ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Physics

Α.Μ: *201100079*	**Επώνυμο:** *Κορομηλάς*	**Όνομα:** *Χρήστος*
Ημ/νία(Ημέρα,Ώρα) εκτέλεσης της άσκησης: ***13/04/2022 (Τετάρτη, 15:30-18:00)***		
Ημ/νία παράδοσης της άσκησης: 21/04/2022		

ΣΥΝΕΛΙΞΗ, ΑΥΤΟΣΥΣΧΕΤΙΣΗ, και ΕΤΕΡΟΣΥΣΧΕΤΙΣΗ ΣΗΜΑΤΩΝ

Μια εργασία η οποία επικεντρώνεται στην υλοποίηση προγραμμάτων στη MATLAB, και πιο συγκεκριμένα στην ανάλυση συνέλιξης, αυτοσυσχέτισης και ετεροσυσχέτισης σημάτων.

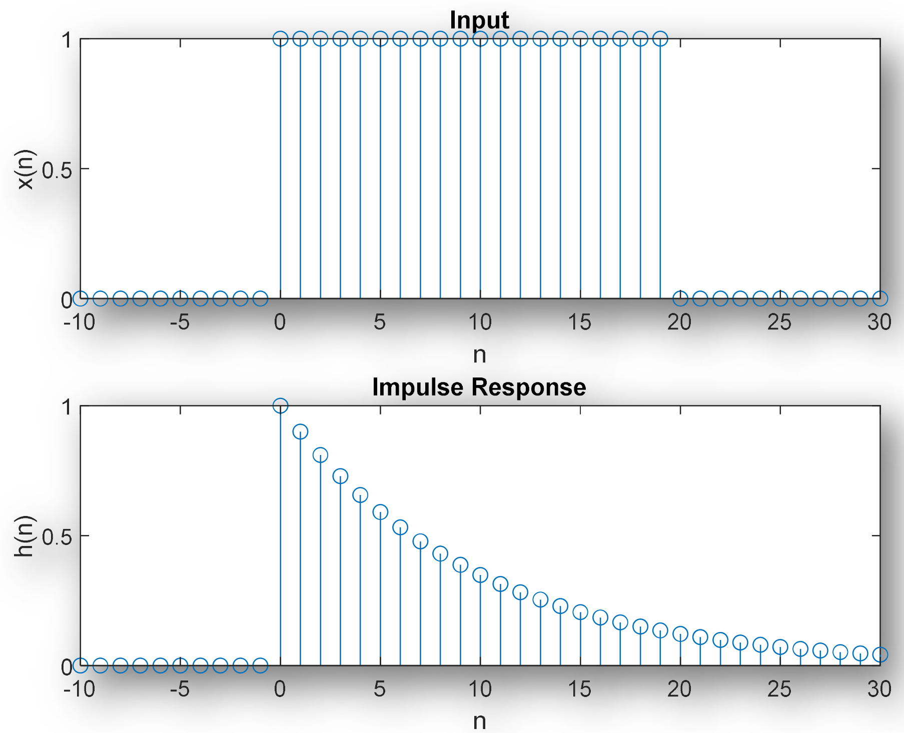
➢ ***Άσκηση 1η :***

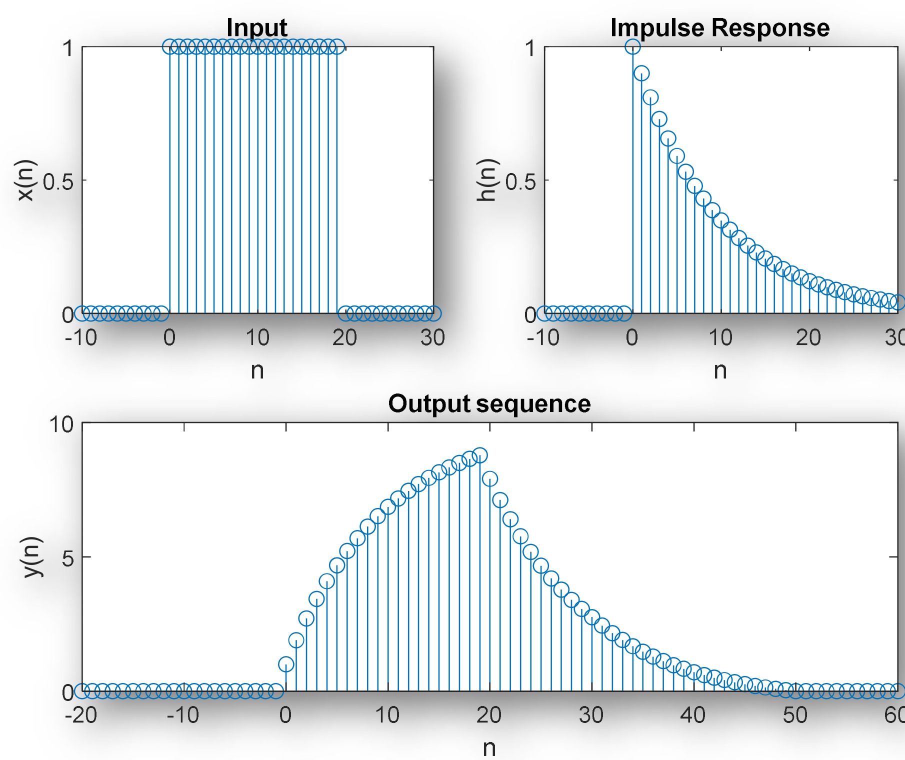
Ακολουθούν τα προγράμματα $\boldsymbol{init1.m}$, $\boldsymbol{convol1.m}$ στη λειτουργική τους μορφή και τα αποτελέσματα εκτέλεσης, μαζί και η συνάρτηση $\boldsymbol{step\_f.m}$:

```
%%init1.m

clear all;
close all;
clc;

n1 = -10;
n2 = 30;


[u1,n] = step_f(0, n1, n2);
[u2,n] = step_f(20, n1, n2);

xn = u1-u2;
hn = (0.9.^n).*u1;

figure(1);

subplot(211);
stem(n,xn);
title('Input');
ylabel('x(n)');
xlabel('n');

subplot(212);
stem(n,hn);
title('Impulse Response');
ylabel('h(n)');
xlabel('n');

print -depsc progr13;
```

```
%%step function

function[u,n]=step_f(n0,n1,n2);
n = [n1:1:n2];
u = [(n-n0)>=0];
```

```
%%First Convolution exercise

clear all;
close all;
clc;

n1 = -10;
n2 = 30;

[u1,n1]=step_f(0,n1,n2);
[u2,n2]=step_f(20,n1,n2);

xn = u1-u2;
hn = (0.9.^n1).*u1;
yn = conv(xn,hn);

n1_new = n1(1)+n2(1);
n2_new = n1(length(u1)) + n2(length(u2));
n_new = [n1_new:1:n2_new];

figure(1);

subplot(221);
stem(n1,xn);
title('Input');
ylabel('x(n)');
xlabel('n');

subplot(222);
stem(n1,hn);
title('Impulse Response');
ylabel('h(n)');
xlabel('n');

subplot(212);
stem(n_new,yn);
title('Output sequence');
ylabel('y(n)');
xlabel('n');

print -depsc progr14;
```


