Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Signals and Systems II
Course Number:	ELE632
Semester/Year (e.g.F2016)	W2022

Instructor:	Dimitri Androutsos

Assignment/Lab Number:	1
Assignment/Lab Title:	Time-Domain Analysis of Discrete-Time Systems - Part 1

Submission Date:	Tuesday, February 8th, 2022
Due Date:	Tuesday, February 8th, 2022

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Fahmy	Ahmad	500913092	9	A.F.

**By signing above, you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*

https://www.ryerson.ca/senate/policies/pol60.pdf

Appendix

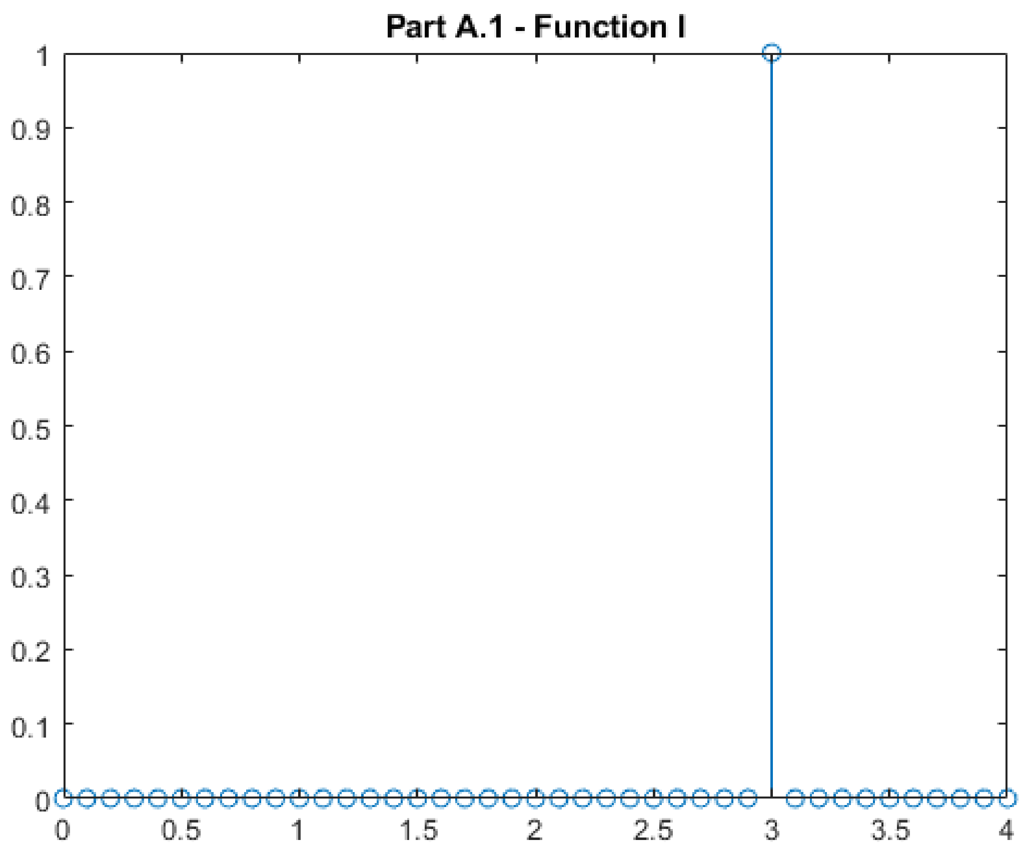
A) Signal Transformation 3
Part 1 - Plotting Discrete Time Signals 3
Function I 3
Function II 3
Function III, IV, & V 4
Part 2 - Scaling Discrete Time Signals 7
Part 3 - Sampling Continuous Signals 10
B) Recursive Solution of Difference Equation 11
Part I - Compound Interest of Balance with Monthly Contributions 11
Part II - Compound Interest of Balance, Zero Input Response 12
Part III - n Growth Deposits 14
C) Design a filter: casual N-point maximum filter 15
Part I - maximum filtering 15
Part II - maximum filter with x[n] 16
D) Energy and power of a discrete signal 19
Part I - energy and power of a vector 19
Part II - energy and power of x[n] (Fig.P3.1-1 (c) of the textbook) 19

A) Signal Transformation

Part 1 - Plotting Discrete Time Signals

Function I

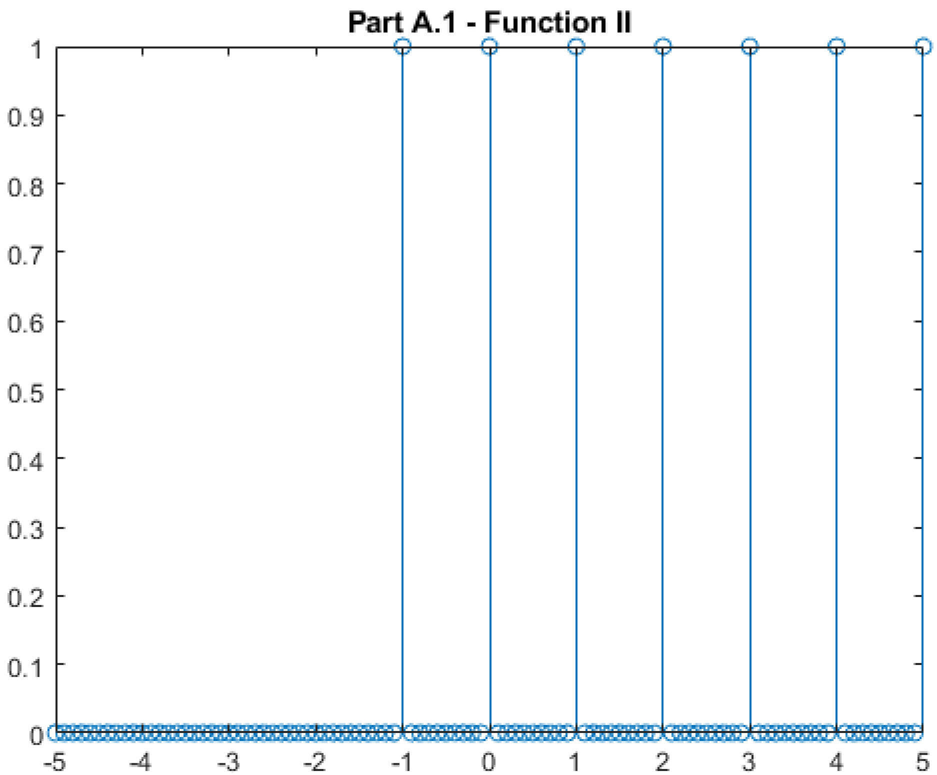
```
x = 0:0.1:4;
y = dirac(x-3);
idx = y == Inf;
y(idx) = 1;
figure;
stem(x,y)
title('Part A.1 - Function I');
```



