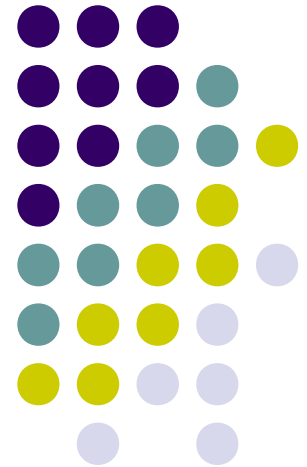
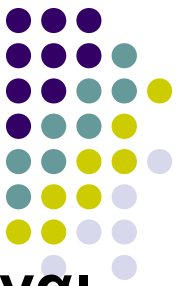


# Πίνακες

Ι.Ε.Κ ΓΛΥΦΑΔΑΣ  
Τεχνικός Τεχνολογίας Internet  
**Αλγοριθμική Ι (Ε)**  
Σχολ. Έτος 2012-13  
Α' Εξάμηνο

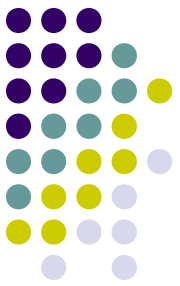




# Πίνακες

- Η πιο γνωστή και διαδομένη στατική δομή είναι ο πίνακας. Οι πίνακες αποτελούνται από στοιχεία του ίδιου τύπου. Η αναφορά στα στοιχεία του πίνακα γίνεται με τη χρήση του συμβολικού ονόματος του πίνακα το οποίο ακολουθείται από την τιμή ενός ή περισσότερων **δεικτών** σε παρένθεση ή αγκύλη. Ο δείκτης μας καθορίζει την θέση του στοιχείου στον πίνακα.
- Οι πίνακες που χρησιμοποιούν ένα δείκτη για την αναφορά των στοιχείων τους λέγονται μονοδιάστατοι.
- Παράδειγμα: Μονοδιάστατος πίνακας 9 στοιχείων

<b>A[1]</b>	<b>A[2]</b>	...						<b>A[9]</b>
17	2	3	1	5	9	4	11	7



# Μονοδιάστατοι πίνακες (1)

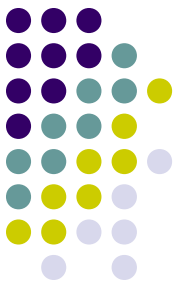
- Δημιουργία πίνακα
- Για να δημιουργήσουμε ένα πίνακα π.χ. 30 θέσεων από τα στοιχεία που δίνει ο χρήστης γράφουμε:

Για  $i$  από 1 μέχρι 30

Εμφάνισε 'Δώσε το ' $i$ ',' στοιχείο του πίνακα'

Διάβασε ΠΙΝ[ $i$ ]

Τέλος\_επανάληψης



## Μονοδιάστατοι πίνακες (2)

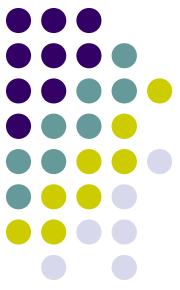
- Εμφάνιση στοιχείων πίνακα
- Για να εμφανίσουμε τα στοιχεία ενός πίνακα π.χ. 30 θέσεων γράφουμε:

Για  $i$  από 1 μέχρι 30

Εμφάνισε  $ΠΙΝ[i]$

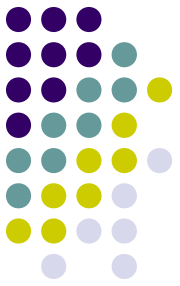
Τέλος\_επανάληψης

# Σειριακή αναζήτηση σε Μονοδιάστατο πίνακα



- Σε ένα μη ταξινομημένο πίνακα *table* αναζητούμε την τιμή *key*. Ο αλγόριθμος παίρνει ένα-ένα τα στοιχεία του πίνακα *table* και ελέγχει αν είναι αυτό που αναζητεί. Η διαδικασία συνεχίζεται μέχρι να βρεθεί το στοιχείο οπότε στη μεταβλητή *position* αποθηκεύεται η θέση που βρέθηκε το στοιχείο του πίνακα ή μέχρι να τελειώσει ο πίνακας. Επιπλέον ο αλγόριθμος χρησιμοποιεί τη λογική μεταβλητή *done* για να ελέγχει αν έχει ή όχι βρεθεί το στοιχείο που αναζητάμε.

