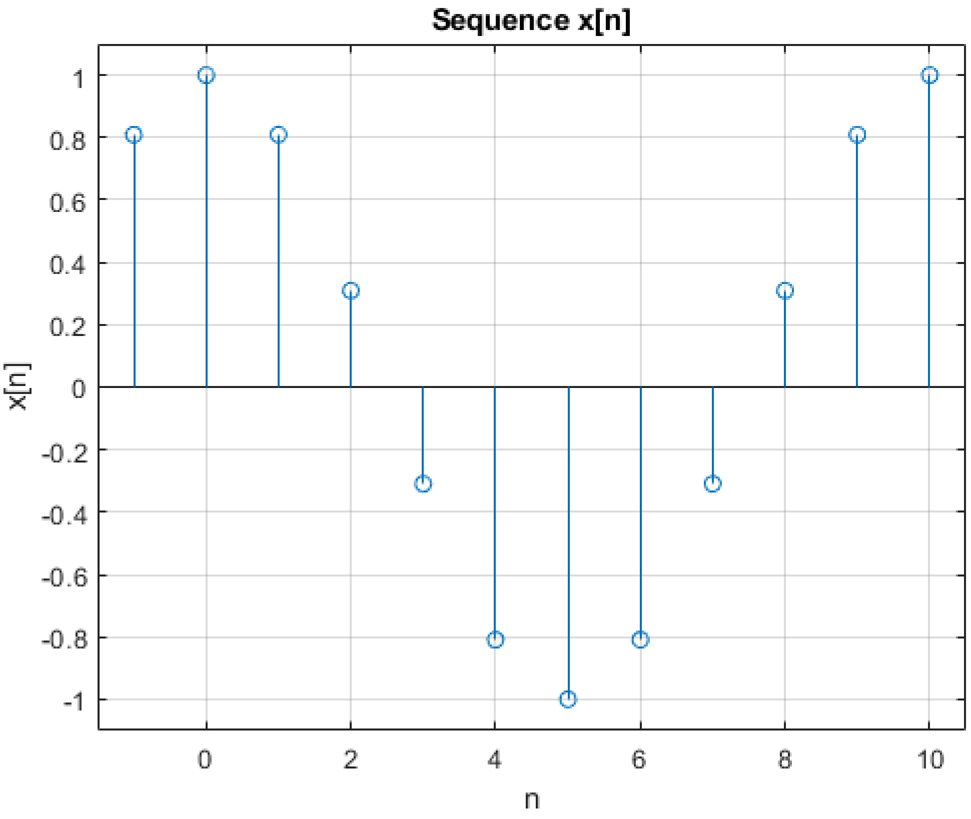
Table of Contents

```
% A pure sinusoidal sequence can be written as
% x[n] = A * cos(?*n + ?) = A * cos(2*pi*(f/fs)*n + ?)
%
% where ? = 2?f/fs is normalized angular frequency with frequency f
 and
% sampling frequency fs , amplitude A, and phase shift ?.
%
% The code for generating signal:
n = (-2 : 15);
A = 1;
theta = 0;
omega = 0.2*pi;
x = A * cos(omega * n + theta);
figure(1); clf; % open/activate Fig. 1, clean it
stem(n,x);
axis([-1.5 10.5 -1.1 1.1]); % zoom [xmin xmax ymin ymax]
grid on; xlabel('n'); ylabel('x[n]'); title('Sequence x[n]');
```



Task a)

The sampling frequency tells us how many samples there are during one second. (How many values have been sampled in one second.) If the sampling frequency fs = 10000 Hz, (I) how many samples are they in 0.5 seconds? (II) how long would 20000 samples last in seconds?