Function II

```
n = -5:0.1:5;
u = (mod(n,1)==0)*1.0.*(n>=-1);
```

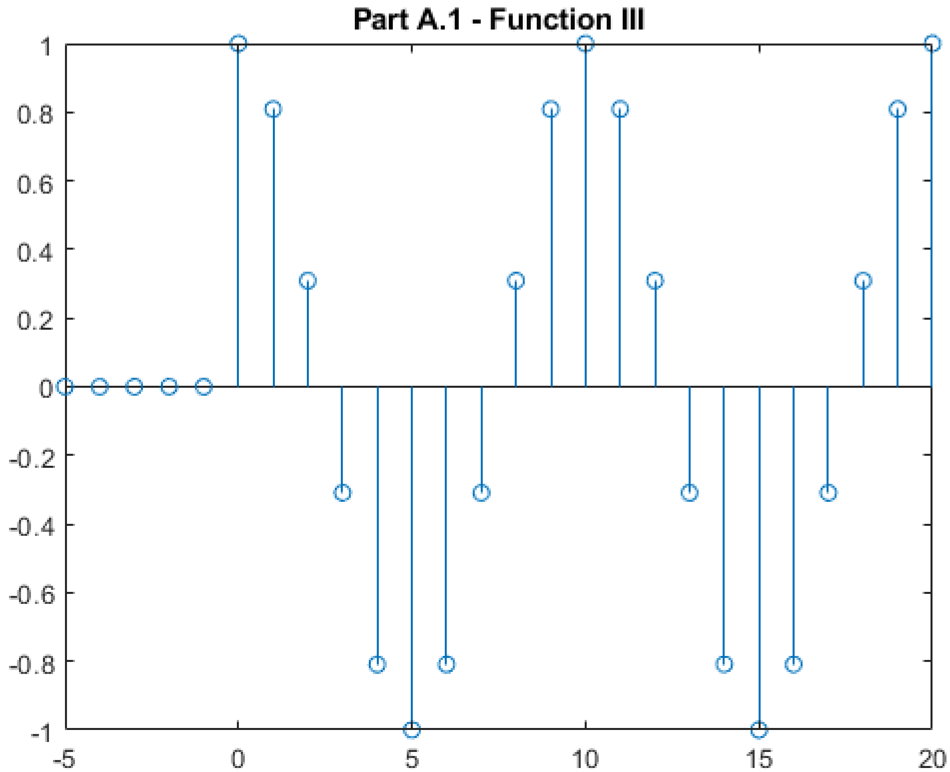
```
figure;
stem(n,u);
title('Part A.1 - Function II');
```

