

ΥΠΟΥΡΓΕΙΟ ΠΟΛΙΤΙΣΜΟΥ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Κωτσάκης Σ., Ταταράκη Α.

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ
ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ
ΠΕΡΙΒΑΛΛΟΝ

Γ' Τάξη ΓΕ.Λ.

ΠΑΡΑΡΤΗΜΑ Α
ΟΔΗΓΙΕΣ ΜΕΛΕΤΗΣ ΜΑΘΗΤΗ

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΕΚΔΟΣΕΩΝ
«ΔΙΟΦΑΝΤΟΣ»

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Πρόεδρος: Σωτήριος Γκλαβάς

ΓΡΑΦΕΙΟ ΕΡΕΥΝΑΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΕΦΑΡΜΟΓΩΝ Β΄

Προϊστάμενος: Παύλος Μάραντος

ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υπεύθυνος: Θεοδόσιος Τσαπέλας, Σύμβουλος Β΄ Πληροφορικής ΙΕΠ

ΣΥΓΓΡΑΦΕΙΣ:

Κωτσάκης Σταύρος, Σχολικός σύμβουλος πληροφορικής-ΠΕ19 Πελοποννήσου
Ταταράκη Αλεξάνδρα, Καθηγήτρια ΠΕ19

ΣΥΝΤΟΝΙΣΤΗΣ

Κωτσάκης Σταύρος, Σχολικός σύμβουλος πληροφορικής-ΠΕ19 Πελοποννήσου

ΠΡΟΕΚΤΥΠΩΤΙΚΕΣ ΕΡΓΑΣΙΕΣ: ΔΙΕΥΘΥΝΣΗ ΕΚΔΟΣΕΩΝ/Ι.Τ.Υ.Ε. «ΔΙΟΦΑΝΤΟΣ»

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	5
1. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ.....	6
1.1. Η εντολή επανάληψης ΟΣΟ,,ΕΠΑΝΑΛΑΒΕ	6
1.1.1. Χαρακτηριστικά της εντολής;	7
1.1.2. Πως συντάσσεται:	7
1.1.3. Που χρησιμοποιείται:	8
1.1.4. Παραδείγματα	8
1.2. Η εντολή επανάληψης ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ.....	9
1.2.1. Σύνταξη:.....	9
1.2.2. Χαρακτηριστικά της εντολής;	9
1.2.3. Που χρησιμοποιείται:	9
1.2.4. Παραδείγματα	10
1.3. Η εντολή επανάληψης ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ	15
1.3.1. Τρόπος εκτέλεσης.....	15
1.3.2. Παραδείγματα	16
1.3.3. Παρατηρήσεις	16
1.3.4. Εμφωλευμένοι Βρόχοι.....	16
1.4. Αλγόριθμος ελάχιστου ή μεγίστου από μια ομάδα αριθμών.....	17
1.4.1. Παραδείγματα	18
1.5. Μετατροπές από μια δομή επανάληψης σε άλλη.....	23
1.5.1. Μετατροπή από ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ ... ΣΕ ΟΣΟ...ΚΑΙ ΑΝΤΙΣΤΡΟΦΩΣ	23
1.5.1.1. Παραδείγματα	23
1.5.2. Μετατροπή απο ΓΙΑ... σε ΟΣΟ... ή σε ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ και αντιστρόφως. 24	
1.5.2.1. Παραδείγματα	25
2. ΠΙΝΑΚΕΣ.....	27
2.1. Πλεονεκτήματα χρήσης πινάκων:	27
2.2. Μειονεκτήματα χρήσης πινάκων:	27
2.3. Που χρησιμοποιούνται οι πίνακες;	27
2.4. Δήλωση πίνακα στη ΓΛΩΣΣΑ.	28
2.5. Είσοδος δεδομένων σε πίνακα	29
2.5.1. Σε μονοδιάστατο πίνακα	29
2.5.2. Σε διδιάστατο πίνακα.....	30
2.5.3. Παραδείγματα	30

2.6.	Τυπικές επεξεργασίες πινάκων	37
2.6.1.	Ταξινόμηση πίνακα	37
2.6.1.1.	Ταξινόμηση ευθείας ανταλλαγής ή φυσαλίδας (Bubblesort).....	37
2.6.2.	Αναζήτηση στοιχείου πίνακα	39
2.6.2.1.	Σειριακή αναζήτηση	39
2.6.2.2.	Διαδική αναζήτηση.....	40
2.7.	Αλγόριθμος συγχώνευσης δύο πινάκων.....	43
2.8.	Παράλληλοι πίνακες.....	45
2.9.	Λυμένες ασκήσεις	49
2.9.1.	Άσκηση 1. Μονοδιάστατος πίνακας.....	49
2.9.2.	Άσκηση 2. Δισδιάστατος πίνακας.....	53
2.9.3.	Άσκηση 3	58
3.	ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ-ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ (ΔΙΑΔΙΚΑΣΙΕΣ-ΣΥΝΑΡΤΗΣΕΙΣ)	61
3.1.	Που χρησιμοποιούνται τα υποπρογράμματα;.....	62
3.2.	Τι πρέπει να προσεχθεί;	62
3.3.	Διαδικασίες (Παράδειγμα διαδικασίας)	64
3.3.1.	Πως συντάσσεται (ορίζεται) μια διαδικασία;	66
3.3.2.	Πως το κύριο πρόγραμμα καλεί μια διαδικασία;	66
3.3.3.	Πως εκτελείται το πρόγραμμα και πως λειτουργεί η επικοινωνία του κύριου προγράμματος με τη διαδικασία;	67
3.3.3.1.	Σημεία στα οποία πρέπει να δοθεί ιδιαίτερη σημασία.	71
3.3.4.	Κλήση Διαδικασίας από Διαδικασία	73
3.3.5.	Διαδικασίες και Πίνακες.....	75
3.4.	Συναρτήσεις (Παράδειγμα συνάρτησης)	77
3.4.1.	Πως συντάσσεται και πως καλείται μια συνάρτηση;.....	78
3.4.1.1.	Σύνταξη - ορισμός συνάρτησης.....	78
3.4.1.2.	Κλήση συνάρτησης.....	78
3.4.2.	Σε τι διαφέρουν οι διαδικασίες από τις συναρτήσεις;	79
3.4.3.	Διαδικασίες και Συναρτήσεις Μαζί	82

ΠΡΟΛΟΓΟΣ

Το μάθημα της Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον, εισήχθη στην Γ΄ τάξη του Γ.Ε.Λ., το σχολικό έτος 1999 – 2000. Από το πρώτο εκπαιδευτικό έτος διδασκαλίας του (1999-2000), μέχρι σήμερα, έχει παρατηρηθεί δυσκολία στην κατανόηση βασικών εννοιών προγραμματισμού. Το παρόν πρόσθετο εκπαιδευτικό υλικό αναπτύχθηκε με σκοπό να συμβάλλει στην κατανόηση αυτών των εννοιών και να διευκολύνει στη μάθησή τους. Είναι απαραίτητο να διευκρινισθεί ότι δεν υποκαθιστά και πολύ περισσότερο δεν αντικαθιστά το Βιβλίο Μαθητή και το Τετράδιο Εργασιών Μαθητών αλλά φιλοδοξεί να λειτουργήσει αποκλειστικά επικουρικά σε αυτά.

1. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

Η «ΓΛΩΣΣΑ» υποστηρίζει τρεις εντολές επανάληψης:

Την εντολή **ΟΣΟ...** που εκτελεί τις εντολές που περικλείει όσο η συνθήκη είναι ΑΛΗΘΗΣ και αυτή γράφεται στην αρχή της εντολής.

Την εντολή **ΜΕΧΡΙΣ_ΟΤΟΥ...** που εκτελεί τις εντολές, που περικλείει, όσο η συνθήκη είναι ΨΕΥΔΗΣ και αυτή γράφεται στο τέλος της εντολής.

Την εντολή **ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...** που εκτελεί τις εντολές, που περικλείει, προκαθορισμένες φορές.

Που χρησιμοποιούνται οι δομές επανάληψης;

Χρησιμοποιούνται στη περίπτωση που έχουμε επαναλαμβανόμενες εντολές. Υπάρχουν τρεις μορφές επανάληψης γιατί η κάθε μια καλύπτει διαφορετικά είδη επανάληψης. Δηλαδή, όταν ξέρουμε πόσες φορές γίνεται η επανάληψη χρησιμοποιούμε την ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ..., εάν δεν ξέρουμε το πλήθος των επαναλήψεων και η επανάληψη σταματά με μια ερώτηση ή με το πάτημα ενός κουμπιού - επανάληψη υπό συνθήκη - τότε χρησιμοποιούμε την ΟΣΟ... ή την ΜΕΧΡΙΣ_ΟΤΟΥ...

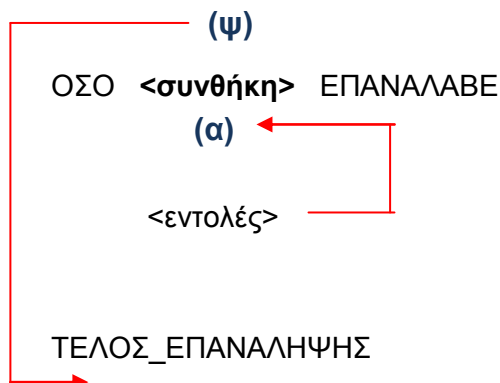
Η ΟΣΟ... χρησιμοποιείται στις περιπτώσεις που έχουμε τιμή φρουρό, δηλαδή να σταματά η επανάληψη όταν πληκτρολογήσουμε 0 ή αρνητικό αριθμό ή # κλπ. ενώ σε όλες τις άλλες περιπτώσεις χρησιμοποιείται η ΜΕΧΡΙΣ_ΟΤΟΥ...

Παρακάτω θα ελέγξουμε αναλυτικά την κάθε μια ξεχωριστά.

1.1. Η εντολή επανάληψης ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

Με την εντολή **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ** οι εντολές που περιλαμβάνει επαναλαμβάνονται όσο η συνθήκη είναι ΑΛΗΘΗΣ και μόνο όταν γίνει ΨΕΥΔΗΣ σταματά να εκτελεί αυτές και εκτελεί την αμέσως επόμενη εντολή μετά το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ.

Σύνταξη:



1.1.1. Χαρακτηριστικά της εντολής:

- Επειδή η συνθήκη βρίσκεται στην αρχή της εντολής υπάρχει περίπτωση να βγει ΨΕΥΔΗΣ από την αρχή οπότε και να μην εκτελεστούν καμία φορά οι εντολές της επανάληψης.
- Αν η συνθήκη δεν βγει ΨΕΥΔΗΣ ποτέ, τότε οι εντολές της επανάληψης εκτελούνται συνεχώς και το πρόγραμμα δεν τελειώνει ποτέ. Τότε λέμε ότι έχουμε **"ατέρμων βρόχο"**.

1.1.2. Πως συντάσσεται:

- Δίνουμε **Αρχική Τιμή στη μεταβλητή** της συνθήκης, γιατί η συνθήκη βρίσκεται στην αρχή της ΟΣΟ. και πρέπει η μεταβλητή να έχει αρχικοποιηθεί για να μπορεί να εκτελέσει ο υπολογιστής τη συνθήκη και να δώσει τη τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ
- Πριν τελειώσει η επανάληψη, δηλαδή πριν το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ πρέπει να **αλλάξουμε τη τιμή της μεταβλητής** της συνθήκης, για να μην έχουμε **ατέρμων βρόχο**.

<p>1^η περίπτωση</p> <p>Αρχική τιμή ΔΙΑΒΑΣΕ α ΟΣΟ α > 0 ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ α</p> <p>Αλλαγή τιμής ΔΙΑΒΑΣΕ α ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p> <p>Στο παράδειγμα αυτό η μεταβλητή της συνθήκης είναι η α. Η μεταβλητή αυτή παίρνει τιμές από το πληκτρολόγιο, γι' αυτό η αρχική τιμή της είναι η εντολή ΔΙΑΒΑΣΕ α την πρώτη φορά πριν την εντολή ΟΣΟ., για να μπορέσει να ελεγχθεί η συνθήκη $\alpha > 0$. Το ΔΙΑΒΑΣΕ α ξαναγράφεται πριν το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ για να αλλάζει τη τιμή της α, με τη νέα τιμή που πληκτρολογούμε. Η παραπάνω επανάληψη κάνει το εξής: 'Πληκτρολογούμε θετικούς αριθμούς'. Στο παράδειγμα αυτό η μεταβλητή της συνθήκης είναι η α.</p>	<p>2^η περίπτωση</p> <p>Αρχική τιμή α ← 100 ΟΣΟ α ≤ 1000 ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ α</p> <p>Αλλαγή τιμής α ← α + 2 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p> <p>Η μεταβλητή αυτή παίρνει τιμές με εντολή καταχώρησης. Αρχική τιμή της α είναι το 100, έτσι πριν το ΟΣΟ γράφουμε την εντολή $\alpha \leftarrow 100$, για να μπορέσει να ελεγχθεί η συνθήκη $\alpha < 1000$. Η αλλαγή της μεταβλητής α γίνεται πριν το τέλος της επανάληψης με την $\alpha \leftarrow \alpha + 2$, έτσι ώστε να αλλάξει τη τιμή της α και να μην έχουμε ατέρμων βρόχο. Η παραπάνω επανάληψη κάνει το εξής: Τυπώνει όλους τους άρτιους αριθμούς από το 100 μέχρι και το 1000'</p>
---	--

1.1.3. Που χρησιμοποιείται:

Σε όλες τις περιπτώσεις που έχουμε επανάληψη υπό συνθήκη, η οποία σταματά με μια τιμή φρουρό.

1.1.4. Παραδείγματα

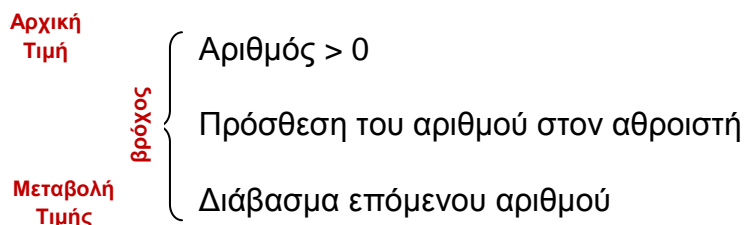
1. Πληκτρολογούμε θετικούς αριθμούς, η πληκτρολόγηση σταματά όταν δώσουμε αρνητικό αριθμό. Στο τέλος τυπώνεται το άθροισμα των θετικών αριθμών.

Ανάλυση:

1. Εφόσον πληκτρολογούμε συνέχεια θετικούς αριθμούς, έχουμε επανάληψη.
2. Η πληκτρολόγηση σταματά με τον αρνητικό αριθμό ή μηδέν, άρα η επανάληψη είναι με συνθήκη που ελέγχει αν ο αριθμός που πληκτρολογήσαμε είναι θετικός. (Τιμή φρουρός: αριθμός>0).
3. Επειδή χρειαζόμαστε το άθροισμα των θετικών αριθμών, που δεν είναι αποθηκευμένοι σε διαφορετικές μεταβλητές αλλά σε μια, θα χρησιμοποιήσουμε αθροιστή που σημαίνει μηδενισμός του αθροιστή στην αρχή του προγράμματος.

Γενική μορφή:

Μηδενισμός Αθροιστή
Διάβασμα του 1^{ου} αριθμού



Εκτύπωση αθροίσματος

ΠΡΟΓΡΑΜΜΑ Άθροισμα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: sum, x

ΑΡΧΗ

sum <- 0

ΔΙΑΒΑΣΕ x

ΟΣΟ x > 0 **ΕΠΑΝΑΛΑΒΕ**

sum <- sum + x

ΔΙΑΒΑΣΕ x

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ sum

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Άθροισμα

1.2. Η εντολή επανάληψης ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ...

Με την εντολή ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ οι εντολές που περιλαμβάνει επαναλαμβάνονται μέχρις ότου η συνθήκη γίνει αληθής. Δηλαδή, όσο η συνθήκη είναι ΨΕΥΔΗΣ έχουμε επανάληψη, ενώ όταν η συνθήκη γίνει ΑΛΗΘΗΣ η επανάληψη σταματά και εκτελείται η αμέσως επόμενη εντολή μετά το ΜΕΧΡΙΣ_ΟΤΟΥ.

1.2.1. Σύνταξη:

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

<εντολές> ← (ψ)

ΜΕΧΡΙΣ_ΟΤΟΥ <συνθήκη>

..... ← (α)

1.2.2. Χαρακτηριστικά της εντολής:

- Επειδή η συνθήκη στην εντολή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ βρίσκεται στο τέλος της εντολής οι εντολές, που περιλαμβάνονται σ' αυτήν, θα εκτελεστούν τουλάχιστον μια φορά.
- Οι εντολές επανάληψης ΟΣΟ. και ΜΕΧΡΙΣ_ΟΤΟΥ. είναι εντολές επανάληψης υπό συνθήκη, δηλαδή τις χρησιμοποιούμε όπου έχουμε επανάληψη εντολών και σταματά η επανάληψη όταν ισχύει ή δεν ισχύει μια συνθήκη. Επίσης μπορεί η μια να αντικατασταθεί από την άλλη σ' όλες τις περιπτώσεις επανάληψης υπό συνθήκη. Όπως θα δούμε και παρακάτω. Όμως μια από τις δύο είναι η βέλτιστη σε κάθε πρόβλημα.

1.2.3. Που χρησιμοποιείται:

- Στα προβλήματα που έχουν *Μενού επιλογής*.
- Στα προβλήματα που έχουμε *Έλεγχο δεδομένων από το πληκτρολόγιο*.
- Στα προβλήματα που θέλουμε να επαναλάβουμε μια ολόκληρη επεξεργασία (π.χ. Αναζήτηση δεδομένων) απαντώντας στην ερώτηση *'Θέλεις να συνεχίσεις (N/O)?'*.

Στις παραπάνω τρεις περιπτώσεις, που αναφέρθηκαν για επιλογή της εντολής ΜΕΧΡΙΣ_ΟΤΟΥ, μπορεί να χρησιμοποιηθεί και η ΟΣΟ αλλά δεν μας συμφέρει γιατί γράφονται περισσότερες εντολές.

1.2.4. Παραδείγματα

1. Έλεγχος δεδομένων

Να δώσετε από το πληκτρολόγιο το μέσο προφορικό βαθμό και το γραπτό βαθμό ενός μαθητή. Να υπολογισθεί και να τυπωθεί ο μέσος όρος του μαθητή. Να γίνεται έλεγχος για τους βαθμούς, ώστε να βρίσκονται στο διάστημα [0,20].

Λύση

Ανάλυση:

1. Διαβάζετε προφορικό βαθμό **με έλεγχο**.
2. Διαβάζετε τον γραπτό βαθμό **με έλεγχο**.
3. Υπολογίζετε τον Μέσο Όρο του μαθητή.

ΠΡΟΓΡΑΜΜΑ Έλεγχος

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Πρ, Γρ, ΜΟ

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε προφορικό βαθμό:'

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Πρ

ΜΕΧΡΙΣ_ΟΤΟΥ Πρ \leq 20 **ΚΑΙ** Πρ \geq 0

ΓΡΑΨΕ 'Δώσε γραπτό βαθμό''

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Γρ

ΜΕΧΡΙΣ_ΟΤΟΥ Γρ \leq 20 **ΚΑΙ** Γρ \geq 0

$ΜΟ \leftarrow (Πρ + Γρ) / 2$

ΓΡΑΨΕ 'Μέσος Όρος=',Μο

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Έλεγχος

Παρατήρηση:

Οι εντολές:

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Πρ

ΜΕΧΡΙΣ_ΟΤΟΥ Πρ ≤ 20 **ΚΑΙ** Πρ ≥ 0

Διαβάζετε από το πληκτρολόγιο έναν αριθμό, λόγω της ΔΙΑΒΑΣΕ Πρ, αν ο αριθμός βρίσκεται στο διάστημα $[0,20]$ τότε το πρόγραμμα εκτελεί την εντολή μετά το ΜΕΧΡΙΣ_ΟΤΟΥ, δηλαδή η συνθήκη βγαίνει ΑΛΗΘΗΣ.

Εάν όμως ο αριθμός δεν βρίσκεται στο διάστημα $[0,20]$, δηλ. η συνθήκη βγει ΨΕΥΔΗΣ, τότε επαναλαμβάνεται η εντολή και το πρόγραμμα περιμένει να ξαναπληκτρολογήσουμε το βαθμό. Αυτό σημαίνει ότι το πρόγραμμα δεν μας επιτρέπει να προχωρήσουμε αν δεν πληκτρολογήσουμε τα σωστά δεδομένα.

Οι εντολές:

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Πρ

ΑΝ Πρ < 0 **Ή** Πρ > 20 **ΤΟΤΕ**

ΓΡΑΨΕ 'Λάθος βαθμός. Ξαναπροσπάθησε!'

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ Πρ ≤ 20 **ΚΑΙ** Πρ ≥ 0

Οι παραπάνω εντολές κάνουν ακριβώς το ίδιο με αυτές στην προηγούμενη περίπτωση, αλλά τυπώνεται και μήνυμα λάθους στην περίπτωση λάθους πληκτρολόγησης.

2. Μενού επιλογής

Δίνετε από το πληκτρολόγιο 2 αριθμούς ακέραιους και πληκτρολογείτε στη συνέχεια την επιλογή της πράξης που θα εκτελεστεί για τους παραπάνω αριθμούς, σύμφωνα με το μενού:

1. Πρόσθεση
2. Αφαίρεση
3. Πολλαπλασιασμός
4. Διαίρεση
5. Έξοδος

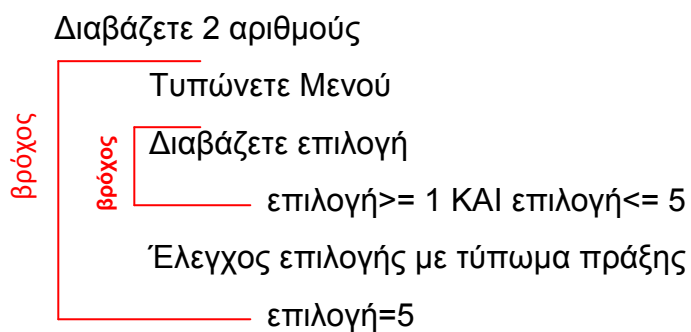
Όταν επιλέξετε '5' τότε το πρόγραμμα τελειώνει, διαφορετικά μπορείτε να επιλέξετε πάλι μια νέα πράξη. Να γίνεται έλεγχος της επιλογής για τα νούμερα 1,2,3,4 και 5.

Λύση

Ανάλυση:

1. Διαβάζετε τους 2 αριθμούς.
2. ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ.
3. Τυπώνετε το ΜΕΝΟΥ.
4. Διαβάζετε επιλογή με έλεγχο.
5. Αν επιλογή = 1 τότε προσθέτετε.
6. Αν επιλογή = 2 τότε αφαιρείτε.
7. Αν επιλογή = 3 τότε πολλαπλασιάζετε.
8. Αν επιλογή = 4 τότε διαιρείτε.
9. Αν επιλογή = 5 ΤΕΛΟΣ προγράμματος αλλιώς επιστρέφει στο Βήμα 2.

Γενική μορφή:



ΠΡΟΓΡΑΜΜΑ Μενού

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: α, β, Αρ

ΑΚΕΡΑΙΕΣ: Ε

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε δύο ακέραιους αριθμούς:'

ΔΙΑΒΑΣΕ α, β

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ '1. Πρόσθεση'

ΓΡΑΨΕ '2. Αφαίρεση'

ΓΡΑΨΕ '3. Πολλαπλασιασμός'

ΓΡΑΨΕ '4. Διαίρεση'

ΓΡΑΨΕ '5. Έξοδος'

ΓΡΑΨΕ 'Επιλογή:'

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Ε

ΜΕΧΡΙΣ_ΟΤΟΥ Ε >= 1 ΚΑΙ Ε <= 5

ΑΝ Ε = 1 **ΤΟΤΕ**

Αρ <- α + β

ΓΡΑΨΕ 'Πρόσθεση=', Αρ

ΑΛΛΙΩΣ_ΑΝ Ε = 2 **ΤΟΤΕ**

Αρ <- α - β

ΓΡΑΨΕ 'Αφαίρεση=', Αρ

ΑΛΛΙΩΣ_ΑΝ Ε = 3 **ΤΟΤΕ**

Αρ <- α * β

ΓΡΑΨΕ 'Πολλαπλασιασμός=', Αρ

ΑΛΛΙΩΣ_ΑΝ Ε = 4 **ΤΟΤΕ**

ΑΝ β <> 0 **ΤΟΤΕ**

Αρ <- α / β

ΓΡΑΨΕ 'Διαίρεση=', Αρ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Η διαίρεση δεν γίνεται'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΜΕΧΡΙΣ_ΟΤΟΥ Ε = 5

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μενού

3. Επανάληψη μετά από ερώτηση

Να διαβάζετε ονοματεπώνυμο και 3 βαθμούς ενός μαθητή. Να υπολογίζετε και να τυπώνετε το ΜΟ και το μήνυμα 'ΕΠΙΤΥΧΩΝ', αν $ΜΟ > 10$, διαφορετικά 'ΑΠΟΤΥΧΩΝ'. Στη συνέχεια να βγαίνει το μήνυμα: 'Θέλεις να συνεχιστεί για άλλο μαθητή (N/O)?' Αν πληκτρολογήσετε 'N' τότε να επαναλαμβάνονται τα παραπάνω και για άλλο μαθητή, διαφορετικά να τελειώνει το πρόγραμμα.