Input
x(n)
n
Impulse Response
h(n)
n


Input
x(n)
n
Impulse Response
h(n)
n
Output sequence
y(n)
n

Απάντηση ερωτήσεων:

Ερώτημα $\mathbf{1}$*:*

Για το πρόγραμμα της βηματικής συνάρτησης, $\boldsymbol{step\_f.m}$*: Το πρόγραμμα για να λειτουργεί χρειάζεται 3 μεταβλητές, τη μεταβλητή σύγκρισης, τη μεταβλητή πρώτης τιμής και τη μεταβλητή τελευταίας τιμής. Το πρώτο βήμα είναι να δημιουργηθεί ένας πίνακας που θα έχει ίσο αριθμό ψηφίων με το πίνακα* $\boldsymbol{n}$*, ο πίνακας* $\boldsymbol{u}$ *ο όποιος παίρνει τιμή* $\mathbf{1}$ *στις θέσεις όπου υπάρχει* $\boldsymbol{n} >=$ *από τη μεταβλητή σύγκρισης και* $\mathbf{0}$ *σε όλες τις άλλες. Καθώς τερματίζει η συνάρτηση επιστρέφονται στην έξοδο ο* $\boldsymbol{u}$ *και ο* $\boldsymbol{n}$*.*

Για το πρόγραμμα $\boldsymbol{init1.m}$*: Σκοπός του είναι να υπολογιστούν και να σχεδιαστούν οι ακολουθίες* $\boldsymbol{x(n)}$ *και* $\boldsymbol{h(n)}$ *. Καθώς το πρόγραμμα εκκινεί αρχικοποιεί τις μεταβλητές* $\boldsymbol{n1}, \boldsymbol{n2}$ *, έπειτα δημιουργούνται με τη βοήθεια της* $\boldsymbol{step\_f}$ *οι βηματικοί συντελεστές* $\boldsymbol{u1}$ *και* $\boldsymbol{u2}$ *, και αντίστοιχα το* $\boldsymbol{hn}$ *με την εντολή* $\boldsymbol{hn = (0.9.\hat{}\,n).*\,u1;}$ *. Τέλος το πρόγραμμα μας τυπώνει και αποθηκεύει τα δεδομένα μας όπως είδαμε και στις προηγούμενες εργασίες.*

Για το $\boldsymbol{convol1.m}$*: Σκοπός του είναι να υπολογιστούν και να σχεδιαστούν οι ακολουθίες* $\boldsymbol{x(n)}$ *και* $\boldsymbol{h(n)}$ *καθώς και η συνέλιξη τους* $\boldsymbol{y(n)}$ *(όπου* $\boldsymbol{x(n)}$ *η είσοδος μας και* $\boldsymbol{y(n)}$ *η έξοδος). Καθώς το πρόγραμμα εκκινεί αρχικοποιεί τις μεταβλητές* $\boldsymbol{u1}$ *με* $\boldsymbol{n1}$ *και* $\boldsymbol{u2}$ *με* $\boldsymbol{n2}$ *με τους οποίους μπορούμε να υπολογίσουμε το* $\boldsymbol{xn}$ *μέσω της εντολής* $\boldsymbol{xn = u2 - u1}$*, και αντίστοιχα το* $\boldsymbol{hn}$ *με την εντολή* $\boldsymbol{hn = (0.9.\hat{}\,n1).*\,u1}$*. Στη συνέχεια, υπολογίζεται η συνέλιξη αυτών των δύο σημάτων μέσω της εντολής* $\boldsymbol{yn = conv(xn, hn)}$*; και καταχωρούνται οι τιμές στο πίνακα* $\boldsymbol{yn}$*. Έπειτα τρέχουν οι 3 γραμμές που περιγράφουμε στο ερώτημα 5 και πριν το πρόγραμμα τερματιστεί, εκτυπώνει και αποθηκεύει τα δεδομένα μας όπως είδαμε και στις προηγούμενες εργασίες.*

