



ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

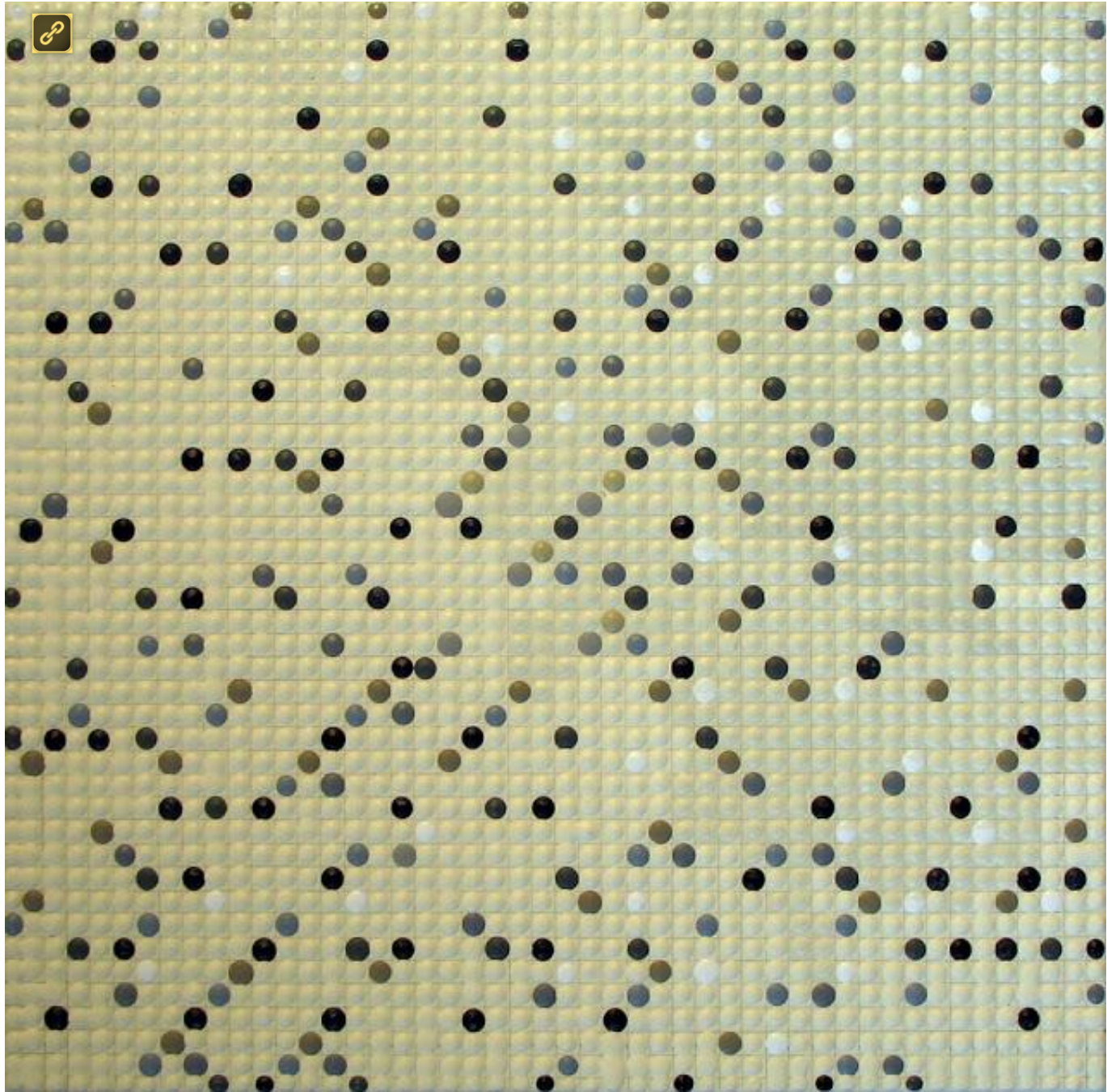


# Εισαγωγή στον Προγραμματισμό Η/Υ για Χημικούς Μηχανικούς

Παρουσίαση Διαλέξεων: 7. Τμήματα Πινάκων  
Καθηγητής Δημήτρης Ματαράς



Copyright © 2014 by Prof. D. S. Mataras ([mataras@upatras.gr](mailto:mataras@upatras.gr)). This work is made available under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license, <http://creativecommons.org/licenses/by-nc-nd/3.0/>



# Τμήματα πινάκων, ορισμοί

---

**ΤΜΗΜΑ ΠΙΝΑΚΑ:** Πίνακας που είναι υποσύνολο του αρχικού και μπορεί να περιγραφεί είτε με *τριάδες ενδεικτών* (αν είναι κανονικό) είτε με *διάνυσμα ενδεικτών*.