Λύση

Ανάλυση:

1. ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ.
2. Διαβάζετε όνομα μαθητή.
3. Διαβάζετε τους τρεις βαθμούς.
4. Υπολογίζετε τον Μέσο Όρο.
5. Τυπώνετε τον Μέσο Όρο και Έλεγχος Μέσου Όρου.
6. Τυπώνετε το μήνυμα: 'Θέλεις να συνεχίσεις για άλλο μαθητή (N/O)?'
7. Διαβάζετε την Απάντηση .
8. Αν η Απάντηση είναι 'N' τότε επαναλαμβάνεται το πρόγραμμα από το 1ο Βήμα.

ΠΡΟΓΡΑΜΜΑ Μαθητές

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Β1, Β2, Β3, ΜΟ

ΧΑΡΑΚΤΗΡΕΣ: Απ, Ον

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Ονοματεπώνυμο:'

ΔΙΑΒΑΣΕ Ον

ΓΡΑΨΕ 'Δώσε 3 βαθμούς:'

ΔΙΑΒΑΣΕ Β1, Β2, Β3

$ΜΟ \leftarrow (Β1 + Β2 + Β3) / 3$

ΑΝ ΜΟ ≥ 10 **ΤΟΤΕ**

ΓΡΑΨΕ ΜΟ, 'ΕΠΙΤΥΧΩΝ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ ΜΟ, 'ΑΠΟΤΥΧΩΝ'

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Θέλεις να συνεχίσεις για άλλο μαθητή (N/O)?'

ΔΙΑΒΑΣΕ Απ

ΜΕΧΡΙΣ_ΟΤΟΥ Απ \neq 'Ο'

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μαθητές

1.3. Η εντολή επανάληψης ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ ...

Η εντολή επανάληψης ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ ... χρησιμοποιείται όταν είναι γνωστό το πλήθος των επαναλήψεων, π.χ. να βρούμε το ΜΟ 35 μαθητών.

Η εντολή αυτή έχει σχεδιαστεί για να απλοποιήσει τον έλεγχο των επαναλήψεων, αν και μπορούν όλα τα προβλήματα να υλοποιηθούν με τις άλλες δύο εντολές επανάληψης υπό συνθήκη.

Σύνταξη:

ΓΙΑ <μεταβλητή> ΑΠΟ <τιμή1> ΜΕΧΡΙ <τιμή2> ΜΕ_ΒΗΜΑ χ

.
<εντολές>

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

1.3.1. Τρόπος εκτέλεσης

Η εντολή αυτή χειρίζεται μια <μεταβλητή> στην οποία καταχωρείται η <τιμή1> αρχικά. Η τιμή που παίρνει η <μεταβλητή> συγκρίνεται με τη <τιμή2> και εφόσον είναι μικρότερη απ' αυτήν, τότε εκτελούνται οι εντολές που βρίσκονται στο βρόχο. Στη συνέχεια η μεταβλητή αυξάνεται κατά την τιμή x που ορίζει το ΒΗΜΑ. Αν η νέα τιμή είναι μικρότερη της τελικής, τότε ο βρόχος εκτελείται ξανά. Η διαδικασία επαναλαμβάνεται μέχρι η <μεταβλητή> γίνει μεγαλύτερη από την <τιμή2> αλλιώς το πρόγραμμα συνεχίζει με την εντολή μετά το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ.

Ο αριθμός του βήματος μπορεί να είναι θετικός ή αρνητικός και αυτό καθορίζει τον έλεγχο για την επανάληψη. Έτσι έχουμε τις περιπτώσεις:

1. **Βήμα =1** : παραλείπεται το ΜΕ_ΒΗΜΑ 1 και η μεταβλητή αυξάνεται κατά +1.
2. **Βήμα > 0** : τότε πρέπει να ισχύει $\text{τιμή1} \leq \text{τιμή2}$ για να εκτελεστεί τουλάχιστον μια φορά ο βρόχος.
3. **Βήμα < 0** : τότε πρέπει να ισχύει $\text{τιμή} \geq \text{τιμή2}$ για να εκτελεστεί τουλάχιστον μια φορά ο βρόχος.
4. **Βήμα = 0** : τότε ο βρόχος είναι **ατέρμων**.

1.3.2. Παραδείγματα

1. ΓΙΑ Κ ΑΠΟ 3 ΜΕΧΡΙ 3
Ο βρόχος εκτελείται ακριβώς μια φορά.
2. ΓΙΑ i ΑΠΟ 10 ΜΕΧΡΙ 5
Ο βρόχος δεν εκτελείται καμία φορά, γιατί $i=10>5$ και Βήμα >0
3. ΓΙΑ i ΑΠΟ 3 ΜΕΧΡΙ 0 ΜΕ_ΒΗΜΑ -1
Ο βρόχος εκτελείται 4 φορές.
4. ΓΙΑ i ΑΠΟ 10 ΜΕΧΡΙ 5 ΜΕ_ΒΗΜΑ -0.5
Ο βρόχος εκτελείται 11 φορές.
5. ΓΙΑ κ ΑΠΟ 5 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ -2
Ο βρόχος δεν εκτελείται καμία φορά, γιατί $κ=5<10$ και Βήμα <0 .
6. ΓΙΑ κ ΑΠΟ 5 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 0
Ο βρόχος εκτελείται άπειρες φορές

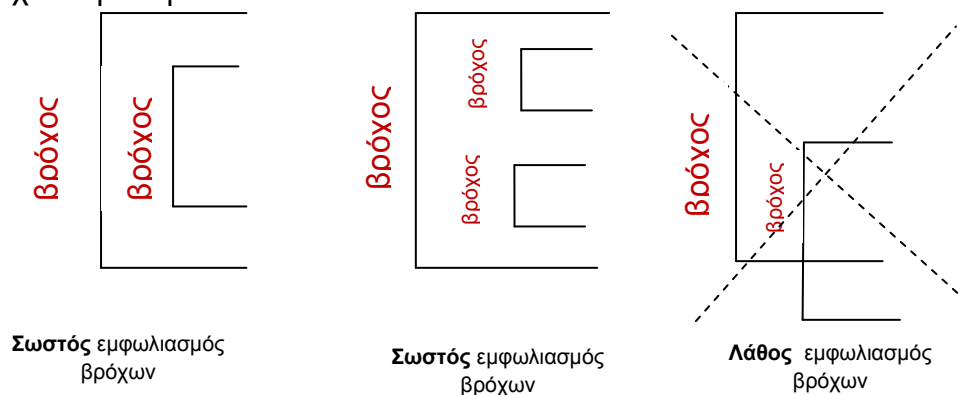
1.3.3. Παρατηρήσεις

1. Μέσα στην εντολή ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ... δεν επιτρέπεται να αλλάζετε την τιμή της <μεταβλητής> του ΓΙΑ, γιατί τότε δεν λειτουργεί σωστά η εντολή επανάληψης.
2. Το βήμα δεν είναι απαραίτητο να είναι ακέραιος αριθμός. Μπορεί να πάρει οποιαδήποτε πραγματική τιμή.
3. Μια εντολή επανάληψης ΓΙΑ ... μπορεί να μετατραπεί σε ΟΣΟ ... και ΜΕΧΡΙΣ_ΟΤΟΥ ... ενώ το αντίστροφο δεν ισχύει πάντα.

1.3.4. Εμφωλευμένοι Βρόχοι

Στην περίπτωση αυτή ο ένας βρόχος βρίσκεται μέσα στον άλλο. Για τη σωστή λειτουργία των εμφωλευμένων βρόχων πρέπει να ισχύουν συγκεκριμένοι κανόνες:

- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό βρόχο. Δηλαδή:



- Η είσοδος σε βρόχο υποχρεωτικά γίνεται από την αρχή του.
- Δεν πρέπει να χρησιμοποιηθεί η ίδια <μεταβλητή> ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

1.4. Αλγόριθμος ελάχιστου ή μεγίστου από μια ομάδα αριθμών

Όταν έχουμε μια ομάδα αριθμών που θέλουμε να βρούμε τον ελάχιστο ή το μέγιστο αριθμό αυτών, δεν χρησιμοποιούμε τόσες μεταβλητές όσοι είναι οι αριθμοί, αφού μερικές φορές δεν γνωρίζουμε και το πλήθος των αριθμών που θα δώσουμε, αλλά χρησιμοποιούμε μια επαναληπτική εντολή που περιέχει την ΔΙΑΒΑΣΕ <μεταβλητή>.

Για το λόγο αυτό δεν μπορούμε με πολλά και εμφωλευμένα AN...ΑΛΛΙΩΣ_ΑΝ... να βρούμε τον ελάχιστο ή τον μικρότερο και έτσι χρησιμοποιούμε τον παρακάτω αλγόριθμο:

1. Σε μια μεταβλητή Max ή Min, που εκφράζουν αντίστοιχα την μεγαλύτερη ή μικρότερη τιμή που θα έχουν στο τέλος, δίνουμε **Αρχική Τιμή**.
2. Κάθε νέα τιμή που δίνουμε στη <μεταβλητή> την ελέγχουμε με το Max ή Min αντίστοιχα, <μεταβλητή> < Min ή <μεταβλητή> > Max, και αν ισχύουν οι παραπάνω συνθήκες καταχωρούμε στο Min ή στο Max τη μεταβλητή με τη νέα τιμή .

Όπου **Αρχική Τιμή** υπάρχουν δύο περιπτώσεις.

1η περίπτωση

Αυθαίρετες τιμές

Στο Max δίνω την μικρότερη δυνατή τιμή. Στο Min δίνω την μεγαλύτερη δυνατή Τιμή.

Αυτό τον τρόπο τον χρησιμοποιούμε στις περιπτώσεις που το πεδίο τιμών της μεταβλητής είναι συγκεκριμένο.

Π.χ δίνω τους βαθμούς μαθητών [0,20]. Τότε σαν αρχικές τιμές

Μπορώ να δώσω

Max ← 0

και Min ← 20

...

2η περίπτωση

Συγκεκριμένες τιμές

Στο Max ή στο Min δίνω την πρώτη τιμή από την ομάδα που θα δώσω.

Αυτό τον τρόπο τον χρησιμοποιούμε στις περιπτώσεις που το πεδίο τιμών της μεταβλητής δεν είναι συγκεκριμένο.

Π.χ. Πληκτρολογούμε αριθμούς να βρούμε τον μεγαλύτερο ή τον μικρότερο απ' αυτούς.

ΔΙΑΒΑΣΕ B

Max ← B

Min ← B

1.4.1. Παραδείγματα

1. Να γίνει πρόγραμμα που να δίνονται 1000 αριθμοί από το πληκτρολόγιο να βρεθεί και να τυπωθεί ο μεγαλύτερος απ' αυτούς.

Λύση

Ανάλυση:

1. Χρησιμοποιούμε την επαναληπτική εντολή ΓΙΑ. για να πληκτρολογήσουμε τους 1000 αριθμούς.
2. Για να βρούμε τον μέγιστο χρησιμοποιούμε τον αλγόριθμο του ΜΕΓΙΣΤΟΥ με αρχική τιμή τον πρώτο αριθμό που πληκτρολογούμε.

Γενική μορφή:

Διαβάζω αριθμό
Καταχωρώ αριθμό
στην Μαχ

βρόχος

$i = 2$
Διαβάζω αριθμό
Ελέγχω αριθμό με την Μαχ
Μέχρι $i = 1000$
Τυπώνω τον Μαχ

ΠΡΟΓΡΑΜΜΑ Μέγιστος ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, I, Μαχ

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε αριθμό'

ΔΙΑΒΑΣΕ A

Αρχική τιμή Μαχ ← A

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 1000

ΓΡΑΨΕ 'Δώσε αριθμό'

ΔΙΑΒΑΣΕ A

έλεγχος

ΑΝ A > Μαχ **ΤΟΤΕ**
 Μαχ ← A
ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Μεγαλύτερος ο:', Μαχ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μέγιστος

Παρατήρηση

Στο παραπάνω παράδειγμα παρατηρούμε ότι όταν στο Max ή αντίστοιχα Min δίνουμε τον πρώτο αριθμό από την λίστα που θα πληκτρολογήσουμε τότε το πρώτο ΔΙΑΒΑΣΕ δίνεται έξω από την επανάληψη, για να δώσω την τιμή μου σαν αρχική τιμή στο Max ή στο Min.

2. Να βρεθεί ο μεγαλύτερος και ο μικρότερος από 1000 αριθμός που πληκτρολογώ.

Λύση

Ανάλυση:

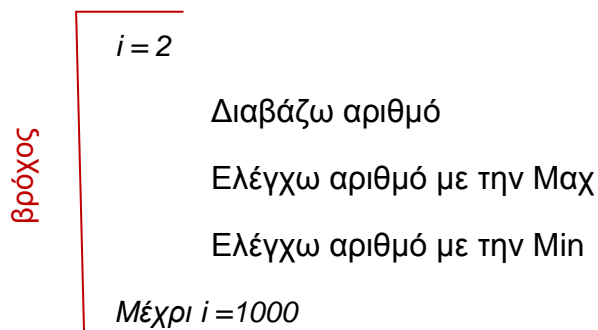
1. Χρησιμοποιούμε την επαναληπτική εντολή ΓΙΑ... για να πληκτρολογήσουμε τους 1000 αριθμούς.
2. Εφαρμόζουμε μέσα στο ΓΙΑ... τον έλεγχο για τον μεγαλύτερο και τον έλεγχο για τον μικρότερο σύμφωνα με τον αλγόριθμο του μεγαλύτερου και του μικρότερου.

Γενική μορφή:

Διαβάζω αριθμό

Καταχωρώ τον αριθμό στην Max

Καταχωρώ τον αριθμό στην Min



Τυπώνω την Max και την Min

ΠΡΟΓΡΑΜΜΑ Max_Min ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, i, Max, Min

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε αριθμό'

ΔΙΑΒΑΣΕ A

Αρχικές
Τιμές

Max ← A

Min ← A

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 1000

ΓΡΑΨΕ 'Δώσε αριθμό'

ΔΙΑΒΑΣΕ A

έλεγχος
Max

ΑΝ A > Max ΤΟΤΕ

Max ← A

ΤΕΛΟΣ-ΑΝ

έλεγχος
Min

ΑΝ A < Min ΤΟΤΕ

Min ← A

ΤΕΛΟΣ_ΑΝ και M

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Μεγαλύτερος:', Max

ΓΡΑΨΕ 'Μικρότερος:', Min

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Max_Min

Παρατήρηση

Όταν έχουμε ταυτόχρονα να ελέγχουμε Max και Min σ' ένα πρόγραμμα δεν πρέπει να χρησιμοποιούμε την ΑΝ ... ΑΛΛΙΩΣ ... γιατί δεν θα έχουμε σωστά αποτελέσματα, επειδή αν ένας αριθμός δεν είναι μεγαλύτερος από την Max, αυτό δεν σημαίνει ότι είναι μικρότερος από την Min.

Δηλαδή, θα γράφουμε δύο διαφορετικά ΑΝ..., ένα για το Max και ένα για το Min.

3. Διαβάζουμε αριθμούς από το πληκτρολόγιο, το τέλος της πληκτρολόγησης δηλώνεται αν δώσουμε τον αριθμό 0. Να βρίσκεται και να τυπώνεται ο μεγαλύτερος και ο μικρότερος απ' αυτούς.

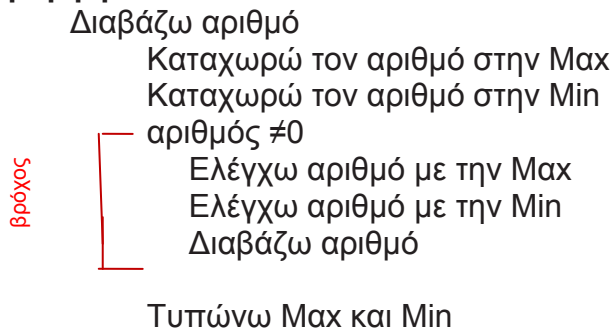
Λύση

Ανάλυση:

1. Χρησιμοποιούμε την επαναληπτική εντολή ΟΣΟ... για να πληκτρολογήσουμε τους αριθμούς, γιατί εδώ η πληκτρολόγηση σταματά όταν δώσουμε την τιμή φρουρό '0'. Δηλαδή, έχουμε επανάληψη υπό συνθήκη.

- Μέσα στην επανάληψη κάνουμε έλεγχο για τον μεγαλύτερο και έλεγχο για τον μικρότερο.

Γενική μορφή:



ΠΡΟΓΡΑΜΜΑ Max_Min

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, Max, Min

ΑΡΧΗ

```

ΓΡΑΨΕ 'Δώσε αριθμό'
ΔΙΑΒΑΣΕ A
Max ← A
Min ← A
ΟΣΟ A <> 0 ΕΠΑΝΑΛΑΒΕ
  ΑΝ A > Max ΤΟΤΕ
    Max ← A
  ΤΕΛΟΣ_ΑΝ
  ΑΝ A < Min ΤΟΤΕ
    Min ← A
  ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ 'Δώσε αριθμό'
ΔΙΑΒΑΣΕ A
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Μεγαλύτερος ',Max
ΓΡΑΨΕ 'Μικρότερος ',Min
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Max_Min
  
```

- Να διαβάζονται τα ονόματα και οι ΜΟ, 70 μαθητών της Β' Λυκείου ενός σχολείου. υπολογισθεί να τυπωθεί το όνομα και ο μέσος όρος του μεγαλύτερου και του χειρότερου μαθητή.

Λύση

Ανάλυση:

- Χρησιμοποιούμε την επαναληπτική εντολή ΓΙΑ... για να διαβάσουμε τα 70 ονόματα και μέσους όρους των μαθητών.
- Οα χρησιμοποιηθεί ο αλγόριθμος του μεγαλύτερου και του μικρότερου με την διαφορά ότι θα χρησιμοποιηθούν και βοηθητικές μεταβλητές που θα κρατούν τα ονόματα του καλύτερου και χειρότερου μαθητή.

2. Σαν αρχική τιμή του μεγαλύτερου και του μικρότερου μπορούμε να δώσουμε το 20 και το 0 αντίστοιχα, εφόσον το πεδίο τιμών είναι [0,20].

Γενική μορφή: Max \leftarrow 0
Min \leftarrow 20

$i = 1$
 Διαβάζω Όνομα και Μέσο Όρο μαθητή
 Ελέγχω τον Μέσο Όρο με την Max
 Ελέγχω τον Μέσο Όρο με την Min
Μέχρι $i = 70$

Τυπώνω την Max και το Όνομα του μεγαλύτερου βαθμού

Τυπώνω την Min και το Όνομα του μικρότερου βαθμού

ΠΡΟΓΡΑΜΜΑ Καλύτερος_Χειρότερος_μαθητης

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i

ΠΡΑΓΜΑΤΙΚΕΣ: MO, Max, Min

ΧΑΡΑΚΤΗΡΕΣ: On, Μαχ_ον, Μιν_ον

ΑΡΧΗ

Max \leftarrow 0

Min \leftarrow 20

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 70

ΓΡΑΨΕ 'Δώσε Όνομα και Μ.Ο. μαθητή'

ΔΙΑΒΑΣΕ On, MO

ΑΝ MO > Max **ΤΟΤΕ**

 Max \leftarrow MO

 Μαχ_ον \leftarrow On

ΤΕΛΟΣ_ΑΝ

ΑΝ MO < Min **ΤΟΤΕ**

 Min \leftarrow MO

 Μιν_ον \leftarrow On

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Καλύτερος ο ',Μαχ_ον, 'με Μ.Ο.=', Max

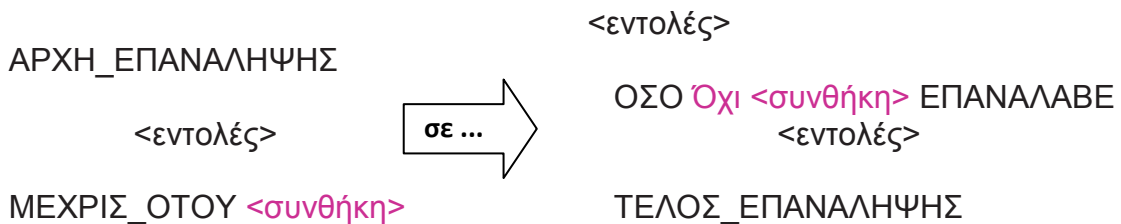
ΓΡΑΨΕ 'Χειρότερος ο ',Μιν_ον, 'με Μ.Ο.=', Min

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Καλύτερος_Χειρότερος_μαθητης

1.5. Μετατροπές από μια δομή επανάληψης σε άλλη

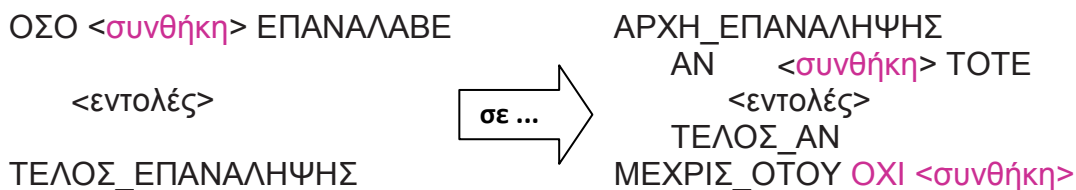
1.5.1. Μετατροπή από ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ ... ΣΕ ΟΣΟ...ΚΑΙ ΑΝΤΙΣΤΡΟΦΩΣ

1.



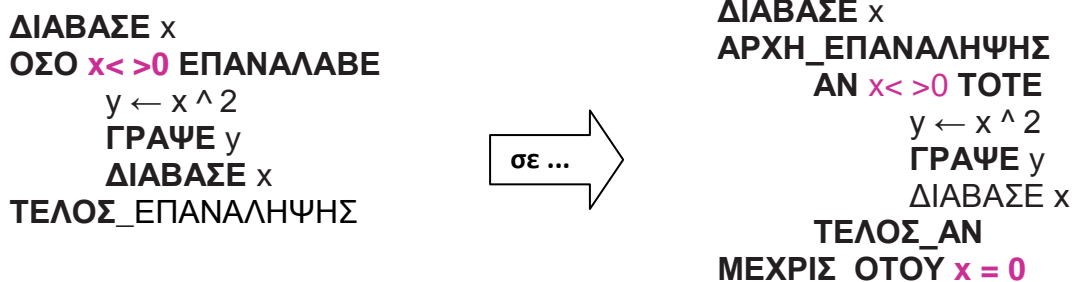
- Επειδή η εντολή επανάληψης ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ εκτελείται όσο η <συνθήκη> είναι Ψευδής, ενώ στην εντολή επανάληψης ΟΣΟ...ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ εκτελείται όσο η <συνθήκη> της είναι Αληθής, ι' αυτό κατά την μετατροπή από την μια εντολή επανάληψης στην άλλη αρκεί να γράφουμε την άρνηση της <συνθήκη> από την μια στην άλλη ή να γράφουμε τον τελεστή ΟΧΙ την συνθήκη.
- Επίσης επειδή η ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ εκτελεί τουλάχιστον μια φορά όλες τις εντολές της γι' αυτό όταν την μετατρέπουμε στην ΟΣΟ...ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ πρέπει όλες τις εντολές της να τις γράφουμε μια φορά πριν την ΟΣΟ...ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ και άλλη μια φορά μέσα σ' αυτήν.

2.



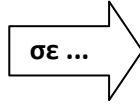
1.5.1.1. Παραδείγματα

1. Μετατροπή από ΟΣΟ.. σε ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ



2. Μετατροπή από ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ σε ΟΣΟ...

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ x
  y <-x ^ 2
  ΓΡΑΨΕ y
ΜΕΧΡΙΣ_ΟΤΟΥ x=0
```



```
ΔΙΑΒΑΣΕ x
y <-x ^ 2
ΓΡΑΨΕ y
ΟΣΟ x<>0 ΕΠΑΝΑΛΑΒΕ
  ΔΙΑΒΑΣΕ x
  y <-x ^ 2
  ΓΡΑΨΕ y
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

1.5.2. Μετατροπή από ΓΙΑ... σε ΟΣΟ... ή σε ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ και αντιστρόφως

1. Περίπτωση $\text{τιμή1} \leq \text{τιμή2}$ και $\beta > 0$

```
ΓΙΑ <μεταβλητή> ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή 2 ΜΕ_ΒΗΜΑ  $\beta$ 
  <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
<μεταβλητή> ← τιμή1
ΟΣΟ <μεταβλητή> <=τιμή2 ΕΠΑΝΑΛΑΒΕ
  <εντολές>
  <μεταβλητή> ← <μεταβλητή>+ $\beta$ 
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

2. Περίπτωση $\text{τιμή1} \geq \text{τιμή2}$ και $\beta < 0$

```
ΓΙΑ <μεταβλητή> ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή2 ΜΕ_ΒΗΜΑ  $\beta$ 
  <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
<μεταβλητή> ← τιμή1
ΟΣΟ <μεταβλητή> >=τιμή2 ΕΠΑΝΑΛΑΒΕ
  <εντολές>
  <μεταβλητή> ← <μεταβλητή>+ $\beta$ 
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- Πριν την εντολή ΟΣΟ...ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ δίνουμε στην <μεταβλητή> της ΓΙΑ... την αρχική τιμή δηλ. την **τιμή1**.

