

<-

[i, j]

top <- top + 1

<-

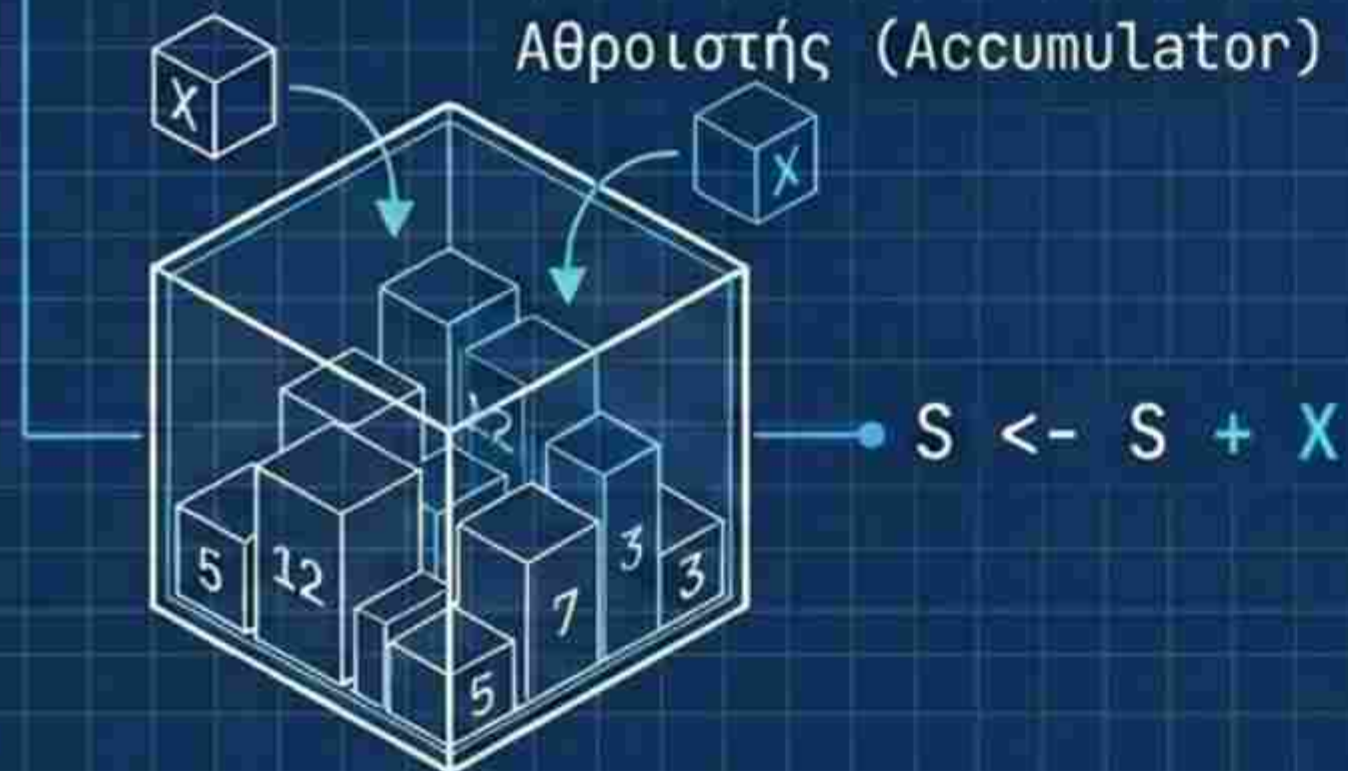
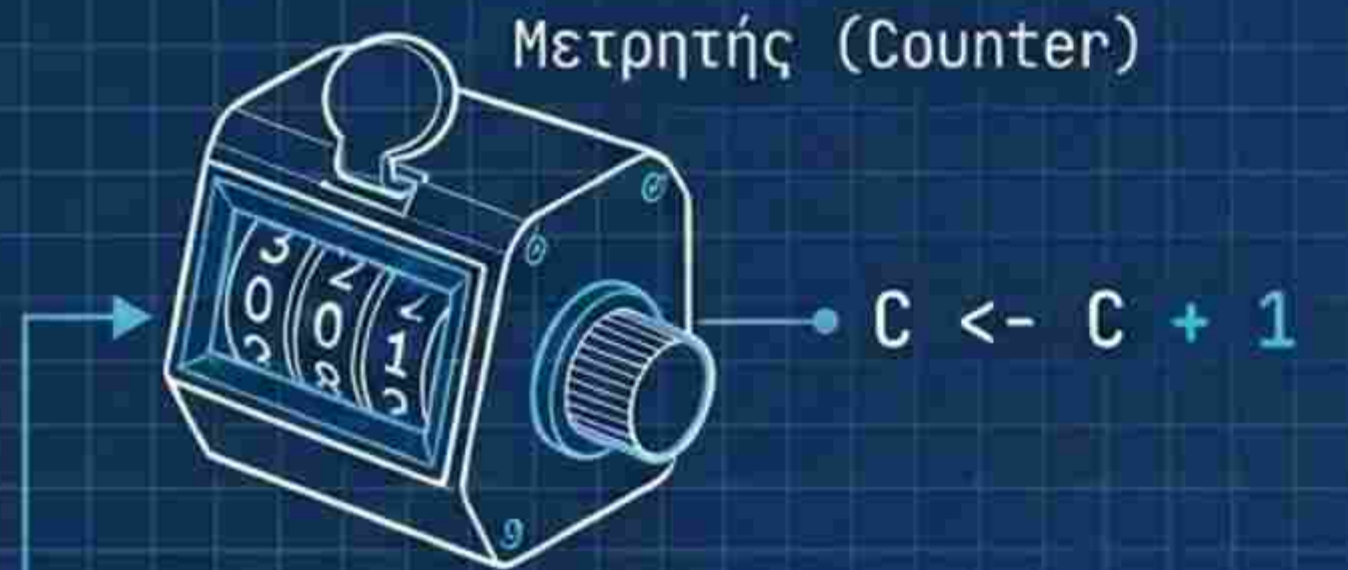
[i, j]

Η Απόλυτη Εργαλειοθήκη Μεθοδολογιών: Α.Ε.Π.Π.

