```
# -*- coding: utf-8 -*-
"""Assignment-1.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1zPMRy0tEFVPnAUtGdtBj2EA1-VfALsz5
"""

#Drawing a triangle given 3 sides
import numpy as np
import matplotlib.pyplot as plt
import math



#Triangle sides
a = 8
h = 4
b = c = math.sqrt((a/2)**2+h**2)

#Coordinates of A
p = (a**2 + c**2-b**2 )/(2*a)
q = np.sqrt(c**2-p**2)
print(p,q)

#Triangle vertices
A = np.array([p,q])
B = np.array([0,0])
C = np.array([a,0])


def line_gen(A,B):
  len =10
  x_AB = np.zeros((2,len))
  lam_1 = np.linspace(0,1,len)
  for i in range(len):
    temp1 = A + lam_1[i]*(B-A)
    x_AB[:,i]= temp1.T
  return x_AB



#Generating all lines
```

```
x_AB = line_gen(A,B)
x_BC = line_gen(B,C)
x_CA = line_gen(C,A)

#Plotting all lines
plt.plot(x_AB[0,:],x_AB[1,:],label='$AB$')
plt.plot(x_BC[0,:],x_BC[1,:],label='$BC$')
plt.plot(x_CA[0,:],x_CA[1,:],label='$CA$')

plt.plot(A[0], A[1], 'o')
plt.text(A[0] * (1 + 0.1), A[1] * (1 - 0.1) , 'A')
plt.plot(B[0], B[1], 'o')
plt.text(B[0] * (1 - 0.2), B[1] * (1) , 'B')
plt.plot(C[0], C[1], 'o')
plt.text(C[0] * (1 + 0.03), C[1] * (1 - 0.1) , 'C')

plt.xlabel('$x$')
plt.ylabel('$y$')
plt.legend(loc='best')
plt.grid() # minor
plt.axis('equal')
```