```
#for questions 5-7

import math

#5
def iterate(u, v):
    #calculate gradient vector at current u, v
    common = 2 * (u * math.exp(v) - 2 * v * math.exp(-u))
    gradu = common * (math.exp(v) + 2 * v * math.exp(-u))
    gradv = common * (u * math.exp(v) - 2 * math.exp(-u))

    #calculate u, v after 1 step
    u -= 0.1 * gradu
    v -= 0.1 * gradv

    #calculate new E for the new u, v
    error = (u * math.exp(v) - 2 * v * math.exp(-u)) ** 2

    #return the number of recursions taken to get the desired error
    if error < 10 ** -14:
        return 1
    return iterate(u, v) + 1

print(iterate(1.0, 1.0))


#6
def iterate6(u, v):
    #calculate gradient vector at current u, v
    common = 2 * (u * math.exp(v) - 2 * v * math.exp(-u))
    gradu = common * (math.exp(v) + 2 * v * math.exp(-u))
    gradv = common * (u * math.exp(v) - 2 * math.exp(-u))

    #calculate u, v after 1 step
    u -= 0.1 * gradu
    v -= 0.1 * gradv

    #calculate new E for the new u, v
    error = (u * math.exp(v) - 2 * v * math.exp(-u)) ** 2

    #return u, v when error < 10 ^ -14
    if error < 10 ** -14:
        return u, v
    return iterate6(u, v)

print(iterate6(1.0, 1.0))


#7
def coor_desc(u, v):
    for i in range(15):
        #step 1: move along u
        common = 2 * (u * math.exp(v) - 2 * v * math.exp(-u))
        gradu = common * (math.exp(v) + 2 * v * math.exp(-u))
        u -= 0.1 * gradu

        #step 2: move along v
        common = 2 * (u * math.exp(v) - 2 * v * math.exp(-u))
        gradv = common * (u * math.exp(v) - 2 * math.exp(-u))
        v -= 0.1 * gradv

    #calculate E after 15 iterations
    error = (u * math.exp(v) - 2 * v * math.exp(-u)) ** 2
    return error

print(coor_desc(1.0, 1.0))
```