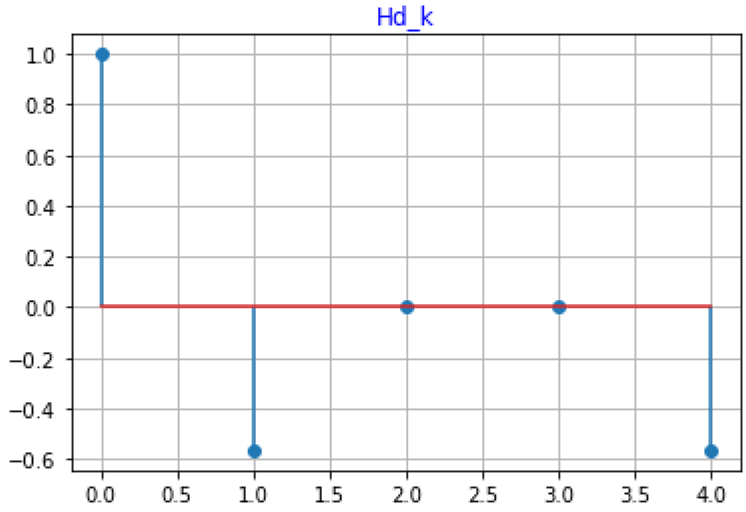
In [ ]:

```
# experiment 6 - Design a low pass filter for following specification
# Assume that we want a filter which is having samples |Hd(k)| = [1,0.7,0,0,0.7]
# Wc = pi/5
# N = 5


import numpy as np
import matplotlib.pyplot as plt
import math

pi = np.pi

hd_k_magn = [1,0.7,0,0,0.7]
hd_k = []
tow  = 2
N = 5
for k in range(len(hd_k_magn)):
    x = np.exp((-1j*2*pi*k*tow)/N)
    hd_k.append(hd_k_magn[k]*x)

# print(hd_k)

n = np.arange(0,5)
plt.stem(n,hd_k)
plt.grid()
plt.title('Hd_k',color='b')
```

```
c:\ProgramData\Anaconda3\lib\site-packages\numpy\ma\core.py:3375: ComplexWarning: Castin
g complex values to real discards the imaginary part
  _data[indx] = dval
c:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_asarray.py:102: ComplexWarning: C
asting complex values to real discards the imaginary part
  return array(a, dtype, copy=False, order=order)
```

Out[ ]:

```
Text(0.5, 1.0, 'Hd_k')
```



In [ ]:

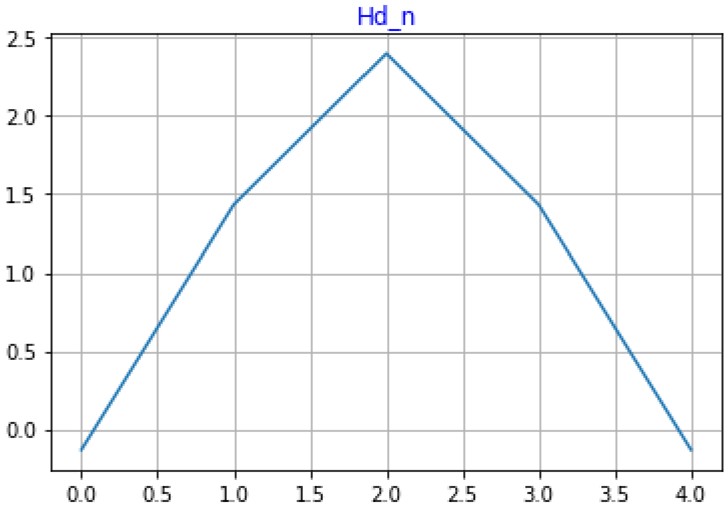
```
# Now hd[n]

# hd[n] = sumation 0 to N-1 (hd_k*np.exp((1j*2*pi*k*n)/N)

Hd_n = []
```

```
N = 5

for n in range(0,N):
    sum = 0
    for k in range(0,N):
        sum += hd_k[k]* np.exp((1j*2*pi*k*n)/N)
    Hd_n.append(sum)

n = np.arange(0,5)
plt.plot(n,Hd_n)
plt.grid()
plt.title('Hd_n',color='b')
```

```
c:\ProgramData\Anaconda3\lib\site-packages\numpy\core\_asarray.py:102: ComplexWarning: C
asting complex values to real discards the imaginary part
  return array(a, dtype, copy=False, order=order)
```

Out[ ]:

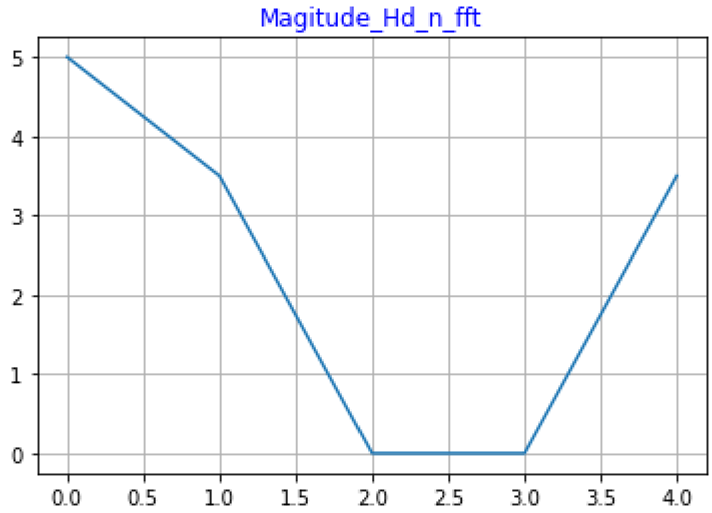
```
Text(0.5, 1.0, 'Hd_n')
```



In [ ]:

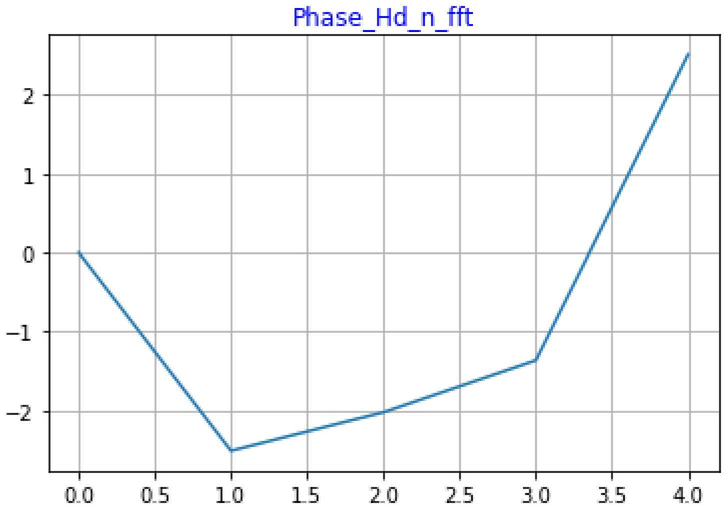
```
# magnitude response
Hd_n_fft =  np.fft.fft(Hd_n)
plt.plot(abs(Hd_n_fft))
plt.grid()
plt.title('Magitude_Hd_n_fft',color='b')
```

```
Out[ ]: Text(0.5, 1.0, 'Magitude_Hd_n_fft')
```



```
In [ ]: # Phase response
        plt.plot(np.angle(Hd_n_fft))
        plt.grid()
        plt.title('Phase_Hd_n_fft',color='b')
```

```
Out[ ]: Text(0.5, 1.0, 'Phase_Hd_n_fft')
```


Phase_Hd_n_fft