

# **ΔΙΣΔΙΑΣΤΑΤΗ ΣΥΝΕΛΙΞΗ ΜΕΣΩ ΤΟΥ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ FFTW**

**ΒΑΣΙΛΗΣ ΚΙΤΣΑΚΗΣ**  
**2014509**

# ΠΕΡΙΕΧΟΜΕΝΑ

- 1) Εισαγωγή
- 2) Οδηγίες εγκατάστασης βιβλιοθηκών
- 3) Οδηγίες χρήσης
- 4) Παραδείγματα
  - A) Τυχαιοί μιγαδικοί πίνακες
  - B) Εισαγωγή πραγματικών πινάκων από αρχεία
- 5) Κώδικας

## 1) Εισαγωγή

Στο πρόγραμμα αυτό εκτελείται η δισδιάστατη συνέλιξη δύο δισδιάστατων πινάκων μέσω του μετασχηματισμού fft με χρήση της βιβλιοθήκης fftw. Έχουμε δύο δισδιάστατους πίνακες a και b βρίσκουμε τις διαστάσεις της συνέλιξης, τον μετασχηματισμό του καθενός A και B τους πολλαπλασιάζουμε μεταξύ τους ανα σημείο  $A*B$  και στο αποτέλεσμα Y ( $Y=A*B$ ) κάνουμε αντίστροφο μετασχηματισμό και έτσι έχουμε υπολογίσει την συνέλιξη τους. Δηλαδή εκτελεί τις εξής εντολές σε matlab για δυο δοσμένα δισδιάστατα διανύσματα a και b

```
m=size(a,1)+size(b,1)-1 ;  
n=size(a,2)+size(b,2)-1;
```

```
A=fft2(a,m,n);  
B=fft2(b,m,n);
```

```
Y=A.*B;
```

```
y=ifft2(Y)
```

ή

```
y=conv2(a,b)
```

## 2) Οδηγίες εγκατάστασης βιβλιοθηκών

Η βιβλιοθήκες που έχουν χρησιμοποιηθεί είναι η fftw και η pthread (posix threads)

### Εγκατάσταση pthread

#### Unix

Στα συστήματα unix υπάρχει η βιβλιοθήκη pthread αλλά σε κάποιες περιπτώσεις ίσως χρειαστεί να γίνει εγκατάσταση. Η εγκατάσταση γίνεται εκτελώντας την εντολή “sudo apt-get install libpthread-stubs0-dev” στο terminal.

#### Windows

Τα posix threads δεν χρησιμοποιούνται από τα windows αλλά υπάρχει όμως υλοποίηση τους η οποία ας πούμε ότι “προσομοιώνει” την λειτουργία τους. Η υλοποίηση αυτής της βιβλιοθήκης πρέπει να ληφθεί από εδώ <https://www.sourceware.org/pthreads-win32> Πρέπει να ληφθεί το αντίστοιχο αρχείο 32-bit ή 64-bit ανάλογα το σύστημά μας. Μετά πρέπει να γίνει αποσυμπίεση του αρχείου που κατεβάσαμε. Τα αποσυμπιεσμένα αρχεία πρέπει να είναι κάποια με επεκτάσεις .dll .lib .def .h . Αν πρόκειται να γίνει το compile μέσο τερματικού θα πρέπει να τοποθετηθούν τα παραπάνω αρχεία στο φάκελο C:\WINDOWS\system32 . Αν το compile πρόκειται να γίνει σε κάποιο περιβάλλον ανάπτυξης (IDE) θα πρέπει τα αρχεία να συνδεθούν-

περαστούν μέσα στις βιβλιοθήκες του.

## Pthread compile

Κατά το compile πρέπει στο πηγαίο αρχείο μας να .c να συνδέσουμε την βιβλιοθήκη μέσω της εντολής -pthread πχ “gcc fft.c -pthread”

## **Εγκατάσταση fftw**

### Ubuntu

Στα συστήματα Ubuntu οι βιβλιοθήκες του fftw υπάρχουν αλλά θα πρέπει να εγκατασταθούν μέσω της εντολής “sudo apt-get install libfftw3-dev libfftw3-doc” (το libfftw3-doc περιέχει κάποιες οδηγίες, η εγκατάσταση του είναι προαιρετική και μπορεί να παραληφθεί) στο terminal.

### Unix

Γενικά σε άλλα συστήματα unix που δεν υπάρχουν οι βιβλιοθήκες θα πρέπει να γίνει λήψη και εγκατάσταση. Για την τωρινή έκδοση αυτό μπορεί να γίνει με τις εξείς εντολές στο terminal

```
mkdir $HOME/usr  
mkdir $HOME/soft
```

```
cd ~/soft
```

```
wget http:// www.fftw.org/fftw-3.3.4.tar.gz
```

```
tar -zxvf fftw-3.3.4.tar.gz  
cd fftw-3.3.4
```

## Windows

Οι βιβλιοθήκες πρέπει να ληφθούν από εδώ <http://www.fftw.org/install/windows.html> Μετά πρέπει να γίνει αποσυμπίεση του αρχείου που κατεβάσαμε. Τα αποσυμπιεσμένα αρχεία πρέπει να είναι κάποια με επεκτάσεις .dll .lib .def .h . Αν πρόκειται να γίνει το compile μέσω τερματικού θα πρέπει να τοποθετηθούν τα παραπάνω αρχεία στο φάκελο C:\WINDOWS\system32 . Αν το compile πρόκειται να γίνει σε κάποιο περιβάλλον ανάπτυξης (IDE) θα πρέπει τα αρχεία να συνδεθούν-περαστούν μέσα στις βιβλιοθήκες του.

## FFTW compile

Κατά το compile πρέπει στο πηγαίο αρχείο μας να .c να συνδέσουμε την βιβλιοθήκη μέσω της εντολής -lfftw3 (το 3 δηλώνει την έκδοση άμα έχουμε κάποια παλαιότερη θα πρέπει να αντικατασταθεί με την αντίστοιχη, μόνο το πρώτο ψηφίο της έκδοσης) πχ “gcc fft.c -lfftw3”

### 3) Οδηγίες χρήσης

Εκτελώντας το πρόγραμμα εμφανίζεται το εξής menu

```
(a) dimiourgia input files  
(r) random complex  
(s) random integer  
(f) random real  
(p) diabasma apo arxeia real  
(c) diabasma apo arxeia complex  
(q) quit
```

(a) Για να εκτελεστεί το πρόγραμμα πρέπει να υπάρχουν στο ίδιο φάκελο (directory) τα εξής αρχεία, τα οποία μπορούν να τροποποιηθούν με κάποιον text editor,

```
input_a  
input_b  
input_a_complex  
input_b_complex  
rand_a  
rand_b  
output
```

Εάν τα αρχεία αυτά δεν υπάρχουν τα 4 πρώτα μπορούν να δημιουργηθούν με αυτήν την εντολή (τα υπόλοιπα δημιουργούνται με την εκτέλεση των εντολών (r),(s),(f) ). Προσοχή εάν είδη υπάρχουν και τα ξαναδημιουργήσουμε θα χάσουμε ότι τυχόν ήταν αποθηκευμένο στα πρώτα μιας και θα αντικατασταθούν με καινούρια κενά!

