

快速傅里叶变换 与 信号频谱分析 实验报告

姓名: Cantjie

一 . 实验目的

1. 在理论学习的基础上, 通过本实验加深对离散傅里叶变换的理解。
2. 熟悉并掌握按时间抽取编写快速傅里叶变换 (FFT) 算法的程序。
3. 了解应用 FFT 进行信号频谱分析过程中可能出现的问题, 例如频谱混淆、泄漏、栅栏效应等, 以便在实际中正确使用 FFT 算法进行信号处理。

二 . 实验内容

1. 仔细分析教材第六章“时间抽取法 FFT 的 FORTRAN 程序”, 编写出相应的使用 FFT 进行信号频谱分析的 Matlab 程序。

主程序 exp1_fft.m (人机交互, 参数设置, 作图) :

```
clc,clear;

F = input('Signal frequency:');
L = input('Length of signal:');
padding_zero = input('padding zero(0/1):');
Fs = input('Sampling frequency(Fs) or Sampling period(T):');
Ph = input('Phase at t = 0 :');
if Fs < 0.01
    T = Fs;
    Fs = 1/T;
else
    T = 1/Fs;
end

t = (0:L-1)*T;          % Time vector
S = sin(2*pi*F*t + Ph);

%补领
if padding_zero == 1
    N2 = input('Numbers of zeros:');
    S(L+1:L+N2) = 0;
    L = L + N2;
end

t = (0:L-1)*T;          % Time vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t_con = (0:4*L-1)*(T/4); %画连续图像时采样密集一点
S_con = sin(2*pi*F*t_con+Ph);
subplot(2,2,1);
plot(t_con,S_con);
```

```

title('Signal')
xlabel('t')
ylabel('S(t)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,2); %时域采样图
stem(t,S);
title('Signal after sampling')
xlabel('t')
ylabel('S(t)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X = abs(my_fft(S,L)); %FFT 结果
subplot(2,2,3);
stem([0:L-1],X);
title('FFT')
xlabel('k')
ylabel('|X(k)|')

```

FFT 实现(时域抽取, 可自动补零):

```

function y = my_fft(signal, N)
%时域抽取实现 fft, 可实现自动补零
%params:
%  signal 信号
%  N 信号长度
%return:
%  vector of length N;
%author: Cantjie(cantjie@xjtu.edu.cn);
%updated at: 2018-10-13

    len = length(signal);

    %通过补 0 的方法使信号达到 2 的整数次幂的长度
    layer = ceil(log2(N));
    N = pow2(layer);
    if(len >= N)
        y = signal(1:N);
    else
        y = signal;
        y(len + 1:N) = 0;
    end
    %信号反转
    y = bitrevorder(y);

    W_N = exp(j * 2 * pi / N);
    for cur_layer = 1:layer
        N_half = pow2(cur_layer - 1); %第 cur_layer 层中每组点数的一半
        g = pow2(layer - cur_layer); %第 cur_layer 层有多少组
        for i = 1:g
            cur_base = 2 * N_half * (i-1); %正在处理的那组的基地址
            W_N_g = power(W_N,g);

            %由蝶形运算得到该组前一半和对应的后一半
            for k = 1:N_half
                W_N_gk = power(W_N_g,k-1);
                temp_k = y(cur_base + k);
                temp_k_j = y(cur_base + N_half + k);
            end
        end
    end
end

