Last name ______________________

First name ______________________

LARSON—MATH 255–CLASSROOM WORKSHEET 15
Riemann Integration & Debugging.

1. Log in to your Sage/Cocalc account.

   (a) Start the Chrome browser.

   (b) Go to `http://cocalc.com` and sign in.

   (c) You should see an existing Project for our class. Click on that.

   (d) Click "New", call it **c15**, then click "Sage Worksheet".

   (e) For each problem number, label it in the Sage cell where the work is. So for Problem 1, the first line of the cell should be `#Problem 1`.

   Riemann Integration

   Given a continuous function $f(x)$ on an interval $[a, b]$ we want to find the *area* between the curve, the x-axis and the lines $y = a$ and $y = b$. One way to do this is to slice up $[a, b]$ into n equal-sized intervals $[a_0, a_1]$, $[a_1, a_2]$,..., $[a_n, a_{n+1}]$ (where $a_1 = a$ and $a_{n+1} = b$), pick a point c_i from each interval $[a_i, a_{i+1}]$ and compute the area $f(c_i) \cdot \Delta$ of a rectangle, where Δ is the interval length $a_{i+1} - a_i$. There are different ways to pick the c_i's. You could pick the leftmost point of the interval, the midpoint, the rightmost point, or even a random point. The *Riemann Integral* is defined to be the *limit* of these area approximations as n goes to infinity of this quantity.

2. Type in the function `leftpoint_riemann(f,a,b,n)` which computes the leftpoint Riemann sums for n equal intervals.

```
def leftpoint_riemann(f,a,b,n):
    area=0
    Delta=(b-a)/n
    for i in [0..n]:
        leftpoint=a+i*Delta
        newarea=f(leftpoint)*Delta
        area=area+newarea
    return area
```

3. Type in the function `rightpoint_riemann(f,a,b,n)` which computes the rightpoint Riemann sums for n equal intervals.

```
def rightpoint_riemann(f,a,b,n):
    area=0
    Delta=(b-a)/n
    for i in [0..n]:
        rightpoint=b-i*Delta
        newarea=f(rightpoint)*Delta
        area=area+newarea
    return area
```

4. Find the value of `leftpoint_riemann(f,a,b,n)` for f(x)=x**2 on [0,3] with $n = 2$, $n = 5$, $n = 10$ and $n = 100$. Here you are making the intervals smaller and smaller, giving a better and better approximation.

5. Find the values of `rightpoint_riemann(f,a,b,n)`for f(x)=x**2 on [0,3] with $n = 2$, $n = 5$, $n = 10$ and $n = 100$. Compare with your results for `leftpoint_riemann(f,a,b,n)`. What do you notice???

6. Now lets add *print statements* to get an idea of what these programs are doing.

```
def leftpoint_riemann(f,a,b,n):
    area=0
    Delta=(b-a)/n
    for i in [0..n]:
        leftpoint=a+i*Delta
        newarea=f(leftpoint)*Delta
        area=area+newarea
        print "i,leftpoint,newarea are {},{},{}".format(i,leftpoint,newarea)
    return area

def rightpoint_riemann(f,a,b,n):
    area=0
    Delta=(b-a)/n
    for i in [0..n]:
        rightpoint=b-i*Delta
        newarea=f(rightpoint)*Delta
        area=area+newarea
        print "i,rightpoint,newarea are {},{},{}".format(i,rightpoint,newarea
    return area
```

   Can you figure out **why** these two programs give the same answers? If not, can you think of any more print statements to add that would be helpful?

7. After you figure out why these programs give the same answer, adjust your code so we are calculating what we want. Then recalculate the values of `leftpoint_riemann()` and `rightpoint_riemann()` for f(x)=x**2 on [0,3] with $n = 2$, $n = 5$, $n = 10$ and $n = 100$.

8. Given a continuous function f(x) on [a,b], define a function `midpoint_riemann(f,a,b,n)` which computes the midpoint Riemann sums for n equal intervals.

9. Find the values of `midpoint_riemann(f,a,b,n)` for f(x)=x**2 on [0,3] with $n = 2$, $n = 5$, $n = 10$ and $n = 100$. Compare with your results for `rightpoint_riemann(f,a,b,n)`.