(r) Με αυτήν την εντολή θα εκτελέσουμε την συνέλιξη δύο τυχαίων μιγαδικών πινάκων. Σαν είσοδο θα πρέπει να δώσουμε τις διαστάσεις των πινάκων και ένα όριο-μέγιστο που θα έχει το κάθε στοιχείο του πίνακα. Οι πίνακες που δημιουργούνται αποθηκεύονται στα αρχεία `rand_a` και `rand_b` και το αποτέλεσμα στο `output` . Προσοχή κάθε φορά που εκτελείται αυτή η εντολή τα προηγούμενα αρχεία χάνονται και αντικαθιστώνται από καινούρια και έτσι μπορεί να χαθούν αποτελέσματα που πιθανών χρειαζόμασταν οπότε φροντίστε να σώσετε ότι χρειάζεστε σε αρχεία με διαφορετικό όνομα. Η τελεία (.) δηλώνει υποδιαστολή και το κόμμα χωρίζει το πραγματικό από το φανταστικό μέλος. Ο αριθμός πχ 3,8 διαβάζεται 3+8i και ο 3.56,0.98 διαβάζεται 3.56+0.98i .

(s) Όμοια με πριν μόνο που τώρα εκτελείται για πραγματικούς πίνακες και πάλι η τελεία (.) δηλώνει υποδιαστολή και οι αριθμοί διαβάζονται όπως φαίνονται πχ 9 ή 5.956 .

(f) Όμοια με πριν μόνο που τώρα εκτελείται για πίνακες με ακέραιους.

(p) Με την εντολή αυτή εκτελείται η συνέλιξη δύο πραγματικών πινάκων οι οποίοι διαβάζονται σαν είσοδος από τα αρχεία `input_a` και `input_b` τα οποία συμπληρώνουμε εμείς σε μορφή πίνακα πχ

9 8 2  
5 7 1.8



τα στοιχεία μιας στήλης χωρίζονται με κενά και οι στήλες χωρίζονται με αλλαγή γραμμής (enter). Το πρόγραμμα τους διαβάσει και βρίσκει μόνο του τις διαστάσεις και τυπώνει το αποτέλεσμα στο αρχείο output. Προσοχή αν το αρχείο output περιέχει στοιχεία θα χαθούν για αυτό φροντίστε να σώσετε σε άλλο αρχείο ότι χρειάζεστε.

(c) Όμοια με πριν μόνο που τώρα εκτελείται για μιγαδικούς πίνακες. Η τελεία (.) δηλώνει υποδιαστολή και το κόμμα χωρίζει το πραγματικό από το φανταστικό μέλος. Οι πίνακες διαβάζονται σαν είσοδος από τα αρχεία input\_a\_complex και input\_b\_complex τα οποία συμπληρώνουμε εμείς σε μορφή πίνακα αυτή τη φορά όμως έτσι

9,8 8,4 2,1

5,0 7,-5 1.7,4.1

και είναι σαν να γράψαμε τον πίνακα

9+8i 8+4i 2+i

5 7-5i 1.7+4.1i

(q) Έξοδος από το πρόγραμμα.

## 4) Παραδείγματα

```
(a) create input files  
(r) random complex  
(s) random real  
(p) diabasma apo arxeia real  
(c) diabasma apo arxeia complex  
(q) quit
```

A) Τυχαίοι μιγαδικοί πίνακες

Επιλέγουμε (r)

```
(a) create input files  
(r) random complex  
(s) random real  
(p) diabasma apo arxeia real  
(c) diabasma apo arxeia complex  
(q) quit  
r
```

```
dose diastaseis n0xn1 n1xn2  
n0=
```

Δίνουμε αριθμό γραμμών του πρώτου πίνακα  
1000

```
dose diastaseis n0xn1 n1xn2  
n0=1000  
n1=
```

Δίνουμε αριθμό στηλών του πρώτου πίνακα 2000

```
dose diastaseis n0xn1 n2xn3  
n0=1000  
n1=2000  
n2=
```

Δίνουμε τον αριθμό των γραμμών του δευτέρου πίνακα 6

```
dose diastaseis n0xn1 n2xn3  
n0=1000  
n1=2000  
n2=6  
n3=
```

Δίνουμε τον αριθμό των στηλών του δευτέρου πίνακα 6

```
dose diastaseis n0xn1 n2xn3  
n0=1000  
n1=2000  
n2=6  
n3=6
```

Και τέλος το όριο-μέγιστο του κάθε αριθμού 10

```
dose orio tuxaion ariston 10
```

Περιμένουμε να εκτυπωθούν οι πίνακες α και β να γίνει η συνέλιξη και να τυπωθεί το αποτέλεσμα. Επίσης μας εμφανίζει τον χρόνο που έκανε να εκτελεστεί η συνέλιξη σε δευτερόλεπτα. Τέλος έχουμε αυτήν την εικόνα και είμαστε έτοιμοι να συνεχίσουμε σε άλλη εντολή.

```
dose diastaseis n0xn1 n2xn3
n0=1000
n1=2000
n2=6
n3=6
dose orio tuxaion ari8mon 10

printing a.....

printing b.....

printing y.....
xronos ektelesis suneliksis 0.840000
(a) dimiourgia input files
(r) random complex
(s) random integer
(f) random real
(p) diabasma apo arxeia real
(c) diabasma apo arxeia complex
(q) quit
```

