```python
# This script plots the frequency response of a low-pass filter and transformed versions
import numpy as np
import matplotlib.pyplot as plt

# frequency array
w_array = np.arange(-np.pi, np.pi + 1, 0.01)

# low-pass filter
H1p = []
for w in w_array:
    if np.abs(w) < 0.2 * np.pi:
        H1p.append(1)
    else:
        H1p.append(0)

# plot low-pass filter
plt.plot(w_array, H1p)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 1.5)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()

# part (a)
H1 = []
for w in w_array:
    if (np.abs(w) < 1.2 * np.pi) and (np.abs(w) > 0.8 * np.pi):
        H1.append(1)
    else:
        H1.append(0)

# plot H1
plt.plot(w_array, H1)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 1.5)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()

# part (b)
H2 = []
for w in w_array:
    if (np.abs(w) < 0.7 * np.pi) and (np.abs(w) > 0.3 * np.pi):
        H2.append(1)
    else:
        H2.append(0)

# plot H2
plt.plot(w_array, H2)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 1.5)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()

# part (c)
H3 = []
for w in w_array:
    if (np.abs(w) < 0.1 * np.pi):
        H3.append(0.1)
    elif (w >= -0.3 * np.pi) and (w <= -0.1 * np.pi):
        H3.append(w / (2 * np.pi) + 0.15)
    elif (w >= 0.1 * np.pi) and (w <= 0.3 * np.pi):
        H3.append(-w / (2 * np.pi) + 0.15)
    else:
        H3.append(0)
```

```
# plot H3
plt.plot(w_array, H3)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 0.15)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()
```