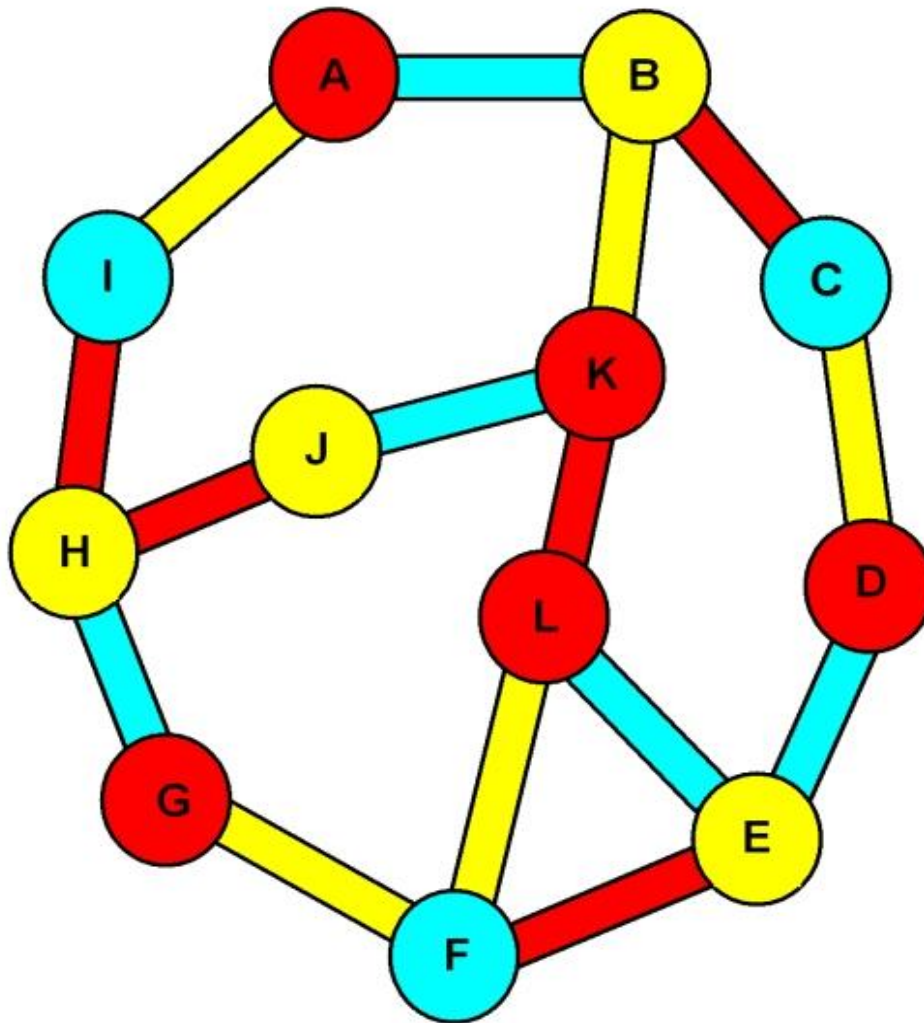


ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

ΤΕΧΝΟΛΟΓΙΚΗΣ ΚΑΤΕΥΘΥΝΣΗΣ



- ΑΝΑΛΥΣΗ ΚΑΙ ΘΕΩΡΙΑ
- ΜΕΘΟΔΟΛΟΓΙΑ ΑΣΚΗΣΕΩΝ
- 115 ΕΡΩΤΗΣΕΙΣ ΚΑΙ ΑΣΚΗΣΕΙΣ
- ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΟΛΩΝ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ

ΚΩΣΤΑΣ-ΠΕΡΙΚΛΗΣ ΖΩΤΟΣ

Ο λαβύρινθος του εξώφυλλου απαιτεί και τα δύο χέρια. Με το αριστερό χέρι ακουμπήστε τον δείκτη στο σημείο Α. Με το δεξί χέρι ακουμπήστε τον δείκτη στο σημείο Β. Σκοπός είναι να εκτελέσετε μια σειρά από κινήσεις που θα οδηγήσουν και τους δύο δείκτες στον ίδιο κύκλο. Σε κάθε μια κίνηση ο ένας δείκτης μετακινείται σ' ένα γειτονικό κύκλο διαμέσου του έγχρωμου δρόμου. Υπάρχει και ένας περιορισμός: το χρώμα του δρόμου, όπου ο δείκτης του χεριού σας μετακινείται, πρέπει να είναι όμοιο με το χρώμα του κύκλου που βρίσκεται ο δείκτης του άλλου χεριού.

Τι μπορεί να «μπλοκάρει» το μυαλό μας;

ΤΟ ΑΓΧΟΣ. Χαρακτηριστική περίπτωση είναι το άγχος των εξετάσεων, όπου -παρότι ένας μαθητής είναι διαβασμένος- τελικά δεν αποδίδει. Αυτό φυσικά δεν σημαίνει ότι δεν είναι έξυπνος, αλλά ότι δεν μπορεί να χειριστεί αποτελεσματικά το άγχος του.

Η ΑΨΥΧΙΑ. Όταν χάνουμε τον ύπνο μας για 2-3 συνεχόμενα 24ωρα, όλοι παρουσιάζουμε ενδείξεις νοητικής αποδιοργάνωσης - ευτυχώς παροδικής: αδυναμία συγκέντρωσης, δυσκολία έκφρασης, κακό προσανατολισμό, απώλεια μνήμης.

Η ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ. Σύμφωνα με έρευνες, οι μαθητές που ζουν σε περιοχές βεβαρημένες με ρύπους ή τοξικές ουσίες (π.χ., κοντά σε εργοστάσιο χημικών) φαίνεται ότι έχουν χαμηλότερη απόδοση στα μαθήματα σε σύγκριση με τους συνομήλικούς τους που ζουν σε περιοχές χωρίς ατμοσφαιρική ρύπανση.

Ο ΚΑΙΡΟΣ. Τους καλοκαιρινούς μήνες επικρατεί μια γενικότερη βραδύτητα του νου. Αντίθετα, όταν ο υδράργυρος είναι σε χαμηλότερα επίπεδα, αποδίδουμε καλύτερα στην εργασία μας.

Η ΡΟΥΤΙΝΑ είναι εχθρός του νου. Όταν οι δραστηριότητες με τις οποίες ασχολούμαστε δεν μας ικανοποιούν τότε θα επηρεαστούν σταδιακά και οι νοητικές μας λειτουργίες. Έχει παρατηρηθεί ότι οι συνταξιούχοι που περνούν μονότονα και παθητικά τις ώρες τους μπροστά στην τηλεόραση, χωρίς να αξιοποιούν δημιουργικά το χρόνο τους, ενδέχεται να παρουσιάσουν εξασθένηση της μνήμης τους, καθώς και να παρουσιάσουν ταχύτερη γήρανση από την αναμενόμενη! Γι' αυτό είναι απαραίτητο το διαρκές παίδεμα του νου με ζητήματα που εξάπτουν τη δημιουργικότητα και τη φαντασία μας.

Η ΨΥΧΙΚΗ ΜΑΣ ΔΙΑΘΕΣΗ. Όταν δεν έχουμε καλή διάθεση, παρουσιάζουμε εικόνα διάχυτης νοητικής υπολειτουργίας. Δεν συγκρατούμε πληροφορίες, δεν επεξεργαζόμαστε σωστά τα δεδομένα, ούτε ακούμε τι μας λένε!

Η ΚΑΚΗ ΔΙΑΤΡΟΦΗ. Υπάρχουν έρευνες που αποδεικνύουν ότι η έλλειψη βασικών θρεπτικών συστατικών μπορεί να επηρεάσει τις νοητικές μας λειτουργίες. Έχει παρατηρηθεί, π.χ., ότι τα παιδιά που έχουν αλλεργία στο σιτάρι, τη σίκαλη και τη βρώμη, με αποτέλεσμα να στερούνται μια σημαντική ομάδα τροφίμων, κινδυνεύουν από νοητική στέρωση, εάν δεν ληφθούν εγκαίρως μέτρα ώστε η διατροφή τους να είναι ισορροπημένη. Τα γλυκά προσφέρουν άμεση τόνωση του εγκεφάλου, επειδή αυξάνουν ταχύτατα τα επίπεδα της γλυκόζης (σακχάρου) στο αίμα. Από την άλλη μεριά, όμως, το σάκχαρο μειώνεται εξίσου γρήγορα, με αποτέλεσμα να έχετε τα αντίθετα από τα αναμενόμενα αποτελέσματα: αδυναμία συγκέντρωσης, υπνηλία, νευρικότητα.

Ο κακός μαθητής δεν είναι έξυπνος;

Εύκολα χαρακτηρίζουμε ένα παιδί «κακό» μαθητή, συχνά όμως ξεχνάμε ότι πίσω από τον κακό μαθητή ίσως κρύβεται ένας «κακός» δάσκαλος ή ένα ανεπαρκές σύστημα εκπαίδευσης, που δεν προάγει ούτε τη φαντασία ούτε τη δημιουργικότητα των παιδιών. Φυσικά και είναι ανησυχητική η χαμηλή απόδοση του παιδιού στο σχολείο. Αντί όμως να αναρωτηθούμε για την ευφυΐα του, είναι προτιμότερο να ελέγξουμε πού μπορεί να οφείλονται οι κακοί του βαθμοί, π.χ., τυχόν οικογενειακά προβλήματα, έλλειψη χρόνου των γονιών ώστε να ασχοληθούν ουσιαστικά με τη βοήθεια του παιδιού, προβλήματα στις σχέσεις με τους συμμαθητές του, το γεγονός ότι το παιδί δεν έχει το δικό του χώρο στο σπίτι ώστε να διαβάσει με ηρεμία κλπ.

Η επιτυχία στις σπουδές είναι και δείκτης επιτυχίας στη ζωή;

Όχι απαραίτητα. Έχει παρατηρηθεί ότι στην Αμερική, όπου προωθήθηκε ένα διαφορετικό σύστημα εκπαίδευσης για τα λεγόμενα χαρισματικά παιδιά, τελικά δεν διαπιστώθηκε ότι αυτά πέτυχαν ιδιαίτερες διακρίσεις στη ζωή τους. Γι' αυτό και το συγκεκριμένο σύστημα εγκαταλείφθηκε. Η επιτυχία στη ζωή δεν καθορίζεται μόνο από την επιτυχία στις σπουδές. Η στήριξη από το οικογενειακό περιβάλλον, οι ευκαιρίες που θα παρουσιαστούν, το ενδιαφέρον για το αντικείμενο απασχόλησης, οι καλές διαπροσωπικές σχέσεις, ο συναισθηματικός κόσμος και φυσικά ο παράγοντας «τύχη» παίζουν επίσης σημαντικό ρόλο!

ΠΕΡΙΕΧΟΜΕΝΑ

- Η έννοια πρόβλημα, σελ.2
- Η έννοια αλγόριθμος, σελ.3-4
- Διαφορές αλγορίθμου και προγράμματος, σελ.5
- Αντιστοιχία εντολών αλγορίθμου – προγράμματος, σελ.6-7
- Μαθηματικές συναρτήσεις, σελ.7
- Τελεστές (συγκριτικοί, αριθμητικοί, λογικοί), σελ.7-8
- Συστήματα αρίθμησης, σελ.9
- Μαθηματική λογική, σελ.9-11
- Δομή επιλογής, σελ.12
- Γλώσσες προγραμματισμού, σελ.13-15
- Μεταγλωττιστές, διερμηνευτές και είδη προγραμματισμού, σελ.16-17
- Δομές δεδομένων, σελ.18-19
- Ασκήσεις με διαστήματα τιμών, σελ.20-21
- Πίνακας τιμών, σελ.22
- **Div** και **Mod**, σελ.23
- Δομή επανάληψης, σελ.24-26
- Ταξινόμηση, σελ.27
- Σειριακή αναζήτηση, σελ.28-32
- Ρυθμός μεταβολής, σελ.33
- Πως επιλύουμε το τέταρτο θέμα, σελ.34-40
- Τμηματικός προγραμματισμός, σελ.41-42
- Ασκήσεις, σελ.43-83
- Απαντήσεις, σελ.84-91
- Κώδικες - μεθοδολογία, σελ.92-97
- Βιβλιογραφία, σελ.98

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

Η ΕΝΝΟΙΑ ΠΡΟΒΛΗΜΑ

Πρόβλημα είναι μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση δεν είναι γνωστή, ούτε προφανής. Για την αντιμετώπιση κάθε προβλήματος πρέπει προηγουμένως να έχει προηγηθεί η κατανόησή του. Αποτελεί συνάρτηση δυο παραγόντων:

- Σωστή διατύπωση εκ μέρους του δημιουργού του
- Σωστή ερμηνεία από αυτόν που θα το επιλύσει

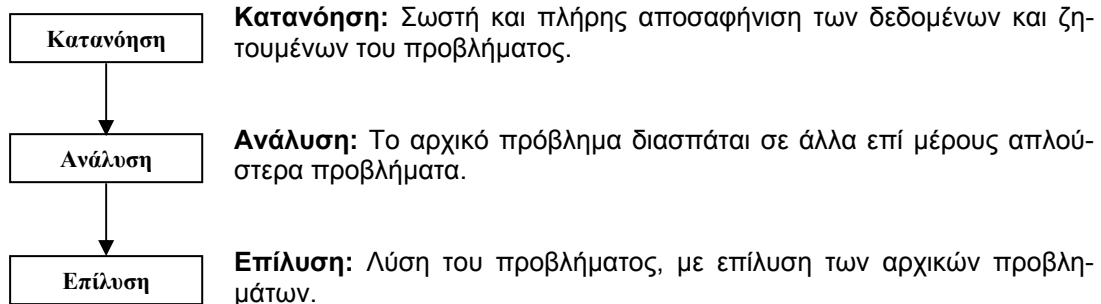
Πρόβλημα και υπολογιστής. Οι υπολογιστές δρουν επικουρικά στην ανθρώπινη δραστηριότητα, δεν έχουν τις δυνατότητες του ανθρώπινου εγκεφάλου. Σε αυτούς αναθέτουμε την επίλυση προβλημάτων λόγω: πολυπλοκότητας υπολογισμών, επαναληπτικότητα διαδικασιών, ταχύτητα εκτέλεσης πράξεων, μεγάλο πλήθος δεδομένων.

Κατηγορίες προβλημάτων:

- Με κριτήριο της δυνατότητα επίλυσης ενός προβλήματος.
Επιλύσιμα. Τα προβλήματα που η λύση τους είναι γνωστή και έχει διατυπωθεί ή η συνάφειά τους με ήδη επιλυμένα προβλήματα επιτρέπει να θεωρούμε βέβαιη την επίλυσή τους.
Ανοικτά. Η λύση τους δεν έχει ακόμη βρεθεί αλλά δεν έχει αποδειχθεί ότι δεν λύνονται.
Άλυτα. Έχει αποδειχθεί ότι δεν επιλύονται.
- Με κριτήριο το βαθμό δόμησης ενός προβλήματος.
Δομημένα. Η επίλυση προέρχεται από μια αυτοματοποιημένη διαδικασία.
Ημιδομημένα. Η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επιλογής της.
Αδόμητα. Οι λύσεις τους δεν μπορούν να δομηθούν ή δεν έχει διερευνηθεί η δυνατότητα δόμησής τους.
- Με κριτήριο το είδος επίλυσης ενός προβλήματος.
Απόφασης. Το πρόβλημα πρέπει πάρει μια απόφαση συνήθως σε μια ερώτηση ναι/όχι.
Υπολογιστικά. Η λύση τους απαιτεί τη διενέργεια πολύπλοκων υπολογισμών.
Βελτιστοποίησης. Το πρόβλημα επιζητά το βέλτιστο αποτέλεσμα για συγκεκριμένο σετ δεδομένων.

ΔΟΜΗ ΠΡΟΒΛΗΜΑΤΟΣ

Με τον όρο **δομή προβλήματος** αναφερόμαστε στα συστατικά του μέρη, στα επιμέρους τμήματα που το αποτελούν καθώς επίσης και στον τρόπο που αυτά συνδέονται μεταξύ τους. Η σωστή επίλυση ενός προβλήματος προϋποθέτει τον επακριβή προσδιορισμό των δεδομένων που παρέχει το πρόβλημα. Απαιτεί επίσης την λεπτομερειακή καταγραφή των ζητούμενων που αναμένονται σαν αποτελέσματα της επίλυσης του προβλήματος. Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία:



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

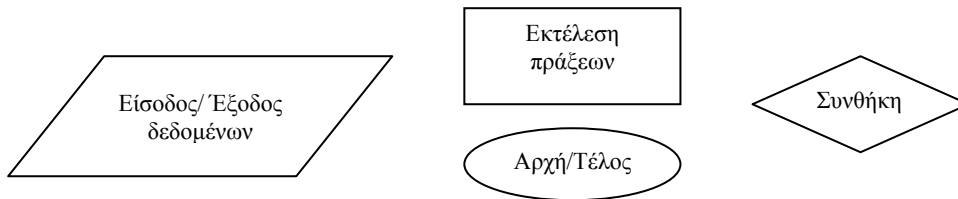
ΤΙ ΕΙΝΑΙ Ο ΑΛΓΟΡΙΘΜΟΣ;

Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Τα πέντε κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος είναι:

- Να έχει δεδομένα εισόδου που θα επεξεργαστεί.
- Να περιέχει απλές εντολές(αποτελεσματικότητα).
- Κάθε εντολή να ορίζεται χωρίς αμφιβολία ως προς την εκτέλεσή της, π.χ. διαίρεση με το μηδέν(καθοριστικότητα).
- Να ολοκληρώνεται μετά από πεπερασμένο αριθμό βημάτων(περατότητα).
- Να επιστρέφει το αποτέλεσμα της επεξεργασίας.

Ένας αλγόριθμος μπορεί να αναπαρασταθεί με:

- Ελεύθερο κείμενο.
- Διαγραμματικές τεχνικές(διάγραμμα ροής).



- Φυσική γλώσσα κατά βήματα.
- Κωδικοποίηση(πρόγραμμα γραμμένο σε ψευδογλώσσα είτε σε γλώσσα προγραμματισμού).

ΜΕΤΑΒΛΗΤΕΣ – ΣΤΑΘΕΡΕΣ

Για την συγγραφή ενός αλγορίθμου χρησιμοποιούνται:

- **Μεταβλητές:** Πρόκειται για ένα γλωσσικό αντικείμενο(το οποίο μπορούμε να χαρακτηρίσουμε και όνομα) όπου χρησιμοποιείται για να αναπαραστήσει ένα στοιχείο εισόδου. Μπορούμε να θεωρήσουμε τις μεταβλητές ως θέσεις μνήμης με συγκεκριμένο όνομα όπου περιέχεται μια τιμή η οποία και μπορεί να μεταβάλλεται κατά την εκτέλεση του αλγορίθμου. Οι μεταβλητές διακρίνονται σε:
 - **Αριθμητικές:** που αποθηκεύουν τιμές όπως 70, -32,5 κ.ο.κ.
 - **Αλφαριθμητικές:** που αποθηκεύουν τιμές όπως "Γεωργία" κ.ο.κ.
 - **Λογικές** με τιμή αληθής ή ψευδής.
- **Σταθερές:** Πρόκειται για προκαθορισμένες τιμές που παραμένουν αμετάβλητες κατά την εκτέλεση του αλγορίθμου. Και αυτές διακρίνονται σε αριθμητικές, αλφαριθμητικές και λογικές.

Τα ονόματα των μεταβλητών μπορούν να περιλαμβάνουν πεζά ή κεφαλαία γράμματα, αριθμούς και τον χαρακτήρα κάτω παύλα(_). Το όνομα πρέπει να ξεκινά από χαρακτήρα και όχι αριθμό. Αν επιθυμούμε την ύπαρξη δυο λέξεων τότε χρησιμοποιείται ή κάτω παύλα π.χ. Μέγιστη_Τιμή. Δεν επιτρέπεται να χρησιμοποιηθεί ως όνομα μεταβλητής κάποια από τις **δεσμευμένες** λέξεις της γλώσσας προγραμματισμού ή ακόμη και της ψευδογλώσσας που χρησιμοποιείται για την απεικόνιση του αλγορίθμου. Ουσιαστικά, χρησιμοποιούμε μια μεταβλητή για να αποθηκεύσουμε σ' αυτήν κάποια τιμή (είσοδο δεδομένων από το χρήστη ή το αποτέλεσμα κάποιας έκφρασης). Όταν χρησιμοποιούμε την μεταβλητή επικαλούμαστε την τιμή που αυτή περιέχει. Για την χρησιμοποίηση της τιμής κάποιας μεταβλητής πρέπει οπωσδήποτε να έχει προηγηθεί εκχώρηση τιμής σε αυτή τη μεταβλητή είτε με εντολή εκχώρησης είτε με είσοδο δεδομένων από τον χρήστη. Πιο πρακτικά, δεν μπορούμε να χρησιμοποιούμε μια μεταβλητή στο δεξί τμήμα μιας εντολής εκχώρησης ή σε μια εκτύπωση αν προηγουμένως δεν έχει λάβει τιμή στον αλγόριθμο (**αρχικοποίηση**).

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΠΑΡΑΔΕΙΓΜΑ

Να γραφεί αλγόριθμος που θα διαβάζει την τιμή ενός προϊόντος χωρίς ΦΠΑ και θα βρίσκει την τελική του τιμή. Ο φόρος είναι 19% .

Λύση

1^{ος} τρόπος

1. φόρος \leftarrow 1.19
2. **Διάβασε** αρχ_τιμή
3. τελ_τιμή \leftarrow αρχ_τιμή*φόρος
4. **Εκτύπωσε** τελ_τιμή

(Οι εντολές εκτελούνται πάντοτε σειριακά.)

2^{ος} τρόπος

Αλγόριθμος ΦΠΑ
φόρος \leftarrow 1.19
Διάβασε αρχ_τιμή
τελ_τιμή \leftarrow αρχ_τιμή*φόρος
Εκτύπωσε τελ_τιμή
Τέλος ΦΠΑ

3^{ος} τρόπος

Πρώτα διαβάζουμε την τιμή του προϊόντος, έπειτα υπολογίζουμε την τελική τιμή πολλαπλασιάζοντας την αρχική επί 1,19. Τέλος εμφανίζουμε την τελική τιμή του προϊόντος.

4^{ος} τρόπος

ΠΡΟΓΡΑΜΜΑ ΦΟΡΟΣ

ΣΤΑΘΕΡΕΣ

φόρος=1.19

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: τελ_τιμή, αρχ_τιμή

ΑΡΧΗ

ΓΡΑΨΕ 'Δώστε την αρχική τιμή: '

ΔΙΑΒΑΣΕ αρχ_τιμή

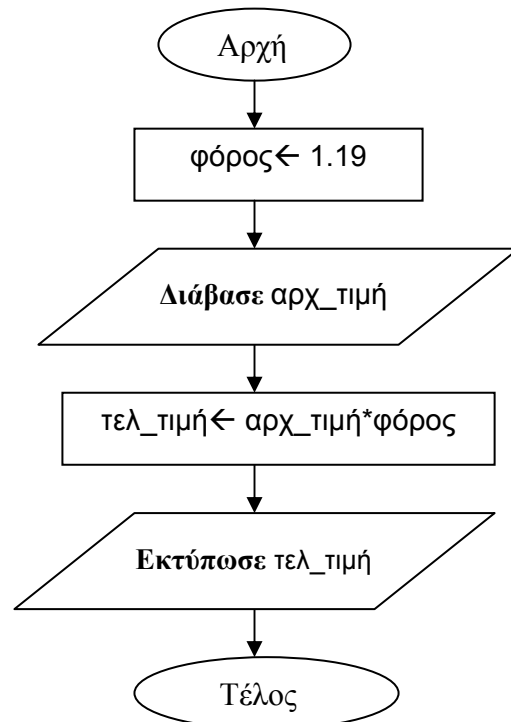
τελ_τιμή \leftarrow αρχ_τιμή*φόρος

ΓΡΑΨΕ 'Η τελική τιμή είναι: ', τελ_τιμή

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ



5^{ος} τρόπος



Ενώ ο "αλγόριθμος" φαίνεται πως είναι όρος που μόνο ένας μαθηματικός θα μπορούσε να αγαπήσει, είναι πραγματικά κάτι που ο καθένας μας χρησιμοποιεί καθημερινά. Ένας αλγόριθμος δεν είναι τίποτα περισσότερο από ένα σύνολο βημάτων για να επιτύχουμε έναν στόχο. Σχεδόν όλοι έχουμε μάθει στα παιδικά μας χρόνια τον αλγόριθμο για να δένουμε με κόμπο τα παπούτσια μας: Κατ' αρχάς, τεντώνουμε την κορδέλα πέρα από το παπούτσι. Κατόπιν κάνουμε έναν βρόχο στη σωστή θέση. Διασχίζουμε το βρόχο πίσω από την άλλη κορδέλα και μέσω της τρύπας τραβάμε...

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον Βασικές γνώσεις

ΟΙ ΔΙΑΦΟΡΕΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΜΕ ΤΟ ΠΡΟΓΡΑΜΜΑ

- Στον Αλγόριθμο ξεκινούμε με:
Αλγόριθμος <Όνομα_αλγορίθμου>
και τελειώνουμε με:
Τέλος <Όνομα_αλγορίθμου>

Στο Πρόγραμμα ξεκινούμε με:
ΠΡΟΓΡΑΜΜΑ <Όνομα_προγράμματος>
και τελειώνουμε με:
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
ή
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ <Όνομα_προγράμματος>
- Στο Πρόγραμμα γράφουμε με κεφαλαία γράμματα. Στον Αλγόριθμο αντίθετα, όλα είναι μικρά εκτός από τον πρώτο χαρακτήρα της κάθε γραμμής(εξαιρείται το **αλλιώς** και το **αλλιώς_αν**).
- Στον Αλγόριθμο χρησιμοποιούμε: **με βήμα** (με κάτω παύλα) ενώ στο Πρόγραμμα: **ΜΕ ΒΗΜΑ** (χωρίς κάτω παύλα)
- Στον Αλγόριθμο έχουμε τους συμβολισμούς: \leq , \geq και \neq στο Πρόγραμμα οι αντίστοιχοι είναι: \leq , \geq και \neq
- Στον Αλγόριθμο έχουμε τις εντολές: **Εμφάνισε**, **Εκτύπωσε** κ.τ.λ. ενώ στο Πρόγραμμα μόνο: **ΓΡΑΨΕ**
- Στο Πρόγραμμα δηλώνουμε πάντοτε τις μεταβλητές, στον Αλγόριθμο δεν είναι απαραίτητο. Όταν η εκφώνηση της άσκησης αναφέρει την ανάγνωση των δεδομένων από το χρήστη τότε χρησιμοποιείται η εντολή **Διάβασε**. Σε διαφορετική περίπτωση, π.χ. όταν η εκφώνηση αναφέρει “*έστω μεταβλητή N που περιέχει το πλήθος*”, εισάγουμε την τιμή της μεταβλητής N στον αλγόριθμο με την εντολή **Δεδομένα**.
- Στο Πρόγραμμα δεν χρησιμοποιούμε ποτέ την εντολή **αντιμετάθεσε** αλλά κάνουμε αντιμετάθεση με χρήση βοηθητικής μεταβλητής.

$$\left. \begin{array}{l} \text{temp} \leftarrow A[j-1] \\ A[j-1] \leftarrow A[j] \\ A[j] \leftarrow \text{temp} \end{array} \right\} \text{αντιμετάθεσε } A[j], A[j-1]$$

Στον Αλγόριθμο μπορούμε να βάλουμε και τα δύο.

- Στο Πρόγραμμα κάθε εντολή γράφεται σε ξεχωριστή γραμμή. Αν μία εντολή πρέπει να συνεχιστεί και στην επόμενη γραμμή, τότε ο πρώτος χαρακτήρας αυτής της γραμμής πρέπει να είναι ο χαρακτήρας **&**.
- Στο Πρόγραμμα αν ο πρώτος χαρακτήρας είναι θαυμαστικό(!) τότε αυτή η εντολή περιέχει σχόλια και όχι εκτελέσιμες εντολές.
- Στο Πρόγραμμα πριν από κάθε εντολή **ΔΙΑΒΑΣΕ**, δίνουμε μια εντολή **ΓΡΑΨΕ** που επεξηγεί τι πρόκειται να διαβάσει.
- Στο Πρόγραμμα έχουμε μονά εισαγωγικά(' ') ενώ στον αλγόριθμο διπλά(" ").
(στις εξετάσεις και τα δύο είναι αποδεκτά.)

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

ΑΝΤΙΣΤΟΙΧΙΑ ΕΝΤΟΛΩΝ ΑΛΓΟΡΙΘΜΟΥ- ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΙ VICE VERSA

ΑΛΓΟΡΙΘΜΟΣ	ΠΡΟΓΡΑΜΜΑ
Διάβασε <όνομα μεταβλητής>	ΔΙΑΒΑΣΕ <όνομα μεταβλητής>
Γράψε <όνομα μεταβλητής> Εμφάνισε <όνομα μεταβλητής> Εκτύπωσε <όνομα μεταβλητής>	ΓΡΑΨΕ <όνομα μεταβλητής>
Αν <συνθήκη> τότε ... αλλιώς ... Τέλος_αν	ΑΝ <συνθήκη> ΤΟΤΕ ... ΑΛΛΙΩΣ ... ΤΕΛΟΣ_ΑΝ
Αν <συνθήκες> τότε ... αλλιώς_αν <συνθήκη> τότε ... αλλιώς_αν <συνθήκη> τότε ... αλλιώς ... Τέλος_αν	ΑΝ <συνθήκη> ΤΟΤΕ ... ΑΛΛΙΩΣ_ΑΝ <συνθήκη> ΤΟΤΕ ... ΑΛΛΙΩΣ_ΑΝ <συνθήκη> ΤΟΤΕ ... ΑΛΛΙΩΣ ... ΤΕΛΟΣ_ΑΝ
Για <μεταβλητή> από <αρχή> μέχρι <τέλος> με_βήμα <βήμα> ... Τέλος_απανάληψης	ΓΙΑ <μεταβλητή> ΑΠΟ <αρχή> ΜΕΧΡΙ <τέλος> ΜΕ ΒΗΜΑ <βήμα> ... ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Όσο <συνθήκη> επανάλαβε ... Τέλος_επανάληψης	ΌΣΟ <συνθήκη> ΕΠΑΝΑΛΑΒΕ ... ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Αρχή_επανάληψης ... Μέχρις_ότου <συνθήκη>	ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ_ΟΤΟΥ <συνθήκη>
Διάβασε <μεταβλητή> Επίλεξε <μεταβλητή> Περίπτωση <τιμή1> ... Περίπτωση <τιμή2> ... Περίπτωση αλλιώς ... Τέλος_επιλογών	ΔΙΑΒΑΣΕ <μεταβλητή> ΕΠΙΛΕΞΕ <μεταβλητή> ΠΕΡΙΠΤΩΣΗ <τιμή1> ... ΠΕΡΙΠΤΩΣΗ <τιμή2> ... ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ ... ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

ΑΝΤΙΣΤΟΙΧΙΑ ΕΝΤΟΛΩΝ ΑΛΓΟΡΙΘΜΟΥ- ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΙ VICE VERSA

ΑΛΓΟΡΙΘΜΟΣ	ΠΡΟΓΡΑΜΜΑ
Αλγόριθμος <όνομα>	ΠΡΟΓΡΑΜΜΑ <όνομα>
Δεν απαιτείται η δήλωση των μεταβλητών εκτός αν έχουμε δεδομένα που τα ορίζουμε ως εξής : Δεδομένα //<μεταβλητή 1>, <μεταβλητή2>, ...//	ΣΤΑΘΕΡΕΣ ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ:
Δεν απαιτείται η Αρχή	ΑΡΧΗ
...ΕΝΤΟΛΕΣ...	... ΕΝΤΟΛΕΣ
Τέλος <όνομα>	ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

Συνάρτηση	Παράδειγμα	Επεξήγηση
HM	HM (A)	Ημίτονο
ΣΥΝ	ΣΥΝ (A)	Συνημίτονο
ΕΦ	ΕΦ (A)	Εφαπτομένη
T_P	T_P (A)	Τετραγωνική ρίζα. Η πρόταση T_P (A) είναι ισότιμη με $A^{(1/2)}$.
ΛΟΓ	ΛΟΓ (A)	Λογάριθμος με βάση το 10.
E	E (A)	$e^A \rightarrow$ Είναι ισοδύναμο με $(2.718281828459045)^A$
A_M	A_M (A)	Ακέραιο μέρος π.χ. A_M (2.718281828459045) = 2
A_T	A_T (A)	Απόλυτη τιμή π.χ. A_T (-1821) = 1821

ΣΥΓΚΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

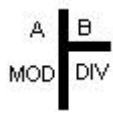
Είναι οι εξής: , , , ,

, ,

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

Σύμβολο	Επεξήγηση	Ιδιότητες	
+	Πρόσθεση	Η πρόσθεση ακεραίων δίνει πάντα ακέραιο αριθμό.	
-	Αφαίρεση	Η αφαίρεση ακεραίων δίνει πάντα ακέραιο αριθμό.	
*	Πολλαπλασιασμός	Ο πολλαπλασιασμός ακεραίων δίνει πάντα ακέραιο αριθμό.	
/	Διαίρεση	Προσοχή στις διαιρέσεις ειδικά όταν χρησιμοποιείτε ακέραιους, γιατί το προϊόν της διαίρεσης είναι συνήθως πραγματικός αριθμός.	
^	Ύψωση σε δύναμη	Ένας αριθμός υψωμένος σε κλάσμα ισούται με την αντίστοιχη ρίζα, για παράδειγμα: $A^{(1/2)} = \sqrt{A}$	
Mod	Ακέραιο υπόλοιπο		Μπορείτε να υπολογίσετε τις τιμές των παραστάσεων $A \bmod B$ και $A \text{ div } B$ αν κάνετε την διαίρεση και σταματήσετε πριν προσθέσετε το δεκαδικό μέρος στο αποτέλεσμα.
Div	Ακέραιο πηλίκο		

Η προτεραιότητα των αριθμητικών τελεστών είναι η εξής: 1.^ 2.*, / 3.+ , -

Όταν η προτεραιότητα είναι ίδια, οι πράξεις εκτελούνται από αριστερά προς τα δεξιά (Προσοχή οι παρενθέσεις μπορούν να αλλάξουν την προτεραιότητα των τελεστών).