Ερώτημα $\mathbf{2}$*:*

Η γραμμή 2 μέσα στο $\boldsymbol{step\_f.m}$ *καταχωρεί τιμές στο μονοδιάστατο πίνακα από τη τιμή* $\boldsymbol{n1}$ *έως τη* $\boldsymbol{n2}$ *με βήμα* $\mathbf{1}$*, ενώ η γραμμή 3 επιστρέφει στο μονοδιάστατο πίνακα τη τιμή* $\mathbf{1}$ *για όσα* $\boldsymbol{n}$ *είναι μεγαλύτερα η ίσα του* $\boldsymbol{n0}$ *ενώ μηδέν για όσα δεν είναι.*

Ερώτημα $\mathbf{3}$*:*

Στο πρόγραμμα $\boldsymbol{1\beta}$ *η γραμμή 8 εκτελεί τη συνάρτηση* $\boldsymbol{step\_f}$ *για τιμές* $\boldsymbol{(0, n1, n2)}$ *και επιστρέφει την έξοδο στους πίνακες* $\boldsymbol{u1}$ *και* $\boldsymbol{n}$*. Αντίστοιχα η γραμμή 9 εκτελεί τη συνάρτηση* $\boldsymbol{step\_f}$ *για τιμές* $\boldsymbol{(20, n1, n2)}$ *και επιστρέφει την έξοδο στους πίνακες* $\boldsymbol{u2}$ *και* $\boldsymbol{n}$*.*

Στο πρόγραμμα $\boldsymbol{1\gamma}$ *η γραμμή 6 εκτελεί τη συνάρτηση* $\boldsymbol{step\_f}$ *για τιμές* $\boldsymbol{(0, n1, n2)}$ *και επιστρέφει την έξοδο στους πίνακες* $\boldsymbol{u1}$ *και* $\boldsymbol{n1}$*. Αντίστοιχα η γραμμή 7 εκτελεί τη συνάρτηση* $\boldsymbol{step\_f}$ *για τιμές* $\boldsymbol{(20, n1, n2)}$ *και επιστρέφει την έξοδο στους πίνακες* $\boldsymbol{u2}$ *και* $\boldsymbol{n2}$*.*

Ερώτημα $\mathbf{4}$*:*

Στο διάγραμμα $\boldsymbol{1\gamma}$ *στη γραμμή 12 υπολογίζεται η συνέλιξη του* $\boldsymbol{xn}$ *και του* $\boldsymbol{hn}$ *και επιστρέφεται η έξοδος στο πίνακα* $\boldsymbol{yn}$*.*

***Ερώτημα* $\mathbf{5}$*:**

Στο πρόγραμμα $\boldsymbol{1\gamma}$ η γραμμή 13 παίρνει τις τιμές των πρώτων στοιχείων των πινάκων $\boldsymbol{n1}$ και $\boldsymbol{n2}$ τις προσθέτει και επιστρέφεται η έξοδος στο στοιχείο $\boldsymbol{n1\_new}$. Στη γραμμή 14 το πρόγραμμα παίρνει τις τιμές των πρώτων στοιχείων των πινάκων $\boldsymbol{n1}$ και $\boldsymbol{n2}$, τις προσθέτει και επιστρέφεται η έξοδος στο στοιχείο $\boldsymbol{n2\_new}$. Στη γραμμή 16 δημιουργείται ένας νέος πίνακας $\boldsymbol{n\_new}$ με στοιχεία από το $\boldsymbol{n1\_new}$ έως το $\boldsymbol{n2\_new}$ με βήμα $\mathbf{1}$ και αυτός ο νέος πίνακας θα αποτελέσει το εύρος στο οποίο θα απεικονιστεί η $\boldsymbol{y(n)}$.

***Ερώτημα* $\mathbf{6}$ & $\mathbf{7}$*:**

Η $\boldsymbol{x(n) = u(n) - u(n-20)}$ είναι μία διακριτή συνάρτηση με σταθερό πλάτος που παίρνει τιμές είτε $\mathbf{1}$ είτε $\mathbf{0}$. Συγκεκριμένα η $\boldsymbol{x(n)}$ έχει τη τιμή $\mathbf{1}$ στο διάστημα $\mathbf{[0, 20]}$ ενώ σε όλο τον υπόλοιπο χώρο είναι μηδενική.

Η $\boldsymbol{h(n) = 0.9^n * u(n)}$ είναι μία διακριτή συνάρτηση η οποία παίρνει τιμές από το $\mathbf{0}$ έως το $\mathbf{1}$ και για $\boldsymbol{n >= 0}$ είναι μία φθίνουσα συνάρτηση με αρχή το $\boldsymbol{h(0) = 1}$.

Η $\boldsymbol{y(n)}$ είναι μία πιο πολύπλοκη συνάρτηση . Για τιμές του $\boldsymbol{n < 0}$ η $\boldsymbol{y(n) = 0}$, για τις τιμές στο διάστημα $\mathbf{[0, 20]}$ η $\boldsymbol{y(n)}$ παίρνει τη μορφή:

$$\boldsymbol{y(n) = 10(1 - 0.9^{n+1})}$$

Όπου η $\boldsymbol{y(n)}$ είναι αύξουσα και μπορούμε να παρατηρήσουμε τη μερική κάλυψη των $\boldsymbol{h(n), x(n)}$.

Για $\boldsymbol{n > 20}$ η $\boldsymbol{y(n)}$ παίρνει τη μορφή:

$$\boldsymbol{y(n) = 10 * 0.9^{n-19}(1 - 0.9^{20})}$$

Όπου η $\boldsymbol{y(n)}$ είναι φθίνουσα και μπορούμε να παρατηρήσουμε τη πλήρη επικάλυψη των $\boldsymbol{h(n)}$ και $\boldsymbol{x(n)}$.

