**ΤΥΠΟΛΟΓΙΟ ΑΕΠΠ**

**ΤΕΛΕΣΤΕΣ & ΕΝΤΟΛΕΣ**

Αριθμητικοί Τελεστές: +, -, \*, /, ^, DIV, MOD

Συγκριτικοί Τελεστές: =, <>, >, >=, <, <=

Λογικοί Τελεστές: ΚΑΙ, Ή, ΟΧΙ

Εισαγωγή από χρήστη: ΔΙΑΒΑΣΕ λίστα μεταβλητών

Έξοδος στην οθόνη ή στον εκτυπωτή: ΓΡΑΨΕ λίστα στοιχείων

Εκχώρηση τιμής: Μεταβλητή ← έκφραση

|  |
| --- |
| **ΚΑΝΟΝΕΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ ΠΡΑΞΕΩΝ**   1. Πρώτα εκτελούνται οι πράξεις μέσα στις παρενθέσεις 2. Έπειτα εκτελούνται οι πράξεις ύψωσης σε δύναμη (^) 3. Έπειτα εκτελούνται οι πολλαπλασιασμοί, διαιρέσεις div και mod. Οι πράξεις αυτές είναι ίδιας προτεραιότητας μεταξύ τους. Οι τελεστές div και mod μπορούν να κάνουν πράξεις μόνο μεταξύ ακεραίων και δίνουν ακέραια αποτελέσματα. 4. Κατόπιν εκτελούνται προσθέσεις και αφαιρέσεις (+,-). Οι πράξεις αυτές είναι ίδιας προτεραιότητας μεταξύ τους. 5. Ύστερα εκτελούνται οι πράξεις σύγκρισης (>, <, >=, <=, =, < >). Δεν επιτρέπεται η χρήση διαδοχικών τελεστών σύγκρισης. 6. Τέλος εκτελούνται οι λογικές πράξεις (ΚΑΙ, Ή, ΟΧΙ). Μεγαλύτερη προτεραιότητα έχει η πράξη ΟΧΙ. Ακολουθεί η πράξη ΚΑΙ, και τέλος η πράξη Ή. 7. Σε μία σειρά πράξεων ίδιας προτεραιότητας, οι πράξεις εκτελούνται από τα αριστερά προς τα δεξιά (με τη σειρά που τις συναντάμε) |

**ΔΟΜΗ ΕΠΙΛΟΓΗΣ**

|  |  |
| --- | --- |
| Δομή απλής επιλογής:  **ΑΝ** συνθήκη **ΤΟΤΕ**  Εντολή 1  …  Εντολή ν **ΤΕΛΟΣ\_ΑΝ** | διάγραμμα ροής ΑΝ |
| Δομή σύνθετης επιλογής:  **ΑΝ** συνθήκη **ΤΟΤΕ** Εντολή 1  …  Εντολή ν **ΑΛΛΙΩΣ**  Εντολή 1  …  Εντολή ν **ΤΕΛΟΣ\_ΑΝ** | διάγραμμα ροής Σύνθετης ΑΝ |
| Δομή πολλαπλής επιλογής:  **ΑΝ** συνθήκη1 **ΤΟΤΕ** Εντολή 1  …  Εντολή ν **ΑΛΛΙΩΣ\_ΑΝ** συνθήκη2 **ΤΟΤΕ** Εντολή 1  …  Εντολή ν  … **ΑΛΛΙΩΣ** Εντολή 1  …  Εντολή ν **ΤΕΛΟΣ\_ΑΝ** | διάγραμμα ροής Πολλαπλής ΑΝ |
| **ΕΠΙΛΕΞΕ** <έκφραση>  **ΠΕΡΙΠΤΩΣΗ** <λίστα\_τιμών\_1>  <εντολές\_1>  **ΠΕΡΙΠΤΩΣΗ** <λίστα\_τιμών\_2>  <εντολές\_2>  ......  **ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ**  <εντολές\_αλλιώς>  **ΤΕΛΟΣ\_ΕΠΙΛΟΓΩΝ** |  |

**ΚΛΙΜΑΚΩΤΗ ΧΡΕΩΣΗ**

Παράδειγμα: κλιμακωτή πολιτική τιμών σύμφωνα με τον ακόλουθο πίνακα

|  |  |
| --- | --- |
| **Διάρκεια σε sec** | **Χρέωση ανά sec** |
| 1-500 | 1,5 |
| 501-800 | 0,9 |
| >800 | 0,5 |

**Κώδικας**

ΑΝ ΔΙΑΡΚΕΙΑ>=1 ΚΑΙ ΔΙΑΡΚΕΙΑ<=500 ΤΟΤΕ

ΧΡΕΩΣΗ← ΔΙΑΡΚΕΙΑ \* 1.5

ΑΛΛΙΩΣ\_ΑΝ ΔΙΑΡΚΕΙΑ <= 800 ΤΟΤΕ

ΧΡΕΩΣΗ← 500 \* 1.5 +(ΔΙΑΡΚΕΙΑ - 500) \* 0.9

ΑΛΛΙΩΣ

ΧΡΕΩΣΗ← 500 \* 1.5 +300 \* 0.9 +(ΔΙΑΡΚΕΙΑ - 800) \* 0.5

ΤΕΛΟΣ\_ΑΝ

|  |
| --- |
| **Ποσοστό=** πλήθος αυτού που ψάχνω το ποσοστό / συνολικό πλήθος \*100 |
| **ΦΠΑ πχ. 23% για μία χρέωση:** ΦΠΑ←23/100\*χρέωση  **Άρα συνολική χρέωση**: χρέωση← χρέωση +ΦΠΑ |
| **έκπτωση πχ. 10% για μία χρέωση:** έκπτωση←10/100\*χρέωση  **Άρα συνολική χρέωση**: χρέωση← χρέωση – έκπτωση |
| **Για να ελέγξω αν ένας αριθμός Χ είναι άρτιος:** Χ mod 2 = 0 |
| **Για να ελέγξω αν ένας αριθμός Χ είναι περιττός:** Χ mod 2 = 1 |
| **Για να ελέγξω αν ένας αριθμός Χ είναι πολλαπλάσιο πχ. του 8:** Χ mod 8 = 0 |
| **Για να ελέγξω αν ένας αριθμός Χ διαιρείται ακριβώς με έναν άλλο αριθμό πχ. το 4:** Χ mod 4 = 0 |
| **Για να ελέγξω αν ένας αριθμός Χ δεν διαιρείται ακριβώς με έναν άλλο αριθμό πχ. το 4:** Χ mod 4 < > 0 |
| **Για να ελέγξω αν ένας αριθμός Χ είναι ακέραιος:** Χ = Α\_Μ(Χ) |

