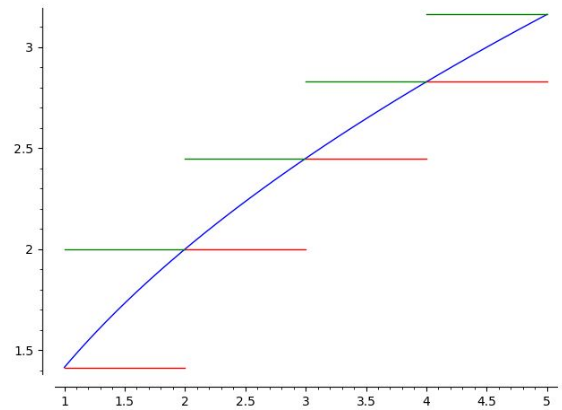
Lab 11
Issac Hertzler - We couldn't
Andrew White

```
def f(x): return sqrt(2*x)
a, b = 1,5
n = 4

print(list(range(n)))
print([i/2 + 1 for i in range(n)])
[0, 1, 2, 3]
[1, 3/2, 2, 5/2]

x_n = [a + (b-a)/n*i for i in range(n+1)]
print(x_n)
[1, 2, 3, 4, 5]
```

Problem 1

a and b are the points for the interval that we would start at. n is how many times the problem is solved, since the n equals 4 then the problem gets solved 4 times using the numbers 1 through 4. i is the numbers that come from the range problems

```
fx_n = [f(i).n() for i in x_n]
print(fx_n)

print("Left endpoints: ", fx_n[:-1])
print("Right endpoints: ", fx_n[1:])

L = (b-a)/n * sum(fx_n[:-1])
R = (b-a)/n * sum(fx_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
Left Approximation:  8.69213042990246
Right Approximation:  10.4401945276977

P_f = plot(f(x), x, a, b)
P_L = sum([plot(fx_n[i], x, x_n[i], x_n[i+1], color = "red") for i in range(n)])
P_R = sum([plot(fx_n[i+1], x, x_n[i], x_n[i+1], color = "green") for i in range(n)])
show(P_f + P_L + P_R)
```



Problem 2

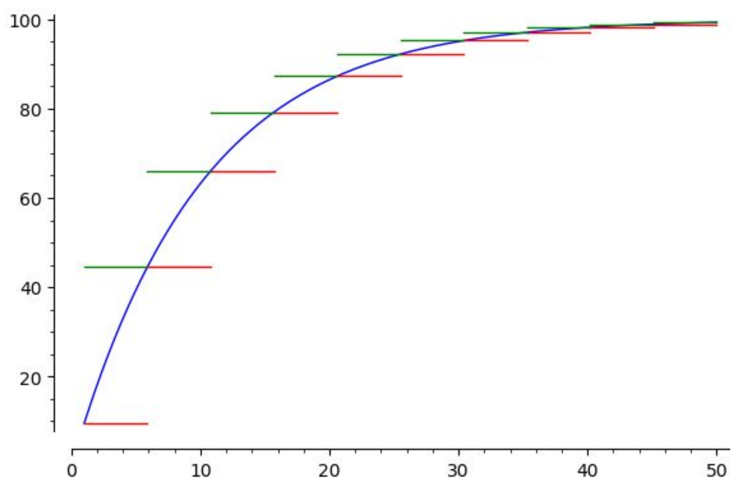
a. Left Approximation: 3763.96818419897
  Right Approximation: 4204.03692500704
b. Left Approximation: 3979.71740437727
  Right Approximation: 4023.72427845807
c. n = 1,000,000
  Left Approximation: 4001.8983286176
  Right Approximation: 4001.90272930505

```
def f(x): return 100*(1-e^(-.1*x))
a, b = 1,50
n = 10

print(list(range(n)))
print([i/2 + 1 for i in range(n)])

x_n = [a + (b-a)/n*i for i in range(n+1)]
print(x_n)

fx_n = [f(i).n() for i in x_n]
print(fx_n)

print("Left endpoints: ", fx_n[:-1])
print("Right endpoints: ", fx_n[1:])
```

```
L = (b-a)/n * sum(fx_n[:-1])
R = (b-a)/n * sum(fx_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
```

