

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

ΠΑΡΑΡΤΗΜΑ Α
ΟΔΗΓΙΕΣ ΜΕΛΕΤΗΣ ΜΑΘΗΤΗ
(2η Έκδοση)

Γ' ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ
Ομάδας Προσανατολισμού Σπουδών Οικονομίας & Πληροφορικής

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Γ΄ Τάξη ΓΕ.Λ.

**ΠΑΡΑΡΤΗΜΑ Α
ΟΔΗΓΙΕΣ ΜΕΛΕΤΗΣ ΜΑΘΗΤΗ
(2η Έκδοση)**

2η Έκδοση

**ΑΝΑΜΟΡΦΩΣΗ,
ΠΡΟΣΑΡΜΟΓΗ
ΚΑΙ ΕΠΙΜΕΛΕΙΑ
ΠΕΡΙΕΧΟΜΕΝΟΥ**

Γραμμένος Νικόλαος, Σύμβουλος Β΄ Ι.Ε.Π.
Γούσιου Ανθή, Σύμβουλος Β΄ Ι.Ε.Π.

Το παρόν εκπονήθηκε αμισθί, με ευθύνη της Μονάδας Φυσικών Επιστημών,
Τεχνολογίας και Μαθηματικών του Ινστιτούτου Εκπαιδευτικής Πολιτικής,
με βάση την υπ' αριθμ. 15/22-04-2019 πράξη του ΔΣ του Ι.Ε.Π.

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Γεράσιμος Κουζέλης

Πρόεδρος του Ινστιτούτου Εκπαιδευτικής Πολιτικής

Γραφείο Επιστημονικών Μονάδων Β΄ Κύκλου
Προϊστάμενη: Γεωργία Φέρμελη, Σύμβουλος Α΄ Ι.Ε.Π.



1η Έκδοση

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

ΑΘΗΝΑ, ΙΟΥΝΙΟΣ 2019 (2^η ΕΚΔΟΣΗ)

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

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Γ΄ Τάξη ΓΕ.Λ.

Ομάδας Προσανατολισμού Σπουδών
Οικονομίας & Πληροφορικής

ΠΑΡΑΡΤΗΜΑ Α
ΟΔΗΓΙΕΣ ΜΕΛΕΤΗΣ ΜΑΘΗΤΗ
(2η Έκδοση)

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ
«ΔΙΟΦΑΝΤΟΣ»

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΠΡΟΓΡΑΜΜΑ – «ΓΛΩΣΣΑ», ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ & ΕΝΤΟΛΕΣ	7
1.1 Δομή προγράμματος σε «ΓΛΩΣΣΑ»	7
1.2 Το αλφάβητο της «ΓΛΩΣΣΑΣ»	7
1.3 Τύποι δεδομένων	7
1.4 Σταθερές - Μεταβλητές	8
1.5 Αριθμητικές Εκφράσεις- Συγκριτικές – Λογικές Πράξεις	11
1.6 Βασικές εντολές και απλά προγράμματα στη «ΓΛΩΣΣΑ»	15
1.7 Μη λυμένες Ασκήσεις	19
2. ΔΟΜΗ ΕΠΙΛΟΓΗΣ	21
2.1 Απλή Επιλογή	21
2.2 Σύνθετη Επιλογή	22
2.3 Πολλαπλή Επιλογή	23
2.4 Γενικά Παραδείγματα	26
2.5 Μη λυμένες ασκήσεις	34
3. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ	38
3.1 Η εντολή επανάληψης «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ»	38
3.2 Η εντολή επανάληψης «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ...»	41
3.3 Η εντολή επανάληψης «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...»	46
3.4 Εμφωλευμένοι Βρόχοι	47
3.5 Παραδείγματα εύρεσης ελάχιστου ή μέγιστου	47
3.6 Μετατροπές από μία δομή επανάληψης σε άλλη	53
3.7 Μη λυμένες ασκήσεις	56
4. ΠΙΝΑΚΕΣ	61
4.1 Δήλωση πίνακα στη «ΓΛΩΣΣΑ»	61
4.2 Είσοδος δεδομένων σε μονοδιάστατο και σε διδιάστατο πίνακα	62
4.3 Τυπικές επεξεργασίες πινάκων	70
4.3.1 Ταξινόμηση στοιχείων πινάκα	70
4.3.2 Αναζήτηση στοιχείου πίνακα	73
4.3.3 Συγκώνευση δύο πινάκων	77
4.4 Παράλληλοι πίνακες	79
4.5 Λυμένες ασκήσεις	83
4.6 Μη λυμένες ασκήσεις	92

5. ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ - ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ	
(ΔΙΑΔΙΚΑΣΙΕΣ-ΣΥΝΑΡΤΗΣΕΙΣ)	97
5.1 Διαδικασίες (Παράδειγμα διαδικασίας)	100
5.2 Συναρτήσεις (Παράδειγμα συνάρτησης).....	114
5.3 Διαδικασίες και Συναρτήσεις	117
5.4 Μη λυμένες ασκήσεις	119

1. ΠΡΟΓΡΑΜΜΑ – «ΓΛΩΣΣΑ», ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ & ΕΝΤΟΛΕΣ

1.1 Δομή προγράμματος σε «ΓΛΩΣΣΑ»

Η δομή του προγράμματος στη «ΓΛΩΣΣΑ» είναι η παρακάτω:

```
ΠΡΟΓΡΑΜΜΑ <όνομα προγράμματος>
ΣΤΑΘΕΡΕΣ
<δήλωση σταθερών>
ΜΕΤΑΒΛΗΤΕΣ
<δήλωση μεταβλητών>
ΑΡΧΗ
    ....
    .... <εντολές προγράμματος>
    ....
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ <όνομα προγράμματος>
```



Παρατηρήσεις:

- Το όνομα του προγράμματος πρέπει να συμφωνεί με τους κανόνες δημιουργίας ονομάτων της «ΓΛΩΣΣΑΣ».
- Οι δεσμευμένες, από τη «ΓΛΩΣΣΑ», λέξεις ΣΤΑΘΕΡΕΣ (για δήλωση σταθερών) ή/και ΜΕΤΑΒΛΗΤΕΣ (για δήλωση μεταβλητών) ενός προγράμματος χρησιμοποιούνται στην περίπτωση που στο πρόγραμμα υπάρχουν σταθερές ή/και μεταβλητές αντίστοιχα, αλλιώς παραλείπονται.
- Οι δεσμευμένες λέξεις ΑΡΧΗ και ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ δηλώνουν αντίστοιχα την αρχή και το τέλος εκτέλεσης του προγράμματος. Μέσα σ' αυτές τοποθετείται το κύριο μέρος του προγράμματος και περιλαμβάνει τις εκτελέσιμες εντολές.
- Κάθε εντολή γράφεται σε ξεχωριστή γραμμή.
- Αν μια εντολή χρειάζεται να συνεχιστεί στην επόμενη γραμμή, τότε ο πρώτος χαρακτήρας αυτής της γραμμής πρέπει να είναι ο χαρακτήρας «&».
- Αν ο πρώτος χαρακτήρας μιας εντολής είναι το θαυμαστικό «!», τότε η γραμμή περιέχει σχόλια και όχι εκτελέσιμες εντολές. Δηλαδή, τη γραμμή αυτή την αγνοεί ο υπολογιστής κατά την εκτέλεση του προγράμματος. Τα σχόλια, μας βοηθούν να γράφουμε επεξηγήσεις σε διάφορα σημεία του προγράμματος.

* Από τη δομή του προγράμματος είναι εμφανές ότι για τη δημιουργία οποιουδήποτε προγράμματος σε «ΓΛΩΣΣΑ», είναι απαραίτητο να γνωρίζουμε τα βασικά στοιχεία της «ΓΛΩΣΣΑΣ» (αλφάβητο, τύπους δεδομένων, σταθερές, μεταβλητές και εντολές).

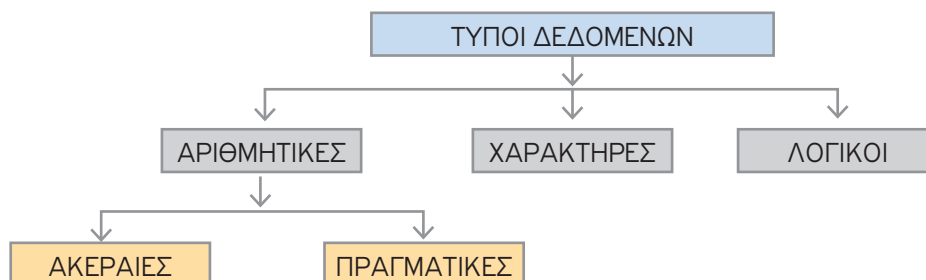
1.2 Το αλφάβητο της «ΓΛΩΣΣΑΣ»

Το αλφάβητο της «ΓΛΩΣΣΑΣ» αποτελείται από:

- **ΓΡΑΜΜΑΤΑ** (Κεφαλαία ελληνικού αλφαβήτου, Πεζά ελληνικού αλφαβήτου, Κεφαλαία λατινικού αλφαβήτου, Πεζά λατινικού αλφαβήτου)
- **ΨΗΦΙΑ** (0,1,2,3,4,5,6,7,8,9)
- **ΕΙΔΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ** (+, -, *, /, =, (,), !, &, το κενό)

1.3 Τύποι δεδομένων

Η «ΓΛΩΣΣΑ» έχει τους εξής τέσσερις βασικούς τύπους δεδομένων, με τους οποίους καθορίζουμε τον τύπο της μεταβλητής ή σταθεράς που θα χρησιμοποιηθεί στο πρόγραμμα.



ΑΚΕΡΑΙΕΣ: Ο τύπος αυτός περιλαμβάνει όλους τους ακέραιους αριθμούς. Μπορεί να είναι θετικοί, αρνητικοί ή μηδέν.

ΠΡΑΓΜΑΤΙΚΕΣ: Ο τύπος αυτός περιλαμβάνει όλους τους πραγματικούς αριθμούς. Μπορεί να είναι θετικοί, αρνητικοί ή μηδέν.

ΧΑΡΑΚΤΗΡΕΣ: Ο τύπος αυτός αναφέρεται τόσο σε ένα χαρακτήρα όσο και σε μια σειρά από χαρακτήρες. Οι χαρακτήρες πρέπει να βρίσκονται υποχρεωτικά μέσα σε εισαγωγικά. Επίσης, σαν χαρακτήρες μπορούμε να χρησιμοποιούμε και αριθμούς, γι' αυτό τα στοιχεία αυτού του τύπου λέγονται και αλφαριθμητικά στοιχεία. Η διαφορά ενός αριθμού που είναι δηλωμένος σαν «Ακέραιος» ή «Πραγματικός», από τον αριθμό που είναι δηλωμένος σαν «Χαρακτήρας», είναι ότι ο πρώτος μπορεί να συμμετάσχει σε αριθμητικές πράξεις, ενώ ο δεύτερος δεν μπορεί.

Παραδείγματα χαρακτήρων: 'Κ', 'Μαρία', 'Αύριο θα πάμε σινεμά', '123', κλπ.

ΛΟΓΙΚΕΣ: Αυτός ο τύπος δέχεται μόνο δύο τιμές: ΑΛΗΘΗΣ και ΨΕΥΔΗΣ.

1.4 Σταθερές - Μεταβλητές

Σταθερές

Οι σταθερές είναι προκαθορισμένες τιμές που δε μεταβάλλονται κατά τη διάρκεια εκτέλεσης του προγράμματος. Οι σταθερές, ανάλογα με την τιμή που «παίρνουν», διακρίνονται σε:

- ΑΚΕΡΑΙΕΣ π.χ. -3, 205, +1097
- ΠΡΑΓΜΑΤΙΚΕΣ π.χ. 0.18, -2.397
- ΧΑΡΑΚΤΗΡΕΣ π.χ. 'Μαρία', 'Πώληση προϊόντων'
- ΛΟΓΙΚΕΣ που είναι μόνο δύο οι τιμές: ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ.

Τρόπος σύνταξης των σταθερών σ' ένα πρόγραμμα:

ΣΤΑΘΕΡΕΣ

Όνομα_1 = τιμή1

Όνομα_2 = τιμή2

...

Όνομα_ν = τιμήν

Παράδειγμα

ΣΤΑΘΕΡΕΣ

ΦΠΑ = 0.19

π = 3.14

Όνομα = 'Μαρία'

Τα ονόματα των σταθερών «ΦΠΑ», «π» ή «Όνομα» μπορούν να χρησιμοποιηθούν μέσα στο πρόγραμμα και να αντιπροσωπεύουν τις τιμές που έχουμε ορίσει. Οι τιμές αυτές δεν μπορούν να αλλάξουν κατά τη διάρκεια εκτέλεσης του προγράμματος. Η χρήση ονομάτων σταθερών κάνει το πρόγραμμα πιο κατανοητό και κατά συνέπεια ευκολότερο να διορθωθεί και να συντηρηθεί.

Μεταβλητές

Μια μεταβλητή παριστάνει μια ποσότητα που η τιμή της μπορεί να μεταβάλλεται.

- Σ' ένα πρόγραμμα χρησιμοποιούμε διάφορες τιμές, τις οποίες επεξεργαζόμαστε. Αυτές οι τιμές πρέπει να αποθηκεύονται στη μνήμη του υπολογιστή. Άρα, κάθε τιμή δεσμεύει μια περιοχή μνήμης. Για να γνωρίζουμε σε ποια περιοχή μνήμης είναι η τιμή που θέλουμε να χρησιμοποιήσουμε, δίνουμε όνομα σε κάθε περιοχή. Το όνομα αυτό είναι το όνομα της μεταβλητής.
- Συνεπώς, η μεταβλητή εκφράζει μια περιοχή μνήμης και όταν δώσουμε τιμή στη μεταβλητή, η τιμή αυτή αποθηκεύεται στην περιοχή μνήμης με το αντίστοιχο όνομα.
- Σε μια μεταβλητή, κατά τη διάρκεια εκτέλεσης του προγράμματος, μπορούμε να αλλάξουμε όσες φορές θέλουμε την τιμή της. Όμως, κάθε φορά χάνεται η προηγούμενη τιμή και αντικαθίσταται με τη νέα τιμή της.
- Η τιμή που μπορεί να πάρει μια μεταβλητή είναι ακέραια, πραγματική, χαρακτήρας ή λογική.
- Οι μεταβλητές, ανάλογα με την τιμή που «παίρνουν», διακρίνονται σε:
 - ΑΚΕΡΑΙΕΣ π.χ. – 3, 205, +1097
 - ΠΡΑΓΜΑΤΙΚΕΣ π.χ. 0.18, – 2.397
 - ΧΑΡΑΚΤΗΡΕΣ π.χ. 'Μαρία', 'Πώληση προϊόντων'
 - ΛΟΓΙΚΕΣ που είναι μόνο δύο οι τιμές: ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ

Τρόπος σύνταξης των μεταβλητών σ' ένα πρόγραμμα:

ΜΕΤΑΒΛΗΤΕΣ

```
τύπος1:Λίστα_μεταβλητών  
τύπος2:Λίστα_μεταβλητών  
...  
τύποςν: Λίστα_μεταβλητών
```

Παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

```
ΠΡΑΓΜΑΤΙΚΕΣ: X, Ψ, M_O  
ΑΚΕΡΑΙΕΣ: N, Τιμή  
ΧΑΡΑΚΤΗΡΕΣ: Όνομα, Επίθετο  
ΛΟΓΙΚΕΣ: Έλεγχος
```

Τα ονόματα «X», «Ψ», «M_O», «N», «Τιμή», «Όνομα», «Επίθετο» και «Έλεγχος» εκφράζουν τα ονόματα των μεταβλητών, που θα χρησιμοποιήσουμε στο πρόγραμμά μας όπως επίσης και ο τύπος που εκφράζει κάθε μια απ' αυτές.

Π.χ. η μεταβλητή X δέχεται πραγματικές τιμές.

Κανόνες ονοματολογίας Μεταβλητών - Σταθερών

Το όνομα μιας μεταβλητής ή σταθεράς μπορεί να αποτελείται:

- από γράμματα, πεζά ή κεφαλαία, του ελληνικού ή λατινικού αλφαβήτου
- από τα ψηφία 0, 1, 2, ..., 9
- από σύμβολα μόνο την κάτω παύλα _

Απαγορεύονται τα ακόλουθα:

- Ο πρώτος χαρακτήρας του ονόματος να είναι αριθμός
- Η ύπαρξη σημείων στίξεως ή ειδικών συμβόλων, όπως: +, -, /, [,], &, # κ.λπ., εκτός από το σύμβολο της κάτω παύλας (_)
- Η χρήση δεσμευμένων λέξεων της «ΓΛΩΣΣΑΣ», όπως: οι λέξεις ΑΚΕΡΑΙΕΣ, ΧΑΡΑΚΤΗΡΕΣ κ.λπ.

Παραδείγματα

1. Ποιο από τα παρακάτω ονόματα μεταβλητών ή σταθερών, είναι σωστό ή λάθος και γιατί;

1) α1	2) B_1	3) B_	4) 1_Φ	5) x23
6) 23x	7) 256	8) x+ψ	9) x/ψ	10) x!1
11) α&	12) ΠΡΟΓΡΑΜΜΑ	13) Άννα	14) Τιμή_Αγοράς	15) 'ΜΑΡΙΑ'
16) ΑΛΗΘΗΣ	17) Ψευδής1			

Λύση

- 1) α1: Σωστό
2) B_1: Σωστό
3) B_: Σωστό
4) 1_Φ: Λάθος, διότι ο 1ος χαρακτήρας είναι αριθμός
5) x23: Σωστό
6) 23x: Λάθος, διότι ο 1ος χαρακτήρας είναι αριθμός
7) 256: Λάθος, διότι ο 1ος χαρακτήρας είναι αριθμός
8) x+ψ: Λάθος, διότι υπάρχει το σύμβολο '+'
9) x/ψ: Λάθος, διότι υπάρχει το σύμβολο '/'
10) x!1: Λάθος, διότι υπάρχει το σύμβολο '!'
11) α&: Λάθος, διότι υπάρχει το σύμβολο '&'
12) ΠΡΟΓΡΑΜΜΑ: Λάθος, διότι η λέξη ΠΡΟΓΡΑΜΜΑ είναι δεσμευμένη λέξη
13) Άννα: Σωστό
14) Τιμή_Αγοράς: Σωστό
15) 'ΜΑΡΙΑ': Λάθος, διότι υπάρχουν τα εισαγωγικά
16) ΑΛΗΘΗΣ: Λάθος, διότι η λέξη ΑΛΗΘΗΣ είναι δεσμευμένη λέξη.
17) Ψευδής1: Σωστό.

2. Στον παρακάτω πίνακα στην 1η στήλη δίνεται ο τύπος της μεταβλητής και στη 2η στήλη η τιμή μιας μεταβλητής. Να αντιστοιχιστούν οι τύποι των μεταβλητών της πρώτης στήλης, στις τιμές των μεταβλητών της δεύτερης στήλης.

Τύπος μεταβλητής	Τιμή μεταβλητής
1. ΑΚΕΡΑΙΕΣ	A. Αληθής
2. ΠΡΑΓΜΑΤΙΚΕΣ	B. 2345
3. ΛΟΓΙΚΕΣ	Γ. 56.9
4. ΧΑΡΑΚΤΗΡΕΣ	Δ. 'ΑΛΗΘΗΣ'
	Ε. Ψευδής
	Στ. '90.89'
	Z. -30,89
	H. Άνω

Λύση

1. → B (διότι σε κάθε ακέραιο τύπο αντιστοιχούν σαν τιμές οι ακέραιοι αριθμοί)

2. $\rightarrow \Gamma, \mathbf{Z}$ (διότι σε κάθε πραγματικό τύπο αντιστοιχίζονται σαν τιμές οι πραγματικοί αριθμοί, δηλ. αριθμοί με δεκαδικό μέρος)

3. $\rightarrow$ A, E (διότι σε κάθε λογικό τύπο αντιστοιχούν οι τιμές Αληθής ή Ψευδής, μόνο).

4. → Δ, Στ, Η (διότι σε κάθε αλφαριθμητικό τύπο αντιστοιχίζονται οι τιμές που βρίσκονται μέσα σε εισαγωγικά).

3. Δίνονται οι μεταβλητές με τις αντίστοιχες τιμές τους

$a \leftarrow 10.89$	$\delta \leftarrow \text{'Αληθής'}$	$\zeta \leftarrow \text{'Νίκος'}$
$\beta \leftarrow \text{'Πρόγραμμα'}$	$x \leftarrow 45$	$\kappa \leftarrow -56$
$\gamma \leftarrow 275$	$y \leftarrow \Psiευδής$	$\lambda \leftarrow -206.9$
		$\varepsilon \leftarrow \text{'-4'}$

Να γραφεί το τμήμα του προγράμματος που ορίζουμε τις μεταβλητές, δηλώνοντας τις παραπάνω στους αντίστοιχους τύπους τους.

Λύση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: γ, x, k

ΠΡΑΓΜΑΤΙΚΕΣ: α. λ

ΧΑΡΑΚΤΗΡΕΣ: δ, ζ, ε, β

ΛΟΓΙΚΕΣ: γ

1.5 Αριθμητικές Εκφράσεις- Συγκριτικές – Λογικές Πράξεις

Τελεστές

Οι τελεστές είναι σύμβολα που χρησιμοποιούνται στις διάφορες πράξεις. Οι τελεστές διακρίνονται σε **αριθμητικούς**, **συγκριτικούς** και **λογικούς**.

Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές καλύπτουν τις βασικές πράξεις: πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση, ενώ υποστηρίζεται η ύψωση σε δύναμη, η ακέραια διαίρεση και το υπόλοιπο της ακέραιας διαίρεσης.

Οι τελεστές με τις αντίστοιχες πράξεις είναι:

Αριθμητικός τελεστής	Πράξη	Παραδείγματα
+	Πρόσθεση	3+5 και επιστρέφει 8
-	Αφαίρεση	5-2 και επιστρέφει 3
*	Πολλαπλασιασμός	2*4 και επιστρέφει 8
/	Διαίρεση	5/2 και επιστρέφει 2.5
^	Ύψωση σε δύναμη	2^3 και επιστρέφει 8
DIV	Ακέραια διαίρεση	5 div 2 και επιστρέφει 2
MOD	Υπόλοιπο ακέραιας διαίρεσης	5 mod 2 και επιστρέφει 1

Συναρτήσεις

Πολλές γνωστές συναρτήσεις από τα μαθηματικά χρησιμοποιούνται συχνά και περιέχονται στη «ΓΛΩΣΣΑ» και στον αλγόριθμο. Οι συναρτήσεις αυτές είναι:

Συνάρτηση	Περιγραφή	Παράδειγμα
HM(x)	Υπολογισμός ημίτονου	HM(45)
ΣΥΝ(x)	Υπολογισμός συνημίτονου	ΣΥΝ(45)
ΕΦ(x)	Υπολογισμός εφαπτομένης	ΕΦ(45)
T_P(x)	Υπολογισμός τετραγωνικής ρίζας	T_P(9) αντιστοιχεί στο $\sqrt{9}$
ΛΟΓ(x)	Υπολογισμός φυσικού λογαρίθμου	ΛΟΓ(32)
E(x)	Υπολογισμός του e^x	E(a+3) αντιστοιχεί στο e^{a+3}
A_M(x)	Ακέραιο μέρος του x	A_M(3.597) δίνει το 3
A_T(x)	Απόλυτη τιμή του x	A_T(-3+1) δίνει το 2

- Για τη σύνταξη μιας αριθμητικής έκφρασης χρησιμοποιούνται αριθμητικοί τελεστές, μεταβλητές, σταθερές, συναρτήσεις και παρενθέσεις.
- Οι αριθμητικές εκφράσεις υλοποιούν απλές ή σύνθετες μαθηματικές πράξεις.
- Κάθε αριθμητική έκφραση μας δίνει σαν αποτέλεσμα μια αριθμητική τιμή, που βρίσκεται μετά την εκτέλεση πράξεων.

- Οι πράξεις σε μια αριθμητική έκφραση εκτελούνται με την παρακάτω ιεραρχία:
 1. Ύψωση σε δύναμη (^)
 2. Πολλαπλασιασμός (*) και διαίρεση (/ , DIV , MOD)
 3. Πρόσθεση (+) και αφαίρεση (-)
- Σε περίπτωση που υπάρχει **παρένθεση** στην αριθμητική έκφραση, τότε εκτελούνται πρώτα οι πράξεις στη παρένθεση σύμφωνα με την παραπάνω ιεραρχία.
- Όταν έχουμε πράξεις της **ίδιας ιεραρχίας**, π.χ. πολλαπλασιασμό και διαίρεση, τότε οι πράξεις εκτελούνται διαδοχικά από τα αριστερά προς τα δεξιά.

Παραδείγματα

Να γράφουν οι παρακάτω αριθμητικές εκφράσεις σε «ΓΛΩΣΣΑ», λαμβάνοντας υπόψη τη σειρά προτεραιότητας εκτέλεσης των πράξεων:

- 1) $a+10$ 2) $1/2a^3$ 3) $\frac{3x+2\psi}{a-\beta}$ 4) $2\eta\mu\psi$ 5) $2+3x+\frac{3}{\psi}$
6) $2\frac{2x+3}{2*\psi}$ 7) $\frac{9}{8+x}+\sqrt{\beta+8}$ 8) $\frac{x+\psi}{\sqrt{7+x}}$ 9) $|a|+\frac{x+9}{10}a$ 10) $\sqrt{a^2+b^2}$

Λύση

- 1) $a+10$
2) $1/2*a^3$
3) $(3*x+2*\psi)/(a-\beta)$
4) $2*HM(\psi)$
5) $2+3*x+3/\psi$
6) $2*(2*x+3)/(2*\psi)$
7) $9/(8+x)+T_P(\beta+8)$
8) $(x+\psi)/T_P(7+x)$
9) $A_T(a)+(x+9)/10*a$
10) $T_P(a^2+\beta^2)$

Συγκριτικοί τελεστές

Οι συγκριτικοί τελεστές είναι:

Τελεστές	Πράξεις	Παραδείγματα
=	Ισότητα	$x = 0$
<>	Διάφορο	ON <> 'Κώστας'
>	Μεγαλύτερο	Τιμή > 10000
>=	Μεγαλύτερο ή ίσο	$A + B \geq 5 / (x+3)$
<	Μικρότερο	$B^2 - 4*A*\Gamma < 0$
<=	Μικρότερο ή ίσο	Βάρος <= 1000

Οι συγκρίσεις σε μια μαθηματική έκφραση βοηθούν στη λήψη αποφάσεων, για παράδειγμα αν η ηλικία σου είναι μεγαλύτερη των 18 ετών τότε είσαι ενήλικος, διαφορετικά δεν είσαι ενήλικος.

Λογικοί τελεστές

Σε πολλά προβλήματα οι επιλογές δεν αρκεί να γίνονται με απλές λογικές παραστάσεις, δηλαδή με μια απλή σύγκριση, αλλά χρειάζεται να συνδυαστούν μια ή περισσότερες λογικές εκφράσεις. Οι λογικές εκφράσεις αποτελούνται από συγκριτικές πράξεις. Αυτό επιτυγχάνεται με τη χρήση τριών βασικών λογικών τελεστών: **ΟΧΙ**, **ΚΑΙ** και **Ή**. Οι λογικές παραστάσεις αποτελούνται από λογικές μεταβλητές, οι οποίες «παίρνουν» μια λογική τιμή, **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**. Το αποτέλεσμα που επιστρέφει μια λογική έκφραση είναι μια λογική τιμή **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**, ανάλογα με τις τιμές των μεταβλητών της.

Στον παρακάτω πίνακα δίνονται τις τιμές των τριών αυτών λογικών πράξεων για όλους τους συνδυασμούς τιμών των μεταβλητών τους A, B:

Μεταβλητή A	Μεταβλητή B	A Ή B	A ΚΑΙ B	ΟΧΙ A
Αληθής	Αληθής	Αληθής	Αληθής	Ψευδής
Αληθής	Ψευδής	Αληθής	Ψευδής	Ψευδής
Ψευδής	Αληθής	Αληθής	Ψευδής	Αληθής
Ψευδής	Ψευδής	Ψευδής	Ψευδής	Αληθής

Οι πράξεις σε μια λογική έκφραση εκτελούνται με την παρακάτω **ιεραρχία**:

1. **ΟΧΙ**
2. **ΚΑΙ**
3. **Ή**

Σε περίπτωση που υπάρχει **παρένθεση** στη λογική έκφραση, τότε εκτελούνται πρώτα οι πράξεις στην παρένθεση σύμφωνα με την παραπάνω ιεραρχία.

Στη περίπτωση που σε μια έκφραση συνυπάρχουν περισσότερα από ένα είδη πράξεων, τότε η **ιεραρχία των πράξεων** είναι η εξής:

1. **Αριθμητικοί τελεστές**
2. **Συγκριτικοί τελεστές και**
3. **Λογικοί τελεστές.**

Παραδείγματα

1. Να βρεθούν οι τιμές που δίνουν οι παρακάτω μαθηματικές εκφράσεις:

Λύση

$$\alpha) (20 \div 2) * 2^2 / 2 =$$

$$10 * 2^2 / 2 =$$

$$10 * 4 / 2 =$$

$$40 / 2 =$$

$$20$$

$$\beta) (T_P(49) + 1) \bmod 3 =$$

$$(7 + 1) \bmod 3 =$$

$$8 \bmod 3 =$$

$$2$$

$$\gamma) A_M(7/2) + 4^2 =$$

$$A_M(3,5) + 4^2 =$$

$$3 + 4^2 =$$

$$3 + 16 =$$

$$19$$

$$\delta) 20 / 2 / 4 =$$

$$10 / 4 =$$

$$2,5$$

$$\epsilon) 12 \bmod 2 + 10 / 4 =$$

$$0 + 2,5 =$$

$$2,5$$

$$\sigma\tau) 12 \div 2 + 6 \bmod 2^2 =$$

$$12 \div 2 + 6 \bmod 4 =$$

$$6 + 2 =$$

$$8$$

2. Να βρεθεί ποια από τις παρακάτω εκφράσεις είναι Αληθής ή Ψευδής, αν η μεταβλητή K έχει την τιμή 12, η μεταβλητή Λ την τιμή 6 και η μεταβλητή Μ την τιμή 2.
(Όπου Α= Αληθής και Ψ= Ψευδής)

α) $K > \Lambda$ ΚΑΙ $(K < M \text{ Ή } M \leq \Lambda) =$ Αληθής

$$12 > 6 \text{ ΚΑΙ } (12 < 2 \text{ Ή } 2 \leq 6) =$$

$$A \text{ ΚΑΙ } (\Psi \text{ Ή } A) =$$

$$A \text{ ΚΑΙ } A = A$$

γ) $K = \Lambda + M + 4$ ΚΑΙ $(K < \Lambda \text{ Ή } K + M > \Lambda) =$ Αληθής

$$12 = 6 + 2 + 4 \text{ ΚΑΙ } (12 < 6 \text{ Ή } 12 + 2 > 6) =$$

$$12 = 12 \text{ ΚΑΙ } (12 < 6 \text{ Ή } 14 > 6) =$$

$$A \text{ ΚΑΙ } (\Psi \text{ Ή } A) =$$

$$A \text{ ΚΑΙ } A = A$$

β) $K > \Lambda$ ΚΑΙ $K < M$ ΚΑΙ $M < \Lambda =$ Ψευδής

$$12 > 6 \text{ ΚΑΙ } 12 < 2 \text{ ΚΑΙ } 2 < 6 =$$

$$A \text{ ΚΑΙ } \Psi \text{ ΚΑΙ } A =$$

$$\Psi \text{ ΚΑΙ } A = \Psi$$

δ) $(K > \Lambda \text{ ΚΑΙ } M < \Lambda) \text{ Ή } (\Lambda < M \text{ ΚΑΙ } K < M) =$ Αληθής

$$(12 > 6 \text{ ΚΑΙ } 2 < 6) \text{ Ή } (6 < 2 \text{ ΚΑΙ } 12 < 2) =$$

$$(A \text{ ΚΑΙ } A) \text{ Ή } (A \text{ ΚΑΙ } \Psi) =$$

$$A \text{ Ή } \Psi = A$$

1.6 Βασικές εντολές και απλά προγράμματα στη «ΓΛΩΣΣΑ»

Βασικές εντολές

Εντολή εκχώρησης

Σύνταξη: Όνομα μεταβλητής $\leftarrow$ <Εκφραση>

όπου <Εκφραση> = κάποια τιμή (ακέραια ή πραγματική ή χαρακτήρας ή λογική) ή κάποια άλλη μεταβλητή ή κάποια μαθηματική παράσταση.

