

joão Costa 199088 j8

## Parte 2

1) c) def numero\_digito\_pares(n):

res = 0

while n ≠ 0:

i = n % 10

n = n // 10

if i % 2 == 0:

res += 1

return res

1) a) def numero\_digito\_pares(n):

if n == 0:

return 0

i = n % 10

if i % 2 == 0:

return numero\_digito\_pares(n // 10) + 1

else:

return numero\_digito\_pares(n // 10)

1) b) def numero\_digito\_pares(n):

def numero\_digito\_pares\_aux(n, res)

if n == 0:

return res

i = n // 10

if i % 2 == 0:

return numero\_digito\_pares\_aux(n // 10, res + 1)

else:

return numero\_digito\_pares\_aux(n // 10, res)

return numero\_digito\_pares\_aux(n, 0)

joão Gato 190088 88

- 2) def caps(c\_p, p):  
    res = 49  
    for p in p:  
        for polerra in p[p]:  
            for c in c\_p:  
                if c\_p[c][0] ≤ polerra ≤ c\_p[c][1]  
                    if c not in res:  
                        res[c] = [p]  
                    else res[c] += [p]  
    return res
- 3) a) def codifica\_polerra(frase):     n-ésimo-primo(n) → tº primo  
    res, num = 1, 1  
    for i in frase:  
        res = res \* (n-ésimo-primo(n) \*\* cord(i))  
    return res
- b) def decodifica\_polerra(n):     n = 2, 3, 5, 7  
    v, t, l = 1, 0, [0]  
    while n != 1:  
        while n % n-ésimo-primo(v) == 0  
            n = n // n-ésimo-primo(v)  
            if len(l) != v:  
                l[v] += [t]  
            else:  
                l[v-1] += 1  
            v += 1  
    return "2\*\*" + str(l[0]) + "3\*\*" + str(l[1]) + ...