Οδηγός Επιβίωσης για τα θέματα Γ & Δ – Μοντέλα
ΓΛΩΣΣΑΣ, Πίνακες, Δομές & Αναζήτηση.

top <- top + 1

Βασικοί Υπολογισμοί: Μετρητές & Αθροιστές & Δομές Επανάληψης



Παράδειγμα με ΟΣΟ (Τιμή Φρουρός):

```
Σ ← 0      ! Αρχικοποίηση Αθροιστή
Π ← 0      ! Αρχικοποίηση Μετρητή
ΔΙΑΒΑΣΕ Χ
ΟΣΟ Χ <> 0 ΕΠΑΝΑΛΑΒΕ
    Σ ← Σ + Χ
    Π ← Π + 1
ΔΙΑΒΑΣΕ Χ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Παράδειγμα με ΜΕΧΡΙΣ_ΟΤΟΥ (Τουλάχιστον μία εκτέλεση):

Χρησιμοποιείται όταν θέλουμε η επανάληψη να εκτελεστεί τουλάχιστον μία φορά, π.χ., μέχρι το άθροισμα να ξεπεράσει το 1000.

```
Σ ← 0      ! Αρχικοποίηση Αθροιστή
Π ← 0      ! Αρχικοποίηση Μετρητή
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ Χ
    Σ ← Σ + Χ
    Π ← Π + 1
ΜΕΧΡΙΣ_ΟΤΟΥ Σ > 1000
```

Μέσοι Όροι & Ποσοστά (ΠΡΟΣΟΧΗ ΣΤΙΣ ΠΑΓΙΔΕΣ)

$$MO \leftarrow \Sigma / \Pi$$

(Μέσος Όρος = Άθροισμα / Πλήθος)



Παγίδα θέματος Γ: Ποτέ μην διαιρείς αν δεν ελέγξεις ότι ο παρονομαστής είναι > 0 !

```
ΑΝ Π > 0 ΤΟΤΕ
    ΜΟ  $\leftarrow$  Σ / Π
    ΓΡΑΨΕ 'Μέσος Όρος:', ΜΟ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Δεν δόθηκαν στοιχεία'
ΤΕΛΟΣ_ΑΝ
```

Μήτρα Επιλογής: Αρχικοποίηση Μέγιστου / Ελάχιστου

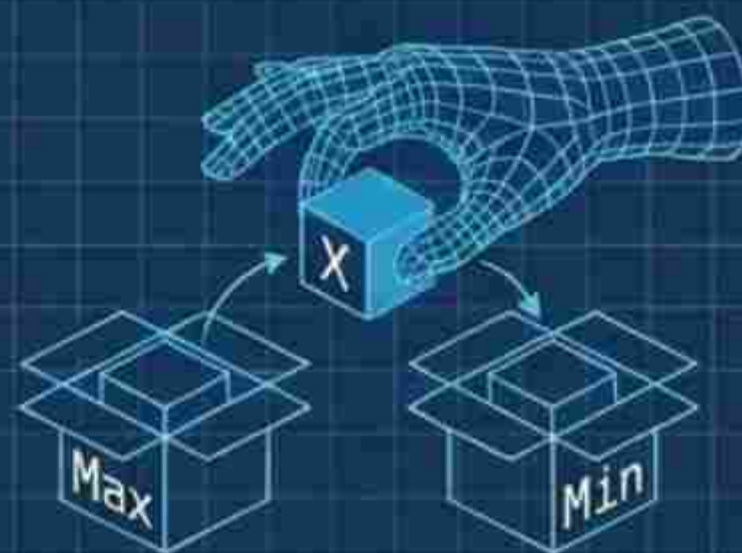
Συγκεκριμένο Πεδίο Τιμών
(Ακραίες Τιμές)



Στο Max δίνουμε τιμές εκτός ορίων κανονικά, δηλαδή αν τα όρια είναι 0-20, το max είναι -1 και το min 21.

```
Max <- -1  
Min <- 21
```

Άγνωστο Πεδίο Τιμών
(Το 1ο Στοιχείο)



Διαβάζουμε το πρώτο στοιχείο σε μια μεταβλητή (π.χ., X) εκτός επανάληψης και το εκχωρούμε στο Max και Min.

```
ΔΙΑΒΑΣΕ X  
Max <- X  
Min <- X
```

Εύρεση Max / Min & Παράλληλη Αποθήκευση (Επεκτάσεις)

Εύρεση Max / Min σε Γνωστό Εύρος (0-20) με Δομή

ΟΣΟ

Max <- -1

Min <- 21

ΔΙΑΒΑΣΕ Όνομα

ΟΣΟ Όνομα <> 'τέλος' ΕΠΑΝΑΛΑΒΕ

ΔΙΑΒΑΣΕ Βαθμός

ΑΝ Βαθμός > Max ΤΟΤΕ

Max <- Βαθμός

ΤΕΛΟΣ_ΑΝ

ΑΝ Βαθμός < Min ΤΟΤΕ

Min <- Βαθμός

ΤΕΛΟΣ_ΑΝ

ΔΙΑΒΑΣΕ Όνομα

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Εύρεση 2 Μεγαλύτερων Τιμών (Max1 & Max2)

Max1 <- -1

Max2 <- -1

ΔΙΑΒΑΣΕ Όνομα

ΟΣΟ Όνομα <> 'τέλος' ΕΠΑΝΑΛΑΒΕ

ΔΙΑΒΑΣΕ Βαθμός

ΑΝ Βαθμός > Max1 ΤΟΤΕ

Max2 <- Max1

Max1 <- Βαθμός

ΑΛΛΙΩΣ_ΑΝ Βαθμός > Max2 ΤΟΤΕ

Max2 <- Βαθμός

ΤΕΛΟΣ_ΑΝ

ΔΙΑΒΑΣΕ Όνομα

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Πλήθος Εμφανίσεων Max

Max <- -1

ΠλήθοςMax <- 0

ΔΙΑΒΑΣΕ Όνομα

ΟΣΟ Όνομα <> 'τέλος' ΕΠΑΝΑΛΑΒΕ

ΔΙΑΒΑΣΕ Βαθμός

ΑΝ Βαθμός > Max ΤΟΤΕ

Max <- Βαθμός

ΠλήθοςMax <- 1

ΑΛΛΙΩΣ_ΑΝ Βαθμός = Max ΤΟΤΕ

ΠλήθοςMax <- ΠλήθοςMax + 1

ΤΕΛΟΣ_ΑΝ

ΔΙΑΒΑΣΕ Όνομα

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Μονοδιάστατοι Πίνακες: Γέμισμα & Προσπέλαση

[1] [2] [3] [N]

42	17	9	56
----	----	---	-----	-----	----

[1][1] [1][2] [M][N]

[1][1]	10	20	30
...		
[M][N]	10	20	<i>i</i>	...	[M][N]

Γέμισμα Μονοδιάστατου Πίνακα A[N]

! Διάβασμα N στοιχείων και
καταχώριση στον Πίνακα A

ΓΙΑ *i* **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΔΙΑΒΑΣΕ A[*i*]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Γέμισμα Δισδιάστατου Πίνακα B[M][N]

! Διάβασμα M*N στοιχείων και
καταχώριση στον Πίνακα B

ΓΙΑ *i* **ΑΠΟ** 1 **ΜΕΧΡΙ** M

ΓΙΑ *j* **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΔΙΑΒΑΣΕ B[*i*][*j*]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Κανόνας: Κάθε φορά που βλέπεις πίνακα, σκέψου δομή **ΓΙΑ**. Η μεταβλητή της **ΓΙΑ** είναι ο Δείκτης σου.

Μεθοδολογίες σε Μονοδιάστατους Πίνακες

Εύρεση Αθροίσματος