```

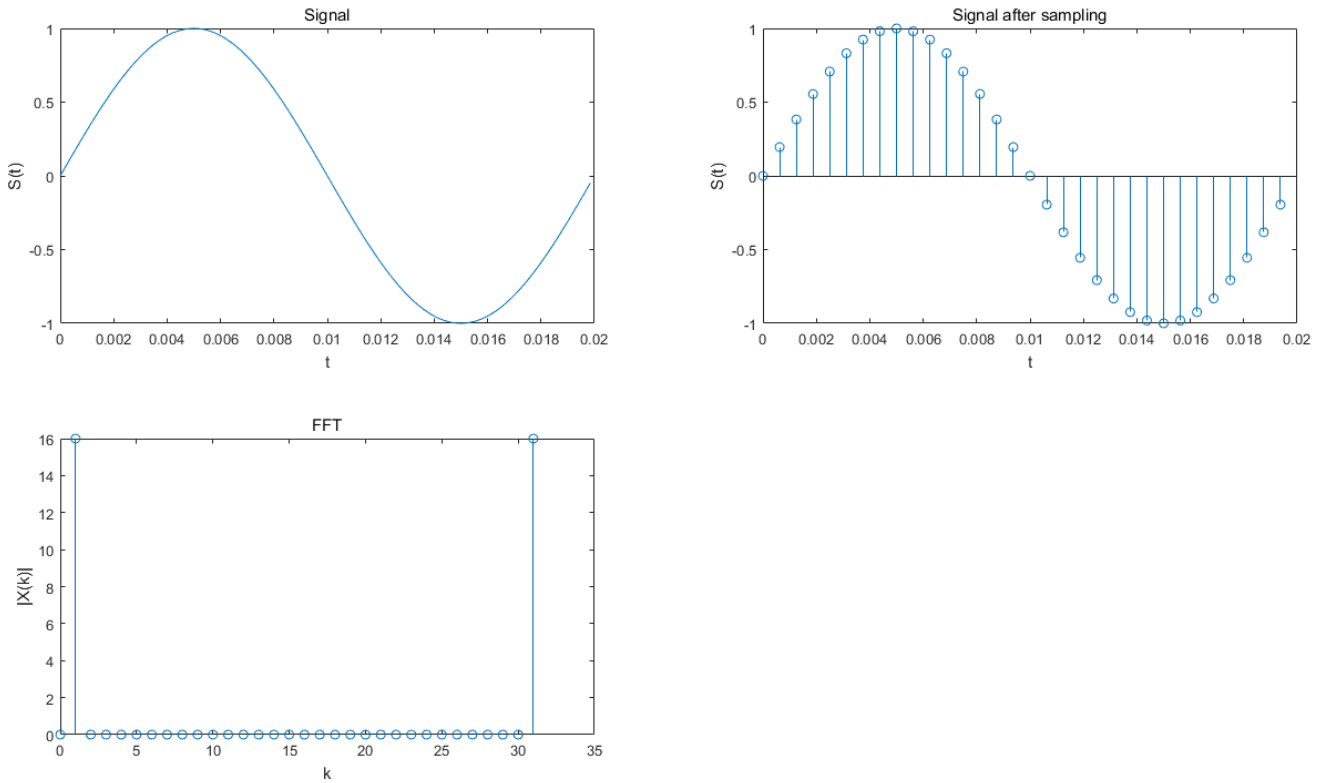
```

y(cur_base + k) = temp_k + temp_k_j * W_N_gk;
y(cur_base + N_half + k) = temp_k - temp_k_j * W_N_gk;
end
end
end
end

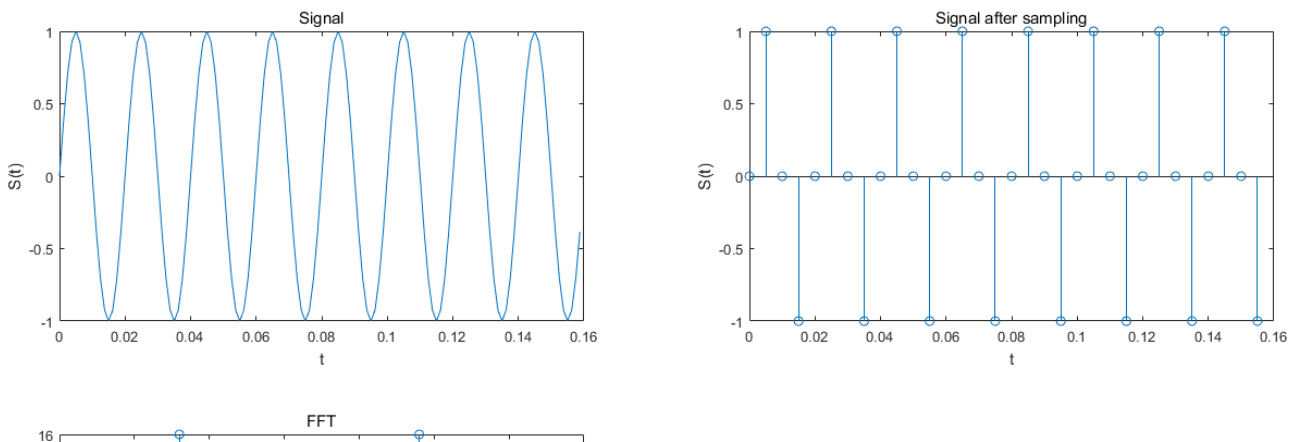
```

2. 用 FFT 程序分析正弦信号，分别在以下情况进行分析，并讨论所得的结果：
(各情况下的 X(k)具体值用列表附在报告结尾)