Τρόπος εκτέλεσης: Με την εντολή αυτή καταχωρίζεται στη μνήμη (στη μεταβλητή που έχουμε ορίσει στο αριστερό μέλος της σχέσης) η τιμή που δίνουμε ή που προκύπτει από πράξεις στο δεξί μέλος της σχέσης.



Παρατηρήσεις:

- Σε μια εντολή εκχώρησης η μεταβλητή που βρίσκεται στο αριστερό μέρος αυτής δύναται να υπάρχει και στο δεξί μέρος της.
Π.χ.
1) $a \leftarrow a + 1$ Αύξηση της μεταβλητής a κατά $+1$. Δηλαδή η νέα τιμή του a είναι η παλιά συν ένα (**μετρητής**)
2) $a \leftarrow a + x$ Αύξηση της μεταβλητής a κατά x . Δηλαδή η νέα τιμή του a είναι η παλιά συν τη τιμή της μεταβλητής x (**αθροιστής**)
- Δεν επιτρέπεται στο αριστερό μέρος της εντολής εκχώρησης να υπάρχει πράξη ή μια οποιαδήποτε παράσταση ή τιμή. Το μόνο που επιτρέπεται είναι το όνομα_μεταβλητής.
- Η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου. Δηλαδή, αν στο αριστερό μέρος έχουμε μεταβλητή που «παίρνει» ακέραιες τιμές, τότε το αποτέλεσμα της έκφρασης στο δεξί μέρος της πρέπει να είναι ακέραια τιμή.

Παραδείγματα

1. Ποιες από τις παρακάτω εντολές είναι σωστές, ποιες είναι λάθος και γιατί;

- | | |
|---|--|
| 1. $ab \leftarrow x1$ | 11. $a_ \leftarrow a1$ |
| 2. $a+b \leftarrow x1$ | 12. $B \leftarrow \text{'ΓΡΑΨΕ'}$ |
| 3. $a1 \leftarrow a+b$ | 13. $x \leftarrow \text{'Όνομα'}$ |
| 4. $a \leftarrow 3+2*6^2$ | 14. $z \leftarrow A>B$ |
| 5. $2\psi \leftarrow \psi$ | 15. $z1 \leftarrow \text{'A>B'}$ |
| 6. $ΑΛΗΘΗΣ \leftarrow a$ | 16. $A \leftarrow 3*(5+6^2)/2$ |
| 7. $\text{Επώνυμο} \leftarrow \text{ΝΙΚΟΣ}$ | 17. $\text{Προϊόν} \leftarrow \text{Όνομα_Προϊόντος}$ |
| 8. $\text{Επώνυμο} \leftarrow \text{'ΝΙΚΟΣ'}$ | 18. $\text{Πρ} \leftarrow \text{'Η/Υ'}$ |
| 9. $a*3 \leftarrow 2*a*\beta$ | 19. $\text{Σημαία} \leftarrow 5>2*5$ |
| 10. $\text{done} \leftarrow \PsiΕΥΔΗΣ$ | 20. $\text{Νίκος} \leftarrow \text{'ΜΑΡΙΑ'}$ |

Λύση

1. Σωστή, στη μεταβλητή ab καταχωρίζεται η τιμή της μεταβλητής $x1$.
2. Λάθος, διότι το όνομα της μεταβλητής $a+b$ δεν είναι αποδεκτό, εφόσον υπάρχει το σύμβολο $+$.
3. Σωστή, στη μεταβλητή $a1$ καταχωρίζεται το αποτέλεσμα της πράξης $a+b$.
4. Σωστή, στη μεταβλητή a καταχωρίζεται το αποτέλεσμα της πράξης $3+2*6^2$.
5. Λάθος, διότι στο όνομα της μεταβλητής 2ψ ο πρώτος χαρακτήρας είναι αριθμός.
6. Λάθος, διότι το όνομα της μεταβλητής $ΑΛΗΘΗΣ$ δεν είναι αποδεκτό, εφόσον είναι δεσμευμένη λέξη.
7. Σωστό, στη μεταβλητή Επώνυμο καταχωρίζεται η τιμή της μεταβλητής ΝΙΚΟΣ .
8. Σωστό, στη μεταβλητή Επώνυμο καταχωρίζεται η τιμή 'ΝΙΚΟΣ' .
9. Λάθος, διότι το όνομα της μεταβλητής $a*3$ δεν είναι αποδεκτό εφόσον υπάρχει το σύμβολο $*$.
10. Σωστό, στη μεταβλητή done καταχωρίζεται η τιμή $\Psi\epsilon\upsilon\delta\eta\varsigma$.
11. Σωστό, στη μεταβλητή $a_$ καταχωρίζεται η τιμή της μεταβλητής $a1$.
12. Σωστό, στη μεταβλητή B καταχωρίζεται η τιμή 'ΓΡΑΨΕ' .
13. Σωστό, στη μεταβλητή x καταχωρίζεται η τιμή της μεταβλητής 'Όνομα' .
14. Σωστό, στη μεταβλητή z καταχωρίζεται η τιμή της λογικής πράξης $A>B$, (Αληθής ή Ψευδής).
15. Σωστό, στη μεταβλητή $z1$ καταχωρίζεται η τιμή 'A > B' .
16. Σωστό, στη μεταβλητή A καταχωρείται το αποτέλεσμα της πράξης.
17. Σωστό, στη μεταβλητή Προϊόν καταχωρείται η τιμή της μεταβλητής Όνομα_Προϊόντος .
18. Σωστό, στη μεταβλητή Πρ καταχωρείται η τιμή 'Η/Υ' .
19. Σωστό, στη μεταβλητή Σημαία καταχωρείται το αποτέλεσμα της λογικής πράξης $3>2*5$, δηλαδή $\Psi\epsilon\upsilon\delta\eta\varsigma$.
20. Σωστό, στη μεταβλητή Νίκος καταχωρείται η τιμή 'ΜΑΡΙΑ' .

2. Ποια είναι η τελική τιμή της μεταβλητής a , όταν εκτελεστούν οι παρακάτω εντολές, αν σαν είσοδο δεδομένων από το πληκτρολόγιο δώσουμε τις τιμές 5, 10 και 25.

ΔΙΑΒΑΣΕ α
 $\beta \leftarrow 2^{\wedge}3 + \alpha$
 ΔΙΑΒΑΣΕ γ, β
 $\alpha \leftarrow \beta + \gamma + \alpha$

Λύση

Πίνακας Τιμών			
Εντολές	α	β	γ
1n	5		
2n		13	
3n		25	10
4n	40		

Η τελική τιμή της μεταβλητής a είναι 40.

3. Δίνεται το παρακάτω τμήμα προγράμματος:

$\alpha \leftarrow 3$
 $\chi \leftarrow 10 * \alpha$
 έλεγχος $\leftarrow$ Αληθής
 $\psi \leftarrow \chi < \alpha$
 έλεγχος $\leftarrow$ έλεγχος ΚΑΙ ψ
 $\alpha \leftarrow \alpha + \chi$

Να γίνει ο πίνακας τιμών.

Λύση

Εντολές	α	χ	έλεγχος	ψ
$\alpha \leftarrow 3$	3			
$\chi \leftarrow 10 * \alpha$		30		
έλεγχος $\leftarrow$ Αληθής			Αληθής	
$\psi \leftarrow \chi < \alpha$				Ψευδής
έλεγχος $\leftarrow$ έλεγχος ΚΑΙ ψ			Ψευδής	
$\alpha \leftarrow \alpha + \chi$	33			

4. Τι εμφανίζει το παρακάτω τμήμα προγράμματος;

```
α ← 3^2
ΓΡΑΨΕ 'α = ', α
β ← 5
ΓΡΑΨΕ 'β = ', β
γ ← α * β
ΓΡΑΨΕ 'α * β = ', γ
```

Λύση

Πίνακας τιμών			
Εντολές	α	β	γ
1n	9		
3n		5	
5n			45

ΜΟΝΑΔΑ ΕΙΣΟΔΟΥ - ΟΘΟΝΗ

```
α=9
β=5
α*β=45
```

5. Σε μια εταιρεία οι αποδοχές ενός υπαλλήλου καθορίζονται από το βασικό μισθό, τα επιδόματα και τις κρατήσεις.

- Ο βασικός μισθός είναι 537€. Το επίδομα για κάθε υπάλληλο είναι 12€ για κάθε χρόνο εργασίας.

- Οι κρατήσεις είναι 15% για φόρο επί του βασικού μισθού και 2% για χαρτόσημο επί του βασικού μισθού.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει το όνομα του υπαλλήλου και τα χρόνια υπηρεσίας του στην εταιρεία. Στη συνέχεια να υπολογίζει το επίδομα και τις κρατήσεις του και τέλος να εμφανίζει το βασικό μισθό, το επίδομα, τις κρατήσεις κάθε μια χωριστά και το μισθό που του καταβάλλεται.

Λύση

ΠΡΟΓΡΑΜΜΑ Μισθός

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Χρ, Επίδομα

ΠΡΑΓΜΑΤΙΚΕΣ: Φόρος, Χαρτόσημο, Πληρωτέο

ΧΑΡΑΚΤΗΡΕΣ: Ον

ΑΡΧΗ

ΓΡΑΨΕ 'Όνομα υπαλλήλου: '

ΔΙΑΒΑΣΕ Ον

ΓΡΑΨΕ 'Χρόνια Υπηρεσίας: '

ΔΙΑΒΑΣΕ Χρ

Επίδομα <- Χρ * 12

Φόρος <- 537 * 15/100

Χαρτόσημο <- 537 * 2/100

Πληρωτέο <- 537 + Επίδομα - (Χαρτόσημο + Φόρος)

ΓΡΑΨΕ 'Βασικός Μισθός 537 €'

ΓΡΑΨΕ 'Επίδομα:', Επίδομα

ΓΡΑΨΕ 'Φόρος:', Φόρος

ΓΡΑΨΕ 'Χαρτόσημο:', Χαρτόσημο

ΓΡΑΨΕ 'Μισθός:', Πληρωτέο

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μισθός

1.7 Μη λυμένες Ασκήσεις

1. Ποιες από τις παρακάτω εντολές είναι σωστές, ποιες είναι λάθος και γιατί;

- | | |
|--|--|
| 1. $x * x \leftarrow a1$ | 11. $\gamma \leftarrow \Psi\epsilon\upsilon\delta\acute{\eta}\varsigma$ |
| 2. $x\psi \leftarrow \beta$ | 12. $\acute{\omicron}\nu\omicron\mu\alpha \leftarrow \text{M}\alpha\text{P}\iota\alpha$ |
| 3. $\psi \leftarrow 5^2+3$ | 13. $\acute{\omicron}\nu\omicron\mu\alpha \leftarrow \text{'M}\alpha\text{P}\iota\alpha\text{'}$ |
| 4. $a \leftarrow \text{'}x > 3\text{'}$ | 14. $\text{Τιμή} \leftarrow \text{'12'}$ |
| 5. $\gamma x \leftarrow x > 3$ | 15. $\text{Τιμή} \leftarrow 12$ |
| 6. $x+\psi \beta \leftarrow 5$ | 16. $a \leftarrow 3 < 2 * 5$ |
| 7. $3\kappa \leftarrow a + \beta$ | 17. $a\&\beta \leftarrow a$ |
| 8. $1x \leftarrow x+\psi$ | 18. $\text{ΑΛΗΘΗΣ} \leftarrow a$ |
| 9. $a1 \leftarrow 5+x$ | 19. $x \leftarrow 3 * 2^2+(10-6)^2$ |
| 10. $x \leftarrow \text{'}\Psi\epsilon\upsilon\delta\acute{\eta}\varsigma\text{'}$ | 20. $\text{ΕΛΕΝΗ} \leftarrow X + \psi$ |

2. Να υπολογιστούν οι τιμές των παρακάτω αριθμητικών εκφράσεων:

1. $(3^2+T_P(9))/2$
2. $A_T(10-5^2)+7$
3. $3 \bmod 2+5^2-2 \bmod 3$
4. $(3*2 \bmod 2) \bmod 5$
5. $T_P(3^2+4^2)/(2+1)$
6. $7/2+6$
10. $(7+6)/2$

3. Να γράψετε τις παρακάτω μαθηματικές εκφράσεις σε «ΓΛΩΣΣΑ»:

- | | | |
|---------------------------|--|---|
| 1. $\frac{x^2+\psi^2}{2}$ | 2. $5\beta+\frac{2a+\beta^2}{a+\beta}$ | 3. $\sqrt{x+\psi}-\frac{4}{3+x}$ |
| 4. $\sqrt{\frac{x}{7}}+5$ | 5. $\sqrt{\frac{x+5}{\psi}}+ a+\beta $ | 6. $a^2+\frac{3+\sqrt{x}}{\sqrt{4+\psi^2}}$ |

4. Να βρεθούν οι τιμές των παρακάτω λογικών εκφράσεων, αν η μεταβλητή A έχει τη τιμή 6, η μεταβλητή B έχει τη τιμή -2 και η μεταβλητή Γ έχει τη τιμή 20.

1. $(A > 12 \text{ ΚΑΙ } B > -6) \text{ Ή } (OXI \ A < 6 \text{ ΚΑΙ } B > 11)$
2. $OXI \ (A < 3 \text{ Ή } B > -1)$
3. $(OXI \ A > -2) \text{ ΚΑΙ } (OXI \ B < -13)$
4. $(A < 43 \text{ Ή } B > \Gamma) \text{ ΚΑΙ } A > \Gamma$
5. $OXI \ (A > B \text{ ΚΑΙ } A > \Gamma)$
6. $OXI \ (A > B) \text{ ΚΑΙ } (A > \Gamma)$

5. Ποια είναι η τιμή της μεταβλητής x, όταν εκτελεστούν οι παρακάτω εντολές; Να γίνει πίνακας τιμών.

```
α ← 10
β ← α+3^2
γ ← α+β
x ← (β + γ* 2)* 100
α ← x
x ← β + α
```

6. Τι εμφανίζουν τα παρακάτω προγράμματα για τιμές εισόδου 12 και 7;

ΠΡΟΓΡΑΜΜΑ Πρόγραμμα1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α, β, γ, δ, χ, ψ

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε 2 ακέραιες τιμές τιμές'

ΔΙΑΒΑΣΕ χ, ψ

γ <- χ **div** ψ

δ <- χ **mod** ψ

ΓΡΑΨΕ 'χ=', χ, ' ψ=', ψ

ΓΡΑΨΕ 'Υπόλοιπο=', δ

ΓΡΑΨΕ 'Πηλίκο=', γ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Πρόγραμμα1

ΠΡΟΓΡΑΜΜΑ Πρόγραμμα2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: x, y

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε το x'

ΔΙΑΒΑΣΕ x

ΓΡΑΨΕ 'Δώσε το y'

ΔΙΑΒΑΣΕ y

x ← 3 ^ 2+10 **div** 2^2

y ← 5+2 ^2 * 2+7 **div** 2

ΓΡΑΨΕ 'Νέα τιμή x =', x

ΓΡΑΨΕ 'Νέα τιμή y =', y

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Πρόγραμμα2

7. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει το πλήθος των άτοκων δόσεων που επιθυμεί να πληρώσει ένας πελάτης για κάποια αγορά του και το ποσό της αγοράς. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα το ποσό της κάθε δόσης.

8. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει έναν τριψήφιο αριθμό. Να υπολογίζει και να εμφανίζει το άθροισμα των ψηφίων του.

9. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει το κεφάλαιο που κατατέθηκε σε κάποια τράπεζα με επιτόκιο 1,6%. Να υπολογίζει και να εμφανίζει τον τόκο που θα εισπράξουμε μετά από 3 χρόνια.

10. Ένας υπάλληλος σε μια εταιρεία, επιπλέον του βασικού μισθού του, παίρνει επίδομα 20€ για κάθε παιδί και 30€ επίδομα γάμου, ενώ έχει κρατήσεις 15% στο βασικό μισθό του. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», που να διαβάζει τον βασικό μισθό του υπαλλήλου, και τον αριθμό των παιδιών του (θεωρούμε ότι ο υπάλληλος είναι παντρεμένος). Να υπολογίζει και να εμφανίζει τον τελικό μισθό του, τυπώνοντας κατάλληλο μήνυμα.

2. ΔΟΜΗ ΕΠΙΛΟΓΗΣ

Σε πολλά προβλήματα χρειάζεται να λαμβάνονται αποφάσεις σχετικά με το ποια θα είναι η επόμενη εντολή με βάση κάποια κριτήρια. Η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ). Ανάλογα με το λογικό αποτέλεσμα της συνθήκης επιλέγονται οι εντολές που θα εκτελεστούν.

2.1 Απλή Επιλογή

Γενική μορφή σύνταξης:

```
AN <συνθήκη> ΤΟΤΕ
.....
..... <εντολές προγράμματος>
.....
ΤΕΛΟΣ_ΑΝ
```

Τρόπος εκτέλεσης:

Ελέγχεται η λογική συνθήκη και αν αυτή είναι **ΑΛΗΘΗΣ** τότε εκτελούνται οι εντολές μετά το **ΤΟΤΕ** μέχρι το **ΤΕΛΟΣ_ΑΝ**. Στην περίπτωση που η λογική συνθήκη πάρει την τιμή **ΨΕΥΔΗΣ** τότε οι εντολές αυτές αγνοούνται και η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το **ΤΕΛΟΣ_ΑΝ**.

Παραδείγματα

1. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει την ακτίνα ενός κύκλου και να εμφανίζει το εμβαδόν του κύκλου. Θεωρήστε ότι η ακτίνα είναι θετικός αριθμός.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Εμβαδόν_κύκλου
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: α, Εμ
ΑΡΧΗ
  ΔΙΑΒΑΣΕ α
  ΑΝ α > 0 ΤΟΤΕ
    Εμ <- 3.14* α^2
    ΓΡΑΨΕ 'Εμβαδόν=', Εμ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Εμβαδόν_κύκλου
```

2. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει έναν ακέραιο αριθμό και να εμφανίζει την απόλυτη τιμή του.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Απόλυτη_Τιμή
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: α
ΑΡΧΗ
  ΔΙΑΒΑΣΕ α
  ΑΝ α < 0 ΤΟΤΕ
    α <- (-1)*α
  ΤΕΛΟΣ_ΑΝ
  ΓΡΑΨΕ ' | α | = ', α
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Απόλυτη_Τιμή
```

2.2 Σύνθετη Επιλογή

Γενική μορφή σύνταξης:

```
AN <συνθήκη> ΤΟΤΕ
    .....
    ..... <εντολές 1>
    .....
ΑΛΛΙΩΣ
    .....
    ..... <εντολές 2>
    .....
ΤΕΛΟΣ_ΑΝ
```

Τρόπος εκτέλεσης:

Ελέγχεται η λογική συνθήκη και αν αυτή είναι **ΑΛΗΘΗΣ** εκτελούνται οι **<εντολές 1>**, ενώ αν αυτή είναι **ΨΕΥΔΗΣ** εκτελούνται οι **<εντολές 2>**. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το **ΤΕΛΟΣ_ΑΝ**.

Παραδείγματα

1. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει την ακτίνα κύκλου και αν αυτή είναι θετικός αριθμός τότε να υπολογίζει και να τυπώνεται το εμβαδόν του κύκλου.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Εμβαδόν_κύκλου
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: α, Εμ
ΑΡΧΗ
    ΔΙΑΒΑΣΕ α
    ΑΝ α > 0 ΤΟΤΕ
        Εμ <- 3.14* α^2
        ΓΡΑΨΕ 'Εμβ_κύκλου=', Εμ
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Λάθος η ακτίνα'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Εμβαδόν_κύκλου
```

2. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει έναν ακέραιο αριθμό και να εμφανίζει το διπλάσιό του αν το τελευταίο ψηφίο του αριθμού είναι 2 ή 5, ενώ σε διαφορετική περίπτωση, να υπολογίζει και να εμφανίζει το τριπλάσιό του.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Ψηφία
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: x, y, z
ΑΡΧΗ
    ΔΙΑΒΑΣΕ x
    y <- x mod 10
    ΑΝ y = 2 Η y = 5 ΤΟΤΕ
        z <- 2*x
    ΑΛΛΙΩΣ
        z <- 3*x
    ΤΕΛΟΣ_ΑΝ
    ΓΡΑΨΕ z
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ψηφία
```

3. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει δύο ακέραιους αριθμούς. Αν είναι και οι δύο άρτιοι ή και οι δύο περιττοί, τότε να εμφανίζει το γινόμενο τους, αλλιώς να εμφανίζει την απόλυτη τιμή της διαφορά τους.



Σημείωση:

Για να ελέγξουμε αν ένας αριθμός είναι άρτιος ή περιττός, αρκεί να ελέγξουμε το υπόλοιπο της διαίρεσης με το 2 αν είναι 0 ή όχι (εφόσον όλοι οι άρτιοι είναι διαιρέτες του 2).

Δηλαδή:

AN $x \bmod 2 = 0$

TOTE «0 x είναι άρτιος»

ενώ

AN $x \bmod 2 \neq 0$

TOTE «0 x είναι περιττός».

Λύση

ΠΡΟΓΡΑΜΜΑ Αριθμοί

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: x, y, α, β, z

ΑΡΧΗ

ΔΙΑΒΑΣΕ x, y

α $\leftarrow x \bmod 2$

β $\leftarrow y \bmod 2$

ΑΝ (α=0 **ΚΑΙ** β=0) **Η** (α≠0 **ΚΑΙ** β≠0) **ΤΟΤΕ**

z $\leftarrow x * y$

ΑΛΛΙΩΣ

z $\leftarrow A_T(x - y)$

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ z

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αριθμοί

2.3 Πολλαπλή Επιλογή

Γενική μορφή σύνταξης:

1η Μορφή	2η Μορφή
ΑΝ <συνθήκη1> ΤΟΤΕ <εντολές1>	ΑΝ <συνθήκη1> ΤΟΤΕ
ΑΛΛΙΩΣ	<εντολές1>
ΑΝ <συνθήκη2> ΤΟΤΕ	ΑΛΛΙΩΣ_ΑΝ <συνθήκη2> ΤΟΤΕ
<εντολές2>	<εντολές2>
ΑΛΛΙΩΣ	ΑΛΛΙΩΣ_ΑΝ <συνθήκη3> ΤΟΤΕ
ΑΝ <συνθήκη3> ΤΟΤΕ	<εντολές3>
<εντολές3>	ΑΛΛΙΩΣ_ΑΝ <συνθήκη4> ΤΟΤΕ
ΑΛΛΙΩΣ	<εντολές4>
ΑΝ <συνθήκη4> ΤΟΤΕ	ΑΛΛΙΩΣ_ΑΝ <συνθήκη5> ΤΟΤΕ
<εντολές4>	<εντολές5>
ΑΛΛΙΩΣ	ΤΕΛΟΣ_ΑΝ
<εντολές5>	
ΤΕΛΟΣ_ΑΝ	
ΤΕΛΟΣ_ΑΝ	
ΤΕΛΟΣ_ΑΝ	

Τρόπος εκτέλεσης:

Αρχικά ελέγχεται η <συνθήκη1> και αν είναι αληθής τότε εκτελούνται οι <εντολές1> ενώ αν είναι ψευδής ελέγχεται η <συνθήκη2>. Στην περίπτωση που η <συνθήκη2> είναι αληθής, τότε εκτελούνται οι <εντολές2> ενώ αν είναι ψευδής ελέγχεται η <συνθήκη3>. Στην περίπτωση που η <συνθήκη3> είναι αληθής, τότε εκτελούνται οι <εντολές3> κ.ο.κ

Στη συνέχεια και σε κάθε περίπτωση, η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το ΤΕΛΟΣ_ΑΝ.



Ένα συχνό λάθος που παρατηρείται στα προγράμματα είναι ο έλεγχος περιττών συνθηκών. Οι παραπάνω έλεγχοι αυξάνουν την πολυπλοκότητα του προγράμματος.

Παραδείγματα

1. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει το πλήθος των υπολογιστών που αγοράζονται από κάποιο κατάστημα. Η τιμή αγοράς των υπολογιστών εξαρτάται από το πλήθος των υπολογιστών που αγοράζονται, σύμφωνα με τον παρακάτω πίνακα:

Πλήθος	Τιμή ανά τεμάχιο
1 μέχρι 10	500 €
11 μέχρι 20	450 €
21 μέχρι 50	360 €
51 και πάνω	210 €

Τέλος, να υπολογίζει και να εμφανίζει τη συνολική αξία των υπολογιστών που αγοράστηκαν.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Υπολογιστές
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Πλήθος, Αξία
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε πλήθος υπολογιστών'
    ΔΙΑΒΑΣΕ Πλήθος
    ΑΝ Πλήθος < 0 ΤΟΤΕ
        ΓΡΑΨΕ 'Λάθος πλήθος'
    ΑΛΛΙΩΣ_ΑΝ Πλήθος <= 10 ΤΟΤΕ
        Αξία <- Πλήθος * 500
    ΑΛΛΙΩΣ_ΑΝ Πλήθος <= 20 ΤΟΤΕ
        Αξία <- Πλήθος * 450
    ΑΛΛΙΩΣ_ΑΝ Πλήθος <= 50 ΤΟΤΕ
        Αξία <- Πλήθος * 360
    ΑΛΛΙΩΣ
        Αξία <- Πλήθος * 210
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Πλήθος >= 0 ΤΟΤΕ
        ΓΡΑΨΕ 'Συνολική Αξία=', Αξία
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Υπολογιστές
```

2. Μια εταιρεία κινητής τηλεφωνίας ακολουθεί ανά μήνα την εξής πολιτική τιμών:

Πάγιο 20€	
Χρόνος ομιλίας σε δευτερ.	Χρονοχρέωση (€/δευτερ.)
1 - 300	0.50
301 - 1000	0.45
1001 και άνω	0.30

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ» το οποίο:

- α) Να διαβάξει τη χρονική διάρκεια σε δευτερόλεπτα των τηλεφωνημάτων ενός συνδρομητή σε διάστημα ενός μήνα.
- β) Να υπολογίζει τη μηνιαία χρέωση του συνδρομητή.
- γ) Να εμφανίζει κατάλληλο μήνυμα με τη μηνιαία χρέωση του συνδρομητή.

Θεωρήστε ότι ο χρόνος είναι πάντα θετικός αριθμός.

Λύση

ΠΡΟΓΡΑΜΜΑ Κινητή_Τηλεφωνία

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Χρόνος, Αξία

ΠΡΑΓΜΑΤΙΚΕΣ: Χρέωση

ΑΡΧΗ

ΓΡΑΨΕ 'Χρονική διάρκεια τηλεφώνου'

ΔΙΑΒΑΣΕ Χρόνος

ΑΝ Χρόνος **>= 1 ΚΑΙ** Χρόνος **<= 300 ΤΟΤΕ**

Χρέωση **<-** Χρόνος * 0.50

ΑΛΛΙΩΣ_ΑΝ Χρόνος **<= 1000 ΤΟΤΕ**

Χρέωση **<-** (300 * 0.50) + (Χρόνος - 300) * 0.45

ΑΛΛΙΩΣ

Χρέωση **<-** (300 * 0.50) + (700 * 0.45) + (Χρόνος - 1000) * 0.30

ΤΕΛΟΣ_ΑΝ

Χρέωση **<-** Χρέωση + 20

ΓΡΑΨΕ 'Χρέωση=', Χρέωση, '€'

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Κινητή_Τηλεφωνία



Στο παραπάνω πρόβλημα η χρέωση γίνεται **κλιμακωτά**. Δηλαδή, αν π.χ. η χρονική διάρκεια των τηλεφωνημάτων είναι 650 δευτερόλεπτα, τότε τα πρώτα 300 δευτερόλεπτα, χρεώνονται με 0,50€ το δευτερόλεπτο, ενώ τα υπόλοιπα 350 δευτερόλεπτα (650-300) χρεώνονται με 0,45€ το δευτερόλεπτο.

2.4 Γενικά Παραδείγματα

1. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τρεις πραγματικούς αριθμούς.

Αν το άθροισμά τους είναι μεγαλύτερο του 10, να υπολογίζει και να εμφανίζει το μέσο όρο τους. Διαφορετικά να βρίσκει τον μεγαλύτερο απ' αυτούς και να τον εμφανίζει με κατάλληλο μήνυμα.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Αριθμοί3
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: α, β, γ, ΜΟ, ΜαX
ΑΡΧΗ
  ΔΙΑΒΑΣΕ α, β, γ
  ΑΝ (α + β + γ) > 10 ΤΟΤΕ
    ΜΟ <- (α + β + γ)/3
    ΓΡΑΨΕ 'ΜΟ=', ΜΟ
  ΑΛΛΙΩΣ
    ΑΝ α > β ΚΑΙ α > γ ΤΟΤΕ
      ΜαX <- α
    ΑΛΛΙΩΣ_ΑΝ β > γ ΤΟΤΕ
      ΜαX <- β
    ΑΛΛΙΩΣ
      ΜαX <- γ
  ΤΕΛΟΣ_ΑΝ
  ΓΡΑΨΕ 'ΜαX=', ΜαX
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αριθμοί3
```

2. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να υπολογίζει τον μισθό ενός εργαζόμενου σε μια εταιρεία που εργάζεται με την ώρα.

Να διαβάζει το όνομά του, τις ώρες εργασίας του για τον συγκεκριμένο μήνα και το ποσό που πληρώνεται για την κάθε ώρα.