```
sum ← 0  
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N  
    sum ← sum + A[i]  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Εύρεση Πλήθους υπό Συνθήκη

```
count ← 0  
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N  
    ΑΝ (συνθήκη για A[i]) ΤΟΤΕ  
        count ← count + 1  
    ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

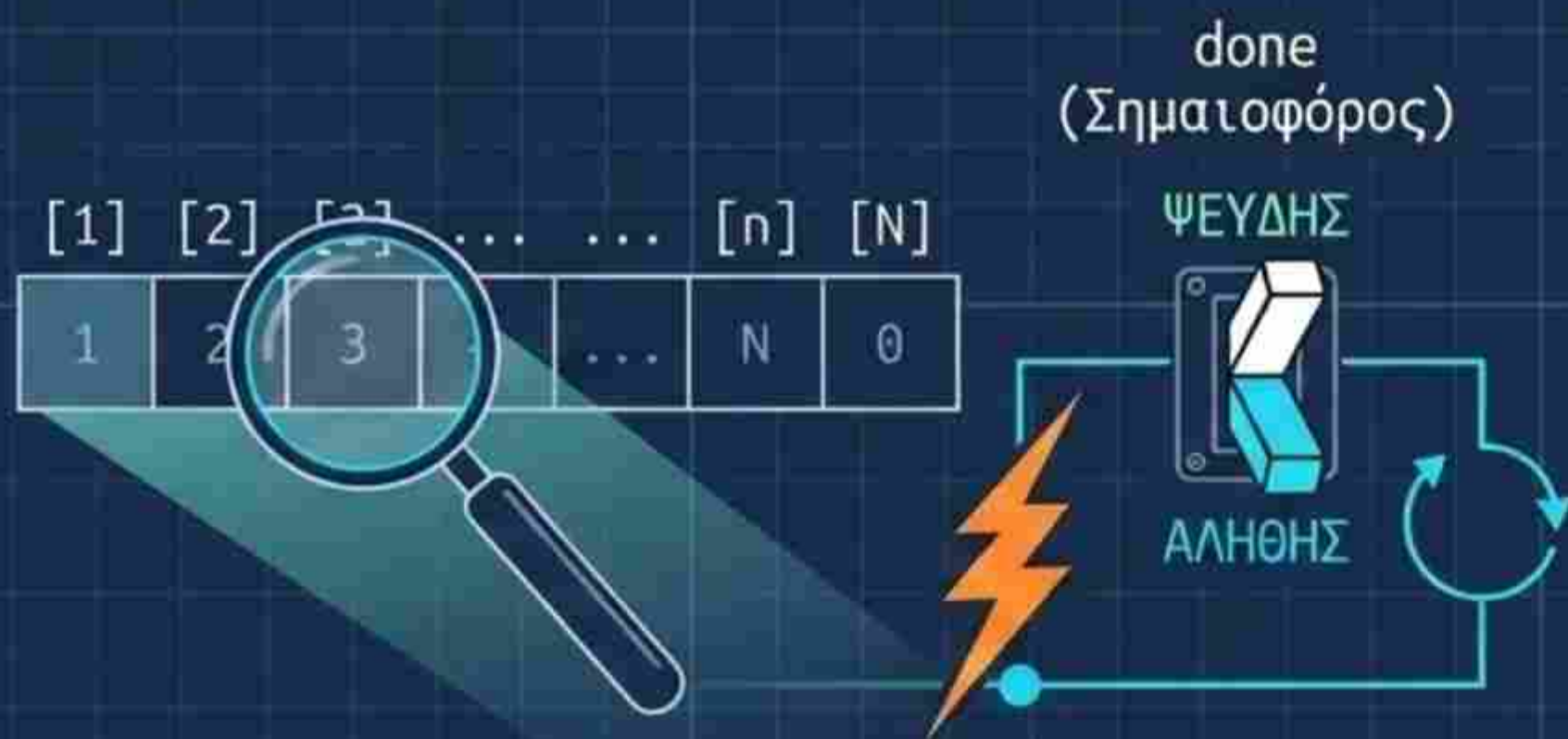
Εύρεση Μέγιστου (Max)