- Στη συνθήκη της ΟΣΟ...ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ συγκρίνουμε την <μεταβλητή> με την **τιμή2**. Η σύγκριση εξαρτάται από το βήμα αν είναι θετικό ή αρνητικό όπως βλέπουμε παραπάνω.
- Πριν το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ της ΟΣΟ... αυξάνουμε την τιμή της μεταβλητής όσο είναι η τιμή του βήματος. Στη περίπτωση που το βήμα δεν υπάρχει, τότε η <μεταβλητή> αυξάνεται κατά 1.
- Ανάλογα, γίνεται η μετατροπή της ΟΣΟ... σε ΓΙΑ...

1.5.2.1. Παραδείγματα

1. Μετατροπή από ΓΙΑ... σε ΟΣΟ...

α)

ΓΙΑ **i** ΑΠΟ **1** ΜΕΧΡΙ **10**
 ΓΡΑΨΕ i^3
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ



i ← 1
 ΟΣΟ **i** ≤ 100 ΕΠΑΝΑΛΑΒΕ
 ΓΡΑΨΕ i^3
i ← i+1
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

β)

ΓΙΑ **i** ΑΠΟ **10** ΜΕΧΡΙ **100** ΜΕ_ΒΗΜΑ **2**
 ΓΡΑΨΕ i^2
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ



i ← 10
 ΟΣΟ **i** ≤ 100 ΕΠΑΝΑΛΑΒΕ
 ΓΡΑΨΕ i^2
i ← i+2
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

γ)

ΓΙΑ **i** ΑΠΟ **1000** ΜΕΧΡΙ **100** ΜΕ_ΒΗΜΑ **-1**
 ΓΡΑΨΕ **i**
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ



i ← 1000
 ΟΣΟ **i** ≥ 100 ΕΠΑΝΑΛΑΒΕ
 ΓΡΑΨΕ **i**
i ← i-1
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

2. Μετατροπή από ΟΣΟ... σε ΓΙΑ...

α)

i ← 10
 ΟΣΟ **i** ≤ 100
 ΕΠΑΝΑΛΑΒΕ
 ΓΡΑΨΕ **i**
i ← i+1
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ



ΓΙΑ **i** ΑΠΟ **10** ΜΕΧΡΙ **100** _ΒΗΜΑ **100**
 ΓΡΑΨΕ **i**
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

β)

$i \leftarrow 10$ ΟΣΟ $i \leq 100$ ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ i $i \leftarrow i + 2$ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	⇒	ΓΙΑ i ΑΠΟ 10 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2 ΓΡΑΨΕ i ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
--	---	---

γ)

$i \leftarrow 100$ ΟΣΟ $i \geq 1$ ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ i $i \leftarrow i - 2$ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	⇒	ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 1 ΜΕ_ΒΗΜΑ -2 ΓΡΑΨΕ i ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
---	---	---

2. ΠΙΝΑΚΕΣ

Στο πρόγραμμα ο πίνακας εκφράζει μια μεταβλητή που έχει περισσότερες από μια θέσεις, δηλαδή μια μεταβλητή πολλαπλών θέσεων. Για το λόγο αυτό όταν ορίζουμε ένα πίνακα, χρησιμοποιούμε το όνομα της μεταβλητής και το δείκτη, που είναι ένας αριθμός και εκφράζει τη θέση του κάθε δεδομένου.

Ένας πίνακας μπορεί να είναι μονοδιάστατος, αλλά στη γενικότερη περίπτωση μπορεί να είναι δισδιάστατος, τρισδιάστατος και γενικά n -διάστατος πίνακας. Όσο αφορά τους δισδιάστατους πίνακες αν το μέγεθος των δύο διαστάσεων είναι ίσο, δηλαδή είναι $n \times n$, τότε ο πίνακας λέγεται **τετραγωνικός**.

Άρα:

Πίνακας είναι ένα σύνολο δεδομένων ίδιου τύπου, τα οποία έχουν το ίδιο όνομα. Κάθε ένα από τα δεδομένα του πίνακα λέγεται στοιχείο του πίνακα. Όταν αναφερόμαστε σε ένα δεδομένο από τον πίνακα τότε γράφουμε το όνομα του πίνακα ακολουθούμενο από έναν δείκτη.

Οι στατικές δομές υλοποιούνται με πίνακες.

Πίνακας είναι μια δομή που περιέχει στοιχεία του ίδιου τύπου.

2.1. Πλεονεκτήματα χρήσης πινάκων:

Η χρήση πίνακα είναι ένας βολικός τρόπος για τη διαχείριση πολλών δεδομένων ίδιου τύπου.

2.2. Μειονεκτήματα χρήσης πινάκων:

- Οι πίνακες απαιτούν μνήμη, η οποία δεσμεύεται από την αρχή του προγράμματος. Αυτό έχει
- σαν αποτέλεσμα, σε μεγάλα και σύνθετα προγράμματα την αδυναμία εκτέλεσης τους.
- Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος.

2.3. Που χρησιμοποιούνται οι πίνακες;

Χρησιμοποιούνται στη περίπτωση που έχουμε να διαχειριστούμε πολλά δεδομένα και πρέπει να είναι αποθηκευμένα για μεταγενέστερη επεξεργασία τους.

2.4. Δήλωση πίνακα στη ΓΛΩΣΣΑ.

<τύπος>: Όνομα_πίνακα [πλήθος θέσεων].

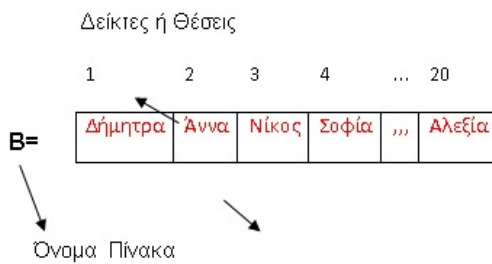
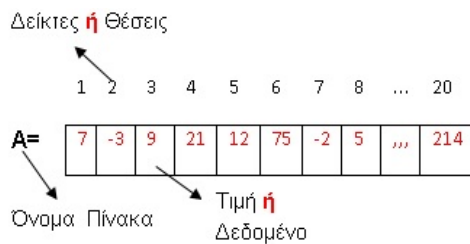
Μονοδιάστατος

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[20]

ΧΑΡΑΚΤΗΡΕΣ: B[20]

Ο πίνακας A[20] δηλώνει μια μεταβλητή A με 20 θέσεις. Δηλ. αν τον αναπαριστούσα θα ήταν:



Όταν γράφουμε στο πρόγραμμα B[2] αναφερόμαστε στο δεδομένο του πίνακα B στη θέση 2, δηλαδή το δεδομένο "ANNA".

- Η αρίθμηση των θέσεων αρχίζει από το ένα και είναι ακέραιος αριθμός με βήμα 1.

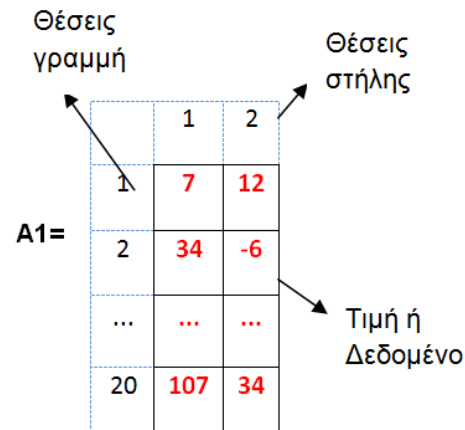
Δισδιάστατος

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A1[20,2]

ΧΑΡΑΚΤΗΡΕΣ: B1[5,3]

Ο πίνακας A1[20,2] δηλώνει μια μεταβλητή A1 με $20 \times 2 = 40$ θέσεις, έτσι ώστε να έχει 20 γραμμές και 2 στήλες. Δηλ. αν τον αναπαριστούσα θα ήταν :



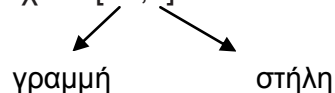
- Για να ορίσουμε μια θέση σ' ένα δισδιάστατο πίνακα πρέπει να ορίσουμε σε ποια γραμμή και μετά σε ποια στήλη βρίσκεται το δεδομένο.

Όταν γράφουμε το A1[15,2] στο πρόγραμμα, αναφερόμαστε στο δεδομένο που βρίσκεται στην 15η γραμμή και τη 2η στήλη.

- Η αρίθμηση των θέσεων γραμμών ή στηλών αρχίζει από το 1 και είναι ακέραιος αριθμός με βήμα 1.

- Όταν ορίζουμε τη θέση ενός δισδιάστατου πίνακα, πάντα γράφουμε πρώτα την γραμμή και μετά τη στήλη του πίνακα.

Π.χ. A[10,2]



2.5. Είσοδος δεδομένων σε πίνακα

2.5.1. Σε μονοδιάστατο πίνακα

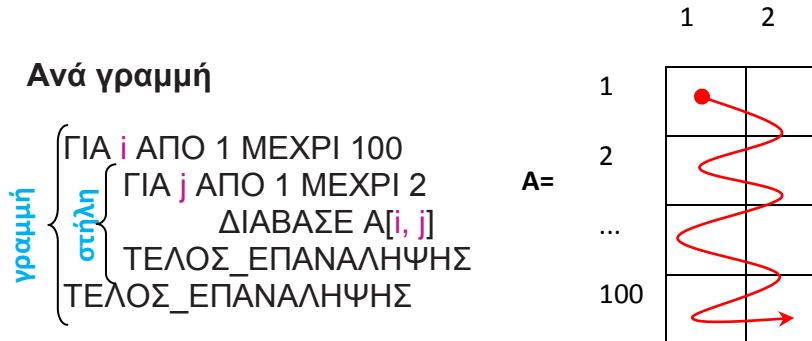
Έστω ότι θέλω να διαβάσω 100 ακέραιους αριθμούς από το πληκτρολόγιο:

1ος τρόπος (χωρίς χρήση πίνακα)	2ος τρόπος (με χρήση πίνακα)
<p>ΔΙΑΒΑΣΕ A1 ΔΙΑΒΑΣΕ A2 ΔΙΑΒΑΣΕ A3 ΔΙΑΒΑΣΕ A4 ... ΔΙΑΒΑΣΕ A100</p> <p>Αυτός ο τρόπος δεν χρησιμοποιείται γιατί γράφουμε πολλές φορές την ίδια εντολή.</p> <p>ή</p> <p>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20 ΔΙΑΒΑΣΕ A ΤΕΛΟΣ_ΕΠΝΑΛΗΨΗΣ</p> <p>Αυτός ο τρόπος δεν εξυπηρετεί στην περίπτωση που θέλω να έχω όλους τους αριθμούς για να τους επεξεργαστώ, εφόσον και οι 100 αριθμοί που πληκτρολογώ καταχωρούνται στην ίδια μεταβλητή με αποτέλεσμα αυτή να κρατά κάθε φορά τον τελευταίο που πληκτρολογήσαμε.</p>	<p>ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: A[100]</p> <p>ΑΡΧΗ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100 ΔΙΑΒΑΣΕ A[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ... ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</p> <p>Κατά την εκτέλεση των παραπάνω εντολών πληκτρολογούμε 100 αριθμούς έτσι ώστε ο κάθε ένας να μπαίνει σε διαφορετική θέση στο πίνακα. Αυτό έχει σαν αποτέλεσμα να μπορούμε στο πρόγραμμά μας παρακάτω, να τους επεξεργαστούμε όλους ή κάθε έναν απ' αυτούς, όποτε τους χρειαζόμαστε.</p>

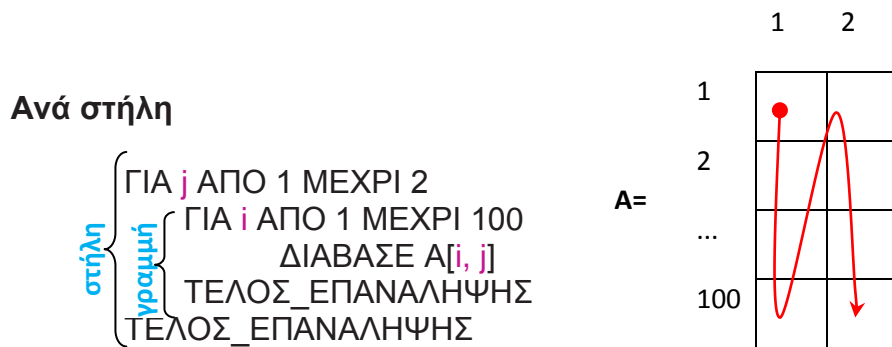
2.5.2. Σε δισδιάστατο πίνακα

Στη περίπτωση αυτή μας ενδιαφέρει αν τα δεδομένα καταχωρηθούν στο πίνακα ανά γραμμή ή ανά στήλη. Οπότε έχουμε:

π.χ. Διάβασμα 20 ακέραιων και καταχώρηση στον πίνακα A[10,2]



Με τις εντολές αυτές γεμίζουμε τον πίνακα ανά γραμμή, που σημαίνει ότι όταν γεμίζει μια γραμμή τότε συνεχίζει από την αρχή της επόμενης.



Με τις εντολές αυτές γεμίζουμε τον πίνακα ανά στήλη, που σημαίνει ότι όταν γεμίζει μια στήλη τότε συνεχίζει από την αρχή της επόμενης.

Παρατήρηση

Η ανάγνωση, η επεξεργασία και η εκτύπωση των στοιχείων ενός πίνακα γίνεται πάντα από βρόχους, που επαναλαμβάνονται προκαθορισμένο αριθμό φορών όσα είναι τα στοιχεία του πίνακα. Επομένως συμφέρει να υλοποιηθούν με την εντολή επανάληψης ΓΙΑ.

2.5.3. Παραδείγματα

1. Να γραφτεί πρόγραμμα που να υπολογίζει τη μέση θερμοκρασία 30 ημερών και στη συνέχεια να υπολογίζει και να τυπώνει πόσες θερμοκρασίες ήταν μικρότερες από τη μέση θερμοκρασία.

Λύση

Ανάλυση

Αν χρησιμοποιήσουμε ένα ΓΙΑ... για να διαβάσουμε τις 30 θερμοκρασίες, τότε θα βρούμε τον μέσο όρο τους αλλά για να ελέγξουμε πόσες είναι μικρότερες από τη μέση θερμοκρασία θα πρέπει να τις ξαναδιαβάσουμε. Αυτό δεν εξυπηρετεί γιατί θα πρέπει να πληκτρολογήσουμε 2 φορές τις 30 θερμοκρασίες.

Έτσι θα χρησιμοποιηθεί μονοδιάστατος πίνακας 30 θέσεων στον οποίο θα καταχωρήσουμε τις 30 θερμοκρασίες και ταυτόχρονα θα βρούμε τον μέσο όρο τους. Στη συνέχεια θα ελέγξουμε κάθε μια από τις 30 θερμοκρασίες αν είναι μικρότερη της μέσης τιμής και θα την μετράμε.

ΠΡΟΓΡΑΜΜΑ θερμοκρασία

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Θ [30], i, M, sum

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΑΡΧΗ

sum ← 0

ΓΡΑΨΕ "Δώσε 30 θερμοκρασίες"

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΔΙΑΒΑΣΕ Θ [i]

 sum ← sum + Θ [i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ ← sum/30

ΓΡΑΨΕ "Μέση θερμοκρασία:", ΜΟ

M ← 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΑΝ Θ[i] < ΜΟ **ΤΟΤΕ**

 M ← M + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ "Ημέρες με θερμοκρασία<Μέση Θερ:", M

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ θερμοκρασία

2. Να γίνει πρόγραμμα που να γεμίζει ένα δισδιάστατο πίνακα 4x3, αν γραμμή με ακέραιους αριθμούς που δίνονται από το πληκτρολόγιο. Στη συνέχεια να υπολογίζει και να τυπώνει:
1. Άθροισμα ανά γραμμή
 2. Άθροισμα ανά στήλη
 3. Το συνολικό άθροισμα όλων των αριθμών του πίνακα.

Λύση

Γενική μορφή:

Δίνω τα δεδομένα

$$\left[\begin{array}{l} i = 1 \\ \left[\begin{array}{l} j = 1 \\ \text{Πληκτρολογώ τα δεδομένα και τα καταχωρώ στον πίνακα} \\ \text{Μέχρι } j = 3 \end{array} \right. \\ \left. \text{Μέχρι } i = 4 \end{array} \right.$$

Άθροισμα ανά γραμμή

$$\left[\begin{array}{l} i = 1 \\ \left[\begin{array}{l} \text{Μηδενισμό αθροιστή γραμμής} \\ J = 1 \\ \text{Πρόσθεση στοιχείου στον αθροιστή γραμμής} \\ \text{Μέχρι } j = 3 \\ \text{Εκτύπωση του αθροιστή γραμμής} \end{array} \right. \\ \left. \text{Μέχρι } i = 4 \end{array} \right.$$

Άθροισμα ανά στήλη

$$\left[\begin{array}{l} J = 1 \\ \left[\begin{array}{l} \text{Μηδενισμό αθροιστή στήλης} \\ i = 1 \\ \text{Πρόσθεση στοιχείου στον αθροιστή στήλης} \\ \text{Μέχρι } i = 4 \\ \text{Εκτύπωση του αθροιστή στήλης} \end{array} \right. \\ \left. \text{Μέχρι } j = 3 \end{array} \right.$$

Συνολικό άθροισμα

$$\left[\begin{array}{l} \text{Μηδενισμός αθροιστή} \\ i = 1 \\ \left[\begin{array}{l} J = 1 \\ \text{Πρόσθεση στοιχείου στον Αθροιστή} \\ \text{Μέχρι } j = 3 \end{array} \right. \\ \left. \text{Μέχρι } i = 4 \\ \text{Εκτύπωση Αθροιστή} \end{array} \right.$$

ΠΡΟΓΡΑΜΜΑ Αθροίσματα
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[4,3], i, j, sum_στ, sum_γρ, sum

ΑΡΧΗ

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 3
    ΓΡΑΨΕ 'Δώσε αριθμό ακέραιο:'
    ΔΙΑΒΑΣΕ A [i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4
  sum_γρ ← 0
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 3
    sum_γρ ← sum_γρ + A [i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Άθροισμα της', i, 'γραμμής', 'είναι', sum_γρ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 3
  sum_στ ← 0
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4
    sum_στ ← sum_στ + A [i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Άθροισμα της', j, 'στήλης', 'είναι:', sum_στ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

