

```

# mylist, recursive with list

def head(a):
    return a[0]

def tail(a):
    b = list(a)
    c = b.pop(0)
    return b

def islist(a):
    return isinstance(a,list)

def cons(a,b):
    b.insert(0,a)
    return b

def enu(a):
    if( a == [] ):
        return
    print(head(a))
    enu(tail(a))

def mylen(a):
    if( a == [] ):
        return 0
    return 1 + mylen(tail(a))

def mylen2(a):
    if( a == [] ):
        return 0
    if( not islist(head(a)) ):
        return 1 + mylen2(tail(a))
    return mylen2(head(a)) +
mylen2(tail(a))

def rev(a):
    return rev2(a,[])

def rev2(a,b):
    if( a == [] ):
        return b
    return rev2(tail(a),
cons(head(a),b))

def copy(a):
    if( a == [] ):
        return a
    return cons(head(a), copy(tail(a)))

def subs(a,b,c):
    e = subs2(a,b,c,[])
    return rev(e)

def subs2(a,b,c,d):
    if( c == [] ):
        return d
    if( head(c) == a ):
        return subs2(a,b, tail(c),
cons(b,d))
    return subs2(a,b, tail(c),
cons(head(c),d))

```

```

def subs3(a,b,c):
    if( c == [] ):
        return c
    if( head(c) == a ):
        return cons(b,
subs3(a,b,tail(c)))
    return cons(head(c),
subs3(a,b,tail(c)))

def sum(a):
    return sum2(0,len(a)-1,a)

def sum2(i,end,a):
    if( i == end ):
        return a[i]
    return a[i] + sum2(i+1,end,a)

def main():
    a = [1,2,3,4]
    print(sum(a))

main()

```

Homework

1. write a printeach() of a complex list.
2. given a list of list, write a program to produce a list of the size of each sub-list.
lenoflist([[1,2,3],[4,5]]) is [3,2]
3. write a program to produce a new list which each element is double of the input list. double([1,2,3]) --> [2,4,6]
4. Here is a difficult one, write a program to reverse a complex list. It means that all sub-list must also be reverse. For example
revx([[1,2],[3,4]]) --> [[4,3],[2,1]]
5. Write a program to sum element-wise of two input lists of the same size, produce the new list.
sumlist([1,2,3],[4,5,6]) --> [5,7,9]
6. Recursion in an array. Do the similar problem as 5. but using index into the input array instead. Observe how different two programs (4 and 5) are.
7. In Ajarn Somchai's slide, he asked you to write a program to "flatten" a complex list. Do it with our four methods. flatten([[1,2],3]) --> [1,2,3]
8. If you really want a mind-bending exercise, write a recursive program to multiply matrix (not using index), representing a matrix with list of list.