**ΕΥΡΕΣΗ 2 ΜΕΓΙΣΤΩΝ ΑΠΟ ΕΝΑ ΣΥΝΟΛΟ ΑΓΝΩΣΤΟΥ ΠΛΗΘΟΥΣ ΣΤΟΙΧΕΙΩΝ**

|  |
| --- |
| Ι<-0  ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ  ΔΙΑΒΑΣΕ ΣΤΟΙΧΕΙΟ  Ι<-Ι+1  ΑΝ Ι=1 ΤΟΤΕ  ΜΑΧ1<- ΣΤΟΙΧΕΙΟ  ΜΑΧ2<-ΣΤΟΙΧΕΙΟ  ΑΛΛΙΩΣ  ΑΝ ΣΤΟΙΧΕΙΟ>ΜΑΧ1 ΤΟΤΕ  ΜΑΧ2<- ΜΑΧ1  ΜΑΧ1<- ΣΤΟΙΧΕΙΟ  ΑΛΛΙΩΣ\_ΑΝ ΣΤΟΙΧΕΙΟ>ΜΑΧ2 ΤΟΤΕ  ΜΑΧ2<- ΣΤΟΙΧΕΙΟ  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΑΝ  ΜΕΧΡΙΣ\_ΟΤΟΥ συνθήκη |

**ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ**

|  |  |
| --- | --- |
| **ΟΣΟ** συνθήκη **ΕΠΑΝΑΛΑΒΕ** Εντολή 1  …  Εντολή ν **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | διάγραμμα ροής ΌΣΟ |
| **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** Εντολή 1  …  Εντολή ν **ΜΕΧΡΙΣ\_ΟΤΟΥ** συνθήκη | διάγραμμα ροής ΌΣΟ |
| **ΓΙΑ** μεταβλητή **ΑΠΟ** ατ **ΜΕΧΡΙ** ττ **ΜΕ\_ΒΗΜΑ** β Εντολή 1  …  Εντολή ν **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | διάγραμμα ροής ΓΙΑ |

**Σημείωση**:

* Οι δομές **ΟΣΟ …ΕΠΑΝΑΛΑΒΕ** και **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** χρησιμοποιούνται όταν δεν είναι γνωστό το πλήθος των επαναλήψεων. Οι διαφορές τους είναι:

(α) οι εντολές της δομής **ΟΣΟ** μπορούν να εκτελεστούν καμία, μία ή περισσότερες φορές ενώ της **ΑΡΧΗΣ\_ΕΠΑΝΑΛΗΨΗΣ** εκτελούνται τουλάχιστον μία φορά.

(β) στην **ΟΣΟ** ο έλεγχος της συνθήκης γίνεται στην αρχή ενώ στο **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** γίνεται στο τέλος

(γ) η **ΟΣΟ** εκτελείται όσο η συνθήκη είναι αληθής ενώ η **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** όσο η συνθήκη είναι ψευδής.

* Η δομή **ΓΙΑ…ΑΠΟ…ΜΕΧΡΙ** χρησιμοποιείται όταν είναι γνωστό το πλήθος των επαναλήψεων.
* Όταν μας ζητείται να κάνουμε έλεγχο ορθής εισαγωγής δεδομένων προτιμάται η δομή ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ όπου μέσα της εισάγουμε μόνο τις εντολές ΓΡΑΨΕ και ΔΙΑΒΑΣΕ.

**ΜΕΤΑΤΡΟΠΕΣ ΜΕΤΑΞΥ ΕΝΤΟΛΩΝ ΕΠΑΝΑΛΗΨΗΣ**

(όπου με=μεταβλητή, ατ=αρχική τιμή, ττ=τελική τιμή, β=βήμα)

|  |  |
| --- | --- |
| **ΓΙΑ …ΑΠΟ...ΜΕΧΡΙ…ΜΕ\_ΒΗΜΑ…** | **ΟΣΟ…ΕΠΑΝΑΛΑΒΕ** |
| Όταν το βήμα είναι θετικό  **ΓΙΑ** με **ΑΠΟ** ατ **ΜΕΧΡΙ** ττ **ΜΕ\_ΒΗΜΑ** β  Εντολές  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | με← ατ  **ΟΣΟ** με <= ττ **ΕΠΑΝΑΛΑΒΕ**  Εντολές  με← με +β  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |
| Όταν το βήμα είναι αρνητικό  **ΓΙΑ** με **ΑΠΟ** ατ **ΜΕΧΡΙ** ττ **ΜΕ\_ΒΗΜΑ** -β  Εντολές  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | με← ατ  **ΟΣΟ** με>=ττ **ΕΠΑΝΑΛΑΒΕ**  Εντολές  με← με - β  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |

|  |  |
| --- | --- |
| **ΟΣΟ…ΕΠΑΝΑΛΑΒΕ** | **ΓΙΑ …ΑΠΟ...ΜΕΧΡΙ…ΜΕ\_ΒΗΜΑ…** |
| με← ατ  **ΟΣΟ** με<τελεστής σύγκρισης>ττ **ΕΠΑΝΑΛΑΒΕ**  Εντολές 1  με← με +β  εντολές 2  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | **ΓΙΑ** με **ΑΠΟ** ατ **ΜΕΧΡΙ** ττ **ΜΕ\_ΒΗΜΑ** β  Εντολές 1  Εντολές 2  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |
| Αν υπάρχουν εντολές πριν την αλλαγή της μεταβλητής κατά το βήμα, αυτές δεν επηρεάζουν τη μεταβλητή της επανάληψης.  Αν υπάρχουν εντολές μετά την αλλαγή της μεταβλητής κατά το βήμα αυτές επηρεάζονται από την αλλαγή τη μεταβλητής | |
| Π.χ.  Χ←20  Ι←1  **ΟΣΟ** Ι<=10 **ΕΠΑΝΑΛΑΒΕ**  **ΓΡΑΨΕ** Ι  Ι←Ι+2  Α← Ι+Χ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | Χ← 20  **ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 10 **ΜΕ\_ΒΗΜΑ** 2  //δεν επηρεάζεται γιατί προηγείται της αλλαγής της μεταβλητής κατά το βήμα  **ΓΡΑΨΕ** Ι  // επηρεάζεται γιατί εκτελείται μετά την αλλαγής της μεταβλητής κατά το βήμα  Α← (Ι+2)\*Χ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |

|  |  |
| --- | --- |
| **ΟΣΟ…ΕΠΑΝΑΛΑΒΕ** | **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** |
| **ΟΣΟ** συνθήκη **ΕΠΑΝΑΛΑΒΕ**  Εντολές  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  Εντολές  **ΜΕΧΡΙΣ‑ ΟΤΟΥ**  συμπληρωματική\_συνθήκη |
| Επειδή στο **ΟΣΟ** οι εντολές μπορεί να μην εκτελεστούν καμία φορά, πρέπει να φροντίσουμε να γίνει το ίδιο και κατά τη μετατροπή στην **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** (όπου είναι γνωστό ότι οι εντολές εκτελούνται τουλάχιστον μία φορά). Αυτό επιτυγχάνεται αν βάλουμε το **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** μέσα σε έλεγχο **ΑΝ** όπου θα ελέγχεται η συνθήκη της ΟΣΟ | |
| Π.χ.  **ΔΙΑΒΑΣΕ** Χ  **ΟΣΟ** Χ>0 **ΕΠΑΝΑΛΑΒΕ**  **ΓΡΑΨΕ** Χ  **ΔΙΑΒΑΣΕ** Χ  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** | **ΔΙΑΒΑΣΕ** Χ  **ΑΝ** Χ>0 **ΤΟΤΕ**  **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΓΡΑΨΕ** Χ  **ΔΙΑΒΑΣΕ** Χ  **ΜΕΧΡΙΣ\_ΟΤΟΥ** Χ<=0  **ΤΕΛΟΣ\_ΑΝ** |

|  |  |
| --- | --- |
| **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** | **ΟΣΟ…ΕΠΑΝΑΛΑΒΕ** |
| Επειδή στο **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** οι εντολές εκτελούνται τουλάχιστον μία φορά, πρέπει να φροντίσουμε να γίνει το ίδιο και κατά τη μετατροπή σε ΟΣΟ. Αυτό επιτυγχάνεται αν εμφανίζουμε τις εντολές της δομής επανάληψης και μία φορά απέξω από τη δομή επανάληψης | |
| Π.χ.  I←0  **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΔΙΑΒΑΣΕ** Χ  **ΓΡΑΨΕ** Χ  I← I+1  **ΜΕΧΡΙΣ\_ΟΤΟΥ** Χ>0 | I←0  **ΔΙΑΒΑΣΕ** Χ  **ΓΡΑΨΕ** Χ  I← I+1  **ΟΣΟ** Χ<=0 **ΕΠΑΝΑΛΑΒΕ**  **ΔΙΑΒΑΣΕ** Χ  **ΓΡΑΨΕ** Χ  I← I+1  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ** |

**ΣΤΑΤΙΚΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ:** το μέγεθος της μνήμης που απαιτείται για την αποθήκευση των κόμβων της καθορίζεται από την αρχή και δεν μεταβάλλεται κατά την εκτέλεση του προγράμματος. Οι κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

**(Α) ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ**

|  |  |
| --- | --- |
| **Δήλωση πίνακα:**  Τύπος δεδομένων: όνομα πίνακα [πλήθος στοιχείων]  **ΑΝΑΓΝΩΣΗ / ΕΜΦΑΝΙΣΗ ΠΙΝΑΚΑ**  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΔΙΑΒΑΣΕ Α[Ι]  ΓΡΑΨΕ Α[Ι]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **ΑΘΡΟΙΣΜΑ /ΜΕΣΟΣ ΟΡΟΣ**  ΑΘΡ←0  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΘΡ← ΑΘΡ+Α[I]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ ΑΘΡ  ΜΟ← ΑΘΡ/Ν  ΓΡΑΨΕ ΜΟ |
| **ΠΛΗΘΟΣ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ ΥΠΟ ΚΡΙΤΗΡΙΟ**  ΠΛΗΘΟΣ←0  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΝ Α[Ι]>50 ΤΟΤΕ //π.χ. ψάχνω αριθμό στοιχείων >50  ΠΛΗΘΟΣ←ΠΛΗΘΟΣ+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ ΠΛΗΘΟΣ | **ΜΕΓΙΣΤΟ/ΕΛΑΧΙΣΤΟ**  MAX←A[1]  ΘΕΣΗ\_ΜΑΧ← 1  MIN←A[1]  ΘΕΣΗ\_ΜΙΝ←1  ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ Ν  ΑΝ Α[Ι]>MAX ΤΟΤΕ  MAX←A[I]  ΘΕΣΗ\_ΜΑΧ← Ι  ΤΕΛΟΣ\_ΑΝ  ΑΝ Α[Ι]<MIN ΤΟΤΕ  MIN←A[I]  ΘΕΣΗ\_ΜΙΝ← Ι  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ MAX, ΘΕΣΗ\_ΜΑΧ,MIN, ΘΕΣΗ\_ΜΙΝ |

