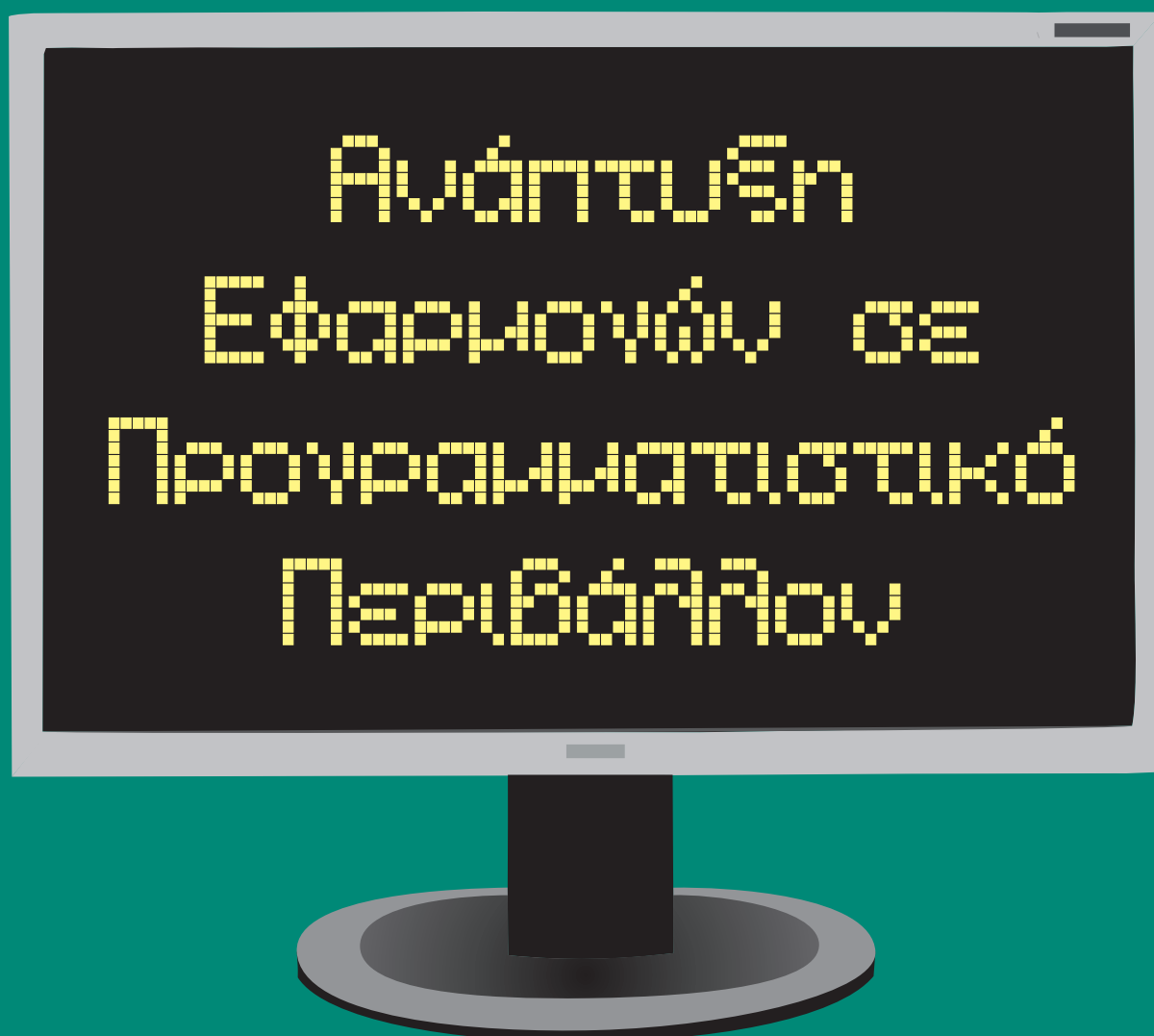




Τετράδιο μαθητή

Γ' ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ

Ομάδας Προσανατολισμού Θετικών Σπουδών και Σπουδών Οικονομίας & Πληροφορικής



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

ΣΤΟΙΧΕΙΑ ΑΡΧΙΚΗΣ ΕΚΔΟΣΗΣ

Ομάδα Συγγραφής	ΑΘΗΝΑ ΒΑΚΑΛΗ, Λέκτωρ Πληροφορικής ΑΠΘ ΗΛΙΑΣ ΓΙΑΝΝΟΠΟΥΛΟΣ, Μηχανικός Πληροφορικής ΝΕΣΤΩΡ ΙΩΑΝΝΙΔΗΣ, Επίκουρος Καθηγητής Τμήματος Πληροφορικής ΤΕΙ Αθήνας ΧΡΗΣΤΟΣ ΚΟΙΛΙΑΣ, Επίκουρος Καθηγητής Τμήματος Πληροφορικής ΤΕΙ Αθήνας ΚΩΝ/ΝΟΣ ΜΑΛΑΜΑΣ, M.Sc. Πληροφορικής, Σύμβουλος Επιχειρήσεων ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΠΟΥΛΟΣ, Αναπληρωτής Καθηγητής Τμήματος Πληροφορικής ΑΠΘ ΠΑΝΑΓΙΩΤΗΣ ΠΟΛΙΤΗΣ, Δρ. Διδακτικής Πληροφορικής, Καθηγητής ΠΕ 19
Υπεύθυνος για το Παιδαγωγικό Ινστιτούτο	ΓΕΩΡΓΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ, Σύμβουλος Π.Ι. (κατά τη συγγραφή)
Επιτροπή Αξιολόγησης	ΚΩΝ/ΝΟΣ ΓΙΑΛΟΥΡΗΣ, Καθηγητής ΠΕ 19 ΕΥΑΓΓΕΛΙΑ ΓΟΥΛΗ, Καθηγήτρια ΠΕ 19 ΔΗΜΗΤΡΙΟΣ ΔΕΣΠΟΤΗΣ, Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς ΚΩΝ/ΝΟΣ ΖΑΧΑΡΗΣ, ΠΛΗΝΕΤ Καρδίτσας ΑΛΕΞΑΝΔΡΟΣ ΣΙΔΕΡΙΔΗΣ, Καθηγητής Γεωπονικού Πανεπιστημίου Αθηνών
Εικονογράφηση	ΑΓΓΕΛΟΣ ΑΓΙΟΣΤΡΑΤΙΤΗΣ
Ηλεκτρονική σελιδοποίηση	ANNA ΑΡΓΥΡΟΠΟΥΛΟΥ - ΧΡΗΣΤΟΣ ΠΙΓΚΑΣ
Εξώφυλλο	ΣΠΥΡΟΣ ΣΙΑΚΑΣ - ΝΙΚΟΣ ΓΕΩΡΓΙΟΥ
Επιμέλεια	ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, Στουρνάρη 49Α, 106 82, Αθήνα, Τηλ. 38.45.594
Φορέας	ΕΛΛΗΝΙΚΗ ΕΤΑΙΡΙΑ ΕΠΙΣΤΗΜΟΝΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ (ΕΠΥ), Μαυρομichάλη 16, Αθήνα, τηλ.: 3645274, e.mail: epy@epy.gr
Συντονιστές έργου	ΣΠ. ΜΠΑΚΟΓΙΑΝΝΗΣ, πρόεδρος Δ.Σ. ΒΑΣ. ΜΙΧΑΛΑΚΟΠΟΥΛΟΣ, μέλος Δ.Σ.

Ενέργεια 1.1.α: «Προγράμματα Βιβλία»

Επιστημονικός Υπεύθυνος Ενέργειας Θεόδωρος Γ. Εξαρχάκος, Καθηγητής του Πανεπιστημίου
Αθηνών, Πρόεδρος του Παιδαγωγικού Ινστιτούτου

Έργο Νο 15: «Αναμόρφωση / εκ νέου σύνταξη και συγγραφή Προγραμμάτων Σπουδών και Σχολικών Βιβλίων για το Ενιαίο Λύκειο»

Επιστημονικός Υπεύθυνος Έργου Γιάννης Σαλβαράς, Επίκουρος Καθηγητής
του Δημοκρίτειου Πανεπιστημίου Θράκης

Καλλιτεχνικός Υπεύθυνος Έργου Σπύρος Ι. Παπασπύρου,
Καθηγητής Εφαρμογών του ΤΕΙ Ηπείρου

ΣΤΟΙΧΕΙΑ ΕΠΑΝΕΚΔΟΣΗΣ

Η επανέκδοση του παρόντος βιβλίου πραγματοποιήθηκε από το Ινστιτούτο Τεχνολογίας Υπολογιστών & Εκδόσεων «Διόφαντος» μέσω ψηφιακής μακέτας, η οποία δημιουργήθηκε με χρηματοδότηση από το ΕΣΠΑ / ΕΠ «Εκπαίδευση & Διά Βίου Μάθηση» / Πράξη «ΣΤΗΡΙΖΩ».



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
ΠΡΟΤΥΠΟ ΠΑΛΕΑΣ & ΠΡΟΚΕΚΜΕΝΩΝ, ΠΟΛΙΤΕΙΣ & ΑΝΑΠΤΕΧΝΩ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Οι διορθώσεις πραγματοποιήθηκαν κατόπιν έγκρισης του Δ.Σ. του Ινστιτούτου Εκπαιδευτικής Πολιτικής

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Α. ΒΑΚΑΛΗ, Η. ΓΙΑΝΝΟΠΟΥΛΟΣ, Ν. ΙΩΑΝΝΙΔΗΣ, Χ. ΚΟΙΛΙΑΣ,
Κ. ΜΑΛΑΜΑΣ, Ι. ΜΑΝΩΛΟΠΟΥΛΟΣ, Π. ΠΟΛΙΤΗΣ

Η συγγραφή και η επιστημονική επιμέλεια του βιβλίου πραγματοποιήθηκε
υπό την αιγίδα του Παιδαγωγικού Ινστιτούτου

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Τετράδιο Μαθητή

Γ' Γενικού Λυκείου

**(Ομάδας Προσανατολισμού Θετικών Σπουδών
και Σπουδών Οικονομίας & Πληροφορικής)**

Περιεχόμενα

Λίγα λόγια για το μαθητή	7
1. Ανάλυση Προβλήματος.....	9
2. Βασικές Έννοιες Αλγορίθμων.....	15
3. Δομές Δεδομένων και Αλγόριθμοι.....	27
4. Τεχνικές Σχεδίασης Αλγορίθμων	35
5. Ανάλυση Αλγορίθμων	43
6. Εισαγωγή στον Προγραμματισμό.....	53
7. Βασικές Έννοιες Προγραμματισμού	57
8. Επιλογή και Επανάληψη	67
9. Πίνακες	81
10. Υποπρογράμματα	93
Απαντήσεις στα τεστ αυτοαξιολόγησης	107

Λίγα λόγια για το μαθητή

Αγαπητέ μαθητή,

Στα χέρια σου κρατάς το Τετράδιο Μαθητή, ένα συμπληρωματικό, αλλά ταυτόχρονα και λειτουργικά απαραίτητο σύγγραμμα για τη διδασκαλία του μαθήματος “Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον”. Σκοπός του μαθήματος δεν είναι να σε διδάξει και να εμβαθύνεις σε κάποια συγκεκριμένη γλώσσα προγραμματισμού. Η έμφαση και η προσπάθεια γίνεται στο να μπορέσεις:

- να αναπτύξεις αναλυτική σκέψη και συνθετική ικανότητα,
- να καλλιεργήσεις αυστηρότητα στη διατύπωση,
- να αναπτύξεις δημιουργικότητα και φαντασία στο σχεδιασμό,
- να αποκτήσεις ικανότητες μεθοδολογικού χαρακτήρα,
- να αναπτύξεις δεξιότητες αλγοριθμικής προσέγγισης,
- να μπορείς να επιλύεις προβλήματα και να υλοποιείς απλά τη λύση τους με χρήση βασικών γνώσεων προγραμματιστικού περιβάλλοντος.

Η θεωρητική πλευρά του μαθήματος καλύπτεται από το βιβλίο μαθητή. Σκοπός του τετραδίου αυτού είναι να σου προσφέρει μέσα από τα έτοιμα παραδείγματα που σου παρουσιάζει, αλλά και τις δραστηριότητες που σου προτείνει, μια πρακτική προσέγγιση των γνώσεων που παραθέτονται στο βιβλίο μαθητή. Με μια σειρά από τρόπους, μεθόδους και απλές τεχνικές χρήσης διαφόρων προγραμματιστικών περιβαλλόντων, σε βοηθάει στη μοντελοποίηση και επίλυση πραγματικών ή ιδεατών προβλημάτων.

Τα παραδείγματα και οι δραστηριότητες που προτείνονται περιγράφονται είτε σε μια υποθετική γλώσσα προγραμματισμού, τη ΓΛΩΣΣΑ, είτε σε πραγματικές γλώσσες προγραμματισμού, την QuickBasic, την Turbo Pascal, τη Visual Basic και την Delphi. Κάποιες από τις γλώσσες αυτές είναι αυτές που θα χρησιμοποιήσεις στο εργαστήριο Πληροφορικής του σχολείου σου. Η αναφορά των ρεπερτορίων εντολών και των τεχνικών καθεμιάς από τις γλώσσες αυτές δεν γίνεται διεξοδικά, αφού σκοπός του μαθήματος, όπως προείπαμε, δεν είναι η εκμάθηση κάποιας γλώσσας προγραμματισμού. Όμως για τη βοήθειά σου έχουμε εντάξει στο τέλος του τετραδίου εργασιών ένα παράρτημα που περιλαμβάνει τέσσερα συνοπτικά εγχειρίδια χρήσης, καθένα από τα οποία αναφέρεται στις παραπάνω γλώσσες προγραμματισμού.