Οι αριθμητικοί τελεστές έχουν υψηλότερη προτεραιότητα από τους λογικούς και τους συγκριτικούς, ενώ οι λογικοί τελεστές χαμηλότερη από τους συγκριτικούς.

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ

Οι λογικοί τελεστές χρησιμοποιούνται κατά πλειονότητα στις συνθήκες των επαναληπτικών δομών με τη μορφή : <συνθήκη_1> ΤΕΛΕΣΤΗΣ <συνθήκη_2>.

Τελεστής	Παράδειγμα	Επεξήγηση
και	A και B	Η πρόταση είναι αληθής όταν και οι δύο συνθήκες ισχύουν. Σε οποιαδήποτε άλλη περίπτωση είναι ψευδής.
ή	A ή B	Ψευδής μόνο όταν δεν ισχύουν και οι δύο συνθήκες. Αν ισχύει η μία από τις δύο, η πρόταση είναι αληθής.
όχι	όχι A	Η πρόταση είναι αληθής όταν η συνθήκη είναι ψευδής, και αντίστροφα.

Η προτεραιότητα των λογικών τελεστών είναι η εξής: 1.όχι 2.και 3.ή

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον Βασικές γνώσεις

ΣΥΣΤΗΜΑΤΑ ΑΡΙΘΜΗΣΗΣ

Το δεκαδικό σύστημα αρίθμησης δεν είναι το μοναδικό, όμως είναι αυτό που χρησιμοποιείται περισσότερο από τους ανθρώπους στις μέρες μας. Κατά καιρούς έχουν χρησιμοποιηθεί και άλλα αριθμητικά συστήματα, με διαφορετικές βάσεις (όπως το οκταδικό ή το δεκαεξαδικό σύστημα). Ένα σύστημα που χρησιμοποιείται αρκετά στις μέρες μας είναι το δυαδικό σύστημα αρίθμησης, το οποίο χρησιμοποιείται στην τεχνολογία των ηλεκτρονικών υπολογιστών. Η μετατροπή ενός δεκαδικού σε δυαδικό (ή δεκαεξαδικό, οκταδικό κ.τ.λ.) σύστημα είναι πολύ εύκολη. Ακολουθούν κάποια παραδείγματα.

Παράδειγμα μετατροπής δυαδικού αριθμού σε δεκαδικό:

$$(10010)_2 = 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = \\ = 1 * 16 + 0 + 0 + 2 + 0 = 16 + 2 = (18)_{10}$$

Παράδειγμα μετατροπής οκταδικού αριθμού σε δεκαδικό:

$$(15102)_8 = 1 * 8^4 + 5 * 8^3 + 1 * 8^2 + 0 * 8^1 + 2 * 8^0 = \\ = 1 * 4096 + 5 * 512 + 1 * 64 + 0 + 2 = \\ = 4096 + 2560 + 64 + 0 + 2 = (6722)_{10}$$

Παράδειγμα μετατροπής δεκαεξαδικού αριθμού σε δεκαδικό:

$$(30C3F)_{16} = 3 * 16^4 + 0 * 16^3 + C * 16^2 + 3 * 16^1 + F * 16^0 = \\ = 3 * 65536 + 0 * 4096 + 12 * 256 + 3 * 16 + 15 * 1 = \\ = 196608 + 3072 + 48 + 15 = (199743)_{10}$$

Γενικά παίρνουμε ένα προς ένα τα ψηφία και πολλαπλασιάζουμε το καθένα με τη βάση υψωμένη στην θέση του ψηφίου από δεξιά. Η πρώτη θέση είναι η θέση μηδέν. Η δεύτερη θέση είναι η θέση ένα κ.ο.κ. Μετά, αθροίζουμε τα γινόμενα και το αποτέλεσμα είναι δεκαδικός αριθμός. Για έλεγχο της μετατροπής του συστήματος αρίθμησης από το 1 μέχρι το 20 χρησιμοποιείστε τον παρακάτω πίνακα:

2δικός	8δικός	10δικός	16δικός	2δικός	8δικός	10δικός	16δικός
1	1	1	1	1011	13	11	B
10	2	2	2	1100	14	12	C
11	3	3	3	1101	15	13	D
100	4	4	4	1110	16	14	E
101	5	5	5	1111	17	15	F
110	6	6	6	10000	20	16	10
111	7	7	7	10001	21	17	11
1000	10	8	8	10010	22	18	12
1001	11	9	9	10011	23	19	13
1010	12	10	A	10100	24	20	14

ΜΑΘΗΜΑΤΙΚΗ ΛΟΓΙΚΗ – ΆΛΓΕΒΡΑ BOOLE

Ο υπολογιστής, εκτός από αριθμητικές πράξεις, έχει τη δυνατότητα να εκτελεί και συγκρίσεις, δηλαδή να επεξεργάζεται λογικά δεδομένα. Τα δεδομένα αυτά μπορεί να είναι απλές ή σύνθετες λογικές προτάσεις. Μια πρόταση λέγεται λογική, όταν μπορεί να χαρακτηριστεί αληθής ή ψευδής. Ένας τέτοιος χαρακτηρισμός μιας πρότασης λέγεται τιμή αλήθειας ή απλά τιμή της πρότασης. Για παράδειγμα:

- Ο αριθμός 4 είναι άρτιος αριθμός - αληθής πρόταση.
- Ο πλανήτης Άρης είναι μεγαλύτερος από τη Γη - ψευδής πρόταση.

Ο πρώτος που διατύπωσε τους βασικούς κανόνες με τους οποίους οι λογικές προτάσεις μπορούν να παρουσιαστούν με μαθηματικά σύμβολα, ήταν ο Άγγλος μαθηματικός George Boole. Στην Άλγεβρα Boole υπάρχουν μόνο δύο είδη προτάσεων, αυτές που είναι αληθείς και αυτές που είναι ψευδείς. Δηλαδή οι μεταβλητές που χρησιμοποιούνται στην άλγεβρα αυτή μπορούν να πάρουν μόνο δύο τιμές.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

Οι βασικές πράξεις της άλγεβρας Boole είναι:

• **ΑΡΝΗΣΗ**

Με την εφαρμογή της άρνησης σε μία λογική πρόταση αντιστρέφεται η αλήθεια των τιμών της, δηλαδή το 1 μετατρέπεται σε 0 και το 0 σε 1. Για παράδειγμα:

A	ΌΧΙ A
Σήμερα θα βρέξει	Σήμερα δε θα βρέξει

Πίνακας αλήθειας	
A	ΌΧΙ A
Αληθής	Ψευδής
Ψευδής	Αληθής

Η πράξη αυτή λέγεται και αντιστροφή.

• **ΣΥΖΕΥΞΗ**

Με την πράξη της σύζευξης συνθέτουμε δύο προτάσεις με τη λέξη <<ΚΑΙ>>, οπότε σχηματίζεται μία νέα πρόταση, η οποία είναι αληθής μόνο όταν και οι δύο αρχικές είναι αληθείς. Για παράδειγμα:

A	B	A ΚΑΙ B
Ο αριθμός 4 είναι άρτιος (Αληθής)	Ο αριθμός 5 είναι μεγαλύτερος του 8 (Ψευδής)	Ο αριθμός 4 είναι άρτιος και ο αριθμός 5 είναι μεγαλύτερος του 8

Πίνακας αλήθειας		
A	B	A ΚΑΙ B
Αληθής	Αληθής	Αληθής
Αληθής	Ψευδής	Ψευδής
Ψευδής	Αληθής	Ψευδής
Ψευδής	Ψευδής	Ψευδής



Η πράξη αυτή λέγεται και λογικός πολλαπλασιασμός και συμβολίζεται με A·B .

• **ΔΙΑΖΕΥΞΗ**

Με την πράξη της διάζευξης συνθέτουμε δύο προτάσεις με τη λέξη <<Η>>, οπότε σχηματίζεται μια νέα πρόταση, η οποία είναι αληθής όταν τουλάχιστον η μία από τις δύο αρχικές είναι αληθής. Για παράδειγμα:

A	B	A Η B
Ο αριθμός 9 είναι μεγαλύτερος του 8 (Αληθής)	Ο αριθμός 5 είναι άρτιος (Ψευδής)	Ο αριθμός 9 είναι μεγαλύτερος του 8 ή ο αριθμός 5 είναι άρτιος

Πίνακας αλήθειας		
A	B	A Η B
Αληθής	Αληθής	Αληθής
Αληθής	Ψευδής	Αληθής
Ψευδής	Αληθής	Αληθής
Ψευδής	Ψευδής	Ψευδής



Η πράξη αυτή λέγεται και λογική πρόσθεση και συμβολίζεται με A+B

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΠΑΡΑΔΕΙΓΜΑ 1

Μια φορά και έναν καιρό ήταν ένας πιλότος ονόματι Γιάννης και έψαχνε κάποιο χαμένο θησαυρό...Καθώς λοιπόν πετούσε με το μονοκινητήριο του πάνω από μια ζούγκλα, χαλάει ο κινητήρας του με αποτέλεσμα να πέσει στο κενό. Ευτυχώς γι' αυτόν, μια φυλή των ζουλού τον βρίσκει και τον πάει στον αρχηγό της. Ζαλισμένος και ταλαιπωρημένος από την περιπέτειά του ο Γιάννης ακούει τον φύλαρχο με ύφος τρομερό να τον ρωτάει:

- Πως σε λένε παιδί μου;

- Γιάννη.

- Κοίταξε να δεις Γιάννη παιδί μου, το ότι θα πεθάνεις είναι σίγουρο. Εμείς όμως θα σου δώσουμε την επιλογή να διαλέξεις πως θα πεθάνεις.

- Άκουσε πως έχουν τα πράγματα, θα μας πεις μια πρόταση. Αν αυτή η πρόταση είναι αληθής θα σε κάψουμε στην πυρά. Αν αυτή η πρόταση είναι ψευδής θα σε σουβλίσουμε. Όμως ο αρχηγός δεν ήξερε καλά τον Γιάννη... Έτσι ο Γιάννης είπε μια πρόταση και τον άφησαν και έφυγε συνεχίζοντας την αναζήτηση του χαμένου θησαυρού. Τι είπε;

Λύση

Ο Γιάννης θα πει: "**Εσείς θα με σουβλίσετε!**". Αν τον σουβλίσουν σημαίνει πως η πρόταση είναι ψευδής, οπότε δεν μπορούν να τον σουβλίσουν. Αν όμως τον κάψουν σημαίνει πως η πρόταση είναι αληθής δηλαδή θα πρέπει να τον σουβλίσουν. Άρα δεν μπορούν ούτε να τον κάψουν!!! Μεταξύ μας δεν νομίζω πως θα κρατήσουν τον λόγο τους...

	Πρόταση	Αποτέλεσμα
Εσείς θα με σουβλίσετε!	Αληθής	δεν μπορούν να τον σουβλίσουν
	Ψευδής	δεν μπορούν να τον σουβλίσουν

ΠΑΡΑΔΕΙΓΜΑ 2

Ένας απογραφέας μπαίνει σ' ένα σπίτι και ρωτάει την νοικοκυρά πόσοι άνθρωποι μένουν εκεί. Εκείνη του απαντάει πως μένει αυτή με τις τρεις κόρες της. Ο απογραφέας έπειτα ρωτάει ποιες είναι οι ηλικίες των κοριτσιών της και εκείνη του λέει πως επειδή της αρέσουν τα μαθηματικά παιχνίδια, θα του απαντήσει με έναν γρίφο: *Το γινόμενο των ηλικιών τους είναι ο αριθμός 36. Ο απογραφέας της λέει πως χρειάζεται και άλλα στοιχεία. Το άθροισμα των ηλικιών τους, προσθέτει, είναι ο αριθμός του σπιτιού μου.* Ο απογραφέας βγαίνει έξω, βλέπει τον αριθμό, αλλά ξαναμπαίνει μέσα και διαμαρτύρεται πως ούτε και πάλι μπορεί να υπολογίσει τις ηλικίες τους. *Η μεγάλη μου κόρη είναι συναχωμένη, συμπληρώνει η κυρία με νόημα.* Ο απογραφέας την ευχαριστεί πολύ και φεύγει. Ποιες είναι οι ηλικίες των τριών κοριτσιών της;

Λύση

Οι πιθανοί συνδυασμοί των ηλικιών των τριών κορών είναι οι εξής:

Κόρη 1	Κόρη 2	Κόρη 3	Άθροισμα
1	2	18	21
2	2	9	13
1	1	36	38
1	3	12	16
1	6	6	13
1	4	9	14
2	3	6	11

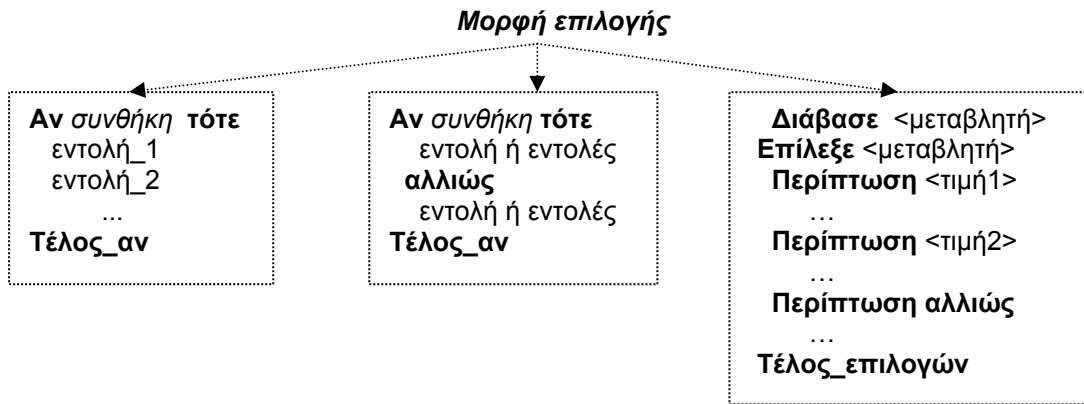
Αφού ο απογραφέας βγαίνοντας να δει τον αριθμό του σπιτιού δεν μπόρεσε να απαντήσει, αυτό σημαίνει ότι ο αριθμός ήταν το 13(αν ήταν οποιοσδήποτε άλλος αριθμός θα μπορούσε να απαντήσει αμέσως αφού θα υπήρχε μία φορά στην τρίτη στήλη του πίνακα, κάτι που δεν συμβαίνει με το 13). Έτσι όταν η νοικοκυρά του είπε πως η μεγάλη της κόρη είναι κρυωμένη κατάλαβε ότι ο σωστός συνδυασμός ήταν ο 2 2 9, πολύ απλά επειδή ο άλλος συνδυασμός έχει δυο κόρες ως τις πιο μεγάλες ενώ θα έπρεπε να ήταν μία.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΔΟΜΗ ΕΠΙΛΟΓΗΣ(ΑΠΛΗΣ ΚΑΙ ΠΟΛΛΑΠΛΗΣ)

Η δομή της επιλογής χρησιμοποιείται στις περιπτώσεις που χρειάζεται να λαμβάνονται κάποιες αποφάσεις με βάση κάποια δεδομένα κριτήρια, που μπορεί να είναι διαφορετικά για κάθε διαφορετικό στιγμιότυπο ενός προβλήματος.



ΠΑΡΑΔΕΙΓΜΑ

Να υλοποιήσετε τον παρακάτω αλγόριθμο με πολλαπλή επιλογή και εμφωλευμένα αν.

Αλγόριθμος Περιπτώσεις
Διάβασε α
Επίλεξε α
 Περίπτωση ≤ 0
 $\beta \leftarrow 0$
 Περίπτωση ≤ 5
 $\beta \leftarrow 5$
 Περίπτωση ≤ 10
 $\beta \leftarrow 10$
 Περίπτωση αλλιώς
 $\beta \leftarrow 100$
Τέλος_επιλογών
Εκτύπωσε β
Τέλος Περιπτώσεις

Λύση



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Ένα **πρόγραμμα** είναι μια σειρά βημάτων, οδηγιών ή εντολών προς τον υπολογιστή, κατανοητών από αυτόν, για τη λύση ενός προβλήματος. Οι εντολές αυτές εκτελούνται από τον υπολογιστή με χρονική και λογική σειρά προκειμένου να δώσουν τα προσδοκώμενα αποτελέσματα. Η διαδικασία σύνταξης των προγραμμάτων αποκαλείται **προγραμματισμός**.

Πώς όμως ένα πρόγραμμα μπορεί να γίνει αντιληπτό από τον υπολογιστή; Στην ιδανική περίπτωση ο άνθρωπος θα έπρεπε να μπορούσε να διατυπώσει στον υπολογιστή το πρόβλημα χρησιμοποιώντας τη φυσική του γλώσσα. Δυστυχώς, λόγω της τεράστιας πολυπλοκότητας των γλωσσών αυτών, προς το παρόν δεν είναι δυνατόν κάτι τέτοιο. Έτσι η εκπόνηση και η δημιουργία των προγραμμάτων γίνεται με τη χρήση ειδικών συμβολισμών, που ονομάζονται γλώσσες προγραμματισμού(programming languages).



Ο προγραμματισμός ενός υπολογιστή μπορεί να γίνει σε πολλά επίπεδα χρησιμοποιώντας σε κάθε επίπεδο διαφορετικού τύπου γλώσσες. Στο χαμηλότερο επίπεδο βρίσκονται τα ίδια τα ηλεκτρονικά κυκλώματα του υπολογιστή. Τα κυκλώματα όλων των υπολογιστών μέχρι σήμερα μπορούν να έχουν δύο μόνο καταστάσεις, στις οποίες αποδίδουμε τα σύμβολα 0 και 1. Έτσι στο επίπεδο αυτό ο προγραμματισμός μπορεί να γίνει μόνο με τη χρήση των δυαδικών αριθμών. Για παράδειγμα σ' έναν υπολογιστή η ακολουθία: 10001010 μπορεί να αποτελεί μία εντολή πρόσθεσης δύο αριθμών. Η "γλώσσα" αυτή των δυαδικών συμβόλων λέγεται **γλώσσα μηχανής**(machine language) ή **κώδικας μηχανής** (machine code) και είναι η μόνη γλώσσα που κατανοεί απ' ευθείας το υλικό του υπολογιστή.

Στα πρώτα χρόνια της ζωής των υπολογιστών η χρήση της γλώσσας μηχανής ήταν ο μόνος δυνατός τρόπος προγραμματισμού. Είναι φανερό ότι η διατύπωση των προβλημάτων στη γλώσσα αυτή είναι ξένη προς την ανθρώπινη φύση και για το λόγο αυτό ο προγραμματισμός στο επίπεδο μηχανής είναι εξαιρετικά επίπονος και ελάχιστα αποδοτικός.

Το επόμενο βήμα ήταν οι **συμβολικές γλώσσες**(assembly languages) ή γλώσσες χαμηλού επιπέδου. Μια συμβολική γλώσσα χρησιμοποιεί σύμβολα στη θέση των δυαδικών αριθμών διευκολύνοντας έτσι σημαντικά τη δουλειά του ανθρώπου. Τα σύμβολα αυτά είναι συνήθως συντμήσεις και αρκτικόλεξα αγγλικών λέξεων που μπορούν να απομνημονευθούν και για αυτό αποκαλούνται μνημονικές εντολές.

Ένα πρόγραμμα σε συμβολική γλώσσα προκειμένου να κατανοηθεί και να εκτελεσθεί από τον υπολογιστή, πρέπει να μετατραπεί σε δυαδικά σύμβολα, δηλαδή σε γλώσσα μηχανής. Το έργο αυτό αναλαμβάνει ο **συμβολομεταφραστής**(assembler), που και αυτός είναι ένα πρόγραμμα σε γλώσσα μηχανής.

Οι ανεπάρκειες των συμβολικών γλωσσών και η προσπάθεια για καλύτερη επικοινωνία ανθρώπου-μηχανής οδήγησαν στα τέλη της δεκαετίας 50 στην εμφάνιση των πρώτων γλωσσών προγραμματισμού υψηλού επιπέδου(high level languages). Στις γλώσσες αυτές που αποτελούν υποσύνολα της αγγλικής γλώσσας, η περιγραφή ενός προβλήματος γίνεται με τρόπο περισσότερο προσιτό στον άνθρωπο. Είναι προφανές ότι, όταν κάποιος ξεκινάει να λύσει ένα πρόβλημα, είναι μάλλον απίθανο να διατυπώσει τη λύση σε συνάρτηση με καταχωρητές και διευθύνσεις μνήμης. Το πιο πιθανό είναι να χρησιμοποιήσει συμβολισμούς από το ίδιο το πρόβλημα.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ

Επειδή κάθε ΚΜΕ(Κεντρική Μονάδα Επεξεργασίας, είναι ένα κύκλωμα στο οποίο εκτελούνται οι αριθμητικές και λογικές πράξεις, αποτελείται από την Αριθμητική και Λογική Μονάδα, τη Μονάδα Ελέγχου και τους Καταχωρητές) έχει διαφορετικό ρεπερτόριο εντολών, τα προγράμματα που εκτελεί πρέπει να είναι διατυπωμένα στη δική της γλώσσα μηχανής. Αυτά τα προγράμματα αποτελούνται από ακολουθίες 0 και 1. Κάθε εντολή στη γλώσσα μηχανής αποτελείται από δύο τμήματα, τον κωδικό λειτουργίας(operation code, OP code) και τον τελεστέο (operand).

Κωδικός λειτουργίας	Τελεστέος
---------------------	-----------

Ο κωδικός λειτουργίας προσδιορίζει τη λειτουργία της εντολής, για παράδειγμα <<πρόσθεσε στο συσσωρευτή>> ή <<αποθήκευσε το περιεχόμενο του συσσωρευτή στην τάδε θέση μνήμης>> και αντιστοιχεί σε μία από τις οδηγίες του συνόλου οδηγιών της ΚΜΕ. Ο τελεστέος αφορά τα δεδομένα στα οποία δρά η εντολή ή τη διεύθυνση στην οποία βρίσκονται αποθηκευμένα. Το μέγεθος(ο αριθμός των bit) κάθε εντολής μπορεί να είναι σταθερός ή μεταβλητός.

Για παράδειγμα, στον επεξεργαστή Z80, η εντολή 1110011010110011, κατευθύνει την ΚΜΕ να προσθέσει στο περιεχόμενο του καταχωρητή με το όνομα accumulator τον αντίστοιχο δυαδικό του 179₍₁₀₎. Τα πρώτα 8 ψηφία είναι ο κωδικός λειτουργίας, ενώ τα επόμενα 8 ψηφία ο τελεστέος της εντολής, ο αριθμός 179₍₁₀₎. Ένα πρόγραμμα γραμμένο σε γλώσσα μηχανής θα έμοιαζε κάπως έτσι:

```
1110011010110011
1011011010111011
101001111011011010100010
0101011010010011
1110011010110110
110001101011011110110100
1101011000110101
10101110
1100011010110010
```

(Κωδικός Λειτουργίας/ Τελεστέος)

Η κάθε εντολή θα μπορούσε, αντί να παρασταθεί στο δυαδικό σύστημα, να παρασταθεί στο οκταδικό ή στο δεκαεξαδικό, ώστε να είναι μικρότερος ο αριθμός των ψηφίων της. Τα πρώτα προγράμματα γράφονταν σε γλώσσα μηχανής. Ωστόσο η γραφή προγραμμάτων άμεσα στη γλώσσα μηχανής του υπολογιστή είναι μία πολύ δύσκολη και αντιπαραγωγική εργασία. Δύο από τις βασικότερες αιτίες είναι η δυσκολία απομνημόνευσης των κωδικών των εντολών και η δυσκολία να εντοπιστεί κάποιο πιθανό λάθος ανάμεσα σ' όλα αυτά τα 0 και 1. Γι' αυτό, από πολύ νωρίς, αναζητήθηκαν τρόποι να γράφονται τα προγράμματα σε γλώσσα πιο προσιπή στον άνθρωπο. Η επιδίωξη αυτή, οδήγησε στην ανάπτυξη των διαφόρων γλωσσών προγραμματισμού, καθώς επίσης και στην ανεύρεση τρόπων για την εκτέλεση από τον υπολογιστή των προγραμμάτων που γράφονται σ' αυτές τις γλώσσες.

Ο υπολογιστής μπορεί να πραγματοποιήσει τις εξής τρεις λειτουργίες:

- Πρόσθεση
- Σύγκριση
- Μεταφορά δεδομένων

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΣΥΜΒΟΛΙΚΕΣ ΓΛΩΣΣΕΣ

Μια πρώτη προσπάθεια για τη διευκόλυνση της γραφής προγραμμάτων ήταν η γραφή τους σε συμβολική γλώσσα(assembly). Στις συμβολικές γλώσσες, σε κάθε εντολή της γλώσσας μηχανής αντιστοιχίζεται μία μνημονική λέξη η οποία θυμίζει το σκοπό της εντολής. Επί πλέον δίνεται η δυνατότητα να χρησιμοποιούνται, στη θέση αριθμών, ονόματα που αντιπροσωπεύουν σταθερές ή διευθύνσεις μνήμης. Έτσι μπορούμε να γράφουμε ΗΛΙΚΙΑ αντί για 18. Η εντολή του προηγούμενου παραδείγματος της γλώσσας μηχανής του Z80 με τη μορφή **1110011010110011**, θα μπορούσε σε συμβολική γλώσσα να έχει τη μορφή: **ADD B3h**.

ΠΑΡΑΔΕΙΓΜΑ 1

Υπολογισμός αθροίσματος των αριθμών 1 έως και 10.

1. Η λύση σε QuickBasic	2. σε συμβολική γλώσσα	3. και σε γλώσσα μηχανής
sum=0 FOR index=1 TO 10 sum=sum+index NEXT index END	INDEX=\$01 SUM=\$02 LDA #10 STA INDEX CLA LOOP ADD INDEX DEC INDEX BNE INDEX STA SUM	10101000 00001010 10001100 00000001 00111100 01010001 00000001 01000011 00000001 11000000 00000001 11000000 11111010 10001100 00000010 11111111

ΠΑΡΑΔΕΙΓΜΑ 2

Έστω ότι θέλουμε να προσθέσουμε το περιεχόμενο δύο θέσεων μνήμης και το αποτέλεσμα να το καταχωρίσουμε σε μία τρίτη. Για λόγους σύγκρισης, ας δούμε πως θα μπορούσε αυτό να γραφεί σε γλώσσα υψηλού επιπέδου, σε συμβολική γλώσσα και σε γλώσσα μηχανής:

Γλώσσα υψηλού επιπέδου

Εντολή	Περιγραφή
A:= B+C	Πρόσθεσε το περιεχόμενο των μεταβλητών B και C και το αποτέλεσμα καταχώρισέ το στη μεταβλητή A.

Συμβολική γλώσσα

Εντολή	Περιγραφή
LDA B	Μετάφερε στο συσσωρευτή το περιεχόμενο της θέσης μνήμης με όνομα B.
ADD C	Πρόσθεσε στο περιεχόμενο του συσσωρευτή το περιεχόμενο της θέσης μνήμης με όνομα C.
STA A	Μετάφερε και αποθήκευσε το περιεχόμενο του συσσωρευτή στη θέση μνήμης με όνομα A.

Γλώσσα μηχανής

Εντολή	Περιγραφή
000001001011010	Μετάφερε στο συσσωρευτή το περιεχόμενο της θέσης μνήμης με διεύθυνση 01011010.
0000101001011110	Πρόσθεσε στο περιεχόμενο του συσσωρευτή το περιεχόμενο της θέσης μνήμης με διεύθυνση 01011110.
0000011011011110	Μετάφερε και αποθήκευσε το περιεχόμενο του συσσωρευτή στη θέση μνήμης με διεύθυνση 11011110.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

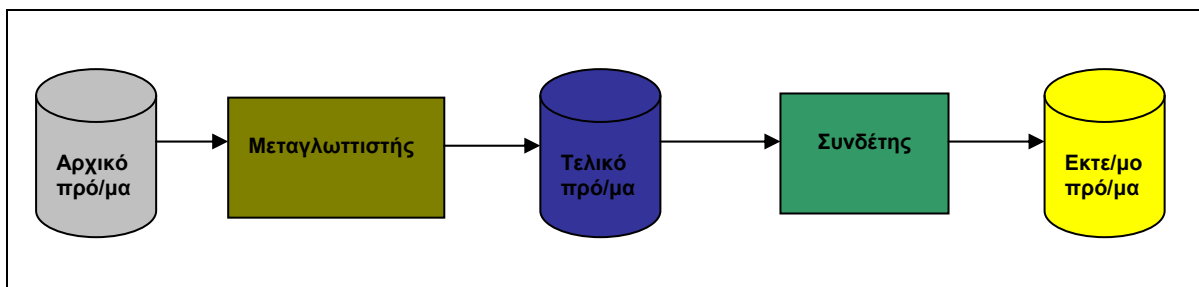
Βασικές γνώσεις

ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ – ΔΙΕΡΜΗΝΕΥΤΕΣ

Για να εκτελεσθεί από τον υπολογιστή ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου, απαιτείται η μετατροπή του σε αντίστοιχο πρόγραμμα σε γλώσσα μηχανής. Η μετατροπή αυτή γίνεται με τη βοήθεια ενδιάμεσων προγραμμάτων-μεταφραστών. Υπάρχουν δύο μεγάλες κατηγορίες μεταφραστικών προγραμμάτων, οι **μεταγλωττιστές**(compilers) και οι **διερμηνευτές**(interpreters). Ο μεταγλωττιστής δέχεται στην είσοδο ένα πρόγραμμα γραμμένο σε μία γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα γραμμένο σε γλώσσα μηχανής. Το τελευταίο μπορεί στη συνέχεια να εκτελεστεί οποτεδήποτε και είναι τελείως ανεξάρτητο από το αρχικό πρόγραμμα. Αντίθετα ο διερμηνευτής διαβάζει μία-μία τις εντολές του αρχικού προγράμματος και για κάθε μία εκτελεί αμέσως μία ισοδύναμη ακολουθία εντολών, σε γλώσσα μηχανής. Άλλες γλώσσες υλοποιούνται με μεταγλωττιστή (π.χ. COBOL, FORTRAN, Pascal, C κ.λ.π.), άλλες με διερμηνευτή (π.χ. APL, LISP, PROLOG) και άλλες και με τους δύο (π.χ. BASIC).

Η χρήση του μεταγλωττιστή έχει το μειονέκτημα ότι προτού χρησιμοποιηθεί ένα πρόγραμμα, πρέπει να περάσει από τη διαδικασία της μεταγλώττισης. Κατά την φάση αυτή, αν εντοπισθούν ορθογραφικά ή συντακτικά λάθη, αλλά και μετά την εκτέλεση, αν προκύψουν σφάλματα, τότε το αρχικό πρόγραμμα πρέπει να διορθωθεί και να ξαναμεταγλωττισθεί.

Ένα πρόγραμμα γραμμένο σε μία γλώσσα υψηλού επιπέδου αποκαλείται **αρχικό ή πηγαίο**. Το πρόγραμμα που παράγεται από τον μεταγλωττιστή λέγεται **τελικό** πρόγραμμα. Συνήθως το τελικό πρόγραμμα δεν είναι ακόμη έτοιμο προς εκτέλεση. Απαιτείται επιπλέον και η διαδικασία της **σύνδεσης**(linking), κατά την οποία από το τελικό πρόγραμμα, που παράγει ο μεταγλωττιστής, δημιουργείται το **εκτελέσιμο** πρόγραμμα(executable). Τη σύνδεση την πραγματοποιεί ο συνδέτης(linker), που είναι και αυτό ένα πρόγραμμα σε γλώσσα χαμηλού επιπέδου. Η διαδικασία της σύνδεσης ακολουθεί πάντα τη μεταγλώττιση, εφόσον δεν υπάρχουν λάθη στο αρχικό πρόγραμμα. Για το λόγο αυτό η συνολική διαδικασία αποκαλείται μεταγλώττιση και σύνδεση(compilation and linking) και παρουσιάζεται σχηματικά παρακάτω:



Στη δεκαετία του 70 παρουσιάζεται για πρώτη φορά η ιδέα του αντικειμενοστραφούς προγραμματισμού(Object Oriented Programming-OOP). Η ιδέα αυτή βρήκε πρόσφορο έδαφος αργότερα και εξελίχθηκε σημαντικά, αλλάζοντας τον παραδοσιακό προγραμματισμό με γλώσσες τρίτης γενιάς. Αργότερα στις αρχές της δεκαετίας του 90 εμφανίζονται τα γραφικά περιβάλλοντα εργασίας. Η ανάπτυξη αυτών δημιούργησε την ανάγκη για δημιουργία προγραμμάτων που θα εκμεταλλεύονται το γραφικό τρόπο επικοινωνίας χρήστη-υπολογιστή. Έτσι παρουσιάστηκαν νέες γλώσσες ή νεότερες εκδόσεις γνωστών γλωσσών προγραμματισμού, που υλοποιούν τις έννοιες του οδηγούμενου από το γεγονός προγραμματισμού(event driven programming) και του οπτικού προγραμματισμού(visual programming).