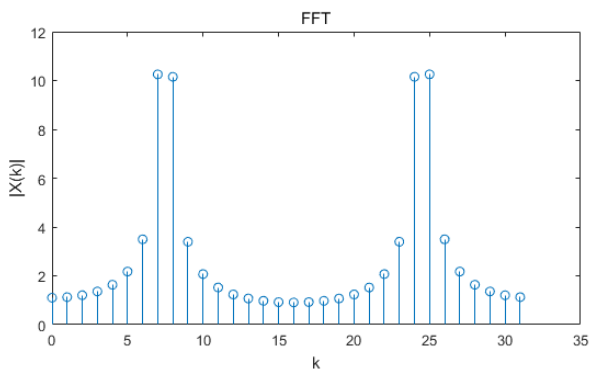
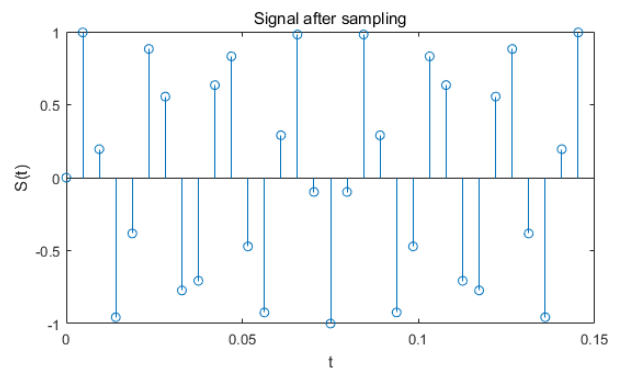
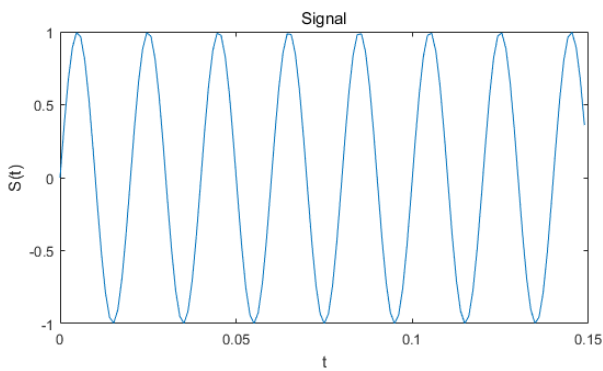
a) 信号频率 $F = 50\text{Hz}$ ，采样点数 $N = 32$ ，采样间隔 $T = 0.000625\text{s}$;
 信号频率 $F = 50\text{Hz}$ ，采样长 $N = 32$ ，采样周期 $T = 0.000625\text{s}$ ， $f_s = 1/T = 1600\text{Hz}$ ，基频为 $f_s/N = 50\text{Hz}$ ， $50/50 = 1$ 。故此在频谱图上的第 1 个点和第 31 个点有值



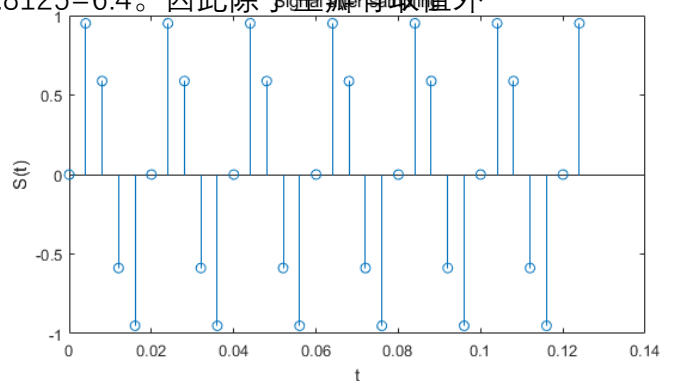
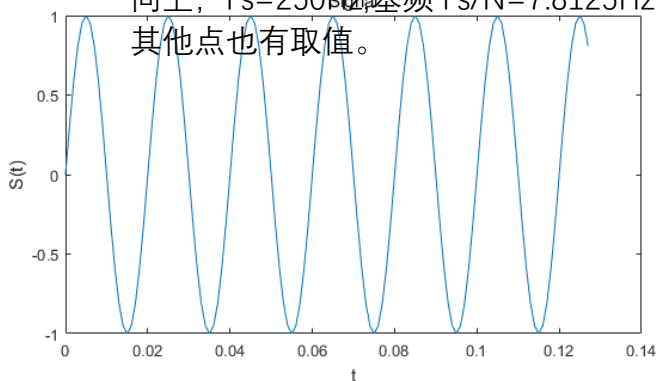
b) 信号频率 $F = 50\text{Hz}$ ，采样点数 $N = 32$ ，采样间隔 $T = 0.005\text{s}$;
 信号频率 $F = 50\text{Hz}$ ，采样长 $N = 32$ ，采样周期 $T = 0.005\text{s}$ ， $f_s = 1/T = 200\text{Hz}$ ，基频为 $f_s/N = 6.25\text{Hz}$ ， $50/6.25 = 8$ 。故此在频谱图上的第 8 个点和第 24 个点有值



