Question 2 & 3

This script multiplies polynomials/integers in vector representation using convolution and the discrete Fourier transform.

In [ ]:
```
import numpy as np

from scipy.signal import convolve
from scipy.fft import fft, ifft
```

Question 2

In [ ]:
```
# Define the polynomial coefficients vectors
x = np.array([1, 2, 6, 11, 15, 12], dtype=np.float64)
y = np.array([1, -3, -3, 7, -7, 3], dtype=np.float64)
```

In [ ]:
```
# Multiply by direct convolution
z_conv = convolve(x, y, mode="full", method="direct")

# Multiply by converting to Fourier domain
x_padded = np.pad(x, (0, len(y) - 1))
y_padded = np.pad(y, (0, len(x) - 1))
z_ffts = ifft(fft(x_padded) * fft(y_padded)).real

print("Q2 by convolution: ", z_conv)
print("Q2 by FFT and IFFT:", z_ffts)
```

```
Q2 by convolution:  [  1.  -1.  -3.  -6. -29. -35. -40.  10.  12. -39.  36.]
Q2 by FFT and IFFT: [  1.  -1.  -3.  -6. -29. -35. -40.  10.  12. -39.  36.]
```

Question 3

In [ ]:
```
# Define the integer multiplicands as vectors
x = np.array([8, 7, 5, 5, 7, 9, 0], dtype=np.float64)
y = np.array([1, 3, 6, 7, 2, 6, 7], dtype=np.float64)
```

In [ ]:
```
# Define the "carry" operation used for integer
def multiply_carry(z: np.array) -> np.array:
    """
    Perform the "carry" steps of the multiplication process. Starting from the
    right end of `z`, each digit is taken modulo 10 and the remainder is added
    to the value immediately to the left. Returns an array of single digits,
    possibly except the first value (though the answer will still be correct).
    """
    r = 0; ret = []
    for n in z[::-1]:
        n += r
        ret.append(n % 10)
        r = n // 10
    ret.append(r)
    return np.array(ret[::-1])
```

In [ ]:
```
# Multiply by direct convolution
z_conv = multiply_carry(convolve(x, y, mode="full", method="direct"))

# Multiply by converting to Fourier domain
x_padded = np.pad(x, (0, len(y) - 1))
y_padded = np.pad(y, (0, len(x) - 1))
z_ffts = multiply_carry(ifft(fft(x_padded) * fft(y_padded)).real)
```

```python
print("Q3 by multiplication:", 8755790 * 1367267)
print("Q3 by convolution:    ", z_conv)
print("Q3 by FFT and IFFT:   ", z_conv)
```

```
Q3 by multiplication: 11971502725930
Q3 by convolution:    [1. 1. 9. 7. 1. 5. 0. 2. 7. 2. 5. 9. 3. 0.]
Q3 by FFT and IFFT:   [1. 1. 9. 7. 1. 5. 0. 2. 7. 2. 5. 9. 3. 0.]
```