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Την τελευταία δεκαετία γνωρίζει μεγάλη διάδοση ο λεγόμενος **αντικειμενοστραφής προγραμματισμός**, ο οποίος βασίζεται στην ιδέα της δόμησης προγραμμάτων ως σύνολο αντικειμένων που υλοποιούνται προγραμματιστικά. Έτσι στον αντικειμενοστραφή προγραμματισμό, έννοια βασικής σημασίας είναι το **αντικείμενο**(object). Ένα αντικείμενο εξωτερικά μπορεί να ειπωθεί ως ένα <<πακέτο>>, το οποίο συμπεριλαμβάνει πληροφορίες που το χαρακτηρίζουν και ενέργειες που μπορούν να γίνουν μ' αυτό.

Οι πληροφορίες που το χαρακτηρίζουν και είναι προσπελάσιμες εξωτερικά λέγονται **ιδιότητες**(properties), ενώ οι επιτρεπτές ενέργειες **μέθοδοι**(methods). Φυσικά, εσωτερικά ένα

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

αντικείμενο στην τυπική περίπτωση, έχει εφοδιαστεί από τον προγραμματιστή του με επιπλέον μεταβλητές και ρουτίνες, οι οποίες του επιτρέπουν να επιτελεί το έργο για το οποίο δημιουργήθηκε. Εξωτερικά όμως όσο αφορά το χειρισμό και τη χρήση του, αρκεί κανείς να γνωρίζει τις ιδιότητες και τις μεθόδους του αντικειμένου.

Στον αντικειμενοστραφή προγραμματισμό, ο προγραμματιστής χρησιμοποιώντας συγκεκριμένες για κάθε γλώσσα εντολές, αρχικά δημιουργεί τις λεγόμενες **κλάσεις** αντικειμένων(classes). Στο σημείο του ορισμού των κλάσεων, ορίζονται οι μεταβλητές και οι ρουτίνες που θα περιέχει το αντικείμενο, καθώς και ποιες από αυτές θα είναι προσπελάσιμες εξωτερικά και ποιες όχι. Ο ορισμός μιας κλάσης αντικειμένου στην ουσία αποτελεί ένα καλούπι, από το οποίο μετά δημιουργείται το αντίστοιχο αντικείμενο, σε όποιο σημείο του προγράμματος χρειαστεί. Στη συνέχεια γίνεται ο χειρισμός του μέσω των μεθόδων και των ιδιοτήτων του, όπως αυτές έχουν οριστεί κατά τον ορισμό της αντίστοιχης κλάσης.

ΟΔΗΓΟΥΜΕΝΟΣ ΑΠΟ ΓΕΓΟΝΟΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Όπως είναι γνωστό, με την λέξη <<γεγονός>> εννοούμε κάτι που έγινε, που συνέβη. Για παράδειγμα το κλικ σ' ένα κουμπί του ποντικιού είναι ένα γεγονός. Γεγονότα στον προγραμματισμό μπορούν να δημιουργηθούν είτε από μία ενεργεία του χρήστη (π.χ. κλικ στο ποντίκι) είτε από το σύστημα (π.χ. αν εμφανιστεί κάποιο λάθος στο γράψιμο μιας δισκέτας). Όταν δημιουργείται μια κλάση σε μια γλώσσα προγραμματισμού, όπως για παράδειγμα ένα κουμπί των Windows, μπορούμε να αντιστοιχίσουμε κώδικα σ' ένα γεγονός.

ΟΠΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Μπορεί κανείς να δημιουργήσει κλάσεις και να χειριστεί αντικείμενα απλά γράφοντας κώδικα. Στα σύγχρονα συστήματα προγραμματισμού, τα οποία κατά κανόνα είναι προσαρμοσμένα στον προγραμματισμό για γραφικά περιβάλλοντα χρήστη, δίνεται επιπλέον η δυνατότητα δημιουργίας κλάσεων αντικειμένων μέσω ειδικών εργαλείων.

Η γενική διαδικασία στον οπτικό προγραμματισμό είναι σχετικά απλή. Βασικά συνίσταται στην ενεργοποίηση του εργαλείου που παρέχεται από το σύστημα ανάπτυξης που χρησιμοποιείται. Με την ενεργοποίηση του εργαλείου, τυπικά εμφανίζεται ένα δειγματολόγιο των αντικειμένων που διατίθενται. Από τα αντικείμενα αυτά, επιλέγεται το κατάλληλο κάθε φορά με κλικ πάνω του και στη συνέχεια σχεδιάζεται στην επιθυμητή θέση και μέγεθος. Με κάποιο χειρισμό, π.χ. διπλό κλικ πάνω στο σχεδιασθέν αντικείμενο, εμφανίζεται παράθυρο που περιέχει όλες τις ιδιότητες π.χ. χρώμα, ετικέτα κ.λ.π. και όλες τις μεθόδους π.χ. γεγονός κλικ του ποντικιού κ.λ.π. του αντικειμένου.

Δεν θα πρέπει να συγχέεται η έννοια <<αντικειμενοστραφής προγραμματισμός>>, με τον οδηγούμενο από γεγονότα προγραμματισμό και τον οπτικό προγραμματισμό. Ο αντικειμενοστραφής προγραμματισμός είναι μία τεχνολογία προγραμματισμού, που χρησιμοποιείται σήμερα κατά κόρον, επειδή με τη χρήση του επιτυγχάνεται ανάπτυξη προγραμμάτων που μπορούν να συντηρηθούν εύκολα, ενώ οι διάφορες κλάσεις που αποτελούν τα δομικά στοιχεία μίας εφαρμογής που έχει δημιουργηθεί με αντικειμενοστραφή προγραμματισμό, μπορούν να επαναχρησιμοποιηθούν με σχετική ευκολία σε άλλες εφαρμογές(reusability). Ο αντικειμενοστραφής προγραμματισμός χρησιμοποιείται ευρέως για ανάπτυξη εφαρμογών στα σύγχρονα γραφικά περιβάλλοντα χρήστη. Αυτό επειδή ως φιλοσοφία ταιριάζει στη φύση των περιβαλλόντων αυτών: επαναχρησιμοποιούμενα αντικείμενα όπως κουμπιά, πλαίσια ελέγχου, κουμπιά επιλογής κ.λ.π. γεγονότα όπως κλικ του ποντικιού, σύρσιμο του ποντικιού κ.λ.π. Ο οπτικός προγραμματισμός είναι μία δυνατότητα που προσφέρουν τα σύγχρονα περιβάλλοντα προγραμματισμού, μέσω της οποίας επιτυγχάνεται η ανάπτυξη κυρίως του τμήματος της εφαρμογής που ασχολείται με την επικοινωνία με το χρήστη.



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Δεδομένα είναι τα στοιχεία που συλλέγουμε χωρίς να έχουν υποστεί καμία επεξεργασία. Οι αλγόριθμοι παίρνουν τα δεδομένα τα επεξεργάζονται και παράγουν πληροφορίες. Για να μπορούμε να αποθηκεύουμε τα δεδομένα που μαζεύουμε και στη συνέχεια να τα επεξεργαστούμε χρησιμοποιούμε τις δομές δεδομένων. **Δομή δεδομένων** είναι ένα σύνολο αποθηκευμένων δεδομένων που μπορούν να υφίστανται επεξεργασία από ένα σύνολο λειτουργιών. Οι λειτουργίες αυτές είναι:

- Εισαγωγή: *Να βάλουμε νέα δεδομένα.*
Διαγραφή: *Να σβήσουμε κάποια δεδομένα.*
Προσπέλαση: *Να <<δούμε>> τα δεδομένα που έχουμε ή αν θέλουμε να τα τροποποιήσουμε.*
Αντιγραφή: *Να αντιγράψουμε κάποια δεδομένα σε μια άλλη δομή.*
Αναζήτηση: *Να βρούμε αν ένα συγκεκριμένο δεδομένο υπάρχει ή όχι μέσα στη δομή που επεξεργαζόμαστε.*
Ταξινόμηση: *Να ταξινομήσουμε τα δεδομένα της δομής.*
Συγχώνευση: *Να ενώσουμε δύο ή περισσότερες δομές σε μία.*
Διαχωρισμός: *Να χωρίσουμε μια δομή σε δύο ή περισσότερες νέες δομές.*

Οι δομές χωρίζονται σε δύο βασικές κατηγορίες:

Δυναμικές δομές δεδομένων, οι οποίες έχουν τα παρακάτω χαρακτηριστικά:

- Μπορούν και αλλάζουν μέγεθος κατά την εκτέλεση του προγράμματος.
- Δεν καταλαμβάνουν συγκεκριμένο χώρο μνήμης αλλά συνεχώς τους παραχωρείται μνήμη αναλόγως με τις ανάγκες τους σε χώρο.
- Τα στοιχεία τους δεν αποθηκεύονται αναγκαστικά σε συνεχόμενες θέσεις μνήμης.

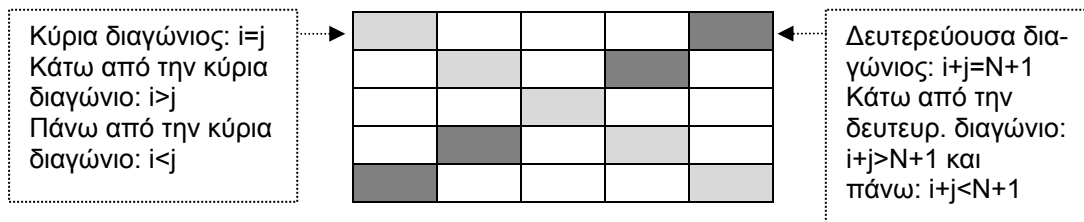
Στατικές δομές δεδομένων, οι οποίες έχουν τα παρακάτω χαρακτηριστικά:

- Έχουν σταθερό μέγεθος κατά τη διάρκεια της εκτέλεσης του προγράμματος.
 - Καταλαμβάνουν συγκεκριμένο χώρο μνήμης.
 - Τα στοιχεία τους αποθηκεύονται πάντα σε συνεχόμενες θέσεις μνήμης.
- Παράδειγμα στατικής δομής είναι οι πίνακες που περιέχουν στοιχεία ίδιου τύπου (ακέραιους, χαρακτήρες κ.τ.λ.). Οι πίνακες χρησιμοποιούνται όταν θέλουμε να χειριστούμε ένα μεγάλο πλήθος δεδομένων.

ΠΙΝΑΚΕΣ

Έχουμε τους **μονοδιάστατους** και τους **δισδιάστατους** πίνακες. Οι μονοδιάστατοι πίνακες είναι αυτοί που αποτελούνται από μια γραμμή και πολλές στήλες ή από μία στήλη και πολλές γραμμές. Για να αναφερθούμε σ' ένα στοιχείο του πίνακα χρησιμοποιούμε το όνομα του πίνακα και το δείκτη που δείχνει σε ποια θέση είναι το συγκεκριμένο στοιχείο.

Οι δισδιάστατοι πίνακες είναι αυτοί που αποτελούνται από πολλές γραμμές και πολλές στήλες. Στους πίνακες αυτούς έχουμε δύο δείκτες τον πρώτο για τις γραμμές και τον δεύτερο για τις στήλες. Όταν ο πίνακας έχει ίδιο αριθμό γραμμών και στηλών ονομάζεται τετραγωνικός.



Σημαντικό είναι να γνωρίζουμε και την έννοια του παράλληλου πίνακα.

Παράλληλοι ονομάζονται δύο ή περισσότεροι πίνακες, όταν σ' αυτούς έχουμε αποθηκεύσει τα χαρακτηριστικά οντοτήτων με τέτοιο τρόπο, ώστε τα δεδομένα κάθε οντότητας να βρίσκονται σε στοιχεία με την ίδια τιμή δείκτη.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

ΠΑΡΑΔΕΙΓΜΑ 1

Να γράψετε πρόγραμμα που να συγχωνεύει τους πίνακες A[28] και B[32] σ' έναν πίνακα.

Λύση

ΠΡΟΓΡΑΜΜΑ Συγχώνευση
ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: i, A[28], B[32], Γ[60]

ΑΡΧΗ

ΓΡΑΨΕ 'Δώστε τα στοιχεία του πίνακα A: '

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 28

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώστε τα στοιχεία του πίνακα B: '

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 32

ΔΙΑΒΑΣΕ B[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Τοποθετούμε στον πίνακα Γ τα στοιχεία του πίνακα A και μετά τα στοιχεία του πίνακα B

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 28

Γ[i] ← A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 32

Γ[i+28] ← B[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 60

ΓΡΑΨΕ Γ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΑΡΑΔΕΙΓΜΑ 2

Να γράψετε πρόγραμμα που να διαχωρίζει τον πίνακα ΑΡΙΘΜΟΙ (περιέχει 100 ακέραιους αριθμούς) σε άρτιους και περιττούς, δηλαδή να δημιουργεί δύο νέους πίνακες.

Λύση

ΠΡΟΓΡΑΜΜΑ Διαχωρισμός
ΣΤΑΘΕΡΕΣ

N=100

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, sum1, sum2, ΑΡΙΘΜΟΙ[N], ΑΡΤΙΟΙ[N], ΠΕΡΙΤΤΟΙ[N]

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 100

ΓΡΑΨΕ 'Δώστε το 'i,' στοιχείο του πίνακα: '

ΔΙΑΒΑΣΕ ΑΡΙΘΜΟΙ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

sum1 ← 0

sum2 ← 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΑΝ (ΑΡΙΘΜΟΙ[i] **MOD** 2)=0 **ΤΟΤΕ**

sum1 ← sum1+1

ΑΡΤΙΟΙ[sum1] ← ΑΡΙΘΜΟΙ[i]

ΑΛΛΙΩΣ

sum2 ← sum2+1

ΠΕΡΙΤΤΟΙ[sum2] ← ΑΡΙΘΜΟΙ[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΑΣΚΗΣΕΙΣ ΜΕ ΔΙΑΣΤΗΜΑΤΑ ΤΙΜΩΝ

Σ' αυτές τις ασκήσεις κάποιες μεταβλητές δεν έχουν σταθερή τιμή αλλά παίρνουν κλιμακωτά τιμές. Γι' αυτό προσέχουμε πολύ τα διαστήματα τιμών που έχουν τεθεί καθώς επίσης και το πώς θα υπολογίσουμε την τελική τους τιμή. Για παράδειγμα:

Ένας δήμος ακολουθεί την εξής τιμολογιακή πολιτική για την κατανάλωση νερού ανά μήνα: χρεώνει πάγιο ποσό 2 ευρώ και εφαρμόζει κλιμακωτή χρέωση σύμφωνα με τον παρακάτω πίνακα:

Κατανάλωση κ.μ.	Χρέωση / κ.μ.
από 0 έως και 5	0,1 ευρώ
από 5 έως και 10	0,4 ευρώ
από 10 έως και 20	0,7 ευρώ
από 20 και άνω	1,0 ευρώ

Στο ποσό που προκύπτει από την αξία του νερού και το πάγιο υπολογίζεται ο Φ.Π.Α. με συντελεστή 18%. Το τελικό ποσό προκύπτει από την άθροιση της αξίας του νερού, το πάγιο, το Φ.Π.Α. και το δημοτικό φόρο που είναι 5 ευρώ.

Να αναπτύξετε αλγόριθμο ή πρόγραμμα το οποίο:

A. Να διαβάζει τη μηνιαία κατανάλωση νερού.

B. Να υπολογίζει την αξία του νερού που καταναλώθηκε σύμφωνα με την παραπάνω τιμολογιακή πολιτική.

Γ. Να υπολογίζει το Φ.Π.Α.

Δ. Να υπολογίζει και εκτυπώνει το τελικό ποσό.

Αλγόριθμος Κατανάλωση_νερού

Αρχή_επανάληψης

Εμφάνισε 'Δώστε τη μηνιαία κατανάλωση νερού: '

Διάβασε MK

Μέχρις_ότου MK ≥ 0

ΚΛΙΜΑΚΩΤΗ ΧΡΕΩΣΗ

Αν MK ≤ 5 **τότε**

Αξία ← 0.1 * MK

αλλιώς_αν MK ≤ 10 **τότε**

Αξία ← 0.1 * 5 + 0.4 * (MK - 5)

αλλιώς_αν MK ≤ 20 **τότε**

Αξία ← 0.1 * 5 + 0.4 * 5 + 0.7 * (MK - 10)

αλλιώς

Αξία ← 0.1 * 5 + 0.4 * 5 + 0.7 * 10 + 1.0 * (MK - 20)

Τέλος_αν

ΦΠΑ ← (Αξία + 2) * 0.18

Τελικό_Ποσό ← Αξία + 2 + ΦΠΑ + 5

από 0 έως και 5	0,1 ευρώ
-----------------	----------

από 0 έως και 5	0,1 ευρώ
από 5 έως και 10	0,4 ευρώ

από 0 έως και 5	0,1 ευρώ
από 5 έως και 10	0,4 ευρώ
από 10 έως και 20	0,7 ευρώ

από 0 έως και 5	0,1 ευρώ
από 5 έως και 10	0,4 ευρώ
από 10 έως και 20	0,7 ευρώ
από 20 και άνω	1,0 ευρώ

Εκτύπωσε Τελικό_Ποσό
Τέλος Κατανάλωση_νερού

Π.χ. Έστω MK=100 τότε:

0..5 χρέωση → 5*0,1= 0,5

5..10 χρέωση → 5*0,4= 2

10..20 χρέωση → 10*0,7= 7

80..100 χρέωση → 80*1 = 80

Σύνολο = 89.5

ΦΠΑ ← (89.5+2)*0.18

Τελικό_Ποσό ← (89.5+2)+16.47+5

Εκτύπωσε 112.97

2^{ος} τρόπος

Αν MK > 20 **τότε**

Αξία ← 0.1 * 5 + 0.4 * 5 + 0.7 * 10 + 1.0 * (MK - 20)

αλλιώς_αν MK ≥ 10 **τότε**

Αξία ← 0.1 * 5 + 0.4 * 5 + 0.7 * (MK - 10)

αλλιώς_αν MK ≥ 5 **τότε**

Αξία ← 0.1 * 5 + 0.4 * (MK - 5)

αλλιώς

Αξία ← 0.1 * MK

Τέλος_αν

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

Δεν λύνονται όλες οι ασκήσεις με διαστήματα τιμών όπως η προηγούμενη άσκηση. Υπάρχουν ασκήσεις όπου οι μεταβλητές δεν παίρνουν κλιμακωτά τιμές. Για παράδειγμα:

Μία εταιρία ταχυδρομικών υπηρεσιών εφαρμόζει για τα έξοδα αποστολής ταχυδρομικών επιστολών εσωτερικού και εξωτερικού, χρέωση σύμφωνα με τον παρακάτω πίνακα:

Βάρος επιστολής σε γραμμάρια	Χρέωση εσωτερικού σε Ευρώ	Χρέωση εξωτερικού σε Ευρώ
Από 0 έως και 500	2,0	4,8
Από 500 έως και 1000	3,5	7,2
Από 100 έως και 2000	4,6	11,5

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

Να γράψετε αλγόριθμο ο οποίος:

- α. Να διαβάζει το βάρος της επιστολής.
- β. Να διαβάζει τον προορισμό της επιστολής. Η τιμή "ΕΣ" δηλώνει προορισμό εσωτερικού και η τιμή "ΕΞ" δηλώνει προορισμό εξωτερικού.
- γ. Να υπολογίζει τα έξοδα αποστολής ανάλογα με τον προορισμό και το βάρος της επιστολής.
- δ. Να εκτυπώνει τα έξοδα αποστολής.

Παρατήρηση: Θεωρείστε ότι ο αλγόριθμος δέχεται τιμές για το βάρος μεταξύ του 0 και του 2000 και για τον προορισμό μόνο τις τιμές "ΕΣ" και "ΕΞ".

Λύση

Αλγόριθμος Θέμα_3_2004

Εμφάνισε ' Δώστε το βάρος: (θετικός αριθμός) '

Διάβασε βάρος

Εμφάνισε ' Δώστε τον προορισμό: ΕΣΩΤΕΡΙΚΟΥ (ΕΣ) ή ΕΞΩΤΕΡΙΚΟΥ (ΕΞ) '

Διάβασε προορισμός

Αν προορισμός= 'ΕΣ' **τότε**

Αν βάρος≤500 **τότε**

Χρέωση← 2

αλλιώς_αν βάρος≤1000 **τότε**

Χρέωση← 3.5

αλλιώς

Χρέωση← 4.6

Τέλος_αν

αλλιώς

Αν βάρος≤500 **τότε**

Χρέωση← 4.8

αλλιώς_αν βάρος≤1000 **τότε**

Χρέωση← 7.2

αλλιώς

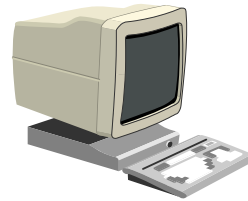
Χρέωση← 11.5

Τέλος_αν

Τέλος_αν

Εμφάνισε 'Τα έξοδα αποστολής είναι: ', Χρέωση, ' Ευρώ'

Τέλος Θέμα_3_2004



Υπάρχει βέβαια και η περίπτωση όπου απαιτείται συνδυασμός των δύο προηγούμενων μεθόδων.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΠΙΝΑΚΑΣ ΤΙΜΩΝ

Ο πίνακας τιμών είναι χρήσιμος όταν θέλουμε να παρακολουθήσουμε με λεπτομέρεια τις αλλαγές στις τιμές των μεταβλητών ενός αλγορίθμου. Ακολουθεί χαρακτηριστικό παράδειγμα.

Άσκηση

Δίνεται μονοδιάστατος πίνακας A, 10 θέσεων, ο οποίος στις θέσεις 1 έως 10 περιέχει αντίστοιχα τους αριθμούς: 15, 3, 0, 5, 16, 2, 17, 8, 19, 1 και τμήμα αλγορίθμου:

Για i από 1 μέχρι 9 με_βήμα 2

$k \leftarrow ((i+10) \bmod 10) + 1$

$A[i] \leftarrow A[k]$

Εμφάνισε i, k, A[i], A[k]

Τέλος_επανάληψης

Ποιες είναι οι τιμές του i, k, A[i], A[k] σε κάθε επανάληψη;

Λύση

1^η επανάληψη (i=1)

$k = (11 \bmod 10) + 1 = 2$

$A[1] = A[2] = 3$

Άρα:

i	k	A[i]	A[k]
1	2	3	3

2^η επανάληψη (i=3)

$k = (13 \bmod 10) + 1 = 4$

$A[3] = A[4] = 5$

Άρα:

i	k	A[i]	A[k]
1	2	3	3
3	4	5	5

3^η επανάληψη (i=5)

$k = (15 \bmod 10) + 1 = 6$

$A[5] = A[6] = 2$

i	k	A[i]	A[k]
1	2	3	3
3	4	5	5
5	6	2	2

κ.τ.λ.

Η τελική μορφή του πίνακα τιμών είναι:

i	k	A[i]	A[k]
1	2	3	3
3	4	5	5
5	6	2	2
7	8	8	8
9	10	1	1

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

DIV και MOD

Το **div** μας δίνει το ακέραιο πηλίκο της διαίρεσης ενώ το **mod** το ακέραιο υπόλοιπο. Για παράδειγμα στην διαίρεση $18/7 = 2*7+4$, το ακέραιο πηλίκο είναι δύο ($18 \text{ div } 7=2$) και το ακέραιο υπόλοιπο είναι 4 ($18 \text{ mod } 7=4$).

Κυρίως χρησιμοποιούνται όταν θέλουμε:

- Να πάρουμε μεμονωμένα τα ψηφία ενός αριθμού.
Π.χ. Έστω x ένας τριψήφιος αριθμός τότε:
Το τελευταίο ψηφίο του αριθμού είναι: $x \text{ mod } 10$
Τα δύο τελευταία ψηφία του αριθμού είναι: $x \text{ mod } 100$
Το πρώτο ψηφίο είναι: $x \text{ div } 100$
- Να βρούμε τα πολλαπλάσια ενός αριθμού.
Π.χ. Το x είναι πολλαπλάσιο του 5 αν $x \text{ mod } 5=0$
- Να ελέγξουμε αν ένας αριθμός είναι άρτιος ή περιττός.

Ακολουθούν δύο παραδείγματα.

ΠΑΡΑΔΕΙΓΜΑ 1

Να δοθεί αλγόριθμος ο οποίος θα διαβάσει έναν αριθμό και θα υπολογίσει αν είναι άρτιος ή περιττός.

Λύση

Αλγόριθμος άρτιος
Εμφάνισε 'Δώστε τον αριθμό: '
Διάβασε x
Αν $(x \text{ mod } 2)=0$ **τότε**
Εμφάνισε 'Ο x είναι άρτιος '
αλλιώς
Εμφάνισε 'Ο x είναι περιττός '
Τέλος_αν
Τέλος άρτιος

ΠΑΡΑΔΕΙΓΜΑ 2

Να δοθεί αλγόριθμος ο οποίος να υπολογίζει και τυπώνει το άθροισμα των ψηφίων ενός θετικού τριψήφιου αριθμού.

Λύση

Αλγόριθμος άθροισμα
Εμφάνισε 'Δώστε έναν τριψήφιο αριθμό: '
Διάβασε αριθμός
Όσο (αριθμός>999) **ή** (αριθμός<100) **επανάλαβε**
Εμφάνισε ' Δώσατε λάθος τριψήφιο αριθμό, ξαναπροσπαθήστε '
Διάβασε αριθμός
Τέλος_επανάληψης

$psif1 \leftarrow$ αριθμός **div** 100
 $υπόλοιπο \leftarrow$ αριθμός **mod** 100
 $psif2 \leftarrow$ υπόλοιπο **div** 10
 $psif3 \leftarrow$ υπόλοιπο **mod** 10
 $sum \leftarrow psif1+psif2+psif3$

Εμφάνισε ' Το άθροισμα του τριψήφιου αριθμού είναι: ',sum
Τέλος άθροισμα

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

Οι δομές επανάληψης είναι οι ακόλουθες τρεις:

❖ **ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...ΜΕ ΒΗΜΑ...**

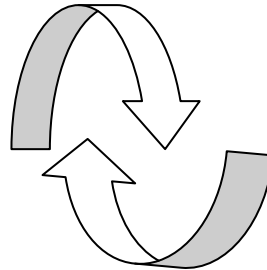
.....
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

❖ **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ**

.....
ΜΕΧΡΙΣ_ΟΤΟΥ <συνθήκη>

❖ **ΌΣΟ** <συνθήκη> **ΕΠΑΝΑΛΑΒΕ**

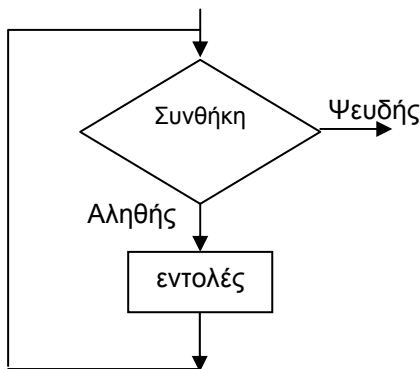
.....
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ



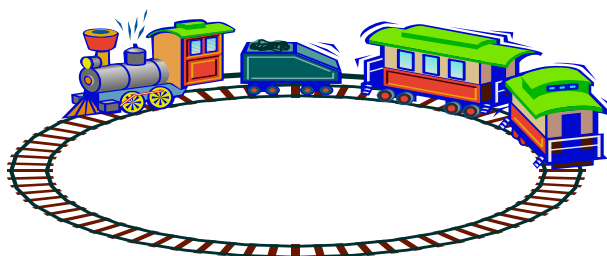
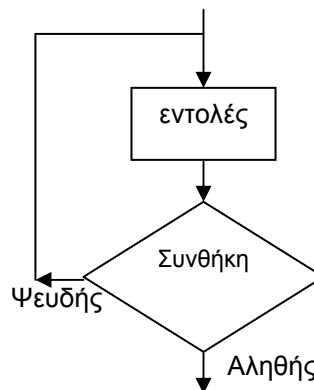
ΠΑΡΑΤΗΡΗΣΕΙΣ

- Η δομή **ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...ΜΕ ΒΗΜΑ...** είναι η μόνη στην οποία γνωρίζουμε εκ των προτέρων το πλήθος των επαναλήψεων και αυτό γιατί ξέρουμε την αρχική και την τελική τιμή καθώς και το βήμα της μεταβλητής επανάληψης. Στις άλλες δύο δομές το πλήθος των επαναλήψεων καθορίζεται από την τιμή της συνθήκης.
- Στη δομή **ΌΣΟ** <συνθήκη> **ΕΠΑΝΑΛΑΒΕ** πρώτα ελέγχεται η συνθήκη και μετά εκτελούνται οι εντολές, επίσης η επανάληψη τερματίζει όταν η συνθήκη γίνει Ψευδής.
- Στη δομή **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ** <συνθήκη> πρώτα εκτελούνται οι εντολές και μετά ελέγχεται η συνθήκη, επίσης η επανάληψη τερματίζει όταν η συνθήκη γίνει Αληθής.

ΌΣΟ <συνθήκη> **ΕΠΑΝΑΛΑΒΕ**



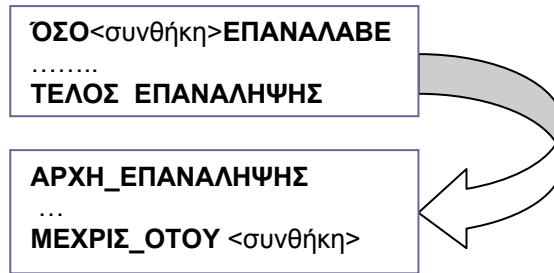
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ... ΜΕΧΡΙΣ_ΟΤΟΥ <συνθήκη>



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

ΜΕΤΑΤΡΟΠΕΣ ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ

- Μετατροπή από **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ** σε **ΜΕΧΡΙΣ_ΟΤΟΥ**



ΤΙ ΚΑΝΟΥΜΕ:

1. Αντιστρέφουμε την συνθήκη, διότι η δομή **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ** εκτελείται όσο είναι Αληθής ενώ η **ΜΕΧΡΙΣ_ΟΤΟΥ** εκτελείται όσο είναι Ψευδής.

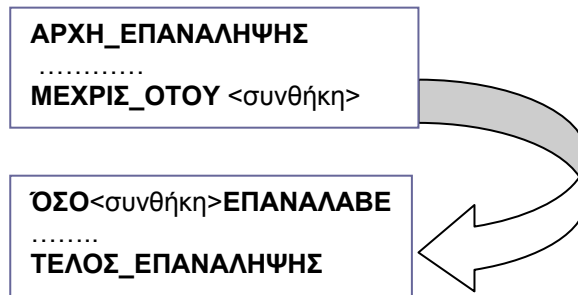
Άρα:

ΌΣΟ...ΕΠΑΝΑΛΑΒΕ		ΜΕΧΡΙΣ_ΟΤΟΥ
>	→	<=
=	→	<>
<	→	>=
>=	→	<
<=	→	>
<>	→	=

2. Ελέγχουμε το πλήθος των επαναλήψεων στις δύο δομές.

Στη δομή επανάληψης **ΜΕΧΡΙΣ_ΟΤΟΥ** πρώτα εκτελούνται οι εντολές και μετά γίνεται ο έλεγχος της συνθήκης, άρα εκτελείται τουλάχιστον μία φορά η δομή είτε ισχύει η συνθήκη είτε όχι. Η δομή **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ** μπορεί να μην εκτελεσθεί αν δεν ισχύει η συνθήκη.

- Μετατροπή από **ΜΕΧΡΙΣ_ΟΤΟΥ** σε **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ**



ΤΙ ΚΑΝΟΥΜΕ:

1. Αντιστρέφουμε την συνθήκη, διότι η δομή **ΜΕΧΡΙΣ_ΟΤΟΥ** εκτελείται όσο είναι Ψευδής ενώ η **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ** όσο είναι Αληθής.

Άρα:

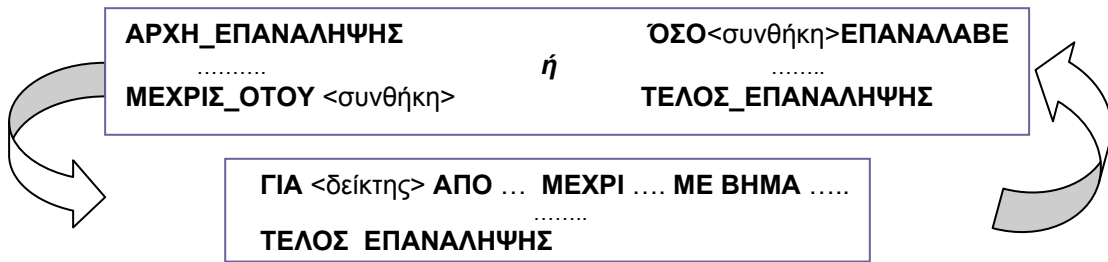
ΜΕΧΡΙΣ_ΟΤΟΥ		ΌΣΟ...ΕΠΑΝΑΛΑΒΕ
>	→	<=
=	→	<>
<	→	>=
>=	→	<
<=	→	>
<>	→	=

2. Ελέγχουμε το πλήθος των επαναλήψεων στις δύο δομές.