**ΤΡΙΑΔΑ :** περιγράφει την έκταση μιας διάστασης του τμήματος με την μορφή:  $\text{όριο}_1 : \text{όριο}_2 : \text{βήμα}$

*όπου :*  $\text{όριο}_1, \text{όριο}_2$  οι ενδείκτες που καθορίζουν την έκταση του τμήματος και  $\text{βήμα}$  η σειρά διαδοχής των ενδεικτών

*αν :* αν κάποιο από τα  $\text{όριο}_1, \text{όριο}_2$  συμπίπτει με τα όρια του αρχικού πίνακα μπορεί να παραληφθεί

*αν :* το  $\text{βήμα} = 1$  μπορεί να παραληφθεί

**ΔΙΑΝΥΣΜΑ :** Ακέραιος μονοδιάστατος πίνακας που περιέχει τους ενδείκτες των στοιχείων του αρχικού πίνακα που συναποτελούν το τμήμα.

# Παραδείγματα τμημάτων

με τριάδες και διανύσματα

**REAL** :: a(5) = 0, x(6,6) = 1

**INTEGER** :: b(3) = [2,3,5]

a = 1.

a(:) = 1. !ακριβώς το ίδιο

a(1:4:1) = 2.

a(:4) = 2. !ακριβώς το ίδιο

a	a(1)	a(2)	a(3)	a(4)	a(5)
	2.000000	2.000000	2.000000	2.000000	1.000000

a(b) = 9.9 !διανυσματικός

a	a(1)	a(2)	a(3)	a(4)	a(5)
	2.000000	9.900000	9.900000	2.000000	9.900000

x(3:4,3:4) = 2.

x(2,:) = 3.

x(:,6) = 4.

**END**

$$\mathbf{x}(3:4,3:4) = 2.$$

<b>x</b>	$x(1,1)$	$x(1,2)$	$x(1,3)$	$x(1,4)$	$x(1,5)$	$x(1,6)$
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	$x(2,1)$	$x(2,2)$	$x(2,3)$	$x(2,4)$	$x(2,5)$	$x(2,6)$
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	$x(3,1)$	$x(3,2)$	<b><math>x(3,3)</math></b>	<b><math>x(3,4)</math></b>	$x(3,5)$	$x(3,6)$
	1.000000	1.000000	<b>2.000000</b>	<b>2.000000</b>	1.000000	1.000000
	$x(4,1)$	$x(4,2)$	<b><math>x(4,3)</math></b>	<b><math>x(4,4)</math></b>	$x(4,5)$	$x(4,6)$
	1.000000	1.000000	<b>2.000000</b>	<b>2.000000</b>	1.000000	1.000000
	$x(5,1)$	$x(5,2)$	$x(5,3)$	$x(5,4)$	$x(5,5)$	$x(5,6)$
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	$x(6,1)$	$x(6,2)$	$x(6,3)$	$x(6,4)$	$x(6,5)$	$x(6,6)$
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