# Αλγόριθμος Σειριακής αναζήτησης



Αλγόριθμος Sequential\_Search

Δεδομένα // n, table, key //

done ← ψευδής

position ← 0

i ← 1

Όσο (done=ψευδής) και (i≤n) επανάλαβε

    Αν table[i]=key τότε

        done ← αληθής

        position ← i

    αλλιώς

        i ← i+1

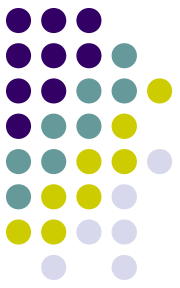
    Τέλος\_αν

Τέλος\_επανάληψης

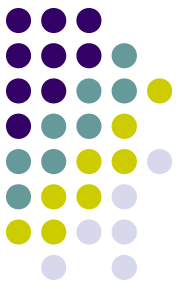
Αποτελέσματα //done, position //

Τέλος Sequential\_Search

# Ταξινόμηση φουσαλίδας (bubble sort) σε μονοδιάστατο πίνακα



- Υπάρχουν διάφορες τεχνικές ταξινόμησης, μία από αυτές είναι η ταξινόμηση ευθείας ανταλλαγής (straight exchange sort) ή ταξινόμηση φουσαλίδας (bubble sort) όπως είναι γνωστότερη.
- Η μέθοδος αυτή βασίζεται στη σύγκριση και την ανταλλαγή ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία. Σύμφωνα με τη μέθοδο αυτή κάθε φορά γίνονται διαδοχικές προσπελάσεις στον πίνακα και μετακινείται το μικρότερο στοιχείο προς την αρχή του πίνακα. Αν ο πίνακας θεωρηθεί σε κατακόρυφη θέση και τα στοιχεία του πίνακα θεωρηθούν ως φουσαλίδες σε μία δεξαμενή νερού με βάρη σύμφωνα με την τιμή τους, τότε κάθε προσπέλαση στον πίνακα έχει ως αποτέλεσμα την άνοδο της φουσαλίδας στο κατάλληλο επίπεδο βάρους.