Η δομή επανάληψης **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ** μπορεί να μην εκτελεσθεί αν δεν ισχύει η συνθήκη. Στη δομή **ΜΕΧΡΙΣ_ΟΤΟΥ** πρώτα εκτελούνται οι εντολές και μετά γίνεται ο έλεγχος της συνθήκης, άρα εκτελείται τουλάχιστον μία φορά η δομή είτε ισχύει η συνθήκη είτε όχι.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
 Βασικές γνώσεις

- Μετατροπή από **ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ** σε **ΌΣΟ** ή σε **ΜΕΧΡΙΣ..ΟΤΟΥ**



ΤΙ ΠΡΟΣΕΧΟΥΜΕ:

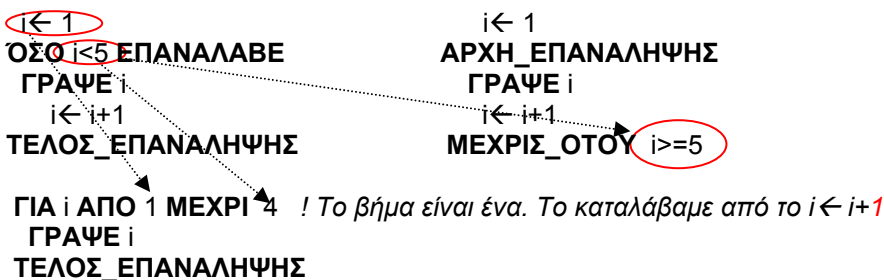
1. Όταν μετατρέπουμε τη δομή επανάληψης **ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ** σε **ΌΣΟ** ή σε **ΜΕΧΡΙΣ..ΟΤΟΥ** δεν ξεχνούμε: **1)** Να βάλουμε μια *μεταβλητή επανάληψης* (π.χ. *i*) η οποία θα μεταβάλλεται ανάλογα με το βήμα της δομής **ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ...ΜΕ ΒΗΜΑ...** (το βήμα σε περίπτωση που δε δοθεί είναι ένα). **2)** Να δώσουμε στη μεταβλητή αυτή αρχική τιμή. **3)** Να ορίσουμε μία συνθήκη για την μεταβλητή επανάληψης που θα μας εξασφαλίζει ίδιο αριθμό επαναλήψεων.

2. Όταν μετατρέπουμε τις δομές επανάληψης **ΌΣΟ** και **ΜΕΧΡΙΣ..ΟΤΟΥ** σε **ΓΙΑ.. ΑΠΟ.. ΜΕΧΡΙ.. ΜΕ ΒΗΜΑ...** ελέγχουμε: **1)** Ποια είναι η αρχική και η τελική τιμή της μεταβλητής επανάληψης. **2)** Ποιο είναι το βήμα της μεταβλητής αυτής.

ΠΑΡΑΔΕΙΓΜΑ 1

Να μετατραπεί η δομή επανάληψης **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ** σε **ΜΕΧΡΙΣ..ΟΤΟΥ** και **ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ..ΜΕ ΒΗΜΑ**.

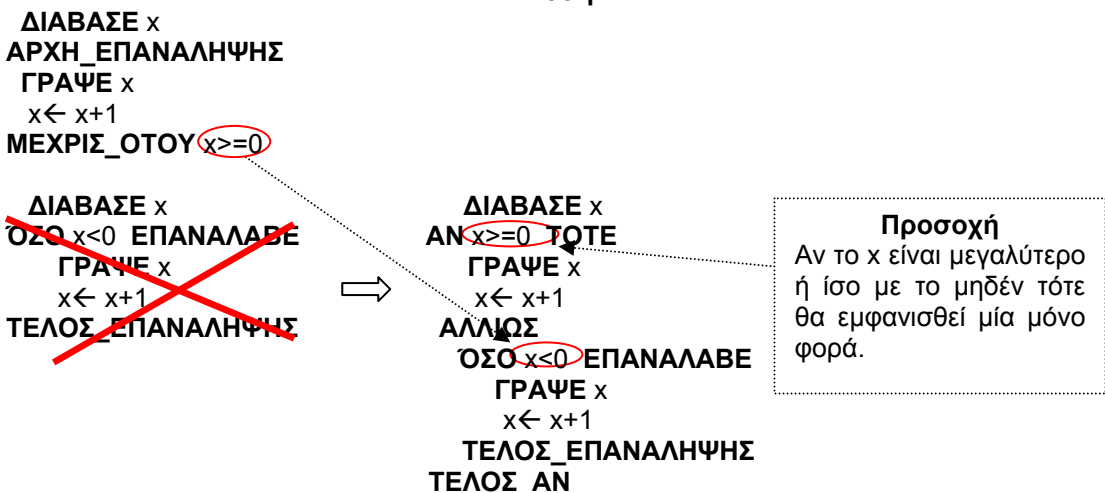
Λύση



ΠΑΡΑΔΕΙΓΜΑ 2

Να μετατραπεί η δομή επανάληψης **ΜΕΧΡΙΣ_ΟΤΟΥ** σε **ΌΣΟ..ΕΠΑΝΑΛΑΒΕ**.

Λύση



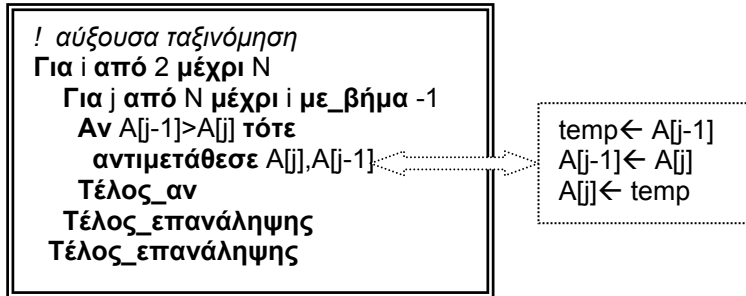
Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

Η ΤΑΞΙΝΟΜΗΣΗ

Η ταξινόμηση συνίσταται στη μετάθεση της θέσης των στοιχείων ώστε να τοποθετηθούν σε σειρά. Υπάρχουν πολλοί αλγόριθμοι ταξινόμησης ανάμεσα σ' αυτούς και ο αλγόριθμος της φυσαλίδας που θα εξετάσουμε παρακάτω. Ο αλγόριθμος αυτός δεν είναι ο βέλτιστος, υπάρχουν άλλοι αλγόριθμοι πιο γρήγοροι και αποτελεσματικοί. Στην πιο απλή μορφή γίνεται ταξινόμηση ενός μονοδιάστατου πίνακα N στοιχείων κατά φθίνουσα ή αύξουσα σειρά.

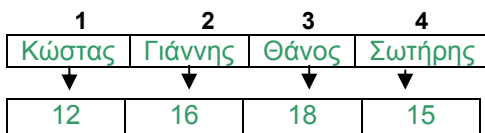
14.B.A



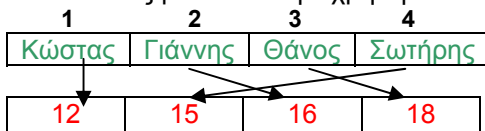
Για φθίνουσα ταξινόμηση αλλάζουμε το $A[j-1] > A[j]$ σε $A[j-1] < A[j]$.

ΤΑΞΙΝΟΜΗΣΗ ΣΕ ΠΑΡΑΛΛΗΛΟΥΣ ΠΙΝΑΚΕΣ

Δύο πίνακες λέμε ότι είναι παράλληλοι όταν οι θέσεις των στοιχείων του κάθε πίνακα, σχετίζονται μεταξύ τους, δηλαδή το πρώτο στοιχείο του πρώτου πίνακα σχετίζεται με το πρώτο στοιχείο του δεύτερου πίνακα κ.ο.κ. Η συσχέτιση μπορεί να είναι όνομα – βαθμολογία ή επίθετο-ηλικία κ.τ.λ. Για παράδειγμα: έστω ότι στον πίνακα B έχουμε τα ονόματα κάποιων μαθητών και στον πίνακα A τη βαθμολογία τους. Αν ταξινομήσουμε τον πίνακα A θα πρέπει ταυτόχρονα να ταξινομήσουμε και τον B διότι υπάρχει **συσχέτιση** ανάμεσα στις **θέσεις** που βρίσκονται οι βαθμοί και στις **θέσεις** που βρίσκονται τα ονόματα.



Οι πίνακες μετά από την χρήση του κώδικα **14.B.A** (χάνεται η συσχέτιση)



Οι πίνακες μετά από την χρήση του κώδικα **14.B.B**

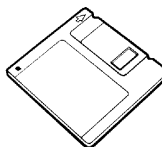


14.B.B

Για i από 2 μέχρι N
Για j από N μέχρι i με_βήμα -1
Αν $A[j-1] > A[j]$ τότε
 αντιμετάθεσε $A[j], A[j-1]$
 αντιμετάθεσε $B[j], B[j-1]$

Τέλος_αν
Τέλος_επανάληψης
Τέλος_επανάληψης

Για φθίνουσα ταξινόμηση αλλάζουμε το $A[j-1] > A[j]$ σε $A[j-1] < A[j]$. Επίσης την αντιμετάθεση μπορούμε να την κάνουμε και με βοηθητική μεταβλητή.



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

Η ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ

Στην πιο απλή μορφή της γίνεται αναζήτηση ενός στοιχείου σ' έναν αταξινόμητο μονοδιάστατο πίνακα. Τότε η αναζήτηση μπορεί να γίνει:

- για την εύρεση ενός στοιχείου{9.B.A} και της θέσης του,
- για την εύρεση του πλήθους του ζητούμενου{9.B.B}.

ΠΑΡΑΔΕΙΓΜΑ

Να γράψετε πρόγραμμα το οποίο θα διαβάσει κάποιους χαρακτήρες και θα τους καταχωρεί σ' έναν πίνακα 15 θέσεων (έστω ο πίνακας A). Έπειτα με την μέθοδο της σειριακής αναζήτησης θα βρίσκει:

- Από τον πίνακα A το ζητούμενο και την θέση του.
- Το πλήθος του ζητούμενου.

Λύση i

ΠΡΟΓΡΑΜΜΑ ΑΣΚ_9_B_A

ΣΤΑΘΕΡΕΣ

N=15

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, θέση

ΧΑΡΑΚΤΗΡΕΣ: ζητούμενο, A[N] ! Ο πίνακας με τους χαρακτήρες

ΛΟΓΙΚΕΣ: βρέθηκε ! Σημαία

ΑΡΧΗ

ΓΡΑΨΕ 'Δώστε τα στοιχεία του πίνακα: '

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

9.B.A

ΓΡΑΨΕ 'Δώστε το ζητούμενο: '

ΔΙΑΒΑΣΕ ζητούμενο

βρέθηκε ← ΨΕΥΔΗΣ

i ← 1

ΟΣΟ (i ≤ N) ΚΑΙ (βρέθηκε = ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ

ΑΝ A[i] = ζητούμενο ΤΟΤΕ

βρέθηκε ← ΑΛΗΘΗΣ

θέση ← i

ΑΛΛΙΩΣ

i ← i + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Εμφανίζουμε τα αποτελέσματα

ΑΝ βρέθηκε = ΨΕΥΔΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'Το ζητούμενο δε βρέθηκε '

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Το ζητούμενο βρέθηκε στην θέση: ', θέση

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Στην αναζήτηση ορισμένες φορές ενδιαφερόμαστε μόνο για τη θέση του ζητούμενου διότι αποτελεί το κλειδί για την εύρεση στοιχείων από παράλληλους πίνακες.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

Στην θέση του **9.B.A** θα μπορούσαμε να βάλουμε:

```
ΓΡΑΨΕ 'Δώστε το ζητούμενο: '  
ΔΙΑΒΑΣΕ ζητούμενο  
  
βρέθηκε ← ΨΕΥΔΗΣ  
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N  
  ΑΝ A[i]=ζητούμενο ΤΟΤΕ  
    βρέθηκε ← ΑΛΗΘΗΣ  
    θέση ← i  
ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΠΑΡΑΤΗΡΗΣΗ

Η λύση αυτή είναι πιο χρονοβόρα διότι κάνει 15 επαναλήψεις ακόμη και στην περίπτωση που έχει βρεθεί το ζητούμενο. Ο προηγούμενος αλγόριθμος κάνει αναζήτηση μέχρι τη θέση του ζητούμενου, συνεπώς στην καλύτερη περίπτωση έχουμε μία επανάληψη και στη χειρότερη 15.

Λύση ii

ΠΡΟΓΡΑΜΜΑ ΑΣΚ_9_B_B

ΣΤΑΘΕΡΕΣ

N=15

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, πλήθος

ΧΑΡΑΚΤΗΡΕΣ: ζητούμενο, A[N] ! Ο πίνακας με τους χαρακτήρες

ΑΡΧΗ

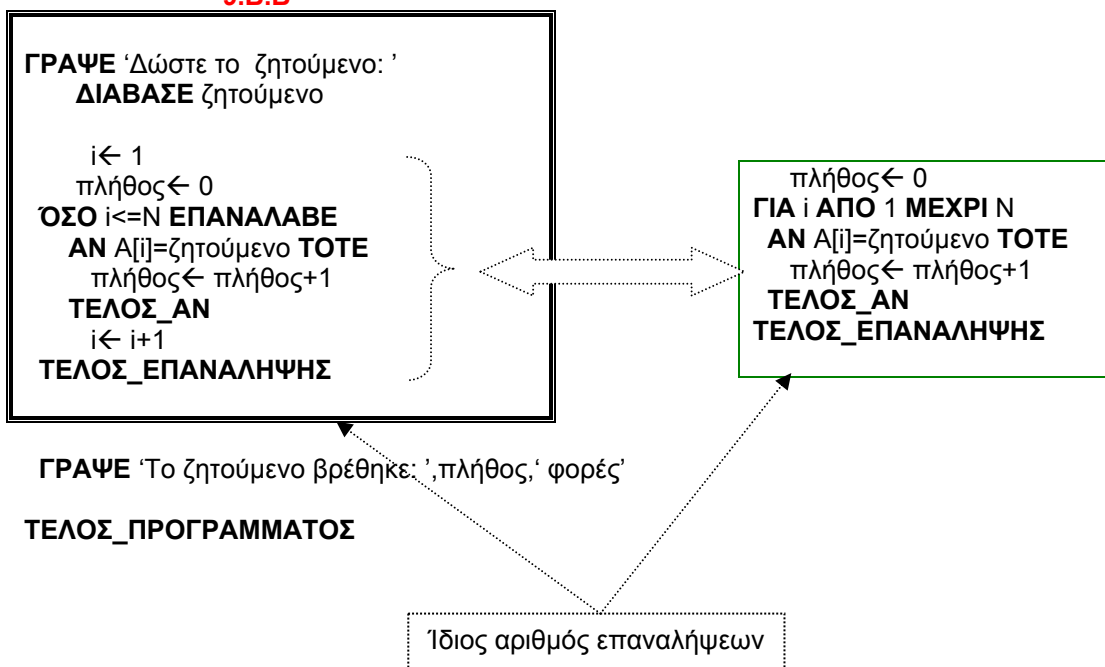
ΓΡΑΨΕ 'Δώστε τα στοιχεία του πίνακα: '

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

9.B.B



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ Σ' ΕΝΑΝ ΤΑΞΙΝΟΜΗΜΕΝΟ ΜΟΝΟΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ

Όταν η αναζήτηση γίνεται σ' έναν ταξινομημένο μονοδιάστατο πίνακα έχουμε τις εξής περιπτώσεις:

- εύρεση ενός στοιχείου{9.Β.Γ} και της θέσης του,
- εύρεση του πλήθους του ζητούμενου{9.Β.Δ}.

Αλγόριθμος Κωδικός **9_B_Γ**

Δεδομένα // N, ΠΙΝΑΚΑΣ //

! Αύξουσα διάταξη

Για i από 2 μέχρι N

Για j από N μέχρι i με_βήμα -1

Αν ΠΙΝΑΚΑΣ[j-1]>ΠΙΝΑΚΑΣ[j] τότε

αντιμετάθεσε ΠΙΝΑΚΑΣ[j], ΠΙΝΑΚΑΣ[j-1]

Τέλος_αν

Τέλος_επανάληψης

Τέλος_επανάληψης

9.Β.Γ

Εμφάνισε 'Δώστε το ζητούμενο: '

Διάβασε ζητούμενο

σημαία ← **Ψευδής**

θέση ← 0

i ← 1

Όσο (σημαία=Ψευδής) και (i<=N) και (ΠΙΝΑΚΑΣ[i]<= ζητούμενο) **επανάλαβε**

! ή (ΠΙΝΑΚΑΣ[i] >= ζητούμενο) για φθίνουσα διάταξη

Αν (ΠΙΝΑΚΑΣ[i]=ζητούμενο) τότε

σημαία ← **Αληθής**

θέση ← i

αλλιώς

i ← i + 1

Τέλος_αν

Τέλος_επανάληψης

Αν (σημαία= **Αληθής**) τότε

Εμφάνισε 'Το στοιχείο ', ζητούμενο, ' ευρέθη στη θέση: ', θέση

αλλιώς

Εμφάνισε 'Το στοιχείο ', ζητούμενο, ' δεν ευρέθη στον δοθέντα πίνακα'

Τέλος_αν

Τέλος Κωδικός **9_B_Γ**

ΠΑΡΑΤΗΡΗΣΕΙΣ:

Η αναζήτηση σταματά όταν η τιμή προς αναζήτηση είναι μικρότερη από την τρέχουσα τιμή του πίνακα. Σε αυτήν την περίπτωση δεν έχει νόημα να συνεχιστεί η αναζήτηση καθώς δεν αναμένεται να έχει επιτυχές αποτέλεσμα.

4	9	12	19	23	45	58	62
---	---	----	----	----	----	----	----

Για παράδειγμα η αναζήτηση για την τιμή 14 στον παραπάνω πίνακα δεν έχει νόημα να συνεχιστεί μετά την 4^η επανάληψη (i=4, ΠΙΝΑΚΑΣ[i]=19) καθώς ο αριθμός 14 που είναι μικρότερος του 19 αποκλείεται να βρίσκεται σε κάποια από τις επόμενες θέσεις του πίνακα. Έτσι, αποφεύγονται άσκοπες επαναλήψεις.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΠΑΡΑΔΕΙΓΜΑ

Να γράψετε πρόγραμμα το οποίο θα διαβάζει 15 ακέραιους αριθμούς και θα τους καταχωρεί σ' έναν πίνακα A. Έπειτα με την μέθοδο της σειριακής αναζήτησης θα βρίσκει το πλήθος ενός αριθμού. (Πρώτα να γίνει αύξουσα ταξινόμηση των στοιχείων του πίνακα)

Λύση

ΠΡΟΓΡΑΜΜΑ ΑΣΚ_9_B_Δ

ΣΤΑΘΕΡΕΣ

N=15

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j, ζητούμενο, πλήθος, A[N], temp

ΑΡΧΗ

ΓΡΑΨΕ 'Δώστε τα στοιχεία του πίνακα: '

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N

ΔΙΑΒΑΣΕ A[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Αύξουσα ταξινόμηση του πίνακα A

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N

ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1

ΑΝ A[j-1]>A[j] ΤΟΤΕ

temp ← A[j-1]

A[j-1] ← A[j]

A[j] ← temp

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

9.B.Δ

ΓΡΑΨΕ 'Δώστε το ζητούμενο: '

ΔΙΑΒΑΣΕ ζητούμενο

πλήθος ← 0

i ← 1

ΟΣΟ (i ≤ N) ΚΑΙ (A[i] ≤ ζητούμενο) ΕΠΑΝΑΛΑΒΕ

! ή (A[i] ≥ ζητούμενο) για φθίνουσα διάταξη

Αν (A[i]=ζητούμενο) τότε

πλήθος ← πλήθος+1

ΤΕΛΟΣ_ΑΝ

i ← i+1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

πλήθος ← 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N

ΑΝ A[i]=ζητούμενο ΤΟΤΕ

πλήθος ← πλήθος+1

ΤΕΛΟΣ_ΑΝ

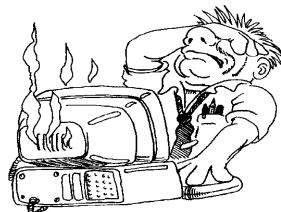
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Εμφανίζουμε τα αποτελέσματα

ΓΡΑΨΕ 'Το ζητούμενο βρέθηκε: ', πλήθος, ' φορές'

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Η λύση αυτή είναι πιο χρονο-
βόρα διότι ελέγχεται όλος ο
πίνακας.



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ ΣΕ ΔΙΣΔΙΑΣΤΑΤΟ ΠΙΝΑΚΑ

Στον παρακάτω αλγόριθμο θα κάνουμε σειριακή αναζήτηση σ' έναν δισδιάστατο πίνακα με N γραμμές και M στήλες. Στο *i_ros* καταχωρούμε τη γραμμή στην οποία ανήκει το κλειδί, ενώ στο *j_ros* τη στήλη.

Αλγόριθμος Σειριακή_2Δ

Δεδομένα // N, M, ΠΙΝΑΚΑΣ, ζητούμενο//

σημαία ← Ψευδής

i_θέση ← 0

j_θέση ← 0

i ← 1

Όσο (σημαία=Ψευδής) και (*i*≤N) επανάλαβε

j ← 1

Όσο (σημαία=Ψευδής) και (*j*≤M) επανάλαβε

Αν ΠΙΝΑΚΑΣ[*i*,*j*]=ζητούμενο τότε

σημαία ← Αληθής

i_θέση ← *i*

j_θέση ← *j*

Τέλος_αν

j ← *j*+1

Τέλος_επανάληψης

i ← *i*+1

Τέλος_επανάληψης

Αποτελέσματα // *i_θέση*,*j_θέση*//

Τέλος Σειριακή_2Δ

2^{ος} Τρόπος (Η λύση αυτή είναι σαφώς πιο χρονοβόρα διότι ελέγχεται όλος ο πίνακας)

```
σημαία ← Ψευδής
i_θέση ← 0
j_θέση ← 0
Για i από 1 μέχρι N
  Για j από 1 μέχρι M
    Αν ΠΙΝΑΚΑΣ[i,j]=ζητούμενο τότε
      σημαία ← Αληθής
      i_θέση ← i
      j_θέση ← j
    Τέλος_αν
  Τέλος_επανάληψης
Τέλος_επανάληψης
```

Υπάρχουν αρκετές μέθοδοι αναζήτησης σε πίνακα που εξαρτώνται κυρίως από το αν ο πίνακας είναι ταξινομημένος ή όχι. Μια άλλη παράμετρος είναι αν ο πίνακας περιέχει στοιχεία που είναι όλα διαφορετικά μεταξύ τους ή όχι. Η πιο απλή και αργή μέθοδος αναζήτησης είναι η σειριακή σ' έναν αταξινόμητο πίνακα.

Όπως είδαμε και στους προηγούμενους αλγορίθμους η χρήση της επαναληπτικής δομής **ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ** στη σειριακή αναζήτηση είναι δυνατή αλλά όχι πιο βέλτιστη από τη δομή **ΌΣΟ...ΕΠΑΝΑΛΑΒΕ**. Η χρήση της καλό είναι να αποφεύγεται όταν κάνουμε αναζήτηση ενός στοιχείου.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΡΥΘΜΟΣ ΜΕΤΑΒΟΛΗΣ

Τα προβλήματα αυτά λύνονται με τη χρήση των δομών επανάληψης:

- α) Για...από ..μέχρι...με_βήμα , παράδειγμα 1
- β) Όσο..επανάλαβε , παράδειγμα 2

Ακολουθούν τα παραδείγματα.

ΠΑΡΑΔΕΙΓΜΑ 1

Να δοθεί πρόγραμμα που να υπολογίζει τον αριθμό των δελφινιών στη μεσόγειο μετά από δέκα έτη. Τα δελφίνια σήμερα είναι 200.000 και αυξάνονται κατά 12% το χρόνο.

λύση

ΠΡΟΓΡΑΜΜΑ Ασκ1
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i

ΠΡΑΓΜΑΤΙΚΕΣ: Άθροισμα

ΑΡΧΗ

Άθροισμα ← 200000 ! αρχική τιμή

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10 ! περίοδος: 10 χρόνια

Άθροισμα ← Άθροισμα*0.12+ Άθροισμα ! ή Άθροισμα ← Άθροισμα*1.12

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Τα δελφίνια θα είναι: ', Άθροισμα
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Στην αρχή το Άθροισμα είναι: 200.000

έπειτα:

200.000*0,12 + 200.000
224.000*0,12 + 224.000
250.880*0,12 + 250.880
κ.τ.λ.

ΠΑΡΑΔΕΙΓΜΑ 2

Να δοθεί πρόγραμμα που να υπολογίζει σε πόσες ημέρες το 1€ θα γίνει ένα εκατομμύριο αν διπλασιάζουμε κάθε μέρα το υπάρχον ποσό.

λύση

ΠΡΟΓΡΑΜΜΑ Ασκ2
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i

ΠΡΑΓΜΑΤΙΚΕΣ: Άθροισμα

ΑΡΧΗ

Άθροισμα ← 1 ! αρχική τιμή: 1Ευρώ

i ← 0 ! οι ημέρες

ΌΣΟ Άθροισμα ≤ 1000000 ΕΠΑΝΑΛΑΒΕ

Άθροισμα ← Άθροισμα*2

i ← i+1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Οι ημέρες θα είναι: ', i
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Το όριο είναι: 1000000

Το ποσό διπλασιάζεται κάθε ημέρα

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

ΠΩΣ ΕΠΙΛΟΥΜΕ ΤΟ ΤΕΤΑΡΤΟ ΘΕΜΑ

1^ο ΔΙΑΒΑΖΟΥΜΕ ΤΗΝ ΕΚΦΩΝΗΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

Κατά τη διάρκεια Διεθνών αγώνων στίβου στον ακοντισμό έλαβαν μέρος 10 αθλητές. Κάθε αθλητής έκανε 6 έγκυρες ρίψεις που καταχωρούνται ως επιδόσεις σε μέτρα. Να αναπτύξετε αλγόριθμο ή πρόγραμμα το οποίο:

- α. Εισάγει σε πίνακα δύο διαστάσεων τις επιδόσεις όλων των αθλητών.
- β. Υπολογίζει και καταχωρεί σε μονοδιάστατο πίνακα την καλύτερη επίδοση κάθε αθλητή.
- γ. Ταξινομεί σε φθίνουσα σειρά τον πίνακα με τις καλύτερες επιδόσεις.
- δ. Βρίσκει την καλύτερη επίδοση του αθλητή που πήρε το χάλκινο μετάλλιο.



2^ο ΥΠΟΓΡΑΜΜΙΖΟΥΜΕ ΤΙΣ ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Κατά τη διάρκεια Διεθνών αγώνων στίβου στον ακοντισμό έλαβαν μέρος 10 αθλητές. Κάθε αθλητής έκανε 6 έγκυρες ρίψεις που καταχωρούνται ως επιδόσεις σε μέτρα. Να αναπτύξετε αλγόριθμο ή πρόγραμμα το οποίο:

- α. Εισάγει σε πίνακα δύο διαστάσεων τις επιδόσεις όλων των αθλητών.
- β. Υπολογίζει και καταχωρεί σε μονοδιάστατο πίνακα την καλύτερη επίδοση κάθε αθλητή.
- γ. Ταξινομεί σε φθίνουσα σειρά τον πίνακα με τις καλύτερες επιδόσεις.
- δ. Βρίσκει την καλύτερη επίδοση του αθλητή που πήρε το χάλκινο μετάλλιο.



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

γ) Στο σημείο αυτό του αλγορίθμου έχει δημιουργηθεί ένας επιπλέον πίνακας (από το β ερώτημα) μονοδιάστατος με τις καλύτερες επιδόσεις του κάθε αθλητή.



→ ταξινομή → ταξινόμηση του νέου πίνακα (φθίνουσα σειρά) → 14.Β.Α (ΥΨΗΛΟΤΕΡΗ)

δ) **χάλκινο μετάλλιο** → ταξινόμηση (ο πίνακας όμως είναι ήδη ταξινομημένος)
βρίσκει → εμφάνισε (τον τρίτο) → 2.Β (A[3])



5° ΔΗΜΙΟΥΡΓΟΥΜΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ ΕΝΩΝΟΝΤΑΣ ΟΛΕΣ ΤΙΣ ΕΠΙΜΕΡΟΥΣ ΛΥΣΕΙΣ

Η λύση της άσκησης είναι:

- ❖ 1.Γ (ΕΠΙΔΟΣΕΙΣ)
- ❖ 12.Γ.Α (ΕΠΙΔΟΣΕΙΣ)
- ❖ 14.Β.Α (ΥΨΗΛΟΤΕΡΗ)
- ❖ 2.Β (A[3])

ή

Αλγόριθμος ακοντισμός

Για i από 1 μέχρι 10

 Για j από 1 μέχρι 6

 Διάβασε ΕΠΙΔΟΣΕΙΣ[i,j] } 1.Γ

 Τέλος_επανάληψης

Τέλος_επανάληψης

Για i από 1 μέχρι 10

 Max ← ΕΠΙΔΟΣΕΙΣ[i,1]

 Για j από 2 μέχρι 6

 Αν (ΕΠΙΔΟΣΕΙΣ[i,j] > Max) τότε

 Max ← ΕΠΙΔΟΣΕΙΣ[i,j] } 12.Γ.Α

 Τέλος_αν

 Τέλος_επανάληψης

 ΥΨΗΛΟΤΕΡΗ[i] ← Max

Τέλος_επανάληψης

Για i από 2 μέχρι 10

 Για j από 10 μέχρι i με_βήμα -1

 Αν ΥΨΗΛΟΤΕΡΗ[j-1] < ΥΨΗΛΟΤΕΡΗ[j] τότε

 αντιμετάθεσε ΥΨΗΛΟΤΕΡΗ[j-1], ΥΨΗΛΟΤΕΡΗ[j] } 14.Β.Α

 Τέλος_αν

 Τέλος_επανάληψης

Τέλος_επανάληψης

 Εμφάνισε ΥΨΗΛΟΤΕΡΗ[3] } 2.Β

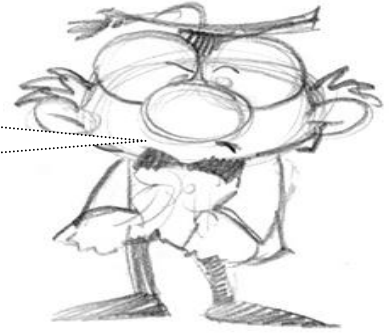
Τέλος Ακοντισμός



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

Σκοπός των ασκήσεων αυτών δεν είναι η εκμάθηση των κωδικών αλλά κυρίως να γίνει αντιληπτό ότι με την χρήση κάποιων βασικών μεθόδων μπορεί να λυθεί οποιαδήποτε άσκηση. **ΌΛΟΙ ΟΙ ΚΩΔΙΚΟΙ ΣΤΙΣ ΣΕΛ. 92-97**



ΠΑΡΑΔΕΙΓΜΑ 1

Για την πρώτη φάση της Ολυμπιάδας Πληροφορικής δήλωσαν συμμετοχή 500 μαθητές. Οι μαθητές διαγωνίζονται σε τρεις γραπτές εξετάσεις και βαθμολογούνται με ακέραιους βαθμούς στη βαθμολογική κλίμακα από 0 έως και 100.

Να γράψετε αλγόριθμο ο οποίος:

- Να διαβάζει τα ονόματα των μαθητών και να τα αποθηκεύει σε μονοδιάστατο πίνακα.
- Να διαβάζει τους τρεις βαθμούς που έλαβε κάθε μαθητής και να τους αποθηκεύει σε δισδιάστατο πίνακα.
- Να υπολογίζει το μέσο όρο των βαθμών του κάθε μαθητή.
- Να εκτυπώνει τα ονόματα των μαθητών και δίπλα τους το μέσο όρο των βαθμών τους ταξινομημένα με βάση το μέσο όρο κατά φθίνουσα σειρά. Σε περίπτωση ισοβαθμίας η σειρά ταξινόμησης των ονομάτων να είναι αλφαβητική.
- Να υπολογίζει και να εκτυπώνει το πλήθος των μαθητών με το μεγαλύτερο μέσο όρο.

Παρατήρηση: Θεωρείστε ότι οι βαθμοί των μαθητών είναι μεταξύ του 0 και του 100 και ότι τα ονόματα των μαθητών είναι γραμμένα με μικρά γράμματα.

(Θέμα 4^ο, Πανελλαδικές 2004)

Λύση

ΟΝΟΜΑ							
1	2					500

	ΒΑΘΜΟΙ			
	Βαθμός 1	Βαθμός 2	Βαθμός 3	ΜΟ
Ονόματα				

Καταχωρούμε σ' αυτόν τον πίνακα το μέσο όρο βαθμολογίας των 500 μαθητών (βρίσκουμε δηλαδή το **μέσο όρο ανά γραμμή**). Έπειτα κάνουμε στον πίνακα ΜΟ **ταξινόμηση** κατά φθίνουσα σειρά, ταυτόχρονα ταξινομούμε και τον πίνακα ΟΝΟΜΑ. ΠΡΟΣΟΧΗ σε περίπτωση ισοβαθμίας η ταξινόμηση των ονομάτων γίνεται κατά αλφαβητική σειρά. Τέλος για να βρούμε το πλήθος των μαθητών με το μεγαλύτερο ΜΟ κάνουμε **σειριακή αναζήτηση** στον **ταξινομημένο** πλέον πίνακα ΜΟ.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

Αλγόριθμος Θέμα_4_2004

Για i από 1 μέχρι 500
 Διάβασε ΟΝΟΜΑ[i]
 Τέλος_επανάληψης } 1.Β

Για i από 1 μέχρι 500
 Για j από 1 μέχρι 3
 Διάβασε ΒΑΘΜΟΙ[i,j]
 Τέλος_επανάληψης } 1.Γ
 Τέλος_επανάληψης

Για i από 1 μέχρι 500
 άθροισμα ← 0
 Για j από 1 μέχρι 3
 άθροισμα ← άθροισμα + ΒΑΘΜΟΙ[i,j]
 Τέλος_επανάληψης } 7.Γ
 ΜΟ[i] ← άθροισμα / 3
 Τέλος_επανάληψης

Για i από 2 μέχρι 500
 Για j από 500 μέχρι i με_βήμα -1
 Αν ΜΟ[j-1] < ΜΟ[j] τότε
 αντιμετάθεσε ΜΟ[j-1], ΜΟ[j]
 αντιμετάθεσε ΟΝΟΜΑ[j-1], ΟΝΟΜΑ[j]
 Τέλος_αν
 Αν ΜΟ[j-1] = ΜΟ[j] τότε
 Αν ΟΝΟΜΑ[j-1] > ΟΝΟΜΑ[j] τότε
 αντιμετάθεσε ΟΝΟΜΑ[j-1], ΟΝΟΜΑ[j]
 Τέλος_αν
 Τέλος_αν
 Τέλος_επανάληψης } 14.Β.Β (ΜΟ, ΟΝΟΜΑ)
 Τέλος_επανάληψης

Για i από 1 μέχρι 500
 Εμφάνισε 'Όνομα: ', ΟΝΟΜΑ[i], ' Μέσος όρος: ', ΜΟ[i]
 Τέλος_επανάληψης } 2.Β

Πλήθος ← 0
i ← 1
Όσο ΜΟ[i] = ΜΟ[1] και i ≤ 500 επανάλαβε } 9.Β.Δ
 Πλήθος ← Πλήθος + 1
 i ← i + 1
Τέλος_επανάληψης

Εμφάνισε 'Το πλήθος των μαθητών με το μεγαλύτερο μέσο όρο είναι: ', Πλήθος } 2.Α

Τέλος Θέμα_4_2004



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΠΑΡΑΔΕΙΓΜΑ 2

Μια εταιρία κινητής τηλεφωνίας έχει ένα εκατομμύριο συνδρομητές και καταχωρεί σ' ένα μονοδιάστατο πίνακα ΟΝΟΜΑΤΑ τα ονόματα των συνδρομητών και σ' ένα δεύτερο δισδιάστατο πίνακα ΧΡΟΝΟΙ το χρόνο σε ώρες που αυτοί μίλησαν κάθε μήνα κατά την διάρκεια του προηγούμενου έτους. Η εταιρία αποφάσισε να προσφέρει στους συνδρομητές της τη δυνατότητα να αλλάξουν συσκευή τηλεφώνου ενισχύοντάς τους με ένα ποσό με βάση το χρόνο που μίλησαν τον προηγούμενο χρόνο. Να γίνει πρόγραμμα το οποίο:

A) Θα αποθηκεύει στους πίνακες τα ονόματα και τους χρόνους που μίλησαν οι συνδρομητές κάθε μήνα κατά τη διάρκεια του προηγούμενου έτους.

B) Θα εμφανίζει τον συνολικό χρόνο που χρησιμοποίησαν οι συνδρομητές το δίκτυο της εταιρίας το μήνα Μάρτιο.

Γ) Θα αποθηκεύει σ' ένα μονοδιάστατο πίνακα ΠΟΣΑ το ποσό που θα προσφερθεί στον πελάτη με βάση το συνολικό χρόνο που μίλησε το προηγούμενο έτος όπως αυτό καθορίζεται στον ακόλουθο πίνακα.

Συνολικός χρόνος (σε ώρες)	Ποσό προσφοράς
0-50	80 Ευρώ
50,1-120	130 Ευρώ
Πάνω από 120	200 Ευρώ

Δ) Θα δέχεται στην είσοδο το όνομα του συνδρομητή και θα εμφανίζει στην οθόνη το ποσό της ενίσχυσης που του αντιστοιχεί. (Υποθέστε ότι κάθε όνομα υπάρχει μία μόνο φορά στον πίνακα).

Λύση

ΟΝΟΜΑΤΑ

1	2			1000000
Κώστας	Περικλής

ΧΡΟΝΟΙ

	Ιαν	Φεβ	Δεκ.
Κώστας										
...										
...										
...										
...										
Περικλής										

	Ιαν	Φεβ	Δεκ.	
Κώστας											Σ
...											
...											
...											
...											
Περικλής											

ΠΟΣΑ

Κώστας				Περικλής
130	200

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Βασικές γνώσεις

ΠΡΟΓΡΑΜΜΑ κινητή_τηλεφωνία
ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ΧΡΟΝΟΙ[1000000,12], Συνολικός_χρόνος, Σ
ΑΚΕΡΑΙΕΣ: συνδρομητής, μήνας, ΠΟΣΑ[1000000], θέση
ΧΑΡΑΚΤΗΡΕΣ: ΟΝΟΜΑΤΑ[1000000], όνομα
ΛΟΓΙΚΕΣ: βρέθηκε

ΑΡΧΗ

ΓΙΑ συνδρομητής ΑΠΟ 1 ΜΕΧΡΙ 1000000
ΓΡΑΨΕ 'Δώστε το όνομα του ',συνδρομητής, 'ου συνδρομητή: ' } 1.B
ΔΙΑΒΑΣΕ ΟΝΟΜΑΤΑ[συνδρομητής]
ΓΙΑ μήνας ΑΠΟ 1 ΜΕΧΡΙ 12
ΓΡΑΨΕ 'Δώστε το χρόνο ομιλίας για τον ',μήνας, 'ο μήνα: '
ΔΙΑΒΑΣΕ ΧΡΟΝΟΙ[συνδρομητής,μήνας]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ } 1.Γ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Συνολικός_χρόνος ← 0
ΓΙΑ συνδρομητής ΑΠΟ 1 ΜΕΧΡΙ 1000000
Συνολικός_χρόνος ← Συνολικός_χρόνος+ΧΡΟΝΟΙ[συνδρομητής,3] } 5.Γ.B
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ συνδρομητής ΑΠΟ 1 ΜΕΧΡΙ 1000000
Σ ← 0

ΓΙΑ μήνας ΑΠΟ 1 ΜΕΧΡΙ 12
Σ ← Σ+ΧΡΟΝΟΙ[συνδρομητής,μήνας]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

~~ΑΝ Σ ≤ 50 ΤΟΤΕ
ΠΟΣΑ[συνδρομητής] ← 80
ΑΛΛΙΩΣ_ΑΝ Σ ≤ 120 ΤΟΤΕ
ΠΟΣΑ[συνδρομητής] ← 50*80+(Σ-50)*130
ΑΛΛΙΩΣ
ΠΟΣΑ[συνδρομητής] ← 50*80+70*130+(Σ-120)*200~~

ΑΝ Σ ≤ 50 ΤΟΤΕ
ΠΟΣΑ[συνδρομητής] ← 80
ΑΛΛΙΩΣ_ΑΝ Σ ≤ 120 ΤΟΤΕ
ΠΟΣΑ[συνδρομητής] ← 130
ΑΛΛΙΩΣ
ΠΟΣΑ[συνδρομητής] ← 200
ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώστε το όνομα του συνδρομητή: '
ΔΙΑΒΑΣΕ όνομα

συνδρομητής ← 1
βρέθηκε ← ΨΕΥΔΗΣ
θέση ← 0

ΌΣΟ (βρέθηκε=ΨΕΥΔΗΣ) ΚΑΙ (συνδρομητής ≤ 1000000) ΕΠΑΝΑΛΑΒΕ

ΑΝ ΟΝΟΜΑΤΑ[συνδρομητής]=όνομα ΤΟΤΕ

βρέθηκε ← ΑΛΗΘΗΣ
θέση ← συνδρομητής

ΑΛΛΙΩΣ

συνδρομητής ← συνδρομητής+1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ βρέθηκε=ΑΛΗΘΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'Ο συνδρομητής θα ενισχυθεί με το ποσό των ',ΠΟΣΑ[θέση], ' Ευρώ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δεν υπάρχει συνδρομητής με όνομα ',όνομα

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βασικές γνώσεις

ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα. Ο τμηματικός προγραμματισμός διευκολύνει την κατανόηση και διόρθωση του προγράμματος και απαιτεί λιγότερο χρόνο και προσπάθεια στην συγγραφή του.

Ένα **υποπρόγραμμα** είναι ένα αυτόνομο τμήμα προγράμματος, το οποίο επιτελεί κάποιες λειτουργίες και έχει αναπτυχθεί χωριστά από το υπόλοιπο πρόγραμμα. Κάθε υποπρόγραμμα πρέπει να ικανοποιεί τις εξής τρεις ιδιότητες:

- Να έχει μια είσοδο και μια έξοδο.
- Να είναι ανεξάρτητο από τα άλλα.
- Να μην είναι πολύ μεγάλο.

Δύο είναι τα βασικά είδη υποπρογραμμάτων οι διαδικασίες και οι συναρτήσεις.

- Μια **διαδικασία** είναι ένα υποπρόγραμμα το οποίο έχει τη δυνατότητα να επιτελεί όλες τις λειτουργίες που μπορεί να επιτελέσει και ένα πρόγραμμα(να διαβάσει, να εμφανίζει στην οθόνη, να κάνει υπολογισμούς κ.λ.π.). Η μορφή της διαδικασίας είναι η παρακάτω:

```
ΔΙΑΔΙΚΑΣΙΑ Όνομα(παράμετροι)
  Τμήμα δηλώσεων
ΑΡΧΗ
  Εντολές
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

- Μία **συνάρτηση** είναι ένα υποπρόγραμμα το οποίο όμως είναι περιορισμένο, ως προς τις λειτουργίες του, σε σχέση με τις διαδικασίες. Συγκεκριμένα, μία συνάρτηση μπορεί να δεχθεί ως είσοδο μία ή περισσότερες τιμές, να τις επεξεργαστεί και στο τέλος να επιστρέψει στο κυρίως πρόγραμμα **μια τιμή** ως αποτέλεσμα.

```
ΣΥΝΑΡΤΗΣΗ Όνομα(παράμετροι): Τύπος συνάρτησης
  Τμήμα δηλώσεων
ΑΡΧΗ
  Εντολές
  Όνομα ← {κάποια τιμή}
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Παράμετρος είναι μια μεταβλητή την οποία χρησιμοποιούμε για να περάσουμε την τιμή της από ένα τμήμα προγράμματος σε κάποιο άλλο.

ΠΑΡΑΔΕΙΓΜΑ 1

Να γράψετε διαδικασία με όνομα **ΜΕΓΙΣΤΟΣ**, που να δέχεται ως είσοδο ένα μονοδιάστατο πίνακα πραγματικών **A**, το πλήθος **N** των στοιχείων του (το πολύ 50) και να υπολογίζει το μέγιστο στοιχείο του καθώς και την θέση του.

Λύση

```
ΔΙΑΔΙΚΑΣΙΑ ΜΕΓΙΣΤΟΣ(N, A, Max, θέση)
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: N, i, θέση
  ΠΡΑΓΜΑΤΙΚΕΣ: A[50], Max
ΑΡΧΗ
```

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
 Βασικές γνώσεις

```

Max ← A[1]
θέση ← 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
  ΑΝ A[i] > Max ΤΟΤΕ
    Max ← A[i]
    θέση ← i
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

ΠΑΡΑΔΕΙΓΜΑ 2

Τι θα εμφανίσει το παρακάτω πρόγραμμα;

ΠΡΟΓΡΑΜΜΑ Παράδειγμα
ΜΕΤΑΒΛΗΤΕΣ

```

ΑΚΕΡΑΙΕΣ: α, β
ΑΡΧΗ
α ← 5
β ← 4
ΚΑΛΕΣΕ Διαδ1(α,β)
ΓΡΑΨΕ α,β
β ← Συν1(α,β)
ΓΡΑΨΕ α,β
ΚΑΛΕΣΕ Διαδ1(α,β)
ΓΡΑΨΕ α,β
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
    
```

Τη διαδικασία την καλούμε με την εντολή **ΚΑΛΕΣΕ**, ενώ την συνάρτηση με το όνομά της.

κ=α και λ=β
 Οποιαδήποτε αλλαγή στο κ ή στο λ προκαλεί αντίστοιχη αλλαγή στο α ή στο β.

```

ΔΙΑΔΙΚΑΣΙΑ Διαδ1(κ,λ)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: κ, λ
ΑΡΧΗ
κ ← κ-1
λ ← λ+1
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

Στην συνάρτηση δηλώνουμε και τι είδους μεταβλητή θα είναι η μοναδική τιμή που θα επιστραφεί.

```

ΣΥΝΑΡΤΗΣΗ Συν1(μ,ν):ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: μ, ν
ΑΡΧΗ
Συν1 ← μ+ν
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
    
```

μ=α και ν=β
 Οποιαδήποτε αλλαγή στο μ ή στο ν **δεν** προκαλεί κάποια αλλαγή στο α ή στο β. Το μόνο που αλλάζει είναι το Συν1.

Λύση

Οι τιμές που θα εμφανισθούν είναι:

α	β
4	5
4	9
3	10

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

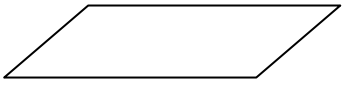
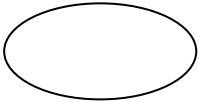
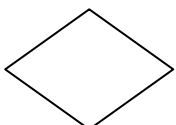
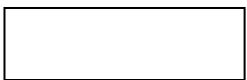
1. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Χαρακτηριστικά</i>	<i>Αλγοριθμικές έννοιες</i>
1. Περατότητα 2. Είσοδος 3. Έξοδος	α. Δεδομένα β. Αποτελέσματα γ. Ακρίβεια στην έκφραση των εντολών δ. Πεπερασμένος χρόνος εκτέλεσης

2. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Δεδομένα</i>	<i>Τύπος δεδομένων</i>
1. Ύψος εφήβου 2. Επώνυμο μαθητή 3. Αριθμός τερμάτων	α. Ακέραιες β. Πραγματικές γ. Αλφαριθμητικές δ. Λογικές

3. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>ΣΧΗΜΑ</i>	<i>ΣΥΜΒΟΛΙΚΟΣ ΣΧΗΜΑΤΙΣΜΟΣ</i>
1. 	α. Ροή εκτέλεσης αλγορίθμου β. Αρχή-Τέλος γ. Υποπρογράμματα δ. Εντολή συνθήκης ε. Είσοδος-Έξοδος στ. Εκτέλεση πράξης
2. 	
3. 	
4. 	

4. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Τιμή</i>	<i>Τύπος δεδομένων</i>
1. 222 2. 'αληθής' 3. ψευδής 4. -10.3	α. Αλφαριθμητικές β. Αριθμητικός (ακέραιες, πραγματικές) γ. Λογικές

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

5. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Τμήμα Αλγορίθμου</i>	<i>Δεσμευμένη λέξη</i>
1. Τμήμα εντολών 2. Τίτλος 3. Τμήμα δηλώσεων	α. Τοποθέτησε β. Εμφάνισε γ. Ακέραιος δ. Εκτύπωσε ε. Αλγόριθμος στ. Ρητός

6. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Τμήμα αλγόριθμου</i>	<i>Τιμή Y μετά την τελευταία εντολή</i>
1. $M \leftarrow 8$ $X \leftarrow 7$ $Y \leftarrow M$ $Y \leftarrow X$	Α. 10 Β. 8
2. $X \leftarrow 20$ $Y \leftarrow (X \text{ div } 8) * 2$ $X \leftarrow Y - 3$ $Y \leftarrow X$	Γ. 1
3. $Z \leftarrow 2$ $Y \leftarrow 10$ $X \leftarrow 4$ $Y \leftarrow (X \text{ mod } Z) * Y$	Δ. 0 Ε. Τίποτε από τα παραπάνω

7. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Προβλήματα</i>	<i>Είδος Προβλημάτων</i>
1. Η διαδικασία λύσης τους είναι αυτοματοποιημένη. 2. Δεν έχει βρεθεί λύση, αλλά δεν έχει αποδειχθεί και η μη ύπαρξη λύσης. 3. Ο τρόπος λύσης τους μπορεί να επιλεγεί από πλήθος δυνατών λύσεων.	Α. Ανοικτά Β. Δομημένα Γ. Άλυτα Δ. Ημιδομημένα

8. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
<i>Προβλήματα</i>	<i>Είδος Προβλήματος</i>
1. Η κατάκτηση του παγκοσμίου πρωταθλήματος ποδοσφαίρου από την εθνική Ελλάδος. 2. Ο τετραγωνισμός του κύκλου. 3. Η επίλυση μιας δευτεροβάθμιας εξίσωσης.	Α. Ανοικτό Β. Αδόμητο Γ. Άλυτο Δ. Τίποτα από τα παραπάνω

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

9. Ποια από τα ακόλουθα αλφαριθμητικά είναι αποδεκτά ως ονόματα μεταβλητών σε έναν αλγόριθμο;

i.	Τιμή	
ii.	Τιμή-1	
iii.	Τιμή_2	
iv.	Egfsst	
v.	Τιμή.δ	
vi.	τ	
vii.	Γ/ε4	
viii.	τιμή 2	
ix.	ΚΟΣΤΑΣ	
x.	2α	

10. Να κρίνετε για την ορθότητά τους τις παρακάτω εντολές εκχώρησης τιμής.

i.	τιμή ← "αρκετά"	
ii.	τιμή ← 14 + 5	
iii.	α+τιμή ← 2	
iv.	τιμή ← υσ ← 2	
v.	τιμή ← 8*7 + 4	
vi.	τιμή ← "τιμή"	
vii.	τιμή ← 4/0	
viii.	2*τιμή ← 8	
ix.	τιμή ← 3*8^0	
x.	Διάβασε τιμή ← 2	
xi.	Διάβασε ← 8	
xii.	τιμή ← 0	
xiii.	τιμή + 5 ← 3	
xiv.	τιμή = 8	



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
 Ασκήσεις 1-115

11. Πώς θα διατυπωθεί σε εντολή εκχώρησης τιμής, η κάθε μία από τις παρακάτω αλγεβρικές παραστάσεις;

i.	$\frac{5x^3 + 7x^2 + 8}{8x - 6}$	
ii.	$x^2 - 32y^3 - 7$	
iii.	$a^{3x+2} + \frac{x+1}{x^3 - 2}$	

12. Ποιο είναι το αποτέλεσμα από την εκτέλεση των παρακάτω πράξεων:

- i. $14 \bmod 5 - 25 \bmod 8 =$
- ii. $3 * (3 \bmod 2) + 4 \operatorname{div} (5 \bmod 3) =$
- iii. $13 \bmod (27 \operatorname{div} 4) =$
- iv. $2^3 + 3 * (27 \bmod (25 \bmod 7)) =$
- v. $13/2 - 3 \bmod 2 - 3 \operatorname{div} 2 =$
- vi. $13/4 + 2 * 5 \bmod 3 * 4 =$
- vii. $25 \bmod 22 \operatorname{div} 4 =$
- viii. $((13 + 2) \operatorname{div} 2) / (7 - 4 + 1) =$
- ix. $3 * (27 \bmod (23 \bmod 6)) =$
- x. $1 \operatorname{div} 1 =$
- xi. $1 \bmod 1 =$

13. Ποιο είναι το λογικό αποτέλεσμα (αληθής ή ψευδής) από την εκτέλεση των παρακάτω πράξεων αν οι εξής μεταβλητές έχουν τιμές: $A = 10, B = 2, \Gamma = -4, \Delta = 9$ και $E = 1$

- i. $(A > B)$ ή $(\Delta = 10)$
- ii. $(\Delta \geq B)$ και $(E < \Gamma)$
- iii. **όχι** $(E \leq \Gamma)$ ή $(\Delta \leq \Gamma)$
- iv. **όχι** $((B \leq \Gamma)$ και $(\Delta < 2))$
- v. **όχι** (**όχι** $(B \leq E)$ ή **όχι** $(\Gamma \leq B)$)
- vi. $((E \leq A)$ και $(E \geq \Gamma))$ και **όχι** $(\Gamma \geq A)$
- vii. **όχι** (**όχι** $(A \geq 2)$ και $(\Gamma < 9)$)

14. Να διατυπώσετε σε λογικές εκφράσεις τις παρακάτω προτάσεις:

i.	Το α ανήκει στο διάστημα $[-5, 6)$	
ii.	Το α είναι μικρότερο του 3 ή μεγαλύτερο του 15	
iii.	Το α είναι ίσο με το β και το γ	
iv.	Το α δεν έχει την τιμή 3	
v.	Το α είναι μικρότερο του 2 ή το β είναι μεγαλύτερο του 78	
vi.	α και β αληθή και γ ψευδές	

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Ασκήσεις 1-115

15. Να επιλέξετε το γράμμα που αντιστοιχεί στη σωστή απάντηση.

1. Ένα από τα παρακάτω δεν αποτελεί τρόπο αναπαράστασης αλγορίθμου :
A. Ελεύθερο κείμενο B. Κωδικοποίηση
Γ. Φυσική γλώσσα κατά βήματα Δ. Ελεύθερο σχέδιο
2. Ποια λειτουργία από τις παρακάτω δεν εκτελεί ο υπολογιστής;
A. Μεταφορά δεδομένων B. Σύγκριση
Γ. Πολλαπλασιασμός Δ. Πρόσθεση
3. Ένα από τα παρακάτω δεν αποτελεί σύμβολο του διαγράμματος ροής:
A. Ρόμβος B. Τετράγωνο
Γ. Ορθογώνιο Δ. Βέλος
4. Ποιο από τα παρακάτω δεν μπορεί να είναι όνομα μεταβλητής:
A. Όνομα μαθητή B. Ανάπτυξη_Εφαρμογών
Γ. Φ Δ. Ασκ1
5. Ποιο από τα παρακάτω δεν μπορεί να είναι όνομα μεταβλητής:
A. Ρητός B. ρ67τ
Γ. Φ1 Δ. Ακέραιες
6. Η τιμή **Ψευδής** θα πρέπει να εκχωρηθεί σε:
A. Ακέραια μεταβλητή B. Λογική μεταβλητή
Γ. Μεταβλητή χαρακτήρα Δ. Τίποτα από τα προηγούμενα
7. Η τιμή **Αληθής** θα πρέπει να εκχωρηθεί σε:
A. Ακέραια μεταβλητή B. Λογική μεταβλητή
Γ. Μεταβλητή χαρακτήρα Δ. Τίποτα από τα προηγούμενα
8. Η έκφραση 'μάθημα' μπορεί να εκχωρηθεί σε μεταβλητή τύπου:
A. Ακέραια μεταβλητή B. Λογική μεταβλητή
Γ. Μεταβλητή χαρακτήρα Δ. Τίποτα από τα προηγούμενα
9. Η έκφραση 'ΝΑΙ' μπορεί να εκχωρηθεί σε:
A. Ακέραια μεταβλητή B. Λογική μεταβλητή
Γ. Μεταβλητή χαρακτήρα Δ. Τίποτα από τα προηγούμενα
10. Η τιμή -10,2 θα πρέπει να εκχωρηθεί σε:
A. Ακέραια μεταβλητή B. Λογική μεταβλητή
Γ. Ακέραια ή λογική μεταβλητή Δ. Τίποτα από τα προηγούμενα
11. Ο αριθμός των μεταβλητών που βρίσκονται δεξιά της εντολής εκχώρησης πρέπει να είναι μέχρι:
A. 1 B. 2
Γ. 3 Δ. Τίποτα από τα προηγούμενα
12. Ποια από τις παρακάτω εκφράσεις εκχωρούν στην μεταβλητή X την τιμή 5;
A. $X \leftarrow \text{"πέντε"}$ B. $X \leftarrow \text{πέντε}$
Γ. $X=5$ Δ. Τίποτα από τα προηγούμενα
13. Ποια από τις παρακάτω εκφράσεις εκχωρούν στις μεταβλητές X,Y,Z την τιμή 3:
A. $X \leftarrow Y \leftarrow Z \leftarrow 3$ B. $X \leftarrow 3, Y \leftarrow 3, Z \leftarrow 3$
Γ. $X=Y=Z=3$ Δ. Τίποτα από τα προηγούμενα
14. Ποιο είναι το αποτέλεσμα της πράξης $25 \text{ div } 4$;
A. 6 B. 1
Γ. 6,25 Δ. Τίποτα από τα προηγούμενα

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Ασκήσεις 1-115

15. Ποιο είναι το αποτέλεσμα της πράξης $33 \bmod 4$;
A. 2
B. 1
Γ. 8
Δ. 25
16. Ποιο είναι το αποτέλεσμα της πράξης $33/4$;
A. 8
B. 25
Γ. 8,25
Δ. 0
17. Στην πράξη $X \bmod 2$ (το $X=2,4,6,8,10$ κ.τ.λ.) το αποτέλεσμα είναι:
A. 0
B. 1
Γ. 2
Δ. Τίποτα από τα προηγούμενα
18. Στην πράξη $X \bmod 2$ (το $X=1,3,5,7,9$ κ.τ.λ.) το αποτέλεσμα είναι:
A. 0
B. 1
Γ. 2
Δ. Τίποτα από τα προηγούμενα
19. Στην πράξη $X \text{ div } 2$ (το $X=1,3,5,7,9$ κ.τ.λ.) το αποτέλεσμα είναι:
A. 0
B. 1
Γ. 2
Δ. Το αποτέλεσμα εξαρτάται από το X
20. Στην πράξη $X \text{ div } X$ (το $X=2,4,6,8,10$ κ.τ.λ.) το αποτέλεσμα είναι:
A. 0
B. 1
Γ. X
Δ. Δε γνωρίζουμε
21. Ποιο είναι το σύμβολο της σύζευξης;
A. ή
B. και
Γ. όχι
Δ. Τίποτα από τα προηγούμενα

16. Αντιστοιχίσετε τη Στήλη Α με τη στήλη Β.

Στήλη Α	Στήλη Β
Εκφράσεις	Αλγοριθμικές έννοιες
1. $X \leftarrow X+2$ 2. $3+A>B$ 3. Εκτύπωσε Β 4. Όσο $K<3$ επανάλαβε Εντολές Τέλος επανάληψης 5. $X-(X/2)*2$	α. αριθμητική έκφραση β. μεταβλητή γ. λογική έκφραση δ. δομή ακολουθίας ε. δομή επανάληψης στ. εντολή εκχώρησης ζ. εντολή εξόδου

17. Συμπληρώστε τον παρακάτω πίνακα:

A	B	A και B	A ή B
Αληθής	Αληθής		
Αληθής	Ψευδής		
Ψευδής	Αληθής		
Ψευδής	Ψευδής		

18. Συμπληρώστε τον παρακάτω πίνακα:

A	B	όχι A	όχι (A ή B)
Αληθής	Αληθής		
Αληθής	Ψευδής		
Ψευδής	Αληθής		
Ψευδής	Ψευδής		

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

19. Συμπληρώστε τον παρακάτω πίνακα:

A	B	όχι (B και A)	όχι (A και B)
Αληθής	Αληθής		
Αληθής	Ψευδής		
Ψευδής	Αληθής		
Ψευδής	Ψευδής		

20. Συμπληρώστε τον παρακάτω πίνακα:

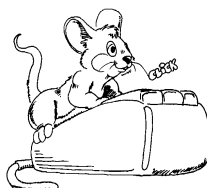
A	B	όχι (B ή A)	όχι (A και (όχι (A ή B)))
1	1		
1	0		
0	1		
0	0		

21. Συμπληρώστε τον παρακάτω πίνακα:

ω	κ	λ	$\omega=2$ και ($\kappa \neq 0$ ή $\lambda > 0$)
2	-7	12	
6	0	-9	
2	1	0	

22. Να συνδέσετε τα στοιχεία της στήλης A με τα στοιχεία της στήλης B.

ΣΤΗΛΗ A	ΣΤΗΛΗ B
Τμήμα Αλγορίθμου	Τελική τιμή του ζ
1. $\zeta \leftarrow 0, v \leftarrow 4$ Αρχή_επανάληψης Αν $v \neq 1$ τότε $\zeta \leftarrow \zeta + v$ Τέλος_αν $v \leftarrow v - 1$ Μέχρις_ότου $v \leq 2$	A. 4 B. -1
2. $\chi \leftarrow -2, \zeta \leftarrow 1$ Όσο $\chi \neq 2$ επανάλαβε $\zeta \leftarrow -2 * \zeta$ $\chi \leftarrow \chi + 2$ Τέλος_επανάληψης	Γ. 2 Δ. 8
3. Για v από 1 μέχρι 2 $\zeta \leftarrow v$ Τέλος_επανάληψης	E. -7
4. $\zeta \leftarrow 3, B \leftarrow 1$ Αρχή_επανάληψης Αν $B \leq 9$ τότε $\zeta \leftarrow \zeta - 2$ Τέλος_αν $B \leftarrow B + 2$ Μέχρις_ότου $\zeta = -7$ και $B > 4$	ΣΤ. 7 Z. Τίποτα από τα παραπάνω



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

23. Ποια θα είναι η τελική τιμή του x ; A) 8 B) 0 Γ) 1

$x \leftarrow 0$
Όσο $x \neq 8$ επανάλαβε
 $x \leftarrow x+2$
Τέλος_επανάληψης
 $x \leftarrow x+2$
Όσο $x \neq 0$ επανάλαβε
 $x \leftarrow x-2$
Τέλος_επανάληψης

24. Ποια θα είναι η τελική τιμή του x ; A) 18 B) 1 Γ) 0

$x \leftarrow -2011$
Όσο $x \neq 18$ επανάλαβε
 $x \leftarrow x+1$
Τέλος_επανάληψης
 $x \leftarrow x+1$
Όσο $x > 0$ επανάλαβε
 $x \leftarrow x-1$
Τέλος_επανάληψης

25. Έστω το παρακάτω τμήμα αλγορίθμου:

$k \leftarrow 19$
 $B \leftarrow 19$
 $i \leftarrow 0$
Αρχή_επανάληψης
Αν $B \leq 19$ τότε
 $k \leftarrow k \text{ div } 2$
 $i \leftarrow i+k$
Τέλος_αν
 $B \leftarrow i+2$
Μέχρις_ότου ($B \geq 16$)



α) Ποια θα είναι η τελική τιμή του k ; β) Ποια θα είναι η τελική τιμή του B ;

26. Έστω το παρακάτω τμήμα αλγορίθμου:

$B \leftarrow 2$
 $k \leftarrow 0$
Αρχή_επανάληψης
Αν $B \leq 9$ τότε
Εκτύπωσε 'Το k είναι: ', k
Τέλος_αν
 $B \leftarrow B+2$
 $k \leftarrow k+k$
Μέχρις_ότου ($B > 17$)

α) Ποια θα είναι η τελική τιμή του k ; A) 0 B) 16 Γ) 2 Δ) 14

β) Ποια θα είναι η τελική τιμή του B ; A) 4 B) 16 Γ) 18 Δ) 14

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

27. Πόσες φορές θα εμφανισθεί το μήνυμα Computer;
Α) 10 Β) 3 Γ) 20 Δ) 30 Ε) Υπάρχει λάθος

Αλγόριθμος πχ27
Για i από 2 μέχρι 3
 Για j από 1 μέχρι 10
 Εμφάνισε 'Computer'
 Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος πχ27



28. Τι θα εκτυπώσει το παρακάτω τμήμα αλγορίθμου;
Α) 9 Β) 10 Γ) Τίποτα από τα προηγούμενα

κ ← 9
Αρχή_επανάληψης
 Αν κ ≤ 9 τότε
 Εκτύπωσε κ
 Τέλος_αν
 κ ← κ + 1
Μέχρις_ότου (κ ≥ 10)

29. Πόσες φορές θα εμφανισθεί το μήνυμα Η/Υ;
Α) 10 Β) 100 Γ) Τίποτα από τα προηγούμενα

Για i από 1 μέχρι 10
 Για j από 1 μέχρι 10
 Εμφάνισε i
 Τέλος_επανάληψης
 Εμφάνισε 'Η/Υ'
Τέλος_επανάληψης

30. Έστω ο παρακάτω αλγόριθμος:

Αλγόριθμος πχ30
Για i από 1 μέχρι 10
 Εμφάνισε 'ΚΕΝΤΑΥΡΟΣ'
 Για j από 1 μέχρι 10
 Εμφάνισε 'Centaur'
 Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος πχ30

- α) Πόσες φορές θα εμφανισθεί το μήνυμα ΚΕΝΤΑΥΡΟΣ;
Α) 1 Β) 10 Γ) 100

- β) Πόσες φορές θα εμφανισθεί το μήνυμα Centaur;
Α) 1 Β) 10 Γ) 100



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

31. Πόσες φορές θα εμφανισθεί το μήνυμα στο παρακάτω τμήμα αλγορίθμου;
A) 30 B) 10 Γ) 40

Για i από 1 μέχρι 10 με_βήμα 3
Για j από 1 μέχρι 10
Εμφάνισε 'Πριν τι κακόν παθέειν. Ρεχθέν δε τε νήπιος έγνων'
Τέλος_επανάληψης
Τέλος_επανάληψης

32. Έστω ο παρακάτω αλγόριθμος:

Αλγόριθμος πχ32
Για χ από 1 μέχρι 100
Εμφάνισε 'Ντόναλντ'
Για ρ από 1 μέχρι 10
Εμφάνισε 'Κνούθ'
Τέλος_επανάληψης
Εμφάνισε 'Ντόναλντ_Κνούθ'
Τέλος_επανάληψης
Εμφάνισε 'Donald'
Τέλος πχ32

α. Πόσες φορές θα εμφανισθεί το μήνυμα Κνούθ;
A) 1 B) 10 Γ) 1000

β. Πόσες φορές θα εμφανισθεί το μήνυμα Ντόναλντ;
A) 100 B) 10 Γ) 1000

γ. Πόσες φορές θα εμφανισθεί το μήνυμα Ντόναλντ_Κνούθ;
A) 100 B) 10 Γ) 1000

δ. Πόσες φορές θα εμφανισθεί το μήνυμα Donald;
A) 1 B) 100 Γ) 1000

33. Πόσες φορές θα εμφανισθεί το μήνυμα Δερτούζος;
A) 1 B) 10 Γ) 100 Δ) 1000

Αλγόριθμος πχ33
Για i από 100 μέχρι 10 με_βήμα -10
Για j από 1 μέχρι 10
Εμφάνισε 'Δερτούζος'
Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος πχ33

34. Πόσες φορές θα εμφανισθεί το μήνυμα;
A) 1 B) 6 Γ) 12 Δ) Καμία φορά

Αλγόριθμος πχ34
Για i από 1 μέχρι 6
Για j από 1 μέχρι 4 με_βήμα 3
Εμφάνισε 'Ο φόβος είναι δείγμα εκφυλισμένου μυαλού – Βιργίλιος (70-19π.Χ.)'
Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος πχ34

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

35. Πόσες φορές θα εμφανισθεί το μήνυμα Ποντίκι;
Α) 1 Β) 6 Γ) 12 Δ) Καμία φορά

Αλγόριθμος πχ35
Για i από 1 μέχρι 6
 Για j από 1 μέχρι 3 με_βήμα 3
 Εμφάνισε 'Ποντίκι'
 Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος πχ35



36. Πόσες φορές θα εμφανισθεί το μήνυμα στο παρακάτω τμήμα αλγορίθμου;
Α) 22 Β) 24 Γ) 23 Δ) 30

Για i από 1 μέχρι 6
 Για j από 1 μέχρι 3 με_βήμα 0.5
 Εμφάνισε 'Άνθρωποι δε μινυνθάδιοι τελέθουσιν'
 Τέλος_επανάληψης
Τέλος_επανάληψης

37. Πόσα @ θα εμφανισθούν στο παρακάτω τμήμα αλγορίθμου;
Α) 30 Β) 45 Γ) 15 Δ) 25

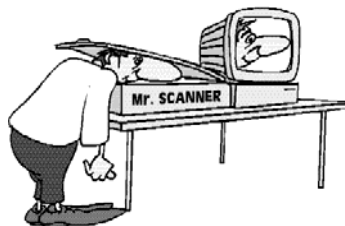
Για i από 1 μέχρι 6 με_βήμα 2
 Για j από 1 μέχρι 3 με_βήμα 0.5
 Εμφάνισε '@@@'
 Τέλος_επανάληψης
Τέλος_επανάληψης

38. Πόσες φορές θα εμφανισθεί το μήνυμα;
Α) 20 Β) 30 Γ) 15 Δ) Υπάρχει λάθος

Αλγόριθμος πχ38
Για i από 1 μέχρι 6
 Για j από 2 μέχρι 4 με_βήμα 0.5
 Εμφάνισε 'Η αμφιβολία δεν με θέλγει λιγότερο από την γνώση - Δάντης (1265-1321)'
 Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος πχ38

39. Πόσες φορές θα εμφανισθεί το μήνυμα;
Α) 4 Β) 5 Γ) 3

Αλγόριθμος πχ39
Για θ από 1 μέχρι 6 με_βήμα 2
 Εμφάνισε 'Η μεγαλοφυΐα είναι 99 τις εκατό ιδρώτας και 1 έμπνευση - Έντισον Τόμας'
 Τέλος_επανάληψης
Τέλος πχ39



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

40. Πόσες φορές θα εμφανισθεί το μήνυμα;
Α) 4 Β) 2 Γ) 3 Δ) 3.2

Αλγόριθμος πχ40

Για i από 3 μέχρι 8 με_βήμα 2

Εμφάνισε 'Ένα πνευματώδες ρητό δεν αποδεικνύει τίποτε – Βολτέρος (1694-1778)'

Τέλος_επανάληψης

Τέλος πχ40

41. Πόσες φορές θα εμφανισθεί το μήνυμα;
Α) 27 Β) 28 Γ) 29 Δ) Υπάρχει λάθος

Αλγόριθμος πχ41

Για i από 1 μέχρι 56 με_βήμα 2

Εμφάνισε 'Η δε αργία πέφυκεν ανθρώποις κακόν'

Τέλος_επανάληψης

Τέλος_επανάληψης

Τέλος πχ41

42. Πόσες φορές θα εμφανισθεί το μήνυμα;
Α) 1 Β) 5 Γ) 0 Δ) Υπάρχει λάθος

Αλγόριθμος πχ42

Για i από 1 μέχρι 5 με_βήμα 25

Εμφάνισε 'Ανήρ δίκαιός εστιν ουχ ο μη αδικών, αλλά όστις αδικείν δυνάμενος μη βούλεται'

Τέλος_επανάληψης

Τέλος πχ42

43. Πόσες φορές θα εμφανισθεί το σύμβολο ♥ ;
Α) Μία Β) Δύο Γ) Καμία

Αλγόριθμος πχ43

Για i από 1 μέχρι 5 με_βήμα 5

Εμφάνισε '♥'

Τέλος_επανάληψης

Τέλος πχ43

44. Πόσες φορές θα εμφανισθεί το μήνυμα;
Α) Μία Β) Δύο Γ) Υπάρχει λάθος

Αλγόριθμος πχ44

Για i από 33 μέχρι 5 με_βήμα -85

Εμφάνισε 'Μένεα πνέω'

Τέλος_επανάληψης

Τέλος πχ44

45. Πόσες φορές θα εμφανισθεί το μήνυμα ΠΗΓΑΣΟΣ;
Α) Δύο Β) Μία Γ) Καμία

Αλγόριθμος πχ45

Για i από 13 μέχρι 15 με_βήμα 5

Εμφάνισε 'ΠΗΓΑΣΟΣ'

Τέλος_επανάληψης

Τέλος πχ45



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

46. Πόσες φορές θα εμφανισθεί το μήνυμα;
 Α) Τέσσερις Β) Έξι Γ) Πέντε Δ) Καμία

Αλγόριθμος πχ46

Για i από 7 μέχρι -3 με_βήμα -2

Εμφάνισε 'Είναι εύκολο να παριστάνεις τον γενναίο από μακριά - Αίσιωπος'

Τέλος_επανάληψης

Τέλος πχ46

47. Συμπληρώστε τον παρακάτω πίνακα:

Όσο ... επανάλαβε	Αρχή_επανάληψης ... Μέχρις_ότου	Για ... από ... μέχρι ... με_βήμα...
$a \leftarrow 8$ Όσο $a > 0$ επανάλαβε Εκτύπωσε a $a \leftarrow a - 1$ Τέλος_επανάληψης		
	$i \leftarrow -5$ $a \leftarrow 1$ Αρχή_επανάληψης $a \leftarrow (a + 2) * i$ $i \leftarrow i + 1$ Μέχρις_ότου $a \geq 0$	

48. Πόσες φορές θα εμφανισθεί το μήνυμα;
 Α) Τέσσερις Β) Έξι Γ) Πέντε Δ) Καμία φορά

Αλγόριθμος πχ48

Για i από 23 μέχρι 11 με_βήμα -4

Εμφάνισε 'Η γλώσσα αμαρτάνουσα τα αληθή λέγει'

Τέλος_επανάληψης

Τέλος πχ48

49. Πόσες φορές θα εμφανισθεί το μήνυμα;
 Α) Τέσσερις Β) Έξι Γ) Πέντε


Αλγόριθμος πχ49

Για i από 10 μέχρι 3 με_βήμα -2

Εμφάνισε 'Τι είναι το πιοτό; Μια απλή διακοπή από το να σκέπτεσαι - Λόρδος Βύρων'


Τέλος_επανάληψης

Τέλος πχ49

50. Πόσες φορές θα εμφανισθεί το σύμβολο  ;
 Α) Δύο Β) Μία Γ) Καμία φορά

Αλγόριθμος πχ50


Για i από 5 μέχρι 5

Εμφάνισε ''

Τέλος_επανάληψης

Τέλος πχ50



Για i από 1 μέχρι 100
 Εμφάνισε ''
 Τέλος_επανάληψης

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

51. Έστω ο παρακάτω πίνακας A με i το δείκτη για τις γραμμές και j το δείκτη για τις στήλες.

3	-1	4	9	0
-5	2	4	0	-3
6	8	5	-2	7
4	0	6	3	1
9	4	8	1	8

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 5
 Για j από 1 μέχρι 5
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3	-1	4	9
-5	2	4	0
6	8	5	-2
4	0	6	3
9	4	8	1

β)

3	-1	4	9	0
-5	2	4	0	-3
6	8	5	-2	7
4	0	6	3	1
9	4	8	1	8

γ) Τίποτα από τα παραπάνω

52. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 4
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3	-1
-5	2
6	8
4	0
9	4

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

β)

3	-1
-5	2

γ)

3	-1
-5	2
6	8
4	0

53. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11
-5	12
62	-18
14	10
9	14

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 1
 Για j από 1 μέχρι 1
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

34

β)

34	-11
----	-----

γ)

-11

54. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11
-5	12
62	-18
14	10
9	14

Το παρακάτω τμήμα αλγορίθμου τι θα εκτυπώσει;

Για i από 1 μέχρι 2
 Για j από 1 μέχρι 1
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

34

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

β)

34	-11
----	-----

γ)

-11
12
-18
10
14

δ) Τίποτα από τα παραπάνω

55. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11
-5	12
62	-18
14	10
9	14
4	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 3 μέχρι 4
 Για j από 1 μέχρι 2
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

62	-18
14	10

β)

34	-11
-5	12
62	-18
14	10

γ)

14	-18
62	10
34	-11

δ) Τίποτα από τα παραπάνω

56. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11	0	8
-5	12	-7	2
6	-18	2	1
14	10	6	7
9	14	4	-19
4	4	5	6

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

Το ακόλουθο τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 5 μέχρι 6
 Για j από 1 μέχρι 2
 Εμφάνισε A[i, j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

9	14
4	4

β)

0	8
-7	2

γ)

9	14	4	-19
4	4	5	6

57. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11	0	8
-5	12	-7	2
6	-18	2	1
14	10	6	7
9	14	4	-19
4	4	5	6

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 3 μέχρι 6
 Για j από 1 μέχρι 5
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

34	-11	0
-5	12	-7
6	-18	2
14	10	6
9	14	4
4	4	5

β) Υπάρχει λάθος

γ)

9	14	4
4	4	5

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

58. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11	0	8
-5	12	-7	2
6	-18	2	1
14	10	6	7
9	14	4	-19
4	4	5	6

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 3 μέχρι 2
 Για j από 1 μέχρι 3
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

34	-11	0
-5	12	-7
6	-18	2
14	10	6
9	14	4
4	4	5

β) Υπάρχει λάθος

γ)

9	14	4
4	4	5

59. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11	0	8
-5	12	-7	2
6	-18	2	1
14	10	6	7
9	14	4	-19
4	4	5	6

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 3 μέχρι 6
 Για j από 1 μέχρι 4
 A[i,j] ← 0
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

34	-11	0
-5	12	-7
6	-18	2
14	10	6
9	14	4
4	4	5

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

β)

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

γ)

9	14	4
4	4	5

60. Έστω ο πίνακας A (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

34	-11	0	8
-5	12	-7	2
6	-18	2	1
14	10	6	7
9	14	4	-19
4	4	5	6

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 3 μέχρι 4
 Για j από 1 μέχρι 4
 Εμφάνισε A[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

34	-11	0
-5	12	-7
6	-18	2
14	10	6
9	14	4
4	4	5

β)

6	-18	2	1
14	10	6	7

γ)

9	14	4
4	4	5

δ) Τίποτα από τα παραπάνω

61. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το ακόλουθο τμήμα αλγορίθμου τι θα εμφανίσει;

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

Για i από 1 μέχρι 4
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 B[i,j] ← -5
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

-5	-5
-5	-5
-5	-5
-5	-5
-5	-5

β)

3	-1
-5	2

γ)

3	-1
-5	2
6	8
4	0

62. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 2
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3	-1
-5	2

β)

6	8
4	0

γ)

3	-1
-5	2
6	8
4	0

δ) Τίποτα από τα παραπάνω

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

63. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 1
 Για j από 1 μέχρι 1
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

-1

β)

3

γ)

1

δ) Τίποτα από τα παραπάνω

64. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 5 με_βήμα 2
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3	-1
6	8
9	4

β)

3	-1
-5	2
6	8

γ) Τίποτα από τα παραπάνω

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

65. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 5 με_βήμα 2
 Για j από 1 μέχρι 2 με_βήμα 10
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3
6
9

β)

3	-1
-5	2
6	8

γ) Τίποτα από τα παραπάνω

66. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 5
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3	-1
6	8
9	4

β)

3	-1
-5	2
6	8

γ) Τίποτα από τα παραπάνω

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

67. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

3	-1
-5	2
6	8
4	0
9	4

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 4
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

3	-1
-5	2
6	8
4	0
9	4

β)

-5	2
6	8
4	0
9	4

γ) Τίποτα από τα παραπάνω

68. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

Γαρύφαλλο	Τουλίπα
Ορχιδέα	Κυκλάμινο
Γιασεμί	Ορτανσία
Αλόη	Γαρύφαλλο

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 1 μέχρι 1
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

Γαρύφαλλο	Τουλίπα
-----------	---------

β)

Γαρύφαλλο	Τουλίπα
Ορχιδέα	Κυκλάμινο

γ) Τίποτα από τα παραπάνω

Ο συμβολισμός των λουλουδιών
Κυκλάμινο= ντροπαλότητα Γιασεμί = γλυκύτητα Γαρύφαλλο= αγάπη, πάθος
Ορτανσία= ματαιοδοξία Ορχιδέα= αρμονία Αλόη= πικρία Τουλίπα= συμπόνια

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

69. Έστω ο πίνακας B (i ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

Κώστας	Περικλής
Στέργιος	Ηλίας
Αθηνά	Αντωνία
Δέσποινα	Γεωργία
Κική	Νίκος

Ο παρακάτω αλγόριθμος τι θα εμφανίσει;

Αλγόριθμος Πχ69
Για i από 2 μέχρι 3
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος Πχ69

α)

Κώστας	Περικλής
--------	----------

β)

Στέργιος

γ)

Κώστας	Περικλής
Στέργιος	Ηλίας

δ) Τίποτα από τα παραπάνω

70. Έστω ο πίνακας B (i είναι ο δείκτης για τις γραμμές και j ο δείκτης για τις στήλες).

B	Γ
A	K
Σ	A
Z	N
A	O

Το παρακάτω τμήμα αλγορίθμου τι θα εμφανίσει;

Για i από 3 μέχρι 1 με_βήμα -1
 Για j από 1 μέχρι 2
 Εμφάνισε B[i,j]
 Τέλος_επανάληψης
Τέλος_επανάληψης

α)

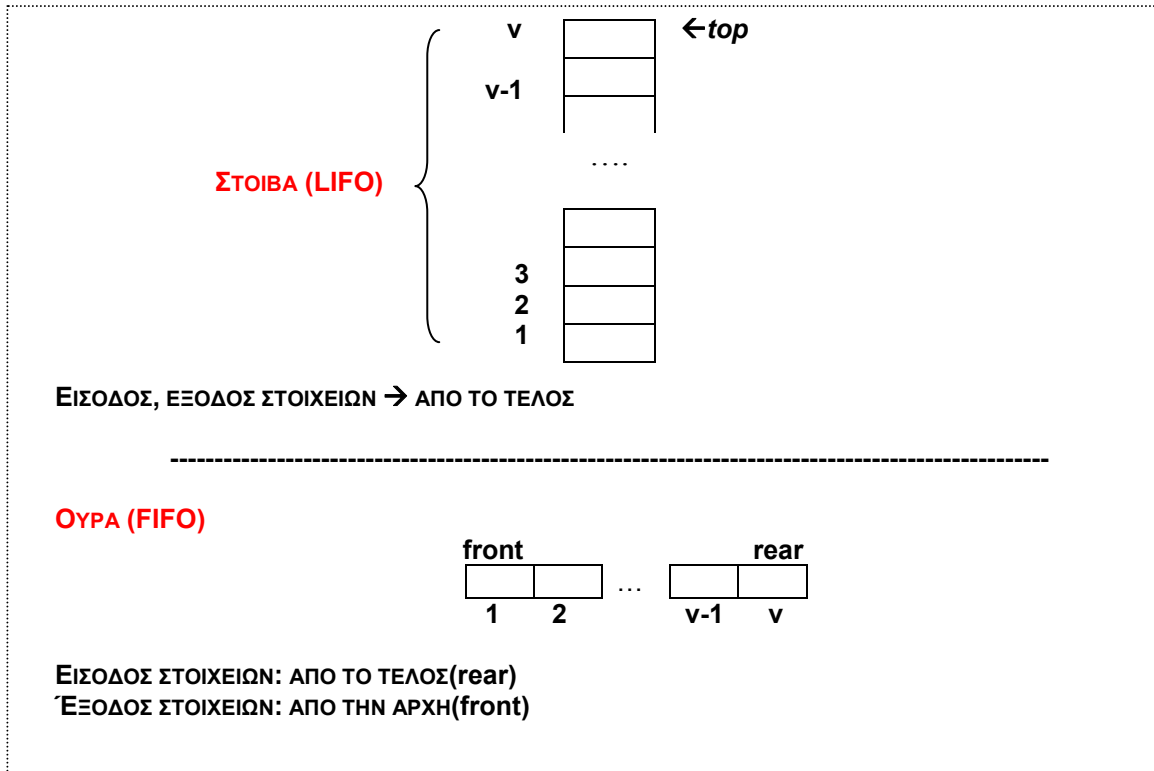
Σ	A
A	K
B	Γ

β)

B	Γ
A	K
Σ	A

γ) Τίποτα από τα παραπάνω

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115



71. Έστω η παρακάτω στοίβα:

Μαρία
Θοδωρής
Δημήτρης
Βασιλική
Αποστόλης

Πώς έγινε η εισαγωγή των ονομάτων;

- α) 1. Μαρία 2. Θοδωρής 3. Δημήτρης 4. Βασιλική 5. Απόστολης
 β) 1. Απόστολης 2. Βασιλική 3. Δημήτρης 4. Θοδωρής 5. Μαρία

72. Έστω η παρακάτω ουρά:

Μαρία	Θοδωρής	Δημήτρης	Βασιλική	Αποστόλης
-------	---------	----------	----------	-----------

Πώς έγινε η εισαγωγή των ονομάτων;

- α) 1. Μαρία 2. Θοδωρής 3. Δημήτρης 4. Βασιλική 5. Απόστολης
 β) 1. Απόστολης 2. Βασιλική 3. Δημήτρης 4. Θοδωρής 5. Μαρία

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

73. Τι θα εμφανίσει ο παρακάτω αλγόριθμος;

