```python
#!/usr/bin/python3

from pylab import *
from numpy import *

from pprint import pprint


def simpson(func, a, b, h=0.1):
    """Approximates integral using simpson method"""
    n = int(abs(b - a) / h)
    n -= 1 if n % 2 == 1 else 0
    I1 = (h / 3.0) * (
        func(a) + func(b) +
        (4.0 * sum([func(a + (k * h)) for k in range(1, n, 2)])) +
        (2.0 * sum([func(a + (k * h)) for k in range(2, n - 1, 2)])))
    h2 = 2 * h
    n2 = int(abs(b - a) / h2)
    n2 -= 1 if n2 % 2 == 1 else 0
    I2 = (h2 / 3.0) * (
        func(a) + func(b) +
        (4.0 * sum([func(a + (k * h2)) for k in range(1, n2, 2)])) +
        (2.0 * sum([func(a + (k * h2)) for k in range(2, n2 - 1, 2)])))
    return (I1, abs(I1 - I2) / 3)


def load_data():
    data = loadtxt('data1.txt')
    return data[0], data[1], data[3], data[5]


T_plot, E1, E2, E3 = load_data()
T_plot = [round(x, 10) for x in list(T_plot)]


def RMI(T, T_max):
    return simpson(lambda x: (2*E1[T_plot.index(round(x,10))]-E2[T_plot.index(round(x,10))]-2*E3[T_plot.index(round(x, 10))]) / x**2, T, T_max)


def p3():
    T_max = T_plot[-1]

    def a():
        print(
            RMI(
                float(input("T[{}-{}]: ".format(T_plot[0], T_plot[-1]))),
                T_max))
        # pprint([round(x, 5) for x in list(T)])
        plot(T_plot, [RMI(x, T_max)[0] for x in T_plot], label="RMI")
        xlabel('Temperature (K)')
        ylabel('RMI(T)')
        xlim(0, 100)
        legend()
        show()

    a()
    print("Error ~10^-4")
    print(
        "I would suggest using the fastest method(Trapazoidal) because the error caused by the data will always be greater than the error caused by any of the integration methods"
    )

```

```python

if __name__ == "__main__":
    p3()
```