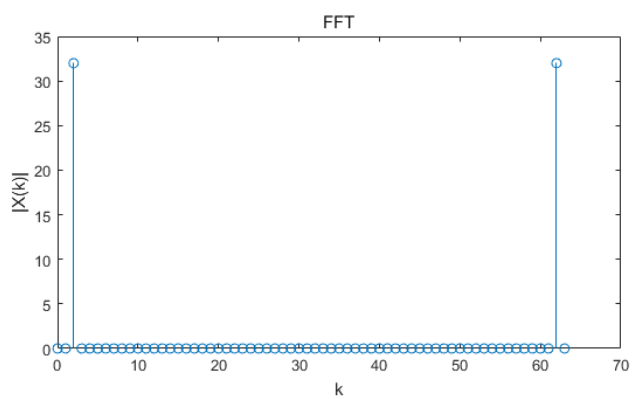
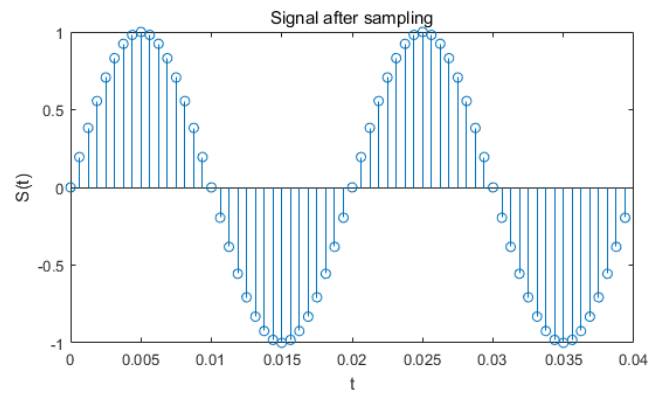
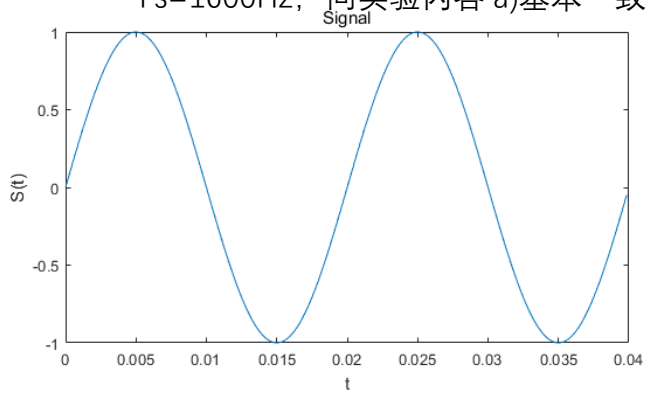
c) 信号频率 $F = 50\text{Hz}$, 采样点数 $N=32$, 采样间隔 $T=0.0046875\text{s}$;
 $F_s=1/T=213.33\text{Hz}$, 基频为 $F_s/N=6.66\text{Hz}$, $50/6.66=7.5$ 。不是整肃周期, 发生了泄露。因此除了主瓣上有取值外还在非整周期处取值。



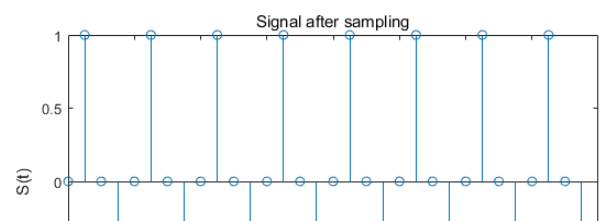
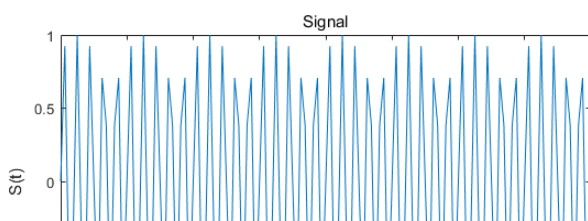
d) 信号频率 $F = 50\text{Hz}$, 采样点数 $N=32$, 采样间隔 $T=0.004\text{s}$;
 同上, $F_s=250\text{Hz}$, 基频 $F_s/N=7.8125\text{Hz}$, $50/7.8125=6.4$ 。因此除了主瓣有取值外其他点也有取值。



e) 信号频率 $F = 50\text{Hz}$, 采样点数 $N=64$, 采样间隔 $T=0.000625\text{s}$;
 $F_s=1600\text{Hz}$, 同实验内容 a)基本一致。