sum ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 4
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 3
    sum ← sum + A [i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Συνολικό άθροισμα:', sum
```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αθροίσματα

Παρατήρηση

- Αν τα ζητούμενα σ' ένα δισδιάστατο πίνακα αφορούν γραμμή ή στήλη τότε οι αρχικές τιμές για άθροισμα, ελάχιστο, μέγιστο, κ.λ.π. δίνονται ανάμεσα στα δύο ΓΙΑ...
- Ενώ αν τα ζητούμενα σ' ένα δισδιάστατο πίνακα αφορούν όλο το πίνακα, τότε οι αρχικές τιμές δίνονται πριν από τα δύο ΓΙΑ...

3. Να γίνει πρόγραμμα στο οποίο να υπολογίζεται και να τυπώνεται ο μεγαλύτερος αριθμός, από 100 ακέραιους αριθμούς που θα διαβαστούν απ' το πληκτρολόγιο.

Λύση

Ανάλυση

1. θα χρησιμοποιηθεί πίνακας 100 θέσεων για να καταχωρήσω τους αριθμούς
2. θα εφαρμοστεί ο αλγόριθμος του μεγαλύτερου, θέτοντας σαν αρχική τιμή την 1η θέση του πίνακα.

Γενική μορφή:

Γέμισμα πίνακα	$i = 1$ Διαβάζω αριθμό $A[i]$ Μέχρι $i = 100$
	Καταχωρώ στην Max την $A[1]$ σαν αρχική τιμή
Εύρεση Max	$i = 2$ Έλεγχος της Max με το $A[i]$ Μέχρι $i = 100$
	Τυπώνω Max

ΠΡΟΓΡΑΜΜΑ $Max_Μονοδιάστατο$ ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: $A[100]$, i , Max

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

 ΓΡΑΨΕ 'Δώσε αριθμό:'

 ΔΙΑΒΑΣΕ $A[i]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

$Max \leftarrow A[1]$

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100

 ΑΝ $A[i] > Max$ ΤΟΤΕ

$Max \leftarrow A[i]$

 ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Μεγαλύτερος ο', Max

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ $Max_Μονοδιάστατο$

4. Να γίνει πρόγραμμα που να γεμίζει ένα δισδιάστατο πίνακα 5x3 με ακέραιους αριθμούς, από το πληκτρολόγιο. Στη συνέχεια να υπολογίζει και να τυπώνει:
- α) το μικρότερο στοιχείο κάθε γραμμής
 - β) το μεγαλύτερο στοιχείο κάθε στήλης
 - γ) το μέσο όρο όλων των αριθμών.

Λύση

Γενική μορφή:

Γέμισμα πίνακα	[$i = 1$ [$j = 1$ Διαβάζω το στοιχείο A [i,j] Μέχρι $j = 3$ Μέχρι $i = 5$
Εύρεση μικρότερου ανά γραμμή	[$i = 1$ Αρχική τιμή στο Min το A [i, 1] J = 2 Έλεγχός της Min με το A [i, j] Μέχρι το $j = 3$ Εκτύπωση της Min Μέχρι το $i = 5$
Εύρεση μεγαλύτερου ανά στήλη	[$j = 1$ Αρχική τιμή στο Max το A [1, j] I = 2 Έλεγχός της Max με το A [i, j] Μέχρι $i = 5$ Εκτύπωση της Max Μέχρι $j = 3$
Υπολογισμός του Μέσου Όρου όλων των αριθμών	[Μηδενισμός αθροιστή $i = 1$ [$j = 1$ Πρόσθεση στον Αθροιστή το A [i, j] Μέχρι $j = 3$ Μέχρι $i = 5$ Υπολογισμός μέσου όρου Αθροιστής/15 Εκτύπωση Μέσου Όρου

ΠΡΟΓΡΑΜΜΑ Min_Max_MO

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[5,3], i, j, Max, Min, sum

ΠΡΑΓΜΑΤΙΚΕΣ: MO

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 3

ΓΡΑΨΕ 'Δώσε ακέραιο αριθμό'

ΔΙΑΒΑΣΕ A [i, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

 Min ← A[i, 1]

ΓΙΑ j **ΑΠΟ** 2 **ΜΕΧΡΙ** 3

ΑΝ A [i, j] < Min **ΤΟΤΕ**

 Min ← A [i, j]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Ο μικρότερος αριθμός της', i, 'γραμμής=', Min

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 3

 Max ← A[1, j]

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 5

ΑΝ A [i, j] > Max **ΤΟΤΕ**

 Max ← A [i, j]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Ο μεγαλύτερος αριθμός της', j, 'στήλης=', Max

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

sum ← 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 3

 sum ← sum + A [i, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

MO ← sum/15

ΓΡΑΨΕ 'Μέσος όρος=', MO

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Min_Max_MO

2.6. Τυπικές επεξεργασίες πινάκων

Τα προγράμματα που χρησιμοποιούν πίνακες έχουν συγκεκριμένες επεξεργασίες, οι οποίες είναι:

- Υπολογισμός αθροισμάτων στοιχείων του πίνακα
- Εύρεση του μέγιστου ή ελάχιστου στοιχείου
- Ταξινόμηση των στοιχείων του πίνακα
- Αναζήτηση ενός στοιχείου του πίνακα
- Συγχώνευση δύο πινάκων.

2.6.1. Ταξινόμηση πίνακα

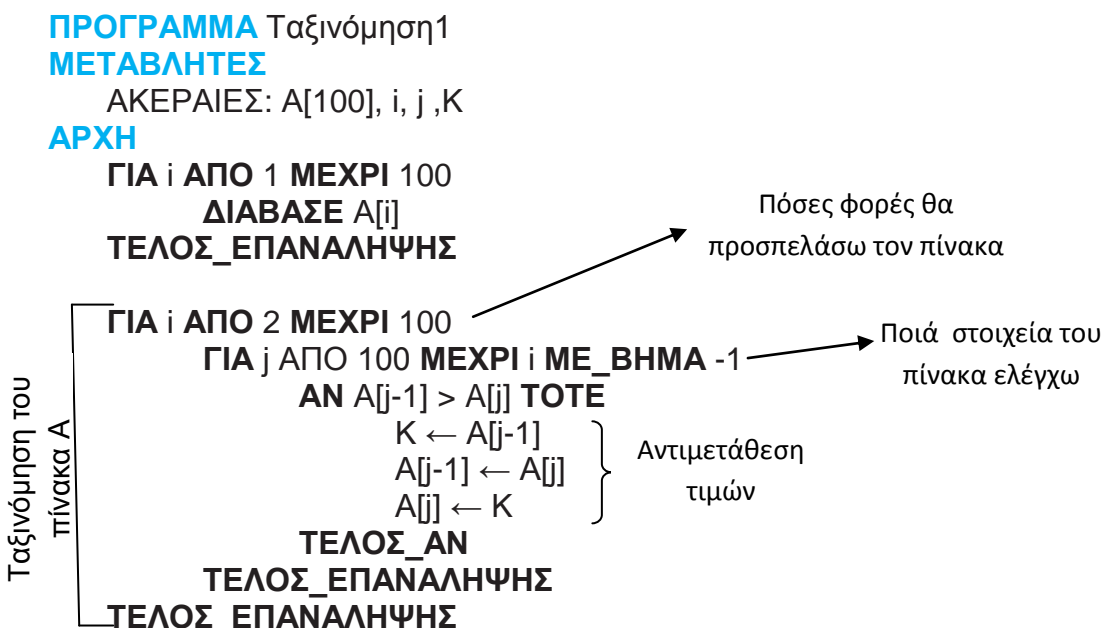
- Ταξινόμηση ονομάζουμε την τακτοποίηση των κόμβων μιας δομής με μια ιδιαίτερη σειρά.
- Η ταξινόμηση μπορεί να γίνει:
 1. κατ' αύξουσα τάξη (δηλ. από το μικρότερο προς το μεγαλύτερο)
 2. κατά φθίνουσα τάξη (δηλ. από το μεγαλύτερο προς το μικρότερο)
- Υπάρχουν διάφοροι αλγόριθμοι ταξινόμησης πίνακα. Εδώ θα αναπτυχθεί ο αλγόριθμος της ταξινόμησης ευθείας ανταλλαγής ή ταξινόμηση φυσαλίδας

2.6.1.1. Ταξινόμηση ευθείας ανταλλαγής ή φυσαλίδας (Bubblesort)

Με τη μέθοδο αυτή της ταξινόμησης συγκρίνουμε και ανταλλάσσουμε ζεύγη γειτονικών στοιχείων μέχρις ότου ταξινομηθούν όλα.

Με τη μέθοδο αυτή κάθε φορά γίνονται διαδοχικές προσπελάσεις στον πίνακα έτσι ώστε το μικρότερο στοιχείο της ακολουθίας να μετακινηθεί προς το αριστερό άκρο του πίνακα.

Περίπτωση 1^η (κατ' αύξουσα τάξη)



ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ταξινόμηση1

- Στην περίπτωση που η ταξινόμηση είναι κατά φθίνουσα τάξη, το μόνο που αλλάζει είναι η σύγκριση των στοιχείων του πίνακα, δηλαδή έχω:
ΑΝ A[j-1] < A[j] ΤΟΤΕ ...

Περίπτωση 2^η

Στην περίπτωση αυτή αναπτύσσεται ο αλγόριθμος ταξινόμησης με τη μέθοδο της φυσαλίδας καλύτερα και εξυπνότερα. Δηλ. όταν ο πίνακας είναι ταξινομημένος, τότε σταματά ο έλεγχος των στοιχείων για ταξινόμηση.

ΠΡΟΓΡΑΜΜΑ Ταξινόμηση1
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], i, j, K
ΛΟΓΙΚΕΣ: Flag

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
ΔΙΑΒΑΣΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Ταξινόμηση του
πίνακα A

Flag ← ΑΛΗΘΗΣ
i ← 2
ΟΣΟ i <= 100 ΚΑΙ flag = ΑΛΗΘΗΣ **ΕΠΑΝΑΛΑΒΕ**
flag ← ΨΕΥΔΗΣ
ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
ΑΝ A[j-1] > A[j] ΤΟΤΕ
K ← A[j - 1]
A[j-1] ← A[j]
A[j] ← K
flag ← ΑΛΗΘΗΣ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
i ← i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ταξινόμηση1

2.6.2. Αναζήτηση στοιχείου πίνακα

Το πρόβλημα της αναζήτησης ενός στοιχείου σε πίνακα είναι ιδιαίτερα χρήσιμο σ' ένα μεγάλο πλήθος εφαρμογών.

Υπάρχουν αρκετοί αλγόριθμοι για την αναζήτηση στοιχείου σε πίνακα που αποσκοπούν στην ταχύτητα εύρεσης του αναζητούμενου στοιχείου. Εμείς θα ασχοληθούμε με δύο αλγορίθμους :

1. Την Σειριακή αναζήτηση, και
2. Την Δυαδική αναζήτηση

2.6.2.1. Σειριακή αναζήτηση

Η πιο απλή μορφή αναζήτησης στοιχείου σε πίνακα είναι η **σειριακή ή γραμμική μέθοδος**.

Με τη μέθοδο αυτή ξεκινάμε από την αρχή του πίνακα και συγκρίνουμε κάθε τιμή του μια προς μια, με το στοιχείο που αναζητάμε.

Παρακάτω θα δοθούν δύο περιπτώσεις αλγόριθμου αναζήτησης για στοιχείο που είναι μοναδικό, δηλ. υπάρχει μόνο μια φορά και για στοιχείο που μπορεί να υπάρχει περισσότερες από μια φορές στον πίνακα.

Περίπτωση 1^η

Αναζήτηση του στοιχείου (key) σε ένα πίνακα A[100]. Το στοιχείο (key) είναι μοναδικό. Σε περίπτωση που βρεθεί να τυπώνεται η αντίστοιχη θέση του και το μήνυμα "Βρέθηκε", αλλιώς να τυπώνεται το μήνυμα "Δεν βρέθηκε".

ΠΡΟΓΡΑΜΑ Αναζήτηση1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], I, KEY

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

i ← 1

ΟΣΟ A[i] <> key ΚΑΙ i <= 100 ΕΠΑΝΑΛΑΒΕ

i ← i + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ A[i] = key ΤΟΤΕ

ΓΡΑΨΕ 'Βρέθηκε στη θέση:', i

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δεν βρέθηκε'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αναζήτηση1

Περίπτωση 2^η

Αναζήτηση του στοιχείου (key) σε ένα πίνακα A[100]. Το στοιχείο key υπάρχει πιθανότητα να βρίσκεται περισσότερες από μια φορές στον πίνακα.

Στην περίπτωση που βρεθεί να τυπώνονται οι θέσεις που βρέθηκε, αλλιώς να τυπώνεται το μήνυμα "Δεν βρέθηκε".

ΠΡΟΓΡΑΜΑ Αναζήτηση2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], I, KEY

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.

ΔΙΑΒΑΣΕ Key

flag ← ΨΕΥΔΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΑΝ key = A[i] **ΤΟΤΕ**

flag ← ΑΛΗΘΗΣ

ΓΡΑΨΕ 'Βρέθηκε στη θέση ', i

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ flag = ΨΕΥΔΗΣ **ΤΟΤΕ**

ΓΡΑΨΕ 'Δεν βρέθηκε'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αναζήτηση2

Παρατήρηση

Η σειριακή αναζήτηση είναι η πιο απλή μέθοδος, αλλά και η λιγότερο αποτελεσματική.

Η χρήση της δικαιολογείται μόνο στις παρακάτω περιπτώσεις:

- Ο πίνακας δεν είναι ταξινομημένος.
- Ο πίνακας έχει μικρό μέγεθος (συνήθως $n \leq 20$).
- Η αναζήτηση στον πίνακα γίνεται σπάνια.

2.6.2.2. Δυαδική αναζήτηση

Η δυαδική αναζήτησης είναι αποτελεσματικότερη από την σειριακή γιατί χρειάζεται πολύ λιγότερο χρόνο εκτέλεσης- κάτω από το μισό χρόνο- από ότι η σειριακή αναζήτηση.

Απαραίτητη προϋπόθεση είναι ότι ο πίνακας πρέπει να είναι ταξινομημένος ως προς το στοιχείο που αναζητούμε.

Η μέθοδος αυτή στηρίζεται στη συνεχή διαίρεση του πίνακα σύμφωνα με τα ακόλουθα βήματα:

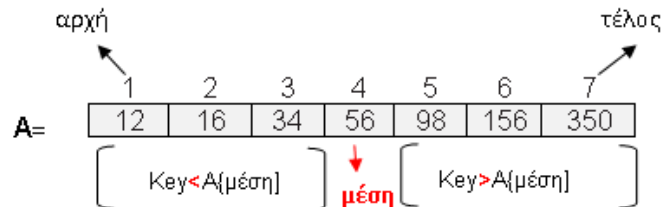
- Ελέγχουμε αν το ζητούμενο στοιχείο είναι ίσο με το στοιχείο του πίνακα που βρίσκεται τη μεσαία θέση.
- Αν όχι, ελέγχουμε αν το ζητούμενο στοιχείο είναι $<$ με το στοιχείο του πίνακα που βρίσκεται τη μεσαία θέση. Αν ισχύει οπότε αλλάζουμε το τέλος του πίνακα (δηλαδή η αναζήτηση γίνεται μόνο στο 1ο μισό του

πίνακα). Διαφορετικά αλλάζουμε την αρχή του πίνακα (δηλαδή ψάχνουμε μόνο στο 2ο μισό του πίνακα).

- Επαναλαμβάνονται τα προηγούμενα βήματα μέχρι να βρούμε το ζητούμενο στοιχείο. Το ζητούμενο στοιχείο δεν υπάρχει όταν μετά από τις διαδοχικές αυξομειώσεις των άκρων του πίνακα μέχρι να ισχύσει αρχή > τέλος.

Η παραπάνω διεργασία στηρίζεται ότι ο πίνακας είναι ταξινομημένος κατ' αύξουσα σειρά.

Π.χ.



Αντίστοιχα γίνεται και όταν ο πίνακας είναι ταξινομημένος κατά φθίνουσα σειρά.

Περίπτωση 1^η

Θεωρούμε ότι τα στοιχεία του πίνακα A είναι **ταξινομημένα κατά αύξουσα σειρά**.

ΠΡΟΓΡΑΜΑ Δυαδική_Αναζήτηση1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], KEY, τέλος, αρχή, μέση

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.

ΔΙΑΒΑΣΕ Key

Αρχή ← 1

Τέλος ← 100

Μέση ← (αρχή + τέλος) div 2

ΟΣΟ αρχή <= τέλος **ΚΑΙ** A[μέση] <> Key **ΕΠΑΝΑΛΑΒΕ**

ΑΝ Key < A[μέση] **ΤΟΤΕ**

Τέλος ← μέση-1

ΑΛΛΙΩΣ

Αρχή ← μέση + 1

ΤΕΛΟΣ_ΑΝ

Μέση ← (αρχή + τέλος) div 2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ A[μέση] = Key **ΤΟΤΕ**

ΓΡΑΨΕ 'Βρέθηκε στη θέση ',μέση

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δεν βρέθηκε'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Δυαδική_Αναζήτηση1

Περίπτωση 2^η

Θεωρούμε ότι τα στοιχεία του πίνακα A είναι ταξινομημένα κατά φθίνουσα σειρά.

ΠΡΟΓΡΑΜΑ Δυαδική_Αναζήτηση2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], KEY, τέλος, αρχή, μέση

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.

ΔΙΑΒΑΣΕ Key

Αρχή ← 1

Τέλος ← 100

μέση ← (αρχή + τέλος) div 2

ΟΣΟ αρχή <= τέλος ΚΑΙ A[μέση] <> Key ΕΠΑΝΑΛΑΒΕ

ΑΝ Key < A[μέση] ΤΟΤΕ

Αρχή ← μέση + 1

ΑΛΛΙΩΣ

Τέλος ← μέση - 1

ΤΕΛΟΣ_ΑΝ

Μέση ← (αρχή + τέλος) div 2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ A[μέση] = Key ΤΟΤΕ

ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δεν βρέθηκε'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Δυαδική_Αναζήτηση2

2.7. Αλγόριθμος συγχώνευσης δύο πινάκων

Η συγχώνευση είναι μια από τις βασικές λειτουργίες σε πίνακες. Σκοπός της είναι η δημιουργία ενός πίνακα από τα δεδομένα δύο πινάκων. Υπάρχουν δύο περιπτώσεις:

Περίπτωση 1^η

Έστω πίνακας A[25] με τα ονόματα του 1ου τμήματος τεχνολογικής κατεύθυνσης της Γ' Λυκείου και B[21] τα ονόματα του 2ου τμήματος τεχνολογικής κατεύθυνσης της Γ' Λυκείου. Να δημιουργηθεί πίνακας με τα ονόματα και των δύο τμημάτων.

Λύση

ΠΡΟΓΡΑΜΜΑ Συγχώνευση1
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[25], B[25], i,

κ

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 25

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 21

ΔΙΑΒΑΣΕ B[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Είσοδος
δεδομένων

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 25

Γ[i] ← A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

κ ← 25

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 21

Γ[i+κ] ← B[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

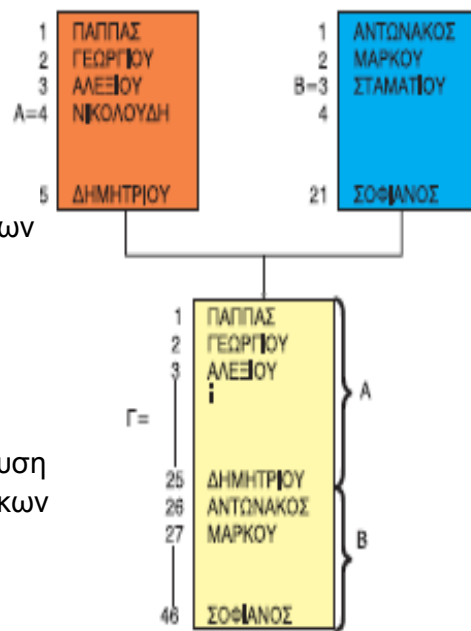
Συγχώνευση
των πινάκων

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 46

ΓΡΑΨΕ Γ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Συγχώνευση1



Περίπτωση 2^η

Έστω ότι έχουμε τα δεδομένα της παραπάνω άσκησης και οι πίνακες A,B είναι ταξινομημένοι κατά αύξουσα τάξη. Να δημιουργηθεί ο πίνακας Γ με τα ονόματα και των δύο τμημάτων με απευθείας ταξινόμηση.

Λύση

Ανάλυση

1. Στη περίπτωση αυτή ελέγχουμε ένα στοιχείο του πίνακα A με ένα στοιχείο του πίνακα B (ξεκινώντας από την πρώτη θέση).

2. Το στοιχείο που είναι μικρότερο καταχωρείται στο πίνακα Γ και αυξάνεται κατά μία μονάδα η θέση του πίνακα του οποίου πήρα το στοιχείο του και το καταχώρησα στον πίνακα Γ.
3. Η διαδικασία αυτή σταματά όταν τελειώσουν τα στοιχεία που συγκρίνουμε σε έναν από δύο πίνακες, τον Α ή Β πίνακα.
4. Στη συνέχεια, από τον πίνακα που περίσσεψαν στοιχεία, τα καταχωρώ στο πίνακα Γ.

ΠΡΟΓΡΑΜΜΑ Συγχώνευση2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[25], B[25], i, κ

ΑΡΧΗ

Είσοδος
δομικών

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 25
ΔΙΑΒΑΣΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 21
ΔΙΑΒΑΣΕ B[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Συνένωση 2
ταξινομημένων
πινάκων με
απευθείας
ταξινόμηση

i ← 1
j ← 1
κ ← 1
ΟΣΟ i ≤ 25 **ΚΑΙ** j ≤ 21 **ΕΠΑΝΑΛΑΒΕ**
ΑΝ A[i] < B[j] **ΤΟΤΕ**
 Γ[κ] ← A[i]
 i ← i + 1
ΑΛΛΙΩΣ
 Γ[κ] ← B[j]
 j ← j + 1
ΤΕΛΟΣ_ΑΝ
κ ← κ + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ i > 25 **ΤΟΤΕ**
 ΓΙΑ i **ΑΠΟ** j **ΜΕΧΡΙ** 21
 Γ[κ] ← B[i]
 κ ← κ + 1
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ
 ΓΙΑ i **ΑΠΟ** i **ΜΕΧΡΙ** 25
 Γ[κ] ← A[i]
 κ ← κ + 1
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ
ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 46
 ΓΡΑΨΕ Γ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Συγχώνευση2

2.8. Παράλληλοι πίνακες

Όταν έχουμε εγγραφές με πεδία διαφορετικού τύπου οι οποίες καταχωρούνται σε πίνακες τότε κάθε πεδίο είναι ένας διαφορετικός πίνακας, λόγω διαφορετικού τύπου. Οι πίνακες αυτοί συνδέονται μεταξύ τους με τέτοιο τρόπο ώστε η κάθε εγγραφή να έχει όλα τα στοιχεία της με την ίδια τιμή δείκτη, δηλ. να βρίσκονται στην ίδια θέση στους πίνακες.

Π.χ. αν έχουμε εγγραφές που αφορούν στοιχεία μισθοδοσίας υπαλλήλων μιας εταιρείας.

ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΟΙΚΟΓΕΝΙΑΚΗ ΚΑΤΑΣΤΑΣΗ	ΑΡΙΘΜΟΣ ΠΑΙΔΙΩΝ	ΜΙΣΘΟΣ	ΚΡΑΤΗΣΕΙΣ
ΠΑΠΠΑΣ	ΓΕΩΡΓΙΟΣ	ΕΓΓΑΜΟΣ	2	1,400	10%
ΓΕΩΡΓΙΟΥ	ANNA	ΑΓΑΜΗ	0	1,150	20%
ΝΑΝΟΥ	ΣΟΦΙΑ	ΕΓΓΑΜΗ	1	1,520	16%
...

Τα παραπάνω δεδομένα μπορώ να τα κρατήσω σε 6 μονοδιάστατος πίνακες, γιατί άλλα στοιχεία είναι χαρακτήρες και άλλα αριθμοί. Οπότε είναι αδύνατη η χρήση διδιάστατου πίνακα. Επίσης, η τοποθέτηση των δεδομένων θα γίνει με τέτοιο τρόπο, ώστε το επώνυμο, όνομα, οικογενειακή κατάσταση, αριθμός παιδιών, μισθός και κρατήσεις ενός υπαλλήλου να βρίσκονται στην ίδια γραμμή σ' όλους τους πίνακες.

Παράδειγμα 1

Έχουμε 50 ονόματα εταιρειών και τις αντίστοιχες εισπράξεις τους. Να τυπωθούν τα ονόματα των εταιρειών με εισπράξεις μεγαλύτερες από το μέσο όρο εισπράξεων και τα ονόματα των εταιρειών με εισπράξεις μικρότερες από το μέσο όρο εισπράξεων.

Λύση

Ανάλυση

1. Το όνομα και η εισπραξη κάθε εταιρείας καταχωρούνται σε διαφορετικούς πίνακες που είναι παράλληλοι. Οπότε τα δεδομένα διαβάζονται ταυτόχρονα.
2. Βρίσκουμε το μέσο όρο από τον πίνακα των εισπράξεων.
3. Ελέγχουμε κάθε εισπραξη αν είναι μεγαλύτερη του μέσου όρου εισπράξης και τυπώνουμε το αντίστοιχο όνομα εταιρείας.
4. Ελέγχουμε κάθε εισπραξη αν είναι μικρότερη του μέσου όρου εισπράξης και τυπώνουμε το αντίστοιχο όνομα εταιρείας.

ΠΡΟΓΡΑΜΜΑ Εταιρεία
ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: ΕΤ[50]
ΠΡΑΓΜΑΤΙΚΕΣ: ΕΙΣ[50], ΜΟ, sum
ΑΚΕΡΑΙΕΣ: i

ΑΡΧΗ

Γέμισμα παράλληλων πινάκων	{	ΓΡΑΨΕ 'Δώσε 50 ονόματα εταιρειών και τις αντίστοιχες εισπράξεις τους' ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50 ΔΙΑΒΑΣΕ ΕΤ[i], ΕΙΣ[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Υπολογισμός ΜΟ εισπράξεων	{	sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50 sum ← sum + ΕΙΣ[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΜΟ ← sum/50
Τυπώνονται ονόματα εταιρειών με εισπράξεις >ΜΟ	{	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50 ΑΝ ΕΙΣ[i] > ΜΟ ΤΟΤΕ ΓΡΑΨΕ ΕΤ[i] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Τυπώνονται ονόματα εταιρειών με εισπράξεις <ΜΟ	{	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50 ΑΝ ΕΙΣ[i] < ΜΟ ΤΟΤΕ ΓΡΑΨΕ ΕΤ[i] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Εταιρεία

Παράδειγμα 2 (ταξινόμηση σε παράλληλους πίνακες)

Όταν ταξινομούμε παράλληλους πίνακες, κάνουμε τον έλεγχο ως προς τον πίνακα που ζητείται η ταξινόμηση, αλλά στην αντιμετάθεση των στοιχείων αντιμετωπίζουμε τις αντίστοιχες θέσεις όλων των παράλληλων πινάκων ως προς αυτόν που γίνεται η ταξινόμηση.

Π.χ. Να γίνει πρόγραμμα που δίνω τα ονόματα και τους μέσους όρους 50 μαθητών της Α΄ Λυκείου ενός σχολείου. Να τυπωθεί μια κατάσταση ταξινομημένη ως προς το μέσο όρο κατά φθίνουσα τάξη, δηλαδή από τον μεγαλύτερο βαθμό προς τον μικρότερο.

Λύση

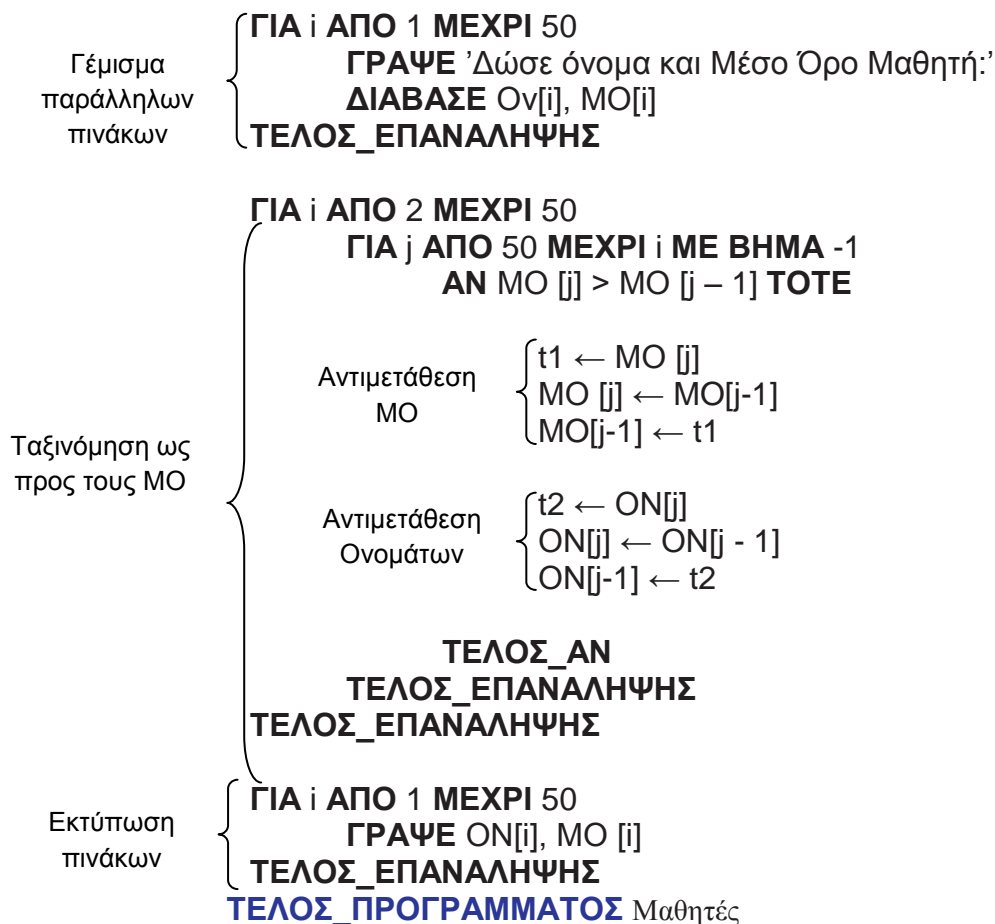
Ανάλυση

1. Καταχωρώ σε δύο παράλληλους πίνακες τα ονόματα και τους μέσους όρους 50 μαθητών
2. Εφαρμόζω τον αλγόριθμο της ταξινόμησης στον πίνακα των μέσων όρων.

ΠΡΟΓΡΑΜΜΑ Μαθητές ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: ON[50], t2
ΠΡΑΓΜΑΤΙΚΟΙ: MO[50], t1
ΑΚΕΡΑΙΕΣ: i, j

ΑΡΧΗ



Παράδειγμα 3 (Διπλή ταξινόμηση σε παράλληλους πίνακες)

Χρησιμοποιώντας το προηγούμενο παράδειγμα να εκτυπωθεί μια κατάσταση μαθητών ταξινομημένη ως προς το Μέσο Όρο τους κατά φθίνουσα τάξη και σε περίπτωση ισοβαθμίας η ταξινόμηση να γίνεται ως προς το όνομα κατά αύξουσα τάξη (αλφαβητικά).

Λύση

Ανάλυση

1. Καταχωρώ σε δύο παράλληλους πίνακες τα ονόματα και τους Μέσους Όρους 50 μαθητών.
2. Ταξινομώ τους πίνακες ως προς τον Μέσο Όρο.
3. Ελέγχω μέσα στην ταξινόμηση αν οι Μέσοι Όροι είναι ίσοι και ταξινομώ ως προς το Όνομα του μαθητή.

ΠΡΟΓΡΑΜΜΑ Διπλή_Ταξινόμηση

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Ον[50], t2

ΠΡΑΓΜΑΤΙΚΟΙ: ΜΟ[50], t1

ΑΚΕΡΑΙΕΣ: i, j

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 50

ΓΡΑΨΕ 'Δώσε όνομα και Μέσο Όρο Μαθητή:'

ΔΙΑΒΑΣΕ Ον[i], ΜΟ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 50

ΓΙΑ j **ΑΠΟ** 50 **ΜΕΧΡΙ** i **ΜΕ_ΒΗΜΑ** -1

ΑΝ ΜΟ [j] > ΜΟ [j - 1] **ΤΟΤΕ**

t1 ← ΜΟ [j]

ΜΟ [j] ← ΜΟ[j-1]

ΜΟ[j-1] ← t1

t2 ← ΟΝ[j]

ΟΝ[j] ← ΟΝ[j - 1]

ΟΝ[j-1] ← t2

ΤΕΛΟΣ_ΑΝ

ΑΝ ΜΟ [j]=ΜΟ[j-1] **ΤΟΤΕ**

ΑΝ ΟΝ[j] < ΟΝ[j-1] **ΤΟΤΕ**

t2 ← ΟΝ[j]

ΟΝ[j] ← ΟΝ[j - 1]

ΟΝ[j - 1] ← t2

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 50

ΓΡΑΨΕ ΟΝ[i], ΜΟ [i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Διπλή_Ταξινόμηση

2.9. Λυμένες ασκήσεις

2.9.1. Άσκηση 1. Μονοδιάστατος πίνακας

Έστω μονοδιάστατος πίνακας ακέραιων A , 100 θέσεων. Να γραφτεί πρόγραμμα που να υπολογίζει τα παρακάτω:

- i. Γέμισμα πίνακα με ακέραιους αριθμούς.
- ii. Εκτύπωση πίνακα από την 1η θέση μέχρι την 100η θέση.
- iii. Εκτύπωση πίνακα από την 100η θέση μέχρι την 1η θέση.
- iv. Να τυπωθεί το άθροισμα των στοιχείων του πίνακα.
- v. Να τυπωθεί ο μέσος όρος των στοιχείων του πίνακα.
- vi. Να τυπωθεί το άθροισμα των στοιχείων του πίνακα που βρίσκεται στις ζυγές θέσεις και το άθροισμα που βρίσκεται στις μονές θέσεις.
- vii. Να τυπωθεί το γινόμενο των 10 πρώτων στοιχείων του πίνακα A .
- viii. Να αντιμεταθέσουμε τα συμμετρικά στοιχεία του πίνακα A και να τυπωθεί.
- ix. Να υπολογισθεί το άθροισμα των 50 πρώτων στοιχείων του πίνακα A και το άθροισμα των 50 τελευταίων στοιχείων του πίνακα A , αν τα δύο αθροίσματα είναι ίσα να τυπώνεται το μήνυμα "Ίσα αθροίσματα".
- x. Αν τα συμμετρικά στοιχεία του πίνακα είναι ίσα τότε να εμφανίζει το μήνυμα "Πίνακας συμμετρικός".
- xi. Να τυπώνεται το πλήθος των στοιχείων που είναι μικρότερα από το Μέσο Όρο τους.
- xii. Να αναζητηθεί η τιμή 42 και αν βρεθεί να τυπωθεί η θέση της.
- xiii. Να ταξινομεί κατά φθίνουσα τάξη τον πίνακα A .
- xiv. Να ελέγχει αν ο πίνακας είναι ταξινομημένος και να βγάζει κατάλληλο μήνυμα.

Λύση

ΠΡΟΓΡΑΜΜΑ Μονοδιάστατος_πίνακας
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], i, j, sum, sum1, sum2, suma, sumμ, K, μ, GIN
ΠΡΑΓΜΑΤΙΚΕΣ: MO
ΛΟΓΙΚΕΣ: flag

ΑΡΧΗ

- i. Γέμισμα πίνακα
- ii.
- iii.
- iv.
- v.
- vi. Ελέγχουμε την κάθε θέση πίνακα A[i] αν έχει άρτια θέση ($i \bmod 2 = 0$) ή όχι και προσθέτουμε το στοιχείο στον κατάλληλο αθροιστή.
- ΓΡΑΨΕ** 'Δώσε 100 ακέραιους αριθμούς:'
ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100
 ΔΙΑΒΑΣΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
- ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100
 ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
- ΓΙΑ** i **ΑΠΟ** 100 **ΜΕΧΡΙ** 1 **ΜΕ_ΒΗΜΑ** -1
 ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
- sum ← 0
ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100
 sum ← sum + A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Αθροισμα στοιχείων:', sum
- MO ← sum/100
ΓΡΑΨΕ 'Μέσος Όρος στοιχείων =', MO
- suma ← 0
sumμ ← 0
ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100
 ΑΝ $i \bmod 2 = 0$ **ΤΟΤΕ**
 suma ← suma + A[i]
 ΑΛΛΙΩΣ
 sumμ ← sumμ + A[i]
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Αθροισμα αριθμών σε άρτιες θέσεις:', suma
ΓΡΑΨΕ 'Αθροισμα αριθμών σε μονές θέσεις:', sumμ

vii. Η μεταβλητή

GIN είναι ο
πολλωστής με
αρχική τιμή το 1, το
οποίο είναι το
ουδέτερο στοιχείο
του πολλωσμού.