Τα παραδείγματα που παρουσιάζονται και οι δραστηριότητες που προτείνονται:

- αντλούν ιδέες από πραγματικές καταστάσεις και από εμπειρίες της καθημερινής ζωής ή θίγουν πολιτιστικά, πολιτισμικά και κοινωνικά θέματα ευρύτερου ενδιαφέροντος, δίνοντάς σου έτσι το ερέθισμα για περαιτέρω προβληματισμό,
- συνδέονται αρκετές φορές με άλλα μαθήματα όπως μαθηματικά, φυσική, χημεία, βιολογία, για να σου υπενθυμίζουν έμμεσα ότι ο υπολογιστής δεν είναι αυτοσκοπός, αλλά εργαλείο επίλυσης προβλημάτων,
- δίνουν μεγαλύτερη έμφαση στην ανάλυση του προβλήματος και στο σχεδιασμό της λύσης παρά στην υλοποίησή της, υποδηλώνοντάς σου έτσι συνέχεια ότι η προσπάθεια που

θα πρέπει να καταβάλεις δεν είναι προς την κατεύθυνση της καλλιέργειας τεχνικής, αλλά προς εκείνη της ανάπτυξης αναλυτικής και συνθετικής σκέψης.

Συμβάσεις

Για την καλύτερη αναγνωσιμότητα του τετραδίου έχουν χρησιμοποιηθεί και μερικά γνωστά εικονίδια από το βιβλίο μαθητή. Εκτός από αυτά, για τη διαβάθμιση των προτεινόμενων δραστηριοτήτων και ασκήσεων χρησιμοποιήθηκαν και τα παρακάτω:

για μέτρια

για προωθημένη

Ευχαριστίες

Για τη δημιουργία των εγχειριδίων χρήσης των Turbo Pascal και Delphi βοήθησαν οι Κώστας Αντωνακόπουλος και Βαγγέλης Χαραλαμπίδης, τους οποίους ευχαριστούμε θερμά και από τη θέση αυτή. Θα θέλαμε επίσης να ευχαριστήσουμε όλους εκείνους τους συναδέλφους καθηγητές Πληροφορικής, και ιδιαίτερα τους **Γιάννη Οικονόμου** και **Ιωάννη Φαμέλη**, για τις εποικοδομητικές παρατηρήσεις που μας έστειλαν κατά τον πρώτο χρόνο χρησιμοποίησης του βιβλίου, βοηθώντας μας με αυτόν τον τρόπο να διορθώσουμε και να βελτιώσουμε το περιεχόμενό του.

Τέλος, θα θέλαμε να ευχαριστήσουμε το συνάδελφο **Νίκο Κασσιμάτη** για τη συνεισφορά του στην παιδαγωγική προσέγγιση του περιεχομένου του βιβλίου.

Οι συγγραφείς

Κεφάλαιο

1

Ανάλυση Προβλήματος

1.1. Προσδοκώμενα αποτελέσματα

Η μελέτη αυτού του πρώτου κεφαλαίου αναμένεται ότι θα σου καταστήσει σαφή την έννοια του προβλήματος. Η σωστή αντιμετώπιση ενός προβλήματος προϋποθέτει την καταρχήν πλήρη κατανόησή του. Η βάση της κατανόησης είναι η σαφής διατύπωσή του και αυτό απαιτεί σωστή χρήση του γραπτού και του προφορικού λόγου. Μέσα από τα παραδείγματα που αναφέρονται και τις δραστηριότητες που προτείνονται, θα μπορέσεις να καλλιεργήσεις την αναλυτική σου ικανότητα, ώστε να είσαι σε θέση να προσδιορίζεις τα συστατικά μέρη ενός προβλήματος και να το αναλύεις στη συνέχεια σε απλούστερα. Θα μάθεις να αναγνωρίζεις τα δεδομένα ενός προβλήματος και να προσδιορίζεις τα ζητούμενα αποτελέσματα στην επιθυμητή μορφή. Τέλος, θα είσαι σε θέση να θέσεις ο ίδιος προβλήματα διατυπώνοντάς τα με πληρότητα και ακρίβεια.

1.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

Στο κεφάλαιο 1.4 Καθορισμός απαιτήσεων του βιβλίου του μαθητή, παρουσιάζεται το πρόβλημα “Αποτελέσματα φοίτησης μαθητών Γ΄ τάξης Τεχνολογικής Κατεύθυνσης στα μαθήματα ειδικότητας”. Εδώ θα προτείνουμε μια επέκτασή του προχωρώντας καταρχήν σε μια πιο αυστηρή διατύπωση του προβλήματος:

“Δίδονται οι βαθμολογίες όλων των μαθητών Γ’ Λυκείου Τεχνολογικής Κατεύθυνσης του σχολικού έτους 1999/2000 στα τέσσερα μαθήματα ειδικότητας. Ζητείται να εκδοθούν στατιστικά αποτελέσματα κατά μάθημα, που περιλαμβάνουν (α) πίνακα συχνοτήτων, (β) τη μέση τιμή και την τυπική απόκλιση”.

Οι απαιτούμενες ενέργειες για την αντιμετώπιση του προβλήματος είναι αυτές που φαίνονται στο σχήμα 1.1.



Σχ. 1.1.

Καταχώριση δεδομένων. Οι βαθμολογίες όλων των μαθητών για ένα μάθημα συγκεντρώνονται και καταγράφονται.

Έλεγχος δεδομένων. Τα δεδομένα ελέγχονται ως προς την ορθότητά τους και γίνονται οι απαραίτητες διορθώσεις, αν απαιτείται.

Επεξεργασία δεδομένων. Γίνονται οι απαραίτητοι υπολογισμοί προκειμένου να βρεθούν τα ζητούμενα αποτελέσματα.

Εξαγωγή αποτελεσμάτων. Δημιουργείται ο πίνακας συχνοτήτων (βλέπε παρ. 1.4 του βιβλίου), σχεδιάζεται το γράφημα και αποτυπώνεται η μέση τιμή και η τυπική απόκλιση.

Από τα παραπάνω μέρη του προβλήματος δεν είναι αρκετά σαφές τι περιλαμβάνει η επεξεργασία δεδομένων. Δηλαδή ποιοι ακριβώς είναι οι απαραίτητοι υπολογισμοί για την εύρεση των αποτελεσμάτων. Έτσι το μέρος αυτό κρίνεται ότι πρέπει να αναλυθεί περισσότερο, όπως στη συνέχεια.

Οι απαιτούμενοι υπολογισμοί είναι:

Βρίσκεται το πλήθος όλων των μαθητών, έστω N .

Καταμετρείται το πλήθος των μαθητών που έχει βαθμολογία ίση ή μικρότερη του 9 έστω $K1$, από 10 έως 13 έστω $K2$, κ.ο.κ.

Το ποσοστό των απορριπτόμενων μαθητών βρίσκεται από τον τύπο $K1/N \cdot 100$.

Αθροίζονται όλες οι βαθμολογίες και έστω S το άθροισμα. Η μέση τιμή μ υπολογίζεται από τη σχέση:

$$\mu = \frac{S}{N}$$

Αθροίζονται επίσης τα τετράγωνα των βαθμολογιών και έστω $S2$ το άθροισμα αυτό. Η τυπική απόκλιση σ βρίσκεται από τον τύπο:

$$\sigma^2 = \frac{S2}{N} - \mu^2$$

Το παράδειγμα εδώ ολοκληρώνεται σε θεωρητικό επίπεδο. Απομένει η υλοποίησή του με πειραματικά ή πραγματικά δεδομένα, όπως ζητείται από τη δραστηριότητα ΔΣ5.



1.3. Συμβουλές - υποδείξεις

Η επιτυχής προσπάθεια αντιμετώπισης ενός προβλήματος εξαρτάται σε πολύ μεγάλο βαθμό από τη σωστή κατανόησή του. Επομένως, πριν αρχίσεις οποιαδήποτε προσπάθεια επίλυσης ενός προβλήματος θα πρέπει να ασχοληθείς επισταμένως με την εκφώνησή του, ώστε να είσαι απολύτως βέβαιος ότι έχεις κατανοήσει σωστά και σε όλο τους το εύρος τα ζητούμενα.

Σημαντική διευκόλυνση για σένα θα είναι να μπορέσεις να αναλύσεις το πρόβλημα σε άλλα απλούστερα. Η αντιμετώπιση απλούστερων προβλημάτων είναι βέβαια πιο εύκολη. Θα πρέπει να έχεις υπόψη σου ότι κάθε κανόνας έχει τις εξαιρέσεις του, οπότε δεν θα πρέπει να εκπλαγείς αν μετά την ανάλυση του προβλήματος κάποιο από τα επιμέρους προβλήματα που προέκυψαν είναι πολύ δύσκολο να αντιμετωπιστεί. Θα πρέπει όμως να είσαι βέβαιος πως η δυσκολία αυτή δεν είναι μεγαλύτερη από αυτήν που έχει το κύριο πρόβλημα.

Αφού κάποτε θα βρεθείς οπωσδήποτε στη θέση να διατυπώσεις ένα πρόβλημα, θα πρέπει να δώσεις προσοχή στη διατύπωσή του, έτσι ώστε να μη δημιουργεί παρερμηνείες και συγχύσεις σε κάποιον που θα κληθεί να το αντιμετωπίσει. Ιδιαίτερα μεγάλη προσοχή απαιτείται αν το πρόβλημα “εκφράζεται” προς υπολογιστή, αφού η μηχανή δεν έχει την ευχέρεια να καταλάβει αυτά που θέλεις να δηλώσεις, αν δεν είναι απόλυτα σωστά διατυπωμένο.

1.4. Δραστηριότητες - ασκήσεις

ΔΤ1. Η διεθνής αντιρατσιστική οργάνωση *SOS Ρατσισμός*, στην προσπάθειά της να συμβάλει στην καταπολέμηση της ξενοφοβίας, διεξήγαγε μια ενημερωτική καμπάνια σε όλες τις ευρωπαϊκές πρωτεύουσες της Ενωμένης Ευρώπης. Κεντρικό σημείο της προσπάθειας αυτής ήταν η ενημέρωση των πολιτών σε θέματα φυλετικών διακρίσεων, αλλά και η υποβολή ερωτημάτων στους πολίτες εκ μέρους της οργάνωσης, με σκοπό τη συλλογή σχετικών απαντήσεων που θα μπορούσαν στη συνέχεια να χρησιμοποιηθούν σαν στατιστικά στοιχεία. Στην Αθήνα τα ενημερωτικά κιόσκια στήθηκαν σε 3 κεντρικά σημεία, όπου μοιραζόντουσαν ενημερωτικά φυλλάδια και ετίθεντο και ερωτήσεις στους διερχόμενους. Οι απαντήσεις των ερωτηθέντων χρησιμοποιήθηκαν για την εξαγωγή συμπερασμάτων και τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

Ερώτηση	Ναι	Όχι	Δεν απάντησαν
Θα στέλνατε το παιδί σας σε ένα σχολείο με μεγάλο αριθμό παιδιών Αλβανών μεταναστών;	27%	61%	12%
Θα νοικιάζατε το διαμέρισμά σας σε μετανάστες της πρώην ανατολικής Ευρώπης ή του τρίτου κόσμου;	65%	32%	3%
Θα παίρνατε για κάποια πρόχειρη δουλειά (βάψιμο, οικιακή βοήθος κ.λπ.) έναν/μία μετανάστη;	89%	7%	4%

Όπως στην Αθήνα, έτσι και στις άλλες ευρωπαϊκές πρωτεύουσες στήθηκαν παρόμοια κιόσκια και τέθηκαν παρόμοιες ερωτήσεις. Τα αποτελέσματα των ερευνών έδωσαν ανάλογα συμπεράσματα για κάθε χώρα.

Στο τέλος θεωρήθηκε σκόπιμο να βγουν κάποια συμπεράσματα συνολικά για την Ευρωπαϊκή Ένωση. Για το σκοπό αυτό χρησιμοποιήθηκαν οι πίνακες αποτελεσμάτων, όπως ο παραπάνω, για τις 15 ευρωπαϊκές πρωτεύουσες.

- α) Να εντοπίσετε σε όλη την παραπάνω περιγραφόμενη ενέργεια ποια στοιχεία αποτελούν δεδομένα και ποια πληροφορίες για ποια διαδικασία. Σχολιάστε τις απαντήσεις.
- β) Να αναλύσετε και να σχολιάσετε το πρόβλημα Ρατσισμός και να εκφράσετε λεκτικά και διαγραμματικά την ανάλυσή σας.

ΔΤ2. Το ενιαίο ευρωπαϊκό νομισματικό σύστημα είναι προ των πυλών. Φανταστείτε πώς θα μπορούσαν οι ευρωπαϊκές χώρες να εφαρμόσουν στην πράξη το σύστημα αυτό αντιμετωπίζοντας με επιτυχία τα προβλήματα αποδοχής που θα συναντήσει από τους υπηκόους τους, οι οποίοι θα αντιδρούν στην αντικατάσταση του εθνικού τους νομίσματος. Αναλύσατε καταρχήν το πρόβλημα φραστικά και στη συνέχεια κάντε τη διαγραμματική του αναπαράσταση.

ΔΤ3. Μία σειρά από γραφειοκρατικές διαδικασίες διαφόρων υπηρεσιών έχουν αρχίσει ήδη να εξαλείφονται με τη χρήση υπολογιστών. Οι πολίτες δεν είναι πλέον υποχρεωμένοι να πηγαίνουν στις δημόσιες υπηρεσίες και να σχηματίζουν ουρές προκειμένου να παραλάβουν ένα πιστοποιητικό. Καλώντας έναν τηλεφωνικό αριθμό, μπορούν να δίνουν κάποια προσωπικά τους στοιχεία και να παραλαμβάνουν το πιστοποιητικό ταχυδρομικά στο σπίτι τους. Προσδιορίστε τα απαραίτητα δεδομένα που θα πρέπει να δίνει τηλεφωνικά ο πολίτης στην περίπτωση που θέλει να πάρει α) πιστοποιητικό γέννησης για έκδοση διαβατηρίου, β) φορολογική ενημερότητα για αγορά αυτοκινήτου.

