```python
import numpy as np
import matplotlib.pyplot as plt

c = 2.998e10    #speed of light
h = 6.626e-27   #Planck constant
k = 1.38e-16    #Stefan Boltzman constant
i = 0
def Planck(x,Temp):

    result = 2*h*x**3/c**2/(np.exp(h*x/k/Temp)-1)
    return result

# Generate a wavelength array in Angstroms, from 1-11000 A in steps of 1 A,
# and corresponding array in Hz
# Generate an array of scaling values unity (of continuum optical depth)

wavelength = []
frequency = []
a = []
while i < 11000:

    wavelength.append(i+1)
    frequency.append(10**8*2.998e10/wavelength[i])

    a.append(1)
    i+=1


# Define a set of rest-wavelengths of hydrogen Balmer lines
linecenter = [6562,4861,4340,4101,3970,3889]

# Calculate Dopller shaped line profiles, of hypothetical strength (10) and
# width (5 A), and add these to the array of scaling values

for i in range (len(linecenter)):
    x = []
    for j in range(21):
        x.append(linecenter[i]+ j-10)
        a[x[j]] = a[x[j]] + 10*np.exp(-(float(j-10)/5)**2)


# Define the Balmer jump, using the bound-free continuum opacity frequency
# dependance and add these to the scaling array

for x in range(3646):
    a[x+1] = a[x+1] + 10*((x+1)/float(3646))**3

# Compute the continuum optical depth where the total optical depth equals 2/3
# Compute the Temperature using the continuum optical depth
# Calculate the flux using the temperature and the frequency
tau = []
Temp = []
Flux = []
Teff = 7300
for i in range (len(a)):
    tau.append((float(2)/3)/(1+a[i]))
    Temp.append((0.75 * 7300**4 * (tau[i] + float(2)/3))**0.25)
    Flux.append(np.pi*Planck(frequency[i],Temp[i]))

#plotting the graph.
plt.plot(wavelength,Flux, c = "black")
plt.yscale('log')
plt.axis([2000,10000,1e-5,5e-4])
plt.xlabel('Wavelength [A]')
plt.ylabel('Flux (at surface in erg/cm^2/Hz)')
plt.yticks([2e-5,5e-5, 1e-4, 2e-4],['2e-5','5e-5', '1e-4',' 2e-4'])
plt.xticks([2000,4000,6000,8000,10000])
plt.suptitle('\nToySpectrum of a star of Teff = 7300, with 5 Balmer lines and edge')
plt.savefig('Toyspectrum.pdf')
plt.show()
```