1.c

def double(lst):

return list(map(lambda x:x\*2, lst))

2.

a.

def flatten(lst):

return [smallItem for item in lst for smallItem in item]

b.

def flatten(lst):

if(len(lst) == 0):

return []

return lst[0] + flatten(lst[1:])

c.

from functools import reduce

def flatten(lst):

return reduce(lambda lst1,lst2:lst1+lst2 ,lst)

3.

a.

def lessThan(n, lst):

return [item for item in lst if(item < n)]

b.

def lessThan(n, lst):

if(len(lst) == 0):

return []

if(lst[0] < n):

return [lst[0]] + lessThan(n, lst[1:])

else:

return lessThan(n, lst[1:])

c.

def lessThan(n, lst):

return list(filter(lambda item: item<n, lst))

4.

b.

from functools import reduce

def increase(x):

return x+1

def square(x):

return x\*\*2

def double(x):

return x\*2

#trả về hàm gộp của 2 hàm

def compose2(func1, func2):

return lambda x: func1(func2(x))

#thực hiện chức năng gộp 2 hàm cho hết list funcs

def compose(\*funcs):

return reduce(compose2, funcs)