```

GIN ← 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
    GIN ← GIN * A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το γινόμενο των 10 πρώτων αριθμών :',GIN
    
```

viii. Αντιμεταθέτει τα:

i	100-i+1
1	100-1+1=100
2	100-2+1=99
...
50	100-50+1=49

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    K ← A[i]
    A[i] ← A[100 - i + 1]
    A[100 - i + 1] ← K
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    
```

Αντιμετάθεση

ix.

```

sum1 ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    sum1 ← sum1 + A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
sum2 ← 0
ΓΙΑ i ΑΠΟ 51 ΜΕΧΡΙ 100
    sum2 ← sum2 + A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ sum1 = sum2 ΤΟΤΕ
    ΓΡΑΨΕ 'Ίσα αθροίσματα'
ΤΕΛΟΣ_ΑΝ
    
```

Υπολογισμός αθροίσματος των 50 πρώτων αριθμών

Υπολογισμός αθροίσματος των 50 επόμενων αριθμών

x. Έλεγχος των

```

flag ← ΨΕΥΔΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    ΑΝ A[i] <> A[100 - i + 1] ΤΟΤΕ
        flag ← ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ flag = ΨΕΥΔΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Πίνακας συμμετρικός'
ΤΕΛΟΣ_ΑΝ
    
```

A[1], A[100]
A[2], A[99]
A[3], A[98]
...
A[49], A[50]

xi. { $\mu \leftarrow 0$
ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100
ΑΝ $A[i] < \mu$ **ΤΟΤΕ**
 $\mu \leftarrow \mu + 1$
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Πλήθος:', μ

xii. Αναζήτηση { **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100
ΑΝ $A[i] = 42$ **ΤΟΤΕ**
ΓΡΑΨΕ i
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

xiii. Ταξινόμηση { **ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 100
ΓΙΑ j **ΑΠΟ** 100 **ΜΕΧΡΙ** i **ΜΕ_ΒΗΜΑ** -1
ΑΝ $A[j] > A[j - 1]$ **ΤΟΤΕ**
 $K \leftarrow A[j]$
 $A[j] \leftarrow A[j-1]$
 $A[j-1] \leftarrow K$
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

xiv. Έλεγχος αν ο πίνακας είναι ταξινομημένος { $flag \leftarrow \text{ΑΛΗΘΗΣ}$
ΓΙΑ i **ΑΠΟ** 100 **ΜΕΧΡΙ** 2 **ΜΕ_ΒΗΜΑ** -1
ΑΝ $A[i] > A[i - 1]$ **ΤΟΤΕ**
 $flag \leftarrow \text{ΨΕΥΔΗΣ}$
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ $flag = \text{ΑΛΗΘΗΣ}$ **ΤΟΤΕ**
ΓΡΑΨΕ 'Πίνακας ταξινομημένος κατά φθίνουσα τάξη'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μονοδιάστατος_πίνακας

2.9.2. Άσκηση 2. Δισδιάστατος πίνακας

Έστω δισδιάστατος πίνακας A , 10×20 με ακέραιους αριθμούς. Να γραφτεί πρόγραμμα που να:

- i. Γεμίζει τον πίνακα A κατά γραμμή.
- ii. Γεμίζει τον πίνακα A κατά στήλη.
- iii. Να τυπώνει τα στοιχεία του πίνακα A , εμφανίζοντάς τα κατά γραμμή.
- iv. Να τυπώνει τα στοιχεία του πίνακα A , εμφανίζοντάς τα κατά στήλη.
- v. Να υπολογίζει τα αθροίσματα των γραμμών του πίνακα A , να τα τυπώνει και να τα καταχωρεί στον πίνακα $AOP_ΓΡ[20]$.
- vi. Να υπολογίζει τα αθροίσματα των στηλών του πίνακα A , να τα τυπώνει και να τα καταχωρεί στον πίνακα $AOP_ΣΤ[20]$.
- vii. Να υπολογίζει το άθροισμα όλων των στοιχείων του πίνακα A .
- viii. Να εμφανίζει το μέσο όρο των στοιχείων του πίνακα A .
- ix. Να τυπώνει τον μεγαλύτερο αριθμό της 12ης στήλης.
- x. Να τυπώνει τον μικρότερο αριθμό της 7ης γραμμής.
- xi. Να τυπώνει τον μεγαλύτερο και τον μικρότερο αριθμό όλου του πίνακα.
- xii. Να τυπώνει πόσα στοιχεία είναι μεγαλύτερα του μέσου όρου.
- xiii. Να διαβάζει έναν ακέραιο και να τον αναζητά στον πίνακα, όταν τον βρει να τυπώνεται η θέση του.
- xiv. Να ταξινομεί τα στοιχεία κάθε γραμμής κατά φθίνουσα τάξη.
- xv. Να ταξινομεί τα στοιχεία κάθε στήλης κατά αύξουσα τάξη.

Λύση

ΠΡΟΓΡΑΜΜΑ Δυσδιάστατος_πίνακας ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[10,20], ΑΘΡ_ΓΡ[10],ΑΘΡ_ΣΤ[20],i, j, sum
ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΑΡΧΗ

- i.** Διάβασμα ανά γραμμή
- ii.** Διάβασμα ανά στήλη
- iii.** Εκτύπωση ανά γραμμή
- iv.** Εκτύπωση ανά στήλη
- v.** Άθροισμα ανά γραμμή.
- ```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10 → γραμμή
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20 → στήλη
 ΓΡΑΨΕ 'Δώσε ακέραιο αριθμό'
 ΔΙΑΒΑΣΕ A[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20 → στήλη
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10 → γραμμή
 ΓΡΑΨΕ 'Δώσε ακέραιο αριθμό'
 ΔΙΑΒΑΣΕ A[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 ΓΡΑΨΕ A[i]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΡΑΨΕ A[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 sum ← 0
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 sum ← sum + A[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ 'Άθροισμα',i, 'γραμμής=',sum
 ΑΘΡ_ΓΡ[i] ← sum
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**vi.** Άθροισμα ανά στήλη.

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 sum ← 0
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 sum ← sum + A[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ 'Άθροισμα', j, 'στήλης=', sum
 ΑΘΡ_ΣΤ[j] ← sum
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

**vii.** Άθροισμα όλων των στοιχείων.

```
sum ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 sum ← sum + A[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Άθροισμα όλων των στοιχείων=', sum
```

**viii.** Ο Μέσος Όρος

```
ΜΟ ← sum/200
ΓΡΑΨΕ 'Μέσος Όρος =', ΜΟ
```

**ix.** Εφαρμόζουμε τον αλγόριθμο του μεγαλύτερου για την στήλη 12.

```
Μαχ ← A[1,12]
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10
 ΑΝ A[i,12] > Μαχ ΤΟΤΕ
 Μαχ ← A[i,12]
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Ο μεγαλύτερος της 12ης στήλης =', Μαχ
```

**x.** Εφαρμόζουμε τον αλγόριθμο του μικρότερου για την γραμμή 7.

```
Μιν ← A[7, 1]
ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 20
 ΑΝ A[7, j] < Μιν ΤΟΤΕ
 Μιν ← A[7, j]
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Ο μικρότερος της 7ης γραμμής =', Μιν
```

**xi.** Εφαρμόζουμε τον αλγόριθμο του μεγαλύτερου και του μικρότερου για τον πίνακα A.

```

Min ← A[1, 1]
Max ← A[1, 1]
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 ΑΝ A[i, j] < Min ΤΟΤΕ
 Min ← A[i, j]
 ΤΕΛΟΣ_ΑΝ
 ΑΝ A[i, j] > Max ΤΟΤΕ
 Max ← A[i, j]
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

**xii.** Μετράμε πόσα στοιχεία είναι μεγαλύτερα του μέσου όρου.

```

μ ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 ΑΝ A[i, j] > MO ΤΟΤΕ
 μ ← μ + 1
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Πλήθος:' μ

```

**xiii.** Αναζήτηση στοιχείου στον πίνακα A.

```

ΓΡΑΨΕ 'Δώσε ακέραιο για αναζήτηση'
ΔΙΑΒΑΣΕ x
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
 ΑΝ x=A[i, j] ΤΟΤΕ
 ΓΡΑΨΕ 'Βρέθηκε στη θέση=', i, j
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

**xiv.** Εφαρμογή της ταξινόμησης για κάθε γραμμή του πίνακα A. Δηλαδή κάθε γραμμή την θεωρώ σαν μονοδιάστατο πίνακα και την ταξινομώ

Επανάληψη γι' όλες τις γραμμές

```

ΓΙΑ gr ΑΠΟ 1 ΜΕΧΡΙ 10
 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 20
 ΓΙΑ j ΑΠΟ 20 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
 ΑΝ A[gr, j] > A[gr, j - 1] ΤΟΤΕ
 K ← A[gr, j]
 A[gr,j] ← A[gr, j - 1]
 A[gr, j - 1] ← K
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```



**xv.** Εφαρμογή της ταξινόμησης για κάθε στήλη του πίνακα A. Δηλαδή κάθε στήλη την θεωρώ σαν μονοδιάστατο πίνακα και την ταξινομώ

```

Επανάληψη γι' όλες τις στήλες
Ταξινόμηση μιας στήλης
ΓΙΑ st ΑΠΟ 1 ΜΕΧΡΙ 20
 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10
 ΓΙΑ j ΑΠΟ 10 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
 ΑΝ A[j,st]<A[j-1, st] ΤΟΤΕ
 K ← A[j, st]
 A[j,st] ← A[j - 1, st]
 A[j - 1, st] ← K
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Δισδιάστατος_πίνακας

```

### 2.9.3. Άσκηση 3

Ένα κτίριο με γραφεία έχει 5 ορόφους με 12 γραφεία ανά όροφο. Να δημιουργηθεί ένας δισδιάστατος πίνακας 5x12 όπου η κάθε γραμμή θα αντιπροσωπεύει τον όροφο και σε κάθε θέση του πίνακα θα αποθηκεύουμε το πλήθος των υπαλλήλων που έχει κάθε γραφείο. Να γραφτεί πρόγραμμα που:

- Να διαβάξει το πλήθος των υπαλλήλων κάθε γραφείου ανά όροφο και να τους καταχωρεί σε πίνακα ΚΤ\_ΓΡ[5,12]
- Να υπολογίζει και να εμφανίζει τον αριθμό και τον όροφο του γραφείου με τους περισσότερους και τους λιγότερους υπαλλήλους.
- Να υπολογίζει και να τυπώνει το πλήθος των υπαλλήλων ανά όροφο.
- Να υπολογίζει και να τυπώνει το σύνολο των υπαλλήλων όλων των γραφείων.

### Λύση

Αν δίνουμε μια γραφική παράσταση των πινάκων που θα δημιουργήσουμε και τη σχέση που έχουν μεταξύ τους, θα είχαμε:

**Σχεδιασμός Πινάκων**

|         |        | Γραφεία |    |    |    |    |    |   |    |   |    |    |    |    |                 |   |     |
|---------|--------|---------|----|----|----|----|----|---|----|---|----|----|----|----|-----------------|---|-----|
|         |        | 1       | 2  | 3  | 4  | 5  | 6  | 7 | 8  | 9 | 10 | 11 | 12 |    |                 |   |     |
| ΚΤ_ΓΡ = | Όροφοι | 1       | 3  | 5  | 10 | 2  | 4  | 7 | 11 | 9 | 2  | 1  | 3  | 5  | (=3+5+10+...+5) | 1 | 62  |
|         |        | 2       | 2  | 6  | 3  | 7  | 9  | 3 | 5  | 6 | 2  | 9  | 10 | 15 | (=2+6+3+...+15) | 2 | 77  |
|         |        | 3       | 24 | 32 | 9  | 10 | 3  | 6 | 5  | 3 | 9  | 8  | 5  | 6  |                 | 3 | 120 |
|         |        | 4       | 7  | 4  | 9  | 3  | 20 | 2 | 4  | 5 | 2  | 6  | 5  | 9  |                 | 4 | 76  |
|         |        | 5       | 5  | 2  | 11 | 12 | 21 | 5 | 9  | 3 | 2  | 4  | 3  | 10 |                 | 5 | 87  |

ΑΘ\_ΟΡ =

### Ανάλυση

- Γεμίζουμε το πίνακα ανά γραμμή, αφού ο κάθε όροφος εκφράζει γραμμή.
- Εφαρμόζουμε τον αλγόριθμο του ελάχιστου και μέγιστου ενώ ταυτόχρονα κρατάμε σε 2 μεταβλητές τη στήλη και τη γραμμή που βρέθηκε αντίστοιχα ο μικρότερος και ο μεγαλύτερος.
- Εφαρμόζουμε άθροισμα ανά γραμμή δημιουργώντας ένα μονοδιάστατο και παράλληλο πίνακα, ως προς τις γραμμές, γιατί ο κάθε όροφος είναι μια γραμμή πίνακα.
- Αθροίζω σ' ένα αθροιστή όλων τα στοιχεία του μονοδιάστατου πίνακα που δημιούργησα παραπάνω.

**ΠΡΟΓΡΑΜΜΑ** Κτίριο\_Γραφείων  
**ΜΕΤΑΒΛΗΤΕΣ**

ΑΚΕΡΑΙΕΣ: ΚΤ\_ΓΡ[5, 12], ΑΘ\_ΟΡ[5], i, j, sum, Min, Min\_i, Min\_j, Max, Max\_i, Max\_j

**ΑΡΧΗ**

Γέμισμα πίνακα ανά γραμμή

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
 ΓΡΑΨΕ 'Όροφος', i
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 12
 ΓΡΑΨΕ 'Δώσε υπαλλήλους γραφείου',j
 ΔΙΑΒΑΣΕ ΚΤ_ΓΡ[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Υπολογισμός Μικροτέρου και Μεγαλυτέρου σε Δισδιάστατο Πίνακα

```
Min ← ΚΤ_ΓΡ[1, 1]
Min_i ← 1
Min_j ← 1
Max ← ΚΤ_ΓΡ [1, 1]
Max_i ← 1
Max_j ← 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 12
 ΑΝ ΚΤ_ΓΡ[i, j] < Min ΤΟΤΕ
 Min ← ΚΤ_ΓΡ[i, j]
 Min_i ← i
 Min_j ← j
 ΤΕΛΟΣ_ΑΝ
 ΑΝ ΚΤ_ΓΡ[i, j] > Max ΤΟΤΕ
 Max ← ΚΤ_ΓΡ [i, j]
 Max_i ← i
 Max_j ← j
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το γραφείο',Min_j, 'στον όροφο',Min_i, 'έχει τους λιγότερους υπαλλήλους'
ΓΡΑΨΕ 'Το γραφείο',Max_j, 'στον όροφο',Max_i, 'έχει τους περισσότερους υπαλλήλους'
```

Άθροισμα ανά γραμμή & καταχώρηση σε παράλληλο ως προς τις γραμμές πίνακα

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
 sum ← 0
 ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 12
 sum ← sum + ΚΤ_ΓΡ[i, j]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 ΓΡΑΨΕ 'Το σύνολο των υπαλλήλων του',i,'όρφου είναι:',sum
 ΑΘΡ_ΟΡ[i] ← sum
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

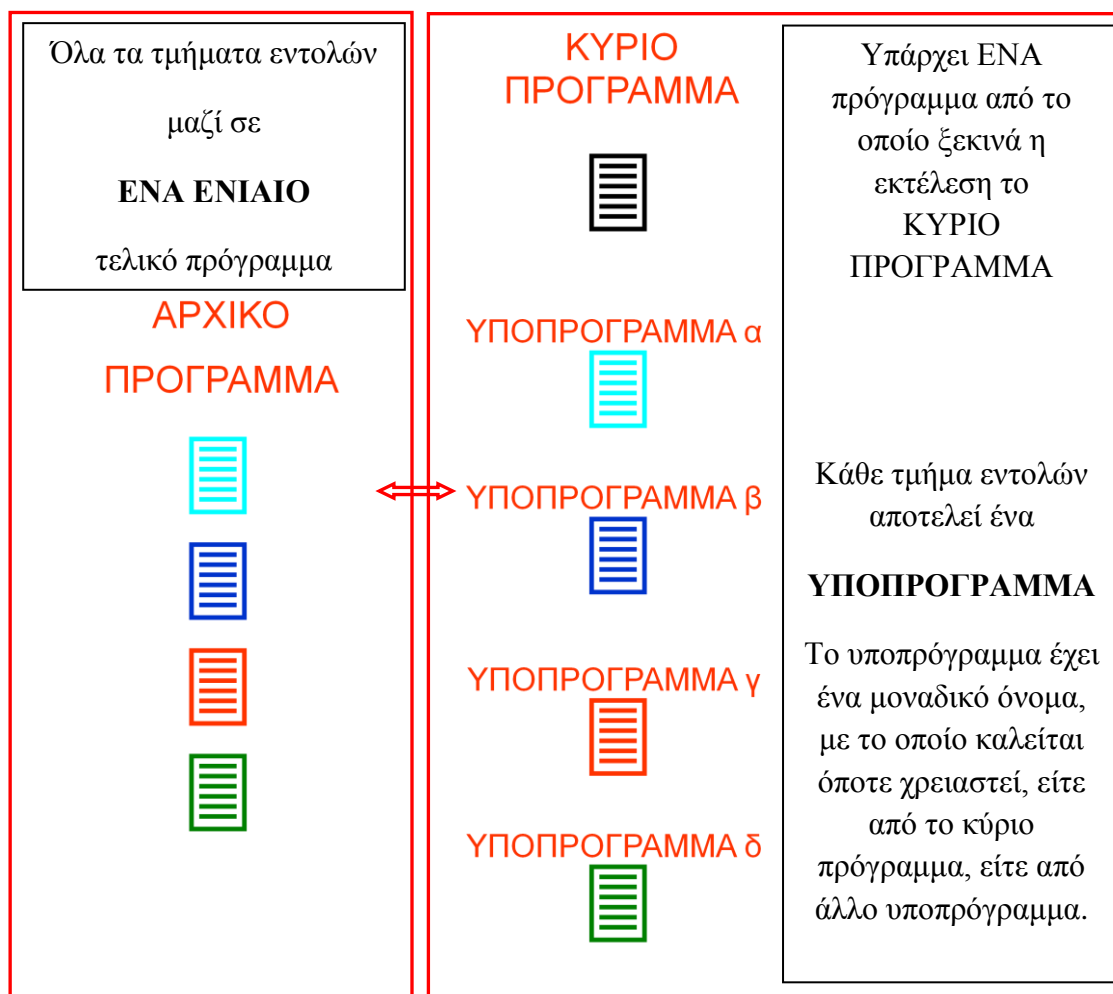
Άθροισμα  
μονοδιάστατου  
πίνακα

```
sum ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
 sum ← sum + ΑΘ_ΟΡ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το σύνολο όλων των υπαλλήλων, είναι:',s um
```

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** Κτίριο\_Γραφείων

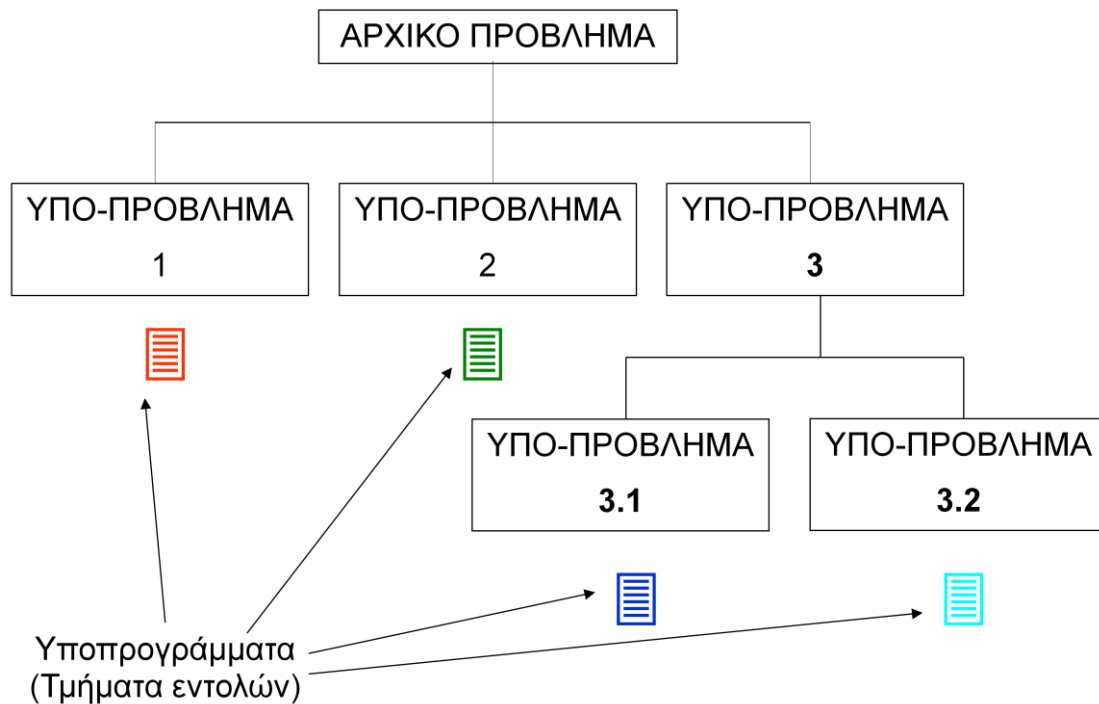
### 3. ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ-ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ (ΔΙΑΔΙΚΑΣΙΕΣ-ΣΥΝΑΡΤΗΣΕΙΣ)

Τμηματικός προγραμματισμός είναι η τεχνική σχεδίασης των προγραμμάτων ώστε ένα πρόγραμμα να αντικαθίσταται από ένα σύνολο απλούστερων μικρότερων και αυτόνομων τμημάτων προγραμμάτων. Και το αρχικό πρόγραμμα και το σύνολο των απλούστερων, μικρότερων και αυτόνομων τμημάτων προγραμμάτων όταν εκτελούνται, παράγουν ακριβώς τα ίδια αποτελέσματα. Στο σύνολο των απλούστερων μικρότερων και αυτόνομων τμημάτων προγραμμάτων, υπάρχει ένα από το οποίο ξεκινά η εκτέλεση των υποπρογραμμάτων, το οποίο καλείται ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ. Τα άλλα απλούστερα μικρότερα και αυτόνομα τμημάτων προγράμματα ονομάζονται υποπρογράμματα.



