1.

```
"""DFT """
import math

def iexp(n):
	return complex(math.cos(n), math.sin(n))

def is_pow2(n):
	return False if n == 0 else (n == 1 or is_pow2(n >> 1))

def dft(xs):
	"naive dft"
	n = len(xs)
	return [sum((xs[k] * iexp(-2 * math.pi * i * k / n) for k in range(n)))
			for i in range(n)]

def dftinv(xs):
	"naive dft"
	n = len(xs)
	return [sum((xs[k] * iexp(2 * math.pi * i * k / n) for k in range(n))) / n
			for i in range(n)]

if __name__ == "__main__":
	wave1 = [1,0,0,0,0,0,0,0]
	wave2 = [1,1,1,1,1,1,1,1]
	wave3 = [1,-1,1,-1,1,-1,1,-1]
	wave4 = [3,0,2,0,2,0,2,0]
	dfreq5 = [1,1,0,0,0,0,1,1]
	dfreq6 = [1,1,0,0,0,0,0,1]
	dfreq1 = dft(wave1)
	dfreq2 = dft(wave2)
	dfreq3 = dft(wave3)
	dfreq4 = dft(wave4)
	wave5 = dftinv(dfreq5)
	wave6 = dftinv(dfreq6)
	print(dfreq1)
	print(dfreq2)
	print(dfreq3)
```

```
    print(dfreq4)
    print(wave5)
    print(wave6)
    pass
```

[(1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (1+0j)]

[(8+0j), (-4.440892098500626e-16+2.220446049250313e-16j), (-4.286263797015736e-16-4.440892098500626e-16j), (-3.3306690738754696e-16+8.881784197001252e-16j), -4.898587196589413e-16j, (-2.1094237467877974e-15-1.2212453270876722e-15j), (-2.9329683544708742e-15-6.661338147750939e-16j), (3.4416913763379853e-15+1.1102230246251565e-15j)]

[0j, -2.220446049250313e-16j, (9.555947231402665e-17-1.1102230246251565e-16j), (9.992007221626409e-16-1.5543122344752192e-15j), (8+3.4290110376125885e-15j), (-2.6645352591003757e-15+1.1102230246251565e-16j), (2.9329683544708742e-15-6.661338147750939e-16j), (-5.218048215738236e-15-2.6645352591003757e-15j)]

[(9+0j), (0.9999999999999997-4.440892098500626e-16j), (1-4.898587196589412e-16j), (1.0000000000000007-6.661338147750939e-16j), (9+2.9391523179536475e-15j), (0.999999999999995-1.3322676295501878e-15j), (1-1.469576158976824e-15j), (0.9999999999999981-1.7763568394002505e-15j)]

[(0.5+0j), (0.30177669529663687-0.12500000000000003j), (-5.35782974626967e-17+5.551115123125783e-17j), (-0.051776695296636796+0.12499999999999997j), 3.0616169978683836e-17j, (-0.05177669529663707-0.12499999999999982j), (-8.906528815257012e-17+1.3877787807814457e-16j), (0.3017766952966372+0.12499999999999961j)]

[(0.375+0j), (0.30177669529663687-2.7755575615628914e-17j), (0.12499999999999994+0j), (-0.051776695296636865-1.3877787807814457e-17j), (-0.125+1.2246467991473532e-16j), (-

0.05177669529663673+1.8041124150158794e-16j), (0.12499999999999992+0j), (0.30177669529663725-3.885780586188048e-16j)]

3.摺積定理指出，函數摺積的傅立葉轉換是函數傅立葉轉換的乘積。即一個域中的摺積對應於另一個域中的乘積，例如時域中的摺積對應於頻域中的乘積。「迴積分定理」亦有譯為「迴旋積分定理」者，其主要應用為幫助 Laplace 積分反轉換之進行。