```
max ← A[1]  
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N  
    ΑΝ A[i] > max ΤΟΤΕ  
        max ← A[i]  
    ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Εύρεση Ελάχιστου (Min)

```
min ← A[1]  
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N  
    ΑΝ A[i] < min ΤΟΤΕ  
        min ← A[i]  
    ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

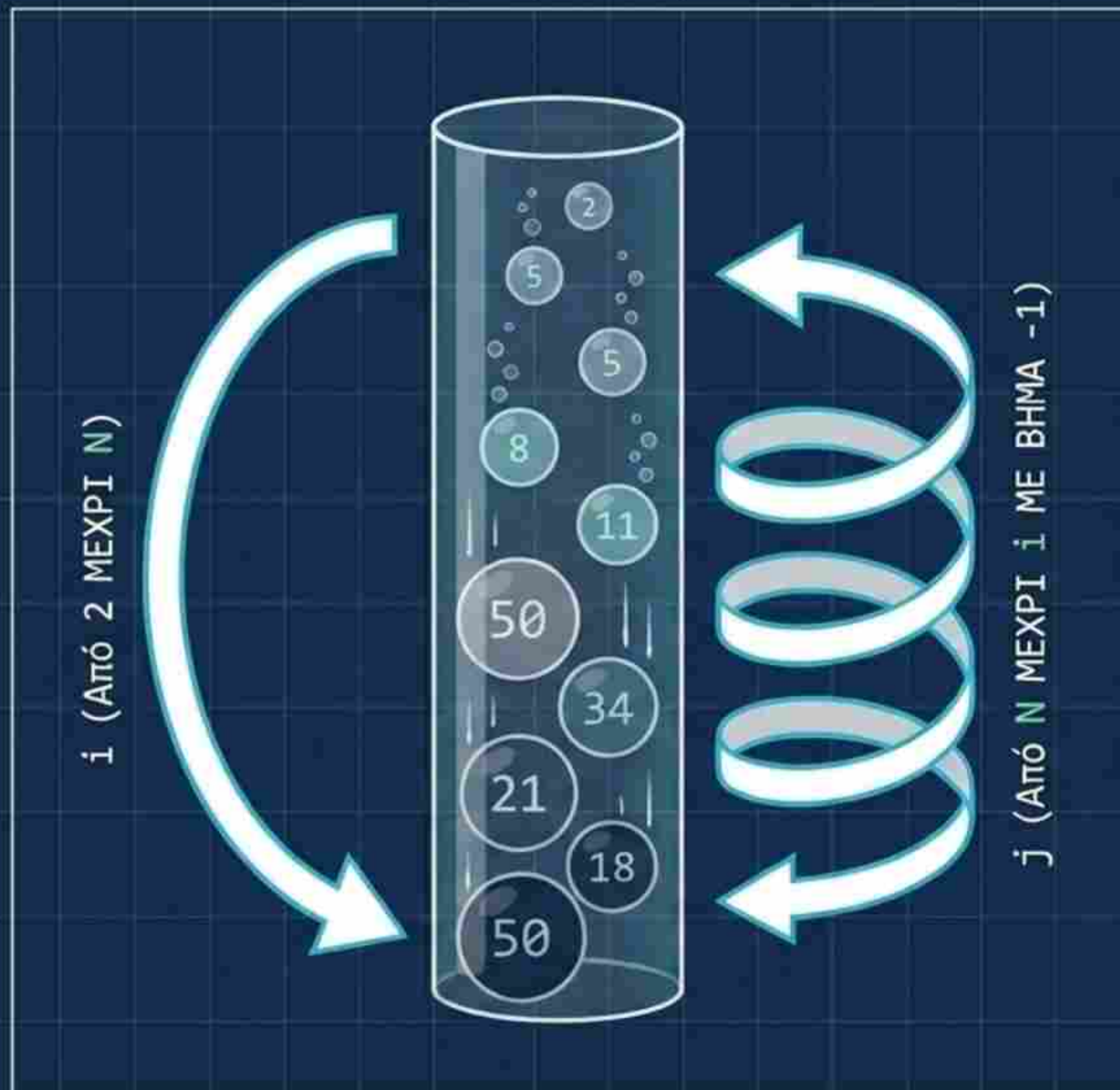
Η Τεχνική της Σειριακής Αναζήτησης (Με Σηματοφόρο)



```
done <- ΨΕΥΔΗΣ  
position <- 0  
i <- 1
```

```
ΟΣΟ (done = ΨΕΥΔΗΣ) ΚΑΙ (i <= N) ΕΠΑΝΑΛΑΒΕ  
  ΑΝ A[i] = key ΤΟΤΕ  
    done <- ΑΛΗΘΗΣ  
    position <- i  
  ΑΛΛΙΩΣ  
    i <- i + 1  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
ΑΝ done = ΑΛΗΘΗΣ ΤΟΤΕ  
  ΓΡΑΨΕ 'Το στοιχείο βρέθηκε στη θέση:',  
  position  
ΑΛΛΙΩΣ  
  ΓΡΑΨΕ 'Το στοιχείο δεν βρέθηκε.'  
ΤΕΛΟΣ_ΑΝ
```


Ταξινόμηση Φυσαλίδας (Bubble Sort)



Φθίνουσα Σειρά

```
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N
  ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1
    ΑΝ A[j-1] < A[j] ΤΟΤΕ ! Για Αύξουσα αλλαγή σε >
      ! Αντιμετάθεση (Swap)
      temp <- A[j-1]
      A[j-1] <- A[j]
      A[j] <- temp
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Παράλληλοι Πίνακες: Ταυτόχρονη Ταξινόμηση

Recipe Cards

Recipe 1
Recipe 2
Recipe 3 Recipes (i)
Recipe 4
Recipe 5
Recipe 6
Recipe 7
Recipe Cards (1)
Recipe 9

Grades

90
95
79
96
60
80
89
77
72

Names

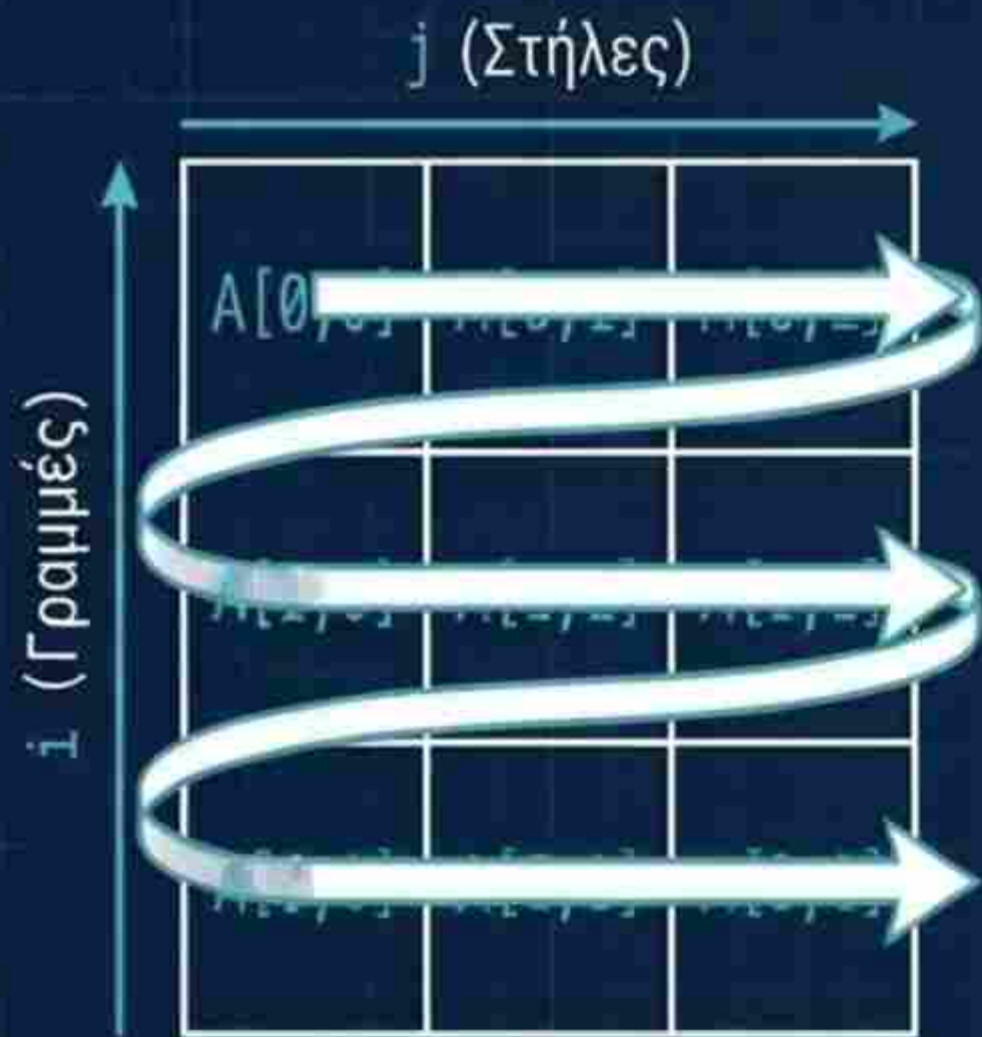
Όνομα 1
Όνομα 2
Όνομα 3
Όνομα 4
Όνομα 5
Όνομα 6
Όνομα 7
Όνομα 8
Names 9

```
! Μέσα στο AN της Φυσαλίδας...
AN Βαθμός[j-1] < Βαθμός[j] ΤΟΤΕ
  ! 1. Αντιμετάθεση Βαθμών
  tempV <- Βαθμός[j-1]
  Βαθμός[j-1] <- Βαθμός[j]
  Βαθμός[j] <- tempV

  ! 2. Υποχρεωτική Αντιμετάθεση Ονομάτων!
  tempN <- Όνομα[j-1]
  Όνομα[j-1] <- Όνομα[j]
  Όνομα[j] <- tempN
ΑΛΛΙΩΣ_ΑΝ Βαθμός[j-1] = Βαθμός[j] ΤΟΤΕ
  ! Σε περίπτωση ισοβαθμίας, αλφαβητική ταξινόμηση
  ΑΝ Όνομα[j-1] > Όνομα[j] ΤΟΤΕ
    ! Αντιμετάθεση Ονομάτων
    tempN <- Όνομα[j-1]
    Όνομα[j-1] <- Όνομα[j]
    Όνομα[j] <- tempN
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
```

