```
import numpy as np
from matplotlib import pyplot as plt


pi = 3.14159
c = 3E8
yd = 1.5E13
wp = 1e16
w0 = 1e18


lrmin = 1E-11
lrmax = 1E-8
lR = np.linspace(lrmin, lrmax, 10)
lR
```

```
array([1.00e-11, 1.12e-09, 2.23e-09, 3.34e-09, 4.45e-09, 5.56e-09,
       6.67e-09, 7.78e-09, 8.89e-09, 1.00e-08])
```

```
luv = np.linspace(2E-8, 400E-9, 10)
lvis = np.linspace(420E-9, 750E-9, 10)
lnir = np.linspace(800E-9, 1.5E-6, 10)
lmir = np.linspace(1.55E-6, 5E-6, 10)
lfir = np.linspace(5E-6, 20E-6, 10)
lter = np.linspace(21E-6, 100E-6, 10)
lmic = np.linspace(110E-6, 990E-6, 10)
lmil = np.linspace(1E-3, 10E-3, 10)


L = np.concatenate((lR, luv, lvis, lnir, lmir, lfir, lter, lmic, lmil), axis=None)#np.array([lR, luv, lvis, lnir, lmir, lfir, lter, lmic,
w = np.divide((2 * pi * c), L)
```

```
array([1.00000000e-11, 1.12000000e-09, 2.23000000e-09, 3.34000000e-09,
       4.45000000e-09, 5.56000000e-09, 6.67000000e-09, 7.78000000e-09,
       8.89000000e-09, 1.00000000e-08, 2.00000000e-08, 6.22222222e-08,
       1.04444444e-07, 1.46666667e-07, 1.88888889e-07, 2.31111111e-07,
       2.73333333e-07, 3.15555556e-07, 3.57777778e-07, 4.00000000e-07,
       4.20000000e-07, 4.56666667e-07, 4.93333333e-07, 5.30000000e-07,
       5.66666667e-07, 6.03333333e-07, 6.40000000e-07, 6.76666667e-07,
       7.13333333e-07, 7.50000000e-07, 8.00000000e-07, 8.77777778e-07,
       9.55555556e-07, 1.03333333e-06, 1.11111111e-06, 1.18888889e-06,
       1.26666667e-06, 1.34444444e-06, 1.42222222e-06, 1.50000000e-06,
       1.55000000e-06, 1.93333333e-06, 2.31666667e-06, 2.70000000e-06,
       3.08333333e-06, 3.46666667e-06, 3.85000000e-06, 4.23333333e-06,
       4.61666667e-06, 5.00000000e-06, 5.00000000e-06, 6.66666667e-06,
       8.33333333e-06, 1.00000000e-05, 1.16666667e-05, 1.33333333e-05,
       1.50000000e-05, 1.66666667e-05, 1.83333333e-05, 2.00000000e-05,
       2.10000000e-05, 2.97777778e-05, 3.85555556e-05, 4.73333333e-05,
       5.61111111e-05, 6.48888889e-05, 7.36666667e-05, 8.24444444e-05,
       9.12222222e-05, 1.00000000e-04, 1.10000000e-04, 2.07777778e-04,
       3.05555556e-04, 4.03333333e-04, 5.01111111e-04, 5.98888889e-04,
       6.96666667e-04, 7.94444444e-04, 8.92222222e-04, 9.90000000e-04,
       1.00000000e-03, 2.00000000e-03, 3.00000000e-03, 4.00000000e-03,
       5.00000000e-03, 6.00000000e-03, 7.00000000e-03, 8.00000000e-03,