Οι κρατήσεις που του γίνονται εξαρτώνται από το ύψος του μισθού ως εξής:

Αν ο μισθός του είναι μέχρι 1000 € τον μήνα έχει 15% κρατήσεις, ενώ διαφορετικά έχει 20% κρατήσεις. Επίσης, αν είναι έγγαμος έχει ένα επίδομα 50 €.

Τέλος, να υπολογίζει και να τυπώνει τον τελικό μισθό που λαμβάνει ο υπάλληλος, όπως επίσης να τυπώνει και όλες τις κρατήσεις και τα επιδόματα που του αντιστοιχούν.

Λύση

ΠΡΟΓΡΑΜΜΑ Υπάλληλος

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Ωρες, Επίδομα

ΠΡΑΓΜΑΤΙΚΕΣ: Ωρομίσθιο, Μισθός, Κρατήσεις, Τελικός_Μισθός

ΧΑΡΑΚΤΗΡΕΣ: Ον, Απ

ΑΡΧΗ

ΓΡΑΨΕ 'Ονοματεπώνυμο Υπαλ'

ΔΙΑΒΑΣΕ Ον

ΓΡΑΨΕ 'Ωρες εργασίας'

ΔΙΑΒΑΣΕ Ωρες

ΓΡΑΨΕ 'Ωρομίσθιο:'

ΔΙΑΒΑΣΕ Ωρομίσθιο

Μισθός <- Ωρομίσθιο* Ωρες

ΓΡΑΨΕ 'Έγγαμος ΝΑΙ ή ΟΧΙ '

ΔΙΑΒΑΣΕ Απ

ΑΝ Απ = 'ΝΑΙ' **TOTE**

Επίδομα <- 50

ΑΛΛΙΩΣ

Επίδομα <- 0

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Μισθός=', Μισθός

ΑΝ Μισθός <= 1000 **TOTE**

Κρατήσεις <- Μισθός* 15/100

ΑΛΛΙΩΣ

Κρατήσεις <- Μισθός* 20/100

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Κρατήσεις:', Κρατήσεις

ΓΡΑΨΕ 'Επίδομα:', Επίδομα

Τελικός_Μισθός <- Μισθός + Επίδομα - Κρατήσεις

ΓΡΑΨΕ 'Τελικός Μισθός=', Τελικός_Μισθός

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Υπάλληλος

3. Σ' ένα σχολείο ένας μαθητής μπορεί να κάνει 60 δικαιολογημένες απουσίες και 55 αδικαιολόγητες, ενώ όσες δικαιολογημένες απουσίες κάνει πέρα των 60 θεωρούνται αδικαιολόγητες.

Αν ο μαθητής κάνει πάνω από 55 αδικαιολόγητες απουσίες, τότε δίνει όλα τα μαθήματα τον Σεπτέμβριο.

Αν ο μαθητής κάνει πάνω από 140 αδικαιολόγητες απουσίες, τότε επαναλαμβάνει την τάξη.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τις δικαιολογημένες και τις αδικαιολόγητες απουσίες ενός μαθητή και ανάλογα με το πλήθος τους να τυπώνει ένα από τα παρακάτω μηνύματα:

- 'Ολική τον Σεπτέμβρη' ή
- 'Επανάληψη τάξης' ή
- 'Προβιβασμός'.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Απουσίες
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Δ, Α
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε (Δ)ικαιολογημένες και '
    ΓΡΑΨΕ '(Α)δικαιολόγητες απουσίες μαθητή:'
    ΔΙΑΒΑΣΕ Δ, Α
    ΑΝ Δ > 60 ΤΟΤΕ
        Α <- Α + (Δ - 60)
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Α > 140 ΤΟΤΕ
        ΓΡΑΨΕ 'Επανάληψη Τάξης'
    ΑΛΛΙΩΣ_ΑΝ Α > 55 ΤΟΤΕ
        ΓΡΑΨΕ 'Ολική τον Σεπτέμβρη'
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Προβιβασμός'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Απουσίες
```

4. Σε μια εξεταστική διαδικασία κάθε γραπτό αξιολογείται αρχικά από δύο βαθμολογητές και υπάρχει περίπτωση το γραπτό να χρειάζεται αναβαθμολόγηση από τρίτο βαθμολογητή. Στην περίπτωση αναβαθμολόγησης ο τελικός βαθμός υπολογίζεται ως εξής:

- Αν ο βαθμός του τρίτου βαθμολογητή είναι ίσος με τον μέσο όρο (Μ.Ο.) των βαθμών των δύο πρώτων βαθμολογητών, τότε ο τελικός βαθμός είναι ο Μ.Ο.
- Αν ο βαθμός του τρίτου βαθμολογητή είναι μικρότερος από το μικρότερο βαθμό MIN των δύο πρώτων βαθμολογητών, τότε ο τελικός βαθμός είναι ο MIN.
- Διαφορετικά, ο τελικός βαθμός είναι ο μέσος όρος του βαθμού του τρίτου βαθμολογητή με τον πλησιέστερο προς αυτόν βαθμό των δύο πρώτων βαθμολογητών.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να υπολογίζει τον τελικό βαθμό ενός γραπτού με αναβαθμολόγηση ως εξής:

- 1) να διαβάζει τους βαθμούς του πρώτου, του δεύτερου και του τρίτου βαθμολογητή ενός γραπτού,
- 2) να υπολογίζει και να εμφανίζει τον μεγαλύτερο (MAX) και τον μικρότερο (MIN) από τους βαθμούς του πρώτου και του δεύτερου βαθμολογητή, και
- 3) να υπολογίζει και να εμφανίζει τον τελικό βαθμό του γραπτού, σύμφωνα με την παραπάνω διαδικασία.

Θεωρήστε ότι και οι τρεις βαθμοί είναι θετικοί ακέραιοι αριθμοί και δεν απαιτείται έλεγχος των δεδομένων.

Λύση

ΠΡΟΓΡΑΜΜΑ Μέσος_Όρος

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: B1, B2, B3, MAX, MIN, Διαφορά1, Διαφορά2

ΠΡΑΓΜΑΤΙΚΕΣ: MO, Τελικός_Βαθμός

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε βαθμό Α', Β' και Γ' βαθμολογητή:'

ΔΙΑΒΑΣΕ B1, B2, B3

ΑΝ B1 > B2 **ΤΟΤΕ**

MAX <- B1

MIN <- B2

ΑΛΛΙΩΣ

MAX <- B2

MIN <- B1

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Μεγαλύτερη βαθμολογία από τις δύο πρώτες είναι η ', MAX

ΓΡΑΨΕ 'Μικρότερη βαθμολογία από τις δύο πρώτες είναι η ', MIN

MO <- (B1 + B2)/2

ΑΝ B3 = MO **ΤΟΤΕ**

Τελικός_Βαθμός <- MO

ΑΛΛΙΩΣ_ΑΝ B3 < MIN **ΤΟΤΕ**

Τελικός_Βαθμός <- MIN

ΑΛΛΙΩΣ

Διαφορά1 <- A_T (B3 - B1)

Διαφορά2 <- A_T (B3 - B2)

ΑΝ Διαφορά1 < Διαφορά2 **ΤΟΤΕ**

Τελικός_Βαθμός <- (B3 + B1)/2

ΑΛΛΙΩΣ

Τελικός_Βαθμός <- (B3 + B2)/2

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Τελικός Βαθμός=', Τελικός_Βαθμός

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μέσος_Όρος

5. Μια εταιρεία ταχυδρομικών υπηρεσιών εφαρμόζει για τα έξοδα αποστολής ταχυδρομικών επιστολών εσωτερικού και εξωτερικού, χρέωση σύμφωνα με τον παρακάτω πίνακα:

Βάρος επιστολής σε γρ.	Χρέωση εσωτερικού σε €	Χρέωση εξωτερικού σε €
Από 0 έως και 500	2	4.8
Από 500 έως και 1000	3.50	7.20
Από 1000 έως και 2000	4.60	11.50

Για παράδειγμα, τα έξοδα αποστολής μιας επιστολής βάρους 800 γραμμαρίων και προορισμού εσωτερικού είναι 3,5 €.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο:

1. Να διαβάζει το βάρος της επιστολής.
2. Να διαβάζει τον προορισμό της επιστολής. Η τιμή 'ΕΣ' δηλώνει προορισμό εσωτερικού και η τιμή 'ΕΞ' δηλώνει προορισμό εξωτερικού.
3. Να υπολογίζει τα έξοδα αποστολής ανάλογα με τον προορισμό και το βάρος της επιστολής.

Λύση

```
ΠΡΟΓΡΑΜΜΑ Ταχυδρομείο
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Β
    ΠΡΑΓΜΑΤΙΚΕΣ: Χρ
    ΧΑΡΑΚΤΗΡΕΣ: Απ
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε βάρος επιστολής:'
    ΔΙΑΒΑΣΕ Β
    ΓΡΑΨΕ 'Δώσε προορισμό επιστολής'
    ΓΡΑΨΕ '(ΕΣ)ωτερικό, (ΕΞ)ωτερικό'
    ΔΙΑΒΑΣΕ Απ
    ΑΝ Απ = 'ΕΣ' ΤΟΤΕ
        ΑΝ Β >= 0 ΚΑΙ Β <= 500 ΤΟΤΕ
            Χρ <- 2
        ΑΛΛΙΩΣ_ΑΝ Β <= 1000 ΤΟΤΕ
            Χρ <- 3.5
        ΑΛΛΙΩΣ
            Χρ <- 4.6
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        ΑΝ Β >= 0 ΚΑΙ Β <= 500 ΤΟΤΕ
            Χρ <- 4.8
        ΑΛΛΙΩΣ_ΑΝ Β <= 1000 ΤΟΤΕ
            Χρ <- 7.2
        ΑΛΛΙΩΣ
            Χρ <- 11.5
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
    ΓΡΑΨΕ 'Έξοδα αποστολής:', Χρ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ταχυδρομείο
```

6. Μια δημοτική αρχή ακολουθεί την εξής τιμολογιακή πολιτική για την κατανάλωση νερού ανά μήνα: Χρεώνει πάγιο ποσό 12€ και εφαρμόζει κλιμακωτή χρέωση σύμφωνα με τον παρακάτω πίνακα:

Κατανάλωση νερού σε κυβικά	Χρέωση ανά κυβικό
Από 0 έως και 5	δωρεάν
Από 5 έως και 10	0,5 €
Από 10 έως και 20	0,7 €
Από 20 και άνω	1,00€

Στο ποσό που προκύπτει από την αξία του νερού και το πάγιο, υπολογίζεται ο ΦΠΑ με συντελεστή 23%. Το τελικό ποσό προκύπτει από το άθροισμα της αξίας του νερού, το πάγιο, το ΦΠΑ και το δημοτικό φόρο που είναι 10€.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο:

1. Να διαβάξει τη μηνιαία κατανάλωση του νερού. (Θεωρήστε ότι η μηνιαία κατανάλωση νερού είναι μη αρνητικός αριθμός).
2. Να υπολογίζει την αξία του νερού που καταναλώθηκε, σύμφωνα με την παραπάνω τιμολογιακή πολιτική.
3. Να υπολογίζει τον ΦΠΑ.
4. Να υπολογίζει και να εμφανίζει το τελικό ποσό πληρωμής.

Λύση

ΠΡΟΓΡΑΜΜΑ Νερό

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: K, Αξία, T_Αξία, ΦΠΑ, Τελ_Ποσό

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε μηνιαία κατανάλωση νερού: '

ΔΙΑΒΑΣΕ K

ΑΝ K >= 0 **ΚΑΙ** K <= 5 **ΤΟΤΕ**

Αξία <- 0

ΑΛΛΙΩΣ_ΑΝ K <= 10 **ΤΟΤΕ**

Αξία <- (K - 5) * 0.5

ΑΛΛΙΩΣ_ΑΝ K <= 20 **ΤΟΤΕ**

Αξία <- 5 * 0.5 + (K - 10) * 0.7

ΑΛΛΙΩΣ

Αξία <- 5 * 0.5 + 10 * 0.7 + (K - 20) * 1

ΤΕΛΟΣ_ΑΝ

T_Αξία <- Αξία + 12

ΦΠΑ <- T_Αξία * 23 / 100

Τελ_Ποσό <- T_Αξία + ΦΠΑ + 10

ΓΡΑΨΕ 'Ποσό πληρωμής νερού: ', Τελ_Ποσό

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Νερό

7. Στο τέλος κάθε έτους οι ιδιοκτήτες των αυτοκινήτων προπληρώνουν τα Τέλη Κυκλοφορίας για την επόμενη χρονιά, σύμφωνα με τον κυβισμό των αυτοκινήτων. Το ποσό που αναλογεί για το κάθε αυτοκίνητο ανάλογα με τον κυβισμό του, φαίνεται στον παρακάτω πίνακα:

Κυβισμός	Τέλη Κυκλοφορίας
Έως 300	22 €
301 – 785	55 €
786 - 1.071	120 €
1.072 - 1.357	135 €
1.358 - 1.548	255 €
1.549 - 1.738	280 €
1.739 - 1.928	320 €
1.929 - 2.357	690 €
2.358 - 3.000	920 €
3.001- 4.000	1.150 €
άνω των 4.000	1.380 €

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τον κυβισμό ενός αυτοκινήτου και να υπολογίζει και να τυπώνει τον φόρο που του αναλογεί.

```

ΠΡΟΓΡΑΜΜΑ Φόρος_Αυτοκινήτου
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Κ, Φ
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε κυβισμό αυτοκινήτου'
    ΔΙΑΒΑΣΕ Κ
    ΑΝ Κ <= 300 ΤΟΤΕ
        Φ <- 22
    ΑΛΛΙΩΣ_ΑΝ Κ <= 785 ΤΟΤΕ
        Φ <- 55
    ΑΛΛΙΩΣ_ΑΝ Κ <= 1071 ΤΟΤΕ
        Φ <- 120
    ΑΛΛΙΩΣ_ΑΝ Κ <= 1357 ΤΟΤΕ
        Φ <- 135
    ΑΛΛΙΩΣ_ΑΝ Κ <= 1548 ΤΟΤΕ
        Φ <- 255
    ΑΛΛΙΩΣ_ΑΝ Κ <= 1738 ΤΟΤΕ
        Φ <- 280
    ΑΛΛΙΩΣ_ΑΝ Κ <= 1928 ΤΟΤΕ
        Φ <- 320
    ΑΛΛΙΩΣ_ΑΝ Κ <= 2357 ΤΟΤΕ
        Φ <- 690
    ΑΛΛΙΩΣ_ΑΝ Κ <= 3000 ΤΟΤΕ
        Φ <- 920
    ΑΛΛΙΩΣ_ΑΝ Κ <= 4000 ΤΟΤΕ
        Φ <- 1150
    ΑΛΛΙΩΣ
        Φ <- 1380
    ΤΕΛΟΣ_ΑΝ
    ΓΡΑΨΕ 'Φόρος=', Φ, '€'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Φόρος_Αυτοκινήτου

```

2.5 Μη λυμένες ασκήσεις

1. Να δημιουργήσετε τον πίνακα τιμών των μεταβλητών του προγράμματος A1.

```
ΠΡΟΓΡΑΜΜΑ A1
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A, B, Γ, χ, ψ
ΑΡΧΗ
    χ <- 100
    ψ <- 200
    A <- χ + 1
    B <- 3 * χ
    Γ <- A + B
    ΑΝ Γ > 400 ΤΟΤΕ
        A <- B
        B <- χ
    ΑΛΛΙΩΣ
        ΑΝ B > 200 ΤΟΤΕ
            B <- A
            A <- 20
        ΑΛΛΙΩΣ
            A <- Γ
            B <- A
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ A1
```

2. Τι θα εμφανίσουν τα παρακάτω προγράμματα αν σαν είσοδο δοθούν οι τιμές 5 και 10, στις μεταβλητές χ και ψ αντίστοιχα;

```
ΠΡΟΓΡΑΜΜΑ Π1
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: χ, ψ
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε τιμή για χ και ψ:'
    ΔΙΑΒΑΣΕ χ, ψ
    χ <- χ div 2
    ψ <- ψ mod 3
    ΑΝ χ = ψ ΤΟΤΕ
        χ <- χ^2 + 1
        ψ <- 2 * ψ - 4^2
    ΑΛΛΙΩΣ
        χ <- χ mod 2
        ψ <- ψ div 3
    ΤΕΛΟΣ_ΑΝ
    ΓΡΑΨΕ 'χ =', χ, ' ψ =', ψ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Π1
```

```
ΠΡΟΓΡΑΜΜΑ Π2
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: χ, ψ
    ΛΟΓΙΚΕΣ: f
ΑΡΧΗ
    ΓΡΑΨΕ 'Δώσε 2 ακέραιες τιμές:'
    ΔΙΑΒΑΣΕ χ, ψ
    χ <- χ^2 + 1
    ψ <- 2 * ψ + χ^2
    f <- (χ mod 2 = 0) ΚΑΙ (ψ mod 2 = 0)
    ΑΝ f = ΑΛΗΘΗΣ ΤΟΤΕ
        ΓΡΑΨΕ 'Αρτιοι:', χ, ψ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Π2
```

3. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάσει από το πληκτρολόγιο τρεις διαφορετικούς ακέραιους αριθμούς και στη συνέχεια να τυπώνει τον αριθμό που η αξία του είναι ανάμεσα στον μεγαλύτερο και στον μικρότερο από αυτούς (δηλαδή, αν δοθούν οι αριθμοί 107,53,42 τότε θα εμφανιστεί ο 53, διότι $107 > 53 > 42$ ή $42 < 53 < 107$).

4. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάσει το όνομα ενός υπαλλήλου, τις ώρες εργασίας και το ωρομίσθιο του. Οι κρατήσεις που γίνονται στον υπάλληλο είναι 15%, αν ο μισθός του είναι μέχρι 1.000 €, διαφορετικά είναι 20%. Σε περίπτωση που ο υπάλληλος έχει λιγότερες από 15 ώρες εργασίας τότε οι παραπάνω κρατήσεις μειώνονται κατά 2%.

Υπόδειξη: Ο μισθός κάθε υπαλλήλου υπολογίζεται από τον τύπο:

Ωρες x Ωρομίσθιο – Κρατήσεις

Και οι κρατήσεις από τον τύπο:

Κρατήσεις = (Ωρες x Ωρομίσθιο) x Ποσοστό / 100

5. Σε ένα σχολείο η φοίτηση ενός μαθητή χαρακτηρίζεται ως επαρκής, εάν το σύνολο όλων των απουσιών του δεν ξεπερνά τις 100, εκ των οποίων οι 60 είναι δικαιολογημένες ή το σύνολο των απουσιών του δεν ξεπερνά τις 160 και οι 120 είναι δικαιολογημένες και ο Μέσος όρος βαθμολογίας του μαθητή μεγαλύτερος του 15. Διαφορετικά η φοίτηση χαρακτηρίζεται ως ανεπαρκής.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο:

- 1) Να διαβάσει από το πληκτρολόγιο τον Μέσο Όρο του μαθητή, το σύνολο των Δικαιολογημένων και το σύνολο των Αδικαιολόγητων απουσιών του.
- 2) Να εμφανίζει το μήνυμα 'Φοίτηση επαρκής' ή 'Φοίτηση ανεπαρκής', ανάλογα με τις απουσίες του.

6. Το Υπουργείο Οικονομικών φορολογεί τους Ελεύθερους Επαγγελματίες σύμφωνα με τον παρακάτω πίνακα:

Από 1€ μέχρι 20.000€	22%
Από 20.001€ μέχρι 30.000€	29%
Από 30.001€ μέχρι 40.000€	37%
Από 40.001€ και άνω	45%

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάσει το εισόδημα ενός ελεύθερου επαγγελματία και στη συνέχεια να υπολογίζει και να εμφανίζει τον φόρο που του αναλογεί. Ο υπολογισμός φόρου γίνεται κλιμακωτά, σύμφωνα με τον παραπάνω πίνακα.

- 7.** Με το σύστημα πληρωμής των διοδίων, οι οδηγοί των τροχοφόρων έχουν τη δυνατότητα να πληρώνουν το αντίτιμο των διοδίων με ειδική μαγνητική κάρτα. Υποθέστε ότι υπάρχει μηχανήμα το οποίο διαθέτει είσοδο για την κάρτα και φωτοκύτταρο. Το μηχανήμα διαβάζει από την κάρτα το υπόλοιπο των χρημάτων και το αποθηκεύει σε μια μεταβλητή Υ και με το φωτοκύτταρο αναγνωρίζει τον τύπο του τροχοφόρου και το αποθηκεύει σε μια μεταβλητή Τ. Υπάρχουν τρεις τύποι τροχοφόρων: δίκυκλα (Δ), επιβατικά (Ε) και φορτηγά (Φ), με αντίτιμο διοδίων 1.20€, 2.40€ και 3.50€ αντίστοιχα.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο:

- 1) Να ελέγχει τον τύπο του τροχοφόρου και να εκχωρεί στη μεταβλητή Α το αντίτιμο των διοδίων, ανάλογα με τον τύπο του τροχοφόρου.
- 2) Να ελέγχει την πληρωμή των διοδίων με τον παρακάτω τρόπο:
 - Αν το υπόλοιπο της κάρτας επαρκεί για την πληρωμή του αντιτίμου των διοδίων, αφαιρεί το ποσό αυτό από την κάρτα.
 - Αν η κάρτα δεν έχει υπόλοιπο, το μηχανήμα ειδοποιεί με μήνυμα για το ποσό που πρέπει να πληρωθεί.
 - Αν το υπόλοιπο δεν επαρκεί, μηδενίζεται η κάρτα και δίνεται με μήνυμα το ποσό που απομένει να πληρωθεί.

- 8.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να δέχεται από το πληκτρολόγιο έναν ακέραιο αριθμό και να ελέγχει εάν είναι διψήφιος και θετικός. Στην περίπτωση που ισχύει, να τυπώνεται το μήνυμα «Θετικός, Διψήφιος αριθμός», ενώ διαφορετικά να τυπώνονται για κάθε περίπτωση «Μόνο θετικός» ή «Μόνο διψήφιος» ή «Ούτε θετικός, ούτε διψήφιος»

- 9.** Ένα μαγαζί με ηλεκτρονικά παιχνίδια χρεώνει τους πελάτες του κλιμακωτά, ανάλογα με το πόση ώρα χρησιμοποιούν τα παιχνίδια, σύμφωνα με τον παρακάτω πίνακα:

Χρόνος σε λεπτά	Χρέωση ανά λεπτό
1-60	0,15€
61-90	0,22€
91-120	0,37€
121-180	0,42€
>180	0,55€

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τον χρόνο χρήσης του ηλεκτρονικού παιχνιδιού και στη συνέχεια να υπολογίζει και να εμφανίζει το ποσό χρέωσης, σύμφωνα με τον παραπάνω πίνακα.

10. Ένα ταξιδιωτικό γραφείο οργανώνει 3ήμερες και 5ήμερες εκδρομές στην Κρήτη, Ρόδο και Μύκονο με αεροπλάνο ή πλοίο. Οι τιμές κατ' άτομο για κάθε περίπτωση, βρίσκονται στον παρακάτω πίνακα.

Προορισμός	Αεροπλάνο		Πλοίο	
	3ήμερη	5ήμερη	3ήμερη	5ήμερη
Κρήτη	400€	600€	550€	750€
Ρόδος	500€	700€	700€	900€
Μύκονος	550€	650€	650€	850€

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τον προορισμό, το μέσο μεταφοράς (Π=πλοίο ή Α=Αεροπλάνο), τις ημέρες διαμονής (3 ή 5) και το πλήθος των ατόμων. Στη συνέχεια να υπολογίζει και τυπώνει το κόστος της εκδρομής.

3. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

Η «ΓΛΩΣΣΑ» υποστηρίζει τρεις εντολές επανάληψης:

- Την εντολή «**ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**» που εκτελεί τις εντολές που περικλείει όσο η συνθήκη είναι ΑΛΗΘΗΣ. Η συνθήκη γράφεται στην αρχή της εντολής.
- Την εντολή «**ΜΕΧΡΙΣ_ΟΤΟΥ...**» που εκτελεί τις εντολές, που περικλείει, όσο η συνθήκη είναι ΨΕΥΔΗΣ. Η συνθήκη γράφεται στο τέλος της εντολής.
- Την εντολή «**ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...**» που εκτελεί τις εντολές που περικλείει, προκαθορισμένες φορές

Πού χρησιμοποιούνται οι δομές επανάληψης;

Χρησιμοποιούνται στη περίπτωση που έχουμε επαναλαμβανόμενες εντολές.

Υπάρχουν **τρεις μορφές επανάληψης**, όπου η κάθε μία καλύπτει διαφορετικά είδη επανάληψης.

Δηλαδή, όταν είναι γνωστό πόσες φορές γίνεται η επανάληψη χρησιμοποιούμε τη «**ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...**», εάν δε γνωρίζουμε το πλήθος των επαναλήψεων και η επανάληψη σταματά με μια ερώτηση ή με το πάτημα ενός «κουμπιού» -επανάληψη υπό συνθήκη- τότε χρησιμοποιούμε την «**ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**» ή τη «**ΜΕΧΡΙΣ_ΟΤΟΥ...**».

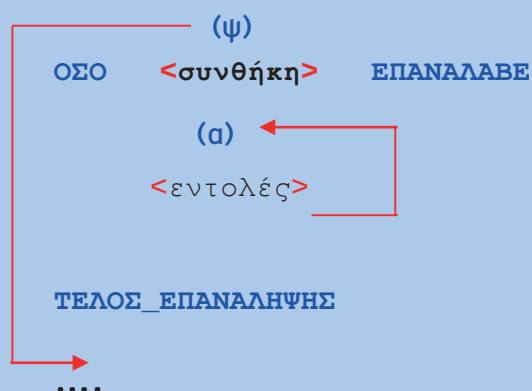
Οι εντολές εκτελούνται σε μια συνθήκη επανάληψης μέχρι να δοθεί μια συγκεκριμένη τιμή (τιμή φρουρός) και τότε μόνο διακόπτεται η επανάληψη. Για παράδειγμα, μια επανάληψη μπορεί να σταματά όταν πληκτρολογήσουμε «Ο» ή αρνητικό αριθμό ή «#» κ.λπ. Επομένως, η τιμή εισόδου που σηματοδοτεί τον τερματισμό της εκτέλεσης μιας επαναληπτικής διαδικασίας ονομάζεται τιμή φρουρός. Συνήθως η τιμή φρουρός χρησιμοποιείται με την εντολή «**ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**».

Παρακάτω θα μελετήσουμε αναλυτικά κάθε μία από τις εντολές επανάληψης.

3.1 Η εντολή επανάληψης «**ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**»

Οι εντολές, που περιλαμβάνει η εντολή «**ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**», επαναλαμβάνονται όσο η συνθήκη είναι ΑΛΗΘΗΣ και μόνο όταν η συνθήκη γίνει ΨΕΥΔΗΣ σταματά η επανάληψή τους και εκτελείται η αμέσως επόμενη εντολή, μετά το «**ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**».

Σύνταξη:



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> δε «βγει» ΨΕΥΔΗΣ ποτέ, τότε οι εντολές της επανάληψης εκτελούνται συνεχώς και το πρόγραμμα δεν τελειώνει ποτέ. Τότε λέμε ότι έχουμε **ατέρμων βρόχο**.

Τρόπος σύνταξης της εντολής

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

1 ^η Περίπτωση	2 ^η Περίπτωση
Αρχική τιμή ΔΙΑΒΑΣΕ α ΟΣΟ α > 0 ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ α ... Μεταβολή τιμής ΔΙΑΒΑΣΕ α ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ Στο παράδειγμα αυτό, η μεταβλητή της συνθήκης είναι η α. Η μεταβλητή αυτή παίρνει τιμές από το πληκτρολόγιο και ως εκ τούτου λαμβάνει αρχική τιμή με την εντολή «ΔΙΑΒΑΣΕ α» πριν την εντολή «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ». Έχοντας τιμή η μεταβλητή α, δύναται να πραγματοποιηθεί ο έλεγχος της συνθήκης «α > 0». Η εντολή «ΔΙΑΒΑΣΕ α» ξαναγράφεται πριν το «ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ» για να αλλάζει τη τιμή της α, με τη νέα τιμή που πληκτρολογούμε. Η παραπάνω επανάληψη διαβάσει και τυπώνει θετικούς αριθμούς που δίνονται από το πληκτρολόγιο.	Αρχική τιμή α <= 100 ΟΣΟ α <= 1000 ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ α ... Μεταβολή τιμής α <= -α + 2 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ Στην περίπτωση αυτή η μεταβλητή α παίρνει τιμές με εντολές καταχώρισης. Απαραίτητη προϋπόθεση είναι η μεταβλητή α να έχει αρχική τιμή (α <= 100), πριν το «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ», προκειμένου να μπορεί να ελεγχθεί η συνθήκη «α <= 1000». Η αλλαγή τιμής της μεταβλητής α γίνεται πριν το τέλος της επανάληψης με την εντολή «α <= -α + 2», έτσι ώστε να αλλάξει τη τιμή της α και να μην έχουμε ατέρμονα βρόχο. Η παραπάνω επανάληψη τυπώνει όλους τους άρτιους αριθμούς από το 100 μέχρι και το 1000.

Παραδείγματα

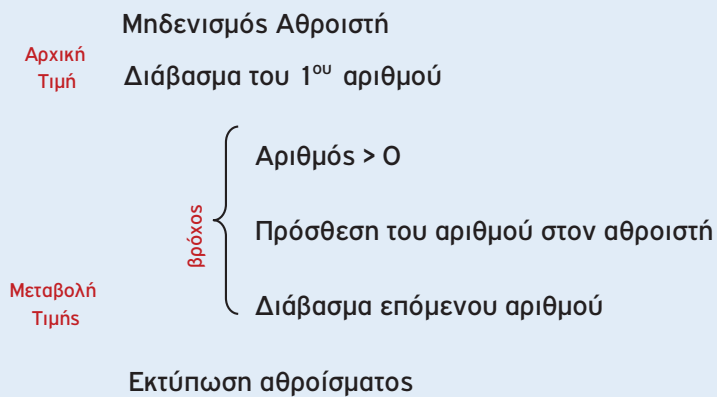
Να γραφεί πρόγραμμα το οποίο δέχεται από το πληκτρολόγιο θετικούς αριθμούς. Η πληκτρολόγηση σταματά με την πληκτρολόγηση αρνητικού αριθμού ή του 0. Τέλος, τυπώνει το άθροισμα των θετικών αριθμών που έχουν πληκτρολογηθεί.

Λύση

Ανάλυση:

1. Εφόσον πληκτρολογούμε συνέχεια θετικούς αριθμούς, έχουμε επανάληψη.
2. Η πληκτρολόγηση σταματά με ένα αρνητικό αριθμό ή το 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 μία, θα χρησιμοποιήσουμε αθροιστή, που σημαίνει μηδενισμός του αθροιστή στην αρχή του προγράμματος.

Γενική μορφή:



```
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
```

3.2 Η εντολή επανάληψης «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ...»

Οι εντολές που περιλαμβάνονται στην εντολή «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ», 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...».

Σύνταξη

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

...<εντολές>

(ψ)

ΜΕΧΡΙΣ_ΟΤΟΥ

<συνθήκη>

...<εντολές>

(α)

Χαρακτηριστικά της εντολής:

- Επειδή η συνθήκη στην εντολή επανάληψης «ΜΕΧΡΙΣ_ΟΤΟΥ...» βρίσκεται στο τέλος της εντολής, οι εντολές που περιλαμβάνονται σ' αυτή θα εκτελεστούν τουλάχιστον μία φορά.
- Οι εντολές επανάληψης «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» και «ΜΕΧΡΙΣ_ΟΤΟΥ...» είναι εντολές επανάληψης υπό συνθήκη και τις χρησιμοποιούμε όπου έχουμε επανάληψη εντολών και η επανάληψη σταματά όταν ισχύει ή δεν ισχύει μία συνθήκη. Επίσης, μπορεί η μια να αντικατασταθεί από την άλλη σ' όλες τις περιπτώσεις επανάληψης υπό συνθήκη, όπως θα δούμε και παρακάτω. Ωστόσο, ανάλογα με το πρόβλημα, μόνο μία από τις δύο είναι η βέλτιστη λύση.

Πού χρησιμοποιείται:

- Στα προβλήματα που έχουν μενού επιλογής.
- Στα προβλήματα που ζητείται έλεγχος δεδομένων από το πληκτρολόγιο.
- Στα προβλήματα που θέλουμε να επαναλάβουμε μια ολόκληρη επεξεργασία (π.χ. αναζήτηση δεδομένων) απαντώντας σε ερώτηση της μορφής 'Θέλεις να συνεχίσεις (Ν/Ο)?'.

Στις παραπάνω τρεις περιπτώσεις, που αναφέρθηκαν για την επιλογή χρήσης της εντολής «ΜΕΧΡΙΣ_ΟΤΟΥ...», μπορεί να χρησιμοποιηθεί και η «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» αλλά χρειάζεται να γραφούν περισσότερες εντολές.

Παράδειγμα 1. Έλεγχος δεδομένων

Να γραφεί πρόγραμμα το οποίο θα δέχεται από το πληκτρολόγιο τον μέσο προφορικό βαθμό και τον γραπτό βαθμό ενός μαθητή και θα υπολογίζει και θα τυπώνει τον μέσο όρο των βαθμών του μαθητή. Κατά την πληκτρολόγηση να γίνεται έλεγχος των βαθμών που πληκτρολογούνται, ώστε να βρίσκονται στο διάστημα [0-20].

Λύση

Ανάλυση:

Το πρόγραμμα χρειάζεται να:

1. Διαβάξει τον μέσο προφορικό βαθμό **με έλεγχο**.
2. Διαβάξει τον μέσο γραπτό βαθμό **με έλεγχο**.
3. Υπολογίζει τον Μέσο Όρο των βαθμών του μαθητή.

ΠΡΟΓΡΑΜΜΑ Έλεγχος

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Πρ, Γρ, ΜΟ

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε προφορικό βαθμό: '

ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Πρ

ΜΕΧΡΙΣ_ΟΤΟΥ Πρ ≤ 20 **ΚΑΙ** Πρ ≥ 0

ΓΡΑΨΕ 'Δώσε γραπτό βαθμό: '

ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Γρ

ΜΕΧΡΙΣ_ΟΤΟΥ Γρ ≤ 20 **ΚΑΙ** Γρ ≥ 0

ΜΟ $\leftarrow (\text{Πρ} + \text{Γρ}) / 2$

ΓΡΑΨΕ 'Μέσος Όρος=', ΜΟ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Έλεγχος

Έλεγχος βαθμού
από **και** 0 έως και
20



Παρατήρηση:

Οι εντολές:

ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Πρ

ΜΕΧΡΙΣ_ΟΤΟΥ Πρ ≤ 20 **ΚΑΙ** Πρ ≥ 0

Διαβάζεται από το πληκτρολόγιο ένας αριθμός, 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 είναι ΑΛΗΘΗΣ.

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

Οι εντολές:

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Πρ

ΑΝ Πρ < 0 **Η** 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.

Παράδειγμα 2. Μενού επιλογής

Να γραφεί πρόγραμμα το οποίο να δέχεται από το πληκτρολόγιο δύο ακέραιους αριθμούς και στη συνέχεια να επιλέγεται η πράξη που πρόκειται να εκτελεστεί για τους παραπάνω αριθμούς, σύμφωνα με το μενού:

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

Όταν επιλεγθεί '5' τότε το πρόγραμμα τερματίζει, διαφορετικά επαναλαμβάνεται η επιλογή με μια νέα πράξη. Να γίνεται έλεγχος της επιλογής για τα νούμερα 1,2,3,4 και 5.

Λύση

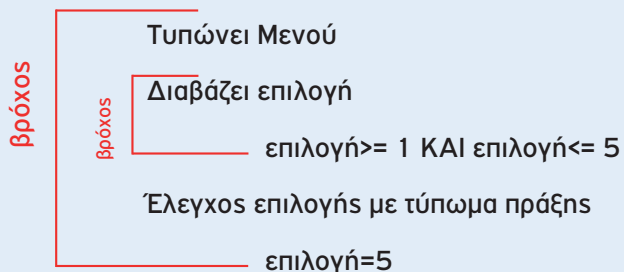
Ανάλυση:

Ακολουθία προγράμματος:

1. Το πρόγραμμα διαβάζει 2 αριθμούς.
2. ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ.
3. Τυπώνεται το ΜΕΝΟΥ.
4. Το πρόγραμμα διαβάζει επιλογή με έλεγχο.
5. Αν επιλογή = 1 τότε γίνεται πρόσθεση.
6. Αν επιλογή = 2 τότε γίνεται αφαίρεση.
7. Αν επιλογή = 3 τότε γίνεται πολλαπλασιασμός.
8. Αν επιλογή = 4 τότε γίνεται διαίρεση.
9. Αν επιλογή = 5 το πρόγραμμα τερματίζει, αλλιώς επιστρέφει στο Βήμα 2.

Γενική Μορφή

Διαβάζει 2 αριθμούς



```

ΠΡΟΓΡΑΜΜΑ Μενού
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: α, β, Αρ
    ΑΚΕΡΑΙΕΣ: Επ
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 Επ
            ΜΕΧΡΙΣ_ΟΤΟΥ Επ >= 1 ΚΑΙ Επ <= 5
            ΑΝ Επ = 1 ΤΟΤΕ
                Αρ <- α + β
                ΓΡΑΨΕ 'Πρόσθεση=', Αρ
            ΑΛΛΙΩΣ_ΑΝ Επ = 2 ΤΟΤΕ
                Αρ <- α - β
                ΓΡΑΨΕ 'Αφαίρεση=', Αρ
            ΑΛΛΙΩΣ_ΑΝ Επ = 3 ΤΟΤΕ
                Αρ <- α * β
                ΓΡΑΨΕ 'Πολλαπλασιασμός=', Αρ
            ΑΛΛΙΩΣ_ΑΝ Επ = 4 ΤΟΤΕ
                ΑΝ β <> 0 ΤΟΤΕ
                    Αρ <- α / β
                    ΓΡΑΨΕ 'Διαίρεση=', Αρ
                ΑΛΛΙΩΣ
                    ΓΡΑΨΕ 'Η διαίρεση δε γίνεται'
                ΤΕΛΟΣ_ΑΝ
            ΤΕΛΟΣ_ΑΝ
            ΜΕΧΡΙΣ_ΟΤΟΥ Επ = 5
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μενού

```

Παράδειγμα 3. Επανάληψη μετά από ερώτηση

Να γραφεί πρόγραμμα το οποίο να διαβάζει το ονοματεπώνυμο και τρεις βαθμούς ενός μαθητή. Στη συνέχεια να υπολογίζει και να τυπώνει τον ΜΟ και το μήνυμα 'ΕΠΙΤΥΧΩΝ' αν $ΜΟ > 10$, διαφορετικά να εμφανίζει το μήνυμα 'ΑΠΟΤΥΧΩΝ'. Στη συνέχεια να τυπώνει το μήνυμα: 'Θέλεις να συνεχίσεις για άλλον μαθητή (Ν/Ο)?' Αν πληκτρολογηθεί 'Ν' τότε να επαναλαμβάνονται τα παραπάνω και για άλλον μαθητή, διαφορετικά να τερματίζει το πρόγραμμα.

Λύση

Ανάλυση:

1. ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ.
2. Διαβάζει όνομα μαθητή.
3. Διαβάζει τους τρεις βαθμούς.
4. Υπολογίζει τον Μέσο Όρο.
5. Ελέγχει και Τυπώνει τον Μέσο Όρο.
6. Τυπώνει το μήνυμα: 'Θέλεις να συνεχίσεις για άλλο μαθητή (Ν/Ο)?'
7. Διαβάζει την Απάντηση . (Με έλεγχο της απάντησης)
8. Αν η Απάντηση είναι 'Ν' 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 **<-** (Β1 + Β2 + Β3) / 3

ΑΝ ΜΟ **>=** 10 **ΤΟΤΕ**

ΓΡΑΨΕ ΜΟ, ' ΕΠΙΤΥΧΩΝ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ ΜΟ, ' ΑΠΟΤΥΧΩΝ'

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ 'Θέλεις να συνεχίσεις για άλλο μαθητή (Ν/Ο)?'

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ Απ

ΜΕΧΡΙΣ_ΟΤΟΥ Απ **=** 'Ν' **Η** Απ **=** 'Ο'

ΜΕΧΡΙΣ_ΟΤΟΥ Απ **=** 'Ο'

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μαθητές

3.3 Η εντολή επανάληψης «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...»

Η εντολή επανάληψης «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» χρησιμοποιείται όταν είναι γνωστό το πλήθος των επαναλήψεων, π.χ. προκειμένου να βρούμε τον ΜΟ των βαθμών 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
```

Τρόπος εκτέλεσης

Στην εντολή αυτή χρησιμοποιείται μια <μεταβλητή> στην οποία αρχικά καταχωρίζεται η <τιμή1>. Η τιμή που παίρνει η <μεταβλητή> συγκρίνεται με τη <τιμή2> και εφόσον είναι μικρότερη ή ίση από αυτή, τότε εκτελούνται οι εντολές που βρίσκονται στον βρόχο.

Στη συνέχεια, η μεταβλητή αυξάνεται κατά την τιμή χ που ορίζει το ΒΗΜΑ. Αν η νέα τιμή είναι μικρότερη ή ίση της τελικής, τότε ο βρόχος εκτελείται ξανά. Η διαδικασία επαναλαμβάνεται μέχρι η <μεταβλητή> να πάρει τιμή μεγαλύτερη από την <τιμή2> αλλιώς το πρόγραμμα συνεχίζει με την εντολή μετά το «ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ».

Η τιμή του βήματος μπορεί να είναι θετικός ή αρνητικός αριθμός και καθορίζει το πλήθος των επαναλήψεων.

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 μία φορά ο βρόχος.
3. **Βήμα < 0** : τότε πρέπει να ισχύει $\text{τιμή1} \geq \text{τιμή2}$ για να εκτελεστεί τουλάχιστον μία φορά ο βρόχος.
4. **Βήμα = 0** : τότε ο βρόχος είναι **ατέρμων**.

Παραδείγματα

1. **ΓΙΑ** κ **ΑΠΟ** 3 **ΜΕΧΡΙ** 3

Ο βρόχος εκτελείται ακριβώς μία φορά.

2. **ΓΙΑ** i **ΑΠΟ** 10 **ΜΕΧΡΙ** 5

Ο βρόχος δεν εκτελείται καμία φορά, διότι $i=10 > 5$ και $\text{Βήμα} > 0$

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, διότι $k=5 < 10$ και $\text{Βήμα} < 0$.

6. **ΓΙΑ** κ **ΑΠΟ** 5 **ΜΕΧΡΙ** 10 **ΜΕ_ΒΗΜΑ** 0

Ο βρόχος εκτελείται άπειρες φορές



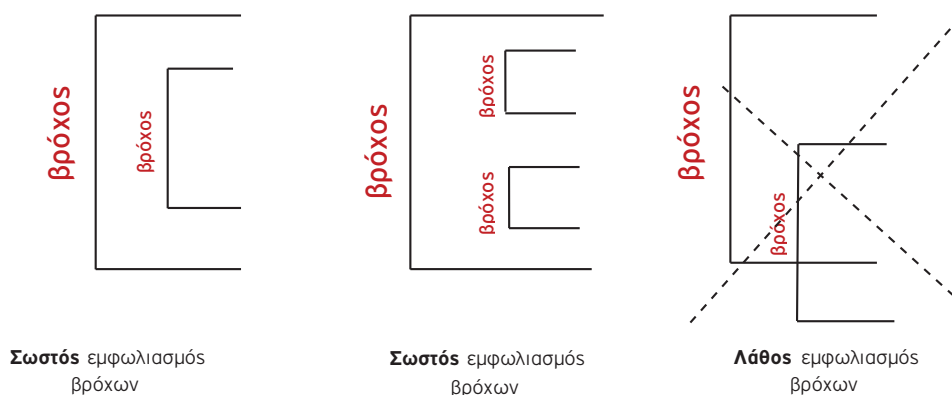
Παρατηρήσεις:

1. Μέσα στην εντολή «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» δεν επιτρέπεται να αλλάξει η τιμή της <μεταβλητής> του ΓΙΑ, διότι τότε δε 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...ΕΠΑΝΑΛΑΒΕ» και «ΜΕΧΡΙΣ_ΟΤΟΥ...», ενώ το αντίστροφο δεν ισχύει πάντα.

3.4 Εμφωλευμένοι Βρόχοι

Στην περίπτωση που ένας βρόχος βρίσκεται μέσα σε άλλον, τότε έχουμε εμφωλευμένους βρόχους. 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.

3.5 Παραδείγματα εύρεσης ελάχιστου ή μέγιστου

Όταν έχουμε μια ομάδα αριθμών από τους οποίους θέλουμε να βρούμε τον ελάχιστο ή τον μέγιστο, δε 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 εντολή «ΔΙΑΒΑΣΕ <μεταβλητή>».

Για τον λόγο αυτό δεν μπορούμε με πολλά και εμφωλευμένα «ΑΝ...ΑΛΛΙΩΣ_ΑΝ...» να βρούμε τον ελάχιστο ή τον μέγιστο και έτσι χρησιμοποιούμε τον παρακάτω αλγόριθμο:

1. Χρησιμοποιούμε δύο μεταβλητές: Max και Min, που εκφράζουν τη μεγαλύτερη και τη μικρότερη τιμή αντίστοιχα που θα έχουν στο τέλος. Στις μεταβλητές αυτές δίνουμε **Αρχική Τιμή**.
2. Συγκρίνουμε κάθε νέα τιμή που δίνουμε στη <μεταβλητή> με τις Max και Min αντίστοιχα (<μεταβλητή> < Min ή <μεταβλητή> > Max) και αν ισχύουν οι συνθήκες αυτές καταχωρίζουμε στη Min ή στη Max τη νέα τιμή της <μεταβλητής>. Για την **Αρχική Τιμή** υπάρχουν δύο περιπτώσεις.

1η περίπτωση (Αυθαίρετες τιμές)	2η περίπτωση (Συγκεκριμένες τιμές)
Στη Max δίνουμε τη μικρότερη δυνατή τιμή. Στη Min δίνουμε τη μεγαλύτερη δυνατή τιμή. Αυτόν τον τρόπο τον χρησιμοποιούμε στις περιπτώσεις που το πεδίο τιμών της μεταβλητής είναι συγκεκριμένο. Π.χ. δίνουμε τους βαθμούς μαθητών [0,20]. Τότε σαν αρχικές τιμές μπορούμε να δώσουμε:	Στη Max ή στη Min δίνουμε την πρώτη τιμή από το σύνολο τιμών που πρόκειται να δώσουμε. Αυτόν τον τρόπο τον χρησιμοποιούμε στις περιπτώσεις που το πεδίο τιμών της μεταβλητής δεν είναι συγκεκριμένο. Π.χ. Πληκτρολογούμε αριθμούς και ζητείται να βρούμε τον μεγαλύτερο ή τον μικρότερο από αυτούς.
Max $\leftarrow -\theta$ και Min $\leftarrow -2\theta$	ΔΙΑΒΑΣΕ B Max $\leftarrow$ B Min $\leftarrow$ B

Παραδείγματα

- 1 Να γραφεί πρόγραμμα το οποίο θα διαβάζει 1000 αριθμούς από το πληκτρολόγιο και θα βρίσκει και θα τυπώνει τον μεγαλύτερο απ' αυτούς.

Λύση

Ανάλυση:

1. Χρησιμοποιούμε την επαναληπτική εντολή «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» 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:

Διαβάζει αριθμό
 Καταχωρίζει αριθμό στη Max
 βρόχος $i = 2$
 Διαβάζει αριθμό
 Ελέγχει αριθμό με τη Max
 Μέχρι $i > 1000$
 Τυπώνει τη Max

```

ΠΡΟΓΡΑΜΜΑ Μέγιστος
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: A, i, Max
ΑΡΧΗ
  ΓΡΑΨΕ 'Δώσε αριθμό'
  ΔΙΑΒΑΣΕ A
  Αρχική Τιμή Max <- A
  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 1000
    ΓΡΑΨΕ 'Δώσε αριθμό'
    ΔΙΑΒΑΣΕ A
    ελέγχος [ AN A > Max ΤΟΤΕ
              Max <- A
            ]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Μεγαλύτερος ο:', Max
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μέγιστος

```



Παρατήρηση:

Στο παραπάνω παράδειγμα παρατηρούμε ότι για να δώσουμε ως αρχική τιμή στη Max (ή αντίστοιχα Min) τον πρώτο αριθμό από τη λίστα αριθμών που θα πληκτρολογήσουμε, τότε το πρώτο ΔΙΑΒΑΣΕ χρειάζεται να γίνει έξω από την επανάληψη.

2. Να γραφεί πρόγραμμα, το οποίο θα υπολογίζει και θα τυπώνει τον μεγαλύτερο και τον μικρότερο από 1000 αριθμούς που εισάγονται από το πληκτρολόγιο.

Λύση

Ανάλυση:

1. Χρησιμοποιούμε την επαναληπτική εντολή «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» 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:

```

Διαβάζει αριθμό
Καταχωρίζει τον αριθμό στη Max
Καταχωρίζει τον αριθμό στη Min
βρόχος [ i = 2
        Διαβάζει αριθμό
        Ελέγχει αριθμό με τη Max
        Ελέγχει αριθμό με τη Min
        Μέχρι i > 1000
      ]
Τυπώνει τη Max και τη 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

  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Μεγαλύτερος: ', Max
  ΓΡΑΨΕ 'Μικρότερος: ', Min
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Max_Min

```



Παρατήρηση:

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

Δηλαδή, χρειάζεται να χρησιμοποιήσουμε δύο διαφορετικά «ΑΝ...», ένα για τη Max και ένα για τη Min.

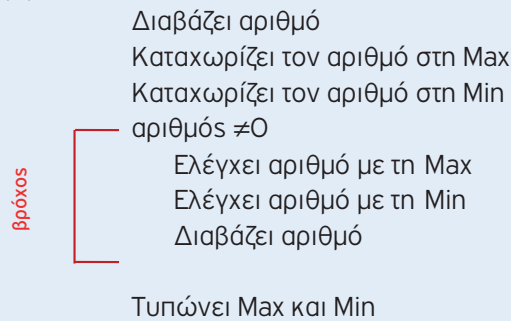
3. Να γραφεί πρόγραμμα, το οποίο να βρίσκει και να τυπώνει τον μεγαλύτερο και τον μικρότερο από τους αριθμούς που εισάγονται από το πληκτρολόγιο. Το τέλος της πληκτρολόγησης δηλώνεται αν πληκτρολογήσουμε τον αριθμό 0.

Λύση

Ανάλυση:

1. Χρησιμοποιούμε την επαναληπτική εντολή «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» για να πληκτρολογήσουμε τους αριθμούς, διότι εδώ η πληκτρολόγηση σταματά όταν δώσουμε την τιμή **φρουρό «0»**.
2. 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

4. Να γραφεί πρόγραμμα, το οποίο θα διαβάζει τα ονόματα και τους αντίστοιχους ΜΟ των βαθμών 70 μαθητών της Β' Λυκείου ενός σχολείου και θα υπολογίζει και θα τυπώνει το όνομα και τον μέσο όρο του μαθητή με τον μεγαλύτερο ΜΟ και του μαθητή με τον μικρότερο ΜΟ. Θεωρήστε ότι ο μέσος όρος κάθε μαθητή είναι μοναδικός.

Λύση

Ανάλυση:

- Χρησιμοποιούμε την επαναληπτική εντολή ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» για να διαβάσουμε τα 70 ονόματα και τους μέσους όρους των μαθητών.
- Θα χρησιμοποιηθεί ο αλγόριθμος του μεγαλύτερου και του μικρότερου, με τη διαφορά ότι θα χρησιμοποιηθούν και βοηθητικές μεταβλητές που θα κρατούν (αποθηκεύουν) αντίστοιχα το όνομα του μαθητή με τον μεγαλύτερο ΜΟ και του μαθητή με τον μικρότερο ΜΟ.
- 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

$i = 1$

Διαβάζει Όνομα και Μέσο Όρο μαθητή

Ελέγχει τον Μέσο Όρο με τη Max

Ελέγχει τον Μέσο Όρο με τη Min

Μέχρι $i = 70$

Τυπώνει τη Max και το Όνομα του μαθητή με τον μεγαλύτερο ΜΟ

Τυπώνει τη Min και το Όνομα του μαθητή με τον μικρότερο ΜΟ

ΠΡΟΓΡΑΜΜΑ Ο_μεγαλύτερος_και_ο_μικρότερος_βαθμός

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, Max, Min

ΧΑΡΑΚΤΗΡΕΣ: Ον, 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 Ον, ΜΟ

ΑΝ ΜΟ > Max **ΤΟΤΕ**

Max $\leftarrow$ ΜΟ

Μαχ_ον $\leftarrow$ Ον

ΤΕΛΟΣ_ΑΝ

ΑΝ ΜΟ < Min **ΤΟΤΕ**

Min $\leftarrow$ ΜΟ

Μιν_ον $\leftarrow$ Ον

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

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

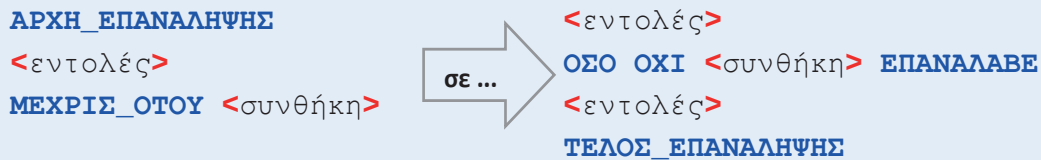
ΓΡΑΨΕ 'Χειρότερος ο ', Μιν_ον, 'με Μ.Ο.=', Min

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ο_μεγαλύτερος_και_ο_μικρότερος_βαθμός

3.6 Μετατροπές από μία δομή επανάληψης σε άλλη

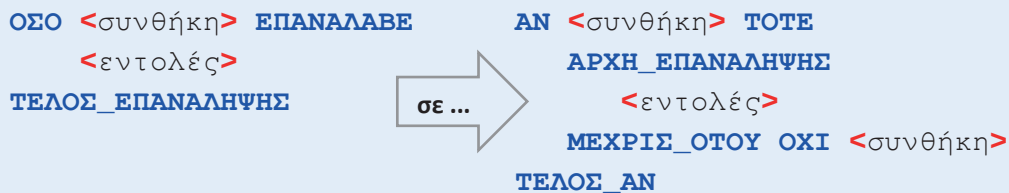
Μετατροπή από «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ» σε «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» και αντιστρόφως

Περίπτωση 1



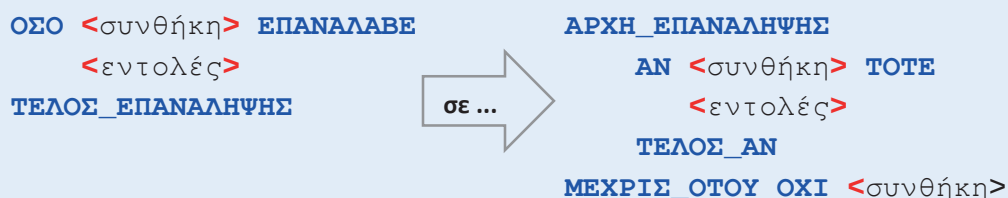
- Οι εντολές στη δομή επανάληψης «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ», εκτελούνται όσο η <συνθήκη> μετά το ΜΕΧΡΙΣ_ΟΤΟΥ είναι Ψευδής, ενώ στη δομή επανάληψης «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ», εκτελούνται όσο η <συνθήκη> ανάμεσα στο ΟΣΟ και το ΕΠΑΝΑΛΑΒΕ είναι Αληθής. Γι' αυτό, κατά τη μετατροπή από τη μια δομή επανάληψης στην άλλη, αρκεί να γράψουμε την άρνηση της <συνθήκη> της πρώτης στη δεύτερη ή να προτάξουμε τον τελεστή **ΟΧΙ** στη <συνθήκη>.
- Επίσης, η «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ» εκτελεί τουλάχιστον μία φορά όλες τις εντολές της, γι' αυτό όταν τη μετατρέπουμε στην «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» πρέπει όλες τις εντολές που περιέχει να τις γράψουμε δύο φορές, μία πριν την «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» και άλλη μία μέσα σ' αυτήν.

Περίπτωση 2



- Η δομή επανάληψης «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» εκτελείται όσο η <συνθήκη> ανάμεσα στο ΟΣΟ και το ΕΠΑΝΑΛΑΒΕ είναι Αληθής, ενώ η δομή επανάληψης «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ» εκτελείται όσο η συνθήκη μετά το ΜΕΧΡΙΣ_ΟΤΟΥ είναι Ψευδής. Γι' αυτό κατά τη μετατροπή από τη μία δομή στην άλλη αρκεί να γράψουμε την άρνηση της <συνθήκη> της πρώτης στη δεύτερη ή να προτάξουμε τον τελεστή **ΟΧΙ** στη <συνθήκη>.
- Επίσης, η «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» μπορεί να μην εκτελεστεί καμία φορά σε αντίθεση με την «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ», η οποία θα εκτελεστεί τουλάχιστον μία φορά, γι' αυτό πριν τη δομή «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ» θα χρησιμοποιηθεί μια εντολή ελέγχου, που αν ισχύει η <συνθήκη> θα εκτελεστεί η «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ».

Ως μη βέλτιστη λύση (διότι η συνθήκη ελέγχεται δύο φορές), για την ίδια μετατροπή μπορεί να δοθεί και η παρακάτω:



Παραδείγματα

1. Μετατροπή από «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» σε «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ»

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



```
ΔΙΑΒΑΣΕ x
ΑΝ x <> 0 ΤΟΤΕ
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    y ← x ^ 2
    ΓΡΑΨΕ y
  ΔΙΑΒΑΣΕ x
ΜΕΧΡΙΣ_ΟΤΟΥ x = 0
ΤΕΛΟΣ_ΑΝ
```

2. Μετατροπή από «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ» σε «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ»

```
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
```

Μετατροπή από «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» σε «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» και αντιστρόφως

1. Περίπτωση τιμή1 <= τιμή2 και β > 0

```
ΓΙΑ <μεταβλητή> ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή2 ΜΕ_ΒΗΜΑ β
  <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
<μεταβλητή> ← τιμή1
ΟΣΟ <μεταβλητή> <= τιμή2 ΕΠΑΝΑΛΑΒΕ
  <εντολές>
  <μεταβλητή> ← <μεταβλητή> + β
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

2. Περίπτωση τιμή1 >= τιμή2 και β < 0

```
ΓΙΑ <μεταβλητή> ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή2 ΜΕ_ΒΗΜΑ β  
  <εντολές>  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
<μεταβλητή> ← τιμή1  
ΟΣΟ <μεταβλητή> >= τιμή2 ΕΠΑΝΑΛΑΒΕ  
  <εντολές>  
  <μεταβλητή> ← <μεταβλητή> + β  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- Πριν την εντολή «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» εκχωρούμε στη <μεταβλητή> την αρχική τιμή της «ΓΙΑ... ΑΠΟ...ΜΕΧΡΙ...», δηλ. την τιμή1.
- Στη συνθήκη της «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» συγκρίνουμε τη <μεταβλητή> με την τιμή2. Η σύγκριση εξαρτάται από το βήμα αν είναι θετικό ή αρνητικό, όπως βλέπουμε παραπάνω.
- Πριν το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ της «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» αυξάνουμε την τιμή της μεταβλητής κατά την τιμή του βήματος. Στην περίπτωση που το βήμα δεν υπάρχει, τότε η <μεταβλητή> αυξάνεται κατά 1.
- Η μετατροπή της «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» σε «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...», γίνεται μόνο στην περίπτωση που στην «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» είναι γνωστός ο αριθμός των επαναλήψεων, σε οποιαδήποτε άλλη περίπτωση δεν είναι δυνατή η μετατροπή της «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» σε «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...».