ΔΤ4. Να επιλέξετε κάποιο από τα σοβαρότερα προβλήματα που απασχολούν τη μαθητική κοινότητα του σχολείου σας ή συνολικά τη μαθητική κοινότητα της χώρας μας. Να το διατυπώσετε με ακρίβεια και πληρότητα. Στη συνέχεια να προσδιορίσετε τα δεδομένα και τα ζητούμενα αποτελέσματα. Τέλος, μέσω της ανάλυσής του, να προτείνετε λύση του.

ΔΤ5. Ας υποθέσουμε ότι σήμερα είναι η γιορτή του Αγίου Γεωργίου και ότι θέλετε να τηλεφωνήσετε σε όλους τους φίλους σας και τις φίλες σας που γιορτάζουν για να τους ευχηθείτε. Θα πρέπει λοιπόν να ψάξετε στο προσωπικό σας σημειωματάριο για να βρείτε τα ονόματα και τα τηλέφωνα όλων όσοι γιορτάζουν. Μια σκέψη που μπορεί να κάνετε μεγαλόφωνα είναι: “Να ψάξω να βρω όλους όσοι λέγονται Γιώργος και Γεωργία”.

Σχολιάστε τη διατύπωση αυτής της σκέψης.

Προβληματιστείτε για τα αποτελέσματα, αν ακριβώς τη σκέψη σας αυτή τη μεταφράζατε σε μία γλώσσα προγραμματισμού και βάζατε τον υπολογιστή να βρει τους εορτάζοντες από το ηλεκτρονικό σας σημειωματάριο που κρατάτε σε αυτόν.

ΔΤ6. Να σχολιασθεί η άποψη: “Οι υπολογιστές δεν είναι ούτε κατάρρα ούτε πανάκεια”.

Δεν είναι στη “φύση” των νέων τεχνολογιών να δημιουργούν προβλήματα ή να συντελούν στην ανθρώπινη πρόοδο. Οι τρόποι χρήσης είναι αυτοί που επηρεάζουν την ανθρώπινη ζωή και τις κοινωνίες. Να διατυπώσετε προβλήματα που δημιουργούνται αλλά και προβλήματα που λύνονται με τη χρήση των υπολογιστών.

ΔΣ1. Η χρήση κεντρικών υπολογιστικών συστημάτων δημιουργεί και μια σειρά από κοινωνικά προβλήματα, ένα από τα οποία είναι ο κίνδυνος καταπάτησης του ιδιωτικού απορρήτου. Η δυνατότητα πρόσβασης σε στοιχεία των πολιτών, από τεχνικής πλευράς, είναι δύσκολο να αποκλειστεί. Τα καταχωρημένα στοιχεία μπορούν να αφορούν ατομικά στοιχεία (ονοματεπώνυμο, έτος και τόπος γέννησης, διεύθυνση κατοικίας κ.λπ.), κοινωνικοπολιτικά στοιχεία (θρήσκευμα, πολιτική τοποθέτηση, συνδικαλιστική δράση κ.λπ.), οικονομικά στοιχεία (ΑΦΜ, στοιχεία φορολογικών δηλώσεων, δάνεια, πιστωτικές κάρτες κ.λπ.), ιατρικά στοιχεία (ασθένειες, νοσηλείες, θεραπείες κ.λπ.) καθώς και άλλα διάφορα στοιχεία (κλήσεις τροχαίας, αεροπορικά ταξίδια κ.λπ.). Καλείστε να προτείνετε τρόπους αντιμετώπισης του προβλήματος.

ΔΣ2. Οι κίνδυνοι εθισμού και εξάρτησης από την αλόγιστη χρήση των υπολογιστών, ειδικά για τα παιδιά και τους εφήβους, είναι μεγάλοι. Πώς θα μπορούσε να αντιμετωπιστεί δραστικά αυτό το πρόβλημα; Ποιες είναι οι γενεσιουργές του αιτίες; Αυτά είναι, μεταξύ πολλών άλλων, μερικά από τα ερωτήματα που θα πρέπει να σας απασχολήσουν ώστε να μπορέσετε να προσδιορίσετε σωστά το πρόβλημα και να το αναλύσετε σε επιμέρους προβλήματα. Καταγράψτε τη φραστική ανάλυση για την αντιμετώπισή του και στη συνέχεια κάντε και τη διαγραμματική του αναπαράσταση.

ΔΣ3. Υποθέστε ότι είστε μέλος της σχολικής επιτροπής του σχολείου σας και ότι πρέπει να αντιμετωπίσετε το θέμα της αγοράς εξοπλισμού υπολογιστικών συστημάτων για τις διαφορετικές ανάγκες του σχολείου. Πρέπει να επιλέξετε εξοπλισμό για το εργαστήριο πληροφορικής, για το εργαστήριο καλλιτεχνικών σπουδών, για τη γραμματεία του σχολείου και για την αίθουσα εκδηλώσεων/παρουσιάσεων. Τι είδους εξοπλισμό θα διαλέγατε για να ικανοποιήσετε τις ανάγκες καθενός από αυτούς τους χώρους; Καταγράψτε το βασικό εξοπλισμό και τις απαραίτητες περιφερειακές μονάδες για κάθε χώρο ξεχωριστά.

ΔΣ4. Επιλέξτε ένα πρόβλημα από τον προσωπικό σας χώρο ή από τον κοινωνικό χώρο που σας απασχολεί. Διατυπώστε το με ακρίβεια και πληρότητα έτσι ώστε παρουσιάζοντάς το στη συνέχεια στην τάξη σας να γίνει απόλυτα κατανοητό από όλους.

ΔΣ5. Με τη συνεργασία του καθηγητή σας συγκεντρώστε στοιχεία βαθμολογίας μαθητών προηγούμενων ετών του σχολείου σας και πραγματοποιήστε την ανάλυση προβλήματος (για τέσσερα μαθήματα της επιλογής σας) που αναφέρεται στο κεφάλαιο 1.4 του βιβλίου και συνεχίζεται στο παράδειγμα του τετραδίου.

1.5. Τεστ αυτοαξιολόγησης



1. Δίνονται οι παρακάτω ομάδες λέξεων. Σε καθεμία από αυτές, να βάλεις τις λέξεις στη σωστή σειρά:

1. Επίλυση, ανάλυση, κατανόηση (αναφορά σε πρόβλημα)
2. Επεξεργασία, έλεγχος, έξοδος, είσοδος (αναφορά σε δεδομένα)

2. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

3. Η επίλυση ενός προβλήματος ξεκινά από την _____ του.
4. _____ είναι το αποτέλεσμα επεξεργασίας δεδομένων.
5. Σημαντικός παράγοντας στην κατανόηση ενός προβλήματος είναι η _____.
6. Με τον όρο _____ προβλήματος αναφερόμαστε στα συστατικά μέρη που το αποτελούν.
7. Για να μπορέσουμε να επιλύσουμε ένα πρόβλημα θα πρέπει να γίνει ο καθορισμός _____.

3. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

8. Πρόβλημα είναι μια οποιαδήποτε κατάσταση που πρέπει να αντιμετωπίσουμε.
9. Ο ανθρώπινος εγκέφαλος είναι ένας μηχανισμός επεξεργασίας δεδομένων.
10. Για την παραγωγή πληροφοριών απαιτούνται δεδομένα.
11. Ο υπολογιστής και το πρόβλημα είναι έννοιες αλληλένδετες.
12. Ένα πρόβλημα μπορεί να αναπαρασταθεί είτε διαγραμματικά είτε φραστικά.

Βασικές Έννοιες Αλγορίθμων

2.1. Προσδοκώμενα αποτελέσματα

Ολοκληρώνοντας αυτό το κεφάλαιο, θα έχεις κατανοήσει ακριβώς την έννοια του αλγορίθμου. Θα έχεις συνειδητοποιήσει τη σπουδαιότητα των αλγορίθμων ως μεθοδολογία σκέψης και ως εργαλείο αντιμετώπισης των προβλημάτων. Θα έχεις διαπιστώσει μέσα από τα παρουσιαζόμενα παραδείγματα και από τις ασκήσεις που θα λύσεις την αναγκαιότητα αλγοριθμικής προσέγγισης κατά τη διαδικασία επίλυσης των προβλημάτων. Θα έχεις μπορέσει να εξασκηθείς στη μορφοποίηση αλγορίθμων με χρήση συγκεκριμένων τεχνικών. Έτσι λοιπόν εισάγεσαι στα “εργαλεία” ανάπτυξης αλγορίθμων, δηλαδή στη μεθοδολογία επίλυσης προβλημάτων με προγραμματισμό.

2.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1. Μετατροπή από βαθμούς Φαρενάιτ σε βαθμούς Κελσίου

Η μετατροπή μίας θερμοκρασιακής τιμής από βαθμούς Φαρενάιτ σε βαθμούς Κελσίου γίνεται με βάση τον τύπο:

$$C = \frac{5(F - 32)}{9}$$

όπου οι μεταβλητές C και F συμβολίζουν τις αντίστοιχες τιμές. Η μετατροπή αυτή γίνεται εύκολα με τον επόμενο αλγόριθμο που έχει ακολουθιακή δομή.

Αλγόριθμος Θερμοκρασία
Διάβασε fahrenheit
celsius $\leftarrow$ (fahrenheit-32) * 5 / 9
Εκτύπωσε celsius
Τέλος Θερμοκρασία

ΠΑΡΑΔΕΙΓΜΑ 2. Υπολογισμός γεωμετρικών μεγεθών

Έστω ότι δεδομένου του μήκους της ακτίνας θέλουμε να υπολογίσουμε το εμβαδόν του αντίστοιχου κύκλου, το εμβαδόν του τετραγώνου που είναι περιγεγραμμένο στο δεδομένο κύκλο και το μήκος της διαγωνίου του τετραγώνου αυτού. Ο επόμενος αλγόριθμος επιλύει το γεωμετρικό αυτό πρόβλημα, όπου τα ονόματα των μεταβλητών είναι προφανή. Τέλος, διευκρινίζεται ότι ο ακόλουθος αλγόριθμος καλεί έναν αλγόριθμο ονομαζόμενο Ρίζα, που επιστρέφει την τετραγωνική ρίζα ενός θετικού αριθμού.

Αλγόριθμος Γεωμετρικός
Διάβασε aktina
emvadon $\leftarrow$ 3.14 * aktina * aktina
plevra $\leftarrow$ 2 * aktina
tetragwno $\leftarrow$ plevra * plevra
diagwnios $\leftarrow$ Ρίζα(2 * tetragwno)
Εκτύπωσε emvadon, tetragwno, diagwnios
Τέλος Γεωμετρικός

ΠΑΡΑΔΕΙΓΜΑ 3. Τιμές θερμοκρασίας από Μετεωρολογικό Κέντρο

Σε ένα μετεωρολογικό κέντρο χρειάζεται να βρεθεί η μέγιστη και η ελάχιστη θερμοκρασία από τις μέσες ημερήσιες θερμοκρασίες ενός μήνα. Να γραφεί ένας αλγόριθμος που θα διαβάσει τη μέση ημερήσια θερμοκρασία για κάθε ημέρα ενός μήνα 30 ημερών και θα υπολογίζει την ελάχιστη και τη μέγιστη από αυτές τις θερμοκρασίες.



Για τον υπολογισμό ελάχιστης και μέγιστης θερμοκρασίας είναι βασικό να δοθούν αρχικές τιμές στις μεταβλητές που θα κρατήσουν τις τιμές για να μπορεί να γίνει σωστά η σύγκριση. Εάν, για παράδειγμα, στη μεταβλητή MIN δώσουμε αρχική τιμή 0, δεν θα καταλήξουμε σε σωστή ελάχιστη θερμοκρασία, εφόσον στο μήνα δεν υπάρχουν αρνητικές θερμοκρασίες. Αντίθετα, εάν στο MAX δώσουμε αρχική τιμή 0, δεν θα καταλήξουμε σε σωστή μέγιστη θερμοκρασία, στην περίπτωση που όλος ο μήνας είχε καθημερινή αρνητική μέση θερμοκρασία. Επομένως είναι χρήσιμο η MIN να έχει αρκετά υψηλή θερμοκρασία ως αρχική τιμή, ενώ αντίθετα η MAX να έχει αρκετά χαμηλή θερμοκρασία ως αρχική τιμή.

```

Αλγόριθμος Ελάχιστη_Μέγιστη1
MIN ← 100
MAX ← -100
Για i από 1 μέχρι 30
    Διάβασε THEP
    Αν THEP < MIN τότε MIN ← THEP
    Αν THEP > MAX τότε MAX ← THEP
Τέλος_επανάληψης
Αποτελέσματα // MIN, MAX//
Τέλος Ελάχιστη_Μέγιστη1

```

ΠΑΡΑΔΕΙΓΜΑ 4. Επίλυση δευτεροβάθμιας εξίσωσης

Η περίπτωση της δευτεροβάθμιας εξίσωσης είναι παρόμοια. Αρχικά είναι απαραίτητο η τιμή του A να είναι μη μηδενική, πράγμα που ελέγχεται κατά την είσοδο. Στη συνέχεια, για την εύρεση πραγματικών ριζών της εξίσωσης $Ax^2+Bx+C=0$, πρέπει να ελεγχθεί αν η διακρίνουσα είναι θετική. Και πάλι καλείται ο αλγόριθμος Ρίζα, που επιστρέφει την τετραγωνική ρίζα ενός θετικού αριθμού.

```

Αλγόριθμος ΕξίσωσηB
Αρχή_επανάληψης
    Διάβασε a
Μέχρις_ότου a≠0
    Διάβασε b
    Διάβασε c
    delta ← b*b-4*a*c
    Αν delta ≥ 0 τότε
        solution1 ← (-b+Πίζα(delta))/(2*a)
        solution2 ← (-b-Πίζα(delta))/(2*a)
        Εκτύπωσε solution1, solution2
    Τέλος_αν
Τέλος ΕξίσωσηB