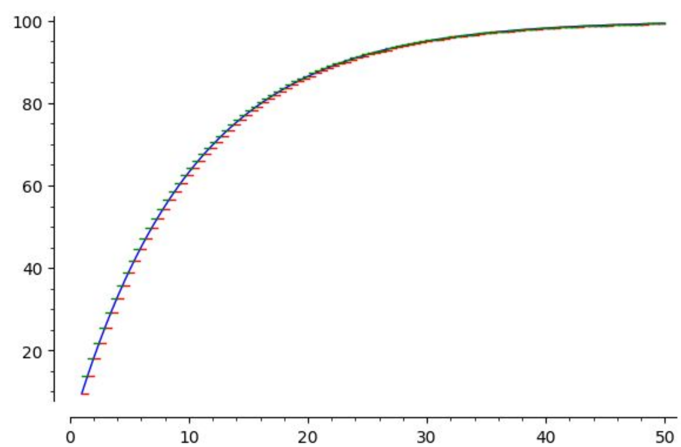
Left Approximation: 3763.96818419897
Right Approximation: 4204.03692500704

```
P_f = plot(f(x), x, a, b)
P_L = sum([plot(fx_n[i], x, x_n[i], x_n[i+1], color = "red") for i in range(n)])
P_R = sum([plot(fx_n[i+1], x, x_n[i], x_n[i+1], color = "green") for i in range(n)])
show(P_f + P_L + P_R)
```




```
def f(x): return 100*(1-(e)^(-.1*x))
a, b = 1,50
n = 100

#print(list(range(n)))
#print([i/2 + 1 for i in range(n)])

x_n = [a + (b-a)/n*i for i in range(n+1)]
#print(x_n)

fx_n = [f(i).n() for i in x_n]
#print(fx_n)

#print("Left endpoints: ", fx_n[:-1])
#print("Right endpoints: ", fx_n[1:])
```

```
L = (b-a)/n * sum(fx_n[:-1])
R = (b-a)/n * sum(fx_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
Left Approximation:  3979.71740437727
Right Approximation:  4023.72427845807

P_f = plot(f(x), x, a, b)
P_L = sum([plot(fx_n[i], x, x_n[i], x_n[i+1], color = "red") for i in range(n)])
P_R = sum([plot(fx_n[i+1], x, x_n[i], x_n[i+1], color = "green") for i in range(n)])
show(P_f + P_L + P_R)
```



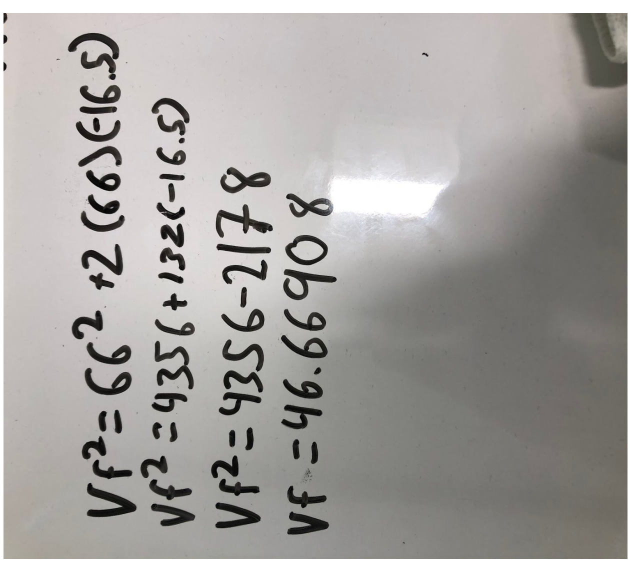

```
def f(x): return 100*(1-(e)^(-.1*x))
a, b = 1,50
n = 1000000

#print(list(range(n)))
#print([i/2 + 1 for i in range(n)])

x_n = [a + (b-a)/n*i for i in range(n+1)]
#print(x_n)

fx_n = [f(i).n() for i in x_n]
#print(fx_n)

#print("Left endpoints: ", fx_n[:-1])
#print("Right endpoints: ", fx_n[1:])
```

```
L = (b-a)/n * sum(fx_n[:-1])
R = (b-a)/n * sum(fx_n[1:])
print("Left Approximation: ", L)
print("Right Approximation: ", R)
Left Approximation:  4001.89832861764
Right Approximation:  4001.90272930505
```