**(Β) ΔΙΣΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ**

|  |  |
| --- | --- |
| **Δήλωση πίνακα:**  Τύπος δεδομένων: όνομα πίνακα [αρ. γραμμών, αρ. στηλών]  **ΑΝΑΓΝΩΣΗ / ΕΜΦΑΝΙΣΗ ΠΙΝΑΚΑ**  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  ΔΙΑΒΑΣΕ Α[Ι,J]  ΓΡΑΨΕ Α[Ι,J]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **ΑΘΡΟΙΣΜΑ /ΜΕΣΟΣ ΟΡΟΣ ΟΛΩΝ ΤΩΝ ΣΤΟΙΧΕΙΩΝ**  ΑΘΡ←0  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΘΡ← ΑΘΡ+Α[I,J]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ ΑΘΡ  ΜΟ← ΑΘΡ/(Ν\*M)  ΓΡΑΨΕ ΜΟ |
| **ΑΘΡΟΙΣΜΑ /ΜΕΣΟΣ ΟΡΟΣ ΑΝΑ ΓΡΑΜΜΗ**  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΑΘΡ[I]← 0  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΘΡ[I]← ΑΘΡ[I]+Α[I,J]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  MO[I] ← ΑΘΡ[I]/M  ΓΡΑΨΕ MO[I]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **ΑΘΡΟΙΣΜΑ /ΜΕΣΟΣ ΟΡΟΣ ΑΝΑ ΣΤΗΛΗ**  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M  ΑΘΡ[J]← 0  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N  ΑΘΡ[J]← ΑΘΡ[J]+Α[I,J]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  MO[J] ← ΑΘΡ[J]/N  ΓΡΑΨΕ MO[J]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |
| **ΠΛΗΘΟΣ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ ΑΝΑ ΓΡΑΜΜΗ**  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΠΛΗΘΟΣ[I]←0  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Α[Ι]>50 ΤΟΤΕ !π.χ. ψάχνω στοιχεία>50  ΠΛΗΘΟΣ[I]←ΠΛΗΘΟΣ[I]+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ ΠΛΗΘΟΣ[I]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **ΜΕΓΙΣΤΟ/ΕΛΑΧΙΣΤΟ ΟΛΩΝ ΤΩΝ ΣΤΟΙΧΕΙΩΝ**  MAX←A[1,1]  ΓΡΑΜΜΗ←1  ΣΤΗΛΗ←1  MIN←A[1,1]  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Α[Ι,J]>MAX ΤΟΤΕ  MAX←A[I,J]  ΓΡΑΜΜΗ← I  ΣΤΗΛΗ←J  ΤΕΛΟΣ\_ΑΝ  ΑΝ Α[Ι,J]<MIN ΤΟΤΕ  MIN←A[I,J]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ MAX,ΓΡΑΜΜΗ, ΣΤΗΛΗ, MIN |
| **ΜΕΓΙΣΤΟ/ΕΛΑΧΙΣΤΟ ΑΝΑ ΓΡΑΜΜΗ**  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Ν  MAX[I]←A[I,1]  MIN[I]←A[I,1]  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  ΑΝ Α[Ι,J]>MAX[I] ΤΟΤΕ  MAX[I]←A[I,J]  ΤΕΛΟΣ\_ΑΝ  ΑΝ Α[Ι,J]<MIN[I] ΤΟΤΕ  MIN[I]←A[I,J]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ MAX[I], MIN[I]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **ΜΕΓΙΣΤΟ/ΕΛΑΧΙΣΤΟ ΑΝΑ ΣΤΗΛΗ**  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ M  MAX[J]←A[1,J]  MIN[J]←A[1,J]  ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N  ΑΝ Α[Ι,J]>MAX[J] ΤΟΤΕ  MAX[J]←A[I,J]  ΤΕΛΟΣ\_ΑΝ  ΑΝ Α[Ι,J]<MIN[J] ΤΟΤΕ  MIN[J]←A[I,J]  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΡΑΨΕ MAX[J], MIN[J]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |

**Σημείωση: τετραγωνικός πίνακας:** ο πίνακας που έχει ίδιο αριθμό γραμμών και στηλών Α[Ν,Ν]

* για να ελέγξω αν είναι στοιχείο της κύριας διαγώνιου: **i=j**
* για να ελέγξω αν είναι στοιχείο της δευτερεύουσας διαγώνιου: **i+j=Ν+1** (όπου Ν η διάσταση του πίνακα)
* για να ελέγξω αν είναι στοιχείο πάνω από την κύρια διαγώνιο: **i<j**
* για να ελέγξω αν είναι στοιχείο κάτω από την κύρια διαγώνιο: **i>j**

**ΔΥΝΑΜΙΚΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ:** το μέγεθος της μνήμης μεταβάλλεται κατά την εκτέλεση του προγράμματος. Οι κόμβοι **μπορούν** να αποθηκεύονται σε μη συνεχόμενες θέσεις μνήμης.

**(Α) ΣΤΟΙΒΑ:** είναι μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία.

Η παραπάνω μέθοδος ονομάζεται **Τελευταίο Μέσα, Πρώτο Έξω (LIFO)**

|  |  |
| --- | --- |
| **ΩΘΗΣΗ ΣΤΟΙΧΕΙΟΥ ΣΕ ΣΤΟΙΒΑ**  top←0  ΑΝ top=N ΤΟΤΕ  ΓΡΑΨΕ ‘ΥΠΕΡΧΕΙΛΙΣΗ ΣΤΟΙΒΑΣ’  ΑΛΛΙΩΣ  ΔΙΑΒΑΣΕ στοιχείο  top←top+1  Α[top] ← στοιχείο  ΤΕΛΟΣ\_ΑΝ | **ΑΠΩΘΗΣΗ ΣΤΟΙΧΕΙΟΥ ΑΠΟ ΣΤΟΙΒΑ**  ΑΝ top>=1 ΤΟΤΕ  ΓΡΑΨΕ Α[top]  top← top-1  ΑΛΛΙΩΣ  ΓΡΑΨΕ ‘ΥΠΟΧΕΙΛΙΣΗ ΣΤΟΙΒΑΣ’  ΤΕΛΟΣ\_ΑΝ |

**(B) ΟΥΡΑ:** είναι μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα. Η παραπάνω μέθοδος ονομάζεται **Πρώτο Μέσα, Πρώτο Έξω (FIFO)**

|  |  |
| --- | --- |
| **ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΟΥ ΣΕ ΟΥΡΑ**  front←0  rear←0  ΑΝ rear = N TOTE  ΓΡΑΨΕ 'Γεμάτη ουρά'  ΑΛΛΙΩΣ  ΔΙΑΒΑΣΕ στοιχείο  ΑΝ (front = 0 KAI rear = 0) ΤΟΤΕ  front← 1  rear←1  ΑΛΛΙΩΣ  rear ← rear + 1  ΤΕΛΟΣ\_ΑΝ  A[rear]←στοιχείο  ΤΕΛΟΣ\_ΑΝ | **ΕΞΑΓΩΓΗ ΣΤΟΙΧΕΙΟΥ ΑΠΟ ΟΥΡΑ**  ΑΝ (front = 0 KAI rear = 0) ΤΟΤΕ  ΓΡΑΨΕ ‘Άδεια ουρά'  ΑΛΛΙΩΣ  ΓΡΑΨΕ 'Εξάγεται το στοιχείο:', A[front]  ΑΝ (front = rear ) ΤΟΤΕ  front ← 0  rear ← 0  ΑΛΛΙΩΣ  front ← front + 1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΑΝ |
|  |  |

**(Γ) Απλή Συνδεδεμένη λίστα** είναι ένα σύνολο κόμβων διατεταγμένων γραμμικά (ο ένας μετά τον άλλο). Κάθε κόμβος περιέχει εκτός από τα **δεδομένα** του και έναν **δείκτη** που δείχνει προς τον επόμενο κόμβο. Ο δείκτης του τελευταίου κόμβου δε δείχνει σε κάποιον κόμβο (δείκτης στο κενό). Για να το δηλώσουμε αυτό λέμε ότι το πεδίο δείκτη του τελευταίου κόμβου έχει την τιμή **NULL**. Για να προσπελάσουμε τους κόμβους της λίστας χρειάζεται να γνωρίζουμε τη διεύθυνση

του πρώτου κόμβου της λίστας που αποθηκεύεται σε μία ειδική μεταβλητή που την ονομάζουμε **Κεφαλή.**

**Διπλή Συνδεδεμένη λίστα:** όταν μπορούμε να διατρέξουμε τη λίστα και προς τις 2 κατευθύνσεις

**(Δ) Δένδρο:** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων και ένα σύνολο ακμών μεταξύ των κόμβων με βάση τους εξής κανόνες:

• Υπάρχει ένας ξεχωριστός κόμβος που ονομάζεται ρίζα. Αυτός είναι ένας κόμβος χωρίς γονέα.

