PS Implementation

In [10]:
```
st=time.time()

golden_output = np.fft.fft(data, samples)

et=time.time()
print(et-st)
```

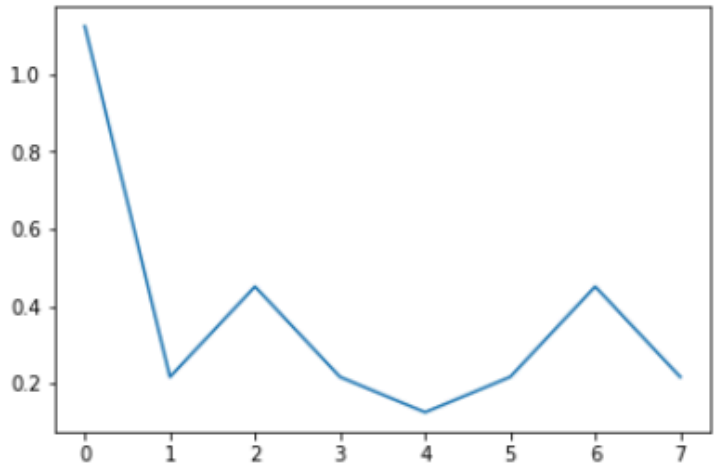
```
0.003037691116333008
```

In [11]:
```
freq = np.fft.fftfreq(samples)
freq
```

Out[11]:
```
array([ 0.   ,  0.125,  0.25 ,  0.375, -0.5  , -0.375, -0.25 , -0.125])
```

In [12]:
```
# freq.sort()
plt.plot((np.abs(golden_output))/samples)
```

Out[12]:
```
[<matplotlib.lines.Line2D at 0xaee0ea30>]
```



PL Implementation

In [13]:
```
from pynq import allocate
```

In [14]:
```
input_buffer = allocate(8,np.csingle)
output_buffer = allocate(8,np.csingle)
```

In [15]:
```
send_channel = data_channel.sendchannel
recv_channel = data_channel.recvchannel
```

In [16]:
```
np.copyto(input_buffer, data)
```

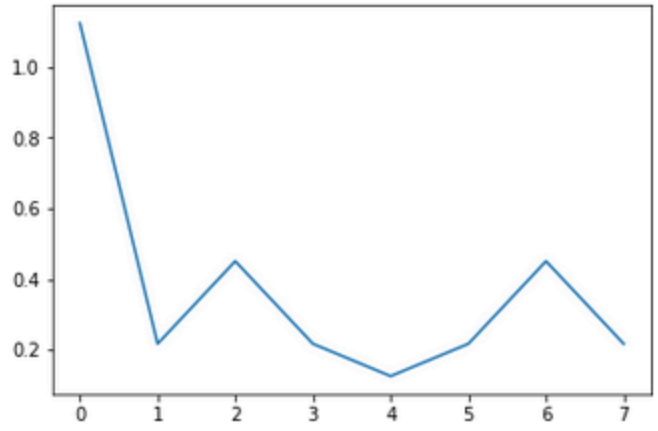
In [17]:
```
input_buffer?
```

In [18]:
```
st=time.time()

send_channel.transfer(input_buffer)
recv_channel.transfer(output_buffer)
send_channel.wait()
recv_channel.wait()

et=time.time()
print(et-st)
```

```
0.0021741390228271484
```

In [19]:
```
output = np.array([0]*8, dtype = np.cdouble)
np.copyto(output, output_buffer)
plt.plot((np.abs(output))/samples)
```

Out[19]:
```
[<matplotlib.lines.Line2D at 0xaed576d0>]
```



Difference between PS and PL Output

In [20]:
```
plt.plot(golden_output - output)
```

```
/usr/lib/python3/dist-packages/numpy/core/numeric.py:531: ComplexWarning: Casting complex values to real discards the imagina
ry part
  return array(a, dtype, copy=False, order=order)
```

Out[20]:
```
[<matplotlib.lines.Line2D at 0xaed128b0>]
```