➢ ***Άσκηση 2η :***

Για την αναλυτική λύση γνωρίζουμε πως το σήμα εισόδου μας είναι:

$$\boldsymbol{x(n) = u(n) - u(n-30)}$$

Ενώ για τη κρουστική απόκριση έχουμε:

$$\boldsymbol{h(n) = 0.9 * \delta(n-5)}$$

Πραγματοποιώντας τη συνέλιξη των δύο σημάτων παίρνουμε:

$$\boldsymbol{y(n) = x(n) * h(n)}$$

$$\boldsymbol{y(n) = \sum_{k=-\infty}^{\infty} \big(u(k) - u(k-30)\big) * 0.9^{n-k} * \delta(n-k-5)}$$

<u>Προκύπτουν δύο περιπτώσεις:</u>

Για τη περίπτωση του $\boldsymbol{n < 5}$ ή $\boldsymbol{n > 35}$: $\boldsymbol{\delta(n-k-5) = 0}$ $\forall \boldsymbol{k} \in [\boldsymbol{0}, \boldsymbol{30}]$
Καθώς ισχύει $\boldsymbol{k \neq n-5}$, και συνεπώς $\boldsymbol{y(n) = 0}$

Για τη περίπτωση του $\boldsymbol{5 < n < 35}$: $\forall \boldsymbol{k} \in [\boldsymbol{0}, \boldsymbol{30}]$ ισχύει πως $\boldsymbol{k = n} -5$
και συνεπώς το $\boldsymbol{y(n)}$ έχει μη μηδενικές τιμές και πιο συγκεκριμένα:

$$\boldsymbol{y(n) = \sum_{k=0}^{30} \big(0.9^{n-k}\big)}$$

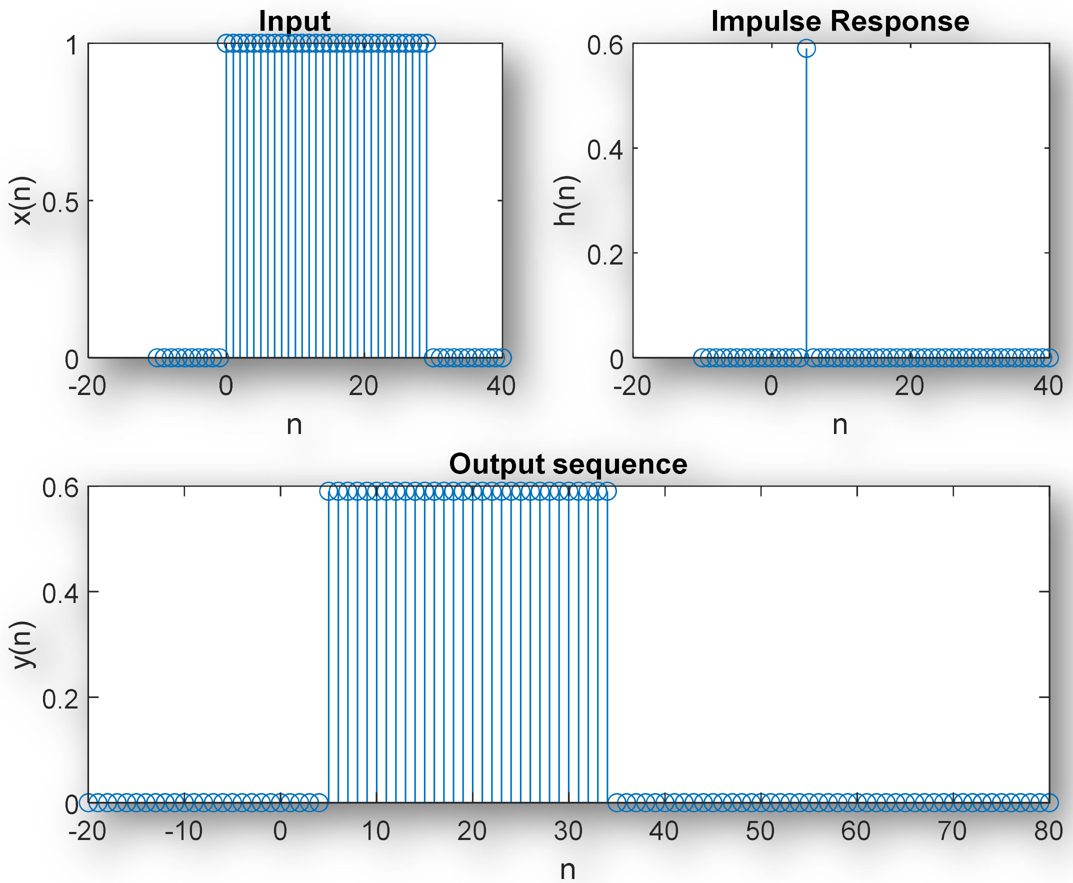
$$\boldsymbol{y(n) = 0.9^{n} * \ 0.9^{-n+5}}$$

$$\boldsymbol{y(n) = 0.9^{5}}$$

Ακολουθεί το πρόγραμμα $\boldsymbol{convol2.m}$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης μαζί και η συνάρτηση $\boldsymbol{stepd\_f.m}$:

```
%%Second Convolution exercise

clear all;
close all;
clc;

n1 = -10;
n2 = 40;

[u1,n1,d1]=stepd_f(0,n1,n2,5);
[u2,n2,d1]=stepd_f(30,n1,n2,5);

xn = u1-u2;
hn = (0.9.^n1).*d1;
yn = conv(xn,hn);

n1_new = n1(1)+n2(1);
n2_new = n1(length(u1)) + n2(length(u2));
n_new = [n1_new:1:n2_new];

figure(1);

subplot(221);
stem(n1,xn);
title('Input');
ylabel('x(n)');
xlabel('n');

subplot(222);
stem(n1,hn);
title('Impulse Response');
ylabel('h(n)');
xlabel('n');

subplot(212);
stem(n_new,yn);
title('Output sequence');
ylabel('y(n)');
xlabel('n');

print -depsc progr15;
```

```
%%function with δ and step

function[u,n,d]=stepd_f(n0,n1,n2,d0);

n = [n1:1:n2];

u = [(n-n0)>=0];

d = [(n-d0)==0];
```