```

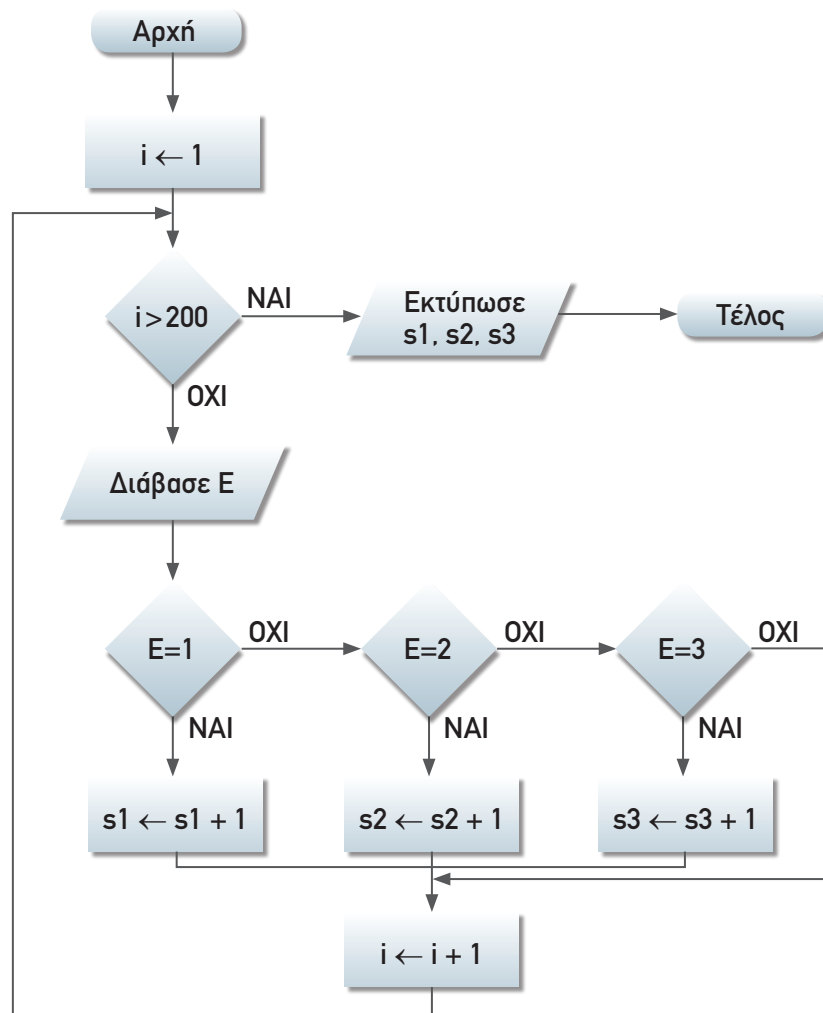
ΠΑΡΑΔΕΙΓΜΑ 5. Φοίτηση στο Πανεπιστήμιο

Σε κάποια Σχολή υπάρχει ένα 3ετές Τμήμα με διαφορετικό αριθμό φοιτητών/φοιτητριών ανά έτος φοίτησης. Συνολικά το Τμήμα αυτό έχει 200 φοιτητές. Να σχεδιασθεί ένα διάγραμμα ροής και να γραφεί ένας αλγόριθμος που θα διαβάζει το έτος κάθε φοιτητή του Τμήματος και θα υπολογίζει τον αριθμό των φοιτητών για κάθε έτος φοίτησης.



Είναι χρήσιμο εδώ να χρησιμοποιηθεί η διαδικασία των πολλαπλών επιλογών διότι είναι ένα πρόβλημα όπου χρειάζεται να γίνει ξεχωριστός υπολογισμός για τις διακριτές τιμές 1, 2, 3 που είναι τα έτη φοίτησης στο συγκεκριμένο Τμήμα.

Διάγραμμα ροής



Αλγόριθμος

Αλγόριθμος Φοιτητές_Ετος

$s1 \leftarrow 0$ $s2 \leftarrow 0$ $s3 \leftarrow 0$

Για i **από** 1 **μέχρι** 200

Διάβασε E

Αν $E = 1$ **τότε** $s1 \leftarrow s1 + 1$

αλλιώς_αν $E = 2$ **τότε** $s2 \leftarrow s2 + 1$

αλλιώς_αν $E = 3$ **τότε** $s3 \leftarrow s3 + 1$

Τέλος_αν

Τέλος_επανάληψης

Αποτελέσματα // $s1, s2, s3$ //

Τέλος Φοιτητές_Ετος

ΠΑΡΑΔΕΙΓΜΑ 6. Διοφαντική ανάλυση

Να εκπονηθεί ένας αλγόριθμος για την εύρεση όλων των ακεραίων λύσεων της εξίσωσης:

$$3x + 2y - 7z = 5$$

για τιμές των x, y, z μεταξύ των 0 και 100. Η επίλυση τέτοιων εξισώσεων με πολλές μεταβλητές που επιδέχονται πολλές λύσεις ονομάζεται διοφαντική ανάλυση. Αλγοριθμικά το πρόβλημα αντιμετωπίζεται ως εξής:

Αλγόριθμος Διοφαντική

Για x **από** 0 **μέχρι** 100

Για y **από** 0 **μέχρι** 100

Για z **από** 0 **μέχρι** 100

Αν $3 * x + 2 * y - 7 * z = 5$ **τότε Εκτύπωσε** x, y, z

Τέλος_επανάληψης

Τέλος_επανάληψης

Τέλος_επανάληψης

Τέλος Διοφαντική



2.3. Συμβουλές - υποδείξεις

Από την αρχή της ενασχόλησής σου με τους αλγορίθμους, είναι χρήσιμο να μάθεις να ακολουθείς κάποιους κανόνες και κάποιες γενικές αρχές, έτσι ώστε να μπορείς να λύσεις πραγματικά προβλήματα με μεθοδικό τρόπο και να βρίσκεις την καλύτερη τεχνική για την επίλυση ενός προβλήματος. Τη σπουδαιότητα των αλγορίθμων καθώς και την αναγκαιότητά τους για την επίλυση προβλημάτων θα την καταλαβαίνεις όλο και καλύτερα όσο τα προβλήματα γίνονται περισσότερο σύνθετα και πολύπλοκα.

Ο αλγόριθμός σου πρέπει να είναι απλός και να προτείνει την εξυπνότερη δυνατή λύση σε ένα πρόβλημα. Είναι χρήσιμο να προσπαθείς κάθε φορά να εντάξεις ένα πρόβλημα σε ένα σύνολο από διαδοχικά βήματα σε φυσική γλώσσα και στη συνέχεια να καταγράφεις αυτά τα βήματα σε κάποια αλγοριθμική δομή.

Θα πρέπει να χρησιμοποιείς επαναληπτικές δομές για προβλήματα στα οποία μία ακριβώς ίδια ενέργεια γίνεται για ένα σύνολο από παρόμοιες οντότητες (π.χ. για 100 μαθητές, για 20 αυτοκίνητα κ.λπ.). Είναι χρήσιμο να αναγνωρίσεις την αλγοριθμική δομή που βολεύει ανάλογα με την εκφώνηση του προβλήματος.

2.4. Δραστηριότητες - ασκήσεις

ΔΤ1. Ο υπολογισμός της περιόδου του εκκρεμούς δίνεται από τον τύπο:

$$T = 2\pi\sqrt{\frac{L}{g}}$$

όπου L είναι το μήκος του εκκρεμούς και g είναι η επιτάχυνση της βαρύτητας. Να γραφεί αλγόριθμος που να υλοποιεί τον τύπο αυτό.

ΔΤ2. Να γράψετε με βήματα αλγορίθμου τη διαδικασία μετατροπής των παρακάτω νομισμάτων σε ευρώ, με δεδομένο ότι έχετε τις παρακάτω πληροφορίες:

1. Το ευρώ έχει τιμή πώλησης 340,75 δρχ.
2. Η λίρα Αγγλίας έχει τιμή πώλησης 550 δρχ.
3. Το δολάριο Αμερικής έχει τιμή πώλησης 280 δρχ.
4. Το μάρκο Γερμανίας έχει τιμή πώλησης 100 δρχ.

Στη συνέχεια να υπολογίσετε σε δραχμές το σύνολο από 1025 λίρες Αγγλίας, 2234 δολάρια Αμερικής και 3459 μάρκα Γερμανίας.

ΔΤ3. Να γράψετε με βήματα αλγορίθμου και με διάγραμμα ροής τα παρακάτω:

1. Το μέσο όρο ηλικιών μίας ομάδας 100 ανθρώπων.
2. Το σύνολο βαθμολογίας όλων των ομάδων που έχουν πάρει περισσότερους από 100 βαθμούς σε ένα διαγωνισμό.

ΔΤ4. Τι τύπου αλγοριθμική συνιστώσα πρέπει να χρησιμοποιήσετε για τα παρακάτω στοιχεία υπολογισμού; Γράψτε το αντίστοιχο τμήμα δηλώσεων.

1. Το σύνολο ποσού για μία λίστα από 100 αντικείμενα.
2. Τη βαθμολογία ενός μαθητή εάν έχει περάσει τα μαθήματά του.
3. Το μέσο όρο βαθμολογίας 100 μαθητών.
4. Διάβασε όνομα και τηλέφωνο ενός μαθητή.
5. Διάβασε όνομα, διεύθυνση και τηλέφωνο 25 μαθητών.
6. Τον αριθμό που προκύπτει όταν ρίξουμε ένα ζάρι.

ΔΤ5. Να διαβάζονται δύο αριθμοί που αντιστοιχούν στο ποσοστό του διοξειδίου του άνθρακα και του αζώτου μίας ημέρας, όπως έχει καταγραφεί στα ειδικά μηχανήματα καταγραφής στην ατμόσφαιρα της πόλης. Να εκτυπώνεται ότι η ατμόσφαιρα είναι «καθαρή», αν το ποσοστό του διοξειδίου του άνθρακα είναι κάτω από 0.35, ή να εκτυπώνεται «μολυσμένη» στην αντίθετη περίπτωση. Επίσης να εκτυπώνεται «διαυγής», αν το άζωτο είναι κάτω από 0.17, αλλιώς να εκτυπώνεται «αδιαυγής».

ΔΤ6. Έστω ότι ένας Πανελλήνιος Διαγωνισμός στα Μαθηματικά δίνει δικαίωμα συμμετοχής στο 1% των μαθητών μίας τάξης με την προϋπόθεση ότι ο μέσος όρος της βαθμολογίας στα Μαθηματικά των μαθητών αυτής της τάξης είναι μεγαλύτερος από

18. Να γραφεί ένας αλγόριθμος που θα ελέγχει τη δυνατότητα συμμετοχής σε έναν τέτοιο διαγωνισμό και να παρακολουθήσετε τον αλγόριθμο για τα δεδομένα της τάξης σας.

ΔΤ7. Οι υπάλληλοι μίας εταιρείας συμφώνησαν για το μήνα Δεκέμβριο να κρατηθούν από το μισθό τους δύο ποσά, ένα για την ενίσχυση του παιδικού χωριού SOS και ένα για την ενίσχυση των σκοπών της UNICEF. Ο υπολογισμός του ποσού των εισφορών εξαρτάται από τον αρχικό μισθό του κάθε υπαλλήλου και υπολογίζεται με βάση τα παρακάτω όρια μισθών:

Μισθός	Εισφορά 1	Εισφορά 2
Έως 500€	5%	4%
501 - 800	7,5%	6%
801 - 1100	9,5%	8%
μεγαλύτερος από 1100	12%	11%

Να γραφεί αλγόριθμος που να δέχεται ως είσοδο το μισθό του και στη συνέχεια να υπολογίζει το ποσό των δύο εισφορών και το καθαρό ποσό που θα πάρει ο υπάλληλος.

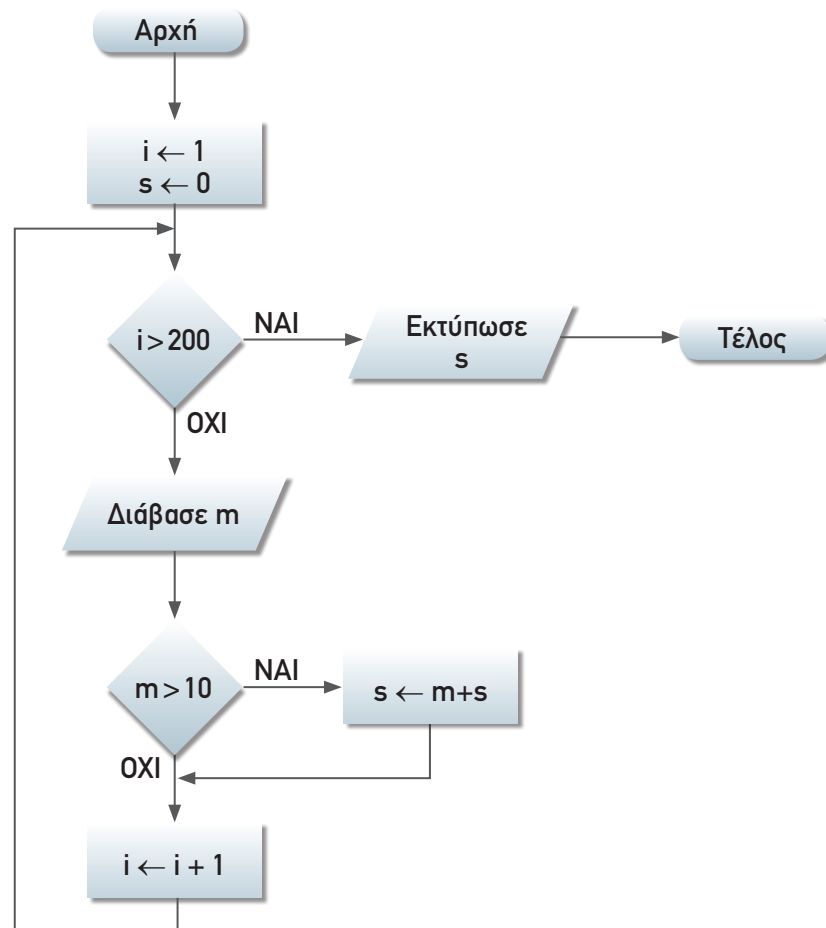
ΔΤ8. Σε 10 σχολεία της περιφέρειας έχουν εγκατασταθεί πειραματικά 10 ηλεκτρονικοί υπολογιστές (εξυπηρέτες) που περιέχουν πληροφοριακές «σελίδες» του Internet και μπορεί να προσπελάσει κανείς την πληροφορία τους μέσα από οποιονδήποτε ηλεκτρονικό υπολογιστή στον κόσμο. Να γραφεί ένας αλγόριθμος που θα διαβάζει το συνολικό αριθμό των προσπελάσεων που πραγματοποιήθηκε σε καθέναν από τους εξυπηρέτες αυτούς για διάστημα μιας ημέρας. Να βρεθεί ο εξυπηρέτης με το μικρότερο αριθμό προσπελάσεων καθώς και ο εξυπηρέτης με το μεγαλύτερο αριθμό προσπελάσεων.