       9.00000000e-03, 1.00000000e-02])
```

```
def func(x):
  return 1+(wp**2)*((w0**2)*(x**2))/((((w0**2)-(x**2))**2)+((x**2)*(yd**2)))
my_func = np.vectorize(func)
n = my_func(w)
n
```

```
array([2.81464051e+27, 8.43511993e+31, 8.76450260e+32, 6.85766170e+31,
       2.66467524e+31, 1.46724115e+31, 9.43290778e+30, 6.62501288e+30,
       4.92892491e+30, 3.81965845e+30, 9.04255888e+29, 9.19407337e+28,
       3.25922076e+28, 1.65227430e+28, 9.96036355e+27, 6.65300183e+27,
       4.75617281e+27, 3.56846010e+27, 2.77586856e+27, 2.22075587e+27,
       2.01428272e+27, 1.70379636e+27, 1.45993416e+27, 1.26491330e+27,
       1.10651114e+27, 9.76102332e+26, 8.67459283e+26, 7.75994577e+26,
       6.98268644e+26, 6.31661594e+26, 5.55170474e+26, 4.61143779e+26,
       3.89128472e+26, 3.32754184e+26, 2.87798835e+26, 2.51374465e+26,
       2.21451564e+26, 1.96570118e+26, 1.75658074e+26, 1.57913902e+26,
       1.47890202e+26, 9.50580411e+25, 6.62025911e+25, 4.87387523e+25,
       3.73732513e+25, 2.95649807e+25, 2.39706749e+25, 1.98260753e+25,
       1.66703462e+25, 1.42122104e+25, 1.42122104e+25, 7.99436734e+24,
       5.11639480e+24, 3.55305183e+24, 2.61040538e+24, 1.99859159e+24,
       1.57913409e+24, 1.27909860e+24, 1.05710628e+24, 8.88262911e+23,
       8.05680644e+23, 4.00697790e+23, 2.39016336e+23, 1.58586909e+23,
       1.12850575e+23, 8.43841418e+22, 6.54725830e+22, 5.22731560e+22,
       4.26972808e+22, 3.55305158e+22, 2.93640627e+22, 8.23006601e+21,
       3.80558252e+21, 2.18410383e+21, 1.41492509e+21, 9.90624355e+20,
       7.32068044e+20, 5.62955994e+20, 4.46329344e+20, 3.62519292e+20,
       3.55305158e+20, 8.88262896e+19, 3.94783509e+19, 2.22065724e+19,
```

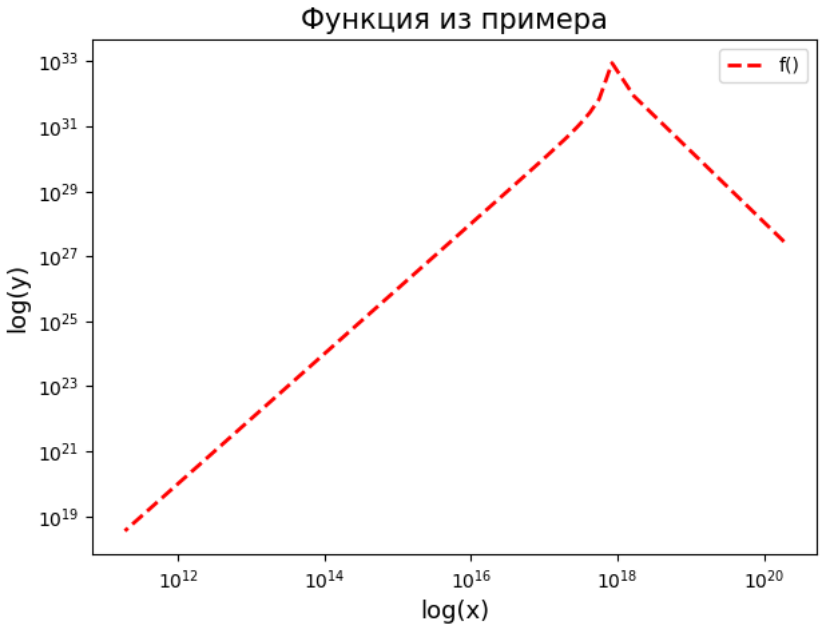
```
           1.42122063e+19, 9.86958773e+18, 7.25112568e+18, 5.55164310e+18,
           4.38648343e+18, 3.55305158e+18])
```

```
#строим
x = w
y = n

# Resizing the figure
plt.figure(figsize=[7, 5])

# Plotting the graph with Log ticks at x and y axis using loglog
plt.loglog(x, y, '--r', linewidth=2,
           label='f()')
plt.title('Функция из примера',
          fontsize=15)
plt.xlabel('log(x)', fontsize=13)
plt.ylabel('log(y)', fontsize=13)

plt.legend()
plt.show()
```




```
# Задание к ЛР1 по ЛАТ
# Необходимо

# 1. Построить зависимость:

# 1) показателя преломления

# 2) показателя поглощения