Input
Impulse Response
Output sequence
x(n)
h(n)
y(n)
n

➢ ***Άσκηση 3η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{eterosys1.m}$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης μαζί και οι συναρτήσεις $\boldsymbol{shift\_f.m}$, $\boldsymbol{rev\_f.m}$, $\boldsymbol{add\_f.m}$ και $\boldsymbol{conv\_f.m}$ αντίστοιχα:

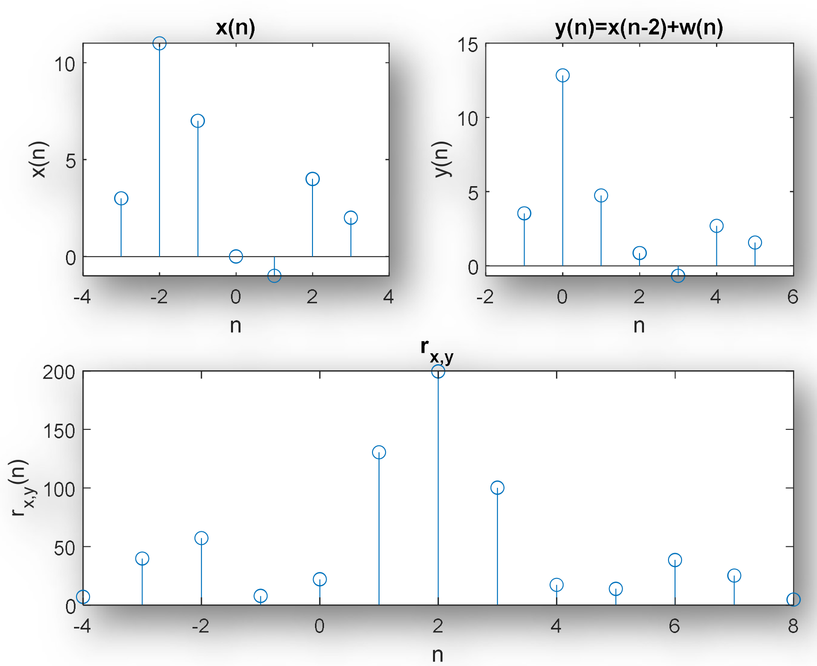
```
%% First Heterocorrelation exercise

clear all;
clc;
close all;

x = [3 11 7 0 -1 4 2];
nx = [-3:1:3];

[y1,ny1]= shift_f(x,nx,2);
w=randn(1,length(y1));
nw = ny1;

[y2,ny2] = add_f(y1,ny1,w,nw);
[x1,nx1] = rev_f(x,nx);

[rxy,nrxy] = conv_f(y2,ny2,x1,nx1);

figure(1);

subplot(2,2,1);
stem(nx,x);
title('x(n)');
xlabel('n');
ylabel('x(n)');

subplot(2,2,2);
stem(ny2,y2);
title('y(n)=x(n-2)+w(n)');
xlabel('n');
ylabel('y(n)');

subplot(2,1,2);
stem(nrxy,rxy);
title('r_{x,y}');
xlabel('n');
ylabel('r_{x,y}(n)');

print -depsc progr19;
```

```
%% shift function: y(n)=x(n+k)

function [y,ny] = shift_f(x,nx,k);
ny=nx+k;
y=x;
```

```
%% y(n)=x(-n)

function[y,ny] = rev_f(x,nx);
in_matr = [x;nx];
fin_matr = fliplr(in_matr);

y = fin_matr(1,:);
ny = -fin_matr(2,:);
```

```
%% y(n)=x1(n)+x2(n)

function[y,ny] = add_f(x1,nx1,x2,nx2);

n_min = min(min(nx1),min(nx2));
n_max = max(max(nx1),max(nx2));
ny=[n_min:1:n_max];

new_x1=zeros(1,length(ny));
new_x2=zeros(1,length(ny));

new_x1(find((ny>=min(nx1))&(ny<=max(nx1))==1)) = x1;
new_x2(find((ny>=min(nx2))&(ny<=max(nx2))==1)) = x2;

y = new_x1 + new_x2;
```

```
%% y=convolution(x1,x2)

function[y,ny] = conv_f(x1,nx1,x2,nx2)

ny1 = nx1(1)+nx2(1);
ny2 = nx1(length(x1)) + nx2(length(x2));
ny = ny1:1:ny2;
y = conv(x1,x2);
```



Απάντηση ερωτήσεων:

Ερώτημα $\mathbf{1}$:

Το πρόγραμμα $\boldsymbol{eterosys1.m}$ *έχει σαν σκοπό τον υπολογισμό της συνάρτησης ετεροσυσχέτισης και το επιτυγχάνει με τα ακόλουθα αλγοριθμικά βήματα.*