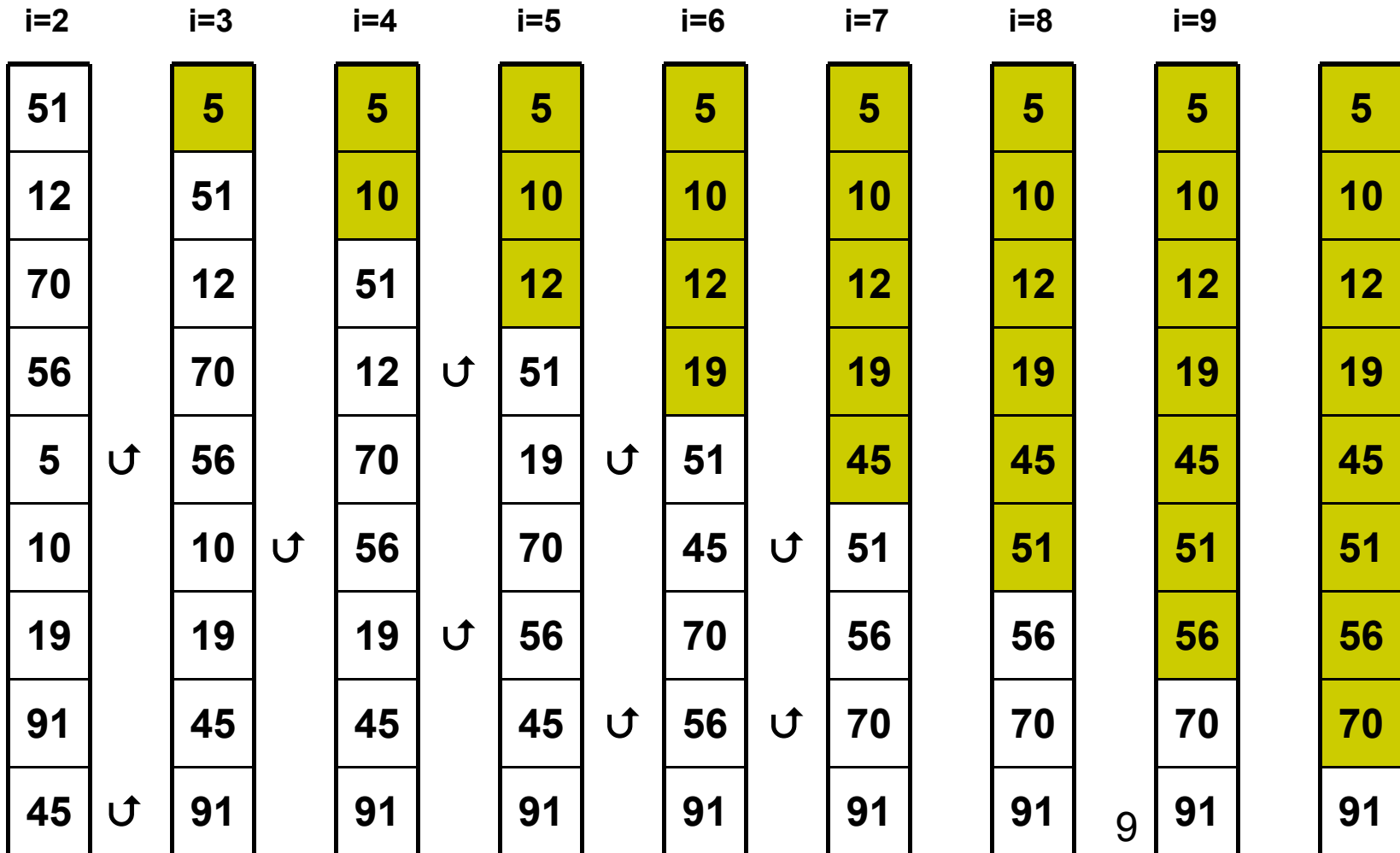
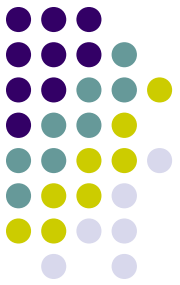


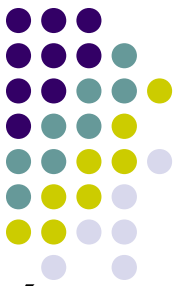
# Παράδειγμα (bubble sort) (1)

- Έστω ότι ο αρχικός πίνακας αποτελείται από εννέα στοιχεία:  
51, 12, 70, 56, 5, 10, 19, 91 και 45.
- Αν εφαρμόσουμε τη μέθοδο στα στοιχεία αυτά εξελίσσεται μια διαδικασία όπως φαίνεται στο επόμενο σχήμα:



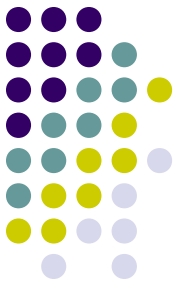
# Παράδειγμα (bubble sort) (2)