# для данного материала в заданном спектральном диапазоне.

# 2. Вычислить толщину скин слоя (материал задан, длина волны - центральная из диапазона)
```

```
pi = 3.14159
c = 3E8
yd = 1.5E13
wp = 1e16
w0 = 1e18
pi, c, yd, wp, w0
```

```
     (3.14159, 300000000.0, 15000000000000.0, 1e+16, 1e+18)
```

```
lrmin = 30E-6#терагерцовый диапазон m
lrmax = 1E-3
lR = np.linspace(lrmin, lrmax, 10)
lR
```

```
     array([3.00000000e-05, 1.37777778e-04, 2.45555556e-04, 3.53333333e-04,
            4.61111111e-04, 5.68888889e-04, 6.76666667e-04, 7.84444444e-04,
            8.92222222e-04, 1.00000000e-03])
```

```
w_my_var = np.divide((2 * pi * c), lR)
w_my_var
```

```
array([6.28318000e+13, 1.36811177e+13, 7.67628326e+12, 5.33477547e+12,
       4.08785205e+12, 3.31339570e+12, 2.78564631e+12, 2.40291586e+12,
       2.11265081e+12, 1.88495400e+12])
```

```
n_my_var = my_func(w_my_var)
n_my_var
```

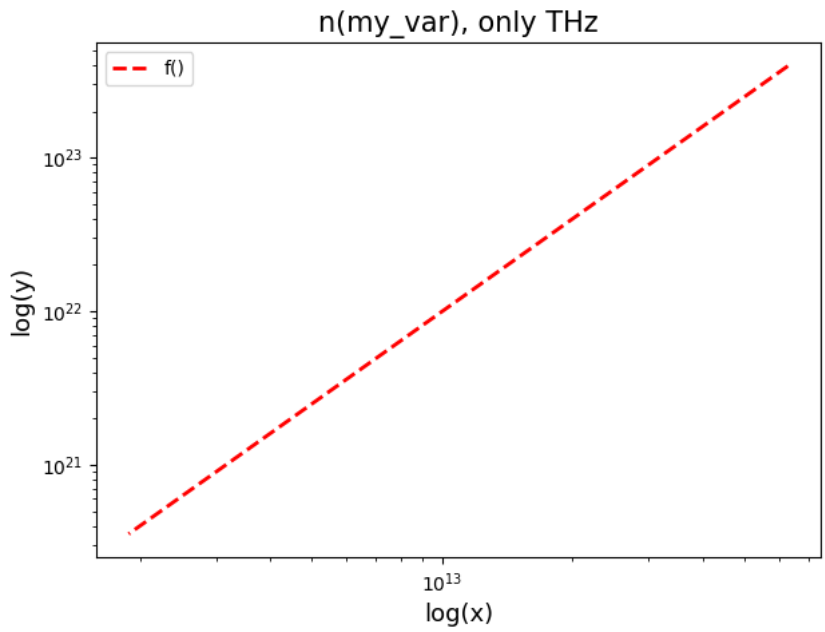
```
array([3.94783512e+23, 1.87172983e+22, 5.89253247e+21, 2.84598293e+21,
       1.67105344e+21, 1.09785911e+21, 7.75982534e+20, 5.77400465e+20,
       4.46329344e+20, 3.55305158e+20])
```

```
#строим my_var
x = w_my_var
y = n_my_var

# Resizing the figure
plt.figure(figsize=[7, 5])

# Plotting the graph with Log ticks at x and y axis using loglog
plt.loglog(x, y, '--r', linewidth=2, label='f()')
plt.title('n(my_var), only THz',
          fontsize=15)
plt.xlabel('log(x)', fontsize=13)
plt.ylabel('log(y)', fontsize=13)

plt.legend()