Αρχικά ορίζονται οι πίνακες $\boldsymbol{x}$ *και* $\boldsymbol{nx}$ *όπου* $\boldsymbol{x}$ *είναι η συνάρτηση* $\boldsymbol{x(n)}$ *και ο* $\boldsymbol{nx}$ *είναι οι τιμές* $\boldsymbol{n}$*. Καλείται η συνάρτηση* $\boldsymbol{shift\_f}$ *με παραμέτρους τα* $(\boldsymbol{x}, \boldsymbol{nx}, \boldsymbol{2})$ *που επιστρέφει τη* $\boldsymbol{x(n-2)}$*. Αποθηκεύεται στο πίνακα w έναν πίνακα* $\boldsymbol{1}\text{X}\boldsymbol{length(y1)}$ *πίνακα με τυχαία απεικόνιση από τη κανονική κατανομή. Στη συνέχεια, η τιμή της μεταβλητής* $\boldsymbol{ny1}$ *που μας υπολογίζεται από συνάρτηση* $\boldsymbol{shift\_f}$ *αποθηκεύεται στη μεταβλητή* $\boldsymbol{nw}$*. καλείται η συνάρτηση* $\boldsymbol{add\_f}$ *με παραμέτρους* $(\boldsymbol{y1}, \boldsymbol{ny1}, \boldsymbol{w})$ *και* $\boldsymbol{nw}$ *η οποία επιστρέφει τα* $\boldsymbol{y2}, \boldsymbol{ny2}$ *και έτσι υπολογίζεται η ποσότητα* $\boldsymbol{y(n)} = \boldsymbol{x(n-2)} + \boldsymbol{w(n)}$*. Η συνάρτηση* $\boldsymbol{rev\_f}$ *με παραμέτρους* $(\boldsymbol{x}, \boldsymbol{xn})$ *επιστρέφει τα* $\boldsymbol{x1}, \boldsymbol{nx1}$ *και έτσι υπολογίζεται το* $\boldsymbol{x(-n)}$*. Τέλος, πριν το τερματισμό του προγράμματος τυπώνονται γραφικά οι ποσότητες* $\boldsymbol{x(n)}, \boldsymbol{y(n)}$ *και* $\boldsymbol{r_{x,y}}$ *και αποθηκεύονται τα δεδομένα του προγράμματος όπως ακριβώς είδαμε και στις προηγούμενες εργασίες.*

Ερώτημα $\mathbf{2}$:

Στη συνάρτηση $\boldsymbol{shift\_f}$ *θέλουμε να πάρουμε μία συνάρτηση* $\boldsymbol{x(n)}$ *και να τη μετατοπίσουμε κατά* $\boldsymbol{k}$ *έτσι ώστε η νέα συνάρτηση που θα προκύψει να είναι* $\boldsymbol{y(n)} = \boldsymbol{x(n+k)}$*. Συνεπώς στη γραμμή 4 με την εντολή* $\boldsymbol{ny} = \boldsymbol{nx} + \boldsymbol{k}$ *μετατοπίζουμε κατά* $\boldsymbol{k}$ *τα* $\boldsymbol{n}$ *της* $\boldsymbol{n(x)}$ *και επιστρέφουμε τις μετατοπίσεις στη μεταβλητή* $\boldsymbol{ny}$*, ενώ στη γραμμή 5 κρατάμε σταθερές τις τιμές της συνάρτησης* $\boldsymbol{x(n)}$ *συνεπώς γράφουμε την εντολή* $\boldsymbol{y} = \boldsymbol{x}$*.*

Ερώτημα $\mathbf{3}$:

Στη συνάρτηση $\boldsymbol{rev\_f}$ *στη γραμμή 5 αποθηκεύονται τα στοιχεία του πίνακα* $\boldsymbol{in\_matr}$ *μετατοπισμένα από αριστερά στα δεξιά μέσα στο πίνακα* $\boldsymbol{fin\_matr}$*. Στη γραμμή 6 τα στοιχεία της πρώτης γραμμής του πίνακα* $\boldsymbol{fin\_matr}$ *αποθηκεύονται στο πίνακα* $\boldsymbol{y}$ *, καθώς τα στοιχεία της πρώτης γραμμής είναι τα στοιχεία που αναφέρονται στις τιμές του* $\boldsymbol{x}$*. Στη γραμμή 7 τα στοιχεία της δεύτερης γραμμής πολλαπλασιασμένα με το* $\boldsymbol{-1}$ *του πίνακα* $\boldsymbol{fin\_matr}$ *αποθηκεύονται στο πίνακα* $\boldsymbol{ny}$*, καθώς τα στοιχεία της δεύτερης γραμμής του πίνακα* $\boldsymbol{fin\_matr}$ *είναι οι τιμές* $\boldsymbol{nx}$*. Με αυτή την αλγοριθμική μέθοδο αυτή η συνάρτηση μπορεί να πραγματοποιήσει τη σχέση* $\boldsymbol{y(n)} = \boldsymbol{x(-n)}$ *, όπως περιγράφεται στο σχόλιο του προγράμματος.*

Ερώτημα $\mathbf{4}$:

Στη συνάρτηση $\boldsymbol{conv\_f}$ *στη γραμμή 4 αποθηκεύονται στη μεταβλητή* $\boldsymbol{ny1}$ *το άθροισμα των πρώτων στοιχείων των πινάκων* $\boldsymbol{nx1}$ *και* $\boldsymbol{nx2}$*. Στη γραμμή 5 αποθηκεύονται στη μεταβλητή* $\boldsymbol{ny2}$ *το άθροισμα των τελευταίων στοιχείων δηλαδή των στοιχείων που βρίσκονται στη θέση* $\boldsymbol{length(x1)}$ *και* $\boldsymbol{length(x2)}$ *των πινάκων* $\boldsymbol{nx1}$ *και* $\boldsymbol{nx2}$*, αντίστοιχα. Στη γραμμή 6 αποθηκεύονται στο νέο πίνακα* $\boldsymbol{ny}$ *στοιχεία από τη τιμή* $\boldsymbol{ny1}$ *έως τη τιμή* $\boldsymbol{ny2}$ *και στη γραμμή 7* *μέσω της εντολής* $\boldsymbol{y = conv(x1, x2)}$ *, υπολογίζεται η συνέλιξη των ακολουθιών* $\boldsymbol{x1}$ *και* $\boldsymbol{x2}$ *αναδρομικά.*