Τα αρχεία μας έχουν την εξής εικόνα

rand\_a

rand_a	rand_b	output
1	6.000,6.000	1.000,5.000
2	1.000,6.000	6.000,6.000
3	6.000,6.000	0.000,4.000
4	0.000,4.000	4.000,7.000
5	6.000,0.000	9.000,1.000
6	3.000,8.000	2.000,8.000
7	4.000,8.000	4.000,8.000
8	3.000,8.000	5.000,6.000
9	9.000,7.000	6.000,7.000
10	6.000,7.000	6.000,6.000
11	4.000,4.000	0.000,9.000
12	5.000,7.000	1.000,8.000
13	8.000,5.000	8.000,5.000
14	8.000,5.000	8.000,5.000
15	6.000,4.000	4.000,7.000
16	7.000,9.000	9.000,8.000
17	9.000,8.000	7.000,2.000
18	0.000,5.000	4.000,4.000
19	2.000,0.000	6.000,0.000
20	6.000,0.000	6.000,0.000
21	9.000,0.000	9.000,0.000
22	7.000,4.000	1.000,1.000
23	6.000,5.000	1.000,5.000
24	9.000,1.000	9.000,1.000
25	3.000,7.000	7.000,3.000
26	3.000,9.000	2.000,3.000
27	2.000,3.000	2.000,2.000
28	8.000,1.000	4.000,0.000
29	7.000,2.000	7.000,2.000
30	3.000,1.000	6.000,1.000
31	8.000,4.000	8.000,4.000
32	3.000,2.000	8.000,6.000
33	1.000,5.000	8.000,9.000
34	6.000,3.000	6.000,1.000
35	3.000,5.000	1.000,0.000
36	9.000,6.000	9.000,6.000
37	4.000,7.000	0.000,3.000
38	5.000,5.000	5.000,5.000
39	8.000,0.000	7.000,4.000
40	7.000,4.000	6.000,6.000
41	5.000,0.000	7.000,5.000
42	6.000,6.000	5.000,0.000
43	7.000,5.000	6.000,6.000
44	9.000,6.000	9.000,6.000
45	2.000,7.000	1.000,1.000
46	0.000,6.000	8.000,9.000
47	6.000,5.000	6.000,5.000
48	8.000,2.000	9.000,2.000
49	9.000,2.000	9.000,2.000
50	9.000,6.000	3.000,6.000
51	3.000,6.000	9.000,7.000
52	5.000,9.000	3.000,7.000
53	1.000,1.000	0.000,6.000
54	8.000,9.000	6.000,5.000
55	6.000,5.000	1.000,4.000
56	1.000,9.000	1.000,9.000
57	2.000,0.000	8.000,4.000
58	3.000,2.000	9.000,6.000
59	4.000,0.000	5.000,3.000
60	5.000,3.000	8.000,7.000
61	8.000,7.000	2.000,6.000
62	0.000,4.000	5.000,4.000
63	6.000,1.000	5.000,7.000
64	2.000,5.000	3.000,0.000
65	3.000,0.000	2.000,1.000
66	4.000,9.000	7.000,9.000
67	6.000,2.000	3.000,9.000
68	9.000,5.000	3.000,2.000
69	8.000,8.000	4.000,5.000
70	4.000,5.000	1.000,8.000
71	1.000,8.000	1.000,1.000
72	8.000,0.000	8.000,0.000
73	7.000,8.000	7.000,8.000
74	4.000,5.000	1.000,1.000
75	6.000,3.000	6.000,3.000
76	1.000,1.000	8.000,0.000
77	8.000,0.000	7.000,8.000
78	9.000,2.000	4.000,0.000
79	2.000,3.000	3.000,0.000
80	5.000,9.000	0.000,1.000
81	0.000,1.000	1.000,8.000
82	6.000,2.000	6.000,2.000
83	6.000,3.000	1.000,1.000
84	1.000,1.000	6.000,7.000
85	7.000,7.000	8.000,0.000
86	0.000,9.000	7.000,8.000
87	7.000,8.000	1.000,7.000
88	9.000,8.000	4.000,7.000
89	4.000,7.000	7.000,1.000
90	7.000,1.000	2.000,7.000
91	0.000,4.000	5.000,3.000
92	5.000,3.000	7.000,4.000
93	7.000,1.000	2.000,5.000
94	2.000,0.000	2.000,0.000
95	9.000,0.000	9.000,0.000
96	0.000,0.000	0.000,0.000
97	1.000,7.000	2.000,0.000
98	2.000,2.000	8.000,6.000
99	8.000,6.000	3.000,5.000
100	5.000,5.000	5.000,6.000

rand\_b

rand_a	rand_b	output
1	5.000,6.000	8.000,5.000
2	1.000,0.000	1.000,0.000
3	8.000,3.000	3.000,9.000
4	9.000,8.000	5.000,7.000
5	5.000,7.000	6.000,3.000
6	4.000,1.000	5.000,9.000
7	5.000,9.000	4.000,2.000
8	5.000,3.000	5.000,3.000
9	2.000,2.000	4.000,6.000
10	8.000,9.000	2.000,1.000
11	4.000,2.000	4.000,2.000
12	5.000,3.000	5.000,3.000
13	4.000,2.000	2.000,1.000
14	8.000,3.000	8.000,9.000
15	2.000,3.000	8.000,6.000
16	8.000,6.000	5.000,0.000
17	5.000,0.000	4.000,9.000
18	1.000,2.000	4.000,9.000
19	4.000,9.000	8.000,3.000
20	5.000,0.000	5.000,0.000