```
Αλγόριθμος Πχ72
top ← 0
item ← 1
Όσο top < 5 επανάλαβε
    item ← item * 2
    top ← top + 1
    Στοιβά[top] ← item
Τέλος_επανάληψης
Για i από 1 μέχρι top
    Εμφάνισε Στοιβά[i]
Τέλος_επανάληψης
Τέλος Πχ72
```

α)

32
16
8
4
2

β)

2
4
8
16
32

74. Τι θα εμφανίσει ο παρακάτω αλγόριθμος;

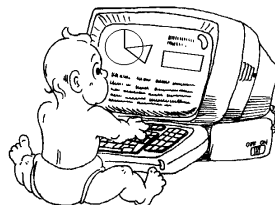
```
Αλγόριθμος Πχ73
top ← 0
Όσο top < 5 επανάλαβε
    top ← top + 1
    A[top] ← top
Τέλος_επανάληψης
Για i από 1 μέχρι top
    Εμφάνισε A[i]
Τέλος_επανάληψης
Τέλος Πχ73
```

α)

5
4
3
2
1

β)

1
2
3
4
5



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Ασκήσεις 1-115

75. Να δοθεί αλγόριθμος ο οποίος θα διαβάζει τον πίνακα ΟΝΟΜΑΤΑ που περιέχει τα ονόματα 100 μαθητών και τον πίνακα ΒΑΘΜΟΙ που περιέχει τους βαθμούς σε 8 μαθήματα των 100 μαθητών. Έπειτα θα υπολογίζει το μέσο όρο βαθμολογίας για κάθε μαθητή και τέλος θα βρίσκει και τυπώνει το όνομα του μαθητή με το μεγαλύτερο μέσο όρο.

ΟΝΟΜΑΤΑ

Κώστας	Δημήτρης	Σοφοκλής
--------	----------	-------	----------

ΒΑΘΜΟΙ

	Κώστας	Δημήτρης	Σοφοκλής
Ιστορία				
Βιολογία				
.....				
Μαθηματικά				

ΜΟ

18,2	14,3	14,5
------	------	-------	------

76. Έστω ο πίνακας ΘΕΡΜΟΚΡΑΣΙΑ που περιέχει τη μέση θερμοκρασία τεσσάρων πόλεων για κάθε μήνα του χρόνου και ο πίνακας ΠΟΛΗ που περιέχει τα ονόματα των πόλεων.

Να γραφεί αλγόριθμος ο οποίος:

- Διαβάζει τα στοιχεία του πίνακα ΘΕΡΜΟΚΡΑΣΙΑ και ΠΟΛΗ.
- Υπολογίζει και εμφανίζει τη μέγιστη θερμοκρασία και την πόλη στην οποία σημειώθηκε.
- Υπολογίζει τη μέση θερμοκρασία κάθε πόλης για όλη τη χρονιά.

ΘΕΡΜΟΚΡΑΣΙΑ

	Αθήνα	Θεσσαλονίκη	Λάρισα	Βόλος
Ιανουάριος				
Φεβρουάριος				
.....				
Δεκέμβριος				

ΠΟΛΗ

Αθήνα	Θεσσαλονίκη	Λάρισα	Βόλος
-------	-------------	--------	-------

77. Να δοθεί αλγόριθμος που θα διαβάζει τον πίνακα ΠΟΝΤΟΙ(περιέχει τους πόντους που έχουν πετύχει 10 παίκτες σε 5 αγώνες καλαθοσφαίρισης) και τον πίνακα ΟΝΟΜΑΤΑ(περιέχει τα ονόματα των παικτών). Έπειτα θα υπολογίζει τον πίνακα ΑΘΡΟΙΣΜΑ που περιέχει τους συνολικούς πόντους του κάθε αθλητή και θα εκτυπώνει τα ονόματα των τριών παικτών που έχουν πετύχει τους περισσότερους πόντους.

ΠΟΝΤΟΙ

	Αγώνας1	Αγώνας2	Αγώνας3	Αγώνας4	Αγώνας5
Παίκτης1					
Παίκτης2					
.....					
Παίκτης10					

ΟΝΟΜΑΤΑ

Σάββας	Ραφαήλ	Νόελ	Μενέλαος
--------	--------	-------	-------	------	----------

ΑΘΡΟΙΣΜΑ

122	36	78	101
-----	----	-------	-------	----	-----

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Ασκήσεις 1-115

78. Το στρατολογικό γραφείο Κορίνθου έχει καλέσει την τρέχουσα κλάση να περάσει περιοδεύον. Για τους 3000 νέους υπάρχουν σε 3 μονοδιάστατους πίνακες ισάριθμων θέσεων τα ονόματα, η χρονολογία γέννησης και το επιλεγμένο σώμα κατάταξης (στρατός ξηράς, ναυτικό, αεροπορία). Να αναπτυχθεί αλγόριθμος όπου:

i. Να διαβάζει το όνομα ενός νέου(μεταβλητή νέος) και να εκτυπώνει τη χρονολογία γέννησης καθώς και το σώμα κατάταξης που επέλεξε.

ii. Να ταξινομεί τους πίνακες με βάση το σώμα κατάταξης(στρατός_ξηράς, ναυτικό, αεροπορία).

ΟΝΟΜΑΤΑ

Στέλιος	Στέφανος	Νίκος	Γιώργος
---------	----------	-------	-------	-------	---------

ΧΡΟΝΟΛΟΓΙΑ ΓΕΝΝΗΣΗΣ

1987	1988	1980	1984
------	------	-------	-------	------	------

ΣΩΜΑ ΚΑΤΑΤΑΞΗΣ

ναυτικό	στρατός ξηράς	αεροπορία	αεροπορία
---------	---------------	-------	-------	-----------	-----------

79. Ένας καθηγητής, έχοντας ως σκοπό να βγάλει κάποια συμπεράσματα από τους βαθμούς των μαθητών του στο μάθημα της Ιστορίας, καταχώρησε τους βαθμούς των 35 μαθητών του σ' ένα μονοδιάστατο πίνακα με το όνομα ΒΑΘΜΟΣ. Για να βγάλει τα συμπεράσματα που θέλει, πρέπει να υπολογίσει τον καλύτερο βαθμό, τον χειρότερο, τον μέσο όρο(MO) καθώς και τη μέγιστη απόκλιση στη βαθμολογία. Να αναπτύξετε αλγόριθμο που θα υπολογίζει τα παραπάνω.

ΒΑΘΜΟΣ

15	18	16	12
----	----	-------	-------	----	----

80. Μια εταιρεία απασχολεί πωλητές. Στον πίνακα ΠΩΛΗΤΗΣ[150] αποθηκεύονται τα ονόματά τους και στον πίνακα ΠΩΛΗΣΕΙΣ[150,365] καταχωρούνται οι ημερήσιες πωλήσεις που πέτυχε κάθε πωλητής τον τελευταίο χρόνο. Να αναπτύξετε αλγόριθμο όπου θα εμφανίζει τον καλύτερο πωλητή της εταιρείας.

ΠΩΛΗΤΗΣ

Σπύρος	Βαγγέλης	Γιάννης	Δήμητρα
--------	----------	-------	-------	---------	---------

ΠΩΛΗΣΕΙΣ

	1 ^η ημέρα	2 ^η ημέρα	364 ^η ημέρα	365 ^η ημέρα
Σπύρος					
Βαγγέλης					
.....					
Γιάννης					
Δήμητρα					

ΣΥΝΟΛΙΚΕΣ ΠΩΛΗΣΕΙΣ

3700	2200	1500	4000
------	------	-------	-------	------	------

81. Στο αγώνισμα των 100 μέτρων συμμετέχουν σ' όλους τους αγώνες 8 αθλητές. Κάθε αθλητής έχει στη φανέλα του ένα νούμερο από το 1 μέχρι το 8. Αν φέτος έγιναν συνολικά 30 αγώνες να δοθεί πρόγραμμα που να καταχωρεί σ' έναν μονοδιάστατο πίνακα το νούμερο του αθλητή που κέρδισε κάθε αγώνα, να υπολογίζει τον αριθμό των νικών κάθε αθλητή και να εμφανίζει τους τρεις αθλητές με τις περισσότερες νίκες.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

ΝΙΚΗΤΗΣ

5	2	4	6
---	---	-------	-------	---	---

ΑΡΙΘΜΟΣ ΝΙΚΩΝ

1	0	3	0
---	---	-------	-------	---	---

ΝΟΥΜΕΡΑ

1	2	7	8
---	---	-------	-------	---	---

82. Μια ομάδα συμμετείχε σε 15 αγώνες. Να γίνει αλγόριθμος ο οποίος θα δέχεται τους πόντους που σημείωσαν οι δέκα παίκτες σε κάθε αγώνα και θα υπολογίζει πόσους πόντους επέτυχαν συνολικά ο καθένας. Έπειτα θα εμφανίζει το νούμερο του παίκτη που σημείωσε τους περισσότερους πόντους καθώς και το σύνολο των πόντων που επέτυχε η ομάδα σε κάθε αγώνα.

ΠΟΝΤΟΙ

	Παίκτης1	Παίκτης2	Παίκτης9	Παίκτης10
Αγώνας1					
Αγώνας2					
.....					
Αγώνας15					

ΣΥΝΟΛΙΚΟΙ ΠΟΝΤΟΙ ΑΓΩΝΩΝ

86	92	91	100
----	----	------	-----	----	-----

ΣΥΝΟΛΙΚΟΙ ΠΟΝΤΟΙ ΠΑΙΚΤΩΝ

150	115	80	121
-----	-----	-------	-------	----	-----

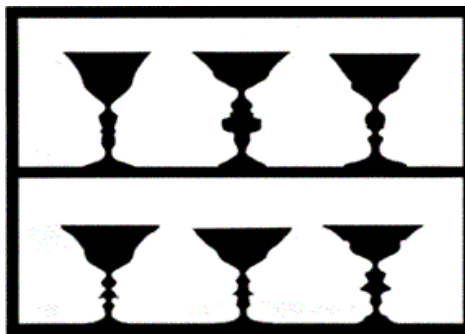
83. Έχουμε σ' έναν πίνακα ονόματα από έξι δισκοπότηρα. Σ' έναν άλλον πίνακα έχουμε έξι πρόσωπα με διαφορετικά ονόματα. Το πρώτο δισκοπότηρο ανήκει στο πρώτο πρόσωπο, το δεύτερο στο επόμενο πρόσωπο κ.ο.κ. Ψάχνουμε να βρούμε, από τον πίνακα με τα πρόσωπα, τον Αντώνη. Θέλουμε όμως και το όνομα από το δισκοπότηρό του...

ΔΙΣΚΟΠΟΤΗΡΑ

Δισκοπότηρο1	Δισκοπότηρο2	Δισκοπότηρο5	Δισκοπότηρο6
--------------	--------------	------	-----	--------------	--------------

ΠΡΟΣΩΠΑ

Αντώνης	Γιώργος	Κώστας	Μαρία
---------	---------	------	-----	--------	-------



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

84. Βρείτε ποιο δεν είναι ισοδύναμο του ακόλουθου κώδικα:

Αν $A[j-1] < A[j]$ τότε
αντιμετάθεσε $A[j-1], A[j]$
Τέλος_αν

α) Αν $A[j-1] < A[j]$ τότε
Βοηθητική $\leftarrow A[j-1]$
 $A[j-1] \leftarrow A[j]$
 $A[j] \leftarrow$ Βοηθητική
Τέλος_αν

β) Αν $A[j-1] < A[j]$ τότε
αντιμετάθεσε $A[j], A[j-1]$
Τέλος_αν

γ) Αν $A[j] > A[j-1]$ τότε
αντιμετάθεσε $A[j-1], A[j]$
Τέλος_αν

δ) Αν $A[j] < A[j-1]$ τότε
αντιμετάθεσε $A[j-1], A[j]$
Τέλος_αν

85. Βρείτε ποιο δεν είναι ισοδύναμο του ακόλουθου κώδικα:

Για i από 1 μέχρι 10
Εμφάνισε $A[i]$
Τέλος_επανάληψης

α) Για j από 1 μέχρι 10
Εμφάνισε $A[j]$
Τέλος_επανάληψης

β) $k \leftarrow 1$
Όσο $k < 11$ επανάλαβε
Εμφάνισε $A[k]$
 $k \leftarrow k+1$
Τέλος_επανάληψης

γ) Για λ από 1 μέχρι 10
Εκτύπωσε $A[\lambda]$
Τέλος_επανάληψης

δ) $\zeta \leftarrow 1$
Εμφάνισε $A[\zeta]$
Αρχή_επανάληψης
 $\zeta \leftarrow \zeta+1$
Εμφάνισε $A[\zeta]$
Μέχρις_ότου $\zeta \geq 10$

ε) Για i από 10 μέχρι 1 με_βήμα -1
Εμφάνισε $A[i]$
Τέλος_επανάληψης

στ) $\zeta \leftarrow 0$
Αρχή_επανάληψης
 $\zeta \leftarrow \zeta+1$
Εμφάνισε $A[\zeta]$
Μέχρις_ότου $\zeta \geq 10$

86. Βρείτε ποιο δεν είναι ισοδύναμο του ακόλουθου κώδικα:

$A \leftarrow 10, B \leftarrow 12$
αντιμετάθεσε A, B

α) $A \leftarrow 10, B \leftarrow 12$
 $A \leftarrow A+B$
 $B \leftarrow A-B$
 $A \leftarrow A-B$

β) $A \leftarrow 10, B \leftarrow 12$
 $A \leftarrow A*B$
 $B \leftarrow A/B$
 $A \leftarrow A/B$

γ) $A \leftarrow 10, B \leftarrow 12$
 $A \leftarrow B$
 $B \leftarrow A$
 $A \leftarrow B$

δ) $A \leftarrow 10, B \leftarrow 12$
 $\Gamma \leftarrow A$
 $A \leftarrow B$
 $B \leftarrow \Gamma$

ε) $A \leftarrow 10, B \leftarrow 12$
 $A \leftarrow 12$
 $B \leftarrow 10$

87. Πώς δηλώνουμε μια σταθερά A με τιμή 5;

α) **ΜΕΤΑΒΛΗΤΕΣ**
 $A \leftarrow 5$

β) **ΣΤΑΘΕΡΕΣ**
 $A \leftarrow 5$

γ) **ΣΤΑΘΕΡΕΣ**
 $A=5$

δ) **ΣΤΑΘΕΡΕΣ**
ΑΚΕΡΑΙΕΣ: A
 $A=5$

ε) **ΣΤΑΘΕΡΕΣ**
ΑΚΕΡΑΙΕΣ: A
 $A \leftarrow 5$

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

88. Έχουμε τις παρακάτω δέκα προτάσεις. Ποιες από αυτές είναι αληθείς και ποιες ψευδείς;

- 1) Μία μόνο από αυτές τις προτάσεις είναι ψευδής.
- 2) Δύο μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 3) Τρεις μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 4) Τέσσερις μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 5) Πέντε μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 6) Έξι μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 7) Επτά μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 8) Οκτώ μόνο από αυτές τις προτάσεις είναι ψευδείς.
- 9) Εννέα από αυτές τις προτάσεις είναι ψευδείς.

89. Κατά τη διάρκεια διεθνών αγώνων αυτοκινήτου έλαβαν μέρος 10 αυτοκίνητα.

- α) Αν κάποιο αυτοκίνητο ξεπεράσει το δεύτερο τι θέση θα πάρει;
- β) Είναι δυνατόν κάποιο αυτοκίνητο να ξεπεράσει το τέταρτο χωρίς προηγουμένως να έχει προσπεράσει το πέμπτο;

90. Αν ο πίνακας A είναι:

11	17	28	21	2
----	----	----	----	---

Ποια θα είναι η τελική μορφή του A μετά την εκτέλεση του ακόλουθου τμήματος αλγορίθμου;

Για i από 2 μέχρι 5
 Για j από 5 μέχρι i με_βήμα -1
 Αν $A[j-1] < A[j]$ τότε
 αντιμετάθεσε $A[j-1], A[j]$
 Τέλος_αν
Τέλος_επανάληψης
Τέλος_επανάληψης

α)

11	17	28	21	2
----	----	----	----	---

β)

2	11	17	21	28
---	----	----	----	----

γ)

28	21	17	11	2
----	----	----	----	---

91. Αν ο πίνακας A είναι:

11	17	28	21	2
----	----	----	----	---

Ποια θα είναι η τελική μορφή του A μετά την εκτέλεση του ακόλουθου τμήματος αλγορίθμου;

Για i από 2 μέχρι 5
 Για j από 5 μέχρι i με_βήμα -1
 Αν $A[j] < A[j-1]$ τότε
 αντιμετάθεσε $A[j-1], A[j]$
 Τέλος_αν
Τέλος_επανάληψης
Τέλος_επανάληψης

α)

11	17	28	21	2
----	----	----	----	---

β)

2	11	17	21	28
---	----	----	----	----

γ)

28	21	17	11	2
----	----	----	----	---

92. Έστω πίνακας A που περιέχει τις χρονολογίες γέννησης κάποιων φοιτητών:

1983	1990	1982	1985	1986
------	------	------	------	------

και πίνακας E με τα αντίστοιχα ονόματα των φοιτητών:

Σπύρος	Ανέστης	Βασίλης	Ελένη	Κώστας
--------	---------	---------	-------	--------

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

Αν ταξινομήσουμε τον πίνακα Α κατά αύξουσα σειρά θα πρέπει προφανώς να ταξινομήσουμε και τον πίνακα Ε. Ποιο θα είναι τότε το τελευταίο στοιχείο του πίνακα Ε;

93. Έχουμε σ' έναν πίνακα Α τα ονόματα κάποιων τραγουδιστών:

Βανδή	Evanescence	Ρέμος	Safri_Duo	Nirvana	Minogue	Madonna
-------	-------------	-------	-----------	---------	---------	---------

και σ' έναν άλλον πίνακα Β κάποια τραγούδια. Το πρώτο τραγούδι αντιστοιχεί στον πρώτο τραγουδιστή κ.ο.κ.

Για	Imaginary	Μείνε	Played_a_live	Smells_like_Teen_Spirit	Slow	Music
-----	-----------	-------	---------------	-------------------------	------	-------

Θέλουμε να γράψουμε ένα CD με τα τραγούδια του Ρέμου και της Minogue. Συμπληρώστε τις τελίτσες στο ακόλουθο τμήμα αλγορίθμου:

Για i από ... μέχρι

Αν (A[i]='Ρέμος') (A[i]='Minogue') τότε

Εμφάνισε 'Το τραγούδι που θα γραφτεί στο CD είναι: ',

Τέλος_αν

Τέλος_επανάληψης

94. Η Σειριακή αναζήτηση: Είναι η πιο απλή, αλλά και η λιγότερο αποτελεσματική μέθοδος αναζήτησης έτσι η χρήση της δικαιολογείται μόνο σε περιπτώσεις όπου:

- Ο πίνακας είναι μη ταξινομημένος (Σωστό ή Λάθος)
- Ο πίνακας είναι μικρού μεγέθους (Σωστό ή Λάθος)
- Η αναζήτηση στοιχείων του πίνακα γίνεται σπάνια (Σωστό ή Λάθος)
- Η αναζήτηση ξεκινάει πάντα από την αρχή (Σωστό ή Λάθος)
- Ο πίνακας περιέχει αριθμούς (Σωστό ή Λάθος)

95. Βρείτε τα δύο λάθη στον παρακάτω αλγόριθμο:

1. Αλγόριθμος πλήθος_ζητούμενου
2. Για θέση από 1 μέχρι 100
3. Διάβασε A[θέση]
4. Τέλος
5. Εμφάνισε 'Δώστε το ζητούμενο: '
6. Διάβασε ζητούμενο

7. βρέθηκε ← Ψευδής
8. φορές ← 0
9. Για θέση από 1 μέχρι 100
10. Αν A[θέση]=ζητούμενο τότε
11. βρέθηκε ← Αληθής
12. φορές ← φορές+1
13. Τέλος_αν
14. Τέλος_επανάληψης

15. Αν βρέθηκε=Ψευδής τότε
16. Εμφάνισε 'Βρέθηκε ', φορές, ' Φορές'
17. αλλιώς
18. Εμφάνισε 'Δε βρέθηκε'
19. Τέλος_αν
20. Τέλος πλήθος_ζητούμενου

α) Τι είδους λάθη είναι αυτά;

β) Αν γράφαμε τον παραπάνω αλγόριθμο σε μία γλώσσα προγραμματισμού ποιο από τα δύο λάθη θα μπορούσε να ανιχνευθεί από το μεταγλωττιστή;

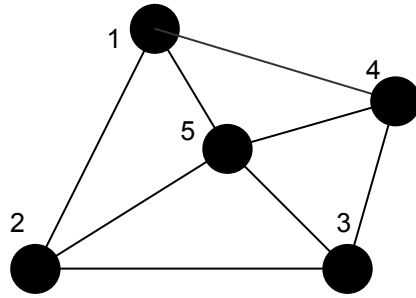
Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

96. Έστω ότι αντιμετωπίζουμε το πρόβλημα ενός ταχυδρομικού διανομέα, που πρέπει να ξεκινήσει από ένα χωριό, να επισκεφθεί έναν αριθμό από γειτονικά χωριά, για να μοιράσει ένα σύνολο επιστολών και να επιστρέψει στο χωριό από όπου ξεκίνησε περνώντας μόνο μία φορά από κάθε χωριό. Το πρόβλημα έγκειται στην εύρεση της καλύτερης διαδρομής, έτσι ώστε ο διανομέας να διανύσει το μικρότερο αριθμό χιλιομέτρων.

Δήμος Μαστοροχωριών

- 1: Δροσοπηγή
- 2: Πλαγιά
- 3: Ζέρμα
- 4: Πυρσόγιαννη
- 5: Λυκόραχη

- Οι αποστάσεις είναι:
- 1-2: 3 Km
 - 1-4: 2 Km
 - 1-5: 7 Km
 - 2-3: 18 Km
 - 2-5: 5 Km
 - 3-4: 3 Km
 - 3-5: 4 Km
 - 4-5: 2 Km



Στο σχήμα οι πέντε αριθμημένοι κόμβοι αντιστοιχούν στα πέντε χωριά, ενώ οι οχτώ συνδέσεις αντιστοιχούν στις οδικές αρτηρίες που ενώνουν τα χωριά αυτά. Τέλος οι ακέραιοι που χαρακτηρίζουν τις συνδέσεις μεταξύ των κύκλων παρουσιάζουν τις αντίστοιχες χιλιομετρικές αποστάσεις μεταξύ των χωριών. Επιπλέον ας υποθέσουμε ότι ο ταχυδρόμος ξεκινά από το χωριό 1 και βέβαια σ' αυτό πρέπει να καταλήξει, αφού επισκεφθεί τα χωριά 2, 3, 4 και 5. Στο πρόβλημα αυτό μπορούν να υπάρξουν διάφορες προσεγγίσεις για την ανάλυση και την επίλυσή του.

α) Μια πρώτη ανάλυση του προβλήματος είναι:

- ❖ να γίνει καταγραφεί όλων των αποστάσεων μεταξύ των χωριών,
- ❖ να ταξινομηθούν οι συνδέσεις των χωριών κατά αύξουσα χιλιομετρική απόσταση,
- ❖ να επιλέγεται κάθε φορά η μετάβαση από το χωριό όπου βρίσκεται ο διανομέας προς το πλησιέστερο χωριό.

Με βάση τα προηγούμενα ο ταχυδρόμος ποια σειρά επίσκεψης χωριών θα διάλεγε;

- A. (1) → (5) → (4) → (3) → (2) → (1)
- B. (1) → (4) → (5) → (3) → (2) → (1)
- Γ. (1) → (2) → (3) → (5) → (4) → (1)
- Δ. (1) → (3) → (4) → (5) → (2) → (1)

β) Μια διαφορετική ανάλυση του προβλήματος είναι:

- ❖ να γίνει καταγραφεί όλων των αποστάσεων μεταξύ των χωριών,
- ❖ να βρεθεί μια σειρά επίσκεψης των χωριών με στόχο την ελαχιστοποίηση της συνολικής απόστασης και όχι την ελαχιστοποίηση της κάθε φορά απόστασης.

Με βάση τα προηγούμενα ο ταχυδρόμος ποια σειρά επίσκεψης χωριών θα διάλεγε;

- A. (1) → (5) → (4) → (3) → (2) → (1)
- B. (1) → (4) → (5) → (3) → (2) → (1)
- Γ. (1) → (2) → (3) → (5) → (4) → (1)
- Δ. (1) → (2) → (5) → (3) → (4) → (1)

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

97. Ταιριάξτε κάθε στοιχείο της στήλης Α μ' ένα από τη στήλη Β.

Στήλη Α	Στήλη Β
Α. Γλώσσα μηχανής	1. BASIC, PASCAL, C
Β. Μεταγλωττιστής	2. Συμβολομεταφραστής
Γ. Συμβολικές γλώσσες	3. Αντικείμενο πρόγραμμα
Δ. Συνδέτης-Φορτωτής	4. Βιβλιοθήκες
Ε. Γλώσσες υψηλού επιπέδου	5. Ακολουθίες από 0 και 1 6. Διερμηνευτής

98. Βάλτε στη σωστή σειρά τα ακόλουθα:

- | | |
|--------------------------|---------|
| Α. Εκτελέσιμο πρόγραμμα | 1. |
| Β. Πηγαίο πρόγραμμα | 2. |
| Γ. Αντικείμενο πρόγραμμα | 3. |
| Δ. Μεταγλωττιστής | 4. |
| Ε. Συνδέτης | 5. |

99. Κάντε τις σωστές αντιστοιχίες ανάμεσα στις δύο στήλες.

Στήλη Α	Στήλη Β
1. COBOL	
2. PASCAL	Μεταγλωττιστής
3. C	
4. BASIC	Διερμηνευτής
5. LISP	
6. PROLOG	
7. FORTRAN	

100. Ποια από τα παρακάτω αποτελούν χαρακτηριστικά των γλωσσών υψηλού επιπέδου;

- 1) Ο φυσικότερος και πιο 'ανθρώπινος' τρόπος έκφρασης των προβλημάτων. Τα προγράμματα σε γλώσσα υψηλού επιπέδου είναι πιο 'κοντά' στα προβλήματα που επιλύουν.
- 2) Η ανεξαρτησία από τον τύπο της μηχανής. Προγράμματα σε μία γλώσσα υψηλού επιπέδου γραμμένα για έναν υπολογιστή μπορούν να εκτελεστούν σε άλλους με μικρές ή καθόλου μετατροπές.
- 3) Η ευκολία της εκμάθησης και εκπαίδευσης ως απόρροια των δύο προηγούμενων.
- 4) Η διόρθωση λαθών και η συντήρηση των προγραμμάτων, δηλαδή η τροποποίηση και η βελτίωσή τους, είναι σημαντικά ευκολότερο έργο.
- 5) Συνολικά οι γλώσσες υψηλού επιπέδου ελάττωσαν σημαντικά το χρόνο και το κόστος παραγωγής νέων προγραμμάτων, αφού λιγότεροι προγραμματιστές μπορούν τώρα σε λιγότερο χρόνο να αναπτύξουν προγράμματα που χρησιμοποιούνται σε περισσότερους υπολογιστές.
- 6) Όλα τα προγράμματα σε γλώσσα υψηλού επιπέδου γράφονται αποκλειστικά από έμπειρους προγραμματιστές.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

101. Να συμπληρωθούν τα κενά με τον κατάλληλο κώδικα.

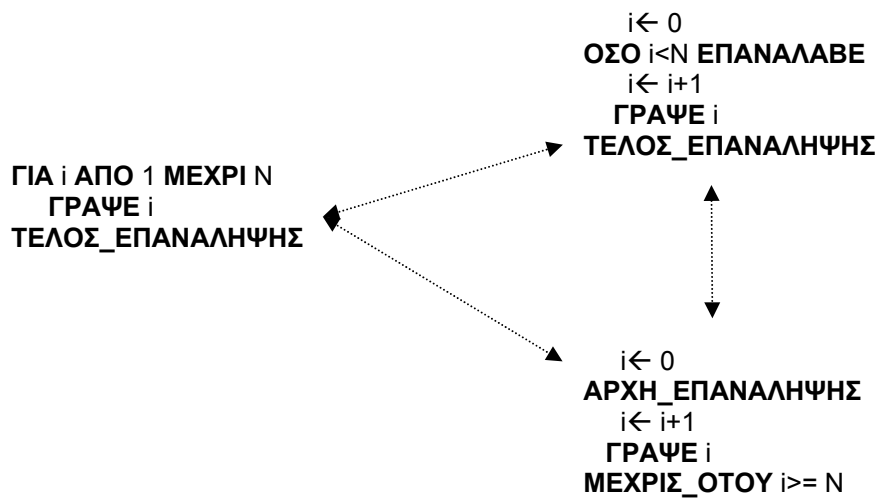
	Πίνακας $N \times M$
Ορισμός πίνακα.	
Εισαγωγή δεδομένων.	
Μηδενισμός πίνακα.	
Άθροισμα 1 ^{ης} Στήλης.	
Άθροισμα K στήλης. ($1 \leq K \leq M$)	
Άθροισμα τελευταίας στήλης.	
Άθροισμα 1 ^{ης} γραμμής.	
Άθροισμα K γραμμής. ($1 \leq K \leq N$)	
Άθροισμα τελευταίας γραμμής.	
Άθροισμα όλων των στοιχείων του πίνακα.	
Άθροισμα των στοιχείων της πρώτης και τρίτης γραμμής.	
Άθροισμα των στοιχείων της πρώτης και τρίτης στήλης.	

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
 Ασκήσεις 1-115

102. Να συμπληρωθούν τα κενά με τον κατάλληλο κώδικα.

Άθροισμα πρώτης διαγωνίου.		
Άθροισμα δεύτερης διαγωνίου.		

103. Τι τιμές πρέπει να πάρει το N ώστε οι παρακάτω δομές επανάληψης να είναι ισοδύναμες;



104. Έστω ο πίνακας A:

4	6
0	1
1	2

Τι θα εμφανίσει το παρακάτω τμήμα προγράμματος αν βάλουμε την εντολή sum ← 0 στη θέση:

- α) 1. β) 2. γ) 3. δ) 4.