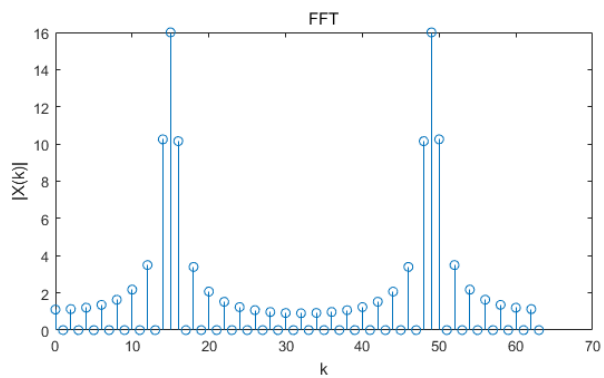
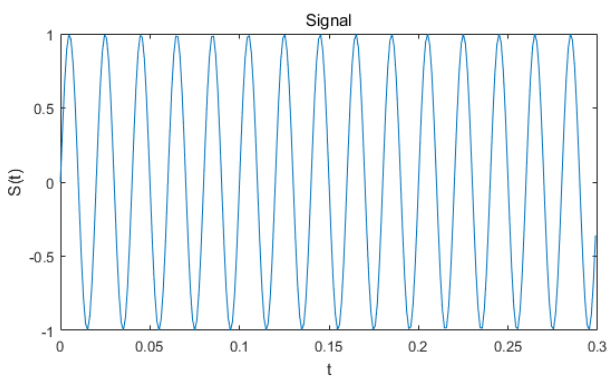


f) 信号频率 $F = 250\text{Hz}$, 采样点数 $N=32$, 采样间隔 $T=0.005\text{s}$;
 $F_s=200\text{Hz}$, $F_s < F$, 产生了失真



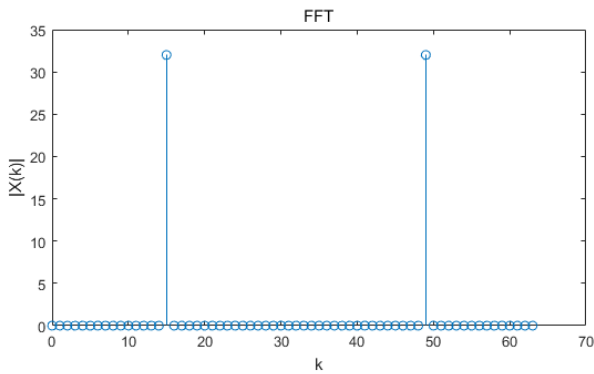
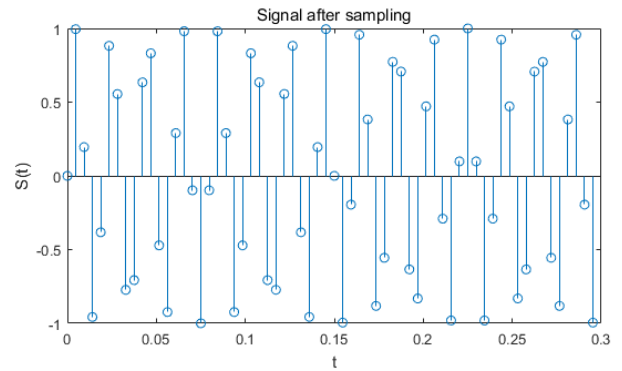
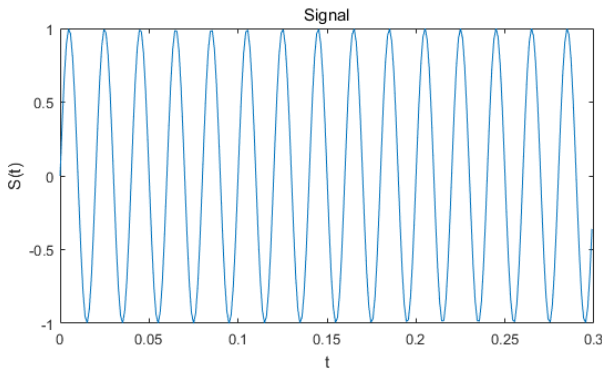
g) 将 c)中信号后补 32 个 0, 做 64 点 FFT, 并与直接采样 64 个点做 FFT 的结果进行对比。

①补零, 64 点 FFT



k	X(k) in task a	X(k) in task b	X(k) in task c	X(k) in task d	X(k) in task f
---	----------------	----------------	----------------	----------------	----------------

②未补零, 64 点 FFT



三. 思考题:

1) 在实验 a)、b)、c)和 d)中, 正弦信号的初始相位对频谱图中的幅度特性是否有影响? 为什么?