$$\mathbf{x}(2,:) = 3.; \quad \mathbf{x}(:,6) = 4.$$

<b>x</b>	$x(1,1)$	$x(1,2)$	$x(1,3)$	$x(1,4)$	$x(1,5)$	<b><math>x(1,6)</math></b>
	1.000000	1.000000	1.000000	1.000000	1.000000	<b>4.000000</b>
	<b><math>x(2,1)</math></b>	<b><math>x(2,2)</math></b>	<b><math>x(2,3)</math></b>	<b><math>x(2,4)</math></b>	<b><math>x(2,5)</math></b>	<b><math>x(2,6)</math></b>
	<b>3.000000</b>	<b>3.000000</b>	<b>3.000000</b>	<b>3.000000</b>	<b>3.000000</b>	<b>4.000000</b>
	$x(3,1)$	$x(3,2)$	$x(3,3)$	$x(3,4)$	$x(3,5)$	<b><math>x(3,6)</math></b>
	1.000000	1.000000	2.000000	2.000000	1.000000	<b>4.000000</b>
	$x(4,1)$	$x(4,2)$	$x(4,3)$	$x(4,4)$	$x(4,5)$	<b><math>x(4,6)</math></b>
	1.000000	1.000000	2.000000	2.000000	1.000000	<b>4.000000</b>
	$x(5,1)$	$x(5,2)$	$x(5,3)$	$x(5,4)$	$x(5,5)$	<b><math>x(5,6)</math></b>
	1.000000	1.000000	1.000000	1.000000	1.000000	<b>4.000000</b>
	$x(6,1)$	$x(6,2)$	$x(6,3)$	$x(6,4)$	$x(6,5)$	<b><math>x(6,6)</math></b>
	1.000000	1.000000	1.000000	1.000000	1.000000	<b>4.000000</b>



**REAL**::**x**(6,6) = 1.; **x**(:,:,2) = 2.0

<b>X</b>	<b>x</b> (1,1)	x(1,2)	<b>x</b> (1,3)	x(1,4)	<b>x</b> (1,5)	x(1,6)
	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000
	<b>x</b> (2,1)	x(2,2)	<b>x</b> (2,3)	x(2,4)	<b>x</b> (2,5)	x(2,6)
	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000
	<b>x</b> (3,1)	x(3,2)	<b>x</b> (3,3)	x(3,4)	<b>x</b> (3,5)	x(3,6)
	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000
	<b>x</b> (4,1)	x(4,2)	<b>x</b> (4,3)	x(4,4)	<b>x</b> (4,5)	x(4,6)
	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000
	<b>x</b> (5,1)	x(5,2)	<b>x</b> (5,3)	x(5,4)	<b>x</b> (5,5)	x(5,6)
	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000
	<b>x</b> (6,1)	x(6,2)	<b>x</b> (6,3)	x(6,4)	<b>x</b> (6,5)	x(6,6)
	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000

**REAL**::**x**(6,6) = 1.; **x**(2::2,2::2) = 2.0

<b>X</b>	x(1,1)	x(1,2)	x(1,3)	x(1,4)	x(1,5)	x(1,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(2,1)	<b>x</b> (2,2)	x(2,3)	<b>x</b> (2,4)	x(2,5)	<b>x</b> (2,6)
	1.000000	2.000000	1.000000	2.000000	1.000000	2.000000
	x(3,1)	x(3,2)	x(3,3)	x(3,4)	x(3,5)	x(3,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(4,1)	<b>x</b> (4,2)	x(4,3)	<b>x</b> (4,4)	x(4,5)	<b>x</b> (4,6)
	1.000000	2.000000	1.000000	2.000000	1.000000	2.000000
	x(5,1)	x(5,2)	x(5,3)	x(5,4)	x(5,5)	x(5,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(6,1)	<b>x</b> (6,2)	x(6,3)	<b>x</b> (6,4)	x(6,5)	<b>x</b> (6,6)
	1.000000	2.000000	1.000000	2.000000	1.000000	2.000000

**REAL**::x(6,6) = 1.; x(:,:,3,2::2) = 2.0

<b>X</b>	x(1,1)	<b>x(1,2)</b>	x(1,3)	<b>x(1,4)</b>	x(1,5)	<b>x(1,6)</b>
	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>
	x(2,1)	x(2,2)	x(2,3)	x(2,4)	x(2,5)	x(2,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(3,1)	x(3,2)	x(3,3)	x(3,4)	x(3,5)	x(3,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(4,1)	<b>x(4,2)</b>	x(4,3)	<b>x(4,4)</b>	x(4,5)	<b>x(4,6)</b>
	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>
	x(5,1)	x(5,2)	x(5,3)	x(5,4)	x(5,5)	x(5,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(6,1)	x(6,2)	x(6,3)	x(6,4)	x(6,5)	x(6,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