• Για κάθε κόμβο c, εκτός από τη ρίζα, υπάρχει μόνο μια ακμή που καταλήγει στον κόμβο αυτόν

ξεκινώντας από κάποιον άλλον κόμβο p. Ο κόμβος p ονομάζεται γονέας του c και ο κόμβος c

παιδί του p.

• Για κάθε κόμβο υπάρχει μία μοναδική διαδρομή, δηλαδή, μια ακολουθία διαδοχικών ακμών,

που ξεκινάει από τη ρίζα και τερματίζει σε αυτόν τον κόμβο.

Σημείωση: Δένδρο θεωρούμε και το **κενό δένδρο**, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

**Δυαδικό δένδρο**: για κάθε κόμβο u, όλοι οι κόμβοι του αριστερού υποδένδρου έχουν τιμές μικρότερες της τιμής του κόμβου u και όλοι οι κόμβοι του δεξιού υποδένδρου έχουν τιμές μεγαλύτερες της τιμής του κόμβου u.

**(Ε) Γράφος:** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων (ή σημείων ή κορυφών) και ένα σύνολο γραμμών (ή ακμών ή τόξων) που ενώνουν μερικούς ή όλους τους κόμβους.

**ΜΕΘΟΔΟΛΟΓΙΑ ΑΝΑΖΗΤΗΣΗΣ**

**(α)** Αν ζητάει να βρούμε αν υπάρχει κάποιο στοιχείο σε ένα πίνακα A[N], χρησιμοποιούμε **σειριακή αναζήτηση** (προτιμάται σε αταξινόμητο πίνακα)

|  |  |
| --- | --- |
| **(i) Σε μονοδιάστατο πίνακα A[N]**  ΔΙΑΒΑΣΕ στοιχείο  βρέθηκε← ψευδής  i← 1  θέση<-0  ΟΣΟ βρέθηκε = ψευδής ΚΑΙ i< =N ΕΠΑΝΑΛΑΒΕ  ΑΝ Α[i]= στοιχείο ΤΟΤΕ  βρέθηκε← αληθής  θέση← i  ΑΛΛΙΩΣ  i← i+1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε = ψευδής ΤΟΤΕ  ΓΡΑΨΕ ‘Δε βρέθηκε το στοιχείο’  ΑΛΛΙΩΣ  ΓΡΑΨΕ θέση  ΤΕΛΟΣ\_ΑΝ | **(ii) Σε δισδιάστατο πίνακα A[N,M]**  ΔΙΑΒΑΣΕ στοιχείο  βρέθηκε← ψευδής  I ← 1  ΟΣΟ βρέθηκε = ψευδής ΚΑΙ I< =N ΕΠΑΝΑΛΑΒΕ  J ←1  ΟΣΟ J<=M ΚΑΙ ΒΡΕΘΗΚΕ=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ  ΑΝ Α[I,J]= στοιχείο ΤΟΤΕ  βρέθηκε← αληθής  γραμμή ← I  στήλη← J  ΑΛΛΙΩΣ  J←J+1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  I← I+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε = ψευδής ΤΟΤΕ  ΓΡΑΨΕ ‘Δε βρέθηκε το στοιχείο’  ΑΛΛΙΩΣ  ΓΡΑΨΕ γραμμή, στήλη  ΤΕΛΟΣ\_ΑΝ |

|  |  |
| --- | --- |
| **(β)** **σειριακής αναζήτηση αγνώστου πλήθους επαναλήψεων**: τοποθετούμε τον αλγόριθμο σειριακής αναζήτησης μέσα σε δομή ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ με ένα κριτήριο τερματισμού | **(γ) Βελτιστοποίηση του αλγόριθμου σειριακής αναζήτησης (δεδομένου ότι ο πίνακας είναι ταξινομημένος)**  ΔΙΑΒΑΣΕ στοιχείο  ΒΡΕΘΗΚΕ← ψευδής  i← 1  ΘΕΣΗ<-0  ΟΣΟ ΒΡΕΘΗΚΕ = ψευδής ΚΑΙ i< =N ΕΠΑΝΑΛΑΒΕ  ΑΝ Α[i]= στοιχείο ΤΟΤΕ  ΒΡΕΘΗΚΕ← ΑΛΗΘΗΣ  ΘΕΣΗ<-Ι  ΑΛΛΙΩΣ\_ΑΝ Α[Ι]>ΣΤΟΙΧΕΙΟ ΤΟΤΕ  ΒΡΕΘΗΚΕ<- ΑΛΗΘΗΣ  ΑΛΛΙΩΣ  i← i+1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ ΒΡΕΘΗΚΕ=ΑΛΗΘΗΣ ΚΑΙ ΘΕΣΗ <>0 ΤΟΤΕ  ΓΡΑΨΕ ΘΕΣΗ  ΑΛΛΙΩΣ  ΓΡΑΨΕ ‘ΔΕ ΒΡΕΘΗΚΕ’  ΤΕΛΟΣ\_ΑΝ |