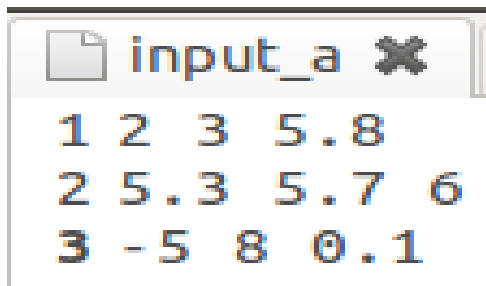
# output

rand_a	rand_b	output
1	-6.000,66.000	-7.000,109.000
	-18.000,135.000	-43.000,180.000
	5.000,180.000	18.000,221.000
	19.000,229.000	126.000,222.000
	101.000,204.000	33.000,237.000
	17.000,277.000	23.000,224.000
	41.000,264.000	29.000,340.000
	61.000,337.000	86.000,336.000
	76.000,303.000	53.000,253.000
	39.000,281.000	8.000,285.000
	34.000,272.000	52.000,306.000
	93.000,296.000	85.000,290.000
	52.000,313.000	112.000,340.000
	155.000,312.000	105.000,243.000
	64.000,241.000	107.000,225.000
	149.000,153.000	133.000,125.000
	157.000,157.000	151.000,180.000
	145.000,189.000	151.000,168.000
	117.000,144.000	120.000,150.000
	144.000,202.000	85.000,193.000
	71.000,227.000	33.000,251.000
	6.000,188.000	75.000,228.000
	77.000,238.000	95.000,247.000
	20.000,272.000	48.000,205.000
	78.000,191.000	80.000,186.000
	146.000,165.000	136.000,181.000
	112.000,138.000	146.000,123.000
	155.000,149.000	150.000,139.000
	133.000,191.000	73.000,179.000
	92.000,234.000	89.000,268.000
	142.000,226.000	106.000,214.000
	88.000,181.000	86.000,202.000
	46.000,133.000	52.000,171.000
	23.000,207.000	84.000,231.000
	136.000,261.000	138.000,225.000
	149.000,238.000	181.000,234.000
	166.000,206.000	139.000,195.000
	150.000,284.000	146.000,294.000
	130.000,236.000	137.000,267.000
	124.000,227.000	170.000,202.000
	171.000,239.000	176.000,235.000
	144.000,231.000	103.000,272.000
	144.000,217.000	109.000,169.000
	27.000,256.000	71.000,296.000
	96.000,252.000	16.000,205.000
	33.000,205.000	61.000,174.000
	61.000,174.000	108.000,180.000
	88.000,193.000	28.000,232.000
	48.000,255.000	70.000,189.000
	27.000,201.000	12.000,237.000
	56.000,224.000	75.000,171.000
	85.000,238.000	92.000,228.000
	141.000,156.000	113.000,211.000
	96.000,230.000	53.000,242.000
	3.000,176.000	-4.000,185.000
	1.000,256.000	11.000,297.000
	-15.000,218.000	-25.000,204.000
	67.000,236.000	107.000,224.000
	32.000,218.000	55.000,194.000
	81.000,126.000	52.000,146.000
	28.000,273.000	67.000,266.000
	38.000,212.000	-12.000,241.000
	49.000,345.000	96.000,262.000
	93.000,242.000	51.000,308.000
	53.000,270.000	47.000,200.000
	84.000,190.000	109.000,223.000
	136.000,258.000	135.000,240.000
	113.000,115.000	138.000,129.000
	101.000,193.000	86.000,131.000
	10.000,151.000	-9.000,167.000
	84.000,201.000	82.000,170.000
	32.000,184.000	45.000,259.000
	159.000,233.000	77.000,199.000
	21.000,240.000	46.000,311.000
	65.000,283.000	42.000,293.000
	88.000,310.000	72.000,273.000
	39.000,215.000	74.000,243.000
	72.000,206.000	141.000,191.000
	82.000,209.000	68.000,151.000
	139.000,166.000	157.000,93.000
	87.000,97.000	29.000,175.000
	55.000,148.000	101.000,161.000
	98.000,208.000	135.000,170.000
	95.000,193.000	22.000,214.000
	64.000,156.000	53.000,190.000
	91.000,219.000	98.000,264.000
	96.000,243.000	141.000,187.000
	101.000,169.000	90.000,216.000
	138.000,264.000	111.000,185.000
	95.000,198.000	94.000,204.000
	125.000,169.000	78.000,217.000
	28.000,250.000	52.000,230.000
	65.000,174.000	59.000,168.000
	19.000,194.000	65.000,182.000
	2.000,177.000	4.000,176.000
	41.000,234.000	-18.000,212.000
	-2.000,164.000	20.000,180.000
	28.000,221.000	22.000,282.000
	21.000,274.000	21.000,254.000
	72.000,253.000	107.000,235.000
	96.000,190.000	133.000,159.000
	77.000,162.000	41.000,157.000
	65.000,186.000	49.000,200.000
	21.000,187.000	51.000,250.000
	83.000,242.000	81.000,166.000
	64.000,192.000	58.000,190.000
	68.000,186.000	-39.000,190.000
	-0.000,216.000	17.000,250.000
	34.000,266.000	83.000,280.000
	128.000,208.000	135.000,247.000
	109.000,266.000	201.000,223.000
	141.000,185.000	74.000,154.000
	114.000,197.000	84.000,225.000
	19.000,227.000	25.000,198.000
	71.000,149.000	52.000,202.000
	41.000,261.000	53.000,247.000
	2.000,226.000	48.000,229.000
	118.000,274.000	81.000,307.000
	22.000,306.000	24.000,319.000
	87.000,315.000	96.000,293.000
	41.000,270.000	90.000,282.000
	124.000,293.000	90.000,262.000
	12.000,254.000	46.000,237.000
	94.000,228.000	40.000,192.000
	-12.000,233.000	7.000,240.000
	75.000,255.000	49.000,237.000
	71.000,140.000	92.000,193.000
	71.000,198.000	106.000,186.000
	88.000,190.000	96.000,188.000
	76.000,266.000	37.000,277.000
	45.000,225.000	30.000,249.000
	56.000,269.000	105.000,272.000
	116.000,305.000	83.000,248.000
	84.000,189.000	117.000,208.000
	101.000,206.000	31.000,185.000
	43.000,215.000	130.000,199.000
	104.000,187.000	35.000,196.000
	26.000,205.000	50.000,243.000
	63.000,271.000	61.000,268.000
	59.000,218.000	86.000,253.000
	102.000,214.000	92.000,188.000
	35.000,221.000	66.000,254.000
	38.000,227.000	-26.000,185.000
	-39.000,285.000	-31.000,293.000
	-7.000,268.000	-41.000,308.000
	-47.000,304.000	-7.000,268.000
	23.000,256.000	100.000,225.000
	147.000,219.000	83.000,175.000
	112.000,129.000	66.000,129.000
	24.000,153.000	18.000,222.000
	11.000,222.000	31.000,237.000
	73.000,225.000	119.000,195.000
	75.000,177.000	67.000,181.000
	88.000,200.000	129.000,168.000
	109.000,223.000	51.000,253.000
	85.000,217.000	123.000,207.000
	108.000,262.000	64.000,324.000
	400.000,300.000	453.000,336.000
	444.000,377.000	404.000,300.000
	453.000,300.000	03.000,437.000
	04.000,405.000	433.000,330.000

## B) Εισαγωγή πραγματικών πινάκων απο αρχεία

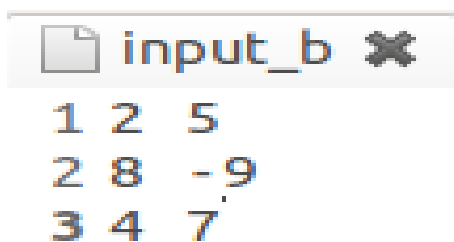
Έστω ότι δίνουμε για είσοδο τους εξείς πίνακες

input\_a



1	2	3	5.8
2	5.3	5.7	6
3	-5	8	0.1

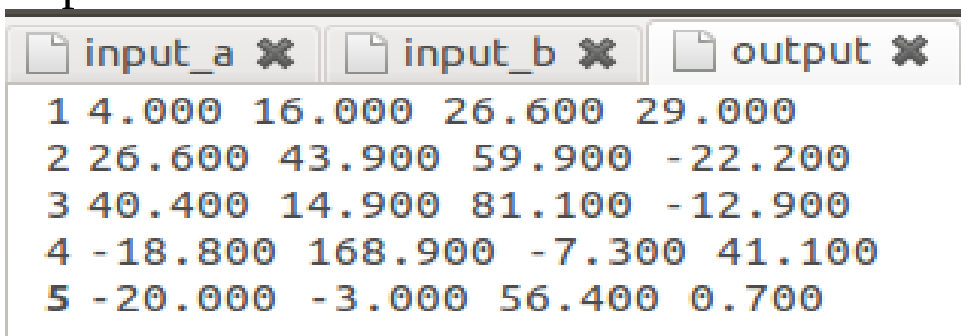
input\_b



1	2	5
2	8	-9
3	4	7

Με την επιλογή (p) το πρόγραμμα βρίσκει τις διαστάσεις των πινάκων και εκτελεί την συνέλιξη και τυπώνει το αποτέλεσμα στον πίνακα output. Επίσεις μας εμφανίζει τον χρόνο που έκανε να εκτελεστεί η συνέλιξη σε δευτερόλεπτα.

output



1	4.000	16.000	26.600	29.000
2	26.600	43.900	59.900	-22.200
3	40.400	14.900	81.100	-12.900
4	-18.800	168.900	-7.300	41.100
5	-20.000	-3.000	56.400	0.700

Τέλος, έχουμε αυτήν την εικόνα και είμαστε έτοιμοι να συνεχίσουμε σε άλλη εντολή.

```
(a) dimiourgia input files
(r) random complex
(s) random integer
(f) random real
(p) diabasma apo arxeia real
(c) diabasma apo arxeia complex
(q) quit
p

oi diastaseis einai 3x3 kai 3x2

oi diastaseis tis suneliksisis einai 5x4

printing y.....
xronos ektelesis suneliksisis 0.000000
(a) dimiourgia input files
(r) random complex
(s) random integer
(f) random real
(p) diabasma apo arxeia real
(c) diabasma apo arxeia complex
(q) quit
```



## 5)Κώδικας

```
#include <stdio.h>
#include <stdlib.h>
#include <complex.h>
#include <fftw3.h>
#include <pthread.h>
#include <time.h>
#define NUM_THREADS 2

struct thread_data{ fftw_plan plan;};// ena struct pou
perixe ta arguments pou 8elw na perasw sto thread

struct thread_data thread_data_array[NUM_THREADS];//
ftiaxnw ena pinaka oste na mporei to ka8e thread na parei
diaforetika arguments

/*
void mult( fftw_complex *c,fftw_complex *a,
fftw_complex *b,int n0,int n1,int n2 )
{ //polaplasiasmos pinakwn complex ari8mwn c[n0]
[n2]=a[n0][n1]*b[n1][n2]
  int i, j, k;
  for (i=0; i<n0; ++i)
    {for (j=0; j<n2; ++j)
      { c[i*n2+j] = 0.0;
        for (k=0; k<n1; ++k)
          {c[i*n2+j] =c[i*n2+j]+a[i*n1+k] * b[k*n2+j];}} } //oi
pinakes einai apo malloc opote diabazontai san
monodiastatoi
```

```

//opou px to stoixeio
c[i][j] tou pinaka c[n0][n2] einai to c[i*n2+j]
}