Παραδείγματα

1. Μετατροπή από «ΓΙΑ... ΑΠΟ...ΜΕΧΡΙ...» σε «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ»

α)

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



```
i ← 1  
ΟΣΟ i <= 10 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...ΕΠΑΝΑΛΑΒΕ» σε «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...»

α)

```
i <- 10
```

```
ΟΣΟ i <= 100 ΕΠΑΝΑΛΑΒΕ
```

```
  ΓΡΑΨΕ i
```

```
  i <- i+1
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
ΓΙΑ i ΑΠΟ 10 ΜΕΧΡΙ 100
```

```
  ΓΡΑΨΕ i
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

β)

```
i <- -10
```

```
ΟΣΟ i <= 100 ΕΠΑΝΑΛΑΒΕ
```

```
  ΓΡΑΨΕ i
```

```
  i <- i + 2
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
ΓΙΑ i ΑΠΟ 10 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
```

```
  ΓΡΑΨΕ i
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

γ)

```
i <- -100
```

```
ΟΣΟ i >= 1 ΕΠΑΝΑΛΑΒΕ
```

```
  ΓΡΑΨΕ i
```

```
  i <- i - 2
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



```
ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 1 ΜΕ_ΒΗΜΑ -2
```

```
  ΓΡΑΨΕ i
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

3.7 Μη λυμένες ασκήσεις

1. Πόσες φορές θα εκτελεστούν οι παρακάτω βρόχοι επανάληψης;

1) ΓΙΑ i ΑΠΟ -3 ΜΕΧΡΙ 1

2) ΓΙΑ i ΑΠΟ 4 ΜΕΧΡΙ 2

3) ΓΙΑ i ΑΠΟ 5 ΜΕΧΡΙ 5

4) ΓΙΑ i ΑΠΟ 6 ΜΕΧΡΙ 6 ΜΕ_ΒΗΜΑ 0

5) ΓΙΑ i ΑΠΟ -9 ΜΕΧΡΙ -15 ΜΕ_ΒΗΜΑ -1

6) ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2

7) ΓΙΑ i ΑΠΟ -4 ΜΕΧΡΙ -1 ΜΕ_ΒΗΜΑ -2

8) ΓΙΑ i ΑΠΟ 5 ΜΕΧΡΙ 12 ΜΕ_ΒΗΜΑ 3

9) ΓΙΑ i ΑΠΟ 0.5 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ 0.5

10) ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ 0

2. Πόσες φορές θα εκτελεστούν οι παρακάτω βρόχοι;

1)

```
α ← 0
ΟΣΟ α < 15 ΕΠΑΝΑΛΑΒΕ
  χ ← α + 1
  α ← α + 3
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

2)

```
α ← -12
ΟΣΟ α > 0 ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ α
  α ← α + 3
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

3)

```
α ← 10
ΟΣΟ α >= 20 ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ α
  α ← α + 5
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

4)

```
α ← 2
ΟΣΟ α <= 11 ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ α
  α ← α + 5
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

3. Πόσες φορές θα εκτελεστούν οι παρακάτω βρόχοι;

1)

```
β ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ β
  β ← β + 3
ΜΕΧΡΙΣ_ΟΤΟΥ β > 13
```

2)

```
α ← -9
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ α
  α ← α - 3
ΜΕΧΡΙΣ_ΟΤΟΥ α > -1
```

3)

```
β ← 200
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ β
  β ← β - 50
ΜΕΧΡΙΣ_ΟΤΟΥ β <= 0
```

4)

```
β ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ β
  β ← β + 3
ΜΕΧΡΙΣ_ΟΤΟΥ β > 7
```

4. Δίνεται το παρακάτω τμήμα προγράμματος σε «ΓΛΩΣΣΑ»:

```
sum ← 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 6 ΜΕ_ΒΗΜΑ 2
  sum ← sum + i
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

i) Να μετατραπεί σε ισοδύναμο με χρήση της δομής ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

ii) Να μετατραπεί σε ισοδύναμο με χρήση της δομής ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ... ΜΕΧΡΙΣ_ΟΤΟΥ

5. Δίνεται το παρακάτω τμήμα προγράμματος σε «ΓΛΩΣΣΑ»:

```
sum ← 0
i ← 0
ΔΙΑΒΑΣΕ α
ΟΣΟ i ≤ 5 ΕΠΑΝΑΛΑΒΕ
    sum ← sum + α
    i ← i + 1
    ΔΙΑΒΑΣΕ α
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

i) Να μετατραπεί σε ισοδύναμο με χρήση της δομής ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ

ii) Να μετατραπεί σε ισοδύναμο με χρήση της δομής ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ

6. Δίνεται το παρακάτω τμήμα προγράμματος σε «ΓΛΩΣΣΑ»:

```
sum ← 10
ΔΙΑΒΑΣΕ x
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    sum ← sum + x
    ΔΙΑΒΑΣΕ x
ΜΕΧΡΙΣ_ΟΤΟΥ x ≤ 0
```

Να μετατραπεί σε ισοδύναμο με χρήση της δομής ΟΣΟ... ΕΠΑΝΑΛΑΒΕ και της δομής ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ

7. Τι εμφανίζεται στην οθόνη κατά την εκτέλεση του παρακάτω προγράμματος σε «ΓΛΩΣΣΑ»; Να συμπληρωθεί ο πίνακας τιμών.

```
ΠΡΟΓΡΑΜΜΑ Ασκηση_Επανάληψης
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: α, β, i, j
ΑΡΧΗ
    α ← 10
    β ← 5
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20 ΜΕ_ΒΗΜΑ 5
        ΓΙΑ j ΑΠΟ i ΜΕΧΡΙ 15 ΜΕ_ΒΗΜΑ 2
            α ← α div 2
            β ← β mod 3
            ΑΝ α <> 0 ΤΟΤΕ
                ΓΡΑΨΕ α
            ΤΕΛΟΣ_ΑΝ
            ΑΝ β <> 0 ΤΟΤΕ
                ΓΡΑΨΕ β
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ασκηση_Επανάληψης
```

8. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ» το οποίο να υπολογίζει τα παρακάτω αθροίσματα:

1. $1 + 3 + 5 + \dots + 99$

2. $\frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \dots + \frac{1}{100}$

3. $2^2 + 4^2 + \dots + 100^2$

9. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει θετικό ακέραιο αριθμό N. Στη συνέχεια, να διαβάζει θετικούς αριθμούς μέχρι το άθροισμά τους να γίνει μεγαλύτερο από τον αριθμό N. Τέλος, να υπολογίζει και να τυπώνει τον μεγαλύτερο αριθμό που πληκτρολογήθηκε. Να γίνεται έλεγχος των αριθμών που πληκτρολογούνται, ώστε να είναι θετικοί, σε αντίθετη περίπτωση να εμφανίζεται μήνυμα λάθους και να ζητείται η πληκτρολόγηση ενός θετικού αριθμού.

10. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τα ονόματα των 26 μαθητών μιας τάξης και 10 βαθμούς για τον κάθε μαθητή. Το πρόγραμμα να υπολογίζει και να εμφανίζει τον Μέσο Όρο για κάθε μαθητή. Στο τέλος το πρόγραμμα να τυπώνει το όνομα του μαθητή με τον μεγαλύτερο μέσο όρο, το όνομα του μαθητή με τον μικρότερο μέσο όρο καθώς και τον μέσο όρο τους αντίστοιχα. Θεωρήστε ότι ο μέσος όρος κάθε μαθητή είναι μοναδικός.

11. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει το πλήθος των μαθητών ενός σχολείου. Στη συνέχεια να διαβάζει τον μέσο όρο κάθε μαθητή. Να υπολογίζει και να τυπώνει το πλήθος και το αντίστοιχο ποσοστό των μαθητών που έχουν μέσο όρο < 10, των μαθητών με μέσο όρο στο διάστημα [10, 18] και των μαθητών με μέσο όρο > 18. Ο βαθμός που πληκτρολογείται είναι από 1 μέχρι 20. Το πρόγραμμα να κάνει έλεγχο εγκυρότητας του βαθμού και στην περίπτωση που δίνεται βαθμός έξω από τα επιτρεπτά όρια, να τυπώνει το μήνυμα «Μη αποδεκτή βαθμολογία. Ξαναπροσπαθήστε.....».

12. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει πραγματικούς αριθμούς και να υπολογίζει και να εμφανίζει το άθροισμα και το πλήθος των αριθμών που πληκτρολογήθηκαν. Το πρόγραμμα να διαβάζει πραγματικούς αριθμούς μέχρι να πληκτρολογήσουμε το 100 ή όταν το άθροισμα των αριθμών που έχουν πληκτρολογηθεί γίνει ίσο με 1.000.000. Ο αριθμός 100 που σηματοδοτεί και το τέλος της πληκτρολόγησης θα λαμβάνεται υπόψη στο άθροισμα και στο πλήθος.

13. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει έναν ακέραιο αριθμό και να υπολογίζει και να τυπώνει:

1. το άθροισμα των ψηφίων του, και
2. το πλήθος των ψηφίων του

- 14.** Η σχολή Θετικών Επιστημών ενός Πανεπιστημίου έχει 4 τμήματα. Κάθε τμήμα δέχεται κάθε χρονιά 150 φοιτητές. Η πρυτανεία αποφάσισε για το τρέχον έτος να κάνει μια στατιστική μελέτη για κάθε τμήμα. Έτσι, αποφάσισε να υπολογίσει το ποσοστό των αριστούχων, δηλ. των νεοεισαχθέντων φοιτητών με βαθμό μεγαλύτερο ή ίσο του 19, ανά σχολή.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τους βαθμούς εισαγωγής των φοιτητών και να υπολογίζει και να τυπώνει :

1. το ποσοστό των αριστούχων φοιτητών κάθε τμήματος, και
2. σε ποιο τμήμα εισήχθη ο φοιτητής με τον μεγαλύτερο βαθμό.

Θεωρήστε ότι ο μεγαλύτερος βαθμός είναι μοναδικός.

- 15.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να περιγράφει τη λειτουργία ενός συγκεκριμένου ασανσέρ. Ο θάλαμος του ασανσέρ χωράει μέχρι 8 άτομα με το μέγιστο συνολικά επιτρεπτό ωφέλιμο βάρος των ατόμων να είναι 900 κιλά. Το ασανσέρ ξεκινά όταν το σύνολο των ατόμων στον θάλαμο δεν ξεπερνά τους οκτώ ή όταν το συνολικό βάρος των ατόμων δεν υπερβαίνει τα 900 κιλά ή όταν δεν υπάρχει άλλο άτομο να εισέλθει σε αυτό από τον κάθε όροφο που έχει σταματήσει. Το ασανσέρ κάνει συνολικά 8 στάσεις (μία σε κάθε όροφο). Σε κάθε στάση, εκτός της τελευταίας, το πρόγραμμα εμφανίζει το μήνυμα «Υπάρχει άτομο να εισέλθει; (ΝΑΙ ή ΟΧΙ)». Αν η απάντηση είναι «ΝΑΙ» τότε «εισάγεται» το βάρος του ατόμου και έτσι σηματοδοτείται η πρόθεσή του να εισέλθει στο ασανσέρ. Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να δοθεί η απάντηση «ΟΧΙ».

Επίσης, σε κάθε στάση, εκτός από την πρώτη, το πρόγραμμα εμφανίζει το μήνυμα «Υπάρχει άτομο να εξέλθει; (ΝΑΙ ή ΟΧΙ)», αν η απάντηση είναι «ΝΑΙ» τότε «εξάγεται» το βάρος του ατόμου κι έτσι σηματοδοτείται η έξοδός του από το ασανσέρ. Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να δοθεί η απάντηση «ΟΧΙ».

Το πρόγραμμα υπολογίζει και τυπώνει:

- το σύνολο των ατόμων που εισέρχονται στον θάλαμο του ασανσέρ σε κάθε στάση (εκτός της τελευταίας στάσης),
- το σύνολο των ατόμων που εξέρχονται από κάθε στάση (εκτός της πρώτης στάσης),
- σε όλες τις στάσεις, πλην της 8^{ης}, το πλήθος των ατόμων που βρίσκονται στον θάλαμο αμέσως πριν την εκκίνηση του ασανσέρ για την επόμενη στάση.
- σε όλες τις στάσεις, πλην της 8^{ης}, το συνολικό βάρος των ατόμων που βρίσκονται στον θάλαμο αμέσως πριν την εκκίνηση του ασανσέρ για την επόμενη στάση.

Σημειώνεται ότι:

- η λειτουργία του ασανσέρ επιτρέπει μία μόνο μετάβαση από την 1^η μέχρι και την 8^η στάση
- στην 1^η στάση επιτρέπεται μόνο η είσοδος και όχι η έξοδος από τον θάλαμο,
- στην 8^η στάση εξέρχονται όλοι από τον θάλαμο χωρίς να επιτρέπεται νέα είσοδος.

4. ΠΙΝΑΚΕΣ

Πίνακας, είναι ένα σύνολο αντικειμένων **ίδιου τύπου** τα οποία αναφέρονται με κοινό όνομα. Κάθε αντικείμενο καλείται και στοιχείο του πίνακα. Ένας πίνακας μπορεί να είναι μονοδιάστατος, δισδιάστατος, τρισδιάστατος και στη γενικότερη περίπτωση μπορεί να είναι ν-διάστατος.

Για να αναφερθούμε σε ένα στοιχείο ενός πίνακα, χρησιμοποιούμε το όνομα μιας μεταβλητής και έναν ακέραιο δείκτη που είναι ένας αριθμός και δείχνει τη θέση του κάθε στοιχείου (για την περίπτωση μονοδιάστατου πίνακα), π.χ. A[20]

δύο ακέραιους δείκτες που δείχνουν τη θέση του κάθε στοιχείου (για την περίπτωση δισδιάστατου πίνακα), π.χ. A[2,3]

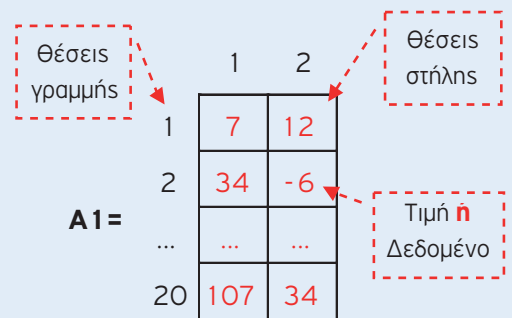
Στην περίπτωση των δισδιάστατων πινάκων, όταν το μέγεθος των δύο διαστάσεων είναι ίσο, δηλαδή ο πίνακας είναι nxn, τότε ο πίνακας λέγεται **τετραγωνικός**.

Οι πίνακες αποτελούν **στατικές δομές**.

Πλεονεκτήματα χρήσης Πινάκων: Οι πίνακες χρησιμοποιούνται στην περίπτωση που χρειάζεται να διαχειριστούμε πολλά δεδομένα του ίδιου τύπου και τα οποία απαιτείται να είναι αποθηκευμένα για μεταγενέστερη χρήση τους.

Μειονεκτήματα χρήσης Πινάκων: Οι πίνακες, από την αρχή του προγράμματος, δεσμεύουν συγκεκριμένο πλήθος θέσεων μνήμης -ανάλογα με το μέγεθός τους- και περιορίζουν τις δυνατότητες του προγράμματος, καθώς το μέγεθός τους είναι προκαθορισμένο (δε μεταβάλλεται).

4.1 Δήλωση πίνακα στη «ΓΛΩΣΣΑ».

Μονοδιάστατος	Δισδιάστατος
«τύπος»: Όνομα_πίνακα [πλήθος θέσεων]	«τύπος»: Όνομα_πίνακα [πλήθος γραμμών, πλήθος στηλών].
ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: A[20] ΧΑΡΑΚΤΗΡΕΣ: B[20]	ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: A1[20, 2] ΧΑΡΑΚΤΗΡΕΣ: B1[5, 3]
Ο πίνακας A[20] θα μπορούσε να θεωρηθεί ως μια μεταβλητή A με 20 θέσεις. Σχηματικά:	Ο πίνακας A1[20,2] θα μπορούσε να θεωρηθεί ως μια μεταβλητή A1 με 20x2=40 θέσεις (20 γραμμές και 2 στήλες). Σχηματικά:
	
	Μια θέση στοιχείου (ή δεδομένου) σε έναν δισδιάστατο πίνακα καθορίζεται από τη γραμμή και από τη στήλη του πίνακα που βρίσκεται το στοιχείο (ή δεδομένο).

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.

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 τη γραμμή και μετά τη στήλη του πίνακα.

Π.χ. A[10,2]



4.2 Είσοδος δεδομένων σε μονοδιάστατο και σε δισδιάστατο πίνακα

Είσοδος δεδομένων σε μονοδιάστατο πίνακα

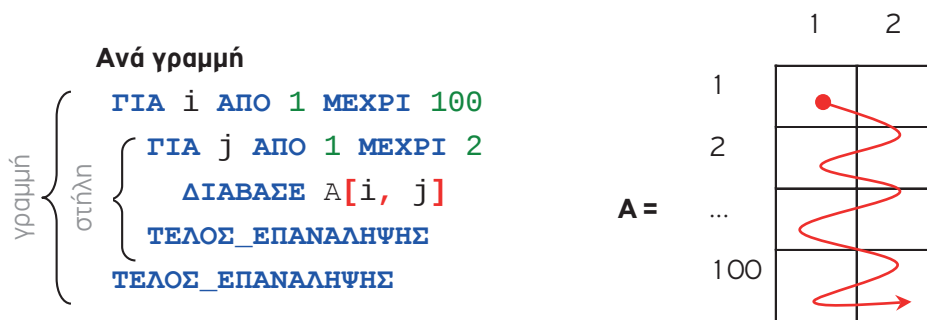
Έστω ότι θέλουμε να διαβάσουμε 100 ακέραιους αριθμούς από το πληκτρολόγιο:

1ος τρόπος (χωρίς χρήση πίνακα)	2ος τρόπος (με χρήση πίνακα)
<pre>ΔΙΑΒΑΣΕ A1 ΔΙΑΒΑΣΕ A2 ΔΙΑΒΑΣΕ A3 ΔΙΑΒΑΣΕ A4 ... ΔΙΑΒΑΣΕ A100</pre> Αυτός ο τρόπος δε χρησιμοποιείται, διότι γράφουμε πολλές φορές την ίδια εντολή. ή <pre>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100 ΔΙΑΒΑΣΕ A ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</pre> 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.	<pre>ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: A[100] ΑΡΧΗ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100 ΔΙΑΒΑΣΕ A[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ... ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ</pre> Κατά την εκτέλεση των παραπάνω εντολών πληκτρολογούμε 100 αριθμούς, έτσι ώστε ο κάθε ένας να μπαίνει σε διαφορετική θέση στον πίνακα. Αυτό έχει σαν αποτέλεσμα να μπορούμε στο πρόγραμμά μας παρακάτω, να τους επεξεργαστούμε όλους ή κάθε έναν απ' αυτούς, όποτε τους χρειαζόμαστε.

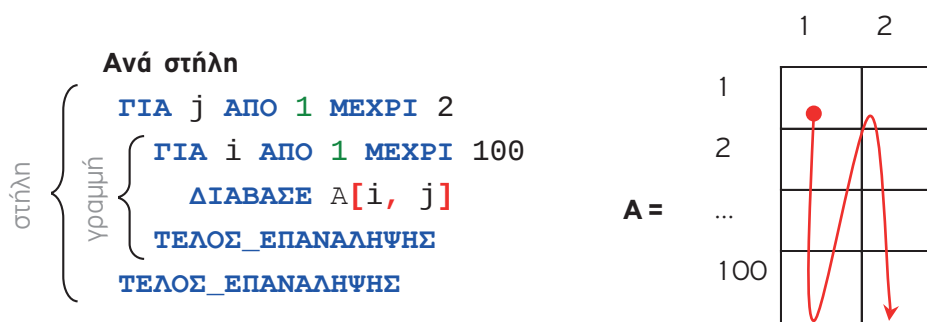
Είσοδος δεδομένων σε δισδιάστατο πίνακα

Στην περίπτωση αυτή μας ενδιαφέρει αν τα δεδομένα καταχωριστούν στον πίνακα ανά γραμμή ή ανά στήλη.

π.χ. Διάβασμα 20 ακέραιων και καταχώριση στον πίνακα A[100,2]



Με τις παραπάνω εντολές **γεμίζουμε τον πίνακα ανά γραμμή** (όταν γεμίζει μία γραμμή, τότε συνεχίζει το γέμισμα από την αρχή της επόμενης γραμμής).



Με τις παραπάνω εντολές **γεμίζουμε τον πίνακα ανά στήλη** (όταν γεμίζει μία στήλη, τότε συνεχίζει το γέμισμα από την αρχή της επόμενης στήλης).



Παρατήρηση:

Η ανάγνωση, η επεξεργασία και η εκτύπωση των στοιχείων ενός πίνακα γίνεται πάντα από βρόχους, που επαναλαμβάνονται προκαθορισμένο αριθμό φορές όσα είναι και τα στοιχεία του πίνακα. Επομένως, η επεξεργασία των πινάκων είναι προτιμότερο να γίνεται με την εντολή επανάληψης «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...».

Παράδειγμα 1

Να γραφεί πρόγραμμα, το οποίο να υπολογίζει και να τυπώνει 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 την εντολή επανάληψης «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...» 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, διότι 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: MO

ΑΡΧΗ

sum <- 0

ΓΡΑΨΕ 'Δώσε 30 θερμοκρασίες'

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΔΙΑΒΑΣΕ Θ[i]

 sum <- sum + Θ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

MO <- sum/30

ΓΡΑΨΕ 'Μέση θερμοκρασία:', MO

M <- 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΑΝ Θ[i] < MO **ΤΟΤΕ**

 M <- M + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Ημέρες με θερμοκρασία<Μέση θερ:', M

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ θερμοκρασία

Παράδειγμα 2

Να γραφεί πρόγραμμα, το οποίο να γεμίζει έναν δισδιάστατο πίνακα 4×3 , ανά γραμμή, με ακέραιους αριθμούς που δίνονται από το πληκτρολόγιο. Στη συνέχεια να υπολογίζει και να τυπώνει:

1. Το άθροισμα των στοιχείων του πίνακα ανά γραμμή
2. Το άθροισμα των στοιχείων του πίνακα ανά στήλη
3. Το συνολικό άθροισμα όλων των αριθμών του πίνακα.

Λύση

Γενική μορφή:

Δίνω τα δεδομένα

- $i = 1$
- $j = 1$
- Μέχρι $j = 3$
- Μέχρι $i = 4$

Πληκτρολόγηση δεδομένων και καταχώρησή τους στον πίνακα

Άθροισμα ανά γραμμή

- $i = 1$
- Μηδενισμός αθροιστή γραμμής
- $j = 1$
- Πρόσθεση στοιχείου στον αθροιστή γραμμής
- Μέχρι $j = 3$
- Εκτύπωση του αθροιστή γραμμής
- Μέχρι $i = 4$

Άθροισμα ανά στήλη

- $j = 1$
- Μηδενισμός αθροιστή στήλης
- $i = 1$
- Πρόσθεση στοιχείου στον αθροιστή στήλης
- Μέχρι $i = 4$
- Εκτύπωση του αθροιστή στήλης
- Μέχρι $j = 3$

Μηδενισμός αθροιστή

Συνολικό άθροισμα

- $i = 1$
- $j = 1$
- Πρόσθεση στοιχείου στον αθροιστή
- Μέχρι $j = 3$
- Μέχρι $i = 4$
- Εκτύπωση αθροιστή

```

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 'Αθροισμα 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 'Αθροισμα 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

```



Παρατηρήσεις:

- Αν τα ζητούμενα σ' έναν δισδιάστατο πίνακα αφορούν γραμμή ή στήλη, τότε οι αρχικές τιμές για άθροισμα, ελάχιστο, μέγιστο, κ.λπ. δίνονται ανάμεσα στις δύο «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...».
- Αν τα ζητούμενα σ' έναν δισδιάστατο πίνακα αφορούν όλο τον πίνακα, τότε οι αρχικές τιμές δίνονται πριν από τις δύο «ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...».

Παράδειγμα 3

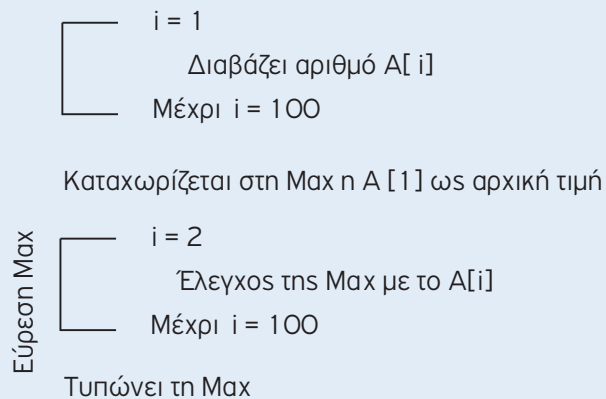
Να γραφεί πρόγραμμα, το οποίο να υπολογίζει και να τυπώνει τον μεγαλύτερο αριθμό, από 100 ακέραιους αριθμούς που θα διαβαστούν από το πληκτρολόγιο.

Λύση

Ανάλυση:

- θα χρησιμοποιηθεί πίνακας 100 θέσεων για να καταχωριστούν οι αριθμοί.
- θα εφαρμοστεί ο αλγόριθμος του μεγαλύτερου, θέτοντας ως αρχική τιμή την 1η θέση του πίνακα.

Γενική μορφή:



```
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 ← A[1]

    ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
        ΑΝ A[i] > Max ΤΟΤΕ
            Max ← A[i]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    ΓΡΑΨΕ 'Μεγαλύτερος ο ', Max
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Max_Μονοδιάστατο
```

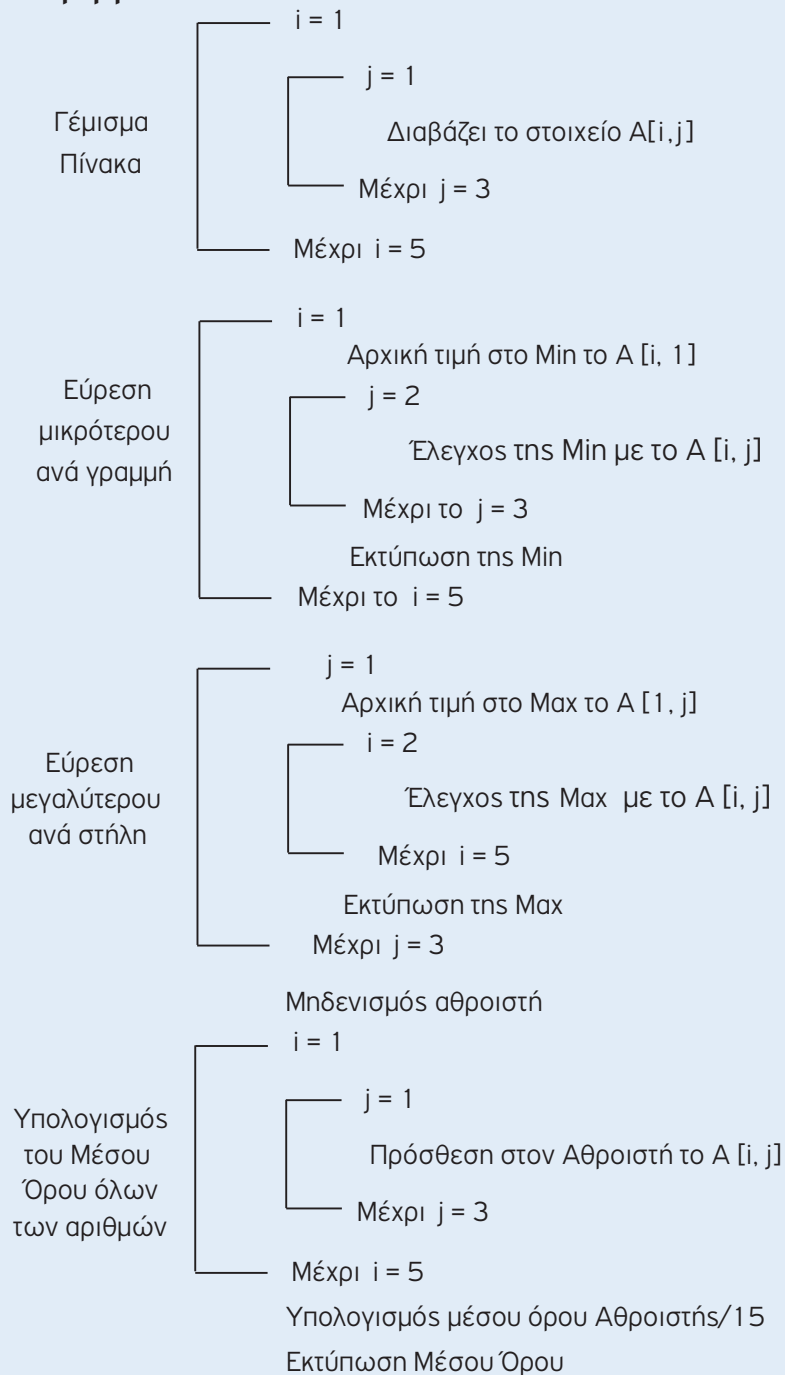
Παράδειγμα 4

Να γραφεί πρόγραμμα, το οποίο να γεμίζει έναν 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) τον μέσο όρο όλων των αριθμών.

Λύση

Γενική μορφή:



```

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

```

4.3 Τυπικές επεξεργασίες πινάκων

Τα προγράμματα τα οποία χρησιμοποιούν πίνακες συχνά απαιτούν συγκεκριμένες επεξεργασίες στα στοιχεία του πίνακα, 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.

4.3.1 Ταξινόμηση στοιχείων πινάκα

Ταξινόμηση ονομάζουμε την τακτοποίηση των κόμβων μιας δομής με μια ιδιαίτερη σειρά.

Η ταξινόμηση μπορεί να γίνει:

- κατ' αύξουσα σειρά (δηλ. από τον μικρότερο προς το μεγαλύτερο)
- κατά φθίνουσα σειρά (δηλ. από τον μεγαλύτερο προς το μικρότερο)

Υπάρχουν διάφοροι αλγόριθμοι ταξινόμησης πίνακα. Εδώ θα αναπτυχθεί ο αλγόριθμος της ταξινόμησης ευθείας ανταλλαγής ή ταξινόμηση φυσαλίδας.

Ταξινόμηση ευθείας ανταλλαγής ή φυσαλίδας (Bubblesort)

Με τη μέθοδο αυτή της ταξινόμησης συγκρίνουμε και ανταλλάσσουμε ζεύγη γειτονικών στοιχείων μέχρι να ταξινομηθούν όλα τα στοιχεία.

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 σειρά)

```
ΠΡΟΓΡΑΜΜΑ Ταξινομήση1
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: A[100], i, j, K
ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΔΙΑΒΑΣΕ A[i]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100
    ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
      ΑΝ A[j - 1] > A[j] ΤΟΤΕ
        Κ <- A[j - 1]
        A[j - 1] <- A[j]
        A[j] <- Κ
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΡΑΨΕ A[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ταξινομήση1
```

Ταξινομήση του πίνακα A

Πόσες φορές θα προσπελάσω τον πίνακα

Αντιμετάθεση τιμών

Ποια στοιχεία του πίνακα ελέγχω



Παρατηρήσεις:

Στην περίπτωση που η ταξινομήση είναι κατά φθίνουσα σειρά, 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.

Η μεταβλητή Flag ελέγχει από ποιο σημείο και μετά ο πίνακας είναι ταξινομημένος (όταν αλλάξει η τιμή της και συγκεκριμένα όταν η Flag από ΑΛΗΘΗΣ γίνει ΨΕΥΔΗΣ, τότε και τερματίζει η διαδικασία ταξινόμησης).

Ταξινόμηση
του πίνακα A

ΠΡΟΓΡΑΜΜΑ Ταξινόμηση2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], i, j, K

ΛΟΓΙΚΕΣ: Flag

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

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 Ταξινόμηση2

Όταν δεν ισχύει η συνθήκη, σημαίνει ότι το επόμενο στοιχείο είναι μικρότερο ή ίσο του προηγούμενου. Δηλαδή, βρέθηκε το σημείο, από το οποίο και μετά, ο πίνακας είναι ταξινομημένος. Η Flag από ΑΛΗΘΗΣ γίνεται ΨΕΥΔΗΣ και τερματίζει η διαδικασία

4.3.2 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 ταχύτερη εύρεση του αναζητούμενου στοιχείου. Εμείς θα ασχοληθούμε με δύο αλγόριθμους:

1. τη Σειριακή αναζήτηση, και
2. τη Δυναμική αναζήτηση.

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 μία προς μία, 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 έναν 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 "Δε βρέθηκε".

```
ΠΡΟΓΡΑΜΜΑ Αναζήτηση1
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A[100], i, KEY
ΑΡΧΗ
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
        ΔΙΑΒΑΣΕ A[i]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση: '
    ΔΙΑΒΑΣΕ KEY
    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 'Δε βρέθηκε'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αναζήτηση1
```

Περίπτωση 2^η

Αναζήτηση του στοιχείου (KEY) σε έναν 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 "Δε βρέθηκε".