**(δ)** Αν ζητάει να βρούμε το στοιχείο μετά από συγκεκριμένο πλήθος εμφανίσεων μέσα στον πίνακα: τότε μονό Όσο αν πρόκειται για μονοδιάστατο πίνακα ή διπλό Όσο αν πρόκειται για δισδιάστατο πίνακα και προσθέτουμε ένα μετρητή που κρατάει το πλήθος των εμφανίσεων του στοιχείου στον πίνακα. Όταν το στοιχείο εμφανιστεί τις φορές που θέλουμε τότε το **βρέθηκε** γίνεται αληθής. Π.χ. θέλω να βρω τη θέση του στοιχείου στον πίνακα(Ν) την 3η φορά που εμφανίζεται

|  |  |
| --- | --- |
| **(i) Σε μονοδιάστατο πίνακα A[N]**  ΔΙΑΒΑΣΕ στοιχείο  πλήθος←0  βρέθηκε ← ψευδής  Ι<-1  ΟΣΟ Ι<=Ν ΚΑΙ βρέθηκε=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ  ΑΝ Α[I]=στοιχείο ΤΟΤΕ  πλήθος←πλήθος+1  ΑΝ πλήθος=3 ΤΟΤΕ  βρέθηκε← αληθής  θέση← I  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΑΝ  Ι<- Ι+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε= αληθής ΤΟΤΕ  ΓΡΑΨΕ θέση  ΑΛΛΙΩΣ  ΓΡΑΨΕ ‘Το στοιχείο δε βρέθηκε’  ΤΕΛΟΣ\_ΑΝ | **(ii) Σε δισδιάστατο πίνακα A[N,M]**  ΔΙΑΒΑΣΕ στοιχείο  πλήθος←0  βρέθηκε ← ψευδής  Ι<-1  ΟΣΟ I<=N ΚΑΙ βρέθηκε=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ  J<-1  ΟΣΟ J<=M ΚΑΙ βρέθηκε=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ  ΑΝ Α[I,J]=στοιχείο ΤΟΤΕ  πλήθος←πλήθος+1  ΑΝ πλήθος=3 ΤΟΤΕ  βρέθηκε← αληθής  γραμμή← I  στήλη←J  ΤΕΛΟΣ\_ΑΝ  ΑΛΛΙΩΣ  J<- J+1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  Ι<-Ι+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ βρέθηκε= αληθής ΤΟΤΕ  ΓΡΑΨΕ γραμμή, στήλη  ΑΛΛΙΩΣ  ΓΡΑΨΕ ‘Το στοιχείο δε βρέθηκε’  ΤΕΛΟΣ\_ΑΝ |

**(ε)** Αν ζητάει να βρούμε αν υπάρχει ένα στοιχείο σε **ταξινομημένο** πίνακα χρησιμοποιούμε τη **δυαδική αναζήτηση** γιατί είναι βέλτιστος αλγόριθμος σε σχέση με τη σειριακή αναζήτηση.

Σημείωση:

(α) Ο αλγόριθμος δυαδικής αναζήτησης είναι ταχύτερος από την απλή σειριακή αναζήτηση.

(β) Ο αλγόριθμος της Δυαδικής Αναζήτησης μπορεί να εφαρμοστεί μόνο σε ταξινομημένο πίνακα

Ο **αλγόριθμος της δυαδικής αναζήτησης** για την εύρεση ενός στοιχείου Χ σε έναν ταξινομημένο πίνακα π.χ. Α[Ν]

|  |
| --- |
| ΔΙΑΒΑΣΕ στοιχείο  πρώτη←1  τελευταία ←Ν  θέση←0  ΟΣΟ πρώτη<=τελευταία και θέση=0 ΕΠΑΝΑΛΑΒΕ  μεσαία← (πρώτη+τελευταία) div 2  ΑΝ Α[μεσαία]= στοιχείο ΤΟΤΕ  θέση←μεσαία  ΑΛΛΙΩΣ\_ΑΝ Α[μεσαια] < στοιχείο ΤΟΤΕ  πρώτη← μεσαία+1  ΑΛΛΙΩΣ  τελευταία ←μεσαία-1  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΝ θέση < >0 ΤΟΤΕ  ΓΡΑΨΕ θέση  ΑΛΛΙΩΣ  ΓΡΑΨΕ ‘Δε βρέθηκε το στοιχείο’  ΤΕΛΟΣ\_ΑΝ |

**ΤΑΞΙΝΟΜΗΣΗ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ**

|  |  |
| --- | --- |
| **//μετατρέπω τον δισδιάστατο σε μονοδιάστατο με χρήση βοηθητικής μεταβλητής που κρατάει το μέγεθος του μονοδιάστατου πίνακα**  k←0  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ Μ  k←k+1  Α[k] ← Γ[i, j]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  **//ταξινομώ τον μονοδιάστατο πίνακα**  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ Κ  ΓΙΑ j ΑΠΟ Κ ΜΕΧΡΙ i ΜΕ\_ΒΗΜΑ -1  ΑΝ Α[j-1] > Α[j] ΤΟΤΕ  temp←Α[j-1]  Α[j-1] ←Α[j]  Α[j] ←temp  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **//τον μετατρέπω σε δισδιάστατο πίνακα**  k←0  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Ν  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Μ  k←k+1  Γ[i, j] ← Α[k]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |

**ΤΑΞΙΝΟΜΗΣΗ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ ΑΝΑ ΣΤΗΛΗ / ΑΝΑ ΓΡΑΜΜΗ**

|  |  |
| --- | --- |
| **ΤΑΞΙΝΟΜΗΣΗ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ ΑΝΑ ΣΤΗΛΗ**  ΓΙΑ ΣΤΗΛΗ ΑΠΟ 1 ΜΕΧΡΙ Μ !για κάθε στήλη  ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν !κάνε ταξινόμηση  ΓΙΑ J ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ\_ΒΗΜΑ -1  ΑΝ Α[J-1,ΣΤΗΛΗ ]>Α[J, ΣΤΗΛΗ] TOTE  TEMP<- Α[J-1,ΣΤΗΛΗ ]  Α[J-1,ΣΤΗΛΗ ]<- Α[J,ΣΤΗΛΗ ]  Α[J,ΣΤΗΛΗ]<- TEMP  TEΛΟΣ\_ΑΝ  ΤΕΛΟΣ-ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **ΤΑΞΙΝΟΜΗΣΗ ΔΙΣΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ ΑΝΑ ΓΡΑΜΜΗ**  ΓΙΑ ΓΡΑΜΜΗ ΑΠΟ 1 ΜΕΧΡΙ Ν !για κάθε γραμμή  ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Μ !κάνε ταξινόμηση  ΓΙΑ J ΑΠΟ Μ ΜΕΧΡΙ Ι ΜΕ\_ΒΗΜΑ -1  ΑΝ Α[ΓΡΑΜΜΗ, J-1]>Α[ΓΡΑΜΜΗ, J] TOTE  TEMP<- Α[ΓΡΑΜΜΗ, J-1]  Α[ΓΡΑΜΜΗ, J-1]<- Α[ΓΡΑΜΜΗ, J]  Α[ΓΡΑΜΜΗ, J]<- ΤΕΜP  TEΛΟΣ\_ΑΝ  ΤΕΛΟΣ-ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |

**ΣΥΓΧΩΝΕΥΣΗ /ΔΙΑΧΩΡΙΣΜΟΣ ΠΙΝΑΚΩΝ**

**(α) συγχώνευση σε ταξινομημένους πίνακες: Α[Ν] και Β[Μ] σε C[Ν+Μ]**

|  |  |
| --- | --- |
| I ←1 **//δείκτης πίνακα Α**  J ←1 **//δείκτης πίνακα B**  K ←1 **//δείκτης πίνακα C**  ΟΣΟ I<=Ν ΚΑΙ J<=Μ ΕΠΑΝΑΛΑΒΕ  ΑΝ A[I]<=B[J] ΤΟΤΕ  C[K]← A[I]  I← I+1  ΑΛΛΙΩΣ  C[K]← B[J]  J← J+1  ΤΕΛΟΣ\_ΑΝ  K← K+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | **//τοποθέτηση των υπόλοιπων στοιχείων του πίνακα από τον οποίο περίσσεψαν στον C**  ΑΝ Ι<=Ν ΤΟΤΕ !περίσσεψαν από Α  ΓΙΑ Λ ΑΠΟ Κ ΜΕΧΡΙ Ν+Μ  C[Λ]← Α[Ι]  I← I+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΑΛΛΙΩΣ !περίσσεψαν από Β  ΓΙΑ Λ ΑΠΟ Κ ΜΕΧΡΙ Ν+Μ  C[Λ]← Β[J]  J← J+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΑΝ |

**(β) συγχώνευση σε μη ταξινομημένους πίνακες: Α[Ν] και Β[Μ] σε C[Ν+Μ]**

|  |
| --- |
| ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν  C[Ι]← A[I]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΙΑ Ι ΑΠΟ Ν+1 ΜΕΧΡΙ Ν+Μ  C[Ι]← Β[I-Ν]  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |

**(γ) Διαχωρισμός πινάκα C[Ν+Μ] σε Α[Ν] και Β[Μ]**

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν

A[I] ←C[Ι]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ Ι ΑΠΟ Ν+1 ΜΕΧΡΙ Ν+Μ

Β[I-Ν] ← C[Ι]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

**ΤΑΞΙΝΟΜΗΣΗ**

**(Α) Ταξινόμηση Ευθείας Ανταλλαγής ή Φυσαλίδας (μόνο σε μονοδιάστατο πίνακα)**

|  |  |
| --- | --- |
| ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν  ΓΙΑ J ΑΠΟ Ν ΜΕΧΡΙ I ΜΕ\_ΒΗΜΑ -1  ΑΝ Α[J-1]> A[J] ΤΟΤΕ //αύξουσα ταξινόμηση  TEMP← A[J-1]  A[J-1]← A[J]  A[J] ← TEMP  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |  |

**(Β) Παραλλαγή της Ταξινόμησης Ευθείας Ανταλλαγής ή Φυσαλίδας: (**Σταματάει όταν ο πίνακας είναι ταξινομημένος άρα γίνεται βέλτιστος ο αλγόριθμος)

|  |  |
| --- | --- |
| ΤΑΞΙΝΟΜΗΜΕΝΟΣ← ΨΕΥΔΗΣ  Ι ←2  ΟΣΟ Ι<=Ν ΚΑΙ ΤΑΞΙΝΟΜΗΜΕΝΟΣ= ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ  ΤΑΞΙΝΟΜΗΜΕΝΟΣ←ΑΛΗΘΗΣ  ΓΙΑ J ΑΠΟ Ν ΜΕΧΡΙ Ι ΜΕ\_ΒΗΜΑ -1  ΑΝ Α[J-1]> A[J] ΤΟΤΕ //αύξουσα ταξινόμηση  TEMP← A[J-1]  A[J-1]← A[J]  A[J] ← TEMP  ΤΑΞΙΝΟΜΗΜΕΝΟΣ←ΨΕΥΔΗΣ  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  Ι←Ι+1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Η Μεταβλητή ΤΑΞΙΝΟΜΗΜΝΕΟΣ ελέγχει (όταν αλλάξει η τιμή της) από ποιο σημείο και μετά ο πίνακας είναι ταξινομημένος και σταματάει η διαδικασία ταξινόμησης.  Όταν δεν ισχύει η συνθήκη σημαίνει ότι το επόμενο στοιχείο είναι μικρότερο ή ίσο του προηγούμενου, δηλαδή βρέθηκε το σημείο από το οποίο και μετά ο πίνακας είναι ταξινομημένος.  Η ΤΑΞΙΝΟΜΗΜΕΝΟΣ από ΨΕΥΔΗΣ γίνεται ΑΛΗΘΗΣ και τερματίζει η διαδικασία ταξινόμησης. |

**(Γ) Ταξινόμηση με Εισαγωγή (Insertion Sort)**

Λειτουργεί ως εξής:

1. Συγκρίνω ανά δύο τα στοιχεία του πίνακα, ξεκινώντας από το δεύτερο στοιχείο (έστω Α[2]

με το Α[1]).

2. Επιλέγεται το στοιχείο με τη μικρότερη τιμή και ανταλλάζεται η θέση του με το άλλο.

3. Επαναλαμβάνω αυτά τα βήματα για τον υπόλοιπο πίνακα. Κάθε στοιχείο που εξετάζεται συγκρίνεται διαδοχικά με τα προηγούμενά του και αν είναι μικρότερο από κάποιο προηγούμενο ανταλλάσσεται με αυτό.

|  |  |
| --- | --- |
| 1. ΓΙΑ Ι ΑΠΟ 2 ΜΕΧΡΙ Ν  2. J←I -1  3. STOP← ΨΕΥΔΗΣ  4. ΟΣΟ (STOP=ΨΕΥΔΗΣ) ΚΑΙ (J>0) ΕΠΑΝΑΛΑΒΕ  5. STOP← ΑΛΗΘΗΣ  6. ΑΝ Α[J+1] < Α[J] ΤΟΤΕ  7. TEMP ← A[J+1]  8. A[J+1] ← A[J]  9. A[J] ← TEMP  10. J← J -1  11. STOP ← ΨΕΥΔΗΣ  12. ΤΕΛΟΣ\_ΑΝ  13. ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  14. ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Επεξηγήση:  Η λογική μεταβλητή STOP, που χρησιμοποιείται στα βήματα 3,4,5,11, έχει ως στόχο να εμποδίσει το δείκτη J να βγει εκτός ορίων (δηλαδή να γίνει 0), καθώς στο βήμα 9 μειώνεται κατά 1 και έτσι να εμποδίσει την ατέρμονη εκτέλεση του βρόχου στο βήμα 4. |