```

```

*/

```

```

void mult( fftw_complex *c,fftw_complex *a,
fftw_complex *b,int n0,int n1)

```

```

//polaplasiasmos simiwn twv pinakwn
{
    int i;
    for (i=0; i<(n0*n1); i++)
    {
        c[i] =a[i] * b[i];
    }
}

```

```

void insert(fftw_complex *a,fftw_complex *b,int n0,int
n1,int n11,int n2,int lim,int n)// ena insert opou sto pinaka
a[n0][n1] kai b[n1][n2]
{int i,j;time_t t;srand((unsigned)time(&t)); //ton gemizei me
tuxea migadika stoixeia
for(i=0;i<n0;i++)
{for(j=0;j<n1;j++)
{a[i*n+j]=rand()%lim+rand()%lim*I;
}
}
for(i=0;i<n11;i++)

```

```

    {for(j=0;j<n2;j++)
      {b[i*n+j]=rand()%lim+rand()%lim*I;
      }
    }
}

```

```

void insert_int(fftw_complex *a,fftw_complex *b,int n0,int
n1,int n11,int n2,int lim,int n)// ena insert opou sto pinaka
a[n0][n1] kai b[n1][n2]
{int i,j;time_t t;srand((unsigned)time(&t)); //ton gemizei me
tuxea akeraia stoixeia
for(i=0;i<n0;i++)
  {for(j=0;j<n1;j++)
    {a[i*n+j]=rand()%lim+0*I;
    }
  }
for(i=0;i<n11;i++)
  {for(j=0;j<n2;j++)
    {b[i*n+j]=rand()%lim+0*I;
    }
  }
printf("\n");
}

```

```

void insert_real(fftw_complex *a,fftw_complex *b,int
n0,int n1,int n11,int n2,int lim,int n)// ena insert opou sto
pinaka a[n0][n1] kai b[n1][n2]
{int i,j;time_t t;srand((unsigned)time(&t)); //ton gemizei me
tuxea pragmatika stoixeia
lim--;

```

```

for(i=0;i<n0;i++)
{
    for(j=0;j<n1;j++)
    {
        a[i*n+j]=(rand()%lim+(rand()%999)/1000.0)+0*I;
    }
}
for(i=0;i<n11;i++)
{
    for(j=0;j<n2;j++)
    {
        b[i*n+j]=(rand()%lim+(rand()%999)/1000.0)+0*I;
    }
}
printf("\n");
}

```

void print(fftw\_complex \*a,int n0,int n1,int n,FILE \*f)//  
 ektuposi pinaka a[n0][n1] me morfi (pragmatiko  
 meros,fantastiko meros)  
 {int i,j;

```

for(i=0;i<n0;i++)
{
    for(j=0;j<n1;j++)
    {
        fprintf(f,"%0.3lf,%0.3lf ",creal(a[i*n+j]),cimag(a[i*n+j]));
    }
    fprintf(f,"\n");
}
}

```

void print\_real(fftw\_complex \*a,int n0,int n1,int n,FILE  
 \*f)// ektuposi pinaka a[n0][n1]  
 {int i,j;

```

for(i=0;i<n0;i++)

```

```

{for(j=0;j<n1;j++)
  {fprintf(f,"%0.3lf ",creal(a[i*n+j]));
  }fprintf(f,"\n");
}
}

```

void normalize\_2D(fftw\_complex \*y,int n0,int n2)// mia  
sunartisi pou sto antistrofo metasximatismo diairei ola ta  
stoixeia me

```

{int i;for(i=0;i<(n0*n2);i++){y[i]=y[i]/(n0*n2);}} //ton
ari8mo ton bimatwn tou metasximatismou (n0*n2 epeidi
eimaste se

```

//disdiastato) mias kai den to  
kanei apo moni tis i sunartisi tis biblio8ikis

int plithos(FILE \*f)// mia sunartisi pou metraei ola ta  
stoixeia tou pragmatikou pinaka

```

{int pli8os;double b;
while(!feof(f))
{fscanf(f,"%lf",&b);
pli8os++;
}return pli8os;
}

```

int plithos\_complex(FILE \*f)// mia sunartisi pou metraei  
ola ta stoixeia tou complex pinaka

```

{int pli8os;double im,re;
while(!feof(f))
{fscanf(f,"%lf,%lf",&re,&im);
pli8os++;
}return pli8os;

```

```
}
```

```
int error_real(FILE *f)//elenxos dedomenwn  
{  
int check;double b;  
while(!feof(f))  
{if (fscanf(f,"%lf ",&b)!=1)  
{return 1;}  
}  
return 0;  
}
```

```
int error_complex(FILE *f)// elenxos dedomenwn gia ta  
arxeia complex  
{double im,re;  
while(!feof(f))  
{if(fscanf(f,"%lf,%lf ",&re,&im)!=2)  
{return 1;}  
  
}return 0;  
}
```

```
void insert_file(FILE *f,fftw_complex *a,int n0,int n1,int  
n)  
{int b,c;double d;  
for(b=0;b<n0;b++)  
{  
for(c=0;c<n1;c++)
```

```

    {fscanf(f,"%lf ",&d);a[b*n+c]=d;}
}
}

```

```

void insert_file_complex(FILE *f,fftw_complex *a,int
n0,int n1,int n)
{int b,c;double re,im;
for(b=0;b<n0;b++)
{
    for(c=0;c<n1;c++)
    {fscanf(f,"%lf,%lf ",&re,&im);a[b*n+c]=re+im*I;}
}
}

```

```

void zero(fftw_complex *a, int n0,int n1,int m,int n)
//gemisma me midenika ta upoloipa stoixeia tou pinaka
{int i,j;
for(i=0;i<n0;i++)
{
    for (j=n1;j<n;j++)
        {a[i*n+j]=0+0*I;}
}
for (i=(n0*n);i<(m*n);i++) {a[i]=0+0*I;}
}

```

```

int input_dim() //diabasma kai epali8eusi enos akeraiou apo
to xristi
{int a1,a2;
a2=scanf("%d",&a1);getchar();
if (a2!=1) printf("prepei na einai akeraios ari8mos\n");
while(a2!=1)

```

```

{a2=scanf("%d",&a1);getchar();
}
while (a1<=0)
{printf("prepei na einai 8etikos ari8mos\n");
a1=input_dim();
}

return a1;
}

```

```

void *execute(void *threadarg){//i sunartisi pou ekteloun ta
threads
struct thread_data *data;
data=(struct thread_data *) threadarg;
fftw_execute(data->plan);// i sunartisi tis biblio8ikis fftw3.h
pou ektelei to plano (plano 8a eksgi8ei parakato)
pthread_exit(NULL);}// eksodos tou thread to bazoume gia
asfaleia