答: 没有影响。因为频谱图的幅度特性仅与所对应频率的幅值大小有关, 与初始相位无关。换言之, 频谱图的幅度特性是从各频率分量功率的特性, 而初始相位并不影响功率大小。

2) 信号补零后做 FFT 是否可以提高信号频谱的分辨率? 为什么?

答: 对于频率分辨率, 百度百科有两种解释, 其一是: 在频率轴上的所能得到的最小频率间隔

$$f_0 = \frac{f_s}{N} = \frac{1}{Nt_s} = \frac{1}{T}$$
 , 其中 N 为采样点数, f_s 为采样频率, t_s 为采样间隔。所以 Nt_s 就是采样前模拟信号的时间长度 T , 所以信号长度越长, 频率分辨率越好。而相当于对变换结果做了插值, 克服“栅栏”效应, 使谱外观平滑化, 但并没有增加有效数据长度和时间长度, 因此并没有提高分辨率。相反, 我们可以通过 g 中的两个频谱图发现, 补 0 后得到的频谱图并不是我们期望的仅有 50Hz 峰值的频谱图了。

1	0.0000 + 0.0000i	-0.0000 + 0.0000i	1.1033 + 0.0000i	0.9511 + 0.0000i	0.0000 + 0.0000i
2	0.0000 +16.0000i	0.0000 - 0.0000i	1.1273 + 0.0000i	0.9867 + 0.0854i	0.0000 - 0.0000i
3	0.0000 - 0.0000i	0.0000 + 0.0000i	1.2050 + 0.0000i	1.1052 + 0.1829i	0.0000 + 0.0000i
4	-0.0000 - 0.0000i	-0.0000 - 0.0000i	1.3568 - 0.0000i	1.3526 + 0.3125i	0.0000 - 0.0000i
5	-0.0000 + 0.0000i	-0.0000 + 0.0000i	1.6339 + 0.0000i	1.8670 + 0.5220i	-0.0000 - 0.0000i
6	-0.0000 - 0.0000i	0.0000 + 0.0000i	2.1750 + 0.0000i	3.1952 + 0.9911i	0.0000 - 0.0000i
7	0.0000 + 0.0000i	-0.0000 + 0.0000i	3.4960 - 0.0000i	11.3836 + 3.6858i	-0.0000 - 0.0000i
8	0.0000 - 0.0000i	0.0000 + 0.0000i	10.2519 + 0.0000i	-7.8447 - 2.5301i	-0.0000 - 0.0000i
9	0.0000 - 0.0000i	0.0000 +16.0000i	-10.1532 - 0.0000i	-3.0777 - 0.9511i	-0.0000 +16.0000i
10	-0.0000 - 0.0000i	0.0000 - 0.0000i	-3.3953 - 0.0000i	-2.0004 - 0.5718i	-0.0000 + 0.0000i
11	-0.0000 - 0.0000i	-0.0000 - 0.0000i	-2.0703 + 0.0000i	-1.5376 - 0.3925i	-0.0000 + 0.0000i
12	-0.0000 + 0.0000i	0.0000 - 0.0000i	-1.5226 + 0.0000i	-1.2889 - 0.2826i	0.0000 + 0.0000i
13	-0.0000 - 0.0000i	-0.0000 - 0.0000i	-1.2361 + 0.0000i	-1.1405 - 0.2045i	-0.0000 + 0.0000i
14	-0.0000 + 0.0000i	-0.0000 + 0.0000i	-1.0707 - 0.0000i	-1.0482 - 0.1432i	0.0000 + 0.0000i
15	-0.0000 + 0.0000i	0.0000 - 0.0000i	-0.9739 - 0.0000i	-0.9916 - 0.0912i	0.0000 - 0.0000i
16	0.0000 + 0.0000i	0.0000 + 0.0000i	-0.9225 - 0.0000i	-0.9608 - 0.0445i	0.0000 + 0.0000i
17	0.0000 + 0.0000i	-0.0000 + 0.0000i	-0.9063 + 0.0000i	-0.9511 + 0.0000i	0.0000 + 0.0000i
18	0.0000 + 0.0000i	0.0000 - 0.0000i	-0.9225 + 0.0000i	-0.9608 + 0.0445i	0.0000 - 0.0000i
19	-0.0000 - 0.0000i	0.0000 + 0.0000i	-0.9739 + 0.0000i	-0.9916 + 0.0912i	0.0000 + 0.0000i
20	-0.0000 - 0.0000i	-0.0000 - 0.0000i	-1.0707 + 0.0000i	-1.0482 + 0.1432i	0.0000 - 0.0000i
21	-0.0000 + 0.0000i	-0.0000 + 0.0000i	-1.2361 - 0.0000i	-1.1405 + 0.2045i	-0.0000 - 0.0000i
22	-0.0000 - 0.0000i	0.0000 + 0.0000i	-1.5226 - 0.0000i	-1.2889 + 0.2826i	0.0000 - 0.0000i
23	-0.0000 + 0.0000i	-0.0000 + 0.0000i	-2.0703 - 0.0000i	-1.5376 + 0.3925i	-0.0000 - 0.0000i
24	-0.0000 + 0.0000i	0.0000 + 0.0000i	-3.3953 + 0.0000i	-2.0004 + 0.5718i	-0.0000 - 0.0000i
25	0.0000 + 0.0000i	0.0000 -16.0000i	-10.1532 + 0.0000i	-3.0777 + 0.9511i	-0.0000 -16.0000i
26	0.0000 + 0.0000i	0.0000 - 0.0000i	10.2519 - 0.0000i	-7.8447 + 2.5301i	-0.0000 + 0.0000i
27	0.0000 - 0.0000i	-0.0000 - 0.0000i	3.4960 + 0.0000i	11.3836 - 3.6858i	-0.0000 + 0.0000i
28	-0.0000 + 0.0000i	0.0000 - 0.0000i	2.1750 - 0.0000i	3.1952 - 0.9911i	0.0000 + 0.0000i
29	-0.0000 - 0.0000i	-0.0000 - 0.0000i	1.6339 - 0.0000i	1.8670 - 0.5220i	-0.0000 + 0.0000i
30	-0.0000 + 0.0000i	-0.0000 + 0.0000i	1.3568 + 0.0000i	1.3526 - 0.3125i	0.0000 + 0.0000i
31	0.0000 + 0.0000i	0.0000 - 0.0000i	1.2050 - 0.0000i	1.1052 - 0.1829i	0.0000 - 0.0000i
32	0.0000 -16.0000i	0.0000 + 0.0000i	1.1273 - 0.0000i	0.9867 - 0.0854i	0.0000 + 0.0000i