### 3.1. Που χρησιμοποιούνται τα υποπρογράμματα;

Χρησιμοποιούνται στην περίπτωση όπου έχουμε να επιλύσουμε ένα σύνθετο και πολύπλοκο πρόβλημα με την χρήση Η/Υ. Το αντίστοιχο πρόγραμμα που πρέπει να δημιουργήσουμε είναι σύνθετο, πολύπλοκο και επομένως αρκετά δύσκολο. Ο καλύτερος τρόπος για την επίλυση του σύνθετου προβλήματος είναι να διαιρεθεί σε μικρότερα προβλήματα και αυτά σε άλλα μικρότερα κ.ο.κ. (ιεραρχική σχεδίαση) και να σχεδιάσουμε υποπρογράμματα (απλά μικρά προγράμματα) για τα μικρότερα προβλήματα. Έτσι είναι δυνατό να αντιμετωπισθεί πιο εύκολα η επίλυση του αρχικού σύνθετου προβλήματος.



**Παράδειγμα 1:** Ένα πρόγραμμα εύρεσης και εμφάνισης του μέσου όρου βαθμολογίας της τάξης, μπορεί να διαιρεθεί στο κύριο πρόγραμμα, που θα ξεκινάει την εκτέλεση των υποπρογραμμάτων:

1. Υποπρόγραμμα για την εισαγωγή των δεδομένων (βαθμών).
2. Υποπρόγραμμα για την επεξεργασία των δεδομένων (εύρεση του μέσου όρου της βαθμολογίας).
3. Υποπρόγραμμα για την εξαγωγή των αποτελεσμάτων (εμφάνιση του μέσου όρου).

### 3.2. Τι πρέπει να προσεχθεί;

1. Τα υποπρογράμματα χαρακτηρίζονται από τις παρακάτω τρεις ιδιότητες:
  - Κάθε υποπρόγραμμα έχει μια μόνο είσοδο και μια έξοδο
  - Κάθε υποπρόγραμμα πρέπει να είναι ανεξάρτητο από τα άλλα
  - Όταν σχεδιάζεται ένα υποπρόγραμμα πρέπει να είναι ανεξάρτητο από άλλα υποπρογράμματα, ώστε να συντηρείται αυτόνομα.
  - Το υποπρόγραμμα είναι αυτόνομο και ανεξάρτητο, δηλώνονται όλες οι μεταβλητές που χρησιμοποιεί στις εντολές του και στο κύριο σώμα του γράφονται οι εντολές του.

- Κάθε υποπρόγραμμα πρέπει να μην είναι μεγάλο. Πρέπει να εκτελεί μόνο μια λειτουργία έτσι, ώστε να είναι εύκολα κατανοητό και να ελέγχεται.
2. Παράμετροι
- Ένα πρόγραμμα ενεργοποιείται (δηλ. εκτελεί τις εντολές του), όταν καλεστεί από άλλο υποπρόγραμμα ή πρόγραμμα (κύριο πρόγραμμα).
  - Για να επικοινωνεί το κύριο πρόγραμμα με το υπόλοιπο πρόγραμμα χρησιμοποιεί μεταβλητές που μεταφέρουν τιμές μεταξύ υποπρογράμματος και κύριου προγράμματος, δηλ. χρησιμοποιούνται σαν κανάλια επικοινωνίας, και λέγονται παράμετροι. Δηλαδή παράμετρος είναι μια μεταβλητή που επιτρέπει τη μεταφορά τιμών από ένα υποπρόγραμμα (τυπικές παράμετροι ή ορίσματα), στο κύριο πρόγραμμα (πραγματικές παράμετροι ή παράμετροι) ή σε άλλο υποπρόγραμμα. Δεν έχουν σημασία τα ονόματα των πραγματικών και τυπικών παραμέτρων (μπορεί και να είναι διαφορετικά), παρά μόνο το πλήθος τους, η σειρά τους (η αντιστοιχία τους γίνεται με την σειρά που είναι γραμμένες) και να είναι του ίδιου τύπου.
3. Μεταβλητές
- Το κύριο πρόγραμμα και τα υποπρογράμματα χρησιμοποιούν και μεταβλητές εκτός από τις παραμέτρους. Μετά το τέλος του υποπρογράμματος ο χώρος μνήμης που είχε δεσμευθεί για τις δηλωθείσες μεταβλητές του, στο τμήμα δηλώσεών του, αποδεσμεύεται.
4. Διαφορά μεταξύ παραμέτρων και μεταβλητών
- Η παράμετροι χρησιμοποιούνται για μεταφορά τιμών από τα υποπρογράμματα στο κύριο πρόγραμμα και αντίστροφα.
  - Η μεταβλητές χρησιμοποιούνται για καταχώρηση τιμών ξεχωριστά για το κύριο πρόγραμμα και τα υποπρογράμματα.
5. Είδη υποπρογραμμάτων
- Υπάρχουν δύο είδη υποπρογραμμάτων
- Διαδικασία
  - Συνάρτηση

### 3.3. Διαδικασίες (Παράδειγμα διαδικασίας)

#### Εκφώνηση προβλήματος

Να σχεδιασθεί κύριο πρόγραμμα και διαδικασία, έτσι ώστε όταν εκτελείται το κύριο πρόγραμμα (και με κλήση της αντίστοιχης διαδικασίας), θα ανταλλάσει τις τιμές μεταξύ δύο μεταβλητών, για δύο διαφορετικά ζεύγη τιμών που θα δίνονται από το πληκτρολόγιο. Συγκεκριμένα θα εισάγονται σε δύο μεταβλητές δύο τιμές από το πληκτρολόγιο και το πρόγραμμα θα ανταλλάσσει τις τιμές τους. Στη συνέχεια θα εισάγονται για δύο άλλες μεταβλητές δύο νέες τιμές από το πληκτρολόγιο και το πρόγραμμα θα ανταλλάσσει πάλι τις τιμές τους.

#### Ανάλυση προβλήματος

Σύμφωνα με το Παράδειγμα 1 (σελίδα 2), πρέπει να αναλυθεί το πρόβλημα σε υποπροβλήματα. Ποια όμως είναι τα υποπροβλήματα στο δοθέν πρόβλημα;

Η εμπειρία, που θα αποκτηθεί μετά την ολοκλήρωση αυτής της ενότητας βοηθά στον επιτυχή χωρισμό ενός προβλήματος σε υποπροβλήματα.

Για διευκόλυνση της κατανόησης της έννοιας των διαδικασιών (υποπρογραμμάτων) για αρχή, ακολουθείται η παρακάτω τεχνική, βασιζόμενοι σε πράγματα που έχουν ήδη διδαχθεί.

Χωρίς διαδικασίες η λύση του προβλήματος θα ήταν:

ΠΡΟΓΡΑΜΜΑ Ανταλλαγή

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α, β, ι, κ, temp

ΑΡΧΗ

ΓΡΑΨΕ 'Δωσε τιμές για α και β'

ΔΙΑΒΑΣΕ α, β

temp <-- α

α <-- β

β <-- temp

ΓΡΑΨΕ 'α=', α, 'β=', β

ΓΡΑΨΕ 'Δωσε τιμές για ι και κ'

ΔΙΑΒΑΣΕ ι, κ

temp <-- ι

ι <-- κ

κ <-- temp

ΓΡΑΨΕ 'ι=', ι, 'κ=', κ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ Ανταλλαγή

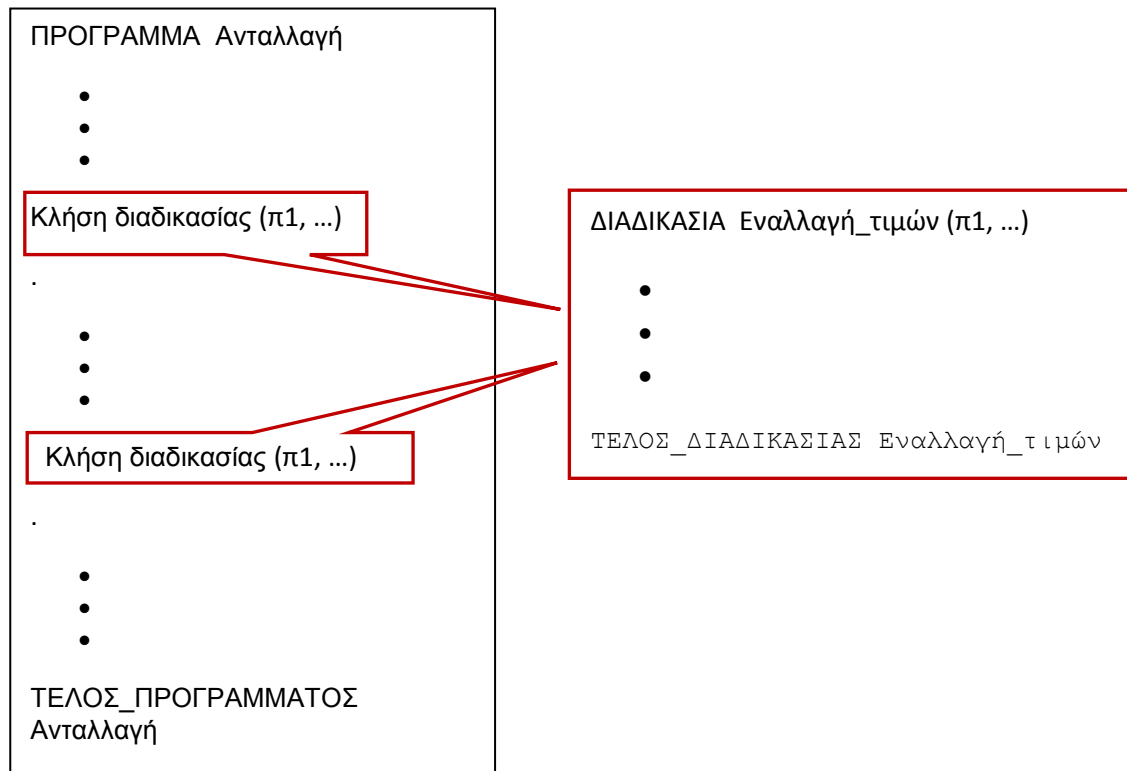
Τμήμα προγράμματος που μπορεί να εκτελεστεί αυτοτελώς.

Μάλιστα επαναλαμβάνεται δύο φορές σε διαφορετικά σημεία στο πρόγραμμα.



Στην περίπτωση χρήσης διαδικασιών και κύριου προγράμματος ποια θα μπορούσε να είναι η λύση;

Παρατηρείται ότι στο πρόγραμμα του παραδείγματος χωρίς διαδικασίες, υπάρχει ένα τμήμα προγράμματος που μπορεί να εκτελεστεί αυτοτελώς και μάλιστα αυτό το τμήμα επαναλαμβάνεται δύο φορές. Η σκέψη που δημιουργείται, για την επίλυση το ίδιου προβλήματος με διαδικασίες, είναι να σχεδιασθεί ένα κύριο πρόγραμμα (Ανταλλαγή) που θα χρησιμοποιεί κατάλληλα και όπου χρειάζεται, μια διαδικασία (Εναλλαγή\_τιμών), η οποία θα υλοποιεί ακριβώς το παραπάνω επαναλαμβανόμενο τμήμα προγράμματος. Η λύση σχηματικά θα μπορούσε να αναπαρασταθεί (Οπτικοποίηση της λύσης), ως εξής;



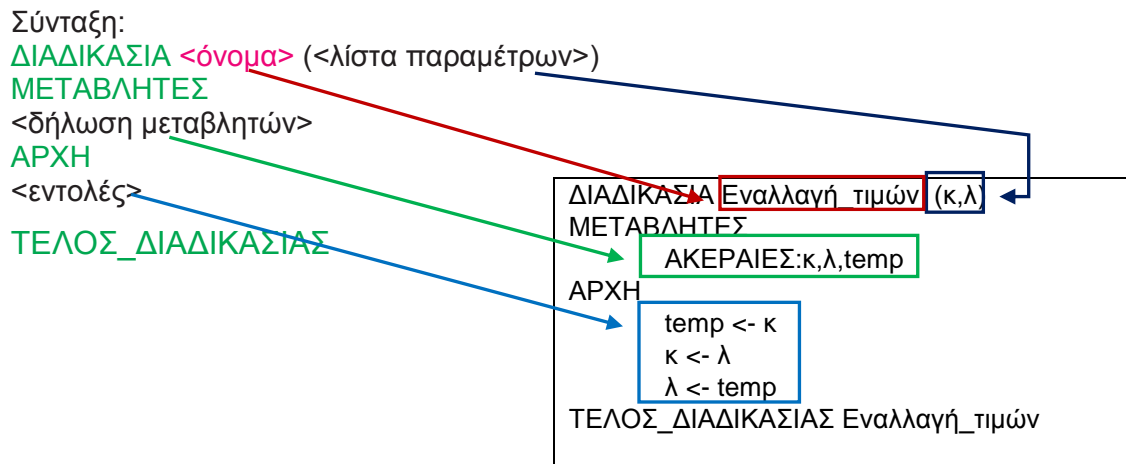
## Λύση

|                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>ΠΡΟΓΡΑΜΜΑ Ανταλλαγή<br/>         ΜΕΤΑΒΛΗΤΕΣ<br/>         ΑΚΕΡΑΙΕΣ: α, β, ι, κ<br/>         ΑΡΧΗ<br/>         ΔΙΑΒΑΣΕ α, β<br/>         ΚΑΛΕΣΕ Εναλλαγή_τιμών(α, β)<br/>         ΓΡΑΨΕ α, β<br/> <br/>         ΔΙΑΒΑΣΕ ι, κ<br/>         ΚΑΛΕΣΕ Εναλλαγή_τιμών(ι, κ)<br/>         ΓΡΑΨΕ ι, κ<br/>         ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ανταλλαγή</p> | <p>Κύριο πρόγραμμα, που εκτελεί δύο φορές κατά σειρά τις εξής λειτουργίες:</p> <ul style="list-style-type: none"> <li>• Διαβάζει σε δύο μεταβλητές, δύο ακεραίους αριθμούς.</li> <li>• Εναλλάσσει τις τιμές μεταξύ των μεταβλητών (διαδοχική κλήση διαδικασίας).</li> <li>• Εμφανίζει το περιεχόμενο των δύο μεταβλητών.</li> </ul> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                     |                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| ΔΙΑΔΙΚΑΣΙΑ Εναλλαγή_τιμών(κ, λ)<br>ΜΕΤΑΒΛΗΤΕΣ<br>ΑΚΕΡΑΙΕΣ:κ, λ,temp<br>ΑΡΧΗ<br>temp <- κ<br>κ <- λ<br>λ <- temp<br>ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Εναλλαγή_τιμών | <i>Διαδικασία που εναλλάσσει τις τιμές δύο μεταβλητών (κ,λ) μέσω της βοηθητικής μεταβλητής temp.</i> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|

## Ανάλυση της λύσης του προβλήματος

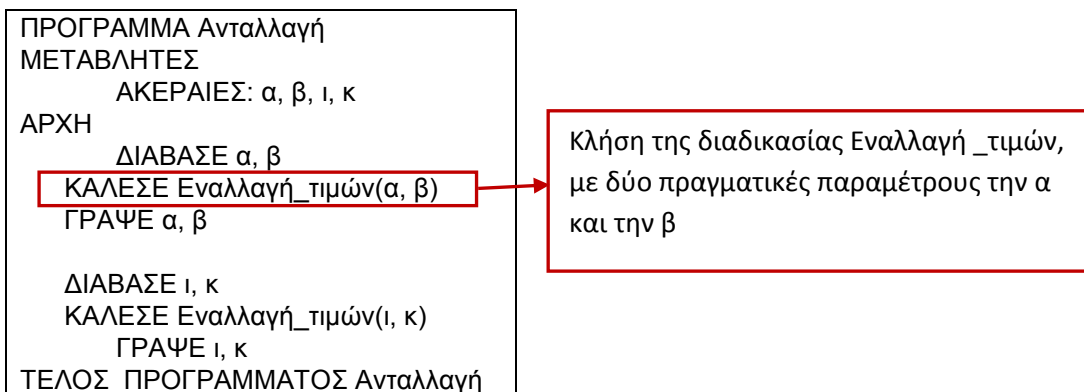
### 3.3.1. Πως συντάσσεται (ορίζεται) μια διαδικασία;



### 3.3.2. Πως το κύριο πρόγραμμα καλεί μια διαδικασία;

Πάντοτε έχουμε μόνο ένα ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ ενώ μπορούμε να έχουμε όσες ΔΙΑΔΙΑΚΑΣΙΕΣ χρειαζόμαστε

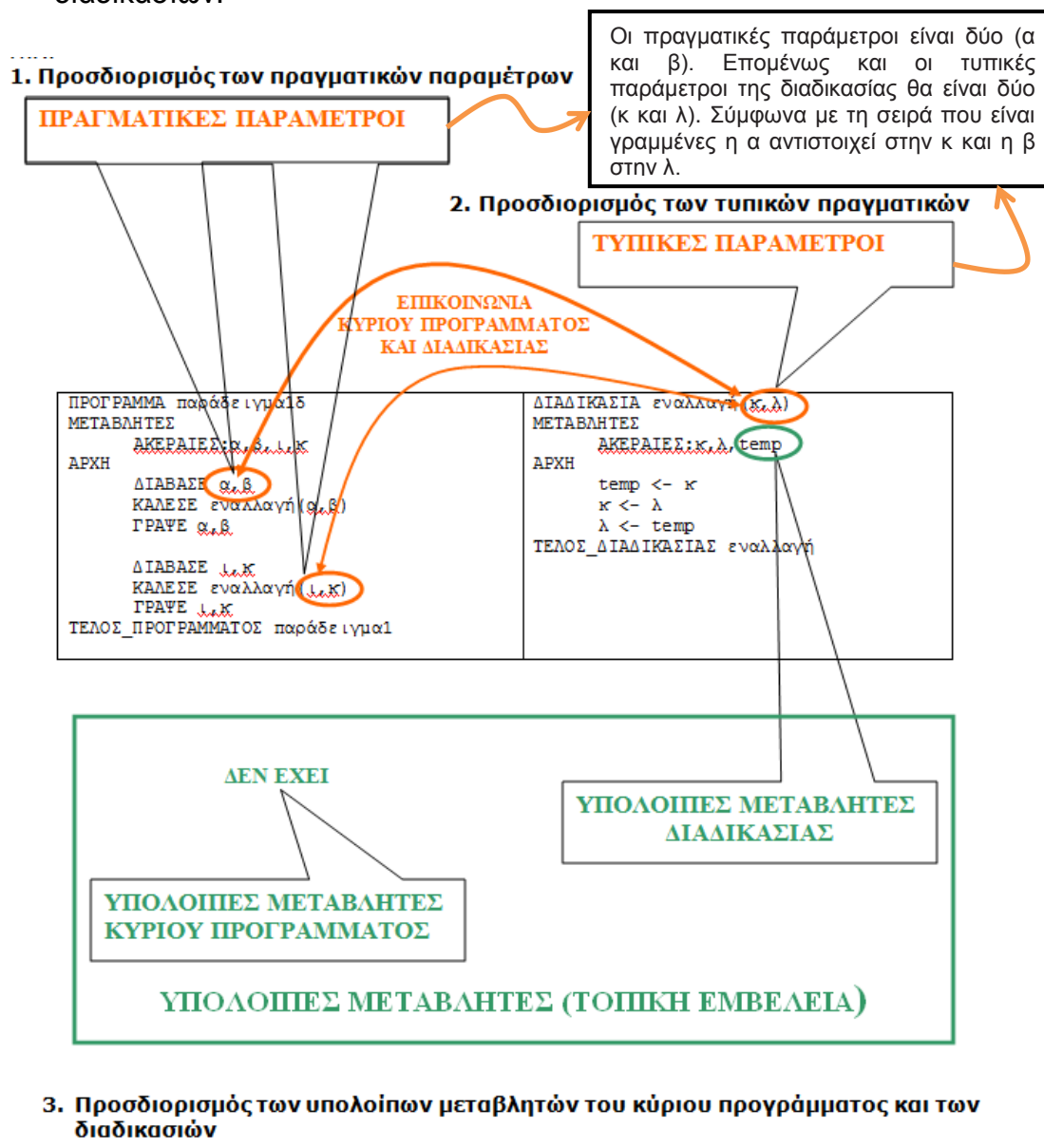
Μια διαδικασία καλείται από το κύριο πρόγραμμα ή άλλη διαδικασία με την εντολή ΚΑΛΕΣΕ, που έχει ως όρισμα το όνομα της καλούμενης διαδικασίας και τις αντίστοιχες πραγματικές παραμέτρους.



