1. The function myawgn() takes PSD, Bandwidth, Sampling Frequency in accordance with Nyquist Theorem, and the length of the sequence as the input argument. To generate an AWGN, we create a random signal which has zero mean and unity variance and make N realizations for every signal. Now, we use the formula,

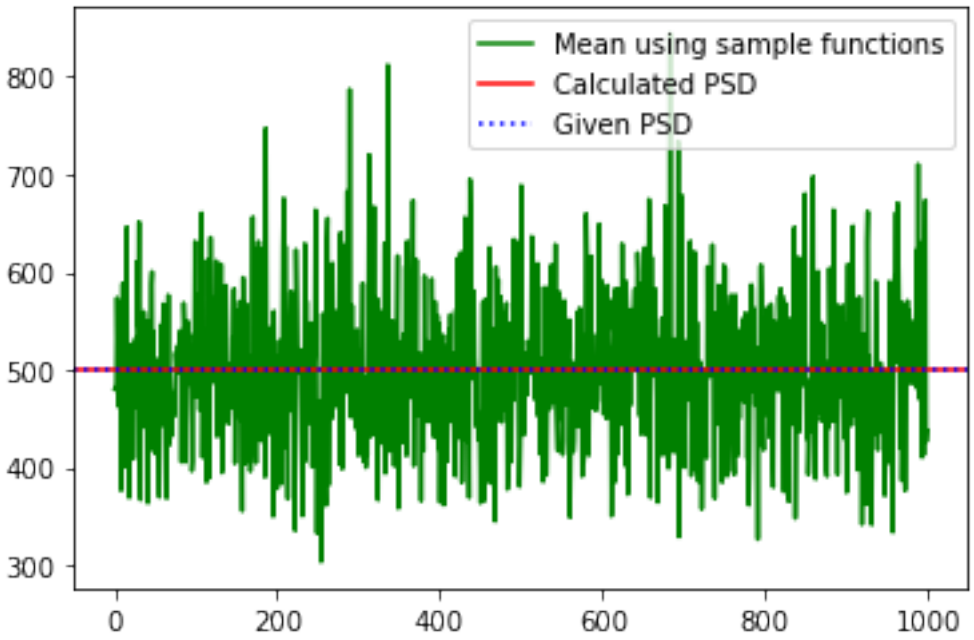
$$variance = 2 \times PSD \times Bandwidth$$

and find the standard deviation by taking it's square root. We multiply this value to the signal so that the variance of this random signal becomes equal to that of the signal with given PSD. Now we take the Fourier transform of this signal followed by multiplying the signal with its conjugate to find it's norm. Then, we take the mean of all the column vectors/experiments to get a row vector with dimension equal to length of the sequence. Next, we again take the mean of all these values and divide it by total time T, given by

$$T = Fs \times l$$

Where l is the length of the sequence and Fs is the sampling frequency.
This gives us one value for PSD. The PSD obtained using these calculations is equal to the value of PSD that we took as the input argument hence verifying our code.

- Sampling Frequency, fs = 200
- Bandwidth, B = 100
- Power Spectral Density, PSD = 500
- Length of sequence, l = 1000



2. We have to generate samples of Gaussian distribution using mygauss() which takes Nx1 mean vector, NxN covariance matrix and number of samples, which is a large number as inputs. We will generate Nx(number of samples) as the Gaussian distribution, lets call them Y. We use the randn.m function which considers the mean to be 0 and variance to be 1 in the output. This gives us some values, say X having mean = 0 and variance = 1. Our desired mean and covariance are the ones we passed as argumemts to the function. We use library functions to find the eigen vector and eigen values of covariance matrix given as input and we also use the eigen decomposition to decompose the input covariance matrix as

$$C = \Lambda\Sigma^2\Lambda^T$$

$$C = (\Lambda\Sigma)(\Sigma^T\Lambda^T)$$

$$C = AA^T$$

where,

$$A = \Lambda\Sigma$$

Now we know,

$$Y = AX$$

Thus we get our gaussian distribution, with covariance as C instead of 1 now. However the mean of Y will still remain approximately 0. Thus we need to add the Nx1 input mean vector to the this in order to get the correct values. These will now be compared to the given input mean and covariance matrix. They are observed to be very close in values thus verifying our code. We also checked if this covariance matrix was SPD and it indeed was.

```
Entered Mean Vector:
[ 1  3 -5]

Entered Covariance Matrix:
[[ 2 -1  0]
 [-1  2 -1]
 [ 0 -1  2]]

Samples:
[[ 1.07126392  1.57325674 -1.99812429]
 [ 1.8310989   2.95882931 -5.24555917]
 [ 0.24695964  1.16748971 -4.66019658]
 ...
 [ 1.80900553  6.01039805 -6.02386182]
 [ 3.8905174   0.82114425 -6.22760604]
 [ 1.70671262  5.53382482 -9.03190338]]

Calculated Mean Vector:
[1.0353198096946992, 2.953741212848046, -4.984730914314471]

Calculated Covariance Matrix:
[[ 2.0194349  -1.02033005  0.01733812]
 [-1.02033005  2.0414831  -1.02638963]
 [ 0.01733812 -1.02638963  2.01643282]]

Calculated matrix SPD?: True
```