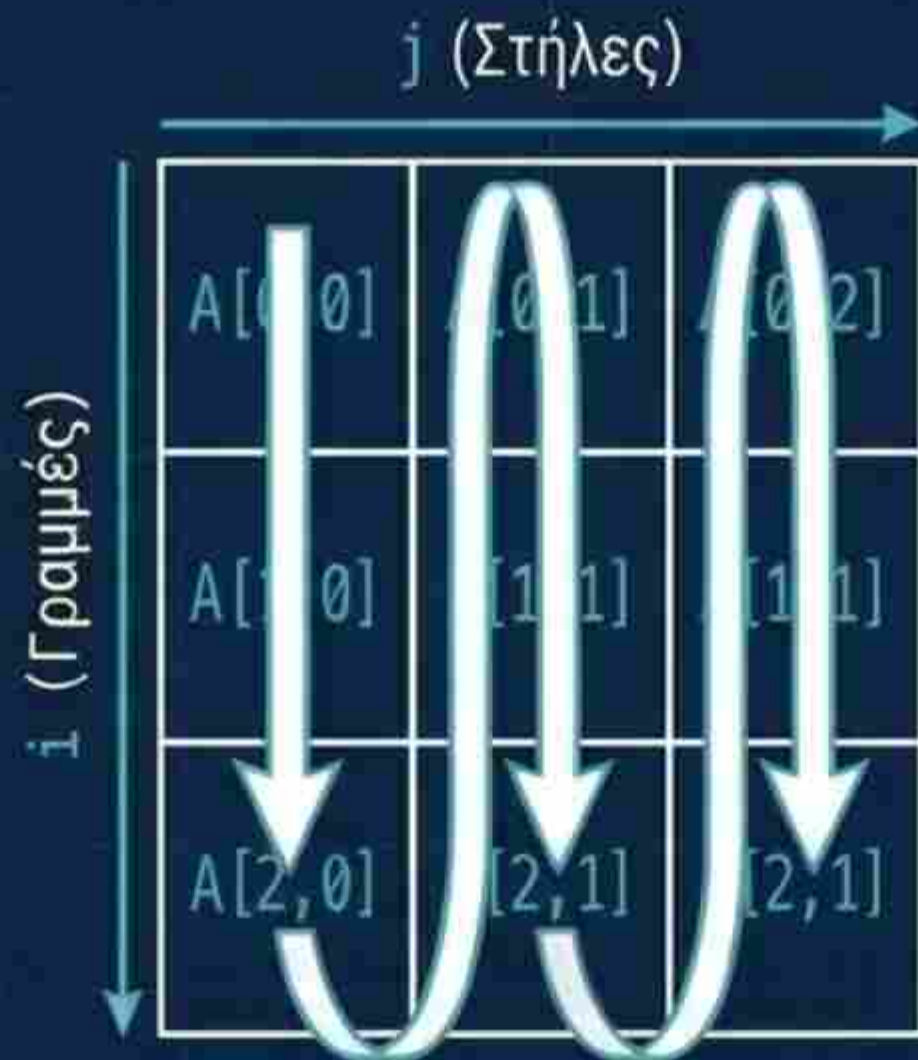
Δισδιάστατοι Πίνακες (2D Matrices): Μηχανική Προσπέλασης

Προσπέλαση Κατά Γραμμή (Row-by-Row)



ΓΙΑ $i \dots$ ΓΙΑ $j \dots$ ΔΙΑΒΑΣΕ $A[i,j]$

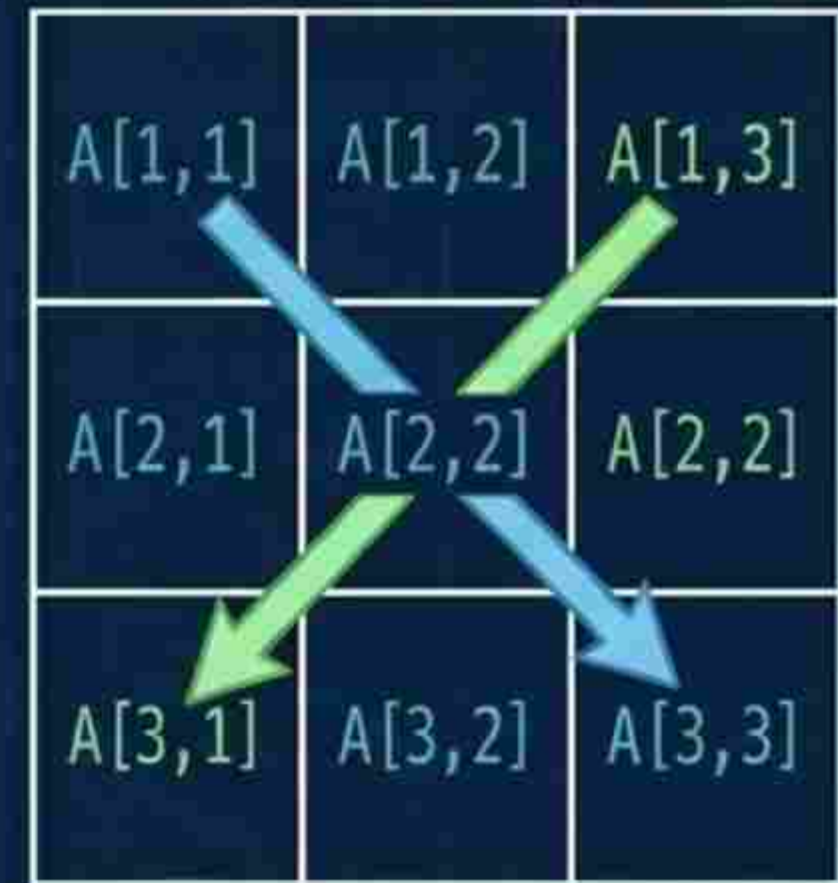
Προσπέλαση Κατά Στήλη (Col-by-Col)