ΔΤ9. Σε ένα φυτώριο υπάρχουν 3 είδη δένδρων που θα δοθούν για δενδροφύτευση. Το 1ο είδος δένδρου θα δοθεί στην περιοχή της Μακεδονίας, το 2ο στην περιοχή της Θράκης και το 3ο είδος στην περιοχή της Πελοποννήσου. Να σχεδιασθεί το διάγραμμα ροής και να γραφεί ένας αλγόριθμος που θα διαβάζει τον αριθμό του είδους του δένδρου και θα εκτυπώνει την περιοχή στην οποία θα γίνει η δενδροφύτευση.

ΔΤ10. Σε ένα μουσείο υπάρχουν 10 διαφορετικές αίθουσες που περιέχουν διάφορα έργα της ελληνιστικής περιόδου. Κάθε αίθουσα έχει το δικό της αριθμό που είναι από 101, 102, ..., έως 110. Να γράψεις έναν αλγόριθμο που θα διαβάζει τον αριθμό των επισκεπτών κάθε αίθουσας για μία ημέρα και θα υπολογίζει το μέσο όρο των επισκεπτών από όλες τις αίθουσες. Στη συνέχεια ο αλγόριθμος θα πρέπει να εκτυπώνει τους αριθμούς των αιθουσών που είχαν περισσότερους επισκέπτες από το μέσο όρο των επισκεπτών.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Δίνεται το παρακάτω διάγραμμα ροής:



Να δώσετε την εκφώνηση του προβλήματος που εκφράζεται με το συγκεκριμένο διάγραμμα ροής.

ΔΣ2. Έστω ότι σου έχουν δώσει ένα μεταχειρισμένο ηλεκτρονικό υπολογιστή για 6 μήνες. Θέλεις να διαπραγματευτείς την τιμή αυτού του υπολογιστή για να δεις αν μπορείς να τον αλλάξεις με κάποιο άλλο μοντέλο. Η αρχική τιμή του υπολογιστή που πήρες είναι 295.600 δρχ. και σου τον προσφέρουν για 256.000 δρχ. Είναι χρήσιμο να υπολογίσεις το ποσοστό της απαξίωσης για τον υπολογιστή αυτό δεδομένου ότι το ετήσιο ποσοστό υποτίμησης υπολογίζεται από τον παρακάτω τύπο:

$$\text{Ποσοστό_Απαξίωσης} = 1 - \left(\frac{\text{Τιμή_προσφοράς}}{\text{Αρχική_Τιμή}} \right)^{\frac{1}{\text{Αριθμός_ετών}}}$$

Να σχεδιασθεί το διάγραμμα ροής και να γραφεί ένας αλγόριθμος που θα υπολογίζει το ποσοστό απαξίωσης για τον υπολογιστή που πήρες για τους 6 μήνες. Στη συνέχεια

να γενικεύσεις τον αλγόριθμο, έτσι ώστε να δουλεύει επαναληπτικά για έναν αριθμό από διαφορετικά είδη των οποίων ξέρεις το αρχικό ποσό, το ποσό της προσφοράς και το χρονικό διάστημα για το οποίο θέλεις να υπολογίσεις τα ποσοστά απαξίωσης.

ΔΣ3. Ένας καταναλωτής πηγαίνει στο πολυκατάστημα και έχει στην τσέπη του 5.000 ευρώ. Ξεκινά να αγοράζει διάφορα είδη και ταυτόχρονα κρατά το συνολικό ποσό στο οποίο έχει φθάσει κάθε στιγμή που αγοράζει κάποιο είδος. Οι τιμές των ειδών που αγοράζει είναι σε δραχμές και είναι δεδομένο ότι 1 ευρώ=340,75 δραχμές. Να γραφεί σε φυσική γλώσσα, με ακολουθία βημάτων και με διάγραμμα ροής, ένας αλγόριθμος για τον υπολογισμό του ποσού από τα ψώνια που έγιναν και να σταματά η αγορά ειδών έτσι ώστε να μην ξεπεραστεί το ποσό που έχει διαθέσιμο ο καταναλωτής.

ΔΣ4. Δίνεται ο παρακάτω αλγόριθμος:

Αλγόριθμος Ελεγχος_Ανάθεσης

Διάβασε x

Όσο x > 1 **επανάλαβε**

Αν x είναι άρτιος **τότε**

 x ← x/2

αλλιώς

 x ← 3*x+1

Τέλος_αν

Τέλος_επανάληψης

Αποτελέσματα // x //

Τέλος Ελεγχος_Ανάθεσης

Να γράψεις τα αποτελέσματα αυτού του αλγορίθμου για x=13, x=9 και x=22. Τι παρατηρείς;

ΔΣ5. Σε ένα Λύκειο κάθε μαθητής αξιολογείται με βάση το μέσο όρο που θα έχει σε 5 βασικά μαθήματα. Να γραφεί ένας αλγόριθμος που θα διαβάζει τη βαθμολογία για καθένα από τα 5 αυτά μαθήματα και θα υπολογίζει το μέσο όρο του μαθητή.

Να αναλυθεί το πρόβλημα και να προταθεί λύση του με ακολουθία βημάτων και με διάγραμμα ροής.



Υπόδειξη: Για τον υπολογισμό του συνολικού μέσου όρου η χρήση επαναληπτικής δομής είναι σημαντική λόγω της “ελάφρυνσης” του κώδικα από παρόμοιες εντολές και από χρήση πολλαπλών μεταβλητών.

ΔΣ6. Πηγαίνεις σε ένα πολυκατάστημα και παρατηρείς τις παρακάτω τιμές για 4 διαφορετικά είδη γάλακτος.

Είδος	Τιμή	Ποσότητα
ΓΑΛΑ_Α	0,60€	300ml
ΓΑΛΑ_Β	0,65€	400ml
ΓΑΛΑ_Γ	1,20€	500ml
ΓΑΛΑ_Δ	1,35€	550ml

Να γράψεις έναν αλγόριθμο που θα υπολογίζει και θα εμφανίζει το είδος γάλακτος που έχει την πλέον συμφέρουσα τιμή.

ΔΣ7. Έστω ότι θέλεις να υπολογίσεις το ποσό που θα έχεις στο μέλλον με βάση το ποσό που τώρα έχεις αποταμιεύσει στην τράπεζα. Δίνεται ο παρακάτω τύπος υπολογισμού:

$$\text{Τελικό_Ποσό} = \text{Αρχικό_Ποσό} \cdot \left(1 + \frac{\frac{\text{ΕΠΙΤΟΚΙΟ}}{100}}{2} \right)^{2 \cdot \text{χρόνια}}$$

Να γράψεις έναν αλγόριθμο που να υπολογίζει το ποσό που θα έχεις μετά από 5 χρόνια με δεδομένο ότι το ετήσιο επιτόκιο είναι 6,5%. Να επεκτείνεις τον αλγόριθμο έτσι ώστε να υπολογίζει το ποσό που θα έχεις για 5 διαφορετικά ποσά που έχει κρατήσει σε ξεχωριστούς τραπεζικούς λογαριασμούς. Να βρεθεί και το τελικό ποσό που θα έχεις από όλους αυτούς τους λογαριασμούς.

ΔΣ8. Έστω ότι έχεις να επεκτείνεις το πρόβλημα της δενδροφύτευσης που δόθηκε στις δραστηριότητες για την τάξη (ΔΤ9). Να επεκτείνεις τον αλγόριθμο έτσι ώστε να διαβάζεις ένα σύνολο από 100 τιμές που αφορούν το είδος του δένδρου και να υπολογίζεις πόσα από τα δένδρα αυτά θα φυτευτούν στη Μακεδονία, πόσα στη Θράκη και πόσα στην Πελοπόννησο.

ΔΣ9. Έστω ότι θέλεις να οργανώσεις μία εκδήλωση για την παγκόσμια ημέρα περιβάλλοντος και έχεις τη χωρητικότητα (σε αριθμό ατόμων) και τις τιμές που θα κοστίσει η ενοικίαση χώρου από 3 διαφορετικούς χώρους στους οποίους μπορεί να γίνει η εκδήλωση. Επιπλέον έχεις προσφορές από 5 διαφορετικούς χορηγούς που διαθέτουν χρήματα για την υποστήριξη της εκδήλωσης. Να γραφεί ένας αλγόριθμος που θα υπολογίζει πόσοι χορηγοί μπορούν να καλύψουν το κόστος της αίθουσας με τη δυνατή μεγαλύτερη χωρητικότητα.

2.5. Τεστ αυτοαξιολόγησης

1. Δίνονται οι παρακάτω ομάδες προτάσεων. Σε καθεμία από αυτές, να κάνετε τις απαραίτητες διορθώσεις ώστε να ισχύουν οι προτάσεις:

1. Η αναπαράσταση αλγορίθμου με ελεύθερο κείμενο (free text) αποτελεί τον πιο καλά δομημένο τρόπο παρουσίασης αλγορίθμου.
2. Τα διαγράμματα ροής (flow charts) αποτελούν έναν ακολουθιακό τρόπο παρουσίασης ενός αλγορίθμου με χρήση βημάτων.
3. Η κωδικοποίηση (coding) ενός αλγορίθμου γίνεται με ένα πρόγραμμα που όταν εκτελεσθεί μπορεί και να μη δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

2. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

4. Η _____ δομή (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών.
5. Η δομή της _____ χρησιμοποιείται όταν υπάρχει αναγκαιότητα απόφασης μεταξύ ενός συνόλου περιπτώσεων.
6. Η _____ ενός αλγορίθμου γίνεται με ένα πρόγραμμα που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.
7. Τα _____ αποτελούν ένα γραφικό τρόπο παρουσίασης ενός αλγορίθμου.
8. Οι _____ διαδικασίες συνδυάζουν και χρησιμοποιούν περισσότερες από μία περιπτώσεις αλγοριθμικών συνιστωσών.

3. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

9. Η αλγοριθμική υποστήριξη βοηθά στην επίλυση προβλημάτων.
10. Οι αλγοριθμικές δομές αποτελούνται από ένα ενιαίο κομμάτι και διαφέρουν μόνο στα στοιχεία εισόδου.
11. Για τον υπολογισμό ενός αθροίσματος ακεραίων μπορώ να χρησιμοποιήσω τη δομή της επιλογής.
12. Οι διαδικασίες πολλαπλών επιλογών χρησιμοποιούνται για τις διαφορετικές ενέργειες που πρέπει να γίνουν με βάση τον αριθμό των διακριτών ακεραίων τιμών μίας μεταβλητής.

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

13. Τα χρησιμοποιούμενα γεωμετρικά σχήματα για την αναπαράσταση των διαγραμμάτων ροής είναι τα εξής:
Α) έλλειψη
Β) ρόμβος

- Γ) ορθογώνιο
 - Δ) κύκλος
14. Ποια από τα παρακάτω κριτήρια πρέπει να ικανοποιεί απαραίτητα ένας αλγόριθμος:
- Α) είσοδος/έξοδος
 - Β) ύπαρξη βρόχου ή συνθήκης επανάληψης
 - Γ) μη-περατότητα
 - Δ) καθοριστικότητα
 - Ε) αποτελεσματικότητα

5. Βάλε έναν κύκλο στα σωστά:

15. Οι αλγοριθμικές συνιστώσες περιλαμβάνουν:
- Α) δομή επιλογής
 - Β) δομή ακολουθίας
 - Γ) δομή δεδομένων
 - Δ) δομή εισόδου
16. Ο πολλαπλασιασμός αλά ρωσικά περιλαμβάνει:
- Α) πολλαπλασιασμό επί 4
 - Β) πολλαπλασιασμό επί 2
 - Γ) διαίρεση διά 4
 - Δ) διαίρεση διά 2
17. Η Πληροφορική είναι η επιστήμη που μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές:
- Α) Υλικού
 - Β) Θεωρητική
 - Γ) Πιθανολογική
 - Δ) Αναλυτική

Δομές Δεδομένων και Αλγόριθμοι

3.1. Προσδοκώμενα αποτελέσματα

Στο τέλος αυτού του κεφαλαίου προσδοκάται ότι θα έχεις συνειδητοποιήσει τη σπουδαιότητα των δεδομένων για την επίλυση ενός προβλήματος. Θα έχεις ενστερνισθεί τη θεώρηση ότι οι αλγόριθμοι και οι δομές δεδομένων αποτελούν αδιάσπαστη ενότητα. Θα μπορείς να χειρίζεσαι με ευχέρεια προβλήματα σχετικά με εργασίες με πίνακες. Ακόμα θα μπορείς να κάνεις μια περιληπτική αναφορά σε άλλες δομές δεδομένων (στοίβα, ουρά, λίστα, δένδρο). Τέλος εκτιμάται ότι θα έχεις κατανοήσει τη λειτουργία της αναδρομής. Έτσι έρχεσαι σε επαφή με ένα πανόραμα δομών και αλγορίθμων, που αποτελεί ένα ικανοποιητικό σύνολο εργαλείων για την επίλυση πρακτικών προβλημάτων.