**REAL**::x(6,6) = 1.; **INTEGER**::k = 2; x(k+1:,k-1::k) = 2.0

<b>X</b>	x(1,1)	x(1,2)	x(1,3)	x(1,4)	x(1,5)	x(1,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	x(2,1)	x(2,2)	x(2,3)	x(2,4)	x(2,5)	x(2,6)
	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
	<b>x(3,1)</b>	x(3,2)	<b>x(3,3)</b>	x(3,4)	<b>x(3,5)</b>	x(3,6)
	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000
	<b>x(4,1)</b>	x(4,2)	<b>x(4,3)</b>	x(4,4)	<b>x(4,5)</b>	x(4,6)
	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000
	<b>x(5,1)</b>	x(5,2)	<b>x(5,3)</b>	x(5,4)	<b>x(5,5)</b>	x(5,6)
	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000
	<b>x(6,1)</b>	x(6,2)	<b>x(6,3)</b>	x(6,4)	<b>x(6,5)</b>	x(6,6)
	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000	<b>2.000000</b>	1.000000

# Πράξεις με πίνακες

```
real      :: a(5) = 2., b(5,3) = 1.  
integer  :: i
```

```
b(:,1) = a           !έχουν την ίδια μορφή 1x5  
b(:,2) = a + b(:,1)
```

```
do i = 1, size(b,1)  
    b(:,3) = sum(b(i,:2))  
    print*,b (i,:) !τυπώνει μια γραμμή κάθε φορά  
enddo
```

```
end
```

```
2.00000000  4.00000000  6.00000000  
2.00000000  4.00000000  6.00000000  
2.00000000  4.00000000  6.00000000  
2.00000000  4.00000000  6.00000000  
2.00000000  4.00000000  6.00000000
```



Τμήματα πινάκων ή ολόκληροι πίνακες που συμμετέχουν στην ίδια έκφραση **πρέπει να έχουν ακριβώς την ίδια μορφή!**



# Πράξεις με πίνακες παραλλαγή

με μεταβλητό format

```
real          :: a(5) = 2., b(5,3) = 1.  
integer      :: i  
character(50) :: string
```

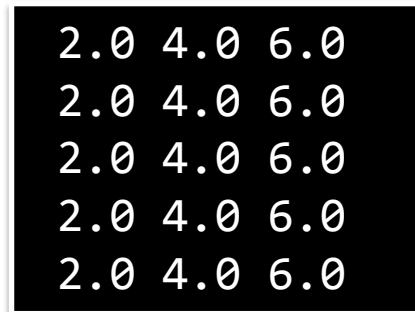
```
b(:,1) = a      !έχουν την ίδια μορφή 1x5
```

```
b(:,2) = a + b(:,1)
```

```
write(string,*) '(' , size(a) , 'f4.1) ' !μεταβλητό format
```

```
do i = 1, size(b,1)  
    b(:,3) = sum(b(i,:2))  
    print string, b(i,:)   
enddo
```

```
end
```



2.0	4.0	6.0
2.0	4.0	6.0
2.0	4.0	6.0
2.0	4.0	6.0
2.0	4.0	6.0

# Οι κατασκευαστές στις 2 διαστάσεις

```
real :: x(2,3) = 0.0
```

```
x(1,:) = [1.,4.,8.]
```

```
x(2,:) = [2.,6.,9.]
```

```
do i =1, size(x,1)
```

```
    print *, x(i,:) ! τυπώνει τον πίνακα ανά γραμμή
```

```
end do
```

```
end
```

<b>x</b>	x(1,1)	x(1,2)	x(1,3)
	1.000000	4.000000	8.000000
	x(2,1)	x(2,2)	x(2,3)
	2.000000	6.000000	9.000000