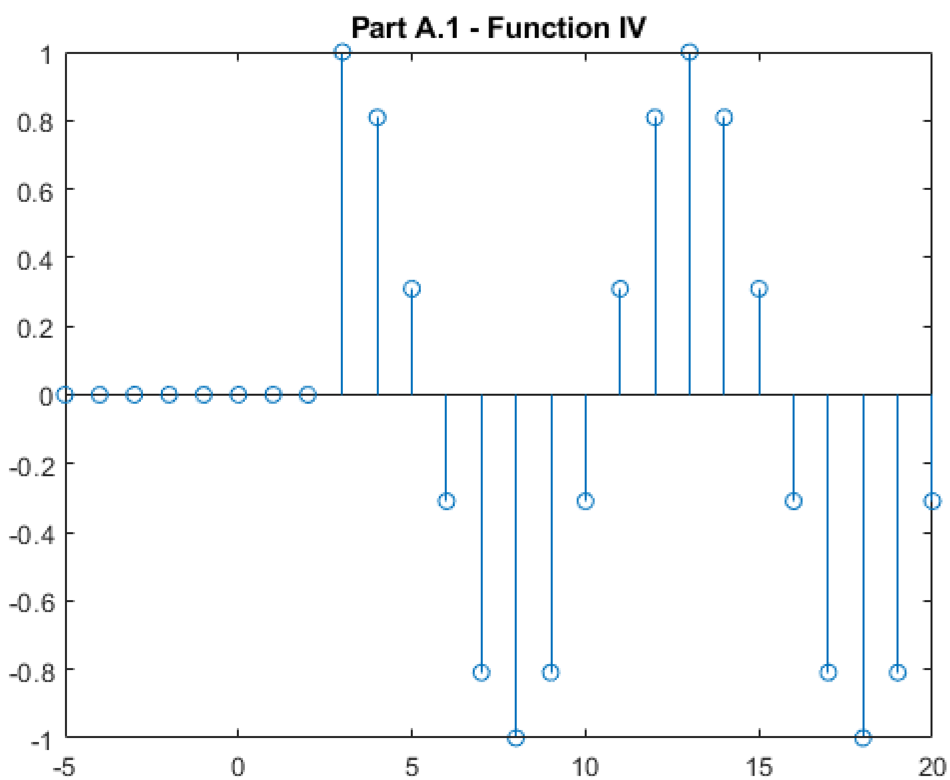


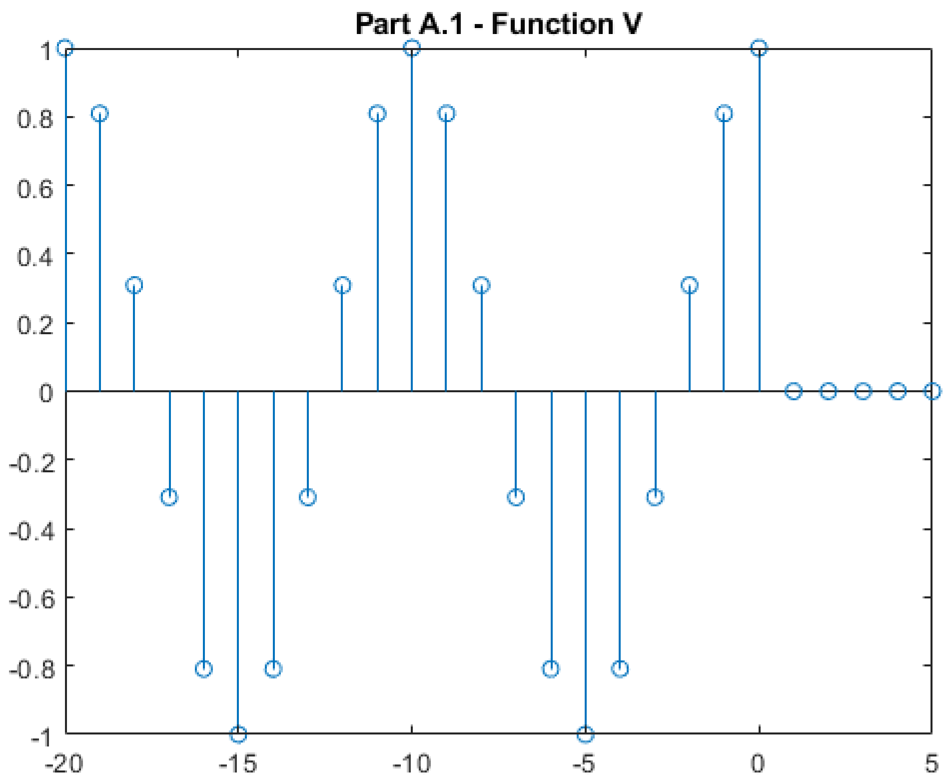
Function III

```
x = -5:20;
u = (mod(x,1)==0)*1.0.*(x>=0);
y = cos(pi.*x./5).*(u);
figure;
stem(x,y)
title('Part A.1 - Function III')



% % * Function IV
u = (mod(x,1)==0)*1.0.*(x>=3);
```

```
y = cos(pi.*(x-3)./5).*(u);
figure;
stem(x,y)
title('Part A.1 - Function IV')



% % * Function V
x = -20:5;
u = (mod(x,1)==0)*1.0.*(x<=0);
y = cos(pi.*(-x)./5).*(u);
figure;
stem(x,y)
title('Part A.1 - Function V')

%
% %The transformation performed in Function IV was a rightward-shift of 3
% %units along the x-axis on Function III.
% %The transformation performed in Function V was a mirrored flip along the
% %y-axis on Function III.
```


Part A.1 - Function III
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-5
0
5
10
15
20

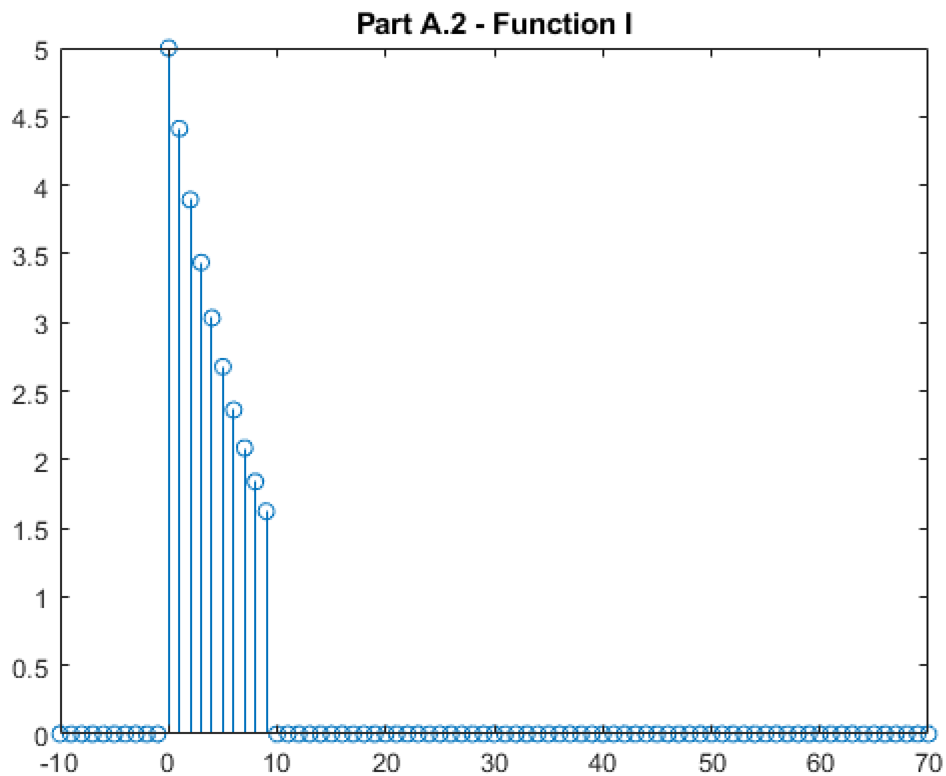

Part A.1 - Function IV
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-5
0
5
10
15
20