ΓΙΑ $j \dots$ ΓΙΑ $i \dots$ ΔΙΑΒΑΣΕ $A[i,j]$

Τετραγωνικοί Πίνακες (Square Matrices)

Κύρια Διαγώνιος (Main Diagonal) Δευτερεύουσα Διαγώνιος (Secondary Diagonal)



$i = j$

$i + j = N + 1$

Υπολογισμοί σε Δισδιάστατους: Αθροίσματα & Μέσοι Όροι Γραμμών & Στηλών

Άθροισμα & Μέσος Όρος Ανά Γραμμή

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
  S <- 0 ! Προσοχή: Επαναμηδενισμός ΕΔΩ!
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
    S <- S + A[i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! Αποθήκευση σε Μονοδιάστατο
  Άθροισμα_Γρ[i] <- S
  Μέσος_Γρ[i] <- S / Cols
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Άθροισμα & Μέσος Όρος Ανά Στήλη

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
  S <- 0 ! Προσοχή: Επαναμηδενισμός ΕΔΩ!
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
    S <- S + A[i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ! Αποθήκευση σε Μονοδιάστατο
  Άθροισμα_Στ[j] <- S
  Μέσος_Στ[j] <- S / Rows
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Απόλυτο Μέγιστο Πίνακα

Rule: Αρχικοποίηση $Max \leftarrow A[1,1]$ πριν από τις δύο επαναλήψεις.

```
Max <- A[1,1]
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
    ΑΝ  $A[i,j] > Max$  ΤΟΤΕ
      Max <- A[i,j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Μέγιστο Ανά Γραμμή