3.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1. Υπολογισμός μέγιστου ποσού

Σε μία εταιρεία εργάζονται 200 υπάλληλοι και είναι γνωστός ο μισθός του καθενός. Να χρησιμοποιηθεί η δομή του πίνακα για να αποθηκεύονται οι μισθοί των υπαλλήλων και να βρεθεί ο κατάλληλος αλγόριθμος υπολογισμού του μεγαλύτερου μισθού.

```

Αλγόριθμος Μεγαλύτερος_Μισθός
Διάβασε MIS[1]
MAX ← MIS[1]
Για i από 2 μέχρι 200
    Διάβασε MIS[i]
    Αν MIS[i] > MAX τότε
        MAX ← MIS[i]
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // MAX//
Τέλος Μεγαλύτερος_Μισθός

```

Ένα παρόμοιο παράδειγμα είχε χρησιμοποιηθεί στο προηγούμενο κεφάλαιο (παράδειγμα 3). Η δομή του πίνακα χρησιμοποιείται ώστε οι μισθοί των υπαλλήλων να αποθηκεύονται στις θέσεις του πίνακα και να μπορούν να χρησιμοποιηθούν και στη συνέχεια. Αντίθετα, στο προηγούμενο κεφάλαιο δεν αποθηκεύονταν κάπου οι τιμές των θερμοκρασιών αλλά απλά χρησιμοποιούνταν στις συγκρίσεις για την αναγνώριση της μέγιστης θερμοκρασίας.

ΠΑΡΑΔΕΙΓΜΑ 2. Υπολογισμός αριθμού συνδυασμών

Είναι γνωστό από τα μαθηματικά ότι ο αριθμός των συνδυασμών των n πραγμάτων ανά k δίνεται από τον τύπο:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Να προταθεί ένας αλγόριθμος για τον υπολογισμό του αριθμού των συνδυασμών αυτών. Ένας απλός τρόπος είναι να εφαρμοσθεί ο προηγούμενος μαθηματικός τύπος. Έτσι, δεδομένων των παραλλαγών του αλγορίθμου για τον υπολογισμό του $n!$, προκύπτει ο επόμενος αλγόριθμος που καλεί κάποια από αυτές τις παραλλαγές.

```

Αλγόριθμος Συνδυασμός
Διάβασε n
Διάβασε k
a ← Παραγοντικό(n)
b ← Παραγοντικό(k)
c ← Παραγοντικό(n-k)
combination ← a / (b*c)
Γράψε combination
Τέλος Συνδυασμός

```

Αν και ο αλγόριθμος αυτός είναι ιδιαίτερα απλός στην κατανόηση και στον προγραμματισμό του, εντούτοις δεν είναι ο καλύτερος δυνατός από την άποψη της αποτελεσματικότητας γιατί εκτελεί περιττούς πολλαπλασιασμούς. Αυτό φαίνεται ανάγλυφα αν θεωρήσουμε και πάλι

το μαθηματικό τύπο του αριθμού των συνδυασμών. Στο κλάσμα του τύπου αυτού μπορεί να γίνει κάποια απλοποίηση μεταξύ αριθμητή και παρονομαστή. Για παράδειγμα, για τον υπολογισμό του αριθμού των συνδυασμών των 10 πραγμάτων ανά 5, με τον προηγούμενο τρόπο θα εκτελέσουμε 9 πολλαπλασιασμούς για τον υπολογισμό του αριθμητή (10!), και 4+4 πολλαπλασιασμούς για τον υπολογισμό του παρονομαστή (δύο φορές το 5!). Ενώ εύκολα προκύπτει ότι:

$$\frac{10!}{5!5!} = \frac{10 \cdot 9 \cdot 8 \cdot 7 \cdot 6}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}$$

όπου εκτελούνται από τέσσερις πολλαπλασιασμοί σε αριθμητή και παρονομαστή. Ο αντίστοιχος αλγόριθμος έχει ως εξής:

```
Αλγόριθμος Συνδυασμός2
Διάβασε n
Διάβασε k
a ← 1
Για i από n μέχρι n-k+1 με_βήμα -1
    a ← a*i
Τέλος_επανάληψης
b ← Παράγοντικό(k)
combination ← a/b
Εκτύπωσε combination
Τέλος Συνδυασμός2
```

ΠΑΡΑΔΕΙΓΜΑ 3. Υπολογισμός μέσου όρου

Σε ένα Λύκειο υπάρχουν τρία τμήματα για την Γ΄ Λυκείου και κάθε τμήμα έχει 35 μαθητές. Να γραφεί ένας αλγόριθμος που θα διαβάζει το μέσο όρο βαθμολογίας κάθε μαθητή και θα υπολογίζει το γενικό μέσο όρο βαθμολογίας για όλη την τάξη της Γ΄ Λυκείου.

Ο αλγόριθμος που ακολουθεί υπολογίζει τον παραπάνω μέσο όρο με χρήση της δομής του πίνακα.

```
Αλγόριθμος Μέσος_Όρος
S ← 0
Για i από 1 μέχρι 105
    Διάβασε M[i]
    S ← S+M[i]
Τέλος_επανάληψης
MO ← S/105
Αποτελέσματα // MO //
Τέλος Μέσος_Όρος
```

ΠΑΡΑΔΕΙΓΜΑ 4. Χρήση δισδιάστατων πινάκων

Έστω ότι δίνονται δύο δισδιάστατοι πίνακες A και B διαστάσεων 5x5 ο καθένας. Να γραφεί ένας αλγόριθμος που θα διαβάζει τα στοιχεία των πινάκων και θα υπολογίζει το άθροισμα των πινάκων, το οποίο θα αποθηκεύεται σε ένα νέο πίνακα.

```

Αλγόριθμος Αθροισμα_Πινάκων
Για i από 1 μέχρι 5
  Για j από 1 μέχρι 5
    Διάβασε A[i,j], B[i,j]
    C[i,j] ← A[i,j] + B[i,j]
  Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτελέσματα // C //
Τέλος Αθροισμα_Πινάκων

```

ΠΑΡΑΔΕΙΓΜΑ 5. Αραιοί πίνακες

Ένας πίνακας λέγεται **αραιός** (sparse) αν ένα μεγάλο ποσοστό των στοιχείων του έχουν μηδενική τιμή. Δεν υπάρχει ακριβές ποσοστό σε σχέση με τον αριθμό των μηδενικών στοιχείων, επάνω από το οποίο ένας πίνακας χαρακτηρίζεται ως αραιός. Αρκεί όμως, για παράδειγμα, να πούμε ότι με περισσότερο από 80% μηδενικά ένας πίνακας χαρακτηρίζεται ως αραιός.

Αραιοί πίνακες συναντώνται συχνά σε μεγάλα επιστημονικά προβλήματα (επίλυση εξισώσεων κ.λπ.). Το πρόβλημα με τη διαχείριση των αραιών πινάκων είναι ότι δαπανάται πολύ χώρος για την αποθήκευση μηδενικών. Άρα πρέπει να βρεθεί ένας οικονομικός τρόπος αποθήκευσης των αραιών πινάκων. Στην πράξη έχουν προταθεί αρκετοί τρόποι. Ένας από αυτούς τους τρόπους περιγράφεται στη συνέχεια. Έστω, λοιπόν, ότι δίνεται ο επόμενος πίνακας, που θέλουμε να τον διαχειρισθούμε ως αραιό.

0	7	0	0	0
1	2	0	0	-3
0	0	4	0	0
12	0	0	0	0

Αντί να αποθηκεύσουμε αυτόν το δισδιάστατο πίνακα 4x5, θα θεωρήσουμε ένα μονοδιάστατο πίνακα όπου θα τοποθετήσουμε μόνο τα μη μηδενικά στοιχεία, για τα οποία όμως χρειαζόμαστε τα στοιχεία των αντίστοιχων γραμμών και στηλών. Έτσι καταλήγουμε κάθε μη μηδενικό στοιχείο να αντιπροσωπεύεται από μία τριάδα στοιχείων, δηλαδή <γραμμή,στήλη,τιμή>. Για το λόγο αυτό δημιουργούμε ένα μονοδιάστατο πίνακα 18 θέσεων για τα 6 μη μηδενικά στοιχεία του αρχικού πίνακα. Ο νέος πίνακας έχει τη μορφή:

1	2	7	2	1	1	2	2	2	2	5	-3	3	3	4	4	1	12
---	---	---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	----

Πλέον, το πρόβλημα έγκειται στην αναγνώριση της τιμής μίας θέσης του παλαιού πίνακα, δεδομένου ότι ο πίνακας είναι αποθηκευμένος με τη νέα του μορφή. Ο επόμενος αλγόριθμος "Αραιός" επιστρέφει την τιμή του στοιχείου που βρίσκεται στη θέση <γραμμή l, στήλη m> του αρχικού πίνακα επεξεργαζόμενος τη νέα μορφή του πίνακα που αποτελείται από 3n θέσεις, όπου n ο αριθμός των μη μηδενικών στοιχείων.

```

Αλγόριθμος Αραιός
Δεδομένα // sparse, n //
flag ← 0
k ← 0
Οσο flag=0 επανάλαβε
    i ← sparse[3*k+1]
    j ← sparse[3*k+2]
    Αν i=L και j=M τότε
        result ← sparse[3*k+3]
        flag ← 1
    αλλιώς_αν i > L ή (i=L και j > M) τότε
        result ← 0
        flag ← 1
    αλλιώς
        k ← k+1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // result //
Τέλος Αραιός

```

3.3. Δραστηριότητες - ασκήσεις

- ΔΤ1.** Σε μία κατασκήνωση υπάρχουν 300 παιδιά και καθένα από αυτά έχει μοναδικό αριθμό από το 1 έως και το 300 που του αντιστοιχεί. Για κάθε παιδί είναι γνωστή η ηλικία του. Να χρησιμοποιηθεί η δομή του πίνακα για να αποθηκεύονται οι ηλικίες των παιδιών και να βρεθεί ο κατάλληλος αλγόριθμος υπολογισμού του μικρότερου και μεγαλύτερου σε ηλικία παιδιού και να εκτυπώνεται τόσο η ηλικία όσο και ο κωδικός του μικρότερου και μεγαλύτερου παιδιού.
- ΔΤ2.** Ο αλγόριθμος της φυσαλίδας όπως διατυπώθηκε στην παράγραφο 3.7 έχει το μειονέκτημα ότι δεν είναι αρκετά “έξυπνος” ώστε να διαπιστώνει στην αρχή ή στο μέσο της διαδικασίας αν ο πίνακας είναι ταξινομημένος. Να σχεδιασθεί μία παραλλαγή του αλγορίθμου αυτού που να σταματά όταν διαπιστωθεί ότι τα στοιχεία του πίνακα είναι ήδη ταξινομημένα.
- Υπόδειξη:** Να χρησιμοποιήσετε μία βοηθητική μεταβλητή που να ελέγχει το τέλος κάθε επανάληψης του εξωτερικού βρόχου (“Για i από 2 μέχρι n”) αν για την τρέχουσα τιμή του i έγιναν αντιμεταθέσεις στοιχείων.
- ΔΤ3.** Να δοθούν οι αλγόριθμοι Ώθηση (Push) και Απώθηση (Pop) που αντίστοιχα εκτελούν τις προφανείς λειτουργίες σε μία στοίβα. Να δοθεί ένα παράδειγμα στο οποίο να χρησιμοποιείται μία στοίβα από ακέραιους. Η στοίβα αντιπροσωπεύεται από έναν πίνακα μέχρι 100 θέσεων.
- ΔΤ4.** Να δοθούν οι αλγόριθμοι Εισαγωγή_σε_Ουρά (Enqueue) και Εξαγωγή_από_Ουρά (Dequeue) που αντίστοιχα εκτελούν τις προφανείς λειτουργίες σε μία ουρά. Να δοθεί ένα παράδειγμα στο οποίο να χρησιμοποιείται μία ουρά από ακέραιους. Η ουρά αντιπροσωπεύεται από έναν πίνακα μέχρι 100 θέσεων.

ΔΤ5. Έστω ότι η τάξη σας θα συμμετάσχει στην ημερήσια εθελοντική αιμοδοσία που πραγματοποιεί ο Δήμος της πόλης σας. Είναι γνωστό το επίθετο κάθε μαθητή και όλοι οι μαθητές θα συμμετάσχουν στην αιμοδοσία. Να γραφεί αλγόριθμος για τη δημιουργία ουράς των μαθητών έξω από το Κέντρο αιμοδοσίας με δεδομένο ότι η ουρά θα δημιουργηθεί με βάση την αλφαβητική σειρά των επιθέτων των μαθητών.

ΔΤ6. Μία οικολογική οργάνωση διαθέτει στοιχεία για το ποσοστό δασών για 50 διαφορετικές χώρες. Χρειάζεται να πάρει απόφαση για να διοργανώσει μία εκδήλωση διαμαρτυρίας στις 10 χώρες που έχουν το χαμηλότερο ποσοστό δασών. Να δοθεί αλγόριθμος που θα ταξινομεί τα ποσοστά δασών των χωρών με χρήση της μεθόδου της ευθείας ανταλλαγής και θα εκτυπώνει τις 10 χώρες στις οποίες θα διοργανωθούν οι εκδηλώσεις.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Δίνεται ο παρακάτω πίνακας από αντιστοιχίσεις νομισμάτων διαφόρων κρατών:

Νόμισμα Χώρας	Αγορά	Πώληση
Δολάριο ΗΠΑ	1,01	1,03
Δολάριο Καναδά	0,65	0,66
Λίρα Αγγλίας	1,57	1,59
Γεν Ιαπωνίας	0,01	0,012
Φράγκο Ελβετίας	0,67	0,68

Να γραφεί ένας αλγόριθμος που θα κάνει μετατροπές ενός ποσού από τα ξένα νομίσματα σε ευρώ και από ευρώ στο αντίστοιχο ξένο νόμισμα.