k	X(k) in task e	X(k) in task g(padding zeros)	X(k) in task g(not padding zeros)
1	-0.0000 + 0.0000i	1.1033 + 0.0000i	0.0000 + 0.0000i
2	-0.0000 + 0.0000i	0.0000 - 0.0000i	0.0000 - 0.0000i
3	0.0000 +32.0000i	1.1273 + 0.0000i	-0.0000 - 0.0000i
4	-0.0000 - 0.0000i	-0.0000 - 0.0000i	-0.0000 - 0.0000i
5	0.0000 - 0.0000i	1.2050 + 0.0000i	-0.0000 + 0.0000i
6	0.0000 + 0.0000i	-0.0000 + 0.0000i	-0.0000 + 0.0000i

7	0.0000 - 0.0000i	1.3568 - 0.0000i	0.0000 + 0.0000i
8	0.0000 - 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
9	-0.0000 + 0.0000i	1.6339 + 0.0000i	0.0000 - 0.0000i
10	0.0000 - 0.0000i	-0.0000 - 0.0000i	0.0000 - 0.0000i
11	-0.0000 - 0.0000i	2.1750 + 0.0000i	-0.0000 - 0.0000i
12	-0.0000 - 0.0000i	-0.0000 - 0.0000i	0.0000 - 0.0000i
13	-0.0000 - 0.0000i	3.4960 - 0.0000i	0.0000 - 0.0000i
14	-0.0000 - 0.0000i	0.0000 + 0.0000i	-0.0000 - 0.0000i
15	-0.0000 + 0.0000i	10.2519 + 0.0000i	-0.0000 - 0.0000i
16	0.0000 + 0.0000i	-0.0000 +16.0000i	-0.0000 +32.0000i
17	0.0000 - 0.0000i	-10.1532 - 0.0000i	-0.0000 + 0.0000i
18	-0.0000 + 0.0000i	0.0000 + 0.0000i	-0.0000 + 0.0000i
19	-0.0000 + 0.0000i	-3.3953 - 0.0000i	0.0000 + 0.0000i
20	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 - 0.0000i
21	0.0000 - 0.0000i	-2.0703 + 0.0000i	-0.0000 - 0.0000i
22	-0.0000 - 0.0000i	0.0000 - 0.0000i	-0.0000 + 0.0000i
23	-0.0000 + 0.0000i	-1.5226 + 0.0000i	0.0000 + 0.0000i
24	-0.0000 + 0.0000i	-0.0000 - 0.0000i	-0.0000 - 0.0000i
25	0.0000 - 0.0000i	-1.2361 + 0.0000i	-0.0000 + 0.0000i
26	-0.0000 - 0.0000i	-0.0000 + 0.0000i	0.0000 + 0.0000i
27	-0.0000 + 0.0000i	-1.0707 - 0.0000i	0.0000 + 0.0000i
28	0.0000 + 0.0000i	-0.0000 + 0.0000i	0.0000 + 0.0000i
29	0.0000 + 0.0000i	-0.9739 - 0.0000i	0.0000 + 0.0000i
30	0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
31	0.0000 + 0.0000i	-0.9225 - 0.0000i	0.0000 - 0.0000i
32	0.0000 - 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i