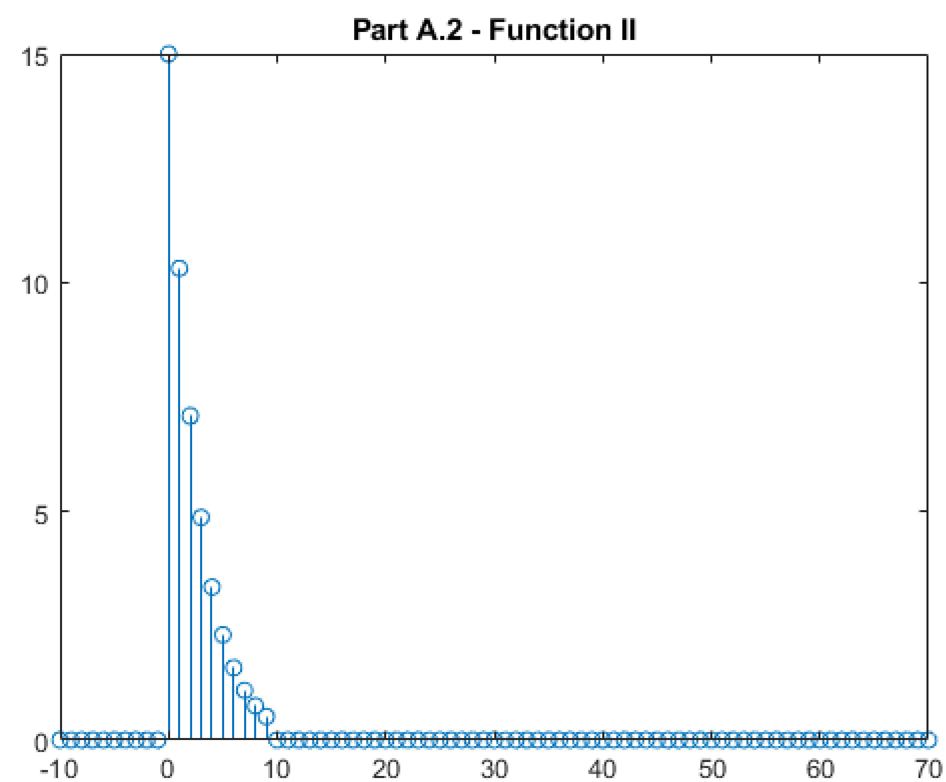


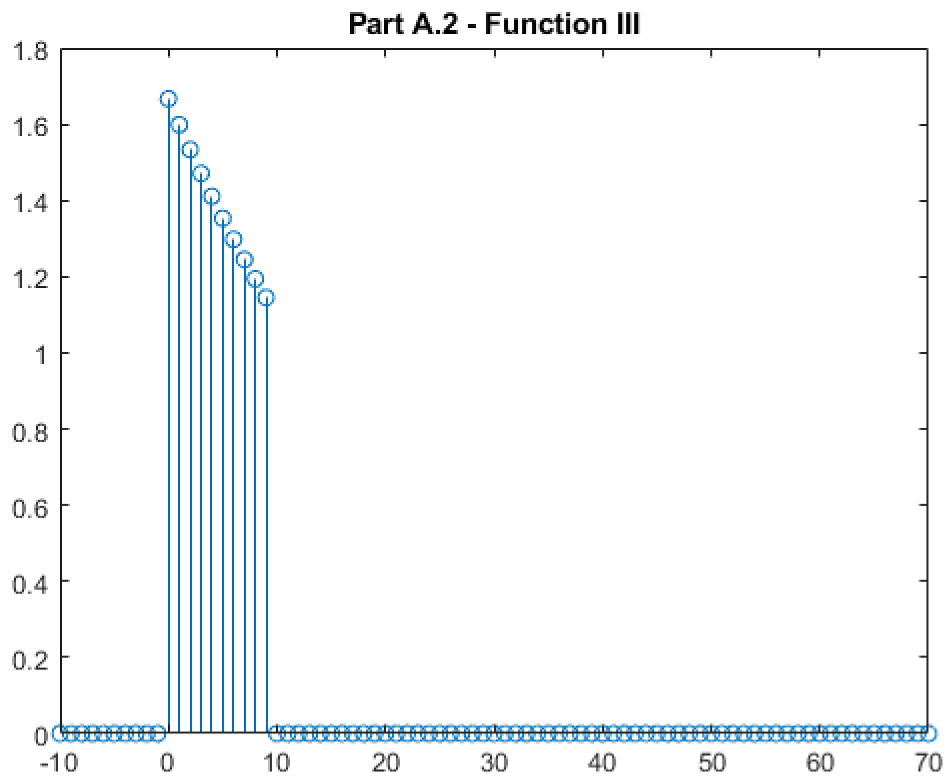
Part 2 - Scaling Discrete Time Signals

```
x = -10:70;
u = (mod(x,1)==0)*1.0.*(x>=0);
u2 = (mod(x,1)==0)*1.0.*(x>=10);
d=(mod(x,1)==0)*1.0;
y = 5.*exp(-x/8).*(u-u2);
figure;
stem(x,y);
title('Part A.2 - Function I')

% * Section II
u = (mod(x,1)==0)*3.0.*(x>=0);
u2 = (mod(x,1)==0)*3.0.*(x>=10);
y = 5.*exp(-3.*x/8).*(u-u2);
figure;
```

```
stem(x,y);
title('Part A.2 - Function II')

% * Section III
u = (mod(x,1)==0)*(1.0./3).*(x>=0);
u2 = (mod(x,1)==0)*(1.0./3).*(x>=10);
y = 5.*exp(-x/(8.*3)).*(u-u2);

figure;
stem(x,y);
title('Part A.2 - Function III')

%The transformation performed in Function II was a horizontal compression on Function I,
%however, since the exponential function is being multiplied by the difference of two
%unit step functions the result was a vertical stretch by 3 from x = [0:10]
%then back to 0 since the two unit step functions result in a
%multiplication by 0.
%The transformation performed in Function III was a horizontal stretch on
%Function I, however, since the exponential function is being multiplied by the difference of
two
%unit step functions the result was a vertical compression by 3 from x = [0:10]
%then back to 0 since the two unit step functions result in a
%multiplication by 0.
```


