### Área do Triângulo

def area\_triang( a, b, c ):

p = (a+b+c) / 2

return (p\*(p-a)\*(p-b)\*(p-c)) \*\* 0.5

### Prestação mensal

def prestacao( P, i, n ):

return P \* (i \* (1+i)\*\*n) / ((1+i)\*\*n – 1)

### Suavização de pedaço da imagem

soma = 0

for i in range( 0, n ):

for j in range( 0, n ):

soma = soma + M[i][j]

media = soma // (n \* n)

for i in range( 0, n ):

for j in range( 0, n ):

M[i][j] = media

### Suavização de imagem completa

m, n = M.shape

for i in range( 0, m, 3 ):

for j in range( 0, n, 3 ):

soma = 0

for i1 in range( i, i+3 ):

for j1 in range( j, j+3 ):

soma = soma + M[i1][j1]

media = round( soma / 9 )

for i1 in range( i, i+3 ):

for j1 in range( j, j+3 ):

M[i1][j1] = media

### Soma de duas matrizes

Leia m, n

para i = 0 até m-1

para j = 0 até n-1

Leia a[i][j]

para i = 0 até m-1

para j = 0 até n-1

Leia b[i][j]

para i = 0 até m-1

para j = 0 até n-1

Escreva a[i][j] + b[i][j]

Escreva fim\_de\_linha

### Função para somar dois vetores

def somaVetores( A, B ):

n = len( A )

C = np.empty( n )

for i in range( 0, n ):

C[i] = A[i] + B[i]

return C

### Função para somar matrizes

def somaMatrizes( A, B ):

m, n = A.shape

C = np.empty( (m,n) )

for i in range( 0, m ):

for j in range( 0, n ):

C[i][j] = A[i][j] + B[i][j]

return C

### Última ocorrência de caractere em string

print("Escreva um texto qualquer:")

texto = input()

n = len( texto ) *# n = quantidade de caracteres digitados*

c = input("Entre com o caractere a procurar: ")

i = n-1

while i >= 0 and texto[i] != c:

i = i - 1

if i >= 0:

print("A última ocorrência de '", c, "' é na posição ", i, sep="")

else:

print("Caractere '", c, "' não encontrado", sep="")

### Função para detectar retas paralelas

def retas\_paralelas( a1, b1, c1, a2, b2, c2 ):

if b1 == 0 and b2 == 0 or b1 != 0 and b2 != 0 and a1/b1 == a2/b2:

return True

else:

return False