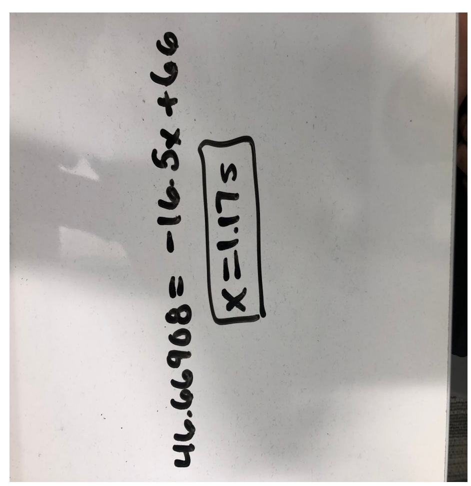
Problem 3

a. 73.333 feet
b. 25.3834 miles per hour
c. 4 seconds
d. 1.17 seconds

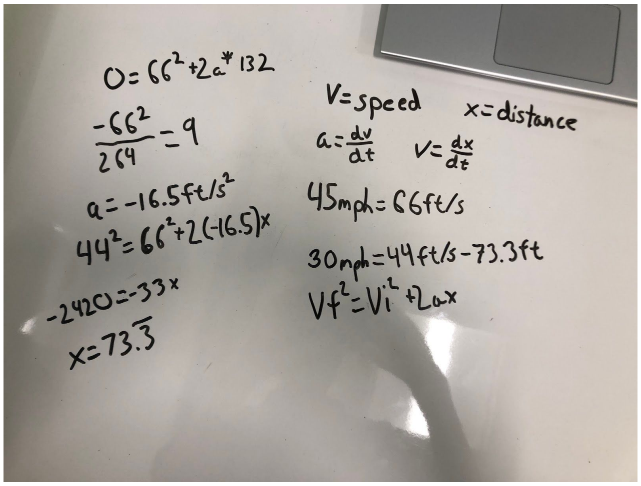
3c).



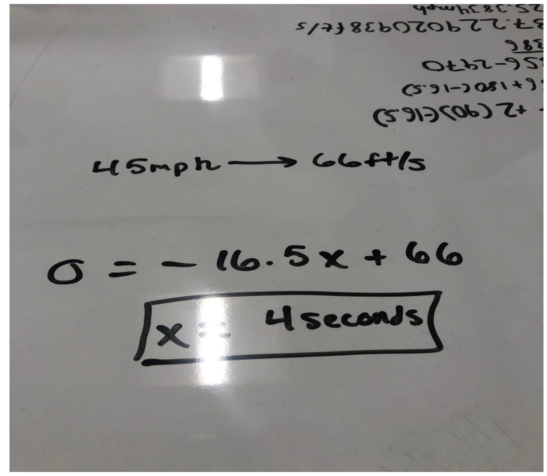
3c).



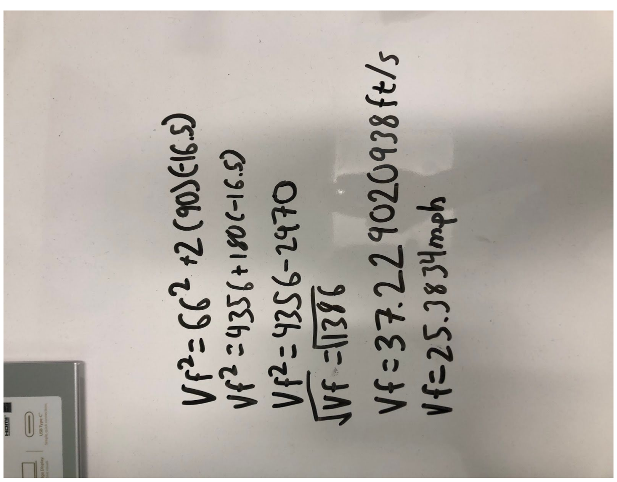
3a).


