

Introdução à Programação (CC,ERSI) 2009/2010 – exame, 2ª parte

1. Seja *li* uma lista de inteiros. Escreva uma função `positivos(li)` que retorna uma nova lista constituída pelos inteiros existentes em *li* que são positivos. Exemplos:

```
positivos([3,-1,9,0,11,-1]) -> [3,9,11]
positivos([0,-1,0,0,-1])    -> []
```

```
def positivos(li):
    r=[]
    for x in li:
        if x>0:
            r.append(x)
    return r
```

2. Escreva uma função `mcm(m,n)` que retorna o mínimo múltiplo comum aos inteiros positivos *m* e *n*. *Deverá seguir obrigatoriamente o seguinte método: testar todos os inteiros m, 2m, 3m, ... até se encontrar um inteiro que seja (também) múltiplo de n. Exemplo: mcm(15,18) -> 90 (inteiros considerados neste exemplo: 15, 30, 45, 60, 75 e 90).*

```
def mcm(m,n):
    t=m
    while t%n!=0:
        t=t+m
    return t
```

3. Implemente uma função `mesmas(s,t)` que retorna

- True, se as letras minúsculas que ocorrem na string *s* são as mesmas que ocorrem na string *t*.
- False, caso contrário.

Exemplos:

```
mesmas("as batatas!!!","bt: assas OS PERUS?") -> True
mesmas("as batatas!!!","Bt: assa OS PERUS")    -> False
mesmas("!!!","BT: ???")                        -> True
```

Note que o número de ocorrências das letras minúsculas em *s* e em *t* não é importante; o que interessa é se ocorrem ou não. **Descreva resumidamente a estratégia usada para resolver este exercício.**

```
def mesmas(s,t):
    return incluído(s,t) and incluído(t,s)

def incluído(s,t):
    letras="abcdefghijklmnopqrstuvwxy"
    for c in s:
        if c in letras and not(c in t):
            return False
    return True
```

Outra solução (mais eficiente):

```
def mesmas2(s,t):
    letras="abcdefghijklmnopqrstuvwxy"
    ms=[c for c in letras if c in s] # minúsculas por ordem em 's'
    mt=[c for c in letras if c in t] # minúsculas por ordem em 't'
    return ms==mt
```