5(a) and 5(b): (Note: 5(a)'s .m code is shown below 5(b))

5(b) Task 2:

Code for plotting Sawtooth:

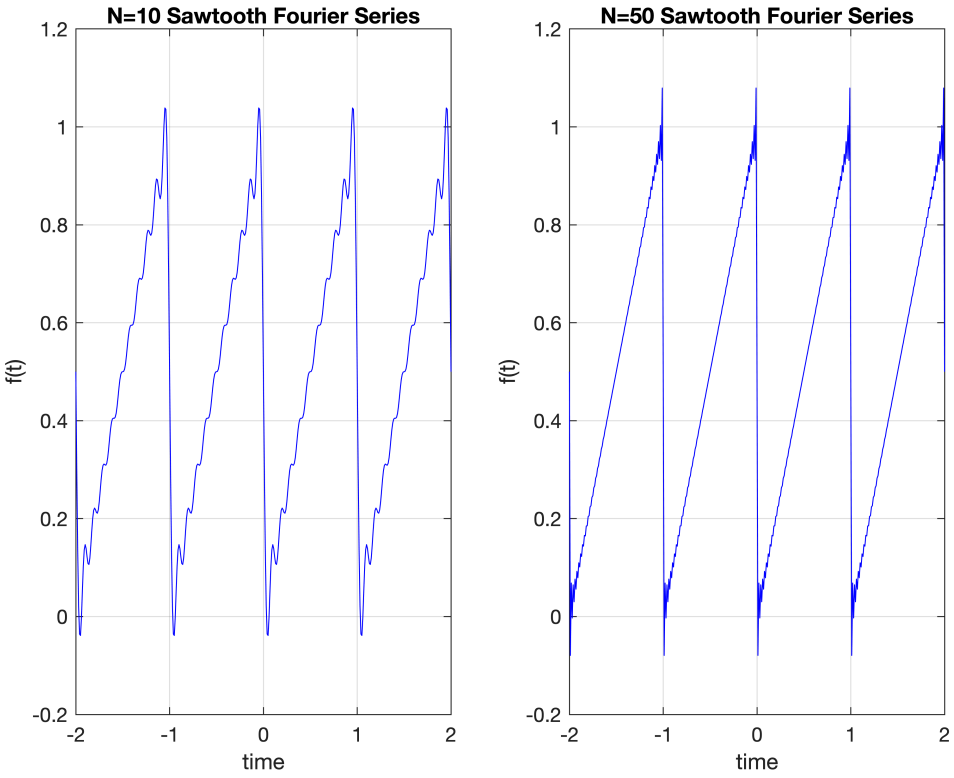
```
t = -2:0.01:2;
dn1 = -10:1:10;
dn1 = j./(dn1*2*pi);
dn1((length(dn1)+1)/2) = 0.5; %give C0 the appropriate value of 1/2
omega0 = 2*pi; %since we're doing Sawtooth for T=1
y1 = myfs(dn1, omega0,t);
dn2 = -50:1:50;
dn2 = j./(dn2*2*pi);
dn2((length(dn2)+1)/2) = 0.5; %give C0 the appropriate value of 1/2
y2 = myfs(dn2, omega0,t);
subplot(1,2,1);
plot(t, y1,'b'); grid on;
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

```
title('N=10 Sawtooth Fourier Series'); xlabel('time');ylabel('f(t)');
subplot(1,2,2);
plot(t, y2,'b'); grid on;
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```

```
title('N=50 Sawtooth Fourier Series'); xlabel('time');ylabel('f(t)');
```



5(a) Task 1:

```
function fn = myfs(Dn,omega0,t)
DnL = length(Dn); %obtain the length of Dn, which is 2N+1
N = (DnL-1)/2; %obtain N
fn = zeros(1,length(t)); %create a vector with length of desired time points
%stores value at each time point from each element of fourier series to fn
for n=1:DnL
    fn = fn + Dn(n)*exp(j*(n-1-N)*omega0*t); %the nth element in Dn corresponds to k va
end
end
```