# RESHAPE και καταχώριση στη μνήμη

```
real::x(2,3) = 0.0
```

```
x = reshape([1.,2.,4.,6.,8.,9.], [2,3])
```

```
do i = 1, 2
```

```
  print '(3f7.1)', x(i,:)
```

```
end do
```

```
print '(6f7.1)', x !ο πίνακας όπως καταχωρίζεται στη μνήμη
```

end

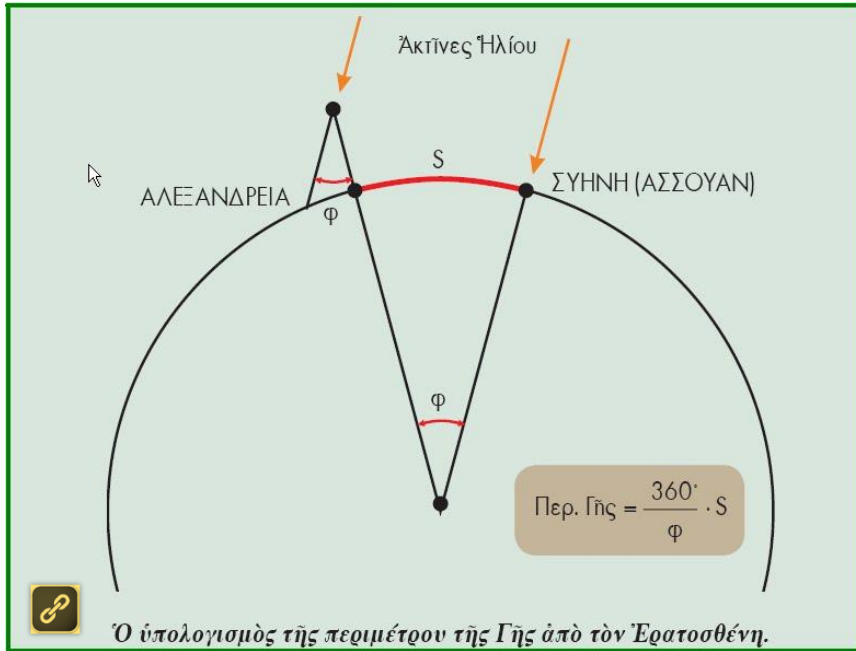
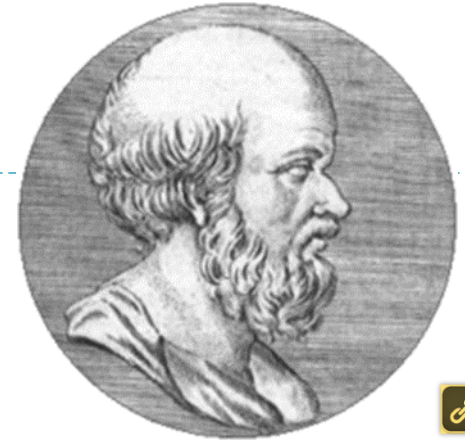
	x(1,1)	x(1,2)	x(1,3)
x	1.0000000	4.0000000	8.0000000
	x(2,1)	x(2,2)	x(2,3)
	2.0000000	6.0000000	9.0000000

1.0	4.0	8.0			
2.0	6.0	9.0			
1.0	2.0	4.0	6.0	8.0	9.0

Όπως βλέπετε τα στοιχεία του πίνακα καταχωρίζονται στη μνήμη **με προτεραιότητα στήλης** και όχι γραμμής.

# Ερατοσθένης

276-194 π.Χ.



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

# Το κόσκινο του Ερατοσθένη

---

```
program Eratosthenes
  integer,parameter :: max_nr = 100
  integer           :: nrs(max_nr), i

  call execute_command_line('chcp 1253')
  ! Δίνουμε αρχικές τιμές 0, 2, 3, ..., max_number
  nrs = [0, (i, i = 2 , max_nr)] ! με κατασκευαστή

  do i = 2, max_nr
    if(nrs(i) /= 0) then      ! Αν ο i είναι πρώτος
      nrs(2*i:max_nr:i) = 0 !μηδενίζουμε τα πολλαπλάσια του
    endif
  enddo

  print '(a40)', 'Υπάρχουν οι εξής πρώτοι αριθμοί:'
  do i = 2, max_nr
    if(nrs(i) /= 0) print *, nrs(i)
  enddo
end program Eratosthenes
```

