```python
##############################################################
#week 5 homework  Statistical Methods in Experimental Physics#
##############################################################
#author Qiyu Chen
#date 2024.3.21

import random
import math
import matplotlib.pyplot as plt

'''A plot function to plot hist'''
def plot(list, title, filename):
    plt.hist(list, bins=100, density=True)
    plt.title(title)
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.savefig('Desktop/{0}.jpg'.format(filename))
    plt.cla()

##############################################################
#exer3.1 started
tot = 1000000;  '''1000 is too small to create a satisfying distribution'''

theta=[2*math.pi*(random.random()) for _ in range(tot)]
r=[math.sqrt(random.random()) for _ in range(tot)]

plot(r, 'exer3.1  distribution of r', 'rdistribution')
plot(theta, 'exer3.1  distribution of theta', 'thetadistribution')

rtheta = zip(r, theta); '''rtheta is the list of 100000 sets of coordinate'''
#exer3.1 finished
##############################################################

##############################################################
#exer3.2 started

def f(x):
    return math.exp(-x)/math.sqrt(x)
def integrate(x_min, x_max, tot):
    y_min=0
    y_max=max(f(x_min),f(x_max))
    x=[x_min + (x_max-x_min)*random.random() for _ in range(tot)]
    y=[y_min + (y_max-y_min)*random.random() for _ in range(tot)]
    correct=[]
    for i, j in zip(x, y):
        if j < f(i):
            correct.append(i)
    return len(correct)/tot*(x_max-x_min)*y_max

def part1():
    return integrate(0.4, 1, tot)
def part2():
    return integrate(0.1, 0.4, tot)
def part3():
    return integrate(0.0000001, 0.1, tot)

print(part1()+part2()+part3())

#exer3.2 finished
#############################################################

#############################################################
#exer3.3 started
D, R, c, beta, tau=0.14, 0.07, 3e8, 0.826968, 8.954e-11
all_z=[random.expovariate(1/beta/c/tau*math.sqrt(1-beta**2)) for _ in range(tot)]
costheta=[-1+2*random.random() for _ in range(tot)]
correct=[]
for z, t in zip(all_z, costheta):
    if z<D and math.sqrt(1-beta**2)*math.sqrt(1-t**2)/(1+t)<=R/(D-z)\
    and math.sqrt(1-beta**2)*math.sqrt(1-t**2)/(1-t)<=R/(D-z):
        correct.append(z)
print(len(correct)/tot)
#exer3.3 finished
#############################################################
```

```
#############################################################
#exer3.4 started
a=[random.expovariate(1/2) for _ in range(tot)]
b=[random.expovariate(1) for _ in range(tot)]
correct=[min(i, j) for i, j in zip(a, b)]
print(sum(correct)/len(correct))
#exer3.4 finished
#############################################################

#######
##end##
#######
```