### 3.3.3. Πως εκτελείται το πρόγραμμα και πως λειτουργεί η επικοινωνία του κύριου προγράμματος με τη διαδικασία;

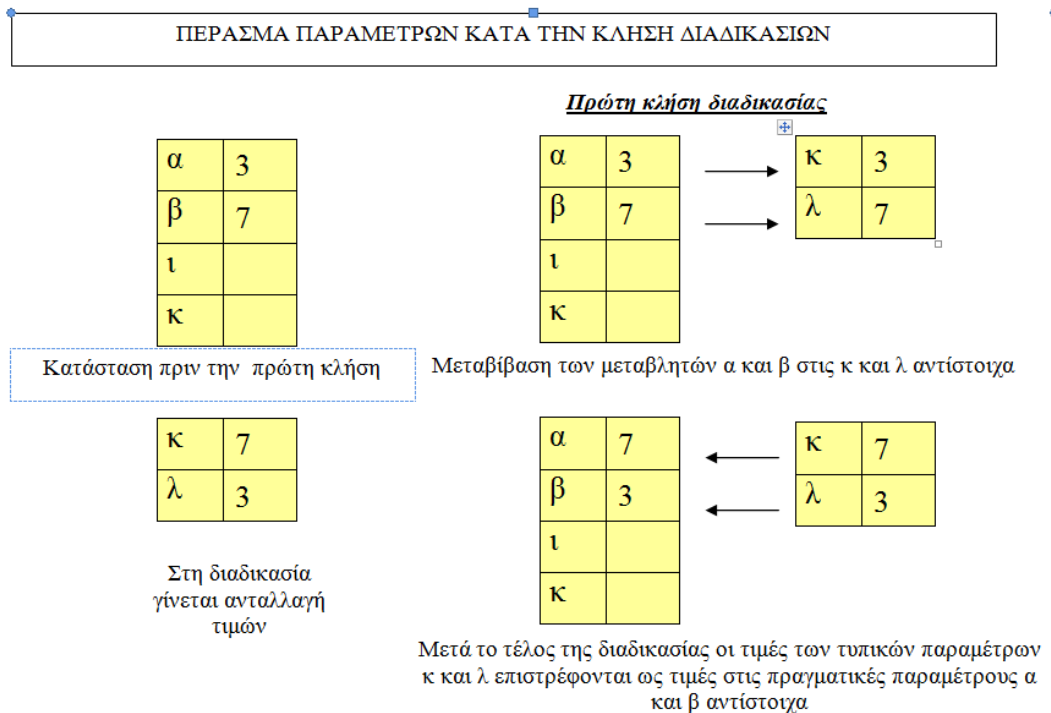
Για την κατανόηση της λειτουργίας θα εκτελεστεί το πρόγραμμα για με δεδομένα εισόδου 3, 7, 2, 9 και θα αναλυθούν η επικοινωνία του κύριου προγράμματος και της διαδικασίας με:

1. Προσδιορισμό των πραγματικών παραμέτρων
2. Προσδιορισμό των τυπικών πραγματικών
3. Προσδιορισμό των υπολοίπων μεταβλητών του κύριου προγράμματος και των διαδικασιών
4. Συμπλήρωση των πινάκων τιμών όλων των μεταβλητών (πραγματικές παράμετροι, τυπικές παράμετροι και υπόλοιπες μεταβλητές) του κύριου προγράμματος και των διαδικασιών με συμπλήρωση των τιμών τους για κάθε εντολή που εκτελείται, του κύριου προγράμματος και των διαδικασιών.



4. Εκτέλεση του προγράμματος και συμπλήρωση των πινάκων τιμών όλων των μεταβλητών (πραγματικές παράμετροι, τυπικές παράμετροι και υπόλοιπες μεταβλητές) του κύριου προγράμματος και των διαδικασιών με συμπλήρωση των τιμών τους για κάθε εντολή που εκτελείται, του κύριου προγράμματος και των διαδικασιών και επηρεάζουν τις τιμές των μεταβλητών.

- Η εκτέλεση του προγράμματος ξεκινά πάντα από το κύριο πρόγραμμα Ανταλλαγή και εκτελούνται οι εντολές του κυρίου προγράμματος (εκτελείται δηλαδή η πρώτη εντολή του κυρίου προγράμματος, η **ΔΙΑΒΑΣΕ**) μέχρι την εντολή **ΚΑΛΕΣΕ Εναλλαγή\_τιμών**. Η εκτέλεση της εντολής αυτής προκαλεί "πάγωμα" του κυρίου προγράμματος. Η συνέχεια της εκτέλεσης του προγράμματος μεταφέρεται στη διαδικασία, **Εναλλαγή\_τιμών** που καλείται. Η τιμές των μεταβλητών και παραμέτρων πριν την πρώτη κλήση της διαδικασίας, φαίνεται στο παρακάτω σχήμα. Κατά την κλήση της διαδικασίας, γίνεται αντιγραφή (μεταβίβαση) των τιμών των πραγματικών παραμέτρων στις αντίστοιχες τυπικές



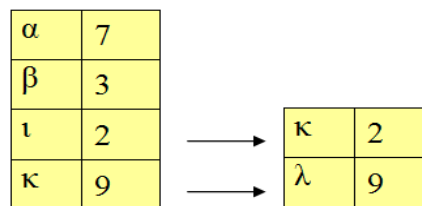
- Οι μεταβλητές του κυρίου προγράμματος, είναι γνωστές μόνο στο πρόγραμμα. Έξω από το πρόγραμμα, κανένα υποπρόγραμμα - διαδικασία δεν τις γνωρίζει και δεν μπορεί να τις χρησιμοποιήσει. Το ίδιο ισχύει και για τις μεταβλητές των υποπρογραμμάτων διαδικασιών. Για αυτό λέμε, ότι οι μεταβλητές έχουν ισχύ τοπική (εντός του προγράμματος-υποπρογράμματος που δηλώνονται). ο Η/Υ διαθέτει διαφορετικό χώρο μνήμης για τις μεταβλητές της διαδικασίας από αυτές του κυρίου προγράμματος)
- Να σημειωθεί ιδιαίτερα ότι μετά το τέλος της διαδικασίας οι τιμές των τυπικών παραμέτρων της διαδικασίας επιστρέφονται ως τιμές στις πραγματικές παραμέτρους του κυρίου προγράμματος (όπως φαίνεται στο

προηγούμενο σχήμα) και ο έλεγχος εκτέλεσης του προγράμματος επανέρχεται στο κύριο πρόγραμμα και συνεχίζει η εκτέλεση από την επόμενη εντολή, από την εντολή που κάλεσε την διαδικασία (σημείο κλήσης της διαδικασίας). Μέχρι εδώ έχουν ανταλλάξει οι τιμές των μεταβλητών α και β. Στη συνέχεια εκτελείται η εντολή ΓΡΑΨΕ. Και τυπώνονται οι τιμές των α, και β. Εκτελείται το δεύτερο ΔΙΑΒΑΣΕ, του κυρίου προγράμματος και φθάνουμε στην εντολή κλήσης της διαδικασίας για δεύτερη φορά, και οι τιμές των μεταβλητών και παραμέτρων διαμορφώνονται όπως ακριβώς δείχνει το παρακάτω σχήμα.

Δεύτερη κλήση διαδικασίας

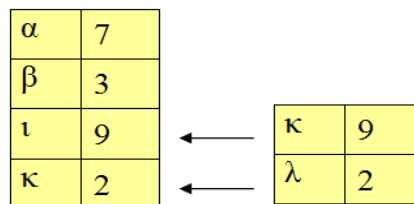
|   |   |
|---|---|
| α | 7 |
| β | 3 |
| ι | 2 |
| κ | 9 |

Κατάσταση πριν τη δεύτερη κλήση



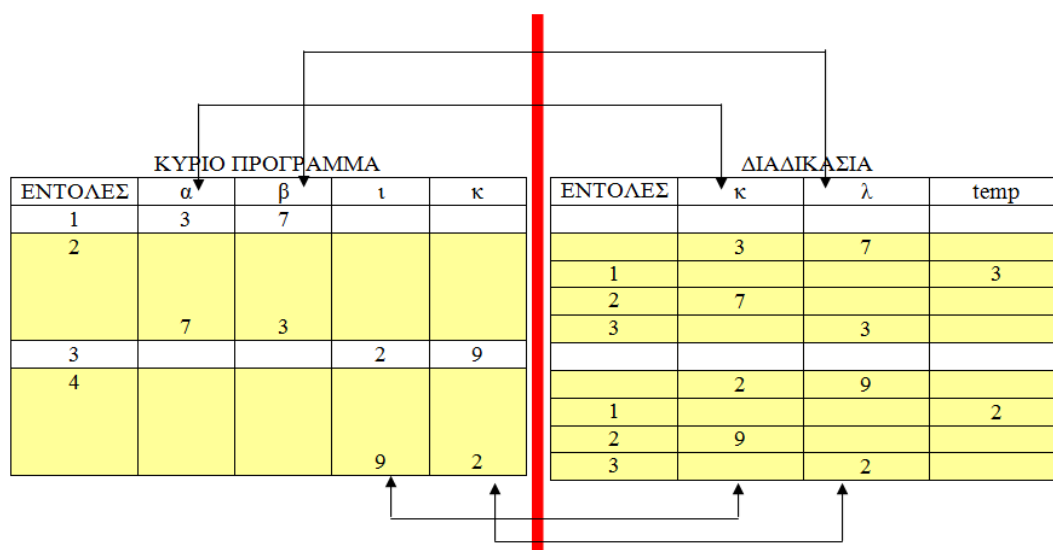
|   |   |
|---|---|
| κ | 9 |
| λ | 2 |

Στη διαδικασία γίνεται ανταλλαγή τιμών

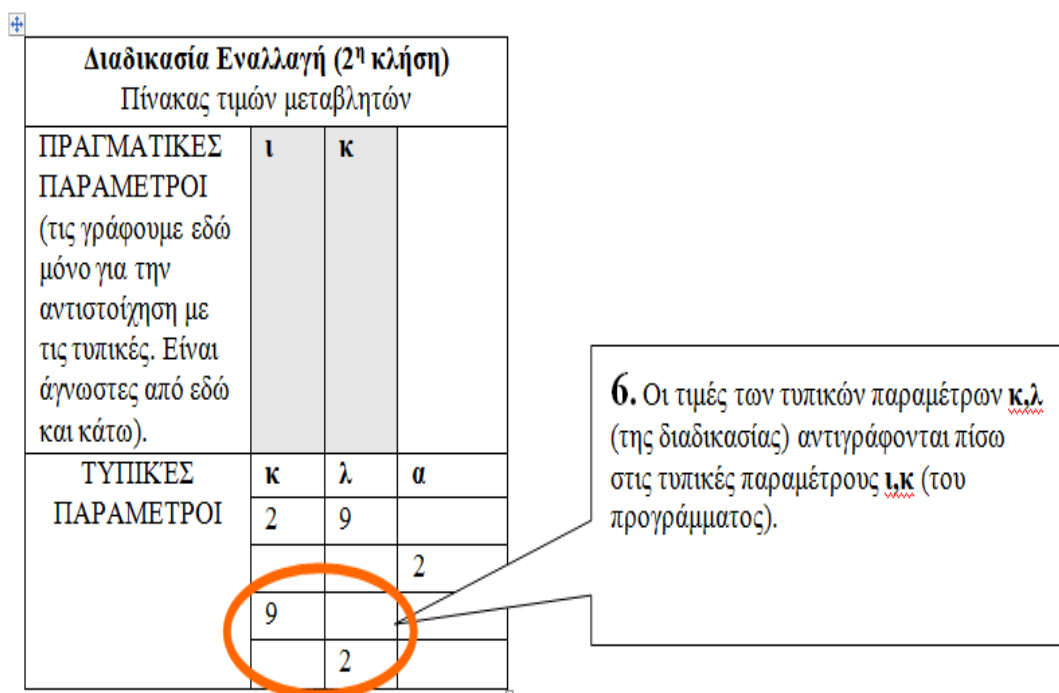
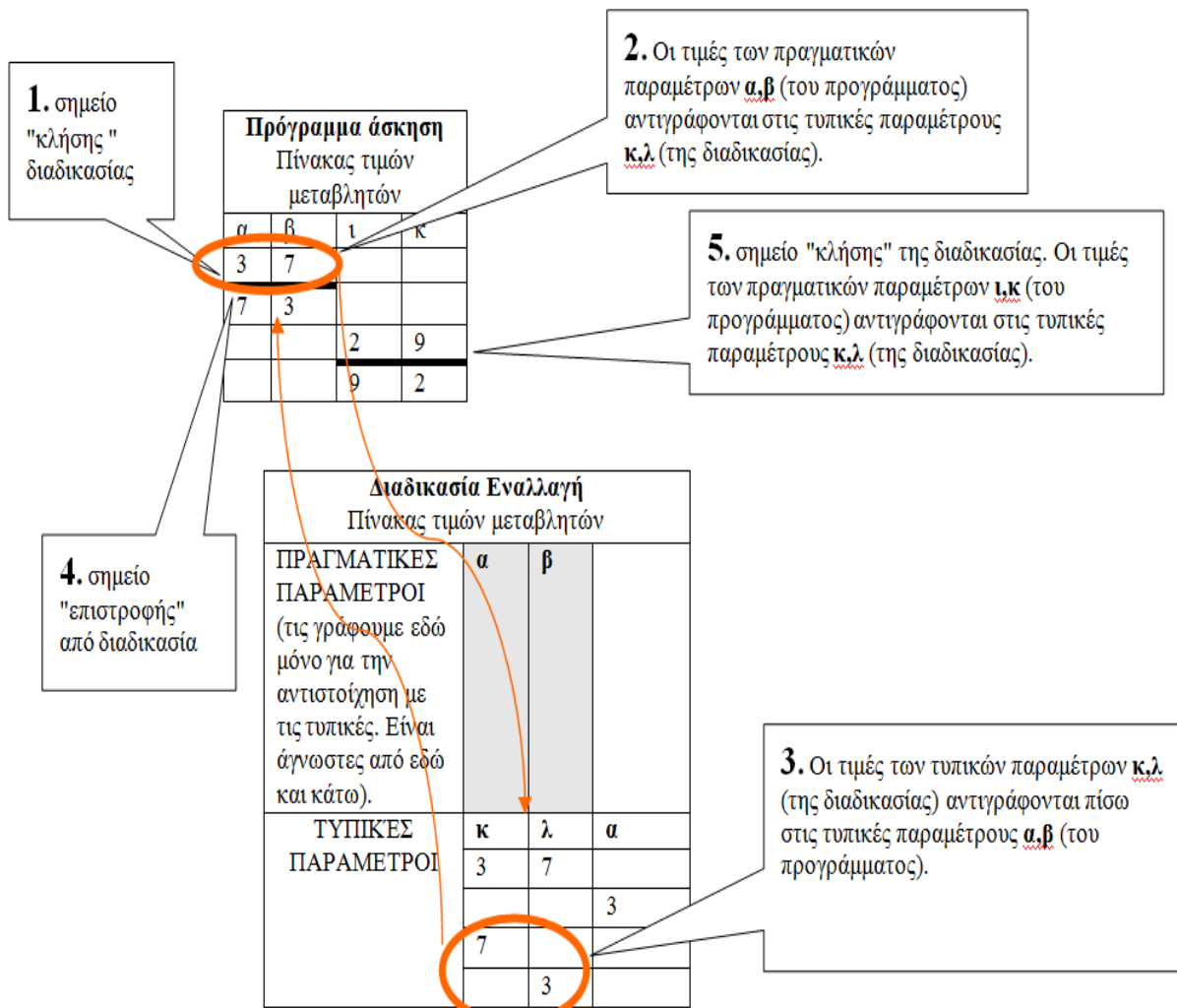


Μετά το τέλος της διαδικασίας οι τιμές των τυπικών παραμέτρων κ και λ επιστρέφονται ως τιμές στις πραγματικές παραμέτρους ι και κ αντίστοιχα

Ο συγκεντρωτικός πίνακας τιμών για το κύριο πρόγραμμα και τις κλήσεις των διαδικασιών φαίνεται στο παρακάτω σχήμα.



Ακολουθώντας τα αριθμημένα βήματα στο παρακάτω σχήμα, γίνεται καλύτερα αντιληπτή η εκτέλεση του προγράμματος.



Στο παρακάτω σχήμα φαίνεται τι θα φανεί στην οθόνη του Η/Υ.

ΜΟΝΑΔΑ ΕΞΟΔΟΥ-ΟΘΟΝΗ

|   |   |
|---|---|
| 7 | 3 |
| 9 | 2 |

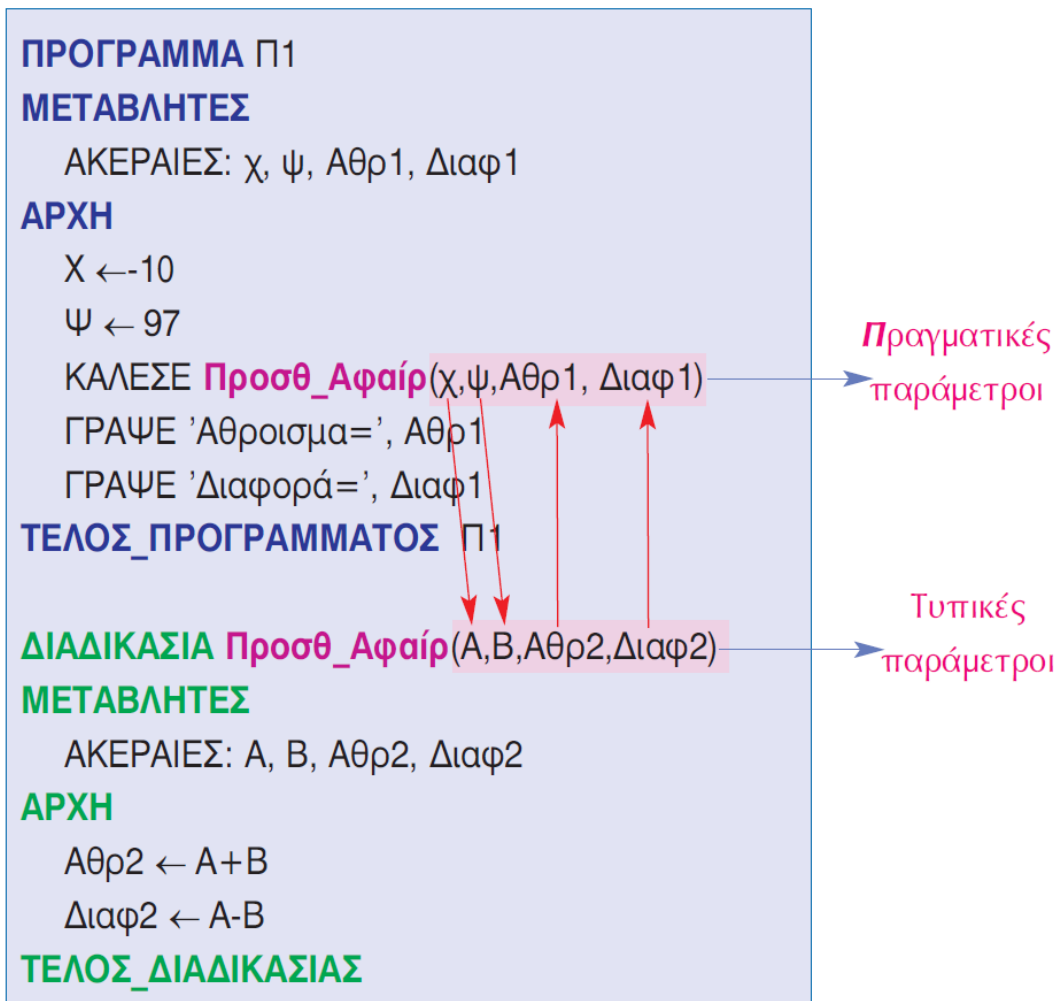
### 3.3.3.1. Σημεία στα οποία πρέπει να δοθεί ιδιαίτερη σημασία.

- **Τοπικότητα μεταβλητών.** Οι μεταβλητές του κυρίου προγράμματος, είναι γνωστές μόνο στο πρόγραμμα. Έξω από το πρόγραμμα, κανένα υποπρόγραμμα - διαδικασία δεν τις γνωρίζει και δεν μπορεί να τις χρησιμοποιήσει. Το ίδιο ισχύει και για τις μεταβλητές των υποπρογραμμάτων διαδικασιών. Για αυτό λέμε, ότι οι μεταβλητές έχουν ισχύ τοπική (εντός του προγράμματος-υποπρογράμματος που δηλώνονται).
- **Επικοινωνία υποπρογραμμάτων.** Ο μοναδικός τρόπος επικοινωνίας είναι μέσω των παραμέτρων (πραγματικών-τυπικών). Οι παράμετροι είναι μεταβλητές του κυρίου προγράμματος και των υποπρογραμμάτων.
- **Αντιστοιχία παραμέτρων-κανόνες τυπικών και πραγματικών παραμέτρων.**
  1. Ο αριθμός των πραγματικών και τυπικών παραμέτρων πρέπει να είναι ο ίδιος.
  2. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση.
  3. Η τυπική παράμετρος και η αντίστοιχη πραγματική παράμετρος πρέπει να είναι του ίδιου τύπου.

Θα μπορούσαμε να το παρομοιάσουμε με ένα βύσμα (π.χ. οθόνης που συνδέεται στην κατάλληλη θύρα του υπολογιστή).

- **Χωροθέτηση.** Πρώτα πάντα είναι γραμμένο το κύριο πρόγραμμα. Έπειτα ακολουθούν τα υποπρογράμματα.
- **Εκτέλεση.** Η εκτέλεση ξεκινά πάντα από το κύριο πρόγραμμα, και προφανώς τελειώνει στο κύριο πρόγραμμα.

Τα σημεία προσοχής που αναφέρονται φαίνονται στο παρακάτω παράδειγμα το οποίο υπολογίζεται και τυπώνεται το άθροισμα και η διαφορά δύο δοθέντων αριθμών με διαδικασία που υπολογίζει το άθροισμα και τη διαφορά τους.



Σύμφωνα με το Πρόγραμμα Π1 του παραδείγματος:

- Οι τιμές των μεταβλητών χ, ψ, Αθρ1 και Διαφ1 δίνονται στις μεταβλητές Α, Β, Αθρ2 και Διαφ2 του υποπρογράμματος Προσθ\_Αφαίρ, κατά την κλήση του.
- Έτσι: η μεταβλητή Α παίρνει την τιμή -10, η μεταβλητή Β παίρνει τη τιμή 97 ενώ οι μεταβλητές Αθρ2 και Διαφ2 δεν παίρνουν καμία τιμή αφού η Αθρ1 και η Διαφ1 δεν έχουν τιμές.
- Μετά την εκτέλεση των εντολών της Διαδικασίας, όταν εκτελεστεί η εντολή ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ, οι τυπικές παράμετροι της διαδικασίας δίνουν τις τιμές που έχουν στις αντίστοιχες πραγματικές παραμέτρους. Έτσι :  
 Η μεταβλητή χ παίρνει τη τιμή της Α(=-10)  
 Η μεταβλητή ψ παίρνει τη τιμή της Β(=97)  
 Η μεταβλητή Αθρ1 παίρνει τη τιμή της Αθρ2(=87)  
 Η μεταβλητή Διαφ1 παίρνει τη τιμή της Διαφ2(=-107)
- Με την επιστροφή στο κύριο πρόγραμμα όλες οι θέσεις μνήμης που είχαν δοθεί στη διαδικασία απελευθερώνονται



### 3.3.4. Κλήση Διαδικασίας από Διαδικασία

Είναι δυνατό μια διαδικασία να καλεί άλλη διαδικασία; Ασφαλώς και μπορεί όπως ακριβώς φαίνεται στο παρακάτω παράδειγμα, όπου μια διαδικασία καλεί μια άλλη διαδικασία. Συγκεκριμένα:

1. Το κύριο πρόγραμμα «παραδειγμα2» εισάγει τιμές από το πληκτρολόγιο, της οποίες μεταβιβάζει στη διαδικασία «υπολογισε» με την εντολή ΚΑΛΕΣΕ.
2. Η διαδικασία «υπολογισε», υπολογίζει την τιμή μιας παράστασης. Αυτή η διαδικασία καλεί τη διαδικασία «κανεΘετικο», που της μεταβιβάζει την τιμή της παράστασης που υπολόγισε.
3. Η διαδικασία «κανεΘετικο» υπολογίζει την απόλυτη τιμή της παράστασης και την επιστρέφει στην διαδικασία «υπολογισε».
4. Η διαδικασία «υπολογισε», με την σειρά της την επιστρέφει στο κύριο πρόγραμμα «παραδειγμα2».
5. Το κύριο πρόγραμμα «παραδειγμα2».εκτυπώνει την απόλυτη τιμή της παράστασης.

|                                                                                                                |                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>ΠΡΟΓΡΑΜΜΑ παραδειγμα2</p> <p>ΜΕΤΑΒΛΗΤΕΣ</p> <p>    ΑΚΕΡΑΙΕΣ: α, β</p> <p>    ΠΡΑΓΜΑΤΙΚΕΣ: τ</p> <p>ΑΡΧΗ</p> |                                                                                                                                                             |
| <p>    ΔΙΑΒΑΣΕ α, β</p>                                                                                        | <p>Το κύριο πρόγραμμα «παραδειγμα2» εισάγει τιμές από το πληκτρολόγιο,</p>                                                                                  |
| <p>    <b>ΚΑΛΕΣΕ υπολογισε(α, β, τ)</b></p>                                                                    | <p>τις οποίες μεταβιβάζει στη διαδικασία «υπολογισε».</p>                                                                                                   |
| <p>    ΓΡΑΨΕ τ</p> <p>ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ παραδειγμα2</p>                                                       | <p>Το κύριο πρόγραμμα «παραδειγμα2», εκτυπώνει την απόλυτη τιμή της παράστασης, που έχει υπολογιστεί από τις άλλες διαδικασίες και του έχει επιστραφεί.</p> |

|                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>ΔΙΑΔΙΚΑΣΙΑ υπολόγισε(β, γ, δ)</p> <p>ΜΕΤΑΒΛΗΤΕΣ</p> <p>ΑΚΕΡΑΙΕΣ: β, γ</p> <p>ΠΡΑΓΜΑΤΙΚΕΣ: δ</p> <p>ΑΡΧΗ</p> <p><math>\delta \leftarrow (\beta + 3 * \gamma) / 2</math></p> <p><b>ΚΑΛΕΣΕ κανεΘετικό(δ)</b></p> <p>ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ υπολόγισε</p> | <p>Η διαδικασία «υπολογισε», υπολογίζει</p> <p>την τιμή μιας παράστασης.</p> <p>Αυτή η διαδικασία καλεί τη διαδικασία «κανεΘετικο», που της μεταβιβάζει την τιμή της παράστασης που υπολόγισε.</p> <p>Μετά την επιστροφή της απόλυτης τιμής της παράστασης, από τη διαδικασία «κανεΘετικο», την επιστρέφει στο κύριο πρόγραμμα.</p> |
| <p>ΔΙΑΔΙΚΑΣΙΑ κανεΘετικό(α)</p> <p>ΜΕΤΑΒΛΗΤΕΣ</p> <p>ΠΡΑΓΜΑΤΙΚΕΣ: α</p> <p>ΑΡΧΗ</p> <p>ΑΝ <math>\alpha &lt; 0</math> ΤΟΤΕ <math>\alpha \leftarrow (-1) * \alpha</math></p> <p>ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ κανεΘετικο</p>                                      | <p>Η διαδικασία «κανεΘετικο» υπολογίζει</p> <p>την απόλυτη τιμή της παράστασης και την επιστρέφει στην διαδικασία «υπολογισε».</p>                                                                                                                                                                                                  |