O=66²+2a*132
-66²/264 = 9
a=-16.5ft/s²
44²=66²+2(-16.5)x
-2420=-33x
x=73.3
V=speed
x=distance
a=dv/dt
V=dx/dt
45mph=66ft/s
30mph=44ft/s-73.3ft
Vf²=Vi²+2ax

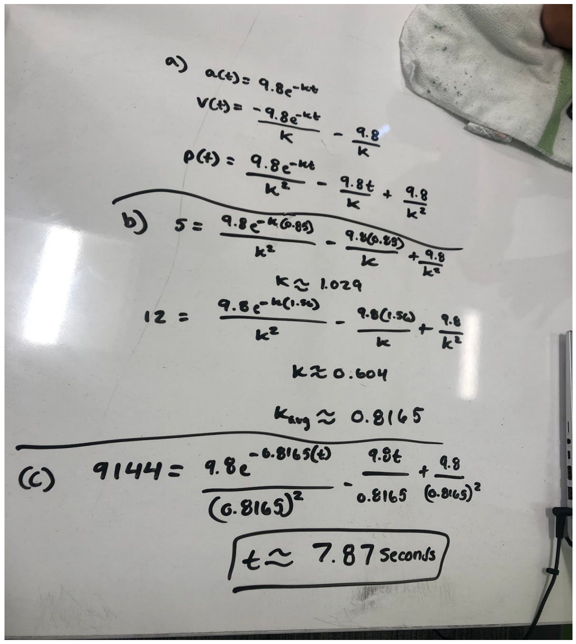
3d).


45mph → 66ft/s
0 = - 16.5x + 66
x = 4seconds

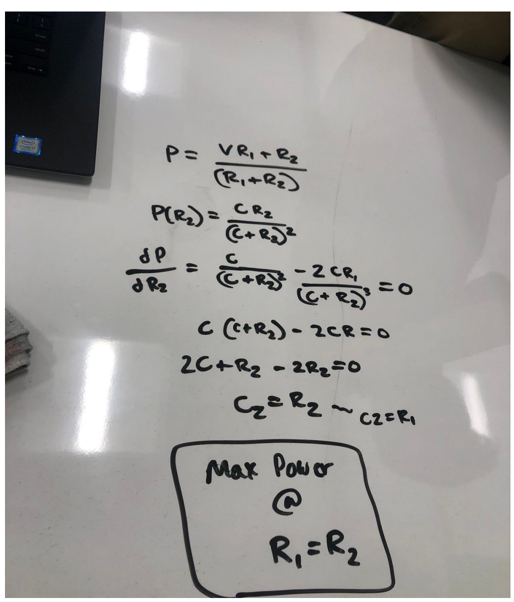
3b).


Vf²=66² +2(90)(-16.5)
Vf² =4356+180(-16.5)
Vf²=4356-2970
Vf²=1386
√Vf =√1386
Vf=37.229020938 ft/s
Vf=25.3834mph

4).


a) a(t)= 9.8e^-kt
V(t)= -9.8e^-kt / k - 9.8/k
p(t)= 9.8e^-kt / k² - 9.8t/k + 9.8/k²
b) 5= 9.8e^-k(0.85) / k² - 9.8(0.85)/k + 9.8/k²
k ≈ 1.029
12 = 9.8e^-k(1.56) / k² - 9.8(1.56)/k + 9.8/k²
k ≈ 0.604
k_avg ≈ 0.8165
(c) 9144= 9.8e^-0.8165(t) / (0.8165)² - 9.8t/0.8165 + 9.8/(0.8165)²
t ≈ 7.87 seconds

5).


P= VR₁+R₂ / (R₁+R₂)
P(R₂)= CR₂ / (C+R₂)²
dP/dR₂ = C/(C+R₂)² - 2CR₁/(C+R₂)³ = 0
C (C+R₂) - 2CR = 0
2C+R₂ - 2R₂=0
C₂=R₂ ~ C2=R₁
Max Power
@
R₁=R₂