# Το κόσκινο του Ερατοσθένη

με count και pack

```
program Eratosthenes
  integer,parameter :: max_nr = 100
  integer           :: nrs(max_nr), i, primes

  call execute_command_line('chcp 1253')
  ! Δίνουμε αρχικές τιμές 0, 2, 3, ..., max_number

  nrs = [0, (i, i = 2, max_nr)] ! με κατασκευαστή

  do i = 2, max_nr
    if(nrs(i) /= 0) then !Αν ο i είναι πρώτος
      nrs(2*i:max_nr:i)=0 !μηδενίζουμε τα πολλαπλάσια του
    endif
  enddo

  ! Μετράμε πόσοι είναι πρώτοι:
  primes = count(nrs /= 0)
  ! Τους μαζεύουμε στην αρχή του πίνακα:
  nrs(1:primes) = pack(nrs, nrs /= 0)
```



# Το κόσκινο του Ερατοσθένη

με count και pack

! και τους τυπώνουμε:

```
print ' (a9,i4,a18,i5) ', 'Υπάρχουν', primes, &  
      & 'πρώτοι αριθμοί <=', max_nr;
```

```
print ' (5i7) ', nrs(1:primes)
```

```
end program Eratosthenes
```

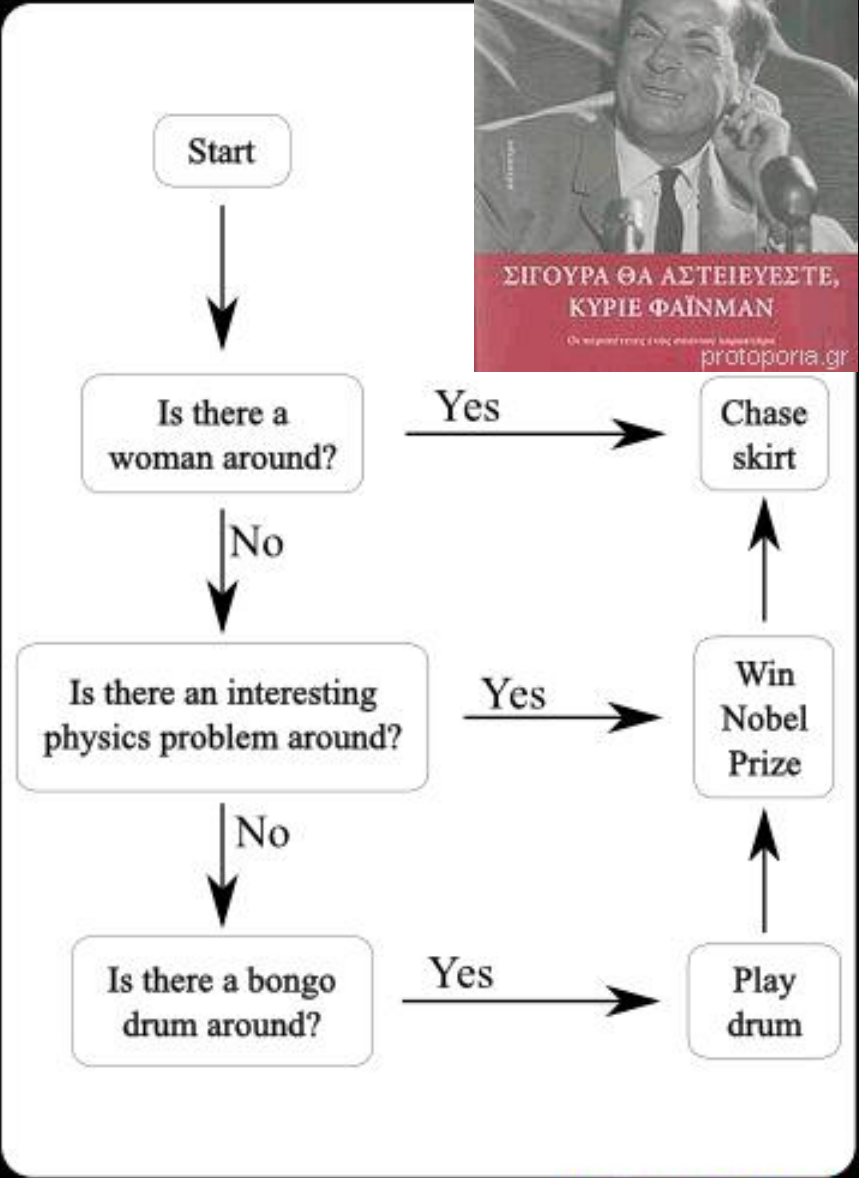
