

Fundamentos de Programação - 2020/2021 Aula Prática 09 (30 minutos) Turno 2ª feira 09:00-10:30 e 10:30-12:00	
Nome:	
Número:	
Data:	
Curso:	

Utilizando funcionais sobre listas, escreva a função quadrado_impares, que recebe uma lista de inteiros positivos e devolve uma outra lista com o quadrado dos inteiros ímpares. A sua função deve conter apenas uma instrução, a instrução return. Por exemplo:



Aula Prática 09 (30 minutos) Turno 5ª feira 10:30-12:00			
Nome:			
Número:			
Data:			
Curso:			

Utilizando os funcionais sobre listas escreva a função produto_multiplos_lista, que recebe uma lista de números e um número e devolve o produto dos elementos da lista que são múltiplos do segundo argumento. A sua função deve conter apenas uma instrução, a instrução return. Por exemplo:



Aula Prática 09 (30 minutos) Turno 5ª feira 14:00-15:30				
Nome:				
Número:				
Data:				
Curso:				

Utilizando os funcionais sobre listas escreva a função conta_multiplos_numero, que recebe dois números inteiros n e d, e devolve o número de inteiros menores que n que são múltiplos de d. A sua função deve conter apenas uma instrução, a instrução return. Por exemplo:

list(range(1, num)))))



Aula Prática 09 (30 minutos) Turno 6ª feira 10:00-11:30	
Nome:	
Número:	
Data:	
Curso:	

Utilizando funcionais sobre listas escreva uma função de ordem superior, conta_p, que recebe um número inteiro positivo n e um predicado unário p, e devolve o número de inteiros positivos menores ou iguais a n que satisfazem p. A sua função deve conter apenas uma instrução, a instrução return. Por exemplo:

```
>>> conta_p(87, lambda x: x % 100 == 0)
0
>>> conta_p(487, lambda x: x % 100 == 0)
4

Solução 1:

def conta_p(num, pred):
    return len(filtra(pred, [x for x in range(1, num+1)]))
```



Aula Prática 09 (30 minutos) Turno 6ª feira 14:00-15:30			
Nome:			
Número:			
Data:			
Curso:			

Utilizando funcionais sobre listas, escreva a função num_pred_digitos, que recebe um predicado *pred* e uma lista *l* de dígitos e devolve o número composto pelos dígitos de *l* que satisfazem o predicado. Pode assumir que a lista *l* tem pelo menos um dígito que satisfaz o predicado *pred*. A sua função deve conter apenas uma instrução, a instrução return. Por exemplo:

```
>>> num_pred_digitos(lambda x : x > 3, [1, 2, 3, 4, 5])
45
>>> num_pred_digitos(lambda x : x > 3, [7, 2, 6, 4, 3])
764

Solução 1:
def num_pred_digitos(pred, l):
    return acumula(lambda x, y: 10*x + y, filtra(pred, l))

Solução 2:
def num_pred_digitos(pred, l):
    return int(acumula(lambda x, y: x+y, transforma(str, filtra(pred, l)))
```



Fundamentos de Programação - 2020/2021 Aula Prática 09 (30 minutos) Turno 6ª feira 15:30-17:00		
Nome:		
Número:		
Data:		
Curso:		

Utilizando funcionais sobre listas, escreva a função concatena, que recebe um predicado *pred* e uma lista *l* e concatena todas as listas em *l* que satisfazem o predicado. Pode assumir que a lista *l* não é vazia e só tem listas como elementos. A sua função deve conter apenas uma instrução, a instrução return. Por exemplo:

```
>>> concatena(lambda l : len(l) >= 2, [[1], [2, 3], [4, 5]])
[2, 3, 4, 5]
>>> concatena(lambda l : 2 in l, [[7, 2], [6, 4, 3], [5, 2]])
[7, 2, 5, 2]

Solução:

def concatena(pred, l):
    return acumula(lambda x,y : x + y, filtra(pred, l))
```