plt.show()
```




```
x
```

```
array([6.28318000e+13, 1.36811177e+13, 7.67628326e+12, 5.33477547e+12,
       4.08785205e+12, 3.31339570e+12, 2.78564631e+12, 2.40291586e+12,
       2.11265081e+12, 1.88495400e+12])
```

```
y
```

```
array([3.94783512e+23, 1.87172983e+22, 5.89253247e+21, 2.84598293e+21,
       1.67105344e+21, 1.09785911e+21, 7.75982534e+20, 5.77400465e+20,
       4.46329344e+20, 3.55305158e+20])
```

```
#kp - коэффициент экстинкции (показатель поглощения) материала
#k(w)
def func_kp(x):
  return (wp**2)*((w0**2)*(x**2))/((((w0**2)-(x**2))**2)+((x**2)*(yd**2)))
my_func2 = np.vectorize(func_kp)
```

```
kp_my_var = my_func2(w_my_var)
kp_my_var
```

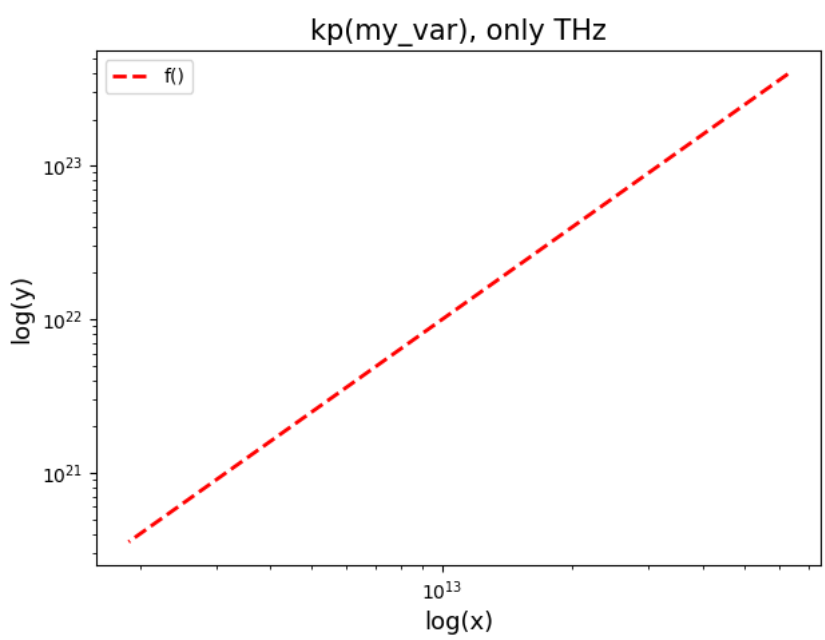
```
array([3.94783512e+23, 1.87172983e+22, 5.89253247e+21, 2.84598293e+21,
       1.67105344e+21, 1.09785911e+21, 7.75982534e+20, 5.77400465e+20,
```

```
#строим my_var 2
x = w_my_var
y = n_my_var

# Resizing the figure
plt.figure(figsize=[7, 5])

# Plotting the graph with Log ticks at x and y axis using loglog
plt.loglog(x, y, '--r', linewidth=2, label='f()')
plt.title('kp(my_var), only THz',
          fontsize=15)
plt.xlabel('log(x)', fontsize=13)
plt.ylabel('log(y)', fontsize=13)

plt.legend()
plt.show()
```




```
# 2. Вычислить толщину скин слоя (d) (материал задан, длина волны - центральная из диапазона)
kp = 12.5#из методички стр 41, таблица 1 (l = 5,0 мкм)
l = np.mean(lR)
l
```

```
0.000515
```

```
d = l / (4 * pi * kp)
d
```

```
3.2785945970034284e-06
```