Part A.2 - Function I
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-10
0
10
20
30
40
50
60
70

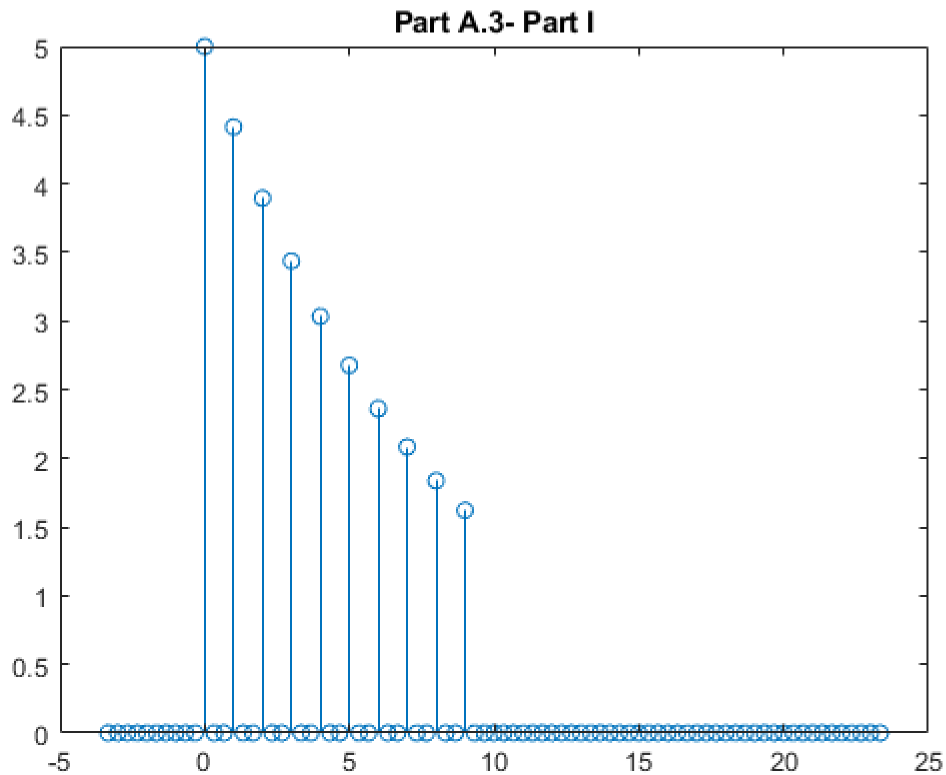

Part A.2 - Function II
15
10
5
0
-10
0
10
20
30
40
50
60
70



Part 3 - Sampling Continuous Signals

```
x = -10:70;
x=x./3;
u = (1.0).*(x>=0);
u2 = (1.0).*(x>=10);
z =  5.*exp(-x/(8)).*(u-u2);
d=(mod(x,1)==0)*1.0;
z = z.*d;
figure;
stem(x,z);
title('Part A.3- Part I')

% * Section II
% The reason y2[n] from PII and y3 from PIII are not the same is because
% the signal in PII was sampled first, then had a transformation applied
```

```
% Whereas, PIII was transformed then sampled which resulted in the
% possibility of new discrete integer values being added after the transform
% and old discrete integer values that are no longer integers being removed
```

