```
# 1 ------------------------ TP07 ------------------------

def conta_c(s,v):
    c=0
    for a in s:    # vê-se se cada um dos carateres de s...
        if a in v: # ... existe em v
            c = c+1
    return c

# Teste
print conta_c("verdes anos","aes")

#--- Exercícios extra
# 1) Escreva uma função "lista(s)" que retorna a lista dos caracteres
#    presentes na string "s", sem repetição. Exemplo
#    lista("As batatas!!!") -> ["A","s"," ","b","a","t","!"]
# 2) Implemente outa vez a função "conta_c", sem usar o predicado "in".

# 2 ------------------------------------------------------

#23456789
def sub_x(s):
    r=""
    for c in s:
        if not c in "aeiouAEIOU ":
            r=r+"p"
        else:
            r=r+c
    return r

# Teste
print sub_x("Pera Banana")

#--- Exercícios extra
# 1) Descreva o efeito da chamada "sub_x(s)", se retirarmos as
#    linhas "else:" e "r=r+c"
# 2) Descreva o efeito da chamada "sub_x(s)", se a instrução
#    "return r" for deslocada ara a direita por forma a
#    iniciar-se na colona 9 (alinhada com o if/else).
```

```
# 3 ------------------------------------------------------

def pares(n):
    li=[]
    for i in range(1,n):         # 1,   2,...,   n-1
        for j in range(i+1,n+1): # i+1, i+2,..., n
            li.append((i,j))
    return li

# Teste
print pares(4)

#--- Exercício extra.
# 1) Qual o comprimento la lista obtida (número de tuplos) como
#    função de n?

# 4 ------------------------------------------------------

def unica(li):
    a=sorted(li)
    r=[]
    for i in range(len(a)):
        if i==0 or a[i]!=a[i-1]:
            r.append(a[i])
    return r

# Teste
print unica([2,3,5,2,5,1,2,2])

#--- Exercício extra.
# 1) Implemente a função "unica(li)" sem usar "sorted", mas
#    podendo usar o predicado "in".
```

```
# 5 ------------------------------------------------------

def cp(a):
    r=[]
    if type(a)==int:
        return a
    for x in a:
        r.append(cp(x))
    return r

# Teste
a=[2, [3, 4], [[5]]]
c=cp(a)
b=a
a[2][0]=1000
print a
print b
print c
#-------------------------------------------------------
```