Practice Problem 1

Complete the function `lastDiv3()` which should return the last number on the list argument `nums` that is equally divisible by 3.

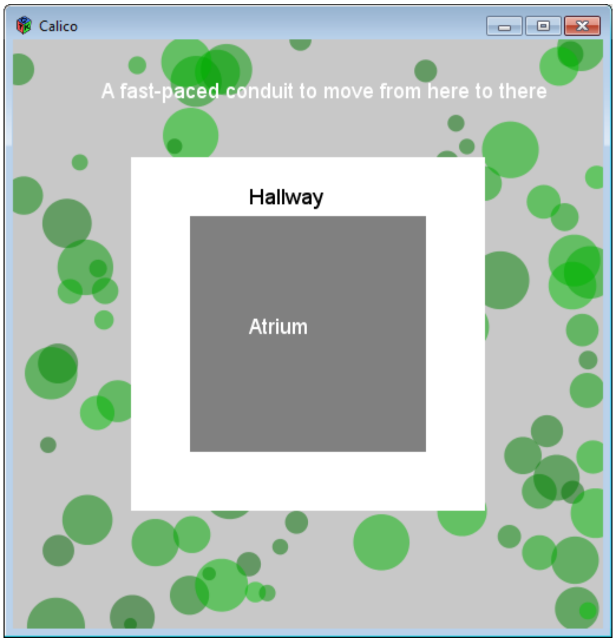
```
def lastDiv3( nums ):

    i = len(nums) - 1
    while i >=0:
        n = nums[i]
        if n % 3 == 0:
            return n
        i = i - 1




nums = [1, 3, 4, 6, 7, 9, 10]
print( lastDiv3( nums ) )
```

Practice Problem 2



A fast-paced conduit to move from here to there

You have been hired by an architecture company to develop an interactive display of their contemplative first-floor design for the 21st century. When a visitor moves the mouse over the Hallway, one message will appear at the top of the screen. When the mouse is moved over the Atrium, another message will be displayed. You are on your way to make the final presentation when you discover that two functions were never completed. The `inHallway()` function returns True when the mouse is over the Hallway, and the `inAtrium()` function returns True when the mouse is over the Atrium. You must complete these functions. Following is the incomplete program.

Hint: consider using `inAtrium()` as part of your `inHallway()` function.

```
from Processing import *
window(500, 500)

atriumInfo = "A spacious place for meditation and reflection"
hallwayInfo = "A fast-paced conduit to move from here to there"

# Return True if the mouse is in the Atrium
def inAtrium():
    x, y = mouseX(), mouseY()
    if x < 150 or x > 350:
        return False
    if y < 150 or y > 350:
        return False
    return True

# Return True if the mouse is in the Hallway
def inHallway():
    x, y = mouseX(), mouseY()
    if inAtrium() == True:
        return False
    if x < 100 or x > 400:
        return False
    if y < 100 or y > 400:
        return False
    return True
```

```
# Init vegetation
treeData = []
for i in range(100):
    x = random( width() )       # Random position
    y = random( height() )
    d = random(10, 50)          # Random diameter
    g = random(100, 200)        # Random shade of green
    treeData.append( [x, y, d, g] )

# Draw all trees
def drawVegetation():
    for i in range( len(treeData) ):
        x = treeData[i][0]
        y = treeData[i][1]
        d = treeData[i][2]
        g = treeData[i][3]
        fill(0, g, 0, 128)
        ellipse(x, y, d, d)

# Draw labels, if appropriate
def drawLabels():
    fill(255)
    if inAtrium() == True:
        text(atriumInfo, 75, 50)
    elif inHallway() == True:
        text(hallwayInfo, 75, 50)

# Draw all aspects of the first floor
def drawFirstFloor():
    rectMode(CENTER)
    fill(255)
    rect(250, 250, 300, 300)
    fill(128)
    rect(250, 250, 200, 200)
    fill(255)
    text("Atrium", 200, 250)
    fill(0)
    text("Hallway", 200, 140)

# Main draw function
def draw(o, e):
    background(200)
    drawVegetation()
    drawFirstFloor()
    drawLabels()

noStroke()
textSize(14)

frameRate(10)
onLoop += draw
loop()
```