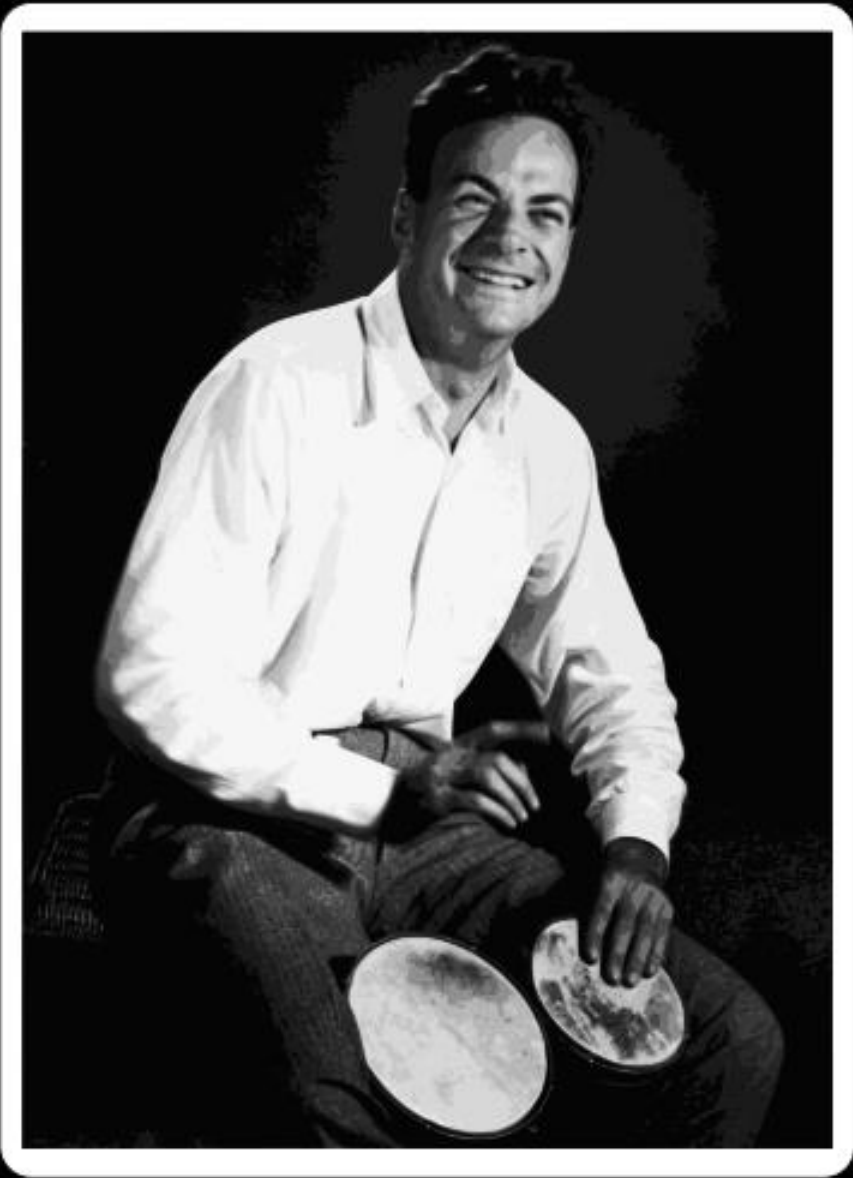
```
Υπάρχουν 25 πρώτοι αριθμοί <= 100  
  2      3      5      7      11  
 13     17     19     23     29  
 31     37     41     43     47  
 53     59     61     67     71  
 73     79     83     89     97
```



## Richard Feynman (1918 - 1988) CalTech

- Albert Einstein Award (1954)
- E. O. Lawrence Award (1962)
- Nobel Prize in Physics (1965)
- Oersted Medal (1972)
- National Medal of Science (1979)

# W.W.R.F.D?



# Παράλληλες δομές

## παράλληλη επιλογή `where`

**WHERE** (συνθήκη)

εκφράσεις1

[ **ELSEWHERE** [ (συνθήκη2) ] ]

εκφράσεις2

**END WHERE**

! Δομή **WHERE**

**WHERE** (συνθήκη) έκφραση

! πρόταση **WHERE**



Μια δομή **WHERE** μπορεί να περιέχει μόνο προτάσεις ανάθεσης τιμών του πίνακα ή προτάσεις και δομές **WHERE** ή **FORALL** που θα δούμε αμέσως. Δεν μπορεί για παράδειγμα να περιέχει μια πρόταση **PRINT**.