Ο πίνακας τιμών του παραδείγματος φαίνεται στο παρακάτω σχήμα.

| Κύριο πρόγραμμα |   |   |      | Διαδικασία 'υπολόγισε' |   |    |       | Διαδικασία 'κονεθετικό' |       | ΟΘΟΝΗ |
|-----------------|---|---|------|------------------------|---|----|-------|-------------------------|-------|-------|
| Εντολές         | α | β | τ    | Εντολές                | β | γ  | δ     | εντολές                 | α     |       |
| 1               | 6 | 9 |      |                        |   |    |       |                         |       | 6     |
| 2               |   |   |      |                        | 6 | -9 |       |                         |       | -9    |
|                 |   |   |      | 1                      |   |    | -10.5 |                         |       | -10.5 |
|                 |   |   |      | 2                      |   |    |       |                         | -10.5 |       |
|                 |   |   |      |                        |   |    |       | 1                       | 10.5  |       |
|                 |   |   |      |                        |   |    | 10.5  | τέλος                   |       |       |
|                 |   |   | 10.5 | τέλος                  |   |    |       |                         |       |       |
| 3               |   |   |      |                        |   |    |       |                         |       |       |

### 3.3.5. Διαδικασίες και Πίνακες

Στο παρακάτω παράδειγμα φαίνεται το γέμισμα και η εκτύπωση δύο πινάκων με κλήση των αντίστοιχων διαδικασιών ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(χ) και ΕΚΤΥΠΩΣΗ\_ΠΙΝΑΚΑ(χ)

**ΠΡΟΓΡΑΜΜΑ gemisma\_pinaka**  
 ΜΕΤΑΒΛΗΤΕΣ  
 ΑΚΕΡΑΙΕΣ: x[100], y[100]  
 ΑΡΧΗ  
 ΓΡΑΨΕ 'Γέμισμα πρώτου πίνακα'  
 ΚΑΛΕΣΕ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(x)  
 ΓΡΑΨΕ 'Γέμισμα δεύτερου πίνακα'  
 ΚΑΛΕΣΕ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(y)  
 ΓΡΑΨΕ 'Εκτύπωση πρώτου πίνακα'  
 ΚΑΛΕΣΕ ΕΚΤΥΠΩΣΗ\_ΠΙΝΑΚΑ(x)  
 ΓΡΑΨΕ 'Εκτύπωση δεύτερου πίνακα'  
 ΚΑΛΕΣΕ ΕΚΤΥΠΩΣΗ\_ΠΙΝΑΚΑ(y)  
 ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Στη λίστα των πραγματικών παραμέτρων εντολής «ΚΑΛΕΣΕ», γράφεται μόνο το όνομα του πίνακα χωρίς το πλήθος των θέσεων που παρακρατούνται στη μνήμη.

**ΔΙΑΔΙΚΑΣΙΑ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(A)**  
 ΜΕΤΑΒΛΗΤΕΣ  
 ΑΚΕΡΑΙΕΣ: A[100], i  
 ΑΡΧΗ  
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100  
 ΓΡΑΨΕ 'A[, i, ]='  
 ΔΙΑΒΑΣΕ A[i]  
 ΤΕΛΟΣ ΕΠΙΣΤΡΟΦΗΣ  
 ΤΕΛΟΣ ΔΙΑΔΙΚΑΣΙΑΣ

Στη λίστα των τυπικών παραμέτρων στο όνομα της διαδικασίας, γράφεται μόνο το όνομα του πίνακα χωρίς το πλήθος των θέσεων που παρακρατούνται στη μνήμη.

**ΔΙΑΔΙΚΑΣΙΑ ΕΚΤΥΠΩΣΗ\_ΠΙΝΑΚΑ(A)**  
 ΜΕΤΑΒΛΗΤΕΣ  
 ΑΚΕΡΑΙΕΣ: A[100], i  
 ΑΡΧΗ  
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100  
 ΓΡΑΨΕ 'A[, i, ]=', A[i]  
 ΤΕΛΟΣ ΕΠΙΣΤΡΟΦΗΣ  
 ΤΕΛΟΣ ΔΙΑΔΙΚΑΣΙΑΣ

Στο τμήμα δηλώσεων των μεταβλητών, τα ονόματα των ονομάτων των πινάκων δηλώνονται κανονικά

**Προσοχή:** στη σύνταξη της εντολής «ΚΑΛΕΣΕ», σε σχέση με τα ονόματα των παραμέτρων (πραγματική και τυπική), όταν αφορά πίνακα. Στη λίστα των παραμέτρων γράφεται μόνο το όνομα του πίνακα χωρίς το πλήθος των θέσεων που παρακρατούνται στη μνήμη. Π.χ.

ΚΑΛΕΣΕ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(χ) και **όχι ΚΑΛΕΣΕ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(χ[100])**

Ο ίδιος κανόνας χρησιμοποιείται και στην τυπική παράμετρο των διαδικασιών

Π.χ. ΔΙΑΔΙΚΑΣΙΑ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(χ) και

**όχι ΔΙΑΔΙΚΑΣΙΑ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(χ[100]).**

Στο τμήμα δηλώσεων των μεταβλητών, τα ονόματα των ονομάτων των πινάκων δηλώνονται κανονικά Π.χ. **ΑΚΕΡΑΙΕΣ:Α[100]**

**Στο παρακάτω παράδειγμα ζητείται να γραφεί διαδικασία που να επιστρέφει τον μικρότερο αριθμό και τη θέση του από ένα πίνακα χ[100] με ακέραιους αριθμούς.**

### Ανάλυση

- Η διαδικασία θα έχει τρεις παραμέτρους, τον πίνακα χ σαν δεδομένο, τον Min που θα επιστρέφει τον μικρότερο αριθμό και τον Min\_θέση που θα επιστρέφει τη θέση της μικρότερης τιμής.
- Θα αναπτυχθεί ο αλγόριθμος του μικρότερου που θα κρατά και την αντίστοιχη θέση του και τις τιμές αυτές θα τις καταχωρεί στις αντίστοιχες παραμέτρους για να μεταφέρουν τα αποτελέσματα στο υποπρόγραμμα που καλεί τη διαδικασία.

### Λύση

**ΔΙΑΔΙΚΑΣΙΑ Μικρότερο** (χ,Min,Min\_θέση)

#### ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: χ[100], i,Min,Min\_θέση

#### ΑΡΧΗ

Min ← χ[1]

Min\_θέση ← 1

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100

    ΑΝ χ[i]<Min ΤΟΤΕ

        Min ← χ[i]

        Min\_θέση ← i

    ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

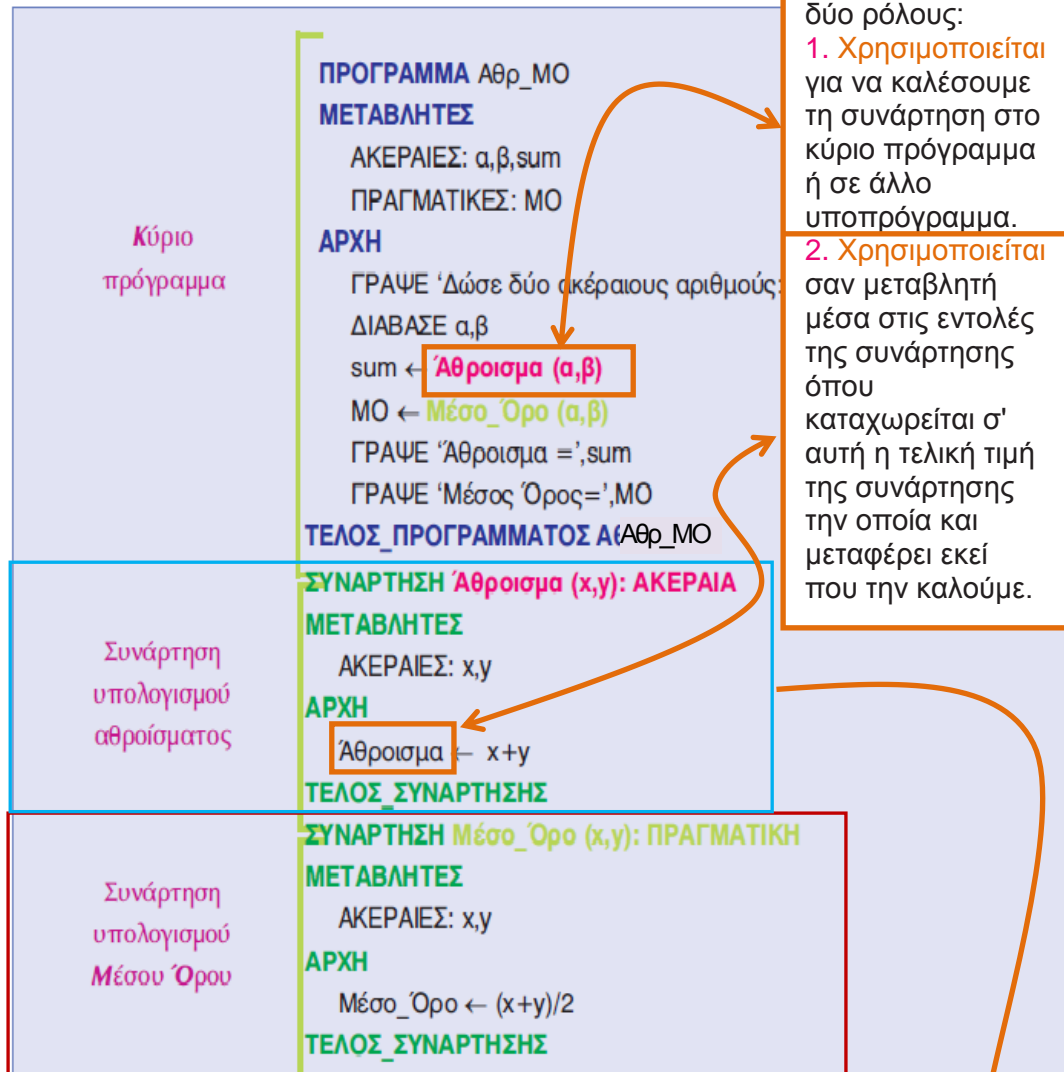
**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

### 3.4. Συναρτήσεις (Παράδειγμα συνάρτησης)

Να γραφτεί πρόγραμμα που να διαβάζει δύο ακέραιους αριθμούς και να τυπώνει το άθροισμα τους και το μέσο όρο τους.

Ο υπολογισμός του αθροίσματος και του μέσου όρου γίνεται από δύο συναρτήσεις.

**Λύση**



Το <όνομα> της συνάρτησης έχει δύο ρόλους:

1. Χρησιμοποιείται για να καλέσουμε τη συνάρτηση στο κύριο πρόγραμμα ή σε άλλο υποπρόγραμμα.

2. Χρησιμοποιείται σαν μεταβλητή μέσα στις εντολές της συνάρτησης όπου καταχωρείται σ' αυτή η τελική τιμή της συνάρτησης την οποία και μεταφέρει εκεί που την καλούμε.

**Σύνταξη συνάρτησης:**

```
ΣΥΝΑΡΤΗΣΗ <όνομα> (<λίστα παραμέτρων>): <τύπος συνάρτησης>
 ΜΕΤΑΒΛΗΤΕΣ
 <δήλωση μεταβλητών>
 ΑΡΧΗ
 <εντολές>
 <όνομα> ← τιμή
 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

### 3.4.1. Πως συντάσσεται και πως καλείται μια συνάρτηση;

#### 3.4.1.1. Σύνταξη - ορισμός συνάρτησης

**ΣΥΝΑΡΤΗΣΗ <όνομα>** (<λίστα παραμέτρων>): <τύπος συνάρτησης>  
**ΜΕΤΑΒΛΗΤΕΣ**  
<δήλωση μεταβλητών>  
**ΑΡΧΗ**  
<εντολές>  
<όνομα> ← τιμή  
**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

Ο τύπος της συνάρτησης, που δηλώνεται στον ορισμό της ( <τύπος της συνάρτησης>) μετά το όνομα της συνάρτησης, εκφράζει τον τύπο της τιμής που μεταφέρει η συνάρτηση.

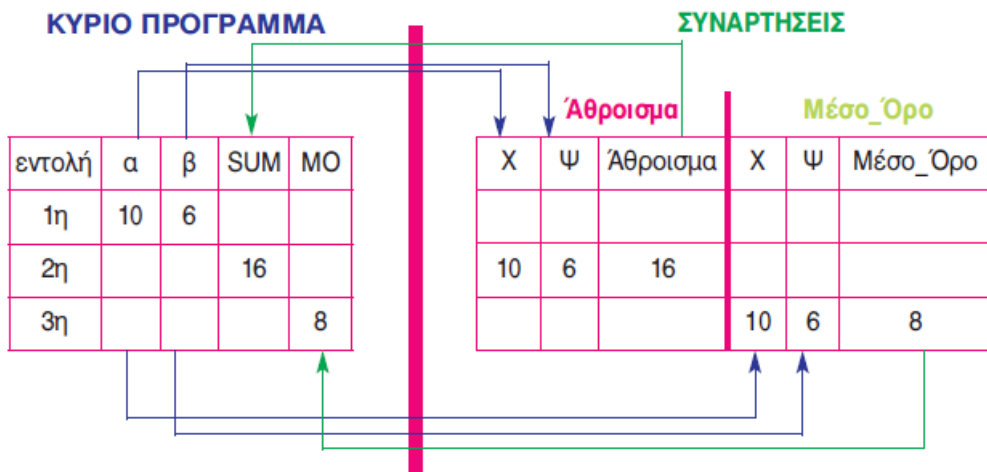
Αν δηλαδή η τιμή που παίρνει είναι ακέραια, πραγματική, λογική, ανεξάρτητα από τον τύπο των δικών της ορισμάτων. Π. χ. η συνάρτηση «Άθροισμα» οι τιμές που παίρνει είναι ακέραιες (η συνάρτησης πρέπει να είναι ακεραίου τύπου). Οι παράμετροί της (χ,ψ) είναι ακέραιες. Αν οι παράμετροί της ήταν πραγματικού τύπου και η συνάρτηση υπολόγιζε το ακεραίο μέρος του αθροίσματός τους, πάλι η συνάρτηση, αφού παίρνει ακέραιες τιμές θα πρέπει να είναι ακεραίου τύπου και όχι πραγματικού τύπου που είναι οι παράμετροί της.

#### 3.4.1.2. Κλήση συνάρτησης

- Η συνάρτηση καλείται από το κύριο πρόγραμμα με το όνομά της, μέσα σε μια εντολή. Π.χ ως τιμή μιας μεταβλητής σε μια εντολή εκχώρησης:  $sum \leftarrow \text{Άθροισμα}(α,β)$
- Το <όνομα> της συνάρτησης έχει δύο ρόλους:
  - Χρησιμοποιείται για να καλέσουμε τη συνάρτηση στο κύριο πρόγραμμα ή σε άλλο υποπρόγραμμα.
  - Χρησιμοποιείται σαν μεταβλητή μέσα στις εντολές της συνάρτησης όπου καταχωρείται σ' αυτή η τελική τιμή της συνάρτησης την οποία και μεταφέρει εκεί που την καλούμε.
- Οι παράμετροι στη συνάρτηση δέχονται τιμές και δεν τις μεταφέρουν από τη συνάρτηση σε άλλο πρόγραμμα ή υποπρόγραμμα.
- Οι τιμές που επιστρέφουν οι συναρτήσεις μπορεί να είναι ΠΡΑΓΜΑΤΙΚΕΣ, ΑΚΕΡΑΙΕΣ, ΧΑΡΑΚΤΗΡΕΣ, ΛΟΓΙΚΕΣ.

#### Επεξήγηση της λύσης του παραπάνω παραδείγματος:

- Ο πίνακας τιμών κατά την εκτέλεση του προγράμματος είναι:



Παρατηρούμε, ότι:

- Το κάθε υποπρόγραμμα, εδώ η κάθε συνάρτηση, δουλεύει αυτόνομα στη μνήμη στη δική του περιοχή μνήμης και έχει επαφή με το πρόγραμμα μόνο μέσω των παραμέτρων.
- Όταν τελειώνει η εκτέλεση των εντολών της ΣΥΝΑΡΤΗΣΗΣ τότε όλες οι μεταβλητές της χάνονται και ελευθερώνεται η αντίστοιχη περιοχή μνήμης.
- Οι μεταβλητές  $a$  και  $b$  δίνουν τη τιμή τους στις μεταβλητές  $x$  και  $y$  αντίστοιχα όταν καλείται η συνάρτηση **άθροισμα**, ενώ δίνουν τη τιμή τους αντίστοιχα στις  $x$  και  $y$  όταν καλείται η συνάρτηση **Μέσος Όρος**.
- Η μεταβλητή **Αθροισμα** δίνει τη τιμή της στη μεταβλητή **sum** του κύριου προγράμματος όταν τελειώσει η συνάρτηση **Αθροισμα** και ομοίως η μεταβλητή **Μέσο\_Όρο** στην **MO** όταν τελειώσει η συνάρτηση **Μέσο\_Όρο**.

### 3.4.2. Σε τι διαφέρουν οι διαδικασίες από τις συναρτήσεις;

| Διαδικασίες                                                                                                                                                                             | Συναρτήσεις                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Οι διαδικασίες μπορούν να εκτελέσουν μια οποιαδήποτε λειτουργία, π.χ. να εισάγουν δεδομένα, να εκτελούν υπολογισμούς, να τυπώνουν αποτελέσματα και να αλλάζουν τις τιμές των μεταβλητών | Οι συναρτήσεις υπολογίζουν μόνο μια τιμή και μόνο αυτή επιστρέφουν στο κύριο πρόγραμμα ή στο υποπρόγραμμα που τις κάλεσε. Η τιμή αυτή μπορεί να είναι ΑΡΙΘΜΗΤΙΚΗ, ΧΑΡΑΚΤΗΡΑΣ ή ΛΟΓΙΚΗ.  |
| Οι διαδικασίες μεταφέρουν τα αποτελέσματα στα άλλα υποπρογράμματα με τη χρήση παραμέτρων                                                                                                | Οι συναρτήσεις μεταφέρουν το αποτέλεσμα στο κύριο πρόγραμμα ή στο υποπρόγραμμα που τις κάλεσε με το όνομά τους και όχι με τη χρήση παραμέτρων. Μοιάζουν με τις μαθηματικές συναρτήσεις. |
| Οι διαδικασίες εκτελούνται αν γράψουμε την εντολή ΚΑΛΕΣΕ και μετά το όνομα της διαδικασίας.                                                                                             | Οι συναρτήσεις εκτελούνται με τη χρήση του ονόματος τους μέσα σε οποιαδήποτε εντολή.                                                                                                    |
| Μια συνάρτηση μπορεί να αντικατασταθεί με αντίστοιχη διαδικασία                                                                                                                         |                                                                                                                                                                                         |

### 3.4.3. Διαδικασίες και Συναρτήσεις Μαζί

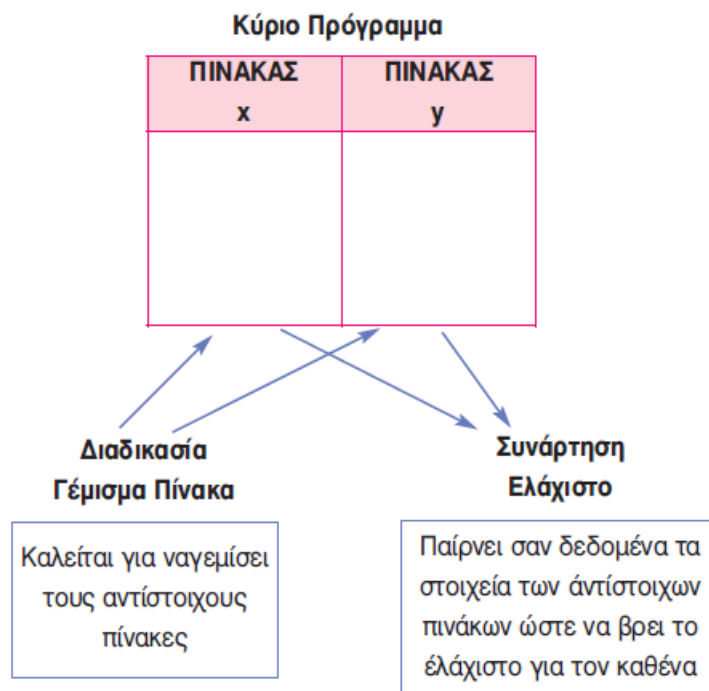
Ένα πρόγραμμα είναι δυνατόν να περιλαμβάνει και διαδικασίες και συναρτήσεις. Π.Χ.

Να γραφτεί πρόγραμμα που να διαβάζει δύο πίνακες  $x[100]$  και  $y[100]$  με ακέραιους αριθμούς και στη συνέχεια να τυπώνει τον ελάχιστο για κάθε πίνακα.

Το γέμισμα των πινάκων να γίνει με χρήση της διαδικασίας ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ και η εύρεση του μικρότερου με χρήση της συνάρτησης ΕΛΑΧΙΣΤΟ.

#### Ανάλυση προβλήματος

Αυτό που ζητείται σχηματικά αναπαρίσταται με το παρακάτω σχήμα.



1. Πρέπει να γεμίσουν τους δύο πίνακες με κάλεσμα της διαδικασίας ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ (x) και ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ (y). Καλείται η ίδια διαδικασία για να γεμίσουν δύο πίνακες. Αυτό γίνεται με αλλαγή της παραμέτρου, όπου την κάθε φορά παίρνει το όνομα του πίνακα που αφορά. Αυτό είναι και το βασικό πλεονέκτημα των υποπρογραμμάτων, ότι μπορούν να χρησιμοποιηθούν όσες φορές χρειάζεται, χρησιμοποιώντας κάθε φορά τις κατάλληλες παραμέτρους.
2. Επίσης θα πρέπει να κληθεί δυο φορές η συνάρτηση ΕΛΑΧΙΣΤΟ (x) και ΕΛΑΧΙΣΤΟ (y), με τη λογική που αναφέρθηκε παραπάνω. Επειδή η συνάρτηση καλείται μέσα από εντολή ή θα χρησιμοποιηθεί η εντολή ΓΡΑΨΕ οπότε και τυπώνεται απευθείας ή θα χρησιμοποιηθεί η εντολή καταχώρησης για να αποθηκευθεί τη τιμή της σε μια μεταβλητή και στη συνέχεια να τυπωθεί η μεταβλητή.
3. Μετά το τέλος του προγράμματος γράφονται τα υποπρογράμματα. Η Διαδικασία ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ και η Συνάρτηση ΕΛΑΧΙΣΤΟ.



Ο κώδικας του προγράμματος είναι:

```
ΠΡΟΓΡΑΜΜΑ Ελάχιστα_Πινάκων
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: x[100], y[100]
ΑΡΧΗ
 ΓΡΑΨΕ 'Καταχώρηση δεδομένων 1ου πίνακα'
 ΚΑΛΕΣΕ Γέμισμα_Πίνακα (x)
 ΓΡΑΨΕ 'Καταχώρηση δεδομένων 2ου πίνακα'
 ΚΑΛΕΣΕ Γέμισμα_Πίνακα (y)
 ΓΡΑΨΕ 'Ελάχιστος αριθμός του 1ου πίνακα:', Ελάχιστο (x)
 ΓΡΑΨΕ 'Ελάχιστος αριθμός του 2ου πίνακα:', Ελάχιστο (y)
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ελάχιστα_Πινάκων
ΔΙΑΔΙΚΑΣΙΑ Γέμισμα_Πίνακα (A)
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: A[100], i
ΑΡΧΗ
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
 ΓΡΑΨΕ 'Δώσε ', i, ' στοιχείο'
 ΔΙΑΒΑΣΕ A[i]
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
ΣΥΝΑΡΤΗΣΗ Ελάχιστο (A): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: Min, i, A[100]
ΑΡΧΗ
 Min ← A[1]
 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
 ΑΝ A[i] < Min ΤΟΤΕ
 Min ← A[i]
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
 Ελάχιστο ← Min
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```