```
ΠΡΟΓΡΑΜΜΑ 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 'Δε βρέθηκε'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αναζήτηση2
```



Παρατήρηση:

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

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.

Δυαδική αναζήτηση

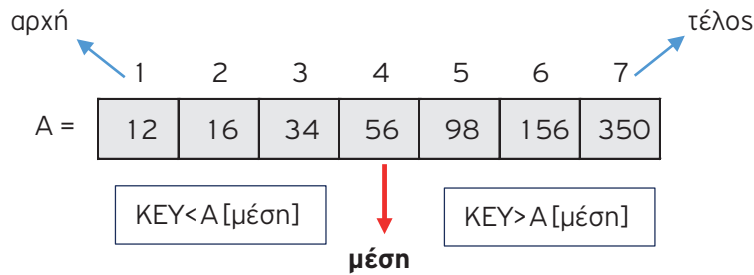
Η δυαδική αναζήτηση είναι αποτελεσματικότερη από τη σειριακή, διότι χρειάζεται πολύ λιγότερο χρόνο εκτέλεσης- κάτω από τον μισό χρόνο- από ότι η σειριακή αναζήτηση.

Απαραίτητη προϋπόθεση είναι ότι ο πίνακας πρέπει να είναι ταξινομημένος.

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 στη μεσαία θέση.
- Αν όχι, ελέγχουμε αν το ζητούμενο στοιχείο είναι μικρότερο από το στοιχείο του πίνακα που βρίσκεται στη μεσαία θέση. Αν ισχύει, 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 συμβεί αρχή > τέλος.

Η παραπάνω διεργασία στηρίζεται στο ότι ο πίνακας είναι ταξινομημένος κατ' αύξουσα σειρά.
Για παράδειγμα:



Η αντίστοιχη διεργασία 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.

```

ΠΡΟΓΡΑΜΜΑ Δυναμική_Αναζήτηση1
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: A[100], KEY, τέλος, αρχ, μέση, ι
    ΛΟΓΙΚΕΣ: δεικτης
ΑΡΧΗ
    ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 100
        ΔΙΑΒΑΣΕ A[ι]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.'
    ΔΙΑΒΑΣΕ KEY
    αρχ <- 1
    τέλος <- 100
    δεικτης <- ΨΕΥΔΗΣ
    ΟΣΟ αρχ <= τέλος ΚΑΙ δεικτης = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
        μέση <- (αρχ + τέλος) div 2
        ΑΝ μέση >= 1 ΚΑΙ μέση <= 100 ΤΟΤΕ
            ΑΝ KEY < A[μέση] ΤΟΤΕ
                τέλος <- μέση - 1
            ΑΛΛΙΩΣ_ΑΝ KEY > A[μέση] ΤΟΤΕ
                αρχ <- μέση + 1
            ΑΛΛΙΩΣ
                δεικτης <- ΑΛΗΘΗΣ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ δεικτης = ΑΛΗΘΗΣ ΤΟΤΕ
        ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Δε βρέθηκε'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Δυναμική_Αναζήτηση1
    
```

Περίπτωση 2^η

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

ΠΡΟΓΡΑΜΜΑ Δυαδική_Αναζήτηση2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[5], KEY, τέλος, αρχ, μέση, i

ΛΟΓΙΚΕΣ: δεικτης

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε αριθμό για αναζήτηση στον πίνακα A.'

ΔΙΑΒΑΣΕ KEY

αρχ **<-** 1

τέλος **<-** 5

δεικτης **<-** ΨΕΥΔΗΣ

ΟΣΟ αρχ **<=** τέλος **ΚΑΙ** δεικτης **=** ΨΕΥΔΗΣ **ΕΠΑΝΑΛΑΒΕ**

μέση **<-** (αρχ + τέλος) **div** 2

ΑΝ μέση **>=** 1 **ΚΑΙ** μέση **<=** 5 **ΤΟΤΕ**

ΑΝ KEY **<** A[μέση] **ΤΟΤΕ**

αρχ **<-** μέση + 1

ΑΛΛΙΩΣ_ΑΝ KEY **>** A[μέση] **ΤΟΤΕ**

τέλος **<-** μέση - 1

ΑΛΛΙΩΣ

δεικτης **<-** ΑΛΗΘΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ δεικτης **=** ΑΛΗΘΗΣ **ΤΟΤΕ**

ΓΡΑΨΕ 'Βρέθηκε στη θέση ', μέση

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δε βρέθηκε'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Δυαδική_Αναζήτηση2

4.3.3 Συγχώνευση δύο πινάκων

Η συγχώνευση είναι μια από τις βασικές επεξεργασίες 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 πληροφορικής της Γ΄ Λυκείου και πίνακας B[21] με τα ονόματα του 2ου τμήματος πληροφορικής της Γ΄ Λυκείου. Να δημιουργηθεί πίνακας με τα ονόματα και των δύο τμημάτων.

Λύση

ΠΡΟΓΡΑΜΜΑ Συγχώνευση1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[25], B[21], i, κ, Γ[46]

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

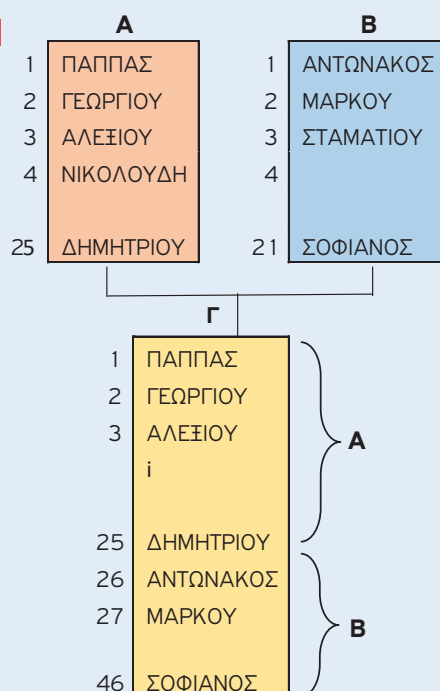
Συγχώνευση
των πινάκων

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 46

ΓΡΑΨΕ Γ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Συγχώνευση1



Περίπτωση 2^η

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

Λύση

Ανάλυση:

1. Στην 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 καταχωρίζεται στον πίνακα Γ και αυξάνεται κατά μία μονάδα η θέση του πίνακα που πήραμε το στοιχείο του και το καταχωρίσαμε στον πίνακα Γ.
3. Η διαδικασία αυτή σταματά όταν τελειώσουν τα στοιχεία που συγκρίνουμε σε έναν από τους δύο πίνακες, τον A ή B.
4. Στη συνέχεια, τα στοιχεία που περίσσεψαν από κάποιον πίνακα καταχωρίζονται στον πίνακα Γ.

```

ΠΡΟΓΡΑΜΜΑ Συγχώνευση2
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: A[25], B[21], i, κ, j, Γ[46]
ΑΡΧΗ
  Είσοδος
  δεδομένων {
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 25
      ΔΙΑΒΑΣΕ A[i]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 21
      ΔΙΑΒΑΣΕ B[i]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  }
  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 l ΑΠΟ j ΜΕΧΡΙ 21
      Γ[κ] <- B[l]
      κ <- κ + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΛΛΙΩΣ
    ΓΙΑ j ΑΠΟ i ΜΕΧΡΙ 25
      Γ[κ] <- A[j]
      κ <- κ + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΑΝ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 46
    ΓΡΑΨΕ Γ[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Συγχώνευση2

```

Συνένωση 2
ταξινομημένων
πινάκων με
απευθείας
ταξινόμηση

4.4 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.

Για παράδειγμα, οι εγγραφές που αφορούν στοιχεία μισθοδοσίας υπαλλήλων μιας εταιρείας.

ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΟΙΚΟΓΕΝΕΙΑΚΗ 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 μπορούμε να τα κρατήσουμε σε 6 μονοδιάστατους πίνακες, διότι τα στοιχεία είναι διαφορετικού τύπου (χαρακτήρες και αριθμοί). Οπότε είναι αδύνατη η χρήση δισδιάστατου 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

Να γραφεί πρόγραμμα που να δέχεται ως είσοδο τα ονόματα και τις εισπράξεις 50 εταιρειών. Στη συνέχεια να τυπώνει τα ονόματα των εταιρειών με εισπράξεις μεγαλύτερες από τον μέσο όρο των εισπράξεων και τα ονόματα των εταιρειών με εισπράξεις μικρότερες από τον μέσο όρο των εισπράξεων.

Λύση

Ανάλυση:

1. Το όνομα και η εισπράξη κάθε εταιρείας καταχωρούνται σε διαφορετικούς πίνακες που είναι παράλληλοι. Επομένως, τα δεδομένα διαβάζονται ταυτόχρονα.
2. Υπολογίζουμε τον μέσο όρο των εισπράξεων από τον πίνακα των εισπράξεων.
3. Ελέγχουμε κάθε εισπράξη αν είναι μεγαλύτερη του μέσου όρου εισπράξης και τυπώνουμε το αντίστοιχο όνομα εταιρείας.
4. Ελέγχουμε κάθε εισπράξη 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
    ΓΡΑΨΕ 'Δώσε 50 ονόματα εταιρειών και '
    ΓΡΑΨΕ 'τις αντίστοιχες εισπράξεις τους'
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
        ΔΙΑΒΑΣΕ Ετ[i], ΕΙΣ[i]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    sum <- 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
        sum <- sum + ΕΙΣ[i]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ <- sum/50

    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
        ΑΝ ΕΙΣ[i] > ΜΟ ΤΟΤΕ
            ΓΡΑΨΕ Ετ[i]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
        ΑΝ ΕΙΣ[i] < ΜΟ ΤΟΤΕ
            ΓΡΑΨΕ Ετ[i]
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Εταιρεία

```

Γέμισμα
παράλληλων
πινάκων

Υπολογισμός
ΜΟ
εισπράξεων

Τυπώνονται
ονόματα
εταιρειών με
εισπράξεις > ΜΟ

Τυπώνονται
ονόματα
εταιρειών με
εισπράξεις
< ΜΟ



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.

Παράδειγμα 2 (ταξινόμηση σε παράλληλους πίνακες)

Να γραφεί πρόγραμμα κατά το οποίο δίνουμε τα ονόματα και τους μέσους όρους 50 μαθητών της Α΄ Λυκείου ενός σχολείου και τυπώνει μια ταξινομημένη κατάσταση, ως προς το μέσο όρο, κατά φθίνουσα σειρά (από τον μεγαλύτερο βαθμό προς τον μικρότερο).

Λύση

Ανάλυση

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

```

ΠΡΟΓΡΑΜΜΑ Μαθητές

ΜΕΤΑΒΛΗΤΕΣ
  ΧΑΡΑΚΤΗΡΕΣ: Ον[50], t2
  ΠΡΑΓΜΑΤΙΚΕΣ: 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
        { Αντιμετάθεση ΜΟ }

        t2 <- Ον[j]
        Ον[j] <- Ον[j - 1]
        Ον[j - 1] <- t2
        { Αντιμετάθεση Ονομάτων }

      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
    ΓΡΑΨΕ Ον[i], ΜΟ[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Μαθητές
  
```

Γέμισμα
παράλληλων
πινάκων {

Ταξινόμηση ως
προς τους ΜΟ {

Εκτύπωση
πινάκων {

Παράδειγμα 3 (Διπλή ταξινόμηση σε παράλληλους πίνακες)

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

Λύση

Ανάλυση

1. Καταχωρούμε σε **δύο παράλληλους πίνακες** τα ονόματα και τους Μέσους Όρους 50 μαθητών.
2. Ταξινομούμε τους πίνακες ως προς τον Μέσο Όρο.
3. Ελέγχουμε μέσα στην ταξινόμηση αν οι Μέσοι Όροι είναι ίσοι και ταξινομούμε ως προς το Όνομα του μαθητή.

ΠΡΟΓΡΑΜΜΑ Διπλή_Ταξινόμηση

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Ον[50], t2
ΠΡΑΓΜΑΤΙΚΕΣ: 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

4.5 Λυμένες ασκήσεις

Άσκηση 1 (Μονοδιάστατος πίνακας)

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

- i. Γέμισμα πίνακα με ακέραιους αριθμούς.
- ii. Εκτύπωση πίνακα από την 1η θέση μέχρι την 100ή θέση.
- iii. Εκτύπωση πίνακα από την 100ή θέση μέχρι την 1η θέση.
- iv. Εκτύπωση του αθροίσματος των στοιχείων του πίνακα.
- v. Εκτύπωση του μέσου όρου των στοιχείων του πίνακα.
- vi. Εκτύπωση του αθροίσματος των στοιχείων του πίνακα που βρίσκονται στις ζυγές θέσεις του πίνακα και του αθροίσματος που βρίσκονται στις μονές θέσεις.
- vii. Εκτύπωση του γινόμενου των 10 πρώτων στοιχείων του πίνακα A.
- viii. Αντιμετάθεση των συμμετρικών στοιχείων του πίνακα A και εκτύπωση του πίνακα A.
- ix. Υπολογισμό του αθροίσματος των 50 πρώτων στοιχείων του πίνακα A και του αθροίσματος των 50 τελευταίων στοιχείων του πίνακα A, αν τα δύο αθροίσματα είναι ίσα να τυπώνεται το μήνυμα «Ίσα αθροίσματα».
- x. Εμφάνιση το μηνύματος «Πίνακας συμμετρικός», αν τα συμμετρικά στοιχεία του πίνακα είναι ίσα.
- xi. Εκτύπωση του πλήθους των στοιχείων που είναι μικρότερα από το Μέσο Όρο τους.
- xii. Αναζήτηση της τιμής «42» και αν βρεθεί να τυπωθεί η θέση της.
- xiii. Ταξινόμηση κατά φθίνουσα σειρά του πίνακα A.
- xiv. Έλεγχο του πίνακα A αν είναι ταξινομημένος κατά φθίνουσα σειρά και να βγάζει κατάλληλο μήνυμα.

Λύση

ΠΡΟΓΡΑΜΜΑ Μονοδιάστατος_πίνακας

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], i, j, sum, sum1, sum2

ΑΚΕΡΑΙΕΣ: sum α , sum μ , K, μ , GIN

ΠΡΑΓΜΑΤΙΚΕΣ: MO

ΛΟΓΙΚΕΣ: flag

ΑΡΧΗ

- i. Γέμισμα πίνακα {
- ```
ΓΡΑΨΕ 'Δώσε 100 ακέραιους αριθμούς:'
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
 ΔΙΑΒΑΣΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```
- ii. {
- ```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```
- iii. {
- ```
ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 1 ΜΕ_ΒΗΜΑ -1
 ΓΡΑΨΕ A[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

iv. {  
 sum <- 0  
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100  
 sum <- sum + A[i]  
 ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  
 ΓΡΑΨΕ 'Αθροισμα στοιχείων:', sum

v. {  
 MO <- sum/100  
 ΓΡΑΨΕ 'Μέσος Όρος στοιχείων =', MO

vi. {  
 sum<sub>α</sub> <- 0  
 sum<sub>μ</sub> <- 0  
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100  
 ΑΝ i mod 2 = 0 ΤΟΤΕ  
 sum<sub>α</sub> <- sum<sub>α</sub> + A[i]  
 ΑΛΛΙΩΣ  
 sum<sub>μ</sub> <- sum<sub>μ</sub> + A[i]  
 ΤΕΛΟΣ\_ΑΝ  
 ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  
 ΓΡΑΨΕ 'Αθροισμα αριθμών σε ζυγές θέσεις:', sum<sub>α</sub>  
 ΓΡΑΨΕ 'Αθροισμα αριθμών σε μονές θέσεις:', sum<sub>μ</sub>

Ελέγχουμε αν η κάθε θέση του πίνακα A[i] είναι άρτιος αριθμός (i mod 2=0) ή όχι και προσθέτουμε το κάθε στοιχείο του πίνακα στον κατάλληλο αθροιστή.

vii. {  
 GIN <- 1  
 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10  
 GIN <- GIN \* A[i]  
 ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ  
 ΓΡΑΨΕ 'Το γινόμενο των 10 πρώτων αριθμών:', GIN

Η μεταβλητή **GIN** είναι ο πολ/στής με αρχική τιμή το 1, το οποίο είναι το ουδέτερο στοιχείο του πολλ/σμού.

viii. {  
 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:

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

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 'Ισα αθροίσματα'  
 ΤΕΛΟΣ\_ΑΝ

Υπολογισμός αθροίσματος των 50 πρώτων αριθμών

Υπολογισμός αθροίσματος των 50 επόμενων αριθμών

x. {

```

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

```

Έλεγχος των:

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

xi. {

```

μ <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
 ΑΝ A[i] < ΜΟ ΤΟΤΕ
 μ <- μ + 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
 Κ <- A[j]
 A[j] <- A[j - 1]
 A[j - 1] <- Κ
 ΤΕΛΟΣ_ΑΝ
 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

xiv. {

```

flag <- ΑΛΗΘΗΣ
ΓΙΑ i ΑΠΟ 100 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ -1
 ΑΝ A[i] > A[i - 1] ΤΟΤΕ
 flag <- ΨΕΥΔΗΣ
 ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ flag = ΑΛΗΘΗΣ ΤΟΤΕ
 ΓΡΑΨΕ 'Πίνακας ταξινομημένος κατά φθίνουσα σειρά'
ΤΕΛΟΣ_ΑΝ

```

Έλεγχος αν ο πίνακας είναι  
ταξινομημένος

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ Μονοδιάστατος\_πίνακας

## Άσκηση 2 (Δισδιάστατος πίνακας)

Έστω δισδιάστατος πίνακας A, 10 x 20, με ακέραιους αριθμούς. Να γραφεί 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 ΑΘΡ\_ΓΡ[20].
- vi. Να υπολογίζει τα αθροίσματα των στηλών του πίνακα A, να τα τυπώνει και να τα καταχωρεί στον πίνακα ΑΘΡ\_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 σειρά.
- xv. Να ταξινομεί τα στοιχεία κάθε στήλης κατά αύξουσα σειρά.

## Λύση

**ΠΡΟΓΡΑΜΜΑ** Δισδιάστατος\_πίνακας

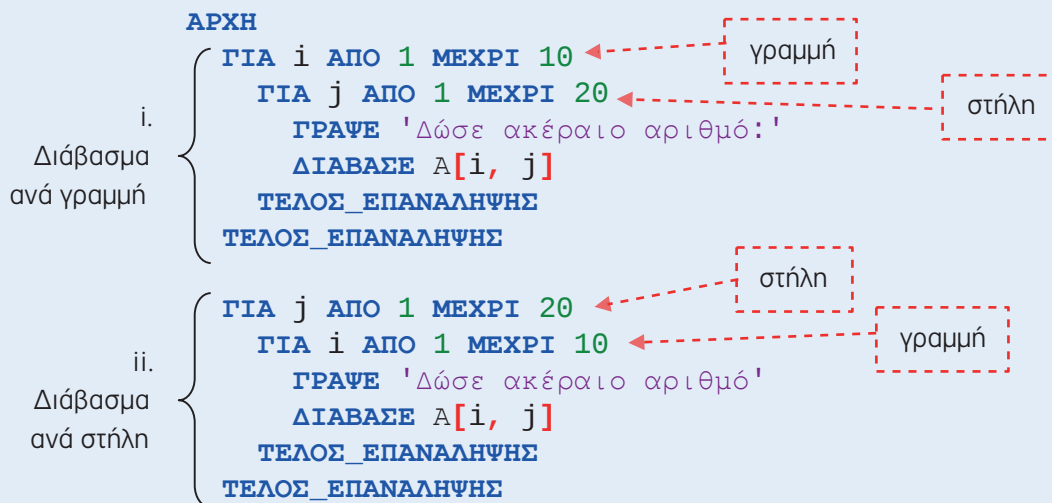
### ΜΕΤΑΒΛΗΤΕΣ

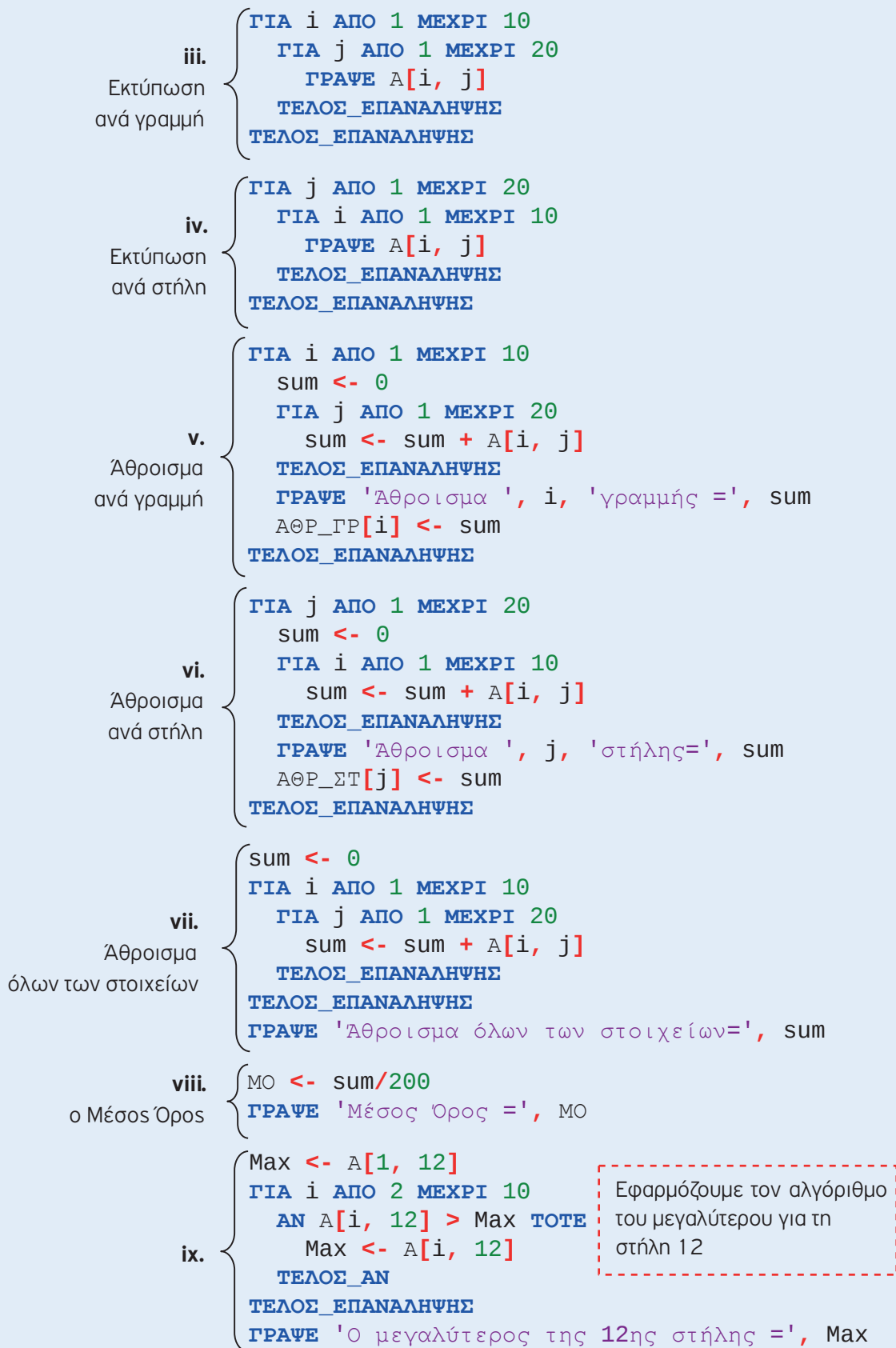
**ΑΚΕΡΑΙΕΣ:** A[10, 20], ΑΘΡ\_ΓΡ[10], Min, Max, st

**ΑΚΕΡΑΙΕΣ:** ΑΘΡ\_ΣΤ[20], i, j, sum, μ, x, gr, K

**ΠΡΑΓΜΑΤΙΚΕΣ:** MO

### ΑΡΧΗ





x. {

```

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

```

Εφαρμόζουμε τον αλγόριθμο του μικρότερου για τη γραμμή 7

xi. {

```

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
ΓΡΑΨΕ Min, Max

```

Εφαρμόζουμε τον αλγόριθμο του μεγαλύτερου και του μικρότερου για τον πίνακα A

xii. {

```

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

```

Μετράμε πόσα στοιχεία είναι μεγαλύτερα του μέσου όρου

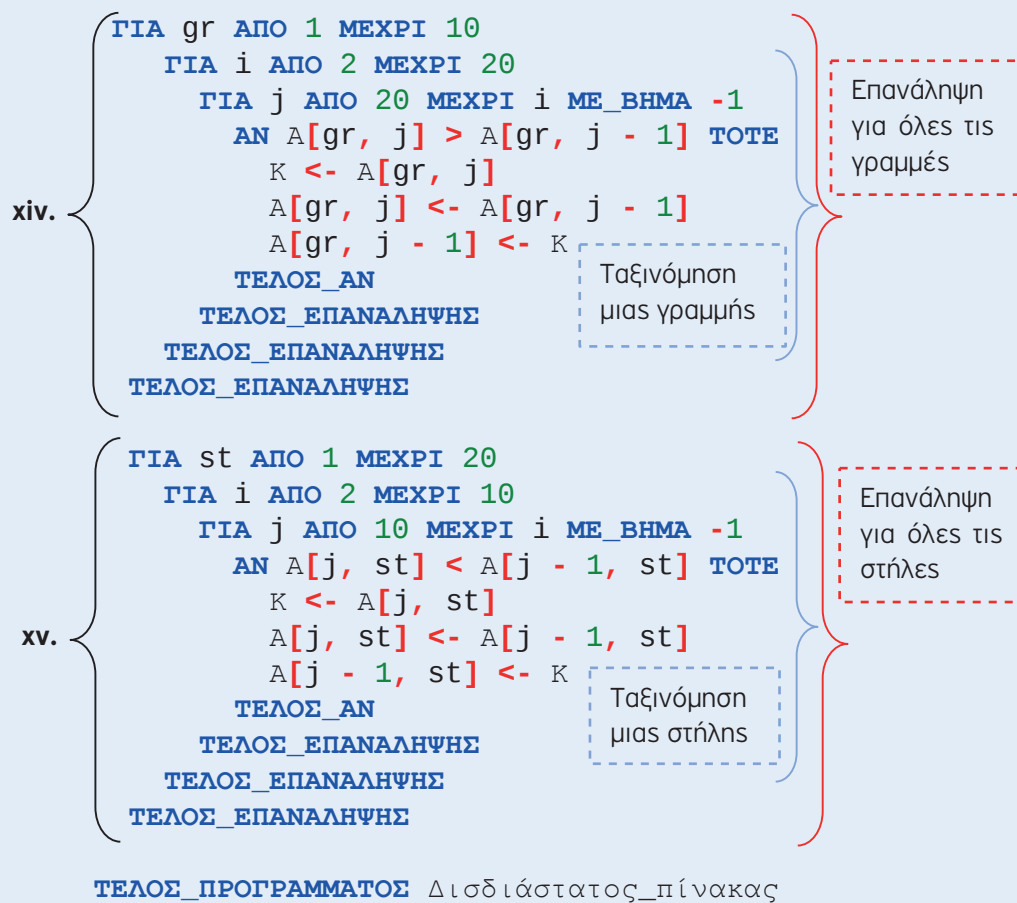
xiii. {

```

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

```

Αναζήτηση στοιχείου στον πίνακα A



### Άσκηση 3

Ένα κτίριο με γραφεία έχει 5 ορόφους με 12 γραφεία ανά όροφο. Να δημιουργηθεί ένας δισδιάστατος πίνακας  $5 \times 12$ , 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 γραφεί πρόγραμμα που:

- Να διαβάξει το πλήθος των υπαλλήλων κάθε γραφείου ανά όροφο και να τους καταχωρεί στον πίνακα  $KT\_GP[5, 12]$ .
- Να γεμίζει, να υπολογίζει και να εμφανίζει τον αριθμό και τον όροφο του γραφείου με τους περισσότερους και τους λιγότερους υπαλλήλους (θεωρήστε ότι είναι μοναδικοί).
- Να υπολογίζει και να τυπώνει το πλήθος των υπαλλήλων ανά όροφο.
- Να υπολογίζει και να τυπώνει το σύνολο των υπαλλήλων όλων των γραφείων.

### Λύση

#### Ανάλυση

Οι πίνακες που θα δημιουργήσουμε και οι μεταξύ τους σχέσεις:

|        |        | Γραφεία |    |    |    |    |    |   |    |   |    |    |    |    |                 |   |     |
|--------|--------|---------|----|----|----|----|----|---|----|---|----|----|----|----|-----------------|---|-----|
|        |        | 1       | 2  | 3  | 4  | 5  | 6  | 7 | 8  | 9 | 10 | 11 | 12 |    |                 |   |     |
| KT_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  | <b>AΘ_OP =</b>  | 4 | 76  |
|        |        | 5       | 5  | 2  | 11 | 12 | 21 | 5 | 9  | 3 | 2  | 4  | 3  | 10 |                 | 5 | 87  |

- Γεμίζουμε τον 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 ο μικρότερος και ο μεγαλύτερος αριθμός υπαλλήλων αντίστοιχα.
- Εφαρμόζουμε άθροισμα ανά γραμμή δημιουργώντας έναν μονοδιάστατο και παράλληλο πίνακα, ως προς τις γραμμές, γιατί ο κάθε όροφος είναι μια γραμμή πίνακα.
- Αθροίζουμε με τη βοήθεια αθροιστή όλα τα στοιχεία του μονοδιάστατου πίνακα που δημιουργήσαμε παραπάνω.