ΔΣ2. Κατά τη διάρκεια ενός πρωταθλήματος μπάσκετ καταγράφεται ο αριθμός των πόντων που έχουν βάλει 5 παίκτες σε 5 διαφορετικά παιχνίδια. Να γραφεί αλγόριθμος που θα σε βοηθήσει να κρατήσεις σε ένα δισδιάστατο πίνακα αυτά τα στοιχεία και στη συνέχεια να υπολογίσεις τον παίκτη που έχει πετύχει το μεγαλύτερο αριθμό πόντων από όλα τα παιχνίδια.

ΔΣ3. Έστω ότι θέλουμε να διατάξουμε τους μαθητές μίας τάξης κατά φθίνουσα σειρά ύψους. Η τεχνική που θα ακολουθήσουμε είναι η εξής: Αρχικά, τοποθετούμε τους μαθητές σε μία τυχαία σειρά. Κατόπιν συγκρίνουμε το δεύτερο με τον πρώτο και αν χρειασθεί τους αντιμεταθέτουμε ώστε πρώτος να είναι ο ψηλότερος. Στη συνέχεια θεωρούμε τον τρίτο και τον τοποθετούμε στη σωστή σειρά σε σχέση με τον πρώτο και το δεύτερο. Κατ' αυτόν τον τρόπο συνεχίζουμε μέχρι να τοποθετήσουμε στη σωστή σειρά όλους τους μαθητές. Να σχεδιασθεί ένας αλγόριθμος που να υλοποιεί αυτή τη μέθοδο ταξινόμησης.

ΔΣ4. Ένας μαθητής έχει μία συλλογή από δίσκους CD και για κάθε CD έχει καταγράψει στον υπολογιστή τον τίτλο και τη χρονιά έκδοσής του. Να ταξινομηθούν τα CD με βάση τη χρονιά τους και να υπολογισθεί ο αριθμός των CD που έχει ο μαθητής με χρονολογία έκδοσης πριν από το 1995.

ΔΣ5. Ας υποθέσουμε ότι έχετε αναλάβει να μοιράσετε ένα σύνολο από βιβλία στους συμμαθητές σας. Αν ορίσετε μία ημέρα για το μοίρασμα των βιβλίων και οι συμμαθητές σας φθάνουν ο ένας μετά τον άλλο φτιάχνοντας μία ουρά, πώς θα ρυθμίσετε την είσοδο και την έξοδό τους από την ουρά; Να δώσετε το σχετικό αλγόριθμο εισαγωγής και εξαγωγής από την ουρά.

3.4. Τεστ αυτοαξιολόγησης

1. *Δίνονται οι παρακάτω ομάδες προτάσεων. Σε καθεμία από αυτές, να κάνετε τις απαραίτητες διορθώσεις ώστε να ισχύουν οι προτάσεις:*

1. Δομή Δεδομένων είναι ένα σύνολο δεδομένων τα οποία δεν υφίστανται επεξεργασία από λειτουργίες, που καλούνται από το υπόλοιπο πρόγραμμα.
2. Οι δυναμικές δομές δεδομένων αποθηκεύονται σε συνεχόμενες θέσεις μνήμης, αλλά στηρίζονται στην τεχνική της λεγόμενης δυναμικής παραχώρησης μνήμης (dynamic memory allocation).
3. Οι πίνακες χρησιμεύουν για την αποθήκευση και διαχείριση τριών βασικών δομών: της αναδρομής, της στοίβας και της ουράς.

2. *Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:*

4. Η _____ είναι η πράξη κατά την οποία όλοι οι κόμβοι ή μερικοί από τους κόμβους μίας δομής αντιγράφονται σε μία άλλη δομή.
5. Η _____ είναι η πράξη κατά την οποία δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
6. Ο _____ αποτελεί την αντίστροφη πράξη της συγχώνευσης.
7. Δύο είναι οι κύριες λειτουργίες σε μία στοίβα: η _____ στοιχείου στην κορυφή της στοίβας και η _____ στοιχείου από τη στοίβα.
8. Δύο είναι οι κύριες λειτουργίες σε μία ουρά: η _____ στοιχείου στο πίσω άκρο της ουράς και η _____ στοιχείου από το εμπρός άκρο της ουράς.
9. Η τακτοποίηση των κόμβων μίας δομής με μία ιδιαίτερη σειρά είναι μία ιδιαίτερη σημαντική λειτουργία που ονομάζεται _____.

3. *Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:*

10. Οι δομές δεδομένων διακρίνονται σε δύο μεγάλες κατηγορίες: τις στατικές και τις δυναμικές.

11. Δύο είναι οι κύριες λειτουργίες που εκτελούνται σε μία ουρά: η εισαγωγή και η διαγραφή.
12. Η τακτοποίηση των κόμβων μίας δομής με μία ιδιαίτερη σειρά είναι μία ιδιαίτερη σημαντική λειτουργία που ονομάζεται εξαγωγή.
13. Η μέθοδος της ταξινόμησης ευθείας ανταλλαγής βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία.

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

14. Οι βασικές λειτουργίες (ή αλλιώς πράξεις) επί των δομών δεδομένων είναι οι ακόλουθες:

A) Προσπέλαση	Γ) Εισαγωγή	E) Αναζήτηση
B) Ανάγνωση	Δ) Διαγραφή	Z) Εκτύπωση
		H) Ταξινόμηση
15. Η σειριακή μέθοδος αναζήτησης δικαιολογεί τη χρήση της μόνο σε περιπτώσεις όπου:

A) ο πίνακας είναι αταξινόμητος
B) ο πίνακας αποτελείται από ακέραιους
Γ) ο πίνακας είναι μικρού μεγέθους
Δ) ο πίνακας δεν είναι δισδιάστατος
E) η αναζήτηση σε ένα συγκεκριμένο πίνακα γίνεται σπάνια
Z) η αναζήτηση γίνεται με βάση την τιμή δευτερεύοντος κλειδιού

5. Βάλε έναν κύκλο στα σωστά:

16. Οι πίνακες χρησιμεύουν για την αποθήκευση και διαχείριση των παρακάτω δομών δεδομένων:

A) ουράς	Γ) στοίβας
B) λίστας	Δ) επιλογής

Τεχνικές Σχεδίασης Αλγορίθμων

4.1. Προσδοκώμενα αποτελέσματα

Ολοκληρώνοντας αυτό το κεφάλαιο προσδοκάται πως θα έχεις λάβει εκείνες τις γνώσεις, ώστε να τεκμηριώνεις την αναγκαιότητα ανάλυσης των προβλημάτων και σχεδίασης των κατάλληλων αλγορίθμων. Θα μπορείς να διατυπώνεις σύγχρονες τεχνικές σχεδίασης αλγορίθμων και να περιγράφεις την ακολουθία βημάτων για την ανάλυση των αλγορίθμων. Ακόμα θα μπορείς να περιγράφεις τις κυριότερες προσεγγίσεις επίλυσης και ανάλυσης προβλημάτων. Τέλος, θα έχεις αποκτήσει εκείνες τις δεξιότητες ώστε να μπορείς να επιλύεις προβλήματα με χρήση των κυριοτέρων προσεγγίσεων.

4.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1. Ταξινόμηση με επιλογή

Σε μία τηλεφωνική εταιρεία χρειάζεται να γίνει ταξινόμηση σε αύξουσα σειρά των αριθμών τηλεφώνων με βάση την εξής παρατήρηση: «Από τους αριθμούς τηλεφώνων που δεν έχουν ταξινομηθεί στη σωστή σειρά, να βρεθεί ο μικρότερος αριθμός και να τοποθετηθεί στη σειρά των αριθμών που έχουν ήδη ταξινομηθεί». Να καταγραφεί ο σχετικός αλγόριθμος για την ταξινόμηση 1000 αριθμών τηλεφώνων.



Για να υλοποιήσεις τον αλγόριθμο, πρέπει να υπάρξει κάποια απόφαση σχετικά με τις δομές δεδομένων που θα χρειασθείς. Είναι δεδομένο ότι θα έχεις 1000 αριθμούς τηλεφώνων που πρέπει να βάλεις σε σωστή αύξουσα σειρά. Στις περιπτώσεις προβλημάτων όπου είναι δεδομένος ο συνολικός αριθμός από «αντικείμενα» που θα έχει το πρόβλημα χρησιμοποιούμε τη δομή του πίνακα, έτσι ώστε κάθε αντικείμενο να έχει τη δική του θέση. Επομένως γίνεται αρχικά η ανάθεση των αριθμών τηλεφώνων σε καθεμία από τις θέσεις του πίνακα, με τον τρόπο που έχει περιγραφεί σε προηγούμενα κεφάλαια. Ο παρακάτω αλγόριθμος υποθέτει ότι ο πίνακας THL έχει πάρει τους αριθμούς τηλεφώνων σε τυχαία σειρά και επομένως δεν είναι ταξινομημένος.

```

Αλγόριθμος Αριθμοί_Τηλεφώνων
Δεδομένα // THL//
Για i από 1 μέχρι 1000
    j ← i
    Για k από i+1 μέχρι 1000
        Αν THL[k] < THL[j] τότε
            j ← k
    Τέλος_αν
Τέλος_επανάληψης
Αντιμετάθεση THL[i], THL[j]
Τέλος_επανάληψης
Αποτελέσματα // THL //
Τέλος Αριθμοί_Τηλεφώνων

```



Ο παραπάνω αλγόριθμος αποτελεί μία απλή πρόταση για την ταξινόμηση στοιχείων και είναι γνωστός ως αλγόριθμος **ταξινόμησης με επιλογή** (selection sort). Η ονομασία του οφείλεται στη λογική που χρησιμοποιεί για την ταξινόμηση, η οποία βασίζεται στην επιλογή του μικρότερου στοιχείου από αυτά που δεν έχουν ταξινομηθεί σε κάθε βήμα.

ΠΑΡΑΔΕΙΓΜΑ 2. Ανεπιτυχής δυαδική αναζήτηση

Σε ένα παραθεριστικό κέντρο υπάρχουν πολλά καταστήματα και εστιατόρια. Ένας επιχειρηματίας θέλει να ανοίξει ένα κατάστημα και θέλει να του δώσει το όνομα «Ανοιξη». Πρέπει πρώτα να ερευνήσει εάν αυτό το όνομα έχει ήδη δοθεί σε κάποιο άλλο κατάστημα. Έστω ότι όλα τα ονόματα καταστημάτων του παραθεριστικού κέντρου έχουν καταγραφεί σε έναν πίνακα 50 θέσεων. Να προτείνετε τον κατάλληλο αλγόριθμο που θα δώσει την απάντηση στον επιχειρηματία για το εάν μπορεί να ανοίξει το κατάστημα με αυτό το όνομα ή όχι.



Για να βρεις τον κατάλληλο αλγόριθμο, πρέπει να χρησιμοποιήσεις την ιδέα της δυαδικής αναζήτησης που έχει παρουσιασθεί με παράδειγμα στο βιβλίο σου (Κεφάλαιο 4). Η διαφοροποίηση σε εκείνο τον αλγόριθμο έχει να κάνει με το ότι εδώ χρειάζεται να υπάρξει κάποια μέριμνα για το εάν το στοιχείο που αναζητούμε βρέθηκε στον πίνακα

ή όχι. Στις περιπτώσεις αυτές συνηθίζεται να υπάρχει κάποια μεταβλητή που αναλαμβάνει αυτόν το ρόλο και ενημερώνεται με την κατάλληλη τιμή. Συχνά οι μεταβλητές αυτές εκφράζουν δύο καταστάσεις (π.χ. εδώ έχουμε βρέθηκε / δεν βρέθηκε) και για αυτό χαρακτηρίζονται ως δυαδικές μεταβλητές «σημαίες» (boolean flags). Η μεταβλητή found στον παρακάτω αλγόριθμο έχει αυτόν το ρόλο. Εάν η τιμή της found είναι 0, δεν έχει βρεθεί το όνομα που ψάχνεις, αν η τιμή της είναι 1, έχει βρεθεί και επομένως το κατάστημα δεν μπορεί να πάρει το όνομα που έχεις δώσει. Επιπλέον, στον αλγόριθμο που ακολουθεί χρησιμοποιείται ο πίνακας KAT, στοιχεία του οποίου είναι τα ονόματα των 50 καταστημάτων που έχουν διαβασθεί όπως έχει περιγραφεί σε προηγούμενο κεφάλαιο και ONOMA “Ανοιξη” είναι το όνομα που αναζητούμε για το κατάστημα.

```
Αλγόριθμος Δυαδική_αναζήτηση
Δεδομένα // ONOMA, KAT //
low ← 0
high ← 50
found ← 0
όσο low ≤ high επανάλαβε
    mid ← (low + high)/2
    Αν KAT[mid] < ONOMA τότε
        low ← mid+1
    αλλιώς_αν KAT[mid] > ONOMA τότε
        high ← mid-1
    αλλιώς
        found ← 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // found //
Τέλος Δυαδική_αναζήτηση
```

ΠΑΡΑΔΕΙΓΜΑ 3. Εύρεση δύο μικρότερων αριθμών

Σε ένα Τμήμα μιας επιχείρησης χρειάζεται να βρεθούν οι δύο χαμηλότεροι μισθοί με δεδομένο ότι το Τμήμα απασχολεί 50 υπαλλήλους και οι μισθοί τους αποθηκεύονται σε κάποιον πίνακα. Να γραφεί ένας αλγόριθμος που θα υπολογίζει τους δύο μικρότερους μισθούς με δεδομένο τον πίνακα των μισθών των υπαλλήλων.

```
Αλγόριθμος Δύο_Μικρότεροι
Δεδομένα // M //
low1 ← M[1]
pos ← 1
Για i από 2 μέχρι 50
    Αν M[i] < low1 τότε
        low1 ← M[i]
        pos ← i
    Τέλος_αν
Τέλος_επανάληψης
Αν pos <> 1 τότε
    low2 ← M[1]
αλλιώς
    low2 ← M[2]
```