Ερώτημα $\mathbf{5}$:

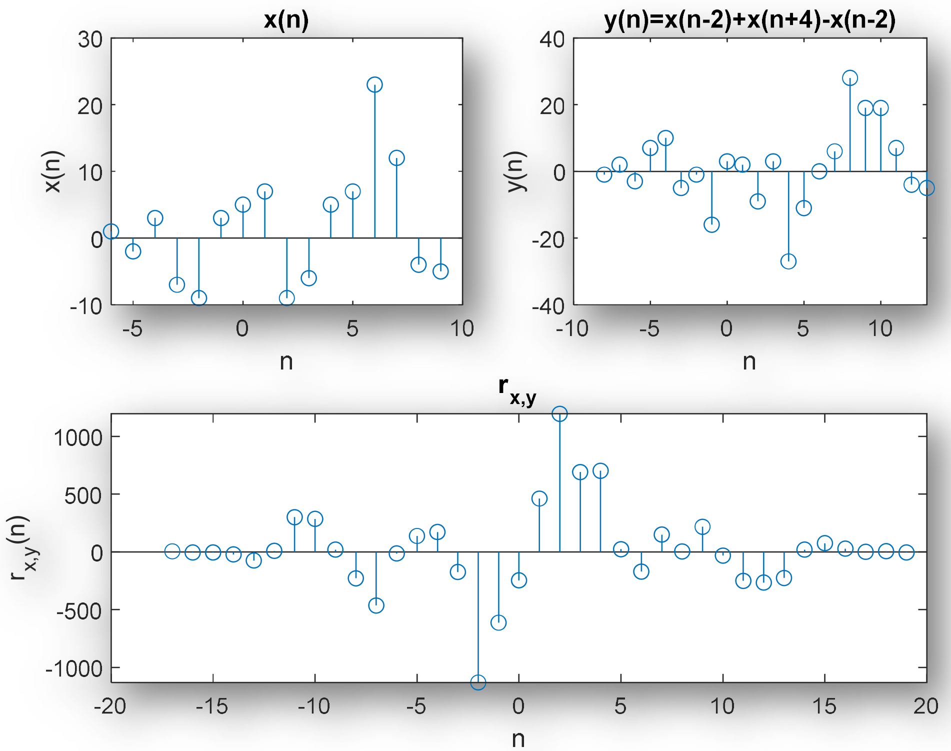
Στο πρόγραμμα $\boldsymbol{eterosys1.m}$ *στη γραμμή 8 αποθηκεύει στο πίνακα* $\boldsymbol{w}$ *ένα πίνακα* $\mathbf{1}\mathrm{X}\boldsymbol{length(y1)}$ *πίνακα με τυχαία απεικόνιση από τη κανονική κατανομή.*

➢ ***Άσκηση 4η :***

Ακολουθεί το πρόγραμμα:$\boldsymbol{cor1.m}$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης μαζί και η συνάρτηση $\boldsymbol{mim\_f.m}$ (επίσης έγινε και χρήση συναρτήσεων από προηγούμενες ασκήσεις):