```

```

int main(int argc,char *argv[])
{time_t start1,end;FILE *f,*fa,*fb;
pthread_t threads[NUM_THREADS];//orizoume ta threads
pthread_attr_t attr;//orizoume oti ta threads 8a exoun kapia
attributes
int n0,n1,n2,n3,i,rc,lim,error,n1_aux,lim1,m,n;
double dif;char c,choise='*';
fftw_complex *a,*A,*b,*B,*y,*Y;//edw einai oi pinakes
pou 8a ftiaksoume (pinakas--F-->PINAKAS parakatw 8a
dilonei metasximatismo)

```



```
fftw_plan plan_a,plan_b,plan_y;//edw orizoume ta plana
void *status;//ena orisma pou xreiazete i sunartisi
pthread_join tis pthread.h
pthread_attr_init(&attr);
pthread_attr_setdetachstate(&attr,
PTHREAD_CREATE_JOINABLE);//ftiaxnoume to
attribute twon thread na einai joinable diladi to
//programa na
proxorisei afou teliwsoun prota ta threads
```

```
while(choise!='q')
{printf("(a) dimiourgia input files\n");
printf("(r) random complex\n");
printf("(s) random integer\n");
printf("(f) random real\n");
printf("(p) diabasma apo arxeia real\n");
printf("(c) diabasma apo arxeia complex\n");
printf("(q) quit\n");
scanf("%c",&choise);
getchar();
if(choise=='a')
{printf("\n");
if((f=fopen("input_a","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
fclose(f);printf("dimiourgia input_a\n");
if((f=fopen("input_a_complex","w"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}
fclose(f);printf("dimiourgia input_a_complex\n");
if((f=fopen("input_b","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
fclose(f);printf("dimiourgia input_b\n");
```

```

if((f=fopen("input_b_complex","w"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}
fclose(f);printf("dimiourgia input_b_complex\n\n");
}
if((choise=='r')||(choise=='s')||(choise=='p')||(choise=='c')||
(choise=='f'))
{
if (choise=='r')
// EISODOS RAND complex
{
printf("\n dose diastaseis n0xn1 n2xn3\n");
printf(" n0=");n0=input_dim();
printf(" n1=");n1=input_dim();
printf(" n2=");n2=input_dim();
printf(" n3=");n3=input_dim();
printf(" dose orio tuxaion ari8mon ");
lim=input_dim();

m=n0+n2-1;
n=n1+n3-1;

a= (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to
pinaka a pou 8a ginei o metasximatismos
A = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to
pinaka A pou einai o metasximatismenos
//exei xreisimopoi8ei
to fftw_malloc tis fftw3.h ka8ws to xirizonte
//kalitera oi sunartiseis
tis fftw3.h

```

```
plan_a = fftw_plan_dft_2d(m,n, a, A, FFTW_FORWARD,
FTTW_ESTIMATE);//ftiaxnoume to plano me orismata n0
n1 oi diastaseis
```

```
//a h eisodos, A h
eksodos kai FFTW_FORWARD o eu8us metasximatismos
//etoimasame dld
na ginei to a--F-->A, DEN exei ginei akoma
```

```
b = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai
edw
B = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);
plan_b = fftw_plan_dft_2d(m,n, b, B, FFTW_FORWARD,
FTTW_ESTIMATE);
```

```
y= (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai
edw me prosoxi omos Y eisodo kai y eksodo
Y = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//kai
FTTW_BACKWARD gia na ektelestei o antistrofos
plan_y = fftw_plan_dft_2d(m,n, Y, y,
FTTW_BACKWARD, FFTW_ESTIMATE);//Y--F(-1)-->y
```

```
insert(a,b,n0,n1,n2,n3,lim,n);//bazoume stoixia ston pinaka
a kai ston b
printf("\nprinting a.....\n");
if((f=fopen("rand_a","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
print(a,n0,n1,n,f);//ektiposi a
```

```
fclose(f);
printf("\nprinting b.....\n");
if((f=fopen("rand_b","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
print(b,n2,n3,n,f);//ektiposi b
fclose(f);
```

```
zero(a,n0,n1,m,n);
zero(b,n2,n3,m,n);
```

```
choise='r';
// END EISODOS RAND complex
}
if (choise=='s')
{
// EISODOS RAND integer
```

```
printf("\n dose diastaseis n0xn1 n2xn3\n");
printf(" n0=");n0=input_dim();
printf(" n1=");n1=input_dim();
printf(" n2=");n2=input_dim();
printf(" n3=");n3=input_dim();
printf(" dose orio tuxaion ari8mon ");
lim=input_dim();
```

```
m=n0+n2-1;
n=n1+n3-1;
```

```
a= (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to
pinaka a pou 8a ginei o metasximatismos
```

```

A = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to
pinaka A pou einai o metasximatismenos
//exei xreisimopoi8ei
to fftw_malloc tis fftw3.h ka8ws to xirizonte
//kalitera oi sunartiseis
tis fftw3.h
plan_a = fftw_plan_dft_2d(m,n, a, A, FFTW_FORWARD,
FFTW_ESTIMATE);//ftiaxnoume to plano me orismata n0
n1 oi diastaseis
//a h eisodos, A h
eksodos kai FFTW_FORWARD o eu8us metasximatismos
//etoimasame dld
na ginei to a--F-->A, DEN exei ginei akoma

```

```

b = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai
edw
B = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);
plan_b = fftw_plan_dft_2d(m,n, b, B, FFTW_FORWARD,
FFTW_ESTIMATE);

```

```

y= (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai
edw me prosoxi omos Y eisodo kai y eksodo
Y = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//kai
FFTW_BACKWARD gia na ektelestei o antistrofos
plan_y = fftw_plan_dft_2d(m,n, Y, y,
FFTW_BACKWARD, FFTW_ESTIMATE);//Y--F(-1)-->y

```

```

insert_int(a,b,n0,n1,n2,n3,lim,n); //bazoume stoixia ston
pinaka a kai ston b
printf("\nprinting a.....\n");
if((f=fopen("rand_a","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
print_real(a,n0,n1,n,f); //ektiposi a
fclose(f);
printf("\nprinting b.....\n");
if((f=fopen("rand_b","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
print_real(b,n2,n3,n,f); //ektiposi b
fclose(f);

```

```

zero(a,n0,n1,m,n);
zero(b,n2,n3,m,n);

```

```

choise='s';
// END EISODOS RAND integer
}

```

```

if (choise=='f')
{
// EISODOS RAND real

```

```

printf("\n dose diastaseis n0xn1 n1xn2\n");
printf(" n0="); n0=input_dim();
printf(" n1="); n1=input_dim();
printf(" n2="); n2=input_dim();

```

```

printf(" n3=");n3=input_dim();
printf(" dose orio tuxaion ari8mon ");
lim=input_dim();

m=n0+n2-1;
n=n1+n3-1;

a= (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to
pinaka a pou 8a ginei o metasximatismos
A = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to
pinaka A pou einai o metasximatismenos
//exei xreisimopoi8ei
to fftw_malloc tis fftw3.h ka8ws to xirizonte
//kalitera oi sunartiseis
tis fftw3.h
plan_a = fftw_plan_dft_2d(m,n, a, A, FFTW_FORWARD,
FFTW_ESTIMATE);//ftiaxnoume to plano me orismata n0
n1 oi diastaseis
//a h eisodos, A h
eksodos kai FFTW_FORWARD o eu8us metasximatismos
//etoimasame dld
na ginei to a--F-->A, DEN exei ginei akoma

b = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai
edw
B = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);
plan_b = fftw_plan_dft_2d(m,n, b, B, FFTW_FORWARD,