```

1. ....
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 3
2. ....
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 2
3. ....
   sum ← sum+A[i,j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ sum
4. ....
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    
```


Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

105. Να αντιστοιχήσετε τα γράμματα της στήλης Α με τους αριθμούς της στήλης Β

A.	B.
A. Διερμηνευτής	1. Δέχεται ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής.
B. Μεταγλωττιστής	2. Μεθοδολογία σύνταξης προγραμμάτων η οποία βοηθάει στην ανάπτυξη σύνθετων προγραμμάτων και στην γενικότερη διαχείριση τους.
Γ. Δομημένος προγραμματισμός	3. Χρησιμοποιείται για την σύνταξη και διόρθωση προγραμμάτων.
Δ. Συντάκτης	4. Διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μία εκτελεί αμέσως μία ισοδύναμη ακολουθία εντολών μηχανής.

106. Απαντήστε αν η κάθε πρόταση είναι σωστή ή λάθος.

- α)** Ιεραρχική σχεδίαση προγράμματος ονομάζεται η τεχνική της διάσπασης του προβλήματος σε μία σειρά από απλούστερα υποπροβλήματα, τα οποία να είναι εύκολο να επιλυθούν, οδηγώντας στην επίλυση του αρχικού προβλήματος.
- β)** Ο πίνακας είναι ένα παράδειγμα στατικής δομής δεδομένων.
- γ)** Οι δυναμικές δομές στηρίζονται στην τεχνική της λεγόμενης δυναμικής παραχώρησης μνήμης.
- δ)** Οι φυσικές γλώσσες εξελίσσονται συνεχώς, ενώ οι τεχνητές χαρακτηρίζονται από στασιμότητα.
- ε)** Όλες οι γλώσσες προγραμματισμού προσφέρουν τη δυνατότητα δυναμικής παραχώρησης μνήμης.
- στ)** Στον οπτικό προγραμματισμό μας δίνεται η δυνατότητα δημιουργίας κλάσεων αντικειμένων μέσω ειδικών εργαλείων.

107. Να συμπληρωθούν τα κενά στο ακόλουθο πρόγραμμα και να υπολογισθεί τι θα εμφανισθεί με την εντολή: ΓΡΑΨΕ Sum+j .

ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:.....

ΠΡΑΓΜΑΤΙΚΕΣ:..... ,

ΑΡΧΗ

Sum ← 0

j ← 1

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ ... ΜΕ ΒΗΜΑ 0.5

Sum ← Sum+ i

j ← j+1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Στο σημείο αυτό του προγράμματος το j είναι 10

ΓΡΑΨΕ Sum+j ←

j=10

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

108. Πόσες αντιμεταθέσεις θα γίνουν στον πίνακα A κατά την εκτέλεση του ακόλουθου κώδικα:

```

Για i από 2 μέχρι 5
  Για j από 5 μέχρι i με_βήμα -1
    Αν A[j-1] < A[j] τότε
      Βοηθητική ← A[j-1]
      A[j-1] ← A[j]
      A[j] ← Βοηθητική
    Τέλος_αν
  Τέλος_επανάληψης
Τέλος_επανάληψης
    
```

Ο πίνακας A:

1 ^η θέση	2 ^η θέση	3 ^η θέση	4 ^η θέση	5 ^η θέση
35	66	12	55	48

109. Αν ο πίνακας A είναι:

Z	B	M	N	K
---	---	---	---	---

Ποια θα είναι η τελική μορφή του A μετά την εκτέλεση του ακόλουθου τμήματος αλγορίθμου;

```

Για i από 2 μέχρι 5
  Για j από 5 μέχρι i με_βήμα -1
    Αν A[j-1] > A[j] τότε
      αντιμετάθεσε A[j-1], A[j]
    Τέλος_αν
  Τέλος_επανάληψης
Τέλος_επανάληψης
    
```

α)

B	Z	K	M	N
---	---	---	---	---

β)

N	M	K	Z	B
---	---	---	---	---

γ)

M	N	K	Z	B
---	---	---	---	---

110. Αν ο πίνακας A είναι:

Z	B	M	N	K
---	---	---	---	---

και ο B είναι:

3	9	3	2	0
---	---	---	---	---

Ποια θα είναι η τελική μορφή του B μετά την εκτέλεση του ακόλουθου τμήματος αλγορίθμου;

```

Για i από 2 μέχρι 5
  Για j από 5 μέχρι i με_βήμα -1
    Αν* A[j-1] < A[j] τότε
      αντιμετάθεσε A[j-1], A[j]
      αντιμετάθεσε B[j-1], B[j]
    Τέλος_επανάληψης
  Τέλος_επανάληψης
    
```

α)

9	3	0	3	2
---	---	---	---	---

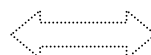
β)

2	3	0	3	9
---	---	---	---	---

γ)

3	9	3	2	0
---	---	---	---	---

* Αν <συνθήκη> τότε
εντολή



Αν <συνθήκη> τότε
εντολή
Τέλος_αν

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

111. Βρείτε ποιες τιμές του λ θα εμφανισθούν.

ΠΡΟΓΡΑΜΜΑ ΑΣΚ111
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: α, ι
ΑΡΧΗ
α ← 4
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 3
ΚΑΛΕΣΕ ΔΙΑΔ(α,ι)
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ(κ,λ)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: κ, λ, ι
ΑΡΧΗ
ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ κ
ΓΡΑΨΕ λ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

α) 111	β) 444	γ) 111
2222	222	222
3333	111	444

112. Ποιες θα είναι οι τελικές τιμές των μεταβλητών α και β;

ΠΡΟΓΡΑΜΜΑ ΑΣΚ112
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: α, β
ΑΡΧΗ
α ← 4
β ← 2
ΚΑΛΕΣΕ ΔΙΑΔ(α,β)
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ(κ,π)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: κ, π
ΑΡΧΗ
ΌΣΟ κ>π ΕΠΑΝΑΛΑΒΕ
κ ← κ-1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

α) α=3 και β=2 β) α=2 και β=3 γ) α=2 και β=2

113. Ποιες θα είναι οι τελικές τιμές των μεταβλητών α και β;

ΠΡΟΓΡΑΜΜΑ ΑΣΚ113
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: α, β
ΑΡΧΗ
α ← 5
β ← 1
α ← Συν1(α,β)
α ← Συν1(β,α)
β ← Συν1(α,β)
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

ΣΥΝΑΡΤΗΣΗ Συν1(κ,π):**ΑΚΕΡΑΙΑ**
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: κ, π

ΑΡΧΗ

ΌΣΟ κ>π **ΕΠΑΝΑΛΑΒΕ**

κ ← κ-3

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Συν1 ← κ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

α) α= -2 β) α= 2 γ) α= -2
β= 2 β= -2 β= -2

114. Αν ο πίνακας A είναι:

2	5	3	-1	0	6	6	3	8	2
---	---	---	----	---	---	---	---	---	---

Ποια θα είναι η τελική μορφή του πίνακα A μετά την εκτέλεση του ακόλουθου προγράμματος;

ΠΡΟΓΡΑΜΜΑ Ταξινόμηση_φυσσαλίδας

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: A[10]

ΑΚΕΡΑΙΕΣ: i, j

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 10

ΓΙΑ j **ΑΠΟ** 10 **ΜΕΧΡΙ** i **ΜΕ ΒΗΜΑ** -1

ΑΝ A[j-1]>A[j] **ΤΟΤΕ**

ΚΑΛΕΣΕ Αντιμετάθεσε(A[j-1],A[j])

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Αντιμετάθεσε(αριθμός1, αριθμός2)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: αριθμός1, αριθμός2, προσωρινή

ΑΡΧΗ

προσωρινή ← αριθμός1

αριθμός1 ← αριθμός2

αριθμός2 ← προσωρινή

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

α)

0	1	2	2	3	3	5	6	6	8
---	---	---	---	---	---	---	---	---	---

β)

-1	0	2	2	3	3	5	6	6	8
----	---	---	---	---	---	---	---	---	---

γ)

8	6	6	5	3	3	2	2	0	-1
---	---	---	---	---	---	---	---	---	----

Υπενθυμίζουμε ότι στο πρόγραμμα δεν μπορούμε να χρησιμοποιήσουμε την εντολή: **αντιμετάθεσε**.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Ασκήσεις 1-115

115. Ποιες τιμές θα πάρει η μεταβλητή x στο παρακάτω πρόγραμμα;

```
ΠΡΟΓΡΑΜΜΑ Collatz
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: x
ΑΡΧΗ
  x ← 5
  ΟΣΟ (x <> 1) ΕΠΑΝΑΛΑΒΕ
    ΑΝ Είναι_άρτιος(x) = ΑΛΗΘΗΣ ΤΟΤΕ
      x ← x div 2
    ΑΛΛΙΩΣ
      x ← x*3+1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
ΣΥΝΑΡΤΗΣΗ Είναι_άρτιος(α): ΛΟΓΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: α
ΑΡΧΗ
  ΑΝ α MOD 2 = 0 ΤΟΤΕ
    Είναι_άρτιος ← ΑΛΗΘΗΣ
  ΑΛΛΙΩΣ
    Είναι_άρτιος ← ΨΕΥΔΗΣ
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

α) 5, 16, 8, 4, 2, 1 β) 5, 2 γ) 5, 16, 0 δ) 5, 12, 6, 3, 1

- Αν η αρχική τιμή του x ήταν 22 και όχι 5, ποιες άλλες τιμές θα έπαιρνε το x ;

α) 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 0

β) 67, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

γ) 11, 34, 17, 52, 26, 13, 6, 3, 1

δ) 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

┌ - ┌ ┌ - ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌

Υπάρχουν κάποιες στιλιστικές οδηγίες τις οποίες μπορείτε να ακολουθήσετε για να σας βοηθήσουν να γράφετε καλύτερα προγράμματα. Οι ακόλουθες είναι μερικές από τις σημαντικότερες:

- Χρησιμοποιείτε σχόλια για να λέτε στους αναγνώστες σας τι χρειάζεται να ξέρουν. Εξηγείτε οτιδήποτε νομίζετε ότι είναι περίπλοκο ή μπορεί να είναι δύσκολο για κάποιον να το καταλάβει διαβάζοντας μόνο το ίδιο το πρόγραμμα.
- Χρησιμοποιείτε οδόντωση για να σημειώνετε τα διάφορα επίπεδα ελέγχου του προγράμματος. Η προσεκτική χρήση της οδόντωσης είναι σημαντική για την αναγνωσιμότητα και καθιστά τη δομή του προγράμματος πολύ πιο σαφή.
- Χρησιμοποιείτε ονόματα με νόημα.
- Αποφεύγετε μη αναγκαία πολυπλοκότητα. Αξίζει συχνά να θυσιάζετε κάποια αποδοτικότητα για χάρη της αναγνωσιμότητας.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Απαντήσεις 1-115

1. 1-δ, 2-α, 3-β

2. 1-β, 2-γ, 3-α

3. 1-ε, 2-β, 3-δ, 4-στ

4. 1-β, 2-α, 3-γ, 4-β

5. 1-β και δ, 2-ε, 3-γ

6. 1-Ε, 2-Γ, 3-Δ

7. 1-Β, 2-Α, 3-Δ

8. 1-Α, 2-Γ, 3-Δ

9.

i.	Τιμή	Αποδεκτό
ii.	Τιμή-1	Μη αποδεκτός ο χαρακτήρας "-", είναι το σύμβολο της αφαίρεσης
iii.	Τιμή_2	Αποδεκτό
iv.	Egfsstt	Αποδεκτό, ο αλγόριθμος δεν ενδιαφέρεται για την ορθογραφία
v.	Τιμή.δ	Μη αποδεκτός ο χαρακτήρας ".", χρησιμοποιείται στους δεκαδικούς αριθμούς
vi.	τ	Αποδεκτό
vii.	Γ/ε4	Μη αποδεκτός ο χαρακτήρας "/", είναι σύμβολο της διαίρεσης
viii.	τιμή 2	Μη αποδεκτός ο χαρακτήρας " "(κενό), πρέπει να έχουμε μία λέξη
ix.	ΚΟΣΤΑΣ	Αποδεκτό, ο αλγόριθμος δεν ενδιαφέρεται για την ορθογραφία
x.	2α	Μη αποδεκτό, δεν επιτρέπεται το όνομα να ξεκινά από αριθμητικό ψηφίο

10.

i.	τιμή ← "αρκετά"	Σωστή
ii.	τιμή ← 14 + 5	Σωστή
iii.	α+τιμή ← 2	Λάθος
iv.	τιμή ← υσ ← 2	Λάθος
v.	τιμή ← 8*7 + 4	Σωστή
vi.	τιμή ← "τιμή"	Σωστή
vii.	τιμή ← 4/0	Λάθος
viii.	2*τιμή ← 8	Λάθος
ix.	τιμή ← 3*8^0	Σωστή
x.	Διάβασε τιμή ← 2	Λάθος
xi.	Διάβασε ← 8	Λάθος
xii.	τιμή ← 0	Σωστή
xiii.	τιμή+5 ← 3	Λάθος
xiv.	τιμή = 8	Λάθος

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Απαντήσεις 1-115

11.

- i) $A \leftarrow (5 \cdot x^3 + 7 \cdot x^2 + 8) / (8 \cdot x - 6)$
 ii) $B \leftarrow x^2 - 32 \cdot y^3 - 7$
 iii) $\Gamma \leftarrow a^3(3 \cdot x + 2) + (x + 1) / (x^3 - 2)$

12.

- i. $14 \bmod 5 - 25 \bmod 8 = 4 - 1 = 3$
 ii. $3 * (3 \bmod 2) + 4 \operatorname{div} (5 \bmod 3) = 3 * 1 + 4 \operatorname{div} 2 = 3 + 2 = 5$
 iii. $13 \bmod (27 \operatorname{div} 4) = 13 \bmod 6 = 1$
 iv. $2^3 + 3 * (27 \bmod (25 \bmod 7)) = 8 + 3 * (27 \bmod 4) = 8 + 3 * 3 = 8 + 9 = 17$
 v. $13/2 - 3 \bmod 2 - 3 \operatorname{div} 2 = 6.5 - 1 - 1 = 4.5$
 vi. $13/4 + 2 * 5 \bmod 3 * 4 = 3.25 + 4 = 7.25$
 vii. $25 \bmod 22 \operatorname{div} 4 = 3 \operatorname{div} 4 = 0$ (οι πράξεις εκτελούνται από αριστερά προς τα δεξιά)
 viii. $((13 + 2) \operatorname{div} 2) / (7 - 4 + 1) = (15 \operatorname{div} 2) / 4 = 7/4 = 1.75$
 ix. $3 * (27 \bmod (23 \bmod 6)) = 3 * (27 \bmod 5) = 3 * 2 = 6$
 x. $1 \operatorname{div} 1 = 1$
 xi. $1 \bmod 1 = 0$

13.

- i. $(A > B)$ ή $(\Delta = 10)$ » αληθής ή ψευδής » αληθής
 ii. $(\Delta \geq B)$ και $(E \leftrightarrow \Gamma)$ » αληθής και αληθής » αληθής
 iii. **όχι** $(E \leq \Gamma)$ ή $(\Delta \leq \Gamma)$ » όχι ψευδής ή ψευδής » αληθής ή ψευδής » αληθής
 iv. **όχι** $((B \leq \Gamma)$ και $(\Delta < 2))$ » όχι (ψευδής και ψευδής) » όχι ψευδής » αληθής
 v. **όχι** (**όχι** $(B \leq E)$ ή **όχι** $(\Gamma \leq B)$) » όχι (όχι ψευδής ή όχι αληθής) » ψευδής
 vi. $((E \leq A)$ και $(E \geq \Gamma))$ και **όχι** $(\Gamma \geq A)$ » (αληθής και αληθής) και όχι ψευδής » αληθής
 vii. **όχι** (**όχι** $(A \geq 2)$ και $(\Gamma < 9)$) » όχι (όχι αληθής και αληθής) » αληθής

14.

i.	Το α ανήκει στο διάστημα [-5, 6)	$(\alpha \geq -5)$ και $(\alpha < 6)$
ii.	Το α είναι μικρότερο του 3 ή μεγαλύτερο του 15	$(\alpha < 3)$ ή $(\alpha > 15)$
iii.	Το α είναι ίσο με το β και το γ	$(\alpha = \beta)$ και $(\alpha = \gamma)$
iv.	Το α δεν έχει την τιμή 3	$\alpha \neq 3$
iv.	Το α δεν έχει την τιμή 3	όχι $(\alpha = 3)$
v.	Το α είναι μικρότερο του 2 ή το β είναι μεγαλύτερο του 78	$(\alpha < 2)$ ή $(\beta > 78)$
vi.	α και β αληθή και γ ψευδές	$(\alpha = \text{αληθής})$ και $(\beta = \text{αληθής})$ και $(\gamma = \text{ψευδής})$

15. 1-Δ, 2-Γ, 3-B, 4-A, 5-Δ, 6-B, 7-B, 8-Γ, 9-Γ, 10-Δ, 11-Δ, 12-Δ, 13-B, 14-A, 15-B, 16-Γ
 17-A, 18-B, 19-Δ, 20-B, 21-B

16. 1-στ, 2-γ, 3-ζ, 4-ε, 5-α

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Απαντήσεις 1-115

17.

A	B	A και B	A ή B
Αληθής	Αληθής	A	A
Αληθής	Ψευδής	Ψ	A
Ψευδής	Αληθής	Ψ	A
Ψευδής	Ψευδής	Ψ	Ψ

18.

A	B	όχι A	όχι (A ή B)
Αληθής	Αληθής	Ψ	Ψ
Αληθής	Ψευδής	Ψ	Ψ
Ψευδής	Αληθής	A	Ψ
Ψευδής	Ψευδής	A	A

19.

A	B	όχι (B και A)	όχι (A και B)
Αληθής	Αληθής	Ψ	Ψ
Αληθής	Ψευδής	A	A
Ψευδής	Αληθής	A	A
Ψευδής	Ψευδής	A	A

20.

A	B	όχι (A ή B)	όχι (A και (όχι (A ή B)))
Αληθής	Αληθής	Ψ	A
Αληθής	Ψευδής	Ψ	A
Ψευδής	Αληθής	Ψ	A
Ψευδής	Ψευδής	A	A

21.

ω	κ	λ	$\omega=2$ και ($\kappa \neq 0$ ή $\lambda > 0$)
2	-7	12	A
6	0	-9	Ψ
2	1	0	Ψ

22. 1-ΣΤ, 2-A, 3-Γ, 4-E

23. B

24. Γ

25. $\kappa=2$, B=17

26. α) A, β) Γ

27. Γ

28. A

29. A

30. α) B, β) Γ

31. Γ

32. α) Γ, β) A γ) A, δ) A

33. Γ

34. Γ

35. B

36. Δ

37. B

38. B

39. Γ

40. Γ

41. Δ

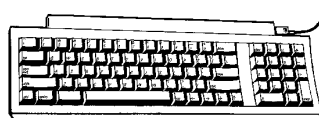
42. A

43. A

44. B

45. B

46. B



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Απαντήσεις 1-115

47.