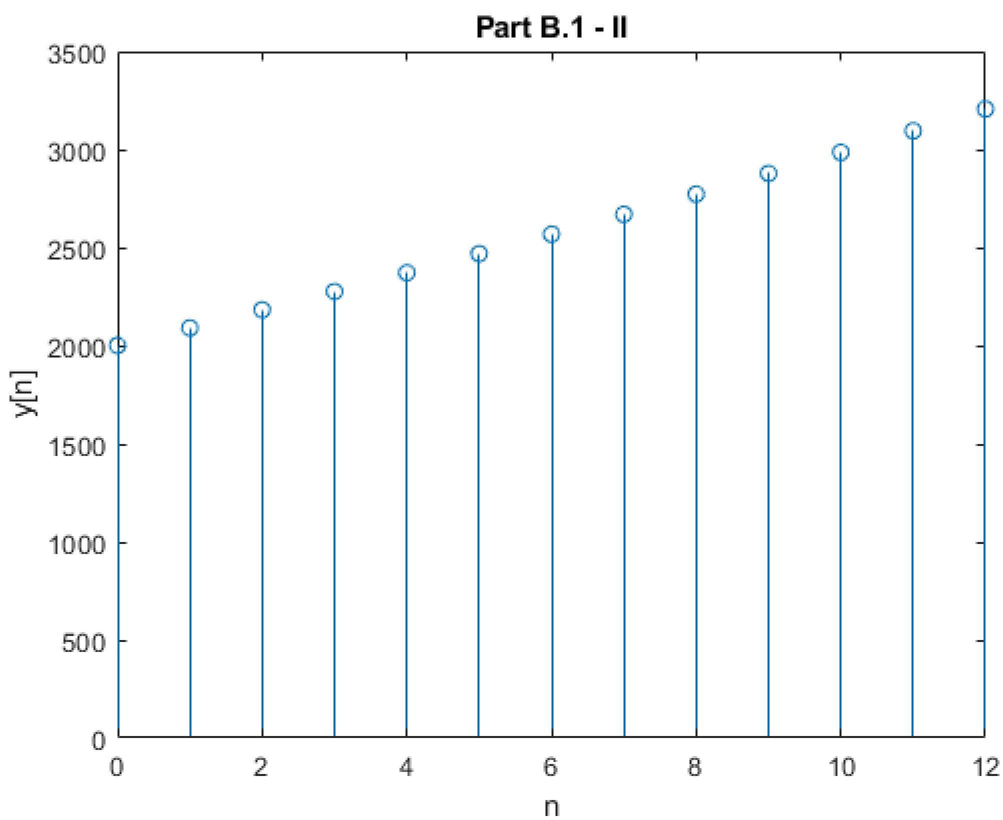


B) Recursive Solution of Difference Equation

Part I - Compound Interest of Balance with Monthly Contributions

```
r = 0.02;
n=0:12;
y = [2000 0 0 0 0 0 0 0 0 0 0 0 0];
x = [0 50 50 50 50 50 50 50 50 50 50 50 50];

for k =2:length(n)
    y(k)=(1+r).*y(k-1)+x(k);
end
```

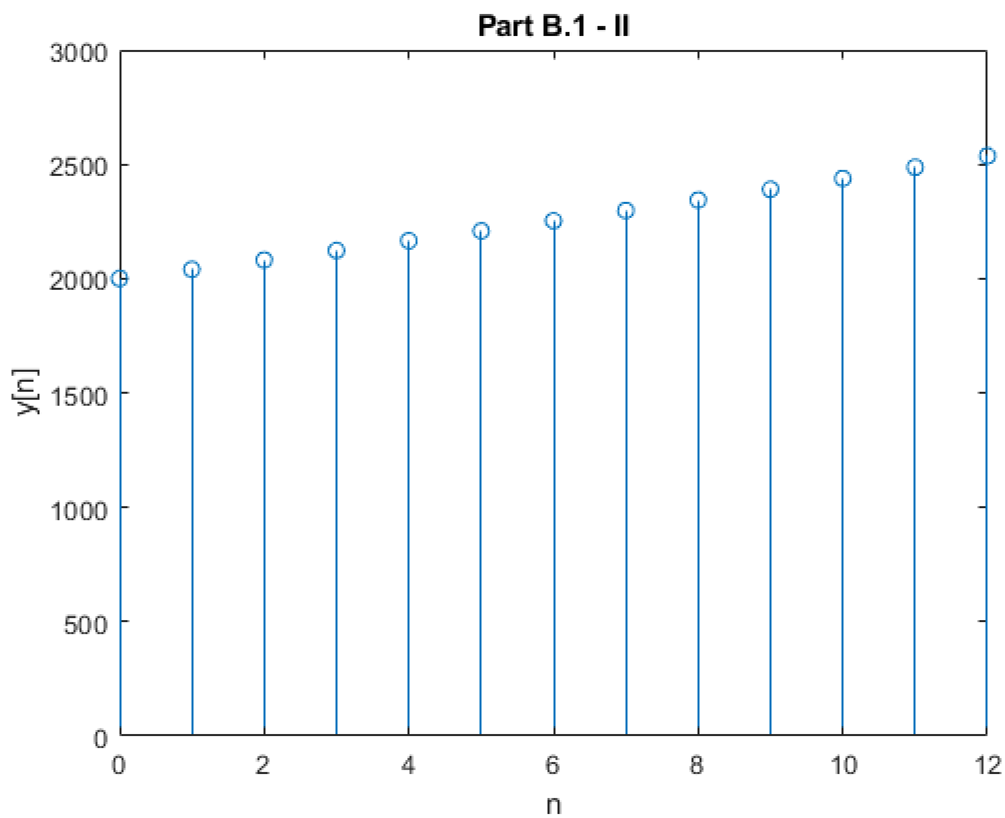
```
figure;
stem(n, y);
title('Part B.1 - II')
xlabel('n')
ylabel('y[n]')
```



