# UseItOrLoseIt.py

def bestPacking(capacity, items):
    """ Given a suitcase capacity and a list of items
    consisting of positive numbers, returns a number
    indicating the largest sum that can be made from a
    subset of the items without exceeding the capacity.
    """
    if capacity <= 0 or items == []:
        return 0
    elif items[0] > capacity:
        return bestPacking(capacity, items[1:])
    else:
        loseIt = bestPacking(capacity, items[1:])
        useIt = items[0] + bestPacking(capacity-items[0],
                                      items[1:])
        return max(loseIt, useIt)

print bestPacking(42,[5,10,18,23,30,45]), "(41)"
print bestPacking(42,[2,5,10,18,23,30,45]), "(42)"
print bestPacking(10,[8,4,6]), "(10)"

# Edit Distance between two strings is
# the minimum number of
# (1) changes of characters +
# (2) addition of characters +
# (3) deletion of characters
# in one string to make it exactly match the other string

# Say,
human = "ATTGCG"
chimp = "ACTCG" # Edit distance = 1
gorilla = "AGGCG" # Edit distance = 2

# == humans closer to chimps than gorillas
# Of course, the sequences and the edit differences are
# more complicated than this.

def distance(s1,s2):
    """ Returns the edit distance between s1 and s2."""
    if s1 == "":
        return len(s2) # must add len(s2) chars to s1
    if s2 == "":
        return len(s1) # must delete len(s1) chars from s1
    elif s1[0] == s2[0]:
        return distance(s1[1:],s2[1:]) # easy, first chars match
    else:
        substitution = 1 + distance(s1[1:],s2[1:])
        deletion = 1 + distance(s1[1:],s2)
        insertion = 1 + distance(s1,s2[1:1])
        return min(substitution,deletion,insertion)

print "distance(human,gorilla) =", distance(human,gorilla), "(2)"
print "distance(chimp,gorilla) =", distance(chimp,gorilla), "(2)"
print "distance(human,chimp) =", distance(human,chimp), "(1)"
print "distance('spam','poems') =", distance('spam','poems'), "(4)"
print "distance('alien','sales') =", distance('alien','sales'), "(3)"