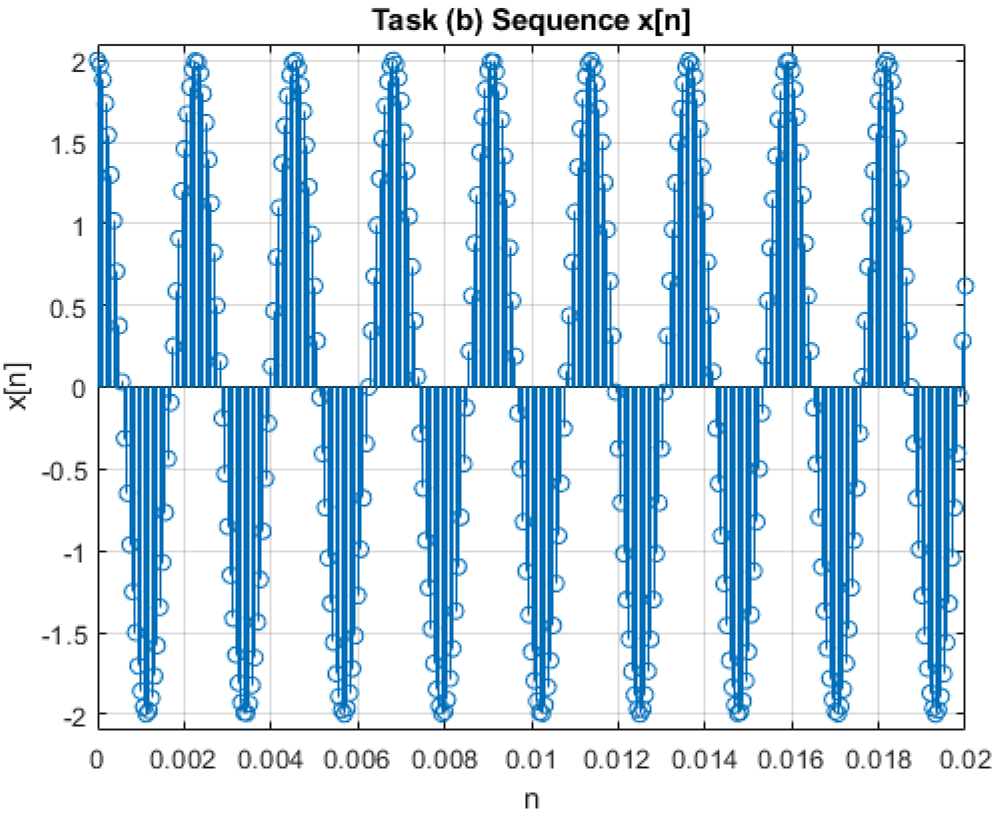
```
I = 5000;
II = 2;
```

Task b)

Generate a pure sinusoidal sequence whose frequency is f = 440 Hz, amplitude A = 2, length is 0.5 seconds, and sampling frequency is fs = 16000 Hz. Visualize it with stem and plot so that figure shows signal from 0 to 0.02 seconds , and listen with soundsc(x, fs).

```
fs = 16000; f = 440; A = 2;

n = (0 : (1/fs) : 0.5);
theta = 0;
omega = 2*f*pi;
x = A * cos(omega * n + theta);
figure(2); clf; % open/activate Fig. 1, clean it
stem(n,x);
axis([0 0.02 -2.1 2.1]); % zoom [xmin xmax ymin ymax]
grid on; xlabel('n'); ylabel('x[n]'); title('Task (b) Sequence x[n]');
soundsc(x, fs)
```

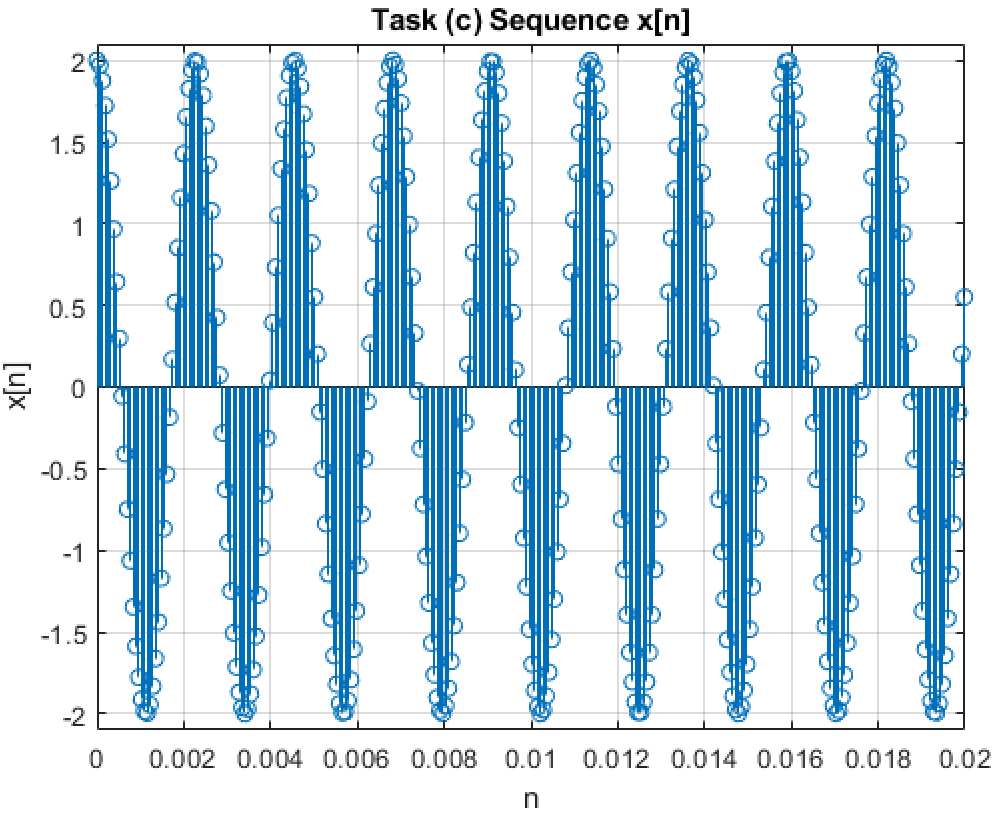


Task c)

Do the same but now f = 15560 Hz.

```
fs2 = 15560; f = 440; A = 2;

n = (0 : (1/fs2) : 0.5);
theta = 0;
omega = 2*f*pi;
x = A * cos(omega * n + theta);
figure(3); clf; % open/activate Fig. 1, clean it
stem(n,x);
axis([0 0.02 -2.1 2.1]); % zoom [xmin xmax ymin ymax]
grid on; xlabel('n'); ylabel('x[n]'); title('Task (c) Sequence x[n]');
soundsc(x, fs)
```



Published with MATLAB® R2020a