Part II - Compound Interest of Balance, Zero Input Response

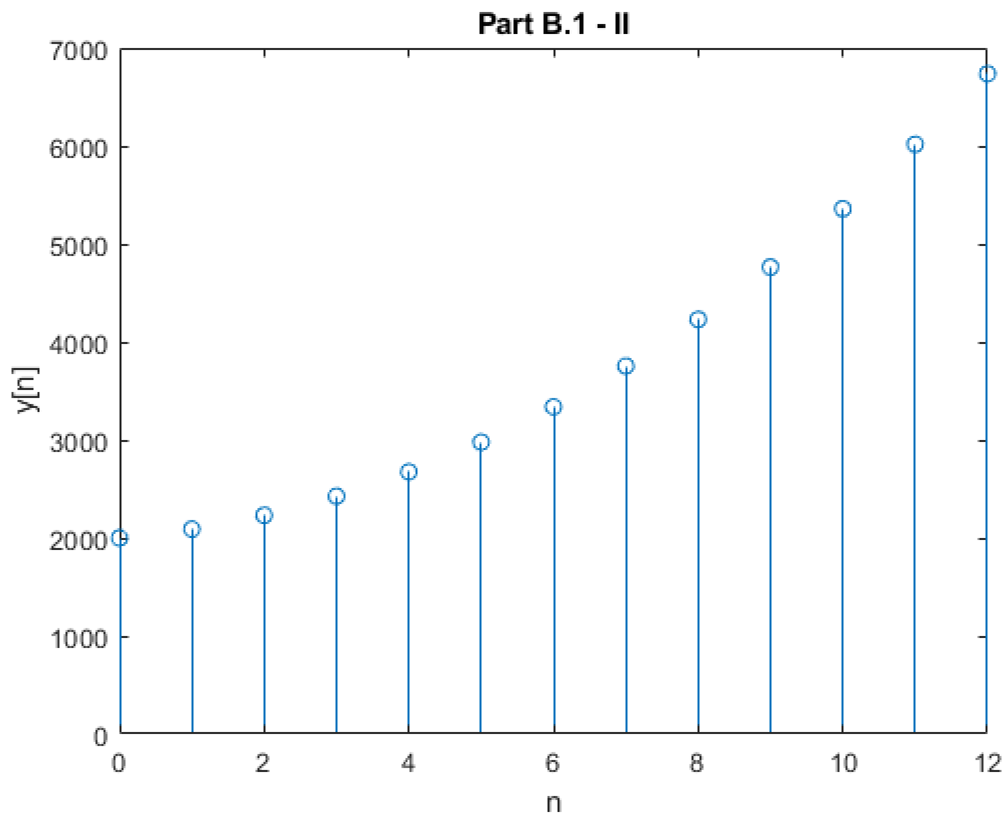
```
r = 0.02;
n=0:12;
y = [2000 0 0 0 0 0 0 0 0 0 0 0 0];
% x = [0 50 50 50 50 50 50 50 50 50 50 50 50];
x = [0 0 0 0 0 0 0 0 0 0 0 0 0];

for k =2:length(n)
    y(k)=(1+r).*y(k-1)+x(k);
```

```
end


%zero input response CE is:
% y0[n] = c1[1/a]^n
% y0[n] = (0)[1/a]^n
% y0[n] = 0



figure;
stem(n, y);
title('Part B.1 - II')
xlabel('n')
ylabel('y[n]')
```



Part III - n Growth Deposits

```
r = 0.02;
n=0:12;
y = [2000 0 0 0 0 0 0 0 0 0 0 0 0];
x = [0 50 100 150 200 250 300 350 400 450 500 550 600];

for k =2:length(n)
    y(k)=(1+r).*y(k-1)+x(k);
end

figure;
stem(n, y);
title('Part B.1 - II')
xlabel('n')
ylabel('y[n]')

clc
clear
```

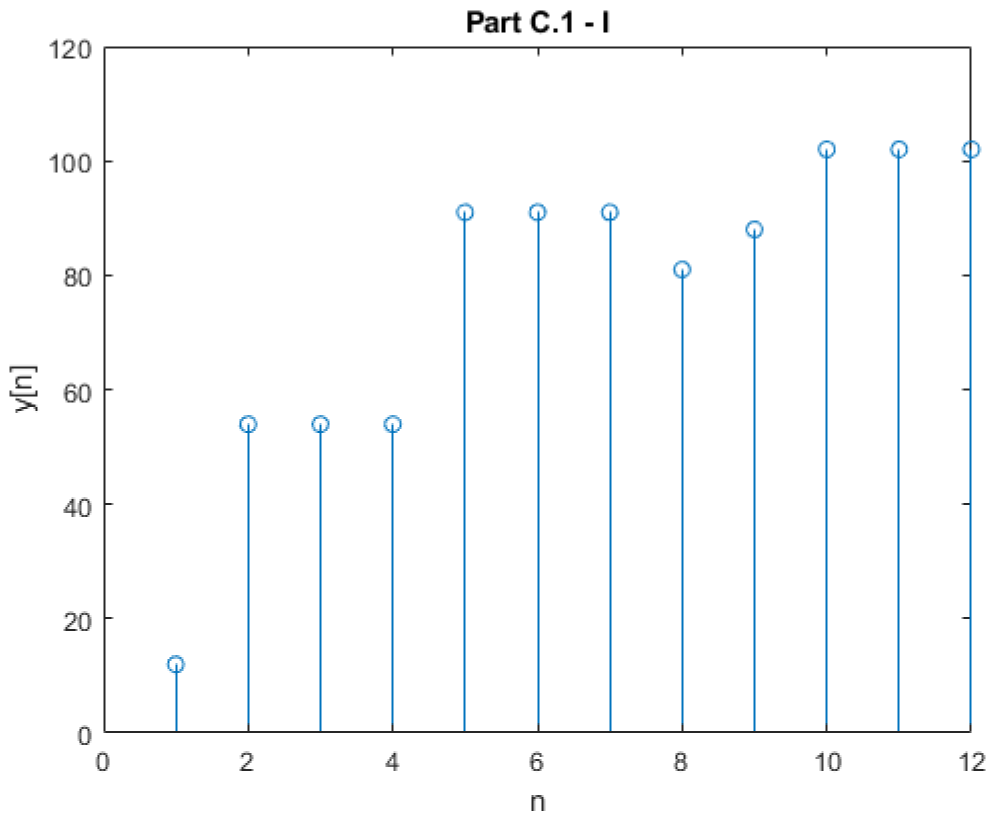


C) Design a filter: casual N-point maximum filter

Part I - maximum filtering

```
filterN = 3;
newval = [12 54 23 10 91 81 30 22 88 102 67 23];
M = length(newval);
x = zeros(1,filterN-1);
for i = 1: M
    x(filterN-1+i) = newval(i);
end


for k=1:M
        temp = x(:, k:k+filterN-1);
    y(k) = max(temp);
end
```

```
n = 1:M;
figure;
stem(n, y);
title('Part C.1 - I')
xlabel('n')
ylabel('y[n]')

clc
clear
```

