sort()
        return t
    return b
def load_children(d):
    all = dict()
    for k in d:
        temp = children(k, d)
        if temp:
            all[k] = temp
    return all
def print_all(d):
    for k, v in d.items():
        print(k, v)
def print_longest(d):
    words = list(d.keys())
    words.sort(key = len, reverse=True)
    b = words[0]
    print(b)
    print(d[b])
d = load_dict()
red = load_children(d)
op = int(input('Insira 1 para ver todas as palavras e seus filhos e 2
para a maior palavra e seus filhos'))
if op == 1:
    print_all(red)
elif op == 2:
    print_longest(red)
```

**Edson Júnior Frota Silva**
**Engenharia de Computação - UFG**