```

```
FFTW_ESTIMATE);
```

```
y= (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai  
edw me prosoxi omos Y eisodo kai y eksodo  
Y = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//kai  
FFTW_BACKWARD gia na ektelestei o antistrofos  
plan_y = fftw_plan_dft_2d(m,n, Y, y,  
FFTW_BACKWARD, FFTW_ESTIMATE);//Y--F(-1)-->y
```

```
insert_real(a,b,n0,n1,n2,n3,lim,n);//bazoume stoixia ston  
pinaka a kai ston b  
printf("\nprinting a.....\n");  
if((f=fopen("rand_a","w"))==NULL){printf("ERROR  
OPENING FILE");exit(1);}  
print_real(a,n0,n1,n,f);//ektiposi a  
fclose(f);  
printf("\nprinting b.....\n");  
if((f=fopen("rand_b","w"))==NULL){printf("ERROR  
OPENING FILE");exit(1);}  
print_real(b,n2,n3,n,f);//ektiposi b  
fclose(f);
```

```
zero(a,n0,n1,m,n);  
zero(b,n2,n3,m,n);
```

```
choise='s';  
// END EISODOS RAND real
```



```
}
```

```
if (choise=='p')
```

```
// EISODOS FILE real
```

```
{
```

```
    if((f=fopen("input_a","r"))==NULL)
```

```
// elenxos an uparxei to arxeio
```

```
    {printf("den brethike to arxeio 'input_a' 8a  
dimiourgi8ei kainourio\n");
```

```
        if((f=fopen("input_a","w"))==NULL){printf("ERROR  
OPENING FILE\n");exit(1);}
        fclose(f);
```

```
        if((f=fopen("input_a","r"))==NULL){printf("ERROR  
OPENING FILE\n");exit(1);}
    }
```

```
error=error_real(f);
```

```
fclose(f);
```

```
while(error==1)
```

```
//elenxos sta dedomena tou arxeiou
```

```
{printf(" la8os sta dedomena tou arxeiou 'input_a' dior8oste  
kai meta patiste enter....\n");
```

```
getchar();
```

```
if((f=fopen("input_a","r"))==NULL){printf("ERROR  
OPENING FILE\n");exit(1);}
error=error_real(f);
```

```
fclose(f);
```

```
}
```

```
if((f=fopen("input_b","r"))==NULL)
```

```

// elenxos an uparxei to arxeio
    {printf("den brethike to arxeio 'input_b' 8a
dimiourgi8ei kainourio\n");
    if((f=fopen("input_b","w"))==NULL){printf("ERROR
OPENING FILE\n");exit(1);}
    fclose(f);
    if((f=fopen("input_b","r"))==NULL){printf("ERROR
OPENING FILE\n");exit(1);}
    }
error=error_real(f);
fclose(f);

while(error==1)
//elenxos sta dedomena tou arxeiou
{printf(" la8os sta dedomena tou arxeiou 'input_b' dior8oste
kai meta patiste enter....\n");
getchar();
if((f=fopen("input_b","r"))==NULL){printf("ERROR
OPENING FILE\n");exit(1);}
error=error_real(f);
fclose(f);
}

if((f=fopen("input_a","r"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
lim=plithos(f); //briskoume to plithos tw n
stoixeiwn tou pinaka a
fclose(f);

if((f=fopen("input_a","r"))==NULL){printf("ERROR
OPENING FILE");exit(1);}n0=0;

```

```

while(!feof(f))
{c = fgetc(f);
  if(c == '\n')
    {n0++;} //diabazei ka8e seira etsi
mporoume na metrisoume tis seires
}
fclose(f);

n1=lim/n0; //etsi oi stoiles tou a einai oi
pli8os/grammes

if((f=fopen("input_b","r"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
lim=plithos(f); //briskoume to plithos tw n
stoixeiwn tou pinaka b
fclose(f);

if((f=fopen("input_b","r"))==NULL){printf("ERROR
OPENING FILE");exit(1);}n1_aux=0;
while(!feof(f))
{c = fgetc(f);
  if(c == '\n')
    {n1_aux++;} //diabazei ka8e seira etsi
mporoume na metrisoume tis seires
}
fclose(f);

n2=lim/n1_aux; //etsi oi stoiles tou b
einai oi pli8os/grammes

```

```
printf("\n oi diastaseis einai %dx%d kai %dx  
%d\n",n0,n1,n1_aux,n2);
```

```
m=n0+n1_aux-1;  
n=n1+n2-1;
```

```
printf("\n oi diastaseis tis suneliksisis einai %dx%d\n",m,n);
```

```
a= (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to  
pinaka a pou 8a ginei o metasximatismos  
A = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to  
pinaka A pou einai o metasximatismenos  
//exei xreisimopoi8ei  
to fftw_malloc tis fftw3.h ka8ws to xirizonte  
//kalitera oi sunartiseis  
tis fftw3.h  
plan_a = fftw_plan_dft_2d(m,n, a, A, FFTW_FORWARD,  
FFTW_ESTIMATE);//ftiaxnoume to plano me orismata n0  
n1 oi diastaseis  
//a h eisodos, A h  
eksodos kai FFTW_FORWARD o eu8us metasximatismos  
//etoimasame dld  
na ginei to a--F-->A, DEN exei ginei akoma
```

```
b = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai  
edw  
B = (fftw_complex*)
```

```
fftw_malloc(sizeof(fftw_complex)*m*n);  
plan_b = fftw_plan_dft_2d(m,n, b, B, FFTW_FORWARD,  
FFTW_ESTIMATE);
```

```
y= (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai  
edw me prosoxi omos Y eisodo kai y eksodo  
Y = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//kai  
FFTW_BACKWARD gia na ektelestei o antistrofos  
plan_y = fftw_plan_dft_2d(m,n, Y, y,  
FFTW_BACKWARD, FFTW_ESTIMATE);//Y--F(-1)-->y
```

```
if((f=fopen("input_a","r"))==NULL){printf("ERROR  
OPENING FILE");exit(1);}  
insert_file(f,a,n0,n1,n);          //eisagogi a  
fclose(f);
```

```
if((f=fopen("input_b","r"))==NULL){printf("ERROR  
OPENING FILE");exit(1);}  
insert_file(f,b,n1_aux,n2,n);      //isagogi b  
fclose(f);  
choise='p';
```

```
zero(a,n0,n1,m,n);  
zero(b,n1_aux,n2,m,n);
```

```
// END EISODOS FILE real  
}
```

```

if(choise=='c')
// EISODOS FILE complex
{
    if((f=fopen("input_a_complex","r"))==NULL)
// elenxos an uparxei to arxeio
    {printf("den brethike to arxeio 'input_a_complex' 8a
dimiourgi8ei kainourio\n");
    if((f=fopen("input_a_complex","w"))==NULL)
{printf("ERROR OPENING FILE\n");exit(1);}
    fclose(f);
    if((f=fopen("input_a_complex","r"))==NULL)
{printf("ERROR OPENING FILE\n");exit(1);}
    }
error=error_complex(f);
fclose(f);

while(error==1)
//elenxos sta dedomena tou arxeiou
{printf(" la8os sta dedomena tou arxeiou 'input_a_complex'
dior8oste kai meta patiste enter....\n");
getchar();
if((f=fopen("input_a_complex","r"))==NULL)
{printf("ERROR OPENING FILE\n");exit(1);}
error=error_complex(f);
fclose(f);
}

    if((f=fopen("input_b_complex","r"))==NULL)
// elenxos an uparxei to arxeio
    {printf("den brethike to arxeio 'input_b_complex' 8a
dimiourgi8ei kainourio\n");