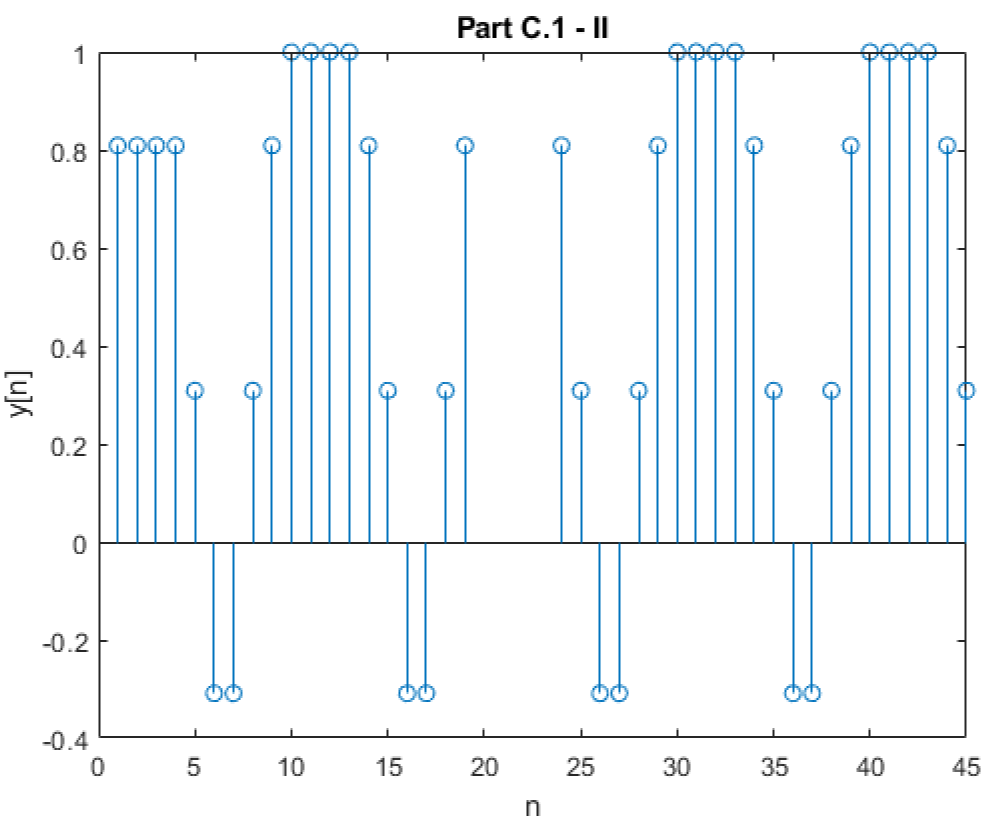


Part II and III - maximum filter with x[n]

```
filterN = 4; %N = 4, 8, 12
M = 45;
n = 1:M;
newval = cos(pi.*n./5)+dirac(n-20)-dirac(n-35);
```

```
x = zeros(1,filterN-1);
for i = 1: M
   x(filterN-1+i) = newval(i);
end

for k=1:M
      temp = x(:, k:k+filterN-1);
   y(k) = max(temp);
end



figure;
stem(n, y);
title('Part C.1 - II')
xlabel('n')
ylabel('y[n]')



%Part III:
%For N = 4, the response of the function begins with 4 units of it's highest
%first x-discrete y-value, the function allows for some negative values
%since the dips are 5 units wide, the high values are held for 4
%units since the next trajectory is negative upholding the high value till
% its no longer in the window, the dirac(n-20) function creates a gap
% starting at x=20 until it's no longer in the window range, and the
% -dirac(n-35) function is immediatly outmaxed since it occurs halfway
% through the dip and there is a higher value within 4 units.

%When N = 8, the initial reponse of the first x-discrete y-value is held
```

```
%for 8 units, the high values are held for the same reason as N=4, the
%negative values dissapear since the window range is larger than the dip
%width allowing there to always be a higher value at the begining of a dip,
%the dirac(n-20) function creates a gap starting at x=20 and creates an
% undefined zone for 8 units, and the -dirac(n-35) function is immediatly
% outmaxed for the same reason as N=4.

%Lastly when N = 12, the initial reponse of the first x-discrete y-value is held
%only for 8 units since the next peak occurs within the window range, the
%initial peak values are only held for 8 units since the dirac(n-20) occurs
%at x=20 and creates an undefined zone for 12 units, beyond the initial
%peak values, the peaks are held until the length of the input vector, and
% the -dirac(n-35) function is immediatly outmaxed for the same reason as N=4.
```



D) Energy and power of a discrete signal

Part I - energy and power of a vector

```
n = 0:20000;
x = (mod(n,1)==0)*1.0.*(n>=10000);
L = length(n);
N = L;
E = (sum(abs(x).^2));
P = (norm(x).^2)./N;

% Since Power is finite, this unit-step function is a power signal, and
% Energy is infinite
```

Part II - energy and power of x[n] (Fig.P3.1-1 (c) of the textbook)

```
x = [0 -9 -6 -3 0 3 6 9 0 0];
L = length(x);
N = L;
E = (sum(abs(x).^2));
P = (norm(x).^2)./N;

% Since Power is finite, this unit-step function is a power signal, and
% Energy is infinite
```