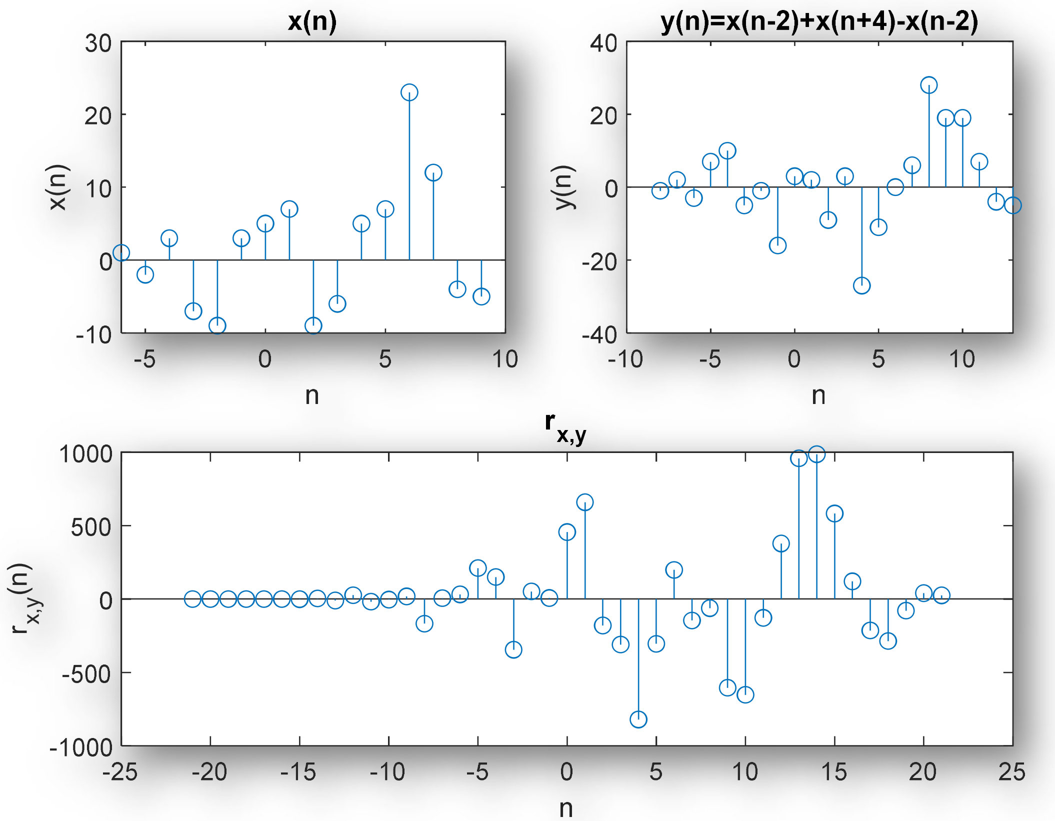
```
%%cor1

clear all;
clc;
close all;

x=[1 -2 3 -7 -9 3 5 7 -9 -6 5 7 23 12 -4 -5];
nx = [-6:1:9];

[y1,ny1] = shift_f(x,nx,2);
[y2,ny2] = shift_f(x,nx,4);
[y3,ny3] = shift_f(x,nx,-2);
[y4,ny4] = add_f(y1,ny1,y2,ny2);
[y5,ny5] = mim_f(y4,ny4,y3,ny3);
[x1,nx1] = rev_f(x,nx);
[rxy,nrxy] = conv_f(y5,ny5,x1,nx1);

figure(1);
subplot(2,2,1);
stem(nx,x);
title('x(n)');
xlabel('n');
ylabel('x(n)');

subplot(2,2,2);
stem(ny5,y5);
title('y(n)=x(n-2)+x(n+4)-x(n-2)');
xlabel('n');
ylabel('y(n)');

subplot(2,1,2);
stem(nrxy,rxy);
title('r_{x,y}');
xlabel('n');
ylabel('r_{x,y}(n)');

print -depsc progr19b;
```

```
%% y(n)=x1(n)-x2(n)
function[y,ny] = mim_f(x1,nx1,x2,nx2);
n_min = min(min(nx1), min(nx2));
n_max = max(max(nx1), max(nx2));
ny = [n_min:1:n_max];

new_x1 = zeros(1,length(ny));
new_x2 = zeros(1,length(ny));

new_x1(find((ny>=min(nx1))&(ny<=max(nx1))==1))=x1;
new_x2(find((ny>=min(nx2))&(ny<=max(nx2))==1))=x2;

y=new_x1-new_x2;
```


x(n)
y(n)=x(n-2)+x(n+4)-x(n-2)
x(n)
y(n)
n
n
r_{x,y}
r_{x,y}(n)
n

Ακολουθεί το πρόγραμμα:$cor2.m$ στην λειτουργική του μορφή και το αποτέλεσμα εκτέλεσης:

```
%%cor2

clear all;
clc;
close all;

x=[1 -2 3 -7 -9 3 5 7 -9 -6 5 7 23 12 -4 -5];
nx = [-6:1:9];

[y1,ny1] = shift_f(x,nx,2);
[y2,ny2] = shift_f(x,nx,4);
[y3,ny3] = shift_f(x,nx,-2);
[y4,ny4] = add_f(y1,ny1,y2,ny2);
[y5,ny5] = mim_f(y4,ny4,y3,ny3);

[x1,nx1] = rev_f(x,nx);
[rxy,nrxy] = xcorr(y5,x1);

figure(1);

subplot(2,2,1);
stem(nx,x);
title('x(n)');
xlabel('n');
ylabel('x(n)');

subplot(2,2,2);
stem(ny5,y5);
title('y(n)=x(n-2)+x(n+4)-x(n-2)');
xlabel('n');
ylabel('y(n)');

subplot(2,1,2);
stem(nrxy,rxy);
title('r_{x,y}');
xlabel('n');
ylabel('r_{x,y}(n)');

print -depsc progr20
```



Παρατηρούμε πως, η διαφορά στα δύο προγράμματα είναι ότι στο $\boldsymbol{cor1.m}$ *έγινε υπολογισμός της ετεροσυσχέτισης μέσω της συνάρτησης που υπολογίζει τη συνέλιξη συναρτήσεων* $\boldsymbol{y(n)}$ *και* $\boldsymbol{x(-n)}$*, ενώ στο* $\boldsymbol{cor2.m}$ *έγινε υπολογισμός της ετεροσυσχέτισης κατευθείαν μέσω της συνάρτησης* $\boldsymbol{xcorr}(\,)$ *η οποία καλείται με παραμέτρους τις τιμές της συνάρτησης* $\boldsymbol{y(n)}$ *και του* $\boldsymbol{x(n)}$*. Ακόμη σημειώνουμε πως, δεν χρειάστηκε η συνάρτηση* $\boldsymbol{rev\_f}$ *για να μας δώσει τη* $\boldsymbol{x(-n)}$*. Ακόμη, παρατηρούμε μία μετατόπιση κατά* $\boldsymbol{n = 2}$ *στις γραφικές παραστάσεις των εξόδων των δύο προγραμμάτων. Αυτή η διαφορά εξαρτάται από τη κανονικοποίηση που έχουμε κάνει και από το πόσο καλά έχουμε προβλέψει τα* $\boldsymbol{n}$ *στα οποία αναμένουμε να εμφανιστεί η συνάρτηση ετεροσυσχέτισης. Βέβαια η μετατόπιση κατά δύο θέσεις δεξιά της συνάρτησης, δεν επηρεάζει το αποτέλεσμα της συνάρτησης ετεροσυσχέτισης.*