Rule: Αρχικοποίηση $Max \leftarrow A[i,1]$ μέσα στην εξωτερική, πριν την εσωτερική επανάληψη.

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
  Max <- A[i,1]
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
    ΑΝ  $A[i,j] > Max$  ΤΟΤΕ
      Max <- A[i,j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Max γραμμής: ', Max
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

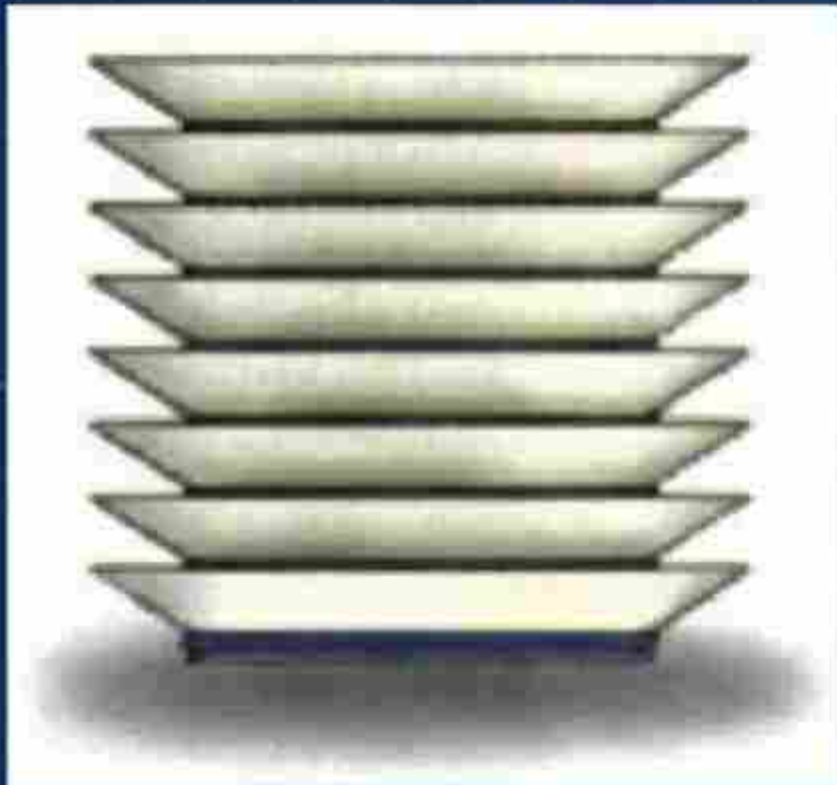
Μέγιστο Ανά Στήλη

Rule: Αρχικοποίηση $Max \leftarrow A[1,j]$ μέσα στην εξωτερική, πριν την εσωτερική επανάληψη.

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
  Max <- A[1,j]
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
    ΑΝ  $A[i,j] > Max$  ΤΟΤΕ
      Max <- A[i,j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Max στήλης: ', Max
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δυναμικές Δομές Δεδομένων: Στοίβα vs. Ουρά

Στοίβα (Stack) - LIFO



- Μηχανισμός: LIFO (Last In, First Out) - Τελευταίο μέσα, πρώτο έξω.
- Δείκτης: Ένας δείκτης (top).
- Πράξεις: Ώθηση (Push), Απώθηση (Pop).

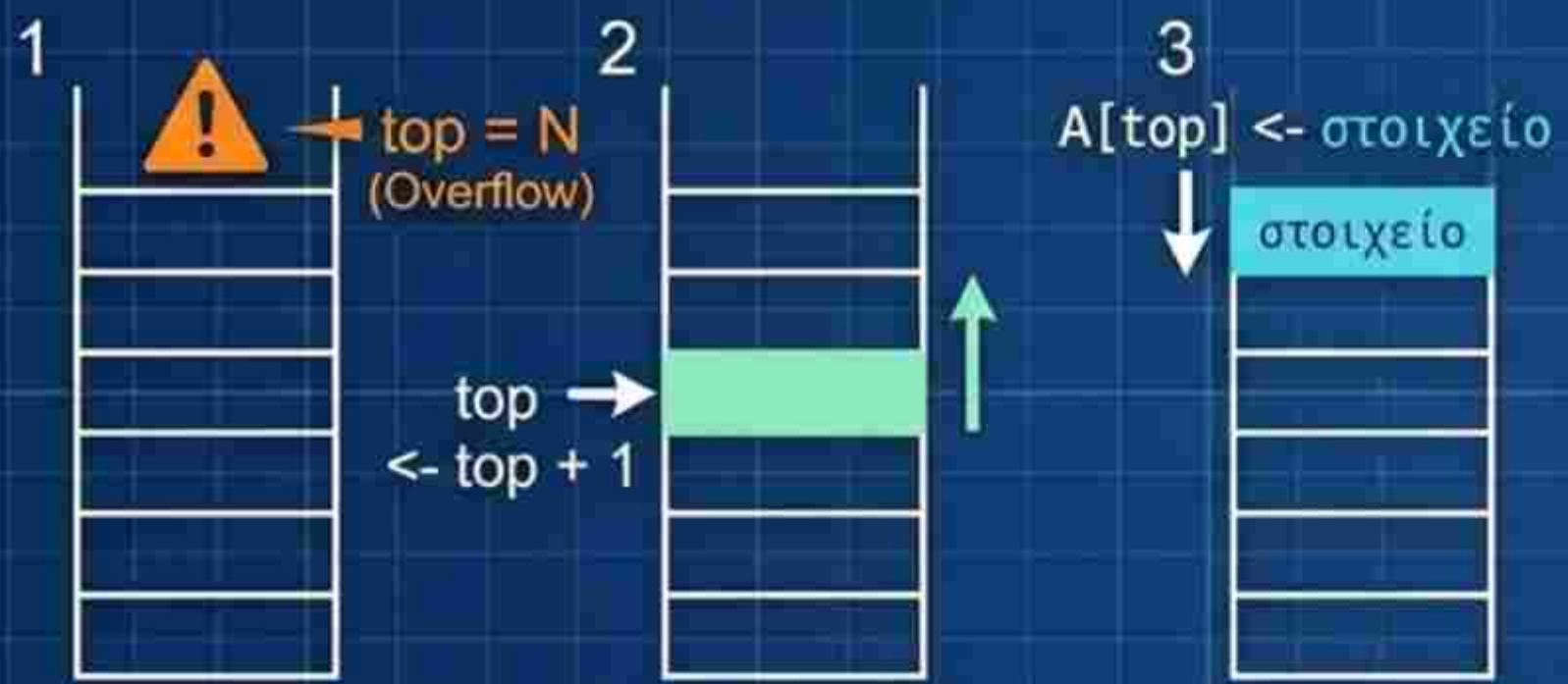
Ουρά (Queue) - FIFO



- Μηχανισμός: FIFO (First In, First Out) - Πρώτο μέσα, πρώτο έξω.
- Δείκτες: Δύο δείκτες (front, rear).
- Πράξεις: Εισαγωγή (Enqueue), Εξαγωγή (Dequeue).

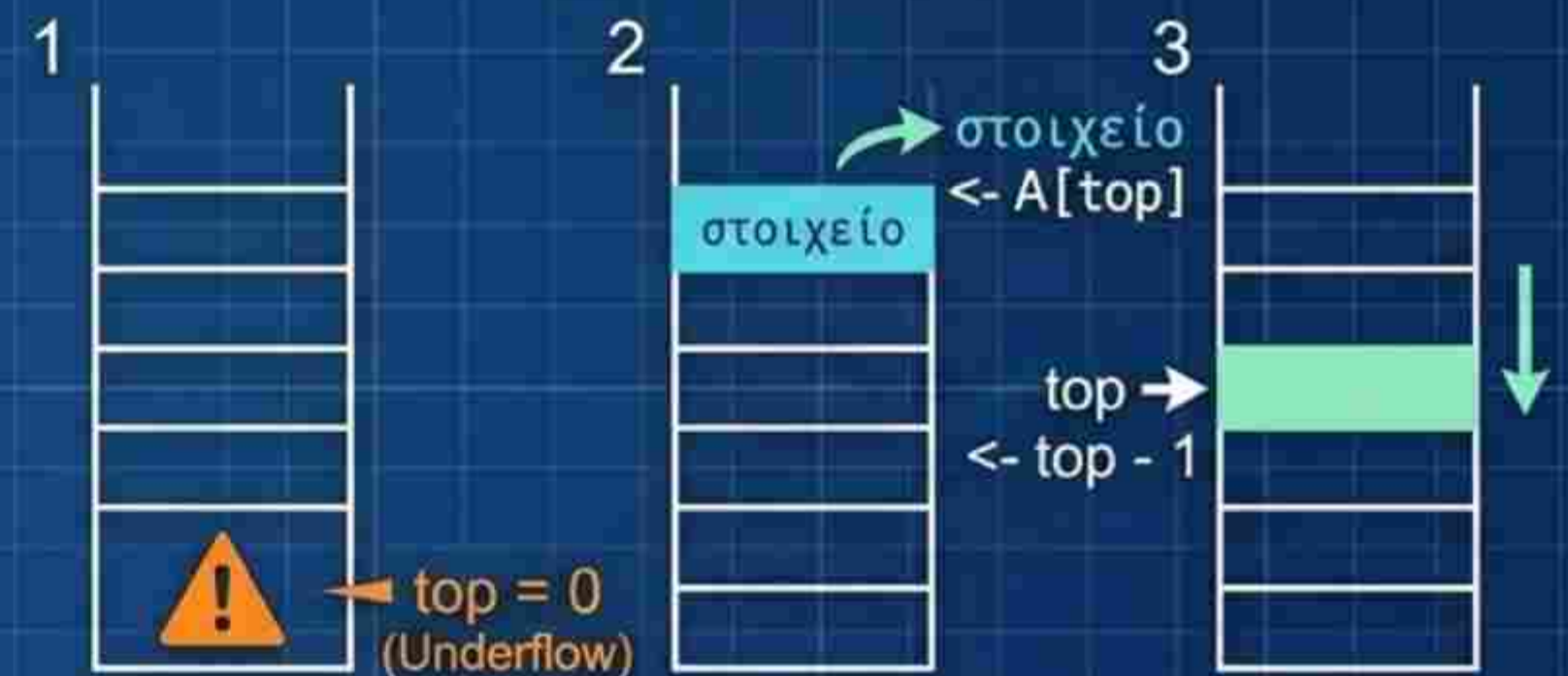
Αλγόριθμοι Στοίβας: Ώθηση (Push) & Απώθηση (Pop)

Ώθηση (Push)