Όσο ... επανάλαβε	Αρχή_επανάληψης ... Μέχρις_ότου	Για ... από ... μέχρι ... με_βήμα...
$\alpha \leftarrow 8$ Όσο $\alpha > 0$ επανάλαβε Εκτύπωσε α $\alpha \leftarrow \alpha - 1$ Τέλος_επανάληψης	$\alpha \leftarrow 8$ Αρχή_επανάληψης Εκτύπωσε α $\alpha \leftarrow \alpha - 1$ Μέχρις_ότου $\alpha \leq 0$	Για α από 8 μέχρι 1 με_βήμα -1 Εκτύπωσε α Τέλος_επανάληψης
$i \leftarrow -5$ $\alpha \leftarrow 1$ Όσο $\alpha < 0$ επανάλαβε $\alpha \leftarrow (\alpha + 2) * i$ $i \leftarrow i + 1$ Τέλος_επανάληψης	$i \leftarrow -5$ $\alpha \leftarrow 1$ Αρχή_επανάληψης $\alpha \leftarrow (\alpha + 2) * i$ $i \leftarrow i + 1$ Μέχρις_ότου $\alpha \geq 0$	$\alpha \leftarrow 1$ Για i από -5 μέχρι 0 $\alpha \leftarrow (\alpha + 2) * i$ Τέλος_επανάληψης

- | | | | |
|-------|-------|-------|-------|
| 48. A | 49. A | 50. B | 51. β |
| 52. γ | 53. α | 54. δ | 55. α |
| 56. α | 57. β | 58. β | 59. β |
| 60. β | 61. γ | 62. α | 63. β |
| 64. α | 65. α | 66. γ | 67. γ |
| 68. α | 69. δ | 70. α | 71. β |
| 72. α | 73. α | 74. α | |

75. **1.B** (ΟΝΟΜΑΤΑ), **1.Γ** (ΒΑΘΜΟΙ), **8.Γ** (ΒΑΘΜΟΙ), **14.B.A** φθίνουσα ταξινόμηση(MO), **2.A** (MO[1])
76. **1.Γ** (ΘΕΡΜΟΚΡΑΣΙΑ), **1.B** (ΠΟΛΗ), **11.Γ** (ΘΕΡΜΟΚΡΑΣΙΑ), **2.A** (ΠΟΛΗ[θέση_MAX_ j]), **8.Γ** (ΘΕΡΜΟΚΡΑΣΙΑ)
77. **1.B** (ΟΝΟΜΑΤΑ), **1.Γ** (ΠΟΝΤΟΙ), **5.Γ.A** (ΠΟΝΤΟΙ), **14.B.B** (ΑΘΡΟΙΣΜΑ, ΟΝΟΜΑΤΑ), **2.A** (ΟΝΟΜΑΤΑ[1], ΟΝΟΜΑΤΑ[2], ΟΝΟΜΑΤΑ[3])
78. **9.B.A** (ΟΝΟΜΑΤΑ), **2.A** (ΧΡΟΝΟΛΟΓΙΑ_ΓΕΝΝΗΣΗΣ[θέση], ΣΩΜΑ_ΚΑΤΑΤΑΞΗΣ[θέση]), **14.B.B** (ΣΩΜΑ_ΚΑΤΑΤΑΞΗΣ, ΟΝΟΜΑΤΑ, ΧΡΟΝΟΛΟΓΙΑ_ΓΕΝΝΗΣΗΣ)
79. **1.B** (ΒΑΘΜΟΣ), **11.B** (ΒΑΘΜΟΣ), **10.B** (ΒΑΘΜΟΣ), **2.A** (Max, Min, Max-Min), **6.B** (ΒΑΘΜΟΣ), **2.A** (MO)
80. **1.B** (ΠΩΛΗΤΗΣ), **1.Γ** (ΠΩΛΗΣΕΙΣ), **5.Γ.A** (ΠΩΛΗΣΕΙΣ), **11.B** (ΣΥΝΟΛΙΚΕΣ_ΠΩΛΗΣΕΙΣ) **2.A** (ΠΩΛΗΤΗΣ[θέση_MAX])
81. **1.B** (ΝΙΚΗΤΗΣ), **0.B** (ΑΡΙΘΜΟΣ_ΝΙΚΩΝ), **0.B** (ΝΟΥΜΕΡΑ), ΑΡΙΘΜΟΣ_ΝΙΚΩΝ[ΝΙΚΗΤΗΣ] ← ΑΡΙΘΜΟΣ_ΝΙΚΩΝ[ΝΙΚΗΤΗΣ]+1, **14.B.B** φθίνουσα ταξινόμηση(ΑΡΙΘΜΟΣ_ΝΙΚΩΝ, ΝΟΥΜΕΡΑ), **2.A** (ΝΟΥΜΕΡΑ[1], ΝΟΥΜΕΡΑ[2], ΝΟΥΜΕΡΑ[3])
82. **1.Γ** (ΠΟΝΤΟΙ), **5.Γ** (ΠΟΝΤΟΙ), **11.B** (ΣΥΝΟΛΙΚΟΙ_ΠΟΝΤΟΙ_ΠΑΙΚΤΩΝ), **2.A** (θέση_MAX), **2.B** (ΣΥΝΟΛΙΚΟΙ_ΠΟΝΤΟΙ_ΑΓΩΝΩΝ)
83. **1.B** (ΔΙΣΚΟΠΟΤΗΡΑ), **1.B** (ΠΡΟΣΩΠΑ), **9.B.A** (ΠΡΟΣΩΠΑ), **2.A** (ΔΙΣΚΟΠΟΤΗΡΑ[θέση])

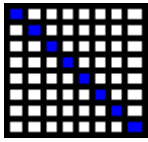
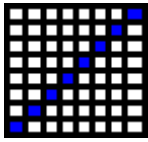
Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Απαντήσεις 1-115

<p>Άθροισμα 1^{ης} Στήλης.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N sum ← sum + A[i, 1] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα K στήλης. (1 ≤ K ≤ M)</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N sum ← sum + A[i, K] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα τελευταίας στήλης.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N sum ← sum + A[i, M] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα 1^{ης} γραμμής.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ M sum ← sum + A[1, i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα K γραμμής. (1 ≤ K ≤ N)</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ M sum ← sum + A[K, i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα τελευταίας γραμμής.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ M sum ← sum + A[N, i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα όλων των στοιχείων του πίνακα.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M sum ← sum + A[i, j] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα των στοιχείων της πρώτης και τρίτης γραμμής.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ M sum ← sum + A[1, i] + A[3, i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>
<p>Άθροισμα των στοιχείων της πρώτης και τρίτης στήλης.</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N sum ← sum + A[i, 1] + A[i, 3] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον
Απαντήσεις 1-115

102.

<p>Άθροισμα πρώτης διαγωνίου. (έστω πίνακας A με 8 γραμμές και 8 στήλες)</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8 sum ← sum + A[i,i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	
<p>Άθροισμα δεύτερης διαγωνίου. (έστω πίνακας A με 8 γραμμές και 8 στήλες)</p>	<p>sum ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8 sum ← sum + A[i,9-i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	

103. $N \geq 1$

104. α) 10 , 11 και 14
β) 10 , 1 και 3
γ) 6 , 1 και 2
δ) Υπάρχει λάθος

105. Α-4, Β-1, Γ-2, Δ-3

106. α)Σ β)Σ γ)Σ δ)Σ ε)Σ στ)Σ

107.

ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: j

ΠΡΑΓΜΑΤΙΚΕΣ: Sum , i

ΑΡΧΗ

Sum ← 0

j ← 1

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5 ΜΕ ΒΗΜΑ 0.5

Sum ← Sum + i

j ← j + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ Sum + j ! 37

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

108. Δημιουργούμε έναν πίνακα τιμών για να παρακολουθήσουμε τις αλλαγές που γίνονται στον πίνακα A. {Τα σκούρα κελιά τονίζουν τις αντιμεταθέσεις}

i	j	1 ^η θέση	2 ^η θέση	3 ^η θέση	4 ^η θέση	5 ^η θέση
2	5	35	66	12	55	48
2	4	35	66	55	12	48
2	3	35	66	55	12	48
2	2	66	35	55	12	48
Άρα ο πίνακας A είναι:		66	35	55	12	48
3	5	66	35	55	48	12
3	4	66	35	55	48	12
3	3	66	55	35	48	12
Άρα ο πίνακας A είναι:		66	55	35	48	12
4	5	66	55	35	48	12
4	4	66	55	48	35	12
Άρα ο πίνακας A είναι:		66	55	48	35	12
5	5	66	55	48	35	12

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Απαντήσεις 1-115

Ο τελικός πίνακας μετά τις πέντε αντιμεταθέσεις θα είναι:

1 ^η θέση	2 ^η θέση	3 ^η θέση	4 ^η θέση	5 ^η θέση
66	55	48	35	12

Όπως διαπιστώνουμε τα στοιχεία του πίνακα έχουν ταξινομηθεί με φθίνουσα σειρά.

109. α

110. β

111. α

112. γ

113. γ

114. γ

115. α,δ

Λύση του λαβύρινθου

Αυτή είναι μια ενδεικτική λύση του προβλήματος:

AB-AC-BC-BD-CD-CE-CL-BL-BF-BG-CG-CH-DH-DJ-CJ-CK-BK-BB

Ο γρίφος δημιουργήθηκε από τον Robert Abbott.

┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌ ┌

Ο υπολογιστής μπορεί να εκτελεί **τρεις** διαφορετικές λειτουργίες: Πρόσθεση, έλεγχο και μεταφορά. Αναλυτικότερα:

Πρόσθεση: Ο υπολογιστής εκτελεί μέσω των ηλεκτρονικών κυκλωμάτων του πρόσθεση δύο αριθμών. Αυτό σημαίνει ότι μπορεί να εκτελεί κάθε πράξη, δεδομένου ότι η αφαίρεση είναι η πρόσθεση του αντιθέτου, πολλαπλασιασμός συνεχείς προσθέσεις, ενώ η διαίρεση συνεχείς αφαιρέσεις. Με τον ίδιο συλλογισμό είναι φανερό ότι και κάθε συνάρτηση αναλυόμενη στα επιμέρους βασικά βήμα καταλήγει να είναι μία σειρά των βασικών πράξεων, επομένως να μπορεί να εκτελεστεί από τον υπολογιστή.

Έλεγχος: Ο υπολογιστής μπορεί να ελέγξει την συνθήκη ανάμεσα σε δύο ποσότητες και να αποφανθεί αν η συνθήκη είναι αληθής ή ψευδής.

Μεταφορά: Ο υπολογιστής εκτελεί μεταφορά δεδομένων. Αυτή η μεταφορά γίνεται είτε μεταξύ της κεντρικής μονάδας με τις περιφερειακές συσκευές, είτε εντός της κεντρικής μονάδας (κεντρική μνήμη). Αναλυτικότερα, το πάτημα ενός πλήκτρου του πληκτρολογίου προκαλεί την μεταφορά του χαρακτήρα που πατήθηκε, σε κωδικοποιημένη μορφή, προς την κεντρική μνήμη του υπολογιστή. Παρόμοια η εμφάνιση αποτελεσμάτων στην οθόνη του υπολογιστή ή στο χαρτί, αποτελεί την αντίστροφη μεταφορά, από την μνήμη προς την περιφερειακή μονάδα αποκωδικοποιούμενο. Τέλος κατά την εκτέλεση πράξεων στον υπολογιστή δημιουργείται η ανάγκη αποθήκευσης του αποτελέσματος. Αυτό αποτελεί μια εσωτερική μεταφορά.



σελ. 92	A. ΜΕΤΑΒΛΗΤΗ x	B. ΠΙΝΑΚΑΣ A[N]	Γ. ΠΙΝΑΚΑΣ A[N,M]	ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ / ΠΑΡΑΔΕΙΓΜΑΤΑ / ΣΧΟΛΙΑ											
0. ΑΡΧΙΚΟΠΟΙΗΣΗ ΜΕΤΑΒΛΗΤΩΝ	x ← 0 ή x ← 1 κ.τ.λ.	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N A[i] ← 0 ! ή A[i] ← 1 κ.τ.λ. ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M A[i,j] ← 0 ! ή A[i,j] ← 1 κ.τ.λ. ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	<u>ΠΑΡΑΤΗΡΗΣΗ</u> Κατά την εκτέλεση ενός προγράμματος οι μεταβλητές δεν έχουν μια συγκεκριμένη αρχική τιμή εκκίνησης αλλά μια τυχαία τιμή του τύπου δεδομένων που υποστηρίζουν. Για το λόγο αυτό όταν τις χρησιμοποιούμε σαν αθροιστές ή μετρητές πρέπει πάντα να τις δίνουμε μια αρχική τιμή.											
1. ΕΙΣΟΔΟΣ ΔΕΔΟΜΕΝΩΝ	ΔΙΑΒΑΣΕ x	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΔΙΑΒΑΣΕ A[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M ΔΙΑΒΑΣΕ A[i,j] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	Π.χ. Να δοθεί πρόγραμμα που να <u>διαβάζει</u> από τον χρήστη τα δεδομένα... ή να <u>διαβάζει</u> τον πίνακα A... ή να <u>εισάγει</u> στον πίνακα τα δεδομένα...κ.τ.λ.											
2. ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ	ΓΡΑΨΕ x	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΡΑΨΕ A[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M ΓΡΑΨΕ A[i,j] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	Π.χ. Να <u>εμφανίζει</u> τον πίνακα A... ή να <u>εκτυπώνει</u> τα τελικά αποτελέσματα ... ή να <u>εμφανίζονται</u> οι τιμές των μεταβλητών... κ.τ.λ.											
3. ΣΥΝΟΛΙΚΟ ΑΘΡΟΙΣΜΑ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ	-	σύνολο ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N σύνολο ← σύνολο + A[i] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	σύνολο ← 0 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M σύνολο ← σύνολο + A[i,j] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	<u>ΠΑΡΑΤΗΡΗΣΗ</u> Συνήθως υπολογίζουμε το συνολικό άθροισμα για να βρούμε στη συνέχεια τον μέσο όρο.	ΠΑΡΑΔΕΙΓΜΑ <table border="1" style="margin: 10px auto;"> <tr><td>3</td><td>0</td><td>0</td><td>5</td><td>2</td></tr> <tr><td>4</td><td>2</td><td>5</td><td>9</td><td>2</td></tr> </table> Συνολικό Άθροισμα = 22	3	0	0	5	2	4	2	5	9	2
3	0	0	5	2											
4	2	5	9	2											

- Ο συμβολισμός π.χ. **2.B** υποδεικνύει τον κώδικα που βρίσκεται στη δεύτερη γραμμή και την τρίτη στήλη(εμφάνιση δεδομένων 2D πίνακα).
- Το σύμβολο – σημαίνει ότι δεν υφίσταται λογική λύση. Ενώ το ~ φανερώνει ότι η λύση παραλείπεται λόγω της πολυπλοκότητάς της.
- Το σύμβολο & υποδηλώνει ότι ο ακόλουθος κώδικας βρίσκεται στην ίδια γραμμή με τον προηγούμενο.

σελ. 93	Α. ΜΕΤΑΒΛΗΤΗ x	Β. ΠΙΝΑΚΑΣ A[N]	Γ. ΠΙΝΑΚΑΣ A[N,M]	ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ / ΠΑΡΑΔΕΙΓΜΑΤΑ / ΣΧΟΛΙΑ																																							
5. ΑΘΡΟΙΣΜΑ ΑΝΑ ΓΡΑΜΜΗ ΚΑΙ ΑΝΑ ΣΤΗΛΗ	-	3	<p><i>1^{ος} τρόπος</i></p> <p>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N $B[i] \leftarrow 0$ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ M $\Gamma[i] \leftarrow 0$ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M $B[i] \leftarrow B[i] + A[i,j]$! ανά γραμμή $\Gamma[j] \leftarrow \Gamma[j] + A[i,j]$! ανά στήλη ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>Π.χ. Να βρεθεί το <u>σύνολο</u> των πόντων που σημείωσε <u>κάθε</u> παίκτης(στήλη) στους αγώνες του πρωταθλήματος(γραμμές)....</p> <p>ή</p> <p>Αντίστροφα το <u>σύνολο</u> των πόντων που σημειώθηκε σ' <u>ένα</u> παιχνίδι(γραμμή) απ' <u>όλους</u> τους παίκτες (στήλες)... κ.τ.λ.</p>	<p>ΠΑΡΑΔΕΙΓΜΑΤΑ</p> <p>↓ Αθροισμα ανά στήλη</p> <table border="1" data-bbox="1742 391 1966 494"> <tr><td>3</td><td>0</td><td>0</td><td>5</td><td>2</td></tr> <tr><td>3</td><td>0</td><td>1</td><td>0</td><td>8</td></tr> <tr><td>2</td><td>0</td><td>2</td><td>9</td><td>1</td></tr> </table> <p>Γ <table border="1" data-bbox="1742 518 1966 558"> <tr><td>8</td><td></td><td></td><td></td><td></td></tr> </table></p> <hr/> <p>Αθροισμα ανά γραμμή</p> <table border="1" data-bbox="1709 662 1933 766"> <tr><td>2</td><td>0</td><td>0</td><td>5</td><td>2</td></tr> <tr><td>7</td><td>0</td><td>1</td><td>0</td><td>8</td></tr> <tr><td>6</td><td>0</td><td>2</td><td>9</td><td>1</td></tr> </table> <p>B <table border="1" data-bbox="1966 646 2016 766"> <tr><td>9</td></tr> <tr><td></td></tr> <tr><td></td></tr> </table></p>	3	0	0	5	2	3	0	1	0	8	2	0	2	9	1	8					2	0	0	5	2	7	0	1	0	8	6	0	2	9	1	9		
			3	0	0	5	2																																				
3	0	1	0	8																																							
2	0	2	9	1																																							
8																																											
2	0	0	5	2																																							
7	0	1	0	8																																							
6	0	2	9	1																																							
9																																											
<p><i>2^{ος} τρόπος</i></p> <p>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N $\beta \leftarrow 0$ ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M $\beta \leftarrow \beta + A[i,j]$! ανά γραμμή } 5.Γ.Α ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΓΡΑΨΕ β ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p> <p>ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M $\gamma \leftarrow 0$ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N $\gamma \leftarrow \gamma + A[i,j]$! ανά στήλη } 5.Γ.Β ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΓΡΑΨΕ γ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p><u>ΠΑΡΑΤΗΡΗΣΗ</u> Η δεύτερη υλοποίηση του αλγορίθμου είναι πιο απλή από την πρώτη όμως δεν γίνεται καταχώρηση όλων των αθροισμάτων, παρά μόνο των τελευταίων. Κατά συνέπεια πολλά αποτελέσματα χάνονται. Αν θέλουμε απλώς να εμφανίσουμε τα αθροίσματα χωρίς να τα αποθηκεύσουμε η δεύτερη λύση είναι πιο ιδανική.</p>	<p>Χωρίς πίνακες, με δύο μεταβλητές μόνο. <u>ΠΡΟΣΟΧΗ</u> όμως γιατί χάνονται οι προηγούμενες τιμές.</p> <p>↓</p> <table border="1" data-bbox="1686 1029 1910 1101"> <tr><td>2</td><td>0</td><td>0</td><td>5</td><td>2</td></tr> <tr><td>3</td><td>0</td><td>1</td><td>0</td><td>8</td></tr> </table> <p>β <table border="1" data-bbox="1955 1029 2004 1069"> <tr><td>9</td></tr> </table></p> <p>5 γ</p>	2	0	0	5	2	3	0	1	0	8	9																														
2	0	0	5	2																																							
3	0	1	0	8																																							
9																																											

σελ. 95	Α. ΜΕΤΑΒΛΗΤΗ x	Β. ΠΙΝΑΚΑΣ A[N]	Γ. ΠΙΝΑΚΑΣ A[N,M]	ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ / ΠΑΡΑΔΕΙΓΜΑΤΑ / ΣΧΟΛΙΑ
<p>9. ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ</p> <p>(Σελ. 28 όλες οι υποκατη- γορίες)</p>	-	<p>ΓΡΑΨΕ 'Δώστε το & ζητούμενο:' ΔΙΑΒΑΣΕ ζητούμενο</p> <p>σημαία ← ΨΕΥΔΗΣ i ← 1 ΟΣΟ (i ≤ N) ΚΑΙ (σημαία = & ΨΕΥΔΗΣ) ΕΠΑΝΑΛΑΒΕ ΑΝ A[i] = ζητούμενο ΤΟΤΕ σημαία ← ΑΛΗΘΗΣ θέση ← i ΑΛΛΙΩΣ i ← i + 1 ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>Σελ. 32</p>	<p>Π.χ. Να γίνει αναζήτηση ενός στοιχείου στον πίνακα A... ή να εμφανισθούν τα ονόματα των μαθητών που <u>ισοψηφούν στην πρώτη θέση</u> (προύποθέτει την ταξινόμηση του πίνακα).. κ.τ.λ.</p> <p>ΠΑΡΑΤΗΡΗΣΗ Αν δε θυμόμαστε τον αλγόριθμο της σειριακής αναζήτησης μπορούμε να κάνουμε την αναζήτηση με ΓΙΑ.. ΑΠΟ...ΜΕΧΡΙ, θα έχουμε όμως περισσότερες επαναλήψεις του αλγορίθμου.</p> <div style="border: 1px dashed black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>βρέθηκε ← ΨΕΥΔΗΣ ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΑΝ A[i] = ζητούμενο ΤΟΤΕ βρέθηκε ← ΑΛΗΘΗΣ θέση ← i ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p> </div>
<p>10. ΕΥΡΕΣΗ ΕΛΑΧΙΣΤΟΥ</p>	-	<p>MIN ← A[1] Θέση_MIN ← 1 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N ΑΝ A[i] < MIN ΤΟΤΕ MIN ← A[i] Θέση_MIN ← i ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>MIN ← A[1,1] Θέση_MIN_i ← 1 Θέση_MIN_j ← 1 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M ΑΝ A[i,j] < MIN ΤΟΤΕ MIN ← A[i,j] Θέση_MIN_i ← i Θέση_MIN_j ← j ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>Π.χ. Να βρεθεί η <u>ελάχιστη</u> θερμοκρασία... ή να βρεθεί η <u>κατώτερη</u> τιμή... ή να βρεθεί το μάθημα στο οποίο κάποιος μαθητής πήρε την <u>χαμηλότερη</u> βαθμολογία... ή να βρεθεί η <u>μέγιστη απόκλιση</u> (MAX-MIN)... κ.τ.λ.</p> <p>ΠΑΡΑΤΗΡΗΣΗ Στη θέση του MIN μπορούμε να βάλουμε και έναν πολύ μικρό αριθμό. Αν για παράδειγμα ψάχνουμε να βρούμε την ελάχιστη τιμή σ' έναν πίνακα με βαθμολογίες από 0 έως 20 δεν είναι απαραίτητο να δώσουμε στο MIN την τιμή A[1], μπορούμε να δώσουμε και την τιμή π.χ. -1 διότι η τιμή αυτή θα αλλάξει σίγουρα μετέπειτα. Κάτι παρόμοιο μπορούμε να κάνουμε και για το MAX.</p>

σελ. 96	Α. ΜΕΤΑΒΛΗΤΗ x	Β. ΠΙΝΑΚΑΣ A[N]	Γ. ΠΙΝΑΚΑΣ A[N,M]	ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ / ΠΑΡΑΔΕΙΓΜΑΤΑ / ΣΧΟΛΙΑ																																													
<p>11. ΕΥΡΕΣΗ ΜΕΓΙΣΤΟΥ</p>	-	<p>MAX ← A[1] Θέση_MAX ← 1 ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N ΑΝ A[i] > MAX ΤΟΤΕ MAX ← A[i] Θέση_MAX ← i ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>MAX ← A[1,1] Θέση_MAX_i ← 1 Θέση_MAX_j ← 1 ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M ΑΝ A[i,j] > MAX ΤΟΤΕ MAX ← A[i,j] Θέση_MAX_i ← i Θέση_MAX_j ← j ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>ΠΑΡΑΔΕΙΓΜΑ</p> <p>Να βρεθεί το μέγιστο στον ακόλουθο πίνακα:</p> <table border="1" data-bbox="1368 392 2009 427"> <tr> <td>8</td> <td>9</td> <td>-1</td> <td>6</td> <td>6</td> </tr> </table> <p>Συγκρίνουμε αυτό με τα υπόλοιπα στοιχεία</p> <table border="1" data-bbox="1368 517 2009 552"> <tr> <td>8</td> <td>9</td> <td>1</td> <td>6</td> <td>6</td> </tr> </table> <p>1 2 3 4 5</p> <p>οι θέσεις του πίνακα (δείκτης i) Τελικά το μέγιστο=9 και βρέθηκε στη θέση=2.</p>		8	9	-1	6	6	8	9	1	6	6																																		
8	9	-1	6	6																																													
8	9	1	6	6																																													
<p>12. ΜΕΓΙΣΤΟ / ΕΛΑΧΙΣΤΟ ΑΝΑ ΓΡΑΜΜΗ</p> <p>(κατά τον ίδιο τρόπο υπολογίζουμε και το ελάχιστο ανά γραμμή. Η μόνη αλλαγή που κάνουμε είναι η μετατροπή του > σε <)</p>	-	11/10	<p>ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N MAX ← A[i,1] ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ M ΑΝ A[i,j] > MAX ΤΟΤΕ MAX ← A[i,j] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ B[i] ← MAX ! ή μόνο ΓΡΑΨΕ MAX ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p> <p>Το μέγιστο ανά γραμμή έχει κωδικό: 12.Γ.Α, ενώ το ελάχιστο ανά γραμμή: 12.Γ.Β.</p>	<p>Π.χ. Να βρεθεί η μεγιστη/ελάχιστη θερμοκρασία για το μήνα Ιανουάριο στην Αθήνα (στις στήλες έχουμε τις 30 ημέρες και στις γραμμές τις πόλεις)...</p> <p>ή</p> <p>Π.χ. Να βρεθεί ο μεγαλύτερος/μικρότερος βαθμός του φοιτητή Α (στις στήλες έχουμε τα μαθήματα και στις γραμμές τους φοιτητές) κ.τ.λ.</p>	<p>ΠΑΡΑΔΕΙΓΜΑΤΑ</p> <p>Μέγιστο ανά γραμμή με καταχώρηση τιμών σε πίνακα.</p> <table border="1" data-bbox="1688 874 2000 1007"> <tr> <td>2</td> <td>1</td> <td>0</td> <td>5</td> <td>2</td> <td rowspan="3">B</td> </tr> <tr> <td>7</td> <td>4</td> <td>6</td> <td>5</td> <td>8</td> </tr> <tr> <td>6</td> <td>3</td> <td>2</td> <td>9</td> <td>1</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>5</td> </tr> </table> <table border="1" data-bbox="1688 1082 2000 1182"> <tr> <td>2</td> <td>9</td> <td>7</td> <td>5</td> <td>8</td> <td rowspan="3">MAX</td> </tr> <tr> <td>7</td> <td>5</td> <td>6</td> <td>5</td> <td>8</td> </tr> <tr> <td>6</td> <td>5</td> <td>2</td> <td>9</td> <td>1</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>9</td> </tr> </table> <p>Μία μεταβλητή μόνο. Χάνεται το προηγούμενο μέγιστο.</p>	2	1	0	5	2	B	7	4	6	5	8	6	3	2	9	1						5	2	9	7	5	8	MAX	7	5	6	5	8	6	5	2	9	1						9
2	1	0	5	2	B																																												
7	4	6	5	8																																													
6	3	2	9	1																																													
					5																																												
2	9	7	5	8	MAX																																												
7	5	6	5	8																																													
6	5	2	9	1																																													
					9																																												

σελ. 97	A. ΜΕΤΑΒΛΗΤΗ x	B. ΠΙΝΑΚΑΣ A[N]	Γ. ΠΙΝΑΚΑΣ A[N,M]	ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ / ΠΑΡΑΔΕΙΓΜΑΤΑ / ΣΧΟΛΙΑ																																				
<p>13. ΜΕΓΙΣΤΟ / ΕΛΑΧΙΣΤΟ ΑΝΑ ΣΤΗΛΗ</p> <p>(κατά τον ίδιο τρόπο υπολογίζουμε και το ελάχιστο ανά στήλη. Η μόνη αλλαγή που κάνουμε είναι η μετατροπή του > σε <)</p>	-	11/10	<p>ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M MAX ← A[1,j] ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N ΑΝ A[i,j] > MAX ΤΟΤΕ MAX ← A[i,j] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ Γ[j] ← MAX ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p> <p>Το μέγιστο ανά στήλη έχει κωδικό: 13.Γ.Α, ενώ το ελάχιστο ανά στήλη: 13.Γ.Β.</p>	<p>Αν θέλουμε να εμφανίσουμε τα μέγιστα, χωρίς να τα κατάρχωρήσουμε σε πίνακα, μπορούμε να χρησιμοποιήσουμε και τον παρακάτω κώδικα:</p> <p>ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ M MAX ← A[1,j] ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N ΑΝ A[i,j] > MAX ΤΟΤΕ MAX ← A[i,j] ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΓΡΑΨΕ MAX ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	<p>ΠΑΡΑΔΕΙΓΜΑΤΑ Αποθήκευση σε πίνακα.</p> <table border="1" style="margin-left: 20px;"> <tr><td>8</td><td>1</td><td>6</td><td>5</td><td>2</td></tr> <tr><td>3</td><td>5</td><td>1</td><td>4</td><td>8</td></tr> <tr><td>2</td><td>2</td><td>0</td><td>1</td><td>1</td></tr> </table> <p>Γ</p> <table border="1" style="margin-left: 20px;"> <tr><td>8</td><td></td><td></td><td></td><td></td></tr> </table> <p>↓</p> <table border="1" style="margin-left: 20px;"> <tr><td>8</td><td>1</td><td>3</td><td>5</td><td>2</td></tr> <tr><td>3</td><td>5</td><td>1</td><td>4</td><td>8</td></tr> <tr><td>2</td><td>2</td><td>0</td><td>1</td><td>1</td></tr> </table> <p>8 MAX Μία μεταβλητή μόνο</p>	8	1	6	5	2	3	5	1	4	8	2	2	0	1	1	8					8	1	3	5	2	3	5	1	4	8	2	2	0	1	1
8	1	6	5	2																																				
3	5	1	4	8																																				
2	2	0	1	1																																				
8																																								
8	1	3	5	2																																				
3	5	1	4	8																																				
2	2	0	1	1																																				
<p>14. ΤΑΞΙΝΟΜΗΣΗ</p> <p>(Σελ. 27 όλες οι υποκατηγορίες)</p>	-	<p>ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ & ΒΗΜΑ -1 ! αύξουσα ταξινόμηση ΑΝ A[j-1] > A[j] ΤΟΤΕ βοηθητική ← A[j-1] A[j-1] ← A[j] A[j] ← βοηθητική ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</p>	~	<p>Π.χ. Να <u>ταξινομηθεί</u> ο πίνακας A... ή να εμφανισθούν σε μία <u>σειρά</u> τα τελικά αποτελέσματα... ή να βρεθεί ή <u>μεγιστη απόκλιση</u>.. ή να βρεθεί ο <u>1^{ος}, 2^{ος}, τελευταίος</u>... κ.τ.λ.</p> <p>Για φθίνουσα ταξινόμηση αλλάζουμε το: A[j-1] > A[j] σε A[j-1] < A[j]. Στον αλγόριθμο μπορούμε να χρησιμοποιήσουμε και την εντολή: αντιμετάθεσε A[j], A[j-1]</p>	<p>ΠΑΡΑΔΕΙΓΜΑ Ο αρχικός πίνακας:</p> <table border="1" style="margin-left: 20px;"> <tr><td>3</td><td>7</td><td>-4</td><td>0</td></tr> </table> <p>Ο ίδιος πίνακας μετά την χρήση του αλγορίθμου: 14.B</p> <table border="1" style="margin-left: 20px;"> <tr><td>-4</td><td>0</td><td>3</td><td>7</td></tr> </table>	3	7	-4	0	-4	0	3	7																											
3	7	-4	0																																					
-4	0	3	7																																					

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Εγκυκλοπαίδεια Πληροφορικής και Τεχνολογίας Υπολογιστών, Νέες Τεχνολογίες.
- Εφαρμογές Πληροφορικής Ενιαίου Λυκείου, ΟΕΔΒ.
- Μανώλης Λουκάκης, Δομές Δεδομένων, Ζυγός.
- Brian P. Flannery, Numerical recipes in C++, Cambridge University Press.
- D. E. Knuth, The Art of Computer Programming, Addison-Wesley.
- Niklaus Wirth, Αλγόριθμοι και Δομές Δεδομένων, Κλειδάριθμος.
- Ph. Breton, Ιστορία της Πληροφορικής, Δίαυλος.
- R. Shackelford, Introduction to Computing and Algorithms, Addison-Wesley.
- T. Cormen, Introduction to Algorithms, MIT.
- Ιωάννης Ζωζάς, Σημειώσεις Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον.

- <http://theory.lcs.mit.edu/groups/algorithms.html>
- <http://users.kor.sch.gr/ptsiotakis>
- <http://users.otenet.gr/~spin1/glossomatheia/>
- <http://www.eecs.umich.edu/courses/index.html>
- <http://www.e-yliko.sch.gr/>
- <http://www.aepp.gr/>

- Τα εικονίδια του βιβλίου φιλοτέχνησε η Αγγελική Χρονοπούλου.

- Ευχαριστώ τους μαθητές μου για τις παρατηρήσεις και τις διορθώσεις τους.

ISBN: 960-631-144-9
Θεσσαλονίκη 2009 – 3^η Έκδοση