**(Δ) Ταξινόμηση με Επιλογή (Selection Sort)**

|  |  |
| --- | --- |
| ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν-1 I=1  min\_θεση ← Ι MIN\_ΘΕΣΗ=3  ΓΙΑ Κ ΑΠΟ Ι+1 ΜΕΧΡΙ Ν Κ=10  ΑΝ Α[Κ] < Α[min\_θεση] ΤΟΤΕ  min\_θεση ← Κ  ΤΕΛΟΣ\_ΑΝ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  TEMP ← A[I]  A[I] ← A[min\_θεση]  A[min\_θεση] ← TEMP  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ | Επεξήγηση**:**  Ο δείκτης min\_θεση δείχνει τη θέση που περιέχει το μικρότερο στοιχείο σε κάθε πέρασμα το οποίο και αντιμετατίθεται με το τρέχον στοιχείο Α[Ι]. |

**ΠΙΝΑΚΑΣ ΣΥΧΝΟΤΗΤΩΝ**

**Παράδειγμα**: έστω ένας πίνακας με βαθμούς 20 φοιτητών σε 40 μαθήματα (κλίμακα 1-10).Συχνότητα εμφάνισης κάθε βαθμού για κάθε φοιτητή ξεχωριστά.

|  |
| --- |
| ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20 ! για κάθε φοιτητή  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10 !για κάθε βαθμό  ΣΥΧΝ[I,J] **←**0  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 20 !για κάθε φοιτητή  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 40 !για κάθε μάθημα  ΣΥΧΝ[Ι, Β[Ι,J]] **←**ΣΥΧΝ[I, B[I,J]] +1  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ |

**ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ**

Οι **παράμετροι** είναι μεταβλητές που επιτρέπουν την ανταλλαγή τιμών μεταξύ προγραμμάτων και υποπρογραμμάτων

|  |  |
| --- | --- |
| **(Α) ΔΙΑΔΙΚΑΣΙΑ:** κάνει ότι κάνει και ένα πρόγραμμα  Σύνταξη**:**  **ΔΙΑΔΙΚΑΣΙΑ** όνομα\_διαδικασίας (τυπικές παράμετροι)  **ΣΤΑΘΕΡΕΣ**  …  **ΜΕΤΑΒΛΗΤΕΣ**  …  **ΑΡΧΗ**  εντολές  **ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**  Κλήση διαδικασίας από κυρίως πρόγραμμα  **ΚΑΛΕΣΕ** όνομα\_διαδικασίας (πραγματικές παράμετροι) | **(Β) ΣΥΝΑΡΤΗΣΗ:** υπολογίζει και επιστρέφει μόνο μία τιμή με το όνομα της στο πρόγραμμα που την κάλεσε  Σύνταξη**:**  **ΣΥΝΑΡΤΗΣΗ** όνομα\_συνάρτησης (τυπικές παράμετροι): τύπος δεδομένων συνάρτησης  **ΣΤΑΘΕΡΕΣ**  …  **ΜΕΤΑΒΛΗΤΕΣ**  …  **ΑΡΧΗ**  Εντολές  Όνομα\_συνάρτησης← έκφραση  **ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**  Κλήση συνάρτησης από κυρίως πρόγραμμα  Καλείται με το όνομα της συνάρτησης μέσα σε μία εντολή  **όνομα\_συνάρτησης (πραγματικές παράμετροι)** |

**Μεθοδολογία Αντικειμενοστραφούς προγραμματισμού**

(α) Αναγνωρίζουμε και καταγράφουμε τα βασικά συστατικά στοιχεία της διαδικασίας επίλυσής του

1. τα **αντικείμενα** που συμμετέχουν με βάση τον ρόλο τους στο συγκεκριμένο σενάριο,

του,

2. οι **ιδιότητες** κάθε αντικειμένου,

3. οι **υπηρεσίες** που προσφέρει ή οι **ενέργειες** που υλοποιεί κάθε αντικείμενο (μέθοδοι) προς

αξιοποίηση από άλλες, ώστε να αναπτυχθούν οι απαραίτητες **συνεργασίες** μεταξύ των αντικειμένων για την επίλυση του προβλήματος.

(β) Ο γενικός τύπος ενός αντικειμένου καλείται **κλάση** και καθορίζει τις αρχικές ιδιότητες και τη συμπεριφορά κάθε αντικειμένου που προέρχεται από αυτή.

**Πολυμορφισμός** είναι μια ιδιότητα του αντικειμενοστραφούς προγραμματισμού με την οποία μια λειτουργία μπορεί να υλοποιείται με πολλούς διαφορετικούς τρόπους.

**Κατηγορίες Λαθών**

(α) Συντακτικά Λάθη

(β) Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος

(γ) Λογικά λάθη

**Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής**

Σε μια δομή επιλογής μπορεί να εμφανιστούν λογικά λάθη που σχετίζονται με:

• τη συνθήκη ή τις συνθήκες

• τις ομάδες εντολών που εκτελούνται όταν μια συνθήκη είναι αληθής ή ψευδής.

**Εκσφαλμάτωση λογικών λαθών στις δομές επανάληψης**

Μπορεί να εμφανιστούν λογικά λάθη που σχετίζονται με:

• τους συγκριτικούς και τους λογικούς τελεστές των συνθηκών επανάληψης ή τερματισμού

• την αρχικοποίηση της συνθήκης

• την ενημέρωση της συνθήκης εντός του βρόχου

• την αλληλουχία των εντολών του βρόχου και στη σειρά εκτέλεσής τους

• το κριτήριο της περατότητας

• την πρώτη επανάληψη και την περίπτωση που ο βρόχος επανάληψης δεν πρέπει να

εκτελεστεί ούτε μία φορά

• την τελευταία επανάληψη.

**Εκσφαλμάτωση λογικών λαθών σε πίνακες:**

• στο μέγεθος των πινάκων κατά τη δήλωσή τους,

• στους δείκτες των πινάκων κατά την προσπέλασή τους,

• στη μη υπέρβαση των ορίων του πίνακα.

**Εκσφαλμάτωση λογικών λαθών στα υποπρογράμματα**:

• την κλήση του υποπρογράμματος και το πέρασμα των παραμέτρων

• τα λοιπά λογικά λάθη που εμφανίζονται και στα προγράμματα.