```
ΑΝ  $top = N$  ΤΟΤΕ  
  ΓΡΑΨΕ 'Υπερχείλιση'  
ΑΛΛΙΩΣ  
   $top \leftarrow top + 1$   
   $A[top] \leftarrow \text{στοιχείο}$   
ΤΕΛΟΣ_ΑΝ
```

Απώθηση (Pop)



```
ΑΝ  $top = 0$  ΤΟΤΕ  
  ΓΡΑΨΕ 'Υποχείλιση'  
ΑΛΛΙΩΣ  
   $\text{στοιχείο} \leftarrow A[top]$   
   $top \leftarrow top - 1$   
ΤΕΛΟΣ_ΑΝ
```

Αλγόριθμοι Ουράς: Εισαγωγή (Enqueue) & Εξαγωγή (Dequeue)

Εισαγωγή (Enqueue)

```
ΔΙΑΒΑΣΕ στοιχείο
ΑΝ rear = N ΤΟΤΕ
    ΓΡΑΨΕ 'Γεμάτη ουρά'
ΑΛΛΙΩΣ_ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
    front ← 1
    rear ← 1
    A[rear] ← στοιχείο
ΑΛΛΙΩΣ
    rear ← rear + 1
    A[rear] ← στοιχείο
ΤΕΛΟΣ_ΑΝ
```

Εξαγωγή (Dequeue)

```
ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
    ΓΡΑΨΕ 'Άδεια ουρά'
ΑΛΛΙΩΣ_ΑΝ (front = rear ) ΤΟΤΕ
    ΓΡΑΨΕ 'Εξάγεται το στοιχείο:', A[front]
    front ← 0
    rear ← 0
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Εξάγεται το στοιχείο:', A[front]
    front ← front + 1
ΤΕΛΟΣ_ΑΝ
```

Δομημένος Προγραμματισμός: Υποπρογράμματα

Διαδικασίες (Procedures)

- **Χρήση:** Καλείται ως αυτόνομη εντολή (π.χ. **ΚΑΛΕΣΕ Ταξινόμηση(A, N)**).
- **Επιστροφή:** Επιστρέφει από 0 έως πολλές τιμές μέσω της λίστας παραμέτρων.
- **Παράμετροι:** Στη λίστα παραμέτρων των διαδικασιών δεν επιτρέπεται να περαστούν σταθερές τιμές.

Συναρτήσεις (Functions)

- **Χρήση:** Καλείται μέσα σε έκφραση (π.χ. $MO \leftarrow \text{Άθροισμα}(X, Y) / 2$).
- **Επιστροφή:** Επιστρέφει ΠΑΝΤΑ μία και μόνο μία τιμή με το όνομά της. **Δεν μπορεί να επιστρέψει πίνακα!**
- **Παράμετροι:** Οι πίνακες μπορούν να περάσουν σε συναρτήσεις, αλλά **δεν μπορούν να επιστραφούν.**

Χρυσός Κανόνας Παραμέτρων στα Υποπρογράμματα

Main Program

Key Rules:

1. Η λίστα παραμέτρων πρέπει να συμφωνεί σε πλήθος, τύπο, και σειρά.
2. Δεν δηλώνουμε το μέγεθος των πινάκων στις τυπικές παραμέτρους (π.χ., γράφουμε ΔΙΑΔΙΚΑΣΙΑ Sort(A, N) και ΟΧΙ Sort(A[100], N)).
3. Κάθε υποπρόγραμμα είναι τυφλό στις μεταβλητές του άλλου (Local Scope). Η επικοινωνία γίνεται μόνο μέσω των παραμέτρων!

Subprogram

Τυπικές
Παράμετροι
(Subprogram)

ΑΚΕΡΑΙΑ

ΠΡΑΓΜΑΤΙΚΗ

Πραγματικές
Παράμετροι
(Main Program)

ΑΚΕΡΑΙΑ

ΠΡΑΓΜΑΤΙΚΗ

Παράδειγμα Επαναληπτικής Κλήσης (2D Array)

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N_ROWS  
  SUM <- Άθροισμα_Γραμμής(A, i, N_COLS)  
  ΓΡΑΨΕ "Άθροισμα γραμμής ", i, ":", SUM  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
// Η συνάρτηση 'Άθροισμα_Γραμμής' καλείται για κάθε γραμμή 'i' του πίνακα 'A'.
```

Το Μοντέλο Εκτέλεσης: Η Ροή των Θεμάτων Γ & Δ

- 1 **Δηλώσεις:** Ορισμός μεταβλητών και πινάκων (ΑΚΕΡΑΙΕΣ, ΠΡΑΓΜΑΤΙΚΕΣ).
- 2 **Αρχικοποιήσεις:** Μηδενισμός αθροιστών/μετρητών, αρχικές τιμές Max/Min.
- 3 **Επιλογή Δομής Επανάληψης:** Καθορισμός της κατάλληλης δομής για τη λύση.
ΓΙΑ: Γνωστό πλήθος επαναλήψεων.
ΟΣΟ: Άγνωστο πλήθος, πιθανότητα μη εκτέλεσης.
ΜΕΧΡΙΣ_ΟΤΟΥ: Άγνωστο πλήθος, τουλάχιστον μία εκτέλεση.
- 4 **Κύρια Επεξεργασία (The Loop):** Υλοποίηση των περισσότερων υπολογισμών που ζητούνται εντός του βρόχου (π.χ., αθροιστές, μετρητές).
Σημείωση: Στις ασκήσεις με πίνακες, τις περισσότερες φορές λύνουμε σε ξεχωριστό βρόχο το κάθε ερώτημα, αφού τα δεδομένα δεν χάνονται όπως στις ασκήσεις χωρίς πίνακες.
- 5 **Έξοδος (Εκτύπωση):** Εμφάνιση αποτελεσμάτων μετά τον βρόχο για τα τελικά αποτελέσματα, ενώ μέσα στον βρόχο εμφανίζουμε ό,τι θέλουμε να εμφανιστεί πολλές φορές.

Η Αλγοριθμική Σκέψη Δεν Μαθαίνεται Απέξω.
Οικοδομείται.

Το κλειδί για το Άριστα (20) στα θέματα Γ & Δ δεν είναι η αποστήθιση κώδικα, αλλά η ικανότητα να σπας ένα πολύπλοκο πρόβλημα στα βασικά του συστατικά: **Μεταβλητές, Πίνακες, Επαναλήψεις.**

Διάβασε την εκφώνηση προσεκτικά.

Σχεδίασε τη δομή δεδομένων σου.

Χτίσε τον αλγόριθμό σου.

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

> compilation successful.