## Παράδειγμα (bubble sort) (3)

- Για να ταξινομήσουμε τον πίνακα ξεκινάμε από το τελευταίο στοιχείο (`table[9]`) και το συγκρίνουμε με το από πάνω του (`table[8]`). Αν το από κάτω είναι μικρότερο από το πάνω τα αντιμεταθέτουμε. Μετά πάμε στο επόμενο (`table[8]`) και το συγκρίνουμε με το στοιχείο που είναι από πάνω του (`table[7]`). Αυτό συνεχίζεται μέχρι να ελέγξουμε το `table[2]` με το `table[1]`. Τελειώνοντας η διαδικασία το ποιο μικρό στοιχείο θα έχει ανέβει στην κορυφή του πίνακα.
- Μετά κάνουμε πάλι τα ίδια μόνο που δουλεύουμε μόνο τα στοιχεία στις θέσεις από 2 μέχρι 9 αφού το πρώτο έχει ταξινομηθεί. Κάθε φορά που ταξινομούμε ένα στοιχείο, την επόμενη ο αλγόριθμος ελέγχει τα υπόλοιπα στοιχεία που δεν έχουν ταξινομηθεί.
- Επομένως ο αλγόριθμος χρειάζεται δύο επαναληπτικές διαδικασίες για να υλοποιηθεί.
- Τελικά η διαδικασία εκτελείται 8 φορές, δηλαδή μία λιγότερη από το πλήθος των στοιχείων του πίνακα.



# Αλγόριθμος bubble sort

Αλγόριθμος Φυσαλίδα

Δεδομένα // n, table //

Για i από 2 μέχρι n

    Για j από n μέχρι i με\_βήμα -1

        Αν table[j-1] > table[j] τότε

            temp ← table[j-1]

            table[j-1] ← table[j]

            table[j] ← temp

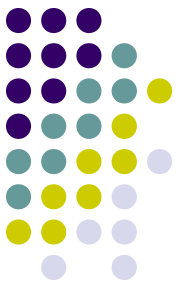
        Τέλος\_αν

    Τέλος\_επανάληψης

Τέλος\_επανάληψης

Αποτελέσματα // table //

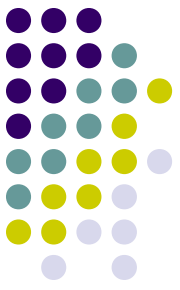
Τέλος Φυσαλίδα



# Δυσδιάστατοι πίνακες

- Εκτός από μονοδιάστατους πίνακες έχουμε δισδιάστατους, τρισδιάστατους και γενικά πολυδιάστατους πίνακες ανάλογα με τον αριθμό των δεικτών που χρησιμοποιούνται για τον καθορισμό των στοιχείων.
- Ένας δισδιάστατος πίνακας λέγεται **τετραγωνικός** όταν το μέγεθος των δύο διαστάσεων του είναι ίσο.





# Δισδιάστατοι πίνακες (1)

- Δημιουργία πίνακα
- Για να δημιουργήσουμε ένα πίνακα  $A$  π.χ. 3 γραμμών και 5 στηλών με στοιχεία που δίνει ο χρήστης από το πληκτρολόγιο γράφουμε:

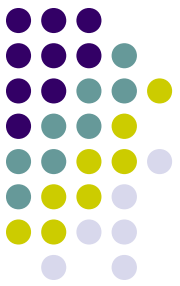
**Για  $i$  από 1 μέχρι 3**

**Για  $j$  από 1 μέχρι 5**

**Διάβασε  $A[i, j]$**

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**



## Δισδιάστατοι πίνακες (2)

- Εμφάνιση στοιχείων πίνακα
- Για να εμφανίσουμε τα στοιχεία του προηγούμενου πίνακα  $A$  γράφουμε:

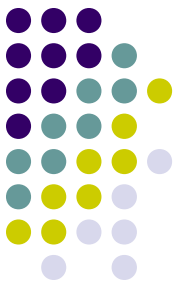
Για  $i$  από 1 μέχρι 3

    Για  $j$  από 1 μέχρι 5

        Εμφάνισε  $A[i, j]$

    Τέλος\_επανάληψης

Τέλος\_επανάληψης



## Δισδιάστατοι πίνακες (3)

- Άθροισμα και μέσος όρος στοιχείων πίνακα
- Για να υπολογίσουμε το άθροισμα και το μέσο όρο των στοιχείων του προηγούμενου πίνακα A γράφουμε:

$sum \leftarrow 0$

Για  $i$  από 1 μέχρι 3

    Για  $j$  από 1 μέχρι 5

$sum \leftarrow sum + A[i, j]$

    Τέλος\_επανάληψης

Τέλος\_επανάληψης

$mo \leftarrow sum/15$

Εμφάνισε  $sum, mo$