33	-0.0000 + 0.0000i	-0.9063 + 0.0000i	-0.0000 + 0.0000i
34	0.0000 + 0.0000i	0.0000 - 0.0000i	0.0000 - 0.0000i
35	0.0000 + 0.0000i	-0.9225 + 0.0000i	0.0000 + 0.0000i
36	0.0000 - 0.0000i	0.0000 - 0.0000i	0.0000 - 0.0000i
37	0.0000 - 0.0000i	-0.9739 + 0.0000i	0.0000 - 0.0000i
38	0.0000 - 0.0000i	-0.0000 - 0.0000i	0.0000 - 0.0000i
39	-0.0000 - 0.0000i	-1.0707 + 0.0000i	0.0000 - 0.0000i
40	-0.0000 + 0.0000i	-0.0000 - 0.0000i	0.0000 - 0.0000i
41	0.0000 + 0.0000i	-1.2361 - 0.0000i	-0.0000 - 0.0000i
42	-0.0000 - 0.0000i	-0.0000 + 0.0000i	-0.0000 + 0.0000i
43	-0.0000 - 0.0000i	-1.5226 - 0.0000i	0.0000 - 0.0000i
44	-0.0000 + 0.0000i	0.0000 + 0.0000i	-0.0000 - 0.0000i
45	0.0000 + 0.0000i	-2.0703 - 0.0000i	-0.0000 + 0.0000i
46	0.0000 - 0.0000i	0.0000 - 0.0000i	0.0000 + 0.0000i
47	-0.0000 - 0.0000i	-3.3953 + 0.0000i	0.0000 - 0.0000i
48	-0.0000 - 0.0000i	0.0000 - 0.0000i	-0.0000 - 0.0000i

49	$0.0000 + 0.0000i$	$-10.1532 + 0.0000i$	$-0.0000 - 0.0000i$
50	$0.0000 - 0.0000i$	$-0.0000 - 16.0000i$	$-0.0000 - 32.0000i$
51	$-0.0000 - 0.0000i$	$10.2519 - 0.0000i$	$-0.0000 + 0.0000i$
52	$-0.0000 + 0.0000i$	$0.0000 - 0.0000i$	$-0.0000 + 0.0000i$
53	$-0.0000 + 0.0000i$	$3.4960 + 0.0000i$	$0.0000 + 0.0000i$
54	$-0.0000 + 0.0000i$	$-0.0000 + 0.0000i$	$0.0000 + 0.0000i$
55	$-0.0000 + 0.0000i$	$2.1750 - 0.0000i$	$-0.0000 + 0.0000i$
56	$0.0000 + 0.0000i$	$-0.0000 + 0.0000i$	$0.0000 + 0.0000i$
57	$-0.0000 - 0.0000i$	$1.6339 - 0.0000i$	$0.0000 + 0.0000i$
58	$0.0000 + 0.0000i$	$0.0000 - 0.0000i$	$0.0000 - 0.0000i$
59	$0.0000 + 0.0000i$	$1.3568 + 0.0000i$	$0.0000 - 0.0000i$
60	$0.0000 - 0.0000i$	$-0.0000 - 0.0000i$	$-0.0000 - 0.0000i$
61	$0.0000 + 0.0000i$	$1.2050 - 0.0000i$	$-0.0000 - 0.0000i$
62	$-0.0000 + 0.0000i$	$-0.0000 + 0.0000i$	$-0.0000 + 0.0000i$
63	$0.0000 - 32.0000i$	$1.1273 - 0.0000i$	$-0.0000 + 0.0000i$
64	$-0.0000 - 0.0000i$	$0.0000 + 0.0000i$	$0.0000 + 0.0000i$