**ΠΡΟΓΡΑΜΜΑ** Κτίριο\_Γραφείων

**ΜΕΤΑΒΛΗΤΕΣ**

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

**ΑΚΕΡΑΙΕΣ:** Min\_i, Min\_j, Max\_i, Max\_j, sum

**ΑΡΧΗ**

i. { **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**

Γέμισμα πίνακα ανά γραμμή

ii. { **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  
    **ΓΡΑΨΕ** 'έχει τους λιγότερους υπαλλήλους'  
    **ΓΡΑΨΕ** 'Το γραφείο', Max\_j, 'στον όροφο', Max\_i  
    **ΓΡΑΨΕ** 'έχει τους περισσότερους υπαλλήλους'

Υπολογισμός μικρότερου και μεγαλύτερου σε δισδιάστατο πίνακα

iii. {

```

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
ΓΡΑΨΕ 'ορόφου είναι:', sum
ΑΘ_ΟΡ[i] <- sum
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Άθροισμα ανά γραμμή και καταχώριση σε παράλληλο, ως προς τις γραμμές, πίνακα

iv. {

```

sum <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
 sum <- sum + ΑΘ_ΟΡ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το σύνολο όλων των υπαλλήλων, είναι:', sum

```

Άθροισμα στοιχείων μονοδιάστατου πίνακα

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ Κτίριο\_Γραφείων

## 4.6 Μη λυμένες ασκήσεις

1. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει τις ηλικίες και τα ονόματα 100 ανθρώπων. Τα δεδομένα να καταχωρούνται σε πίνακες, κάνοντας έλεγχο εγκυρότητας για την ηλικία, η οποία πρέπει να είναι θετικός αριθμός. Κατόπιν το πρόγραμμα να υπολογίζει και να τυπώνει:

- 1) Τον μέσο όρο των ηλικιών.
- 2) Τη μέγιστη ηλικία καθώς και πόσοι και ποιοί την έχουν.
- 3) Το πλήθος και τα ονόματα των ανθρώπων που είναι άνω των 50 ετών.

2. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει για τους 176 υπαλλήλους μιας επιχείρησης τα ονόματα, το έτος πρόσληψης, τα έτη προϋπηρεσίας σε άλλες επιχειρήσεις και τα στοιχεία αυτά να αποθηκεύονται στους πίνακες ΟΝΟΜΑΤΑ, ΕΤΟΣ\_ΠΡΟΣΛΗΨΗΣ και ΕΤΗ\_ΥΠΗΡΕΣΙΑΣ αντίστοιχα. Στη συνέχεια να:

- 1) Εμφανίζει τα ονόματα των υπαλλήλων που θα συνταξιοδοτηθούν την επόμενη πενταετία καθώς και το πλήθος τους. Σύνταξη δίνεται στα 40 έτη συνολικής υπηρεσίας και άνω.
- 2) Δημιουργηθεί πίνακας με το όνομα ΝΕΟΙ\_ΕΡΓΑΖΟΜΕΝΟΙ, με τα ονόματα των εργαζομένων που έχουν συνολικά έτη υπηρεσίας (μαζί και η προϋπηρεσία τους) λιγότερα από 5 και στη συνέχεια να τα εμφανίζει.

3. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάζει 100 πραγματικούς αριθμούς και να τους αποθηκεύει σε μονοδιάστο πίνακα. Στη συνέχεια να κάνει αντιστροφή των στοιχείων του πίνακα:

- 1) Με τη χρήση 2<sup>ου</sup> πίνακα
- 2) Χωρίς τη χρήση 2<sup>ου</sup> πίνακα.

**4.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο διαβάξει τη βαθμολογία ενός τμήματος 26 μαθητών στο μάθημα της Πληροφορικής και την αποθηκεύει σε μονοδιάστατο πίνακα. Στη συνέχεια να υπολογίζει και να εμφανίζει:

- 1) Τον μέσο όρο του τμήματος.
- 2) Την απόκλιση από τον μέσο όρο για κάθε μαθητή
- 3) Το πλήθος των μαθητών με βαθμό μεγαλύτερο του μέσου όρου.

**5.** Στον κεντρικό υπολογιστή του τμήματος μηχανογράφησης ενός ασφαλιστικού ταμείου καταχωρούνται οι ασφαλισμένοι με όλα τα απαραίτητα στοιχεία τους. Για τον σκοπό αυτό χρησιμοποιούνται διαφορετικοί πίνακες, ο πίνακας ΟΝΟΜΑ και ο πίνακας ΕΤΟΣ, που περιέχουν για κάθε ασφαλισμένο το όνομα και το έτος γέννησής του, αντίστοιχα. Γνωρίζοντας ότι κάποιος ασφαλισμένος συνταξιοδοτείται μόλις συμπληρώσει το 67ο έτος της ηλικίας του, να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να:

- 1) Διαβάξει το τρέχον έτος
- 2) Διαβάξει τα ονόματα και τις χρονολογίες γέννησης 100 ασφαλισμένων.
- 3) Εμφανίζει τα ονόματα και το πλήθος εκείνων που θα συνταξιοδοτηθούν σε λιγότερο από 10 χρόνια.

**6.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάξει 100 πραγματικούς αριθμούς και να τους καταχωρεί στον πίνακα A[100]. Στη συνέχεια να εμφανίζει όλες τις τριάδες αριθμών όπου ο μεσαίος αριθμός ισούται με το άθροισμα των άλλων δύο.

**7.** Σε μία πολιτιστική λέσχη τα μέλη της μπορούν να επιλέξουν για να παίξουν Σκάκι (Σ), Μπριτζ (Μ) ή Τάβλι (Τ). Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να διαβάξει τις προτιμήσεις 100 μελών του και στη συνέχεια να υπολογίζει και να τυπώνει το παιχνίδι που είναι η προτίμηση των περισσοτέρων.

Να γίνεται έλεγχος εγκυρότητας των τιμών που δίνονται από το πληκτρολόγιο.

**8.** Μια τράπεζα διαχειρίζεται τους λογαριασμούς πελατών της χρησιμοποιώντας 2 πίνακες. Τον πίνακα ΟΝΟΜΑ που περιέχει τα ονοματεπώνυμα των πελατών της και τον πίνακα ΛΟΓΑΡΙΑΣΜΟΣ που περιέχει τα υπόλοιπα των λογαριασμών τους. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να:

- 1) Διαβάξει τον αριθμό των πελατών της τράπεζας, ελέγχοντας ότι είναι θετικός αριθμός με ανώτατο όριο 10.000.000. Στη περίπτωση που δοθεί λάθος νούμερο, να τυπώνεται το μήνυμα "Μη αποδεκτός αριθμός... Ξαναπροσπαθήστε...." και να επαναλαμβάνεται η διαδικασία μέχρι αποδεκτής τιμής.
- 2) Διαβάξει τα Ονοματεπώνυμο και υπόλοιπο λογαριασμού για κάθε πελάτη της τράπεζας, καταχωρώντας τα αντίστοιχα στους πίνακες ΟΝΟΜΑ και ΛΟΓΑΡΙΑΣΜΟΣ.
- 3) Διαβάξει το ονοματεπώνυμο, την προτίμηση ενός πελάτη για κατάθεση ή ανάληψη χρημάτων (Κ= κατάθεση, Α= ανάληψη) καθώς και το ποσό. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να δοθεί ως ονοματεπώνυμο πελάτη η λέξη "ΤΕΛΟΣ".
- 4) Για κάθε πελάτη που διαβάξει να κάνει τα εξής:
  - i. Να ελέγχει, στην περίπτωση ανάληψης, αν αυτή μπορεί να γίνει και να εμφανίζει το μήνυμα "Δεν επαρκεί το υπόλοιπό σας...." στην περίπτωση που δεν είναι εφικτή.
  - ii. Σε κάθε περίπτωση (κατάθεση ή ανάληψη) να ενημερώνεται ο πίνακας με τον λογαριασμό του πελάτη ανάλογα σε κάθε περίπτωση.
  - iii. Να εμφανίζει το νέο υπόλοιπο λογαριασμού.
- 5) Να εμφανίζει το σύνολο των καταθέσεων που έγιναν.
- 6) Να εμφανίζει το σύνολο των αναλήψεων που έγιναν.

**9.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να δημιουργεί και να εμφανίζει τους παρακάτω πίνακες

Πίνακας 1

|   |    |    |    |    |
|---|----|----|----|----|
| 1 | 6  | 11 | 16 | 21 |
| 2 | 7  | 12 | 17 | 22 |
| 3 | 8  | 13 | 18 | 23 |
| 4 | 9  | 14 | 19 | 24 |
| 5 | 10 | 15 | 20 | 25 |

Πίνακας 2

|    |    |    |    |    |
|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  |
| 6  | 7  | 8  | 9  | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

**10.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να:

- 1) Γεμίζει με ακέραιες τιμές από το πληκτρολόγιο έναν δισδιάστατο πίνακα 20x7
- 2) Υπολογίζει και να τυπώνει τη γραμμή με το μεγαλύτερο άθροισμα.
- 3) Υπολογίζει και να τυπώνει τη στήλη με το μικρότερο άθροισμα.

**11.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να:

- 1) Γεμίζει έναν πίνακα 5x5 με ακέραιους αριθμούς, που δίνονται από το πληκτρολόγιο.
- 2) Υπολογίζει και να εμφανίζει:
  - i. Το μέγιστο στοιχείο της κύριας και της δευτερεύουσας διαγωνίου.
  - ii. Τον μέσο όρο των στοιχείων της κύριας και της δευτερεύουσας διαγωνίου.

**12.** Ένα σχολείο έχει 9 τμήματα με 26 μαθητές το κάθε τμήμα. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να:

- 1) Καταχωρεί τους μέσους όρους των μαθητών ανά τμήμα στο πίνακα A[9,26].
- 2) Υπολογίζει και να εμφανίζει τον μέσο όρο κάθε τμήματος.
- 3) Υπολογίζει και να εμφανίζει το πλήθος και το ποσοστό των αριστούχων ανά τμήμα. Αριστούχος θεωρείται αυτός που έχει μέσο όρο μεγαλύτερο από 18.5.
- 4) Υπολογίζει και να εμφανίζει το μεγαλύτερο μέσο όρο σε όλο το σχολείο.

**13.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο σε ένα μονοδιάστατο πίνακα ΒΑΘΜΟΙ[52] καταχωρεί τους βαθμούς του μαθήματος Πληροφορικής, των μαθητών της Γ΄ Λυκείου ενός σχολείου. Οι βαθμοί θεωρούνται θετικοί και ακέραιοι. Το πρόγραμμα να υπολογίζει και να τυπώνει τη συχνότητα που εμφανίζεται ο κάθε βαθμός, αν θεωρήσουμε ότι όλοι οι βαθμοί είναι από το 1 μέχρι το 20.

**14.** Ένα εμπορικό κατάστημα εμπορεύεται 25 μάρκες και από κάθε μάρκα 10 διαφορετικά είδη. Σε ένα δισδιάστατο πίνακα 25x10, το κατάστημα αποθηκεύει τις τιμές του κάθε είδους από την κάθε μάρκα. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο :

- 1) Να διαβάζει και να αποθηκεύει σ' έναν μονοδιάστατο πίνακα ΜΑΡΚΑ[10], τα ονόματα των μαρκών.
- 2) Να διαβάζει και να αποθηκεύει τα ονόματα των ειδών που εμπορεύεται στον πίνακα ΕΙΔΗ[25].
- 3) Να διαβάζει τη τιμή κάθε είδους ανά μάρκα και να τη καταχωρεί στον πίνακα ΤΙΜΗ[25,10].
- 4) Να δίνεται από το πληκτρολόγιο ένα είδος και η μάρκα του και να τυπώνεται η αντίστοιχη τιμή του.

**15.** Σ' έναν διαγωνισμό ταλέντων διαγωνίστηκαν 250 υποψήφιοι. Η βαθμολογία τους είναι στο διάστημα [1,100]. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο :

- 1) Να καταχωρεί σε πίνακες τα ονόματα και τους βαθμούς των υποψηφίων.
- 2) Να εμφανίζει μια λίστα με τους 10 πρώτους, ταξινομημένους κατά φθίνουσα σειρά ως προς τη βαθμολογία τους.

**16.** Η Γ΄ τάξη έχει 4 τμήματα με 25 μαθητές σε κάθε τάξη.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο :

- 1) Να διαβάζει τους βαθμούς των μαθητών στο μάθημα της Πληροφορικής και να τους καταχωρεί σε έναν πίνακα Β[25,4].
- 2) Να υπολογίζει και να εμφανίζει τον Μέσο όρο ανά τμήμα.
- 3) Να υπολογίζει και να εμφανίζει το τμήμα με τον μεγαλύτερο Μέσο όρο.
- 4) Να εμφανίζει τον μεγαλύτερο βαθμό και σε ποιο τμήμα βρίσκεται.
- 5) Να υπολογίζει και να εμφανίζει τον Μέσο Όρο του μαθήματος σε όλη την Γ΄ Λυκείου.

**17.** Σε πίνακα ακεραίων ΛΟΤΤΟ[1000000,6] βρίσκονται οι 6 αριθμοί του ΛΟΤΤΟ που έχουν παίξει 1.000.000 παίκτες για το παιχνίδι του ΛΟΤΤΟ και στον πίνακα ΠΑΙΧΤΗΣ[1000000] είναι τα ονόματα των παικτών. Στον πίνακα ΚΕΡΔΙΣΕ[6] είναι οι 6 τυχεροί αριθμοί του ΛΟΤΤΟ που κληρώθηκαν.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο :

- 1) Να γεμίζει τους παραπάνω πίνακες με τιμές από το πληκτρολόγιο.
- 2) Να κάνει την τελική διαλογή -πόσοι παίκτες είχαν 6 επιτυχίες, πόσοι είχαν 5 επιτυχίες, πόσοι είχαν 4 επιτυχίες, πόσοι είχαν 3 επιτυχίες, πόσοι είχαν 2 επιτυχίες και πόσοι είχαν 1 επιτυχία- και να τις αποθηκεύει στον πίνακα ΔΙΑΛΟΓΗ[6].
- 3) Να εμφανίζει τα ονόματα των παικτών που είχαν 6 επιτυχίες.

**18.** Για τις ανάγκες μιας έρευνας δημιουργήθηκαν 3 μοναδιάστατοι πίνακες, όπου ο 1ος περιέχει το ονοματεπώνυμο, ο 2ος το φύλο και ο 3ος τον μισθό για τους 245 υπαλλήλους μιας εταιρείας.

Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να υπολογίζει και να εμφανίζει:

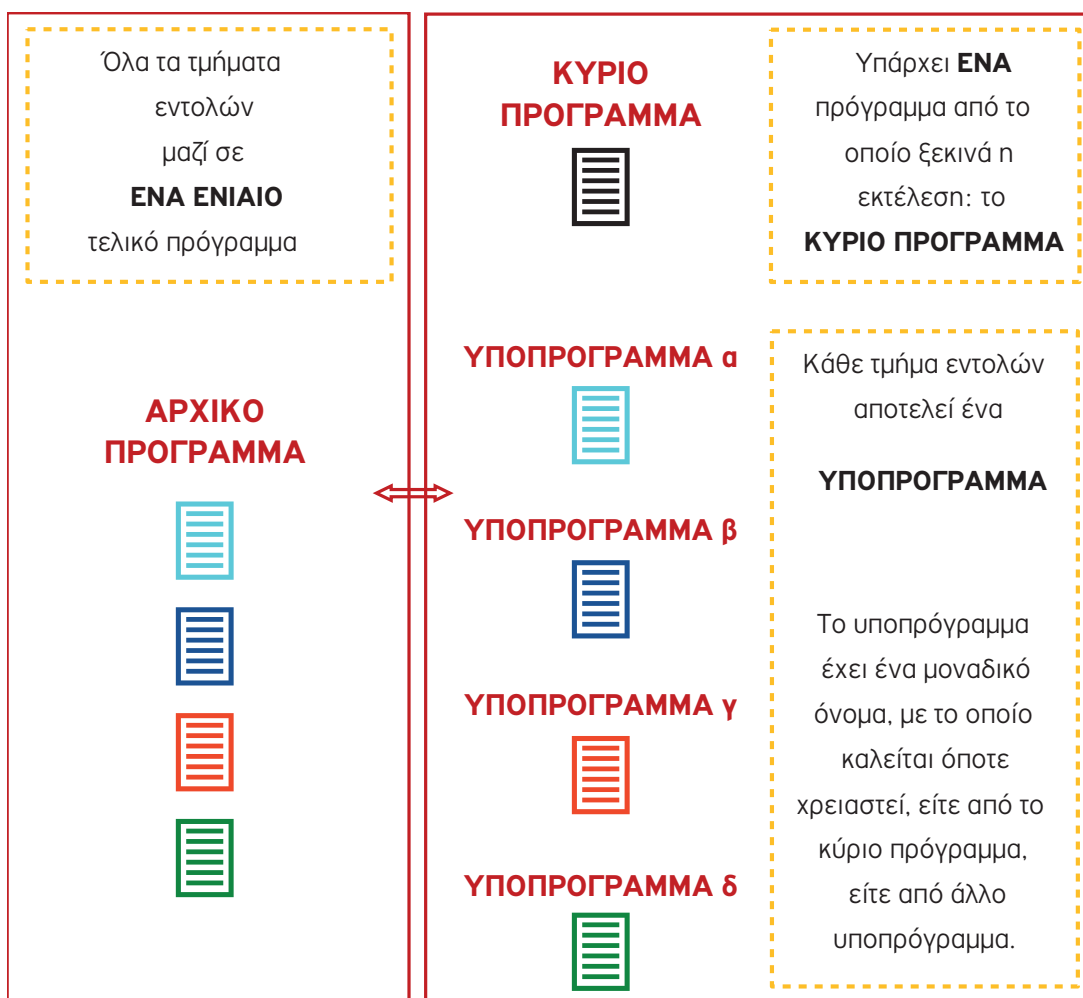
- 1) Πόσοι είναι οι υπάλληλοι με μισθό μεγαλύτερο από 560€.
- 2) Ποιο είναι το % ποσοστό των ανδρών με μισθό μεγαλύτερο του μέσου όρου του μισθού των ανδρών.
- 3) Ποιες είναι οι 10 πιο καλοπληρωμένες γυναίκες.

**19.** Στο αγώνισμα του άλματος εις μήκος συμμετέχουν 12 αθλητές, οι οποίοι κάνουν 6 προσπάθειες ο καθένας. Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ», το οποίο να:

- 1) Διαβάζει και να αποθηκεύει τα ονόματα 12 αθλητών σε μονοδιάστατο πίνακα.
- 2) Διαβάζει και να αποθηκεύει σε δισδιάστατο πίνακα τις επιδόσεις του κάθε αθλητή στις 6 προσπάθειες που έκανε. Σε άκυρη προσπάθεια δίνεται το μηδέν.
- 3) Εμφανίζει για κάθε αθλητή το όνομά του και τις επιδόσεις του ταξινομημένες, από την καλύτερη προς την χειρότερη, εκτός από τις άκυρες.
- 4) Εμφανίζει το όνομα του νικητή του αγωνίσματος άλματος εις μήκος. Νικητής θεωρείται αυτός με την μεγαλύτερη επίδοση. Σε περίπτωση ισοβαθμίας, νικητής θεωρείται αυτός που έχει τη μεγαλύτερη 2η επίδοση.

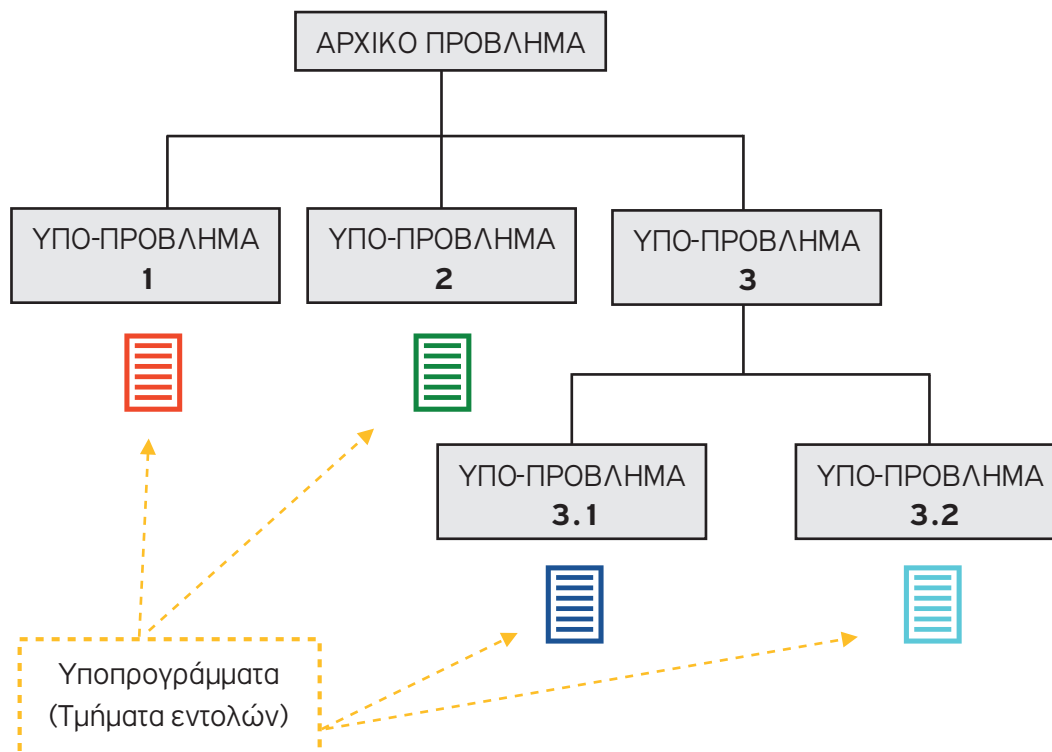
## 5. ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ - ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ (ΔΙΑΔΙΚΑΣΙΕΣ-ΣΥΝΑΡΤΗΣΕΙΣ)

**Τμηματικός προγραμματισμός** είναι η τεχνική σχεδίασης των προγραμμάτων κατά την οποία ένα πρόγραμμα 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 τμήματα-προγράμματα ονομάζονται υποπρογράμματα.



### Πού χρησιμοποιούνται τα υποπρογράμματα;

Τα υποπρογράμματα χρησιμοποιούνται στην περίπτωση όπου έχουμε να επιλύσουμε ένα σύνθετο και πολύπλοκο πρόβλημα με τη χρήση υπολογιστή. 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. Με τον τρόπο αυτό είναι δυνατό να αντιμετωπισθεί πιο εύκολα η επίλυση του αρχικού σύνθετου προβλήματος.



### Παράδειγμα 1

Ένα πρόγραμμα εύρεσης και εμφάνισης του μέσου όρου βαθμολογίας μιας τάξης πώς θα μπορούσε να διαιρεθεί κάνοντας χρήση υποπρογραμμάτων;

### Λύση

Ένα πρόγραμμα εύρεσης και εμφάνισης του μέσου όρου βαθμολογίας μιας τάξης θα μπορούσε να διαιρεθεί στο κύριο πρόγραμμα, το οποίο 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).

### Τι πρέπει να προσέχουμε;



#### 1. Τα **υποπρογράμματα** χαρακτηρίζονται από τις παρακάτω **ιδιότητες**:

- Κάθε υποπρόγραμμα έχει μία μόνο είσοδο και μία 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 κληθεί 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 μεταβλητές σαν κανάλια επικοινωνίας, οι οποίες λέγονται παράμετροι. Δηλαδή, παράμετρος είναι μία 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 αντιστοίχισή τους γίνεται με την σειρά που είναι γραμμένες) και να είναι του ιδίου 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 δηλώσεων του, αποδεσμεύεται.



## 4. **Διαφορές 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:

- οι Διαδικασίες, και
- οι Συναρτήσεις.

## 5.1 Διαδικασίες (Παράδειγμα διαδικασίας)

### Πρόβλημα

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.

### Λύση

#### Ανάλυση προβλήματος

Το πρόβλημα χρειάζεται να αναλυθεί σε υποπροβλήματα.

Για διευκόλυνση της κατανόησης της έννοιας των διαδικασιών (υποπρογραμμάτων), ακολουθείται η παρακάτω τεχνική που βασίζεται σε ήδη γνωστές έννοιες.

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.

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

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:

**ΠΡΟΓΡΑΜΜΑ** Ανταλλαγή

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\alpha, \beta, \iota, \kappa$

**ΑΡΧΗ**

**ΓΡΑΨΕ** 'Δωσε τιμές για  $\alpha$  και  $\beta$ '

**ΔΙΑΒΑΣΕ**  $\alpha, \beta$

Κλήση\_Διαδικασίας ( $\pi 1, \dots$ )

**ΓΡΑΨΕ** ' $\alpha =$ ',  $\alpha$ , ' $\beta =$ ',  $\beta$

**ΓΡΑΨΕ** 'Δωσε τιμές για  $\iota$  και  $\kappa$ '

**ΔΙΑΒΑΣΕ**  $\iota, \kappa$

Κλήση\_Διαδικασίας ( $\pi 1, \dots$ )

**ΓΡΑΨΕ** ' $\iota =$ ',  $\iota$ , ' $\kappa =$ ',  $\kappa$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** Ανταλλαγή

**ΔΙΑΔΙΚΑΣΙΑ** Εναλλαγή\_Τιμών ( $\pi 1, \dots$ )

.....

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ** Εναλλαγή\_Τιμών

### Λύση του Προβλήματος

**ΠΡΟΓΡΑΜΜΑ** Ανταλλαγή

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\alpha, \beta, \iota, \kappa$

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $\alpha, \beta$

**ΚΑΛΕΣΕ** Εναλλαγή\_τιμών( $\alpha, \beta$ )

**ΓΡΑΨΕ**  $\alpha, \beta$

**ΔΙΑΒΑΣΕ**  $\iota, \kappa$

**ΚΑΛΕΣΕ** Εναλλαγή\_τιμών( $\iota, \kappa$ )

**ΓΡΑΨΕ**  $\iota, \kappa$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** Ανταλλαγή

**Κύριο πρόγραμμα**, που εκτελεί δύο φορές κατά σειρά τις εξής λειτουργίες:

- Διαβάζει δύο ακέραιους αριθμούς (σε δύο μεταβλητές)
- Εναλλάσσει τις τιμές μεταξύ των μεταβλητών (διαδοχική κλήση διαδικασίας).
- Εμφανίζει το περιεχόμενο των δύο μεταβλητών.

**ΔΙΑΔΙΚΑΣΙΑ** Εναλλαγή\_τιμών( $\kappa, \lambda$ )

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\kappa, \lambda, temp$

**ΑΡΧΗ**

$temp \leftarrow \kappa$

$\kappa \leftarrow \lambda$

$\lambda \leftarrow temp$

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ** Εναλλαγή\_τιμών

**Διαδικασία** που εναλλάσσει τις τιμές δύο μεταβλητών ( $\kappa, \lambda$ ) μέσω της βοηθητικής μεταβλητής  $temp$ .

## Λεπτομερής ανάλυση της ανωτέρω λύσης του προβλήματος:

### 1. Σύνταξη (ορισμός) διαδικασίας

```
ΔΙΑΔΙΚΑΣΙΑ <όνομα_διαδικασίας> (<λίστα_παραμέτρων>);
ΜΕΤΑΒΛΗΤΕΣ
 <δήλωση μεταβλητών>
ΑΡΧΗ
 <εντολές>
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ <όνομα_διαδικασίας>
```

↓

```
ΔΙΑΔΙΚΑΣΙΑ Εναλλαγή_τιμών (κ, λ)
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: κ, λ, temp
ΑΡΧΗ
 temp <- κ
 κ <- λ
 λ <- temp
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Εναλλαγή_τιμών
```

### 2. Κλήση μιας Διαδικασίας από το Κύριο Πρόγραμμα

Πάντοτε έχουμε μόνο ένα ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ, ενώ μπορούμε να έχουμε όσες ΔΙΑΔΙΚΑΣΙΕΣ 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.

```
ΠΡΟΓΡΑΜΜΑ Ανταλλαγή
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: α, β, ι, κ
ΑΡΧΗ
 ΔΙΑΒΑΣΕ α, β
 ΚΑΛΕΣΕ Εναλλαγή_τιμών (α, β)
 ΓΡΑΨΕ α, β