```

```

        if((f=fopen("input_b_complex","w"))==NULL)
{printf("ERROR OPENING FILE\n");exit(1);}
        fclose(f);
        if((f=fopen("input_b_complex","r"))==NULL)
{printf("ERROR OPENING FILE\n");exit(1);}
    }
    error=error_complex(f);
    fclose(f);

while(error==1)                                     //elenxos
sta dedomena tou arxeiou
{printf(" la8os sta dedomena tou arxeiou 'input_b_complex'
dior8oste kai meta patiste enter...\n");
getchar();
if((f=fopen("input_b_complex","r"))==NULL)
{printf("ERROR OPENING FILE\n");exit(1);}
error=error_complex(f);
fclose(f);
}

if((f=fopen("input_a_complex","r"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}
lim=plithos_complex(f);                             //briskoume to
plithos twn stoixeiwn tou pinaka a
fclose(f);

if((f=fopen("input_a_complex","r"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}n0=0;
while(!feof(f))
{c = fgetc(f);

```

```

    if(c == '\n')
        {n0++;}                //diabazei ka8e seira etsi
mporoume na metrisoume tis seires
}
fclose(f);

n1=lim/n0;                    //etsi oi stoiles tou a einai oi
pli8os/grammes

if((f=fopen("input_b_complex","r"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}
lim=plithos_complex(f);        //briskoume to
plithos tw n stoixeiwn tou pinaka b
fclose(f);

if((f=fopen("input_b_complex","r"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}n1_aux=0;
while(!feof(f))
{ c = fgetc(f);
    if(c == '\n')
        {n1_aux++;}            //diabazei ka8e seira etsi
mporoume na metrisoume tis seires

}
fclose(f);

n2=lim/n1_aux;                //etsi oi stoiles tou b
einai oi pli8os/grammes

```



```
printf("\n oi diastaseis einai %dx%d kai %dx  
%d\n",n0,n1,n1_aux,n2);
```

```
m=n0+n1_aux-1;  
n=n1+n2-1;
```

```
printf("\n oi diastaseis tis suneliksisis einai %dx%d\n",m,n);
```

```
a= (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to  
pinaka a pou 8a ginei o metasximatismos
```

```
A = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//ftiaxnoume to  
pinaka A pou einai o metasximatismenos
```

```
//exei xreisimopoi8ei  
to fftw_malloc tis fftw3.h ka8ws to xirizonte  
//kalitera oi sunartiseis  
tis fftw3.h
```

```
plan_a = fftw_plan_dft_2d(m,n, a, A, FFTW_FORWARD,  
FFTW_ESTIMATE);//ftiaxnoume to plano me orismata n0  
n1 oi diastaseis
```

```
//a h eisodos, A h  
eksodos kai FFTW_FORWARD o eu8us metasximatismos  
//etoimasame dld  
na ginei to a--F-->A, DEN exei ginei akoma
```

```
b = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai  
edw  
B = (fftw_complex*)  
fftw_malloc(sizeof(fftw_complex)*m*n);
```

```
plan_b = fftw_plan_dft_2d(m,n, b, B, FFTW_FORWARD,
FFTW_ESTIMATE);
```

```
y= (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//antistoixa kai
edw me prosoxi omos Y eisodo kai y eksodo
Y = (fftw_complex*)
fftw_malloc(sizeof(fftw_complex)*m*n);//kai
FFTW_BACKWARD gia na ektelestei o antistrofos
plan_y = fftw_plan_dft_2d(m,n, Y, y,
FFTW_BACKWARD, FFTW_ESTIMATE);//Y--F(-1)-->y
```

```
if((f=fopen("input_a_complex","r"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}
insert_file_complex(f,a,n0,n1,n);          //eisagogi a
fclose(f);
```

```
if((f=fopen("input_b_complex","r"))==NULL)
{printf("ERROR OPENING FILE");exit(1);}
insert_file_complex(f,b,n1_aux,n2,n);      //eisagogi
b
fclose(f);
```

```
zero(a,n0,n1,m,n);
zero(b,n1_aux,n2,m,n);
```

```
choise='c';
```

```
// END EISODOS FILE complex
}
```

```
clock_t start=clock();  
//fftw_execute(plan_a);// αυτο θα εκτελούσαμε xwis  
threads για να βρούμε το a--F-->A  
//fftw_execute(plan_b); //και b--F-->B αντίστοιχα
```

```
thread_data_array[0].plan=plan_a;// το πρώτο του προγράμματος  
thread είναι το πρώτο του a  
thread_data_array[1].plan=plan_b;//το δεύτερο του πρώτου  
του b  
rc=pthread_create(&threads[0],&attr,execute,(void *)  
&thread_data_array[0]);//φτιάχνουμε ένα thread που εκτελεί  
το a--F-->A  
rc=pthread_create(&threads[1],&attr,execute,(void *)  
&thread_data_array[1]);//και ένα που εκτελεί το b--F-->B  
pthread_attr_destroy(&attr);  
rc=pthread_join(threads[0],&status);  
rc=pthread_join(threads[1],&status);// τελειώνουν τα thread
```

```
mult(Y,A,B,m,n);//υπολογίζουμε το Y που είναι ο  
πολυπλάσιμος σίμων A*B  
fftw_execute(plan_y);//εκτελούμε το Y--F(-1)-->y  
normalize_2D(y,m,n);//διαίρουμε τα πάντα με m*n  
clock_t end1=clock();float seconds=(float)(end1-  
start)/CLOCKS_PER_SEC;  
printf("\nprinting y.....\n");
```

```
if((choise=='r')||(choise=='c'))
{
if((f=fopen("output","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
print(y,m,n,n,f);// kai ektuponoume to y pou einai pleon i
suneliksi tou a me to b
fclose(f);
}
```

```
if((choise=='s')||(choise=='p'))
{
if((f=fopen("output","w"))==NULL){printf("ERROR
OPENING FILE");exit(1);}
print_real(y,m,n,n,f);// kai ektuponoume to y pou einai
pleon i suneliksi tou a me to b
fclose(f);
}
```

```
fftw_destroy_plan(plan_a);
fftw_destroy_plan(plan_b);
fftw_destroy_plan(plan_y);
fftw_free(a);fftw_free(A);
fftw_free(b);fftw_free(B);
fftw_free(y);fftw_free(Y);//sbinoume oti exoume
dimiourgisei me tis sunartuseis tis fftw3.h
printf("xronos ektelesis suneliksis %f
\n",seconds);choise='*';

}
}
pthread_exit(NULL);// eksodos asfaleias twn thread
```

}