

MIT OpenCourseWare
<http://ocw.mit.edu>

6.00 Introduction to Computer Science and Programming
Fall 2008

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

6.00 Handout, Lecture 4 (Not intended to make sense outside of lecture)

```
#example code for finding square roots
x = 16
ans = 0
if x >= 0:
    while ans*ans < x:
        ans = ans + 1
        print 'ans =', ans
    if ans*ans != x:
        print x, 'is not a perfect square'
    else: print ans
else: print x, 'is a negative number'

def sqrt(x):
    """Returns the square root of x, if x is a perfect square.
    Prints an error message and returns None otherwise"""
    ans = 0
    if x >= 0:
        while ans*ans < x: ans = ans + 1
        if ans*ans != x:
            print x, 'is not a perfect square'
            return None
        else: return ans
    else:
        print x, 'is a negative number'
        return None

def f(x):
    x = x + 1
    return x

x = 3
z = f(x)
print x
print z

def solve(numLegs, numHeads):
    for numChicks in range(0, numHeads + 1):
        numPigs = numHeads - numChicks
        totLegs = 4*numPigs + 2*numChicks
        if totLegs == numLegs:
            return (numPigs, numChicks)
    return (None, None)

def barnYard():
    heads = int(raw_input('Enter number of heads: '))
    legs = int(raw_input('Enter number of legs: '))
    pigs, chickens = solve(legs, heads)
    if pigs == None:
        print 'There is no solution'
    else:
        print 'Number of pigs:', pigs
        print 'Number of chickens:', chickens
```

```

def solve1(numLegs, numHeads):
    for numSpiders in range(0, numHeads + 1):
        for numChicks in range(0, numHeads - numSpiders + 1):
            numPigs = numHeads - numChicks - numSpiders
            totLegs = 4*numPigs + 2*numChicks + 8*numSpiders
            if totLegs == numLegs:
                return (numPigs, numChicks, numSpiders)
    return (None, None, None)

def barnYard1():
    heads = int(raw_input('Enter number of heads: '))
    legs = int(raw_input('Enter number of legs: '))
    pigs, chickens, spiders = solve1(legs, heads)
    if pigs == None:
        print 'There is no solution'
    else:
        print 'Number of pigs:', pigs
        print 'Number of chickens:', chickens
        print 'Number of spiders:', spiders

def solve2(numLegs, numHeads):
    solutionFound = False
    for numSpiders in range(0, numHeads + 1):
        for numChicks in range(0, numHeads - numSpiders + 1):
            numPigs = numHeads - numChicks - numSpiders
            totLegs = 4*numPigs + 2*numChicks + 8*numSpiders
            if totLegs == numLegs:
                print 'Number of pigs: ' + str(numPigs) + ', ',
                print 'Number of chickens: '+str(numChicks)+' ',
                print 'Number of spiders:', numSpiders
                solutionFound = True
    if not solutionFound: print 'There is no solution.'

def isPalindrome(s):
    """Returns True if s is a palindrome and False otherwise"""
    if len(s) <= 1: return True
    else: return s[0] == s[-1] and isPalindrome(s[1:-1])

def fib(x):
    """Return fibonacci of x, where x is a non-negative int"""
    if x == 0 or x == 1: return 1
    else: return fib(x-1) + fib(x-2)

```