 ΔΙΑΒΑΣΕ ι, κ
 ΚΑΛΕΣΕ Εναλλαγή_τιμών (ι, κ)
 ΓΡΑΨΕ ι, κ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ανταλλαγή
```

Κλήση της διαδικασίας  
**Εναλλαγή\_τιμών**, με δύο  
πραγματικές παραμέτρους,  
την α και τη β

### 3. Επικοινωνία κύριου προγράμματος με τη διαδικασία και εκτέλεση του προγράμματος

Για την κατανόηση της λειτουργίας, θα εκτελεστεί το πρόγραμμα 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.

#### 3.1 Προσδιορισμός των πραγματικών παραμέτρων

Οι πραγματικές παράμετροι είναι δύο (α και β). Επομένως, και οι τυπικές παράμετροι της διαδικασίας θα είναι δύο (κ και λ).

Σύμφωνα με τη σειρά που είναι γραμμένες, η **α** αντιστοιχεί στην **κ** και η **β** αντιστοιχεί στη **λ**.

#### 3.2 Προσδιορισμός των τυπικών παραμέτρων

**ΠΡΟΓΡΑΜΜΑ** Ανταλλαγή

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** α, β, ι, κ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** α, β

**ΚΑΛΕΣΕ** Εναλλαγή\_τιμών(**α, β**)

**ΓΡΑΨΕ** α, β

**ΔΙΑΒΑΣΕ** ι, κ

**ΚΑΛΕΣΕ** Εναλλαγή\_τιμών(**ι, κ**)

**ΓΡΑΨΕ** ι, κ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** Ανταλλαγή

**ΔΙΑΔΙΚΑΣΙΑ** Εναλλαγή\_τιμών(**κ, λ**)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** κ, λ, temp

**ΑΡΧΗ**

temp <- κ

κ <- λ

λ <- temp

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ** Εναλλαγή\_τιμών

Επικοινωνία ΚΥΡΙΟΥ  
ΠΡΟΓΡΑΜΜΑΤΟΣ  
και ΔΙΑΔΙΚΑΣΙΑΣ

### 3.3 Προσδιορισμός των υπόλοιπων μεταβλητών του κύριου προγράμματος και των διαδικασιών

**ΠΡΟΓΡΑΜΜΑ** Ανταλλαγή

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** α, β, ι, κ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** α, β

**ΚΑΛΕΣΕ** Εναλλαγή\_τιμών(α β)

**ΓΡΑΨΕ** α, β

**ΔΙΑΒΑΣΕ** ι, κ

**ΚΑΛΕΣΕ** Εναλλαγή\_τιμών(ι, κ)

**ΓΡΑΨΕ** ι, κ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** Ανταλλαγή

Δεν υπάρχουν  
υπόλοιπες μεταβλητές  
στο ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ

**ΔΙΑΔΙΚΑΣΙΑ** Εναλλαγή\_τιμών(κ, λ)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** κ, λ, temp

**ΑΡΧΗ**

temp <- κ

κ <- λ

λ <- temp

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ** Εναλλαγή\_τιμών

Υπόλοιπες μεταβλητές  
ΔΙΑΔΙΚΑΣΙΑΣ

### 3.4 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.

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.

Οι τιμές των μεταβλητών και παραμέτρων πριν την πρώτη κλήση της διαδικασίας, φαίνονται στο παρακάτω σχήμα. Κατά την κλήση της διαδικασίας, γίνεται αντιγραφή (μεταβίβαση) των τιμών των πραγματικών παραμέτρων στις αντίστοιχες τυπικές.

## ΠΕΡΑΣΜΑ ΠΑΡΑΜΕΤΡΩΝ ΚΑΤΑ ΤΗΝ ΚΛΗΣΗ ΔΙΑΔΙΚΑΣΙΩΝ

### 1<sup>η</sup> Κλήση Διαδικασίας

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

Κατάσταση πριν την πρώτη κλήση

|   |   |
|---|---|
| κ | 7 |
| λ | 3 |

Στη διαδικασία γίνεται ανταλλαγή τιμών

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

Μεταβίβαση των μεταβλητών α και β στις κ και λ αντίστοιχα

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

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

- 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 τοπική ισχύ (εντός του προγράμματος-υποπρογράμματος που δηλώνονται). Ο υπολογιστής διαθέτει διαφορετικό χώρο μνήμης για τις μεταβλητές της διαδικασίας από αυτές του κυρίου προγράμματος.
- Έπειτα από 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 (εντολή που κάλεσε τη διαδικασία ή αλλιώς σημείο κλήσης της διαδικασίας) και συνεχίζεται η εκτέλεση από την επόμενη εντολή. Μέχρι εδώ έχουν ανταλλαγή οι τιμές των μεταβλητών α και β. Στη συνέχεια εκτελείται η εντολή ΓΡΑΨΕ και τυπώνονται οι τιμές των α, και β. Έπειτα εκτελείται το δεύτερο ΔΙΑΒΑΣΕ του κυρίου προγράμματος και φθάνουμε στην εντολή κλήσης της διαδικασίας για δεύτερη φορά με τις τιμές των μεταβλητών και παραμέτρων να διαμορφώνονται όπως δείχνει το παρακάτω σχήμα.

## 2<sup>η</sup> Κλήση Διαδικασίας

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

Κατάσταση πριν τη δεύτερη κλήση

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

→

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

→

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

Στη διαδικασία γίνεται ανταλλαγή τιμών

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

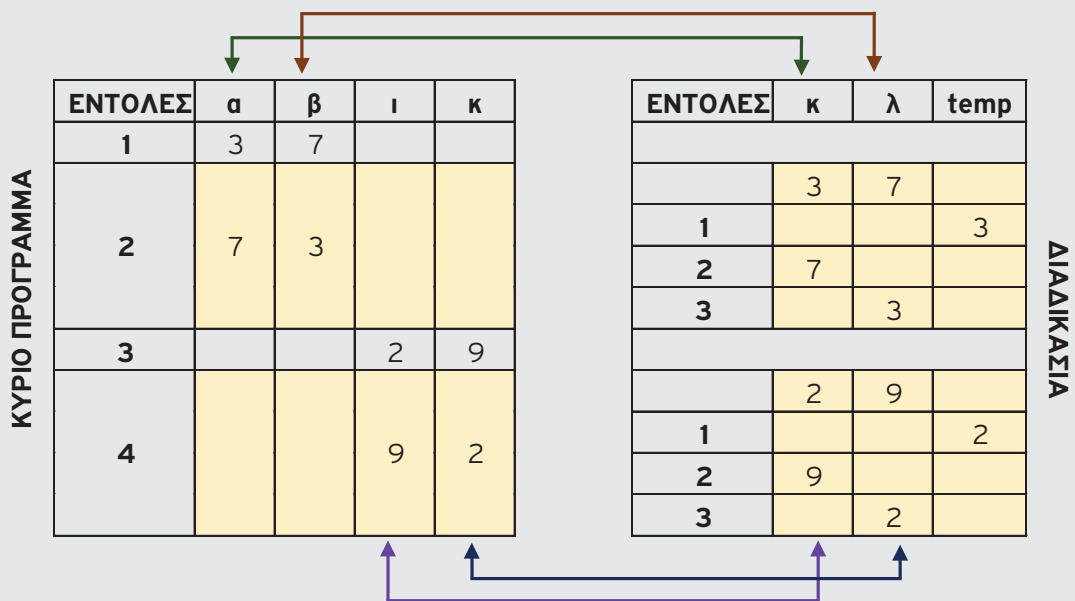
←

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

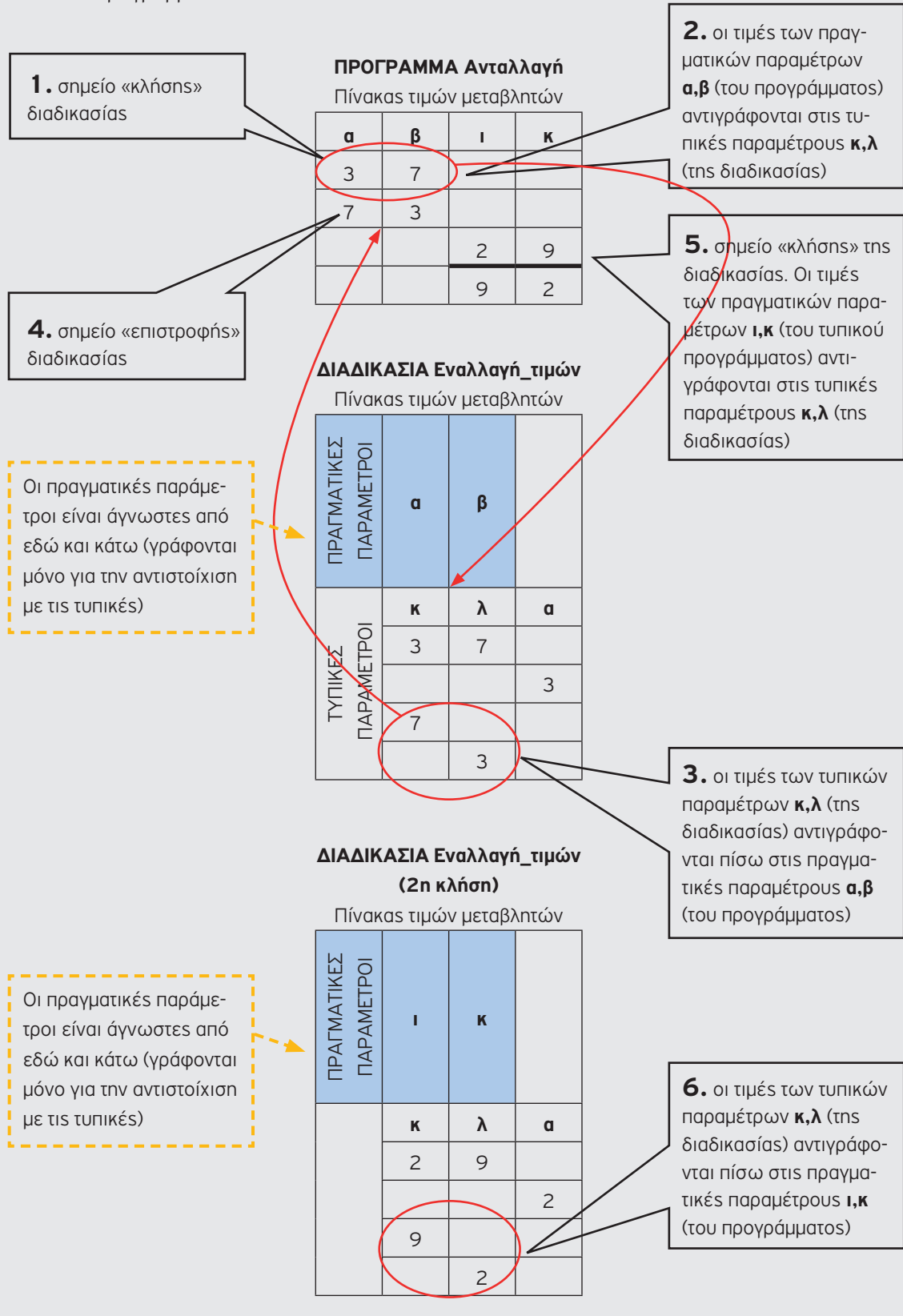
←

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.



Στο παρακάτω σχήμα, ακολουθώντας τα αριθμημένα βήματα, γίνεται καλύτερα αντιληπτή η εκτέλεση του προγράμματος.



Στην οθόνη του Η/Υ θα εμφανιστεί το παρακάτω μήνυμα:

#### ΜΟΝΑΔΑ ΕΞΟΔΟΥ - ΟΘΟΝΗ

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



#### Ιδιαίτερη προσοχή χρειάζονται τα παρακάτω σημεία:

- **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.
- **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 ανωτέρω σημεία προσοχής αναδεικνύονται στο παρακάτω παράδειγμα προγράμματος, το οποίο υπολογίζει και τυπώνει το άθροισμα και τη διαφορά δύο δοθέντων αριθμών με διαδικασία που υπολογίζει το άθροισμα και τη διαφορά τους.

ΠΡΟΓΡΑΜΜΑ Π1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: χ, ψ, Αθρ1, Διαφ1

ΑΡΧΗ

χ <- -10

ψ <- 97

ΚΑΛΕΣΕ Προσθ\_Αφαίρ(χ, ψ, Αθρ1, Διαφ1)

ΓΡΑΨΕ 'Αθροισμα=', Αθρ1

ΓΡΑΨΕ 'Διαφορά=', Διαφ1

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ Π1

ΔΙΑΔΙΚΑΣΙΑ Προσθ\_Αφαίρ(Α, Β, Αθρ2, Διαφ2)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α, Β, Αθρ2, Διαφ2

ΑΡΧΗ

Αθρ2 <- Α + Β

Διαφ2 <- Α - Β

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ Προσθ\_Αφαίρ

Πραγματικές  
Παράμετροι

Τυπικές  
Παράμετροι

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.
  - 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. Επομένως :
  - 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

#### 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, τις οποίες μεταβιβάζει στη διαδικασία «υπολόγισε» με την εντολή ΚΑΛΕΣΕ.
2. Η διαδικασία «υπολόγισε», 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 «υπολόγισε».
4. Η διαδικασία «υπολόγισε», με την σειρά της την επιστρέφει στο κύριο πρόγραμμα «παραδειγμα2».
5. Το κύριο πρόγραμμα «παραδειγμα2» εκτυπώνει την απόλυτη τιμή της παράστασης.

ΠΡΟΓΡΑΜΜΑ παραδειγμα2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: α, β

ΠΡΑΓΜΑΤΙΚΕΣ: τ

ΑΡΧΗ

ΔΙΑΒΑΣΕ α, β

ΚΑΛΕΣΕ υπολόγισε(α, β, τ)

ΓΡΑΨΕ τ

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ παραδειγμα2

ΔΙΑΔΙΚΑΣΙΑ υπολόγισε(β, γ, δ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: β, γ

ΠΡΑΓΜΑΤΙΚΕΣ: δ

ΑΡΧΗ

$\delta \leftarrow (\beta + 3 * \gamma) / 2$

ΚΑΛΕΣΕ κανεΘετικό(δ)

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ υπολόγισε

ΔΙΑΔΙΚΑΣΙΑ κανεΘετικό(α)

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: α

ΑΡΧΗ

ΑΝ α < 0 ΤΟΤΕ

α <- (- 1) \* α

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ κανεΘετικό

Στο ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ «παραδειγμα2» εισάγονται τιμές από το πληκτρολόγιο, οι οποίες μεταβιβάζονται στη διαδικασία «υπολόγισε»

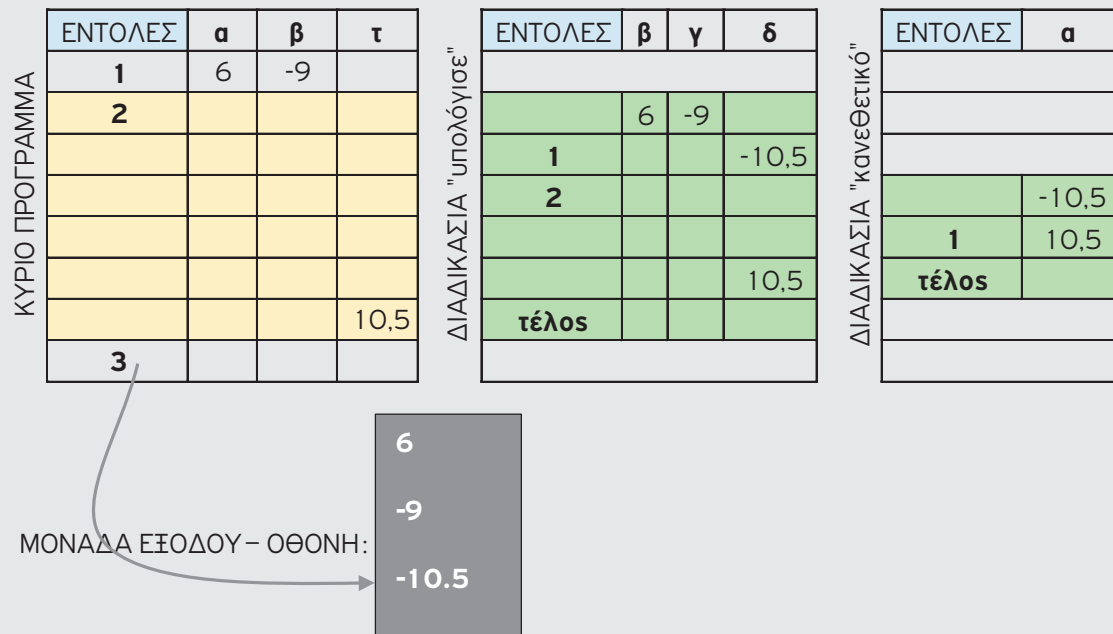
Το ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ «παραδειγμα2» εκτυπώνει την απόλυτη τιμή της παράστασης, που έχει υπολογιστεί από τις άλλες διαδικασίες και του έχει επιστραφεί.

Η ΔΙΑΔΙΚΑΣΙΑ «υπολόγισε», υπολογίζει την τιμή μιας παράστασης

Η ΔΙΑΔΙΚΑΣΙΑ «υπολόγισε» καλεί τη ΔΙΑΔΙΚΑΣΙΑ «κανεΘετικό» και της μεταβιβάζει την τιμή της παράστασης που υπολόγισε. Η ΔΙΑΔΙΚΑΣΙΑ «κανεΘετικό» επιστρέφει την απόλυτη τιμή της παράστασης, την οποία η ΔΙΑΔΙΚΑΣΙΑ «υπολόγισε» επιστρέφει στο ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ

Η ΔΙΑΔΙΚΑΣΙΑ «κανεΘετικό», υπολογίζει την απόλυτη τιμή της παράστασης και την επιστρέφει στη ΔΙΑΔΙΚΑΣΙΑ «υπολόγισε»

Ο πίνακας τιμών του παραδείγματος φαίνεται στο παρακάτω σχήμα.



## 5. Διαδικασίες και Πίνακες

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

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)

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

ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ gemisma\_pinaka

ΔΙΑΔΙΚΑΣΙΑ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(A)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], i

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΓΡΑΨΕ 'A[', i, ']='

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ

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

ΔΙΑΔΙΚΑΣΙΑ ΕΚΤΥΠΩΣΗ\_ΠΙΝΑΚΑ(A)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], i

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΓΡΑΨΕ 'A[', i, ']=' , A[i]

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ ΕΚΤΥΠΩΣΗ\_ΠΙΝΑΚΑ

Στο τμήμα δηλώσεων των μεταβλητών, οι πίνακες δηλώνονται κανονικά



**Ιδιαίτερη προσοχή χρειάζεται 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 δεσμεύονται στη μνήμη, π.χ. **ΚΑΛΕΣΕ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(x)** και **όχι** ΚΑΛΕΣΕ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(x[100]).
- Ο ίδιος κανόνας χρησιμοποιείται και στην τυπική παράμετρο των διαδικασιών, π.χ. **ΔΙΑΔΙΚΑΣΙΑ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(x)** και **όχι** ΔΙΑΔΙΚΑΣΙΑ ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ(x[100]).
- Στο τμήμα δηλώσεων των μεταβλητών, οι πίνακες 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 x[100] με ακέραιους αριθμούς.**

### Λύση

#### Ανάλυση

- Η διαδικασία θα έχει τρεις παραμέτρους, τον πίνακα x σαν δεδομένο, τη Min που θα επιστρέφει τον μικρότερο αριθμό και τη 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.

ΔΙΑΔΙΚΑΣΙΑ Μικρότερο(x, Min, Min\_θέση)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: x[100], i, Min, Min\_θέση

ΑΡΧΗ

Min <- x[1]

Min\_θέση <- 1

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100

ΑΝ x[1] < Min ΤΟΤΕ

Min <- x[i]

Min\_θέση <- i

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ Μικρότερο

## 5.2 Συναρτήσεις (Παράδειγμα συνάρτησης)

Να γραφεί πρόγραμμα που να διαβάζει δύο ακέραιους αριθμούς και να τυπώνει το άθροισμα τους και τον μέσο όρο τους. Ο υπολογισμός του αθροίσματος και του μέσου όρου γίνεται από δύο συναρτήσεις.

### Λύση

Κύριο Πρόγραμμα

```
ΠΡΟΓΡΑΜΜΑ Αθρ_ΜΟ
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: α, β, sum
 ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
ΑΡΧΗ
 ΓΡΑΨΕ 'Δώσε δυο ακέραιους αριθμούς:'
 ΔΙΑΒΑΣΕ α, β
 sum <- Αθροισμα (α, β)
 ΜΟ <- Μέσο_όρο (α, β)
 ΓΡΑΨΕ 'Αθροισμα = ', sum
 ΓΡΑΨΕ 'Μέσος Όρος = ', ΜΟ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Συνάρτηση υπολογισμού αθροίσματος

```
ΣΥΝΑΡΤΗΣΗ Αθροισμα(x, y): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: x, y
ΑΡΧΗ
 Αθροισμα <- x + y
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Συνάρτηση υπολογισμού μέσου όρου

```
ΣΥΝΑΡΤΗΣΗ Μέσο_όρο(x, y): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: x, y
ΑΡΧΗ
 Μέσο_όρο <- (x + y)/2
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

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.

## 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 συνάρτησης>) μετά το όνομα της συνάρτησης, εκφράζει τον τύπο της τιμής που μεταφέρει η συνάρτηση.

Αν δηλαδή η τιμή που επιστρέφει είναι ακέραια, πραγματική, λογική ή χαρακτήρας, ανεξάρτητα από τον τύπο των δικών της ορισμάτων. Π.χ. η συνάρτηση «Αθροισμα» παίρνει τιμές που είναι ακέραιες (η συνάρτηση πρέπει να επιστρέφει τιμή ακέραιου τύπου). Οι παράμετροί της μπορεί να είναι ακέραιου τύπου, αν προσθέτει μόνο ακέραιους αριθμούς.

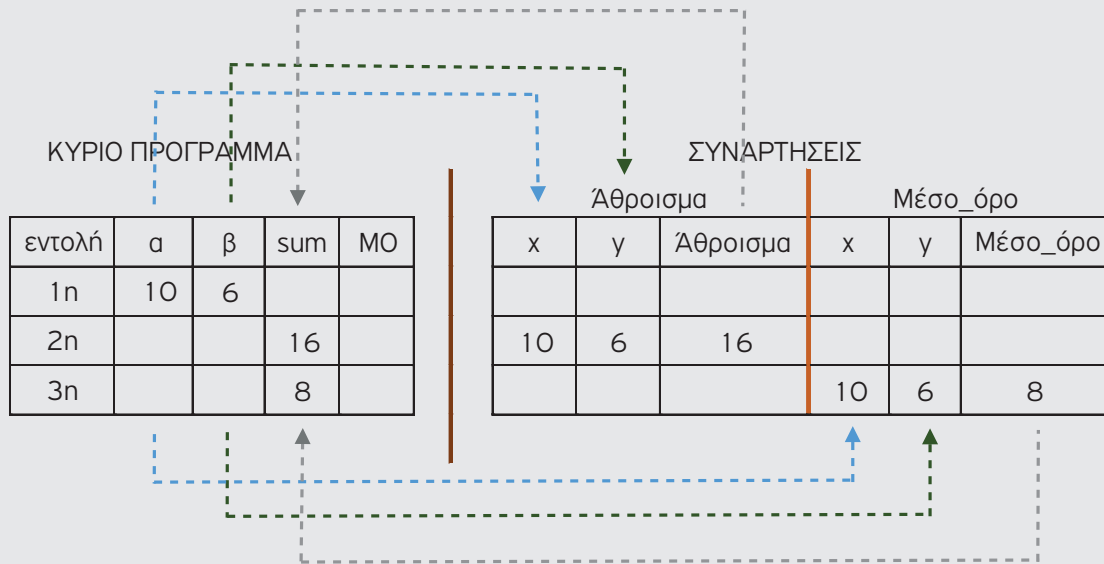
Αν οι παράμετροί της ήταν πραγματικού τύπου και η συνάρτηση υπολόγιζε το ακέραιο μέρος του αθροίσματός τους, τότε η συνάρτηση, θα πρέπει να είναι ακέραιου τύπου και όχι πραγματικού τύπου που είναι οι παράμετροί της.

## 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<-Αθροισμα(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.

### 3. 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 ολοκληρωθεί η εκτέλεση των εντολών της ΣΥΝΑΡΤΗΣΗΣ τότε όλες οι μεταβλητές της χάνονται και ελευθερώνεται η αντίστοιχη περιοχή μνήμης.
- Οι μεταβλητές a και β δίνουν την τιμή τους στις μεταβλητές x και y αντίστοιχα όταν καλείται η συνάρτηση άθροισμα, ενώ δίνουν την τιμή τους αντίστοιχα στις x και y όταν καλείται η συνάρτηση Μέσος Όρος.
- Η μεταβλητή Άθροισμα δίνει την τιμή της στη μεταβλητή sum του κύριου προγράμματος όταν τελειώσει η συνάρτηση Άθροισμα και ομοίως η μεταβλητή Μέσο\_Όρο στη MO όταν τελειώσει η συνάρτηση Μέσο\_Όρο.

## 5.3 Διαδικασίες και Συναρτήσεις

### Διαφορές των Διαδικασιών 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 μία τιμή και μόνο αυτή επιστρέφουν στο κύριο πρόγραμμα ή στο υποπρόγραμμα που τις κάλεσε. Η τιμή αυτή μπορεί να είναι ΑΡΙΘΜΗΤΙΚΗ (ΑΚΕΡΑΙΑ ή ΠΡΑΓΜΑΤΙΚΗ), 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 ονόματός τους μέσα σε οποιαδήποτε εντολή.                                                                                                                          |



**Μια συνάρτηση μπορεί να αντικατασταθεί με αντίστοιχη διαδικασία**

### Διαδικασίες και Συναρτήσεις στο ίδιο Πρόγραμμα

Ένα πρόγραμμα είναι δυνατόν να περιλαμβάνει και διαδικασίες και συναρτήσεις.

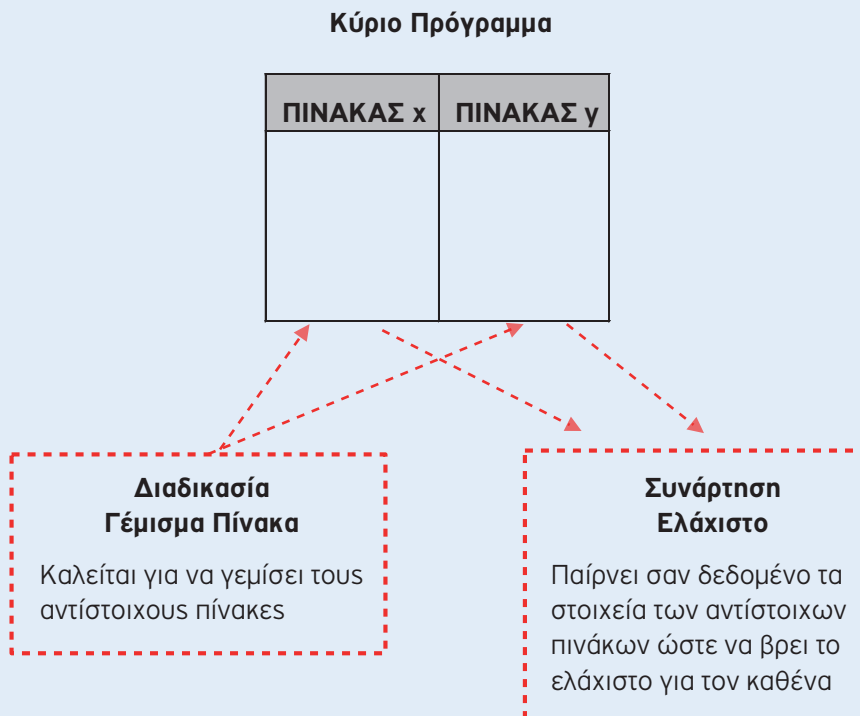
Να γραφεί πρόγραμμα που να διαβάξει δύο πίνακες  $x[100]$  και  $y[100]$  με ακέραιους αριθμούς και στη συνέχεια να τυπώνει τον ελάχιστο για κάθε πίνακα.

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

## Λύση

### Ανάλυση προβλήματος

Σχηματική αναπαράσταση:



1. Πρέπει να γεμίσουν οι δύο πίνακες με κλήση των διαδικασιών ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ (x) και ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ (y). Καλείται η ίδια διαδικασία για να γεμίσουν οι δύο πίνακες. Αυτό γίνεται με αλλαγή της παραμέτρου, όπου κάθε φορά παίρνει το όνομα του πίνακα που αφορά. Αυτό είναι και το βασικό πλεονέκτημα των υποπρογραμμάτων, ότι μπορούν να χρησιμοποιηθούν όσες φορές απαιτείται, 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. Δεδομένου ότι η συνάρτηση καλείται μέσα από εντολή, είτε θα χρησιμοποιηθεί η εντολή ΓΡΑΨΕ οπότε και τυπώνεται απευθείας το αποτέλεσμα είτε θα χρησιμοποιηθεί η εντολή καταχώρισης για να γίνει η αποθήκευση της τιμής της σε μία μεταβλητή και στη συνέχεια να τυπωθεί η μεταβλητή.
3. Τα υποπρογράμματα (η Διαδικασία ΓΕΜΙΣΜΑ\_ΠΙΝΑΚΑ και η Συνάρτηση ΕΛΑΧΙΣΤΟ) γράφονται μετά το τέλος του κύριου προγράμματος.

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**

## 5.4 Μη λυμένες ασκήσεις

1. Να γραφεί συνάρτηση, η οποία να επιστρέφει την τιμή ΑΛΗΘΗΣ, αν ένας αριθμός είναι πολλαπλάσιο του 2 και στην αντίθετη περίπτωση, να επιστρέφει την τιμή ΨΕΥΔΗΣ.
2. Να γραφεί διαδικασία, η οποία να διαβάζει ακέραιους αριθμούς και να επιστρέφει το πλήθος των θετικών, το πλήθος των αρνητικών και το πλήθος των μηδέν. Το πλήθος των ακεραίων που θα διαβαστούν θα ορίζεται σαν παράμετρος της διαδικασίας.

**3.** Να γραφεί υποπρόγραμμα, το οποίο να αναζητεί ένα ακέραιο αριθμό σ' έναν πίνακα x[1000] ακέραιων αριθμών. Όταν τον βρίσκει να επιστρέφει τη θέση που βρέθηκε, διαφορετικά να επιστρέφει μηδέν. Ο πίνακας και ο ακέραιος αριθμός να ορίζονται ως παράμετροι του υποπρογράμματος (οι αριθμοί στον πίνακα είναι μοναδικοί).

**4.** Να γραφεί υποπρόγραμμα το οποίο να ταξινομεί τον πίνακα x[100] με ονόματα, αλφαβητικά. Στην αντιμετάθεση των τιμών να καλείται το υποπρόγραμμα ANTIMETAΘΕΣΗ (x,y), όπου και θα αναπτυχθεί κατάλληλα. Ο πίνακας x να είναι παράμετρος του υποπρογράμματος.

**5.** Να γραφεί υποπρόγραμμα, που να εντοπίζει τους δύο μικρότερους αριθμούς από ένα πίνακα X[50] με ακέραιους αριθμούς και να τους επιστρέφει.

**6.** Να γραφεί διαδικασία, η οποία να υπολογίζει την παρακάτω μαθηματική παράσταση:

$$g(x) = x^2 + x^3 - \frac{\sqrt{2+x^3}}{x + 3*\sqrt{x}}$$

**7.** Να γραφεί υποπρόγραμμα, το οποίο να ελέγχει αν ο αριθμός που πληκτρολογείται είναι θετικός και να τον επιστρέφει στο υποπρόγραμμα που το καλεί. Σε περίπτωση που ο αριθμός δεν είναι θετικός να τυπώνεται μήνυμα λάθους και να επαναλαμβάνεται η πληκτρολόγηση μέχρι να πληκτρολογηθεί ο σωστός αριθμός.

**8.** Να γραφεί υποπρόγραμμα, το οποίο να ελέγχει αν ένας πίνακας A[100] με ακέραιους αριθμούς είναι ταξινομημένος κατ' αύξουσα σειρά ή όχι.

**9.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ» το οποίο να:

- 1) Καταχωρεί στους πίνακες Ημέρα[500] και Μήνα[500] την ημέρα και τον μήνα αντίστοιχα μιας δαπάνης, στον πίνακα Περιγραφή[500] την περιγραφή της δαπάνης και στον πίνακα Κόστος[500] το συνολικό ποσό της αντίστοιχης δαπάνης.
- 2) Ταξινομεί τους πίνακες πρώτα ως προς τον μήνα κατά αύξουσα σειρά και στην περίπτωση ίδιου μήνα η ταξινόμηση να γίνεται ως προς την ημέρα, κατά αύξουσα σειρά.
- 3) Δίνονται από το πληκτρολόγιο 2 ημερομηνίες (εκφράζουν μια χρονική περίοδο). Να εμφανίζεται η περιγραφή κάθε δαπάνης με τα αντίστοιχα ποσά δαπάνης της χρονικής περιόδου που πληκτρολογήσαμε. Επίσης, στο τέλος να εμφανίζεται το συνολικό ποσό που έχει ξοδευτεί την αντίστοιχη χρονική περίοδο.
- 4) Να γίνεται χρήση του υποπρογράμματος ΕΛΕΓΧΟΣ\_ΗΜΕΡΟΜΗΝΙΑΣ, το οποίο να ελέγχει ότι η 1η ημ/νια είναι μικρότερη από τη 2η ημ/νια. Στην αντίθετη περίπτωση να τις αντιμεταθέτει. Να χρησιμοποιηθεί η διαδικασία ANTIMETAΘΕΣΗ, όπου χρειάζεται.

**10.** Να γραφεί πρόγραμμα σε «ΓΛΩΣΣΑ» το οποίο να διαχειρίζεται πίνακα Τηλεφωνικού Καταλόγου 500 γραμμών, με 1η στήλη το Ονοματεπώνυμο, 2η στήλη τη Διεύθυνση και 3η στήλη το Τηλέφωνο. Η διαχείριση γίνεται σύμφωνα με το παρακάτω Μενού επιλογών:

1. Εισαγωγή νέου συνδρομητή
2. Διαγραφή συνδρομητή
3. Διόρθωση δεδομένων συνδρομητή
4. Αναζήτηση συνδρομητή (Με Όνομα ή Τηλέφωνο )
5. Εμφάνιση Τηλεφωνικού Καταλόγου
6. Έξοδος

ΕΠΙΛΟΓΗ: \_\_\_\_

Η κάθε επιλογή αποτελεί ξεχωριστό υποπρόγραμμα.

Στην περίπτωση που έχει γεμίσει ο πίνακας να εμφανίζεται κατάλληλο μήνυμα



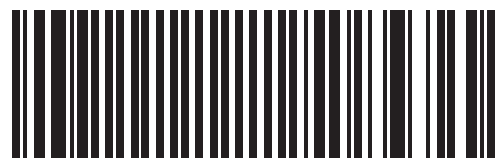


Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔΙΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946,108, Α').

*Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήματος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων / ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ.*



ΙΝΣΤΙΤΟΥΤΟ  
ΤΕΧΝΟΛΟΓΙΑΣ  
ΥΠΟΛΟΓΙΣΤΩΝ & ΕΚΔΟΣΕΩΝ



(01) 000000 0 22 0259 4

Κωδικός Βιβλίου: 0-22-0259  
ISBN 978-960-06-5996-2