Στο **WHERE** η συνθήκη αφορά πάντα ένα πίνακα ενώ τα στοιχεία του πίνακα στα οποία γίνεται μεταβολή στις **εκφράσεις** αναπαρίστανται μόνο από το **όνομα του πίνακα**, χωρίς όρισμα και ενδείκτη.

# Παράλληλες δομές

`forall`: πολλαπλά DO + ένα IF

```
FORALL (τριάδα[, τριάδα] ... [, συνθήκη]) ! Δομή forall
```

εκφράσεις...

```
END FORALL
```

! ή πρόταση forall:

```
FORALL (τριάδα[, τριάδα] ... [, συνθήκη]) έκφραση
```

! όπου:

τριάδα: όνομα ενδείκτη = από:έως[:βήμα]

```
integer:: i; integer, dimension(4):: a = 0 , b = 0
```

```
a(4) = 4
```

```
forall (i = 1:3)
```

```
    a(i) = i; b(i) = a(i+1)
```

```
end forall
```

```
print '(4i4)', a; print '(4i4)', b
```

```
end
```

1	2	3	4
2	3	4	0

# Απλό παράδειγμα forall

```
program forall_use
  implicit none
  ! δηλώσεις:
  integer:: a(4,4) = 0, i, j

  ! αρχή:
  forall (i=1:size(a,1), j=1:size(a,2), i/=j)
    a(i,j) = i * 10 + j
  end forall

  !υπονοούμενη επανάληψη 2 διαστάσεων:
  print 10, ((a(i,j), j=1,size(a,2)), i=1,size(a,1))

10 format (4i5)

end program forall_use
```

0	12	13	14
21	0	23	24
31	32	0	34
41	42	43	0



# Εγγενείς συναρτήσεις ANY και ALL

συνδυάζονται με IF

**ANY**(μάσκα[, διάσταση])

Το Αποτέλεσμα = .TRUE. αν κάποιες τιμές του πίνακα (ή προαιρετικά μίας διάστασης του), πληρούν τις συνθήκες που καθορίζονται στη μάσκα.

**ALL**(μάσκα[, διάσταση])

Το Αποτέλεσμα = .TRUE. αν όλες οι τιμές του πίνακα (ή προαιρετικά μίας διάστασης του), πληρούν τις συνθήκες που καθορίζονται στη μάσκα.

```
real::a(4)=[-1,2,-3,6] !Παράδειγμα 7-34
if(all(a /=0)) a=1./a !αντικαθιστά όλα τα στοιχεία με τα
!αντίστροφα τους υπό την προϋπόθεση
!ότι κανένα δεν είναι μηδέν
if(any(a < 0)) a=abs(a) !αν υπάρχει αρνητικό στοιχείο
!όλα τα στοιχεία γίνονται θετικά
print *, a !1.00000000 0.500000000 0.333333343 0.166666672
end
```



- Spartacus (1960)
- Lolita (1962)
- Dr. Strangelove (1964)
- 2001: A Space Odyssey (1968)
- A Clockwork Orange (1971)
- Barry Lyndon (1975)
- The Shining (1980)
- Full Metal Jacket (1987)
- Eyes Wide Shut (1999)
- Work on A. I. Artificial Intelligence

# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Όνομα μέλους ή μελών ΔΕΠ 2014:  
Δημήτριος Ματαράς. «Εισαγωγή στον Προγραμματισμό Η/Υ». Έκδοση: 1.0.  
Πάτρα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:  
<https://eclass.upatras.gr/courses/CMNG2178>.

# Χρηματοδότηση

- ▶ Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- ▶ Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- ▶ Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.