Το πρόβλημα της ανεύρεσης των δύο μικρότερων στοιχείων ενός πίνακα επιδέχεται διάφορες τεχνικές και τρόπους σχεδίασης. Ο αλγόριθμος που παρουσιάζεται στη συνέχεια είναι αρκετά απλός και δεν έχει καλή αποδοτικότητα.

```

Τέλος_αν
Για i από 2 μέχρι 50
    Αν i <> pos και M[i] < low2 τότε
        low2 ← M[i]
Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // low1 , low2 //
Τέλος Δύο_Μικρότεροι

```

4.3. Δραστηριότητες - ασκήσεις

- ΔΤ1.** Να παρακολουθήσετε την πορεία των αλγορίθμων που προτείνεται στο Παράδειγμα 3 για παραδείγματα πινάκων κάποιων θέσεων. Να συζητήσετε την πιθανότητα να προταθεί κάποια βελτίωση και διαφορετικότητα στην τεχνική σχεδίασης του προτεινόμενου αλγορίθμου.
- ΔΤ2.** Έστω ότι ο κατάλογος των μουσείων της πόλης σας υπάρχει αποθηκευμένος σε έναν πίνακα ο οποίος περιέχει το όνομα κάθε μουσείου. Έστω ότι κάποιος επισκέπτης θέλει να μάθει εάν κάποιο συγκεκριμένο μουσείο (π.χ. Λαογραφικό) υπάρχει στην πόλη σας. Να προτείνετε δύο τρόπους για την αναζήτηση ενός συγκεκριμένου μουσείου από αυτόν τον πίνακα και να συζητήσετε στην τάξη τη διαφορά και τον τρόπο λειτουργίας κάθε αλγορίθμου.
- ΔΤ3.** Να συζητηθεί και να αναλυθεί ο αλγόριθμος που υλοποιεί τη δημιουργία ενός μαγικού τετραγώνου όπως αυτό περιγράφεται στη συνέχεια:
- Ένα μαγικό τετράγωνο είναι ένας $n \times n$ πίνακας από ακέραιους από το 1 μέχρι το n^2 που έχει κατασκευασθεί έτσι ώστε το άθροισμα κάθε γραμμής, κάθε στήλης και κάθε διαγωνίου να είναι το ίδιο, όπως φαίνεται και στο παρακάτω σχήμα:

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

Στο παραπάνω σχήμα παρουσιάζεται ένα μαγικό τετράγωνο με $n=5$, στο οποίο κάθε γραμμή ή στήλη ή διαγώνιος έχει άθροισμα 65. Έχει προταθεί κάποιος συγκεκριμένος κανόνας για τη δημιουργία ενός μαγικού τετραγώνου για n περιττό αριθμό. Ο κανόνας αυτός συνοψίζεται στα εξής:

«Ξεκινούμε τοποθετώντας τον αριθμό 1 στη μεσαία θέση της πρώτης γραμμής. Στη συνέχεια προχωρούμε αναθέτοντας τους αριθμούς 2, 3, 4, ... κ.λπ. μετακινούμενοι συνεχώς προς τα επάνω και αριστερά μέχρι να γεμίσει το μαγικό τετράγωνο. Όταν κάποια μετακίνηση προς τα επάνω ή προς τα αριστερά μας οδηγεί εκτός των ορίων του τετραγώνου, πηγαίνουμε στο αντι-διαμετρικό άκρο της γραμμής ή της στήλης

στην οποία βρεθήκαμε. Επίσης, αν η μετακίνηση μας οδηγεί σε κατειλημμένη θέση, τότε επιλέγεται η θέση κάτω από αυτήν όπου έγινε η τελευταία ανάθεση».

Ο παρακάτω αλγόριθμος υλοποιεί αυτόν τον κανόνα χρησιμοποιώντας τη δομή ενός δισδιάστατου πίνακα (square) για να κρατηθούν οι τιμές των στοιχείων του μαγικού τετραγώνου.

Αλγόριθμος Μαγικό_τετράγωνο

Δεδομένα // n //

Για i από 1 μέχρι n

Για j από 1 μέχρι n
 square[i,j] ← 0

Τέλος_επανάληψης

Τέλος_επανάληψης

i←1

j←(n+1)/2

square[i,j]←1

Για key από 2 μέχρι n*n

Αν i>1 τότε
 k←i-1

αλλιώς
 k←n

Τέλος_αν

Αν j>1 τότε
 l←j-1

αλλιώς
 l←n

Τέλος_αν

Αν square[k,l]>0 τότε
 i←i+1

Αν i=n+1 τότε i←1

αλλιώς
 i←k
 j←l

Τέλος_αν
 square[i,j]←key

Τέλος_επανάληψης

Αποτελέσματα // square //

Τέλος Μαγικό_τετράγωνο

ΔΤ4. Ένα πρακτορείο ταξιδιών διοργανώνει μία εκδρομή για το γύρο του κόσμου σε 80 ημέρες. Για να κάνει το σχεδιασμό του ταξιδιού, χρειάζεται να επιλέξει κάποιες πόλεις και συγκεκριμένη διαδρομή. Να συζητήσετε στην τάξη και να καταγράψετε τα βασικά βήματα ενός αλγορίθμου που θα σχεδιάζει τη διαδρομή που θα πρέπει να ακολουθήσει ο ταξιδιώτης ξεκινώντας από μία πόλη και καταλήγοντας πάλι σε αυτήν αφού περάσει μία φορά από τις πόλεις που έχουν επιλεγεί. Είναι χρήσιμο στο σχεδιασμό της διαδρομής να παίρνει κανείς την απόφαση για τη μικρότερη δυνατή διαδρομή.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Έστω ότι έχεις να παίξεις ένα παιχνίδι όπου προσπαθείς να μαντέψεις έναν αριθμό από το 1 μέχρι το 10 που έχει γράψει κάποιος συμμαθητής σου σε ένα χαρτί. Σε κάθε προσπάθεια, ο συμμαθητής σου απαντά δηλώνοντας αν ο αριθμός του είναι μικρότερος, μεγαλύτερος ή ίσος με το δικό σου. Να γράψεις έναν αλγόριθμο που θα σε οδηγήσει με γρήγορο τρόπο στο να βρεις τον αριθμό που έγραψε ο συμμαθητής σου. Πόσα βήματα θα χρειαστείς μέχρι να βρεις τον αριθμό;

ΔΣ2. Στο προηγούμενο κεφάλαιο είχες ασχοληθεί με την ταξινόμηση των δίσκων των CD σου σε χρονολογική σειρά. Να επεκτείνεις τον αλγόριθμο έτσι ώστε να υπάρχει δυνατότητα να βρίσκεις εάν ένα CD με συγκεκριμένο τίτλο υπάρχει στη συλλογή σου ή όχι, δίνοντας τον τίτλο του (δυαδική αναζήτηση).

ΔΣ3. Έστω ότι έχεις τον παρακάτω αλγόριθμο:

```

Αλγόριθμος Εχρο
Δεδομένα // x, n //
m ← n
row ← 1
z ← x
Οσο m > 0 επανάλαβε
    Οσο ( m MOD 2 ) = 0 επανάλαβε
        m ← m/2
        z ← z * 2
    Τέλος_επανάληψης
    m ← m-1
    row ← row*z
Τέλος_επανάληψης
Αποτελέσματα // row //
Τέλος Εχρο

```

Η «πράξη» MOD έχει ως αποτέλεσμα το υπόλοιπο της ακέραιης διαίρεσης δύο αριθμών (π.χ. $14 \text{ MOD } 3 = 2$, $4 \text{ MOD } 3 = 1$), ενώ το $m/2$ αναφέρεται στον αμέσως μικρότερο ακέραιο από τον αριθμό που προκύπτει ως αποτέλεσμα της διαίρεσης (π.χ. $13/2 = 6$, $25/3 = 8$).

Να παρακολουθήσεις την πορεία του αλγορίθμου για τα εξής ζεύγη τιμών:

$x=4, n=2$

$x=2, n=4$

$x=3, n=3$

$x=5, n=2$

Με βάση την παρακολούθηση που έκανες ποιο είναι το συμπέρασμά σου για το αποτέλεσμα του παραπάνω αλγορίθμου;

ΔΣ4. Στη βιβλιοθήκη ενός σχολείου υπάρχουν πολλά βιβλία σχετικά με τη γεωγραφία και τα ταξίδια. Έστω ότι κάθε βιβλίο έχει ένα μοναδικό κωδικό και καταχωρείται σε ηλεκτρονικό υπολογιστή ο τίτλος και ο συγγραφέας κάθε βιβλίου. Να γραφεί αλγόριθμος που θα διαβάσει το όνομα ενός συγγραφέα και θα βρίσκει τον κωδικό (ή τους κωδικούς) και τον τίτλο (ή τους τίτλους) των βιβλίων αυτού του συγγραφέα που υπάρχουν στη βιβλιοθήκη.

ΔΣ5. Ένας διαγωνισμός τραγουδιού στην Ευρώπη διεξάγεται ως εξής: Γίνεται μία πρώτη ακρόαση των τραγουδιών κάθε χώρας από την Κριτική Επιτροπή η οποία δίνει κάποιους βαθμούς σε κάθε τραγούδι (από 1-100). Έστω ότι είναι γνωστοί οι βαθμοί που δόθηκαν στο τραγούδι κάθε χώρας. Να γραφεί ένας αλγόριθμος που θα επιλέγει για τη συνέχεια στη δεύτερη φάση του διαγωνισμού τις χώρες με τη μεγαλύτερη βαθμολογία κάθε φορά ώστε το άθροισμα της βαθμολογίας όλων των τραγουδιών που θα προχωρήσουν στη δεύτερη φάση να είναι μικρότερο από 1000 βαθμούς.

ΔΣ6. Να παρακολουθήσεις το πρόβλημα για το «Γύρο του κόσμου» που δόθηκε στις δραστηριότητες για την τάξη (ΔΤ4) και να κάνεις ένα σχήμα για 10 πόλεις και των μεταξύ τους αποστάσεων με δεδομένο ότι υπάρχει αεροπορική σύνδεση για κάποιες από αυτές. Στη συνέχεια να δώσεις σχηματικά τη λύση για τη μικρότερη δυνατή διαδρομή.

4.4. Τεστ αυτοαξιολόγησης

1. Δίνονται οι παρακάτω ομάδες προτάσεων. Σε καθεμία από αυτές, να κάνετε τις απαραίτητες διορθώσεις ώστε να ισχύουν οι προτάσεις:

1. Οι μέθοδοι λύσης ενός προβλήματος που προκύπτουν από την υλοποίησή του σε συγκεκριμένο υπολογιστικό σύστημα, οδηγούν στη σχεδίαση ενός αλγορίθμου που συνιστά την ακολουθία βημάτων που πρέπει να ακολουθηθούν για να επιλυθεί το πρόβλημα.
2. Κατά την επίλυση ενός προβλήματος, δεν γίνεται σύγκριση των χαρακτηριστικών και των ιδιοτήτων διαφορετικών τεχνικών σχεδίασης ενός αλγορίθμου αλλά επιλέγεται η πλέον εκτενής τεχνική.
3. Η μέθοδος του Δυναμικού Προγραμματισμού για τη σχεδίαση αλγορίθμων χρησιμοποιείται κυρίως για την επίλυση προβλημάτων υποδιαίρεσεων σε μικρότερα μεγέθη προβλημάτων και κυρίως κατά την ταξινόμηση.

2. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

4. Η _____ μέθοδος προχωρά με την παραδοχή ότι σε κάθε βήμα γίνεται επιλογή της τρέχουσας βέλτιστης επιλογής.
5. Η τεχνική της _____ εντάσσεται στην κατηγορία αντιμετώπισης προβλημάτων που είτε περιλαμβάνουν την αναζήτηση ενός συνόλου λύσεων, είτε αναζητούν τη βέλτιστη λύση υπό κάποιες προϋποθέσεις.
6. Η _____ διευκολύνει την αποδοτική ανεύρεση στοιχείου από πίνακα, υποδιαιρώντας τον πίνακα σε δύο μέρη σε κάθε βήμα και συνεχίζοντας με τον κατάλληλο από τους δύο υπο-πίνακες.
7. Δύο γνωστές τεχνικές για την ταξινόμηση είναι η _____ και η _____.

3. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

8. Γενικότερα, κάθε τεχνική σχεδίασης χρειάζεται να υποστηρίζει τα εξής:
 - να αντιμετωπίζει με το δικό της τρόπο τα δεδομένα.
 - να έχει τη δική της ακολουθία εντολών.
 - να διαθέτει τη δική της αποδοτικότητα.
9. Η γραμμική αναζήτηση διευκολύνει την αποδοτική ανεύρεση στοιχείου από πίνακα και δεν υπάρχει καλύτερος τρόπος αναζήτησης στοιχείου από πίνακα.
10. Η τεχνική του Δυναμικού Προγραμματισμού είναι πιο αποδοτική από την τεχνική Διαίρει και Βασίλευε.

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

11. Κατά την ανάλυση ενός προβλήματος θα πρέπει να δοθεί απάντηση σε καθεμία από τις επόμενες ερωτήσεις:
 - A) Ποια είναι τα δεδομένα και το μέγεθος του προβλήματος;
 - B) Ποια είναι τα περιφερειακά του συστήματος στο οποίο θα επιλυθεί ο αλγόριθμος;
 - Γ) Ποια είναι η χρησιμότητα του αλγορίθμου;
 - Δ) Πώς θα καταγραφεί η λύση σε ένα πρόβλημα; (π.χ. σε ψευδογλώσσα)
 - E) Ποιος είναι ο τρόπος υλοποίησης στο συγκεκριμένο υπολογιστικό σύστημα; (π.χ. επιλογή γλώσσας προγραμματισμού)
12. Κάθε τεχνική σχεδίασης αλγορίθμου χρειάζεται να υποστηρίζει τα εξής:
 - A) Να υλοποιείται σε συγκεκριμένη γλώσσα προγραμματισμού.
 - B) Να αντιμετωπίζει με το δικό της τρόπο τα δεδομένα.
 - Γ) Να έχει τη δική της ακολουθία εντολών.
 - Δ) Να δέχεται συγκεκριμένη είσοδο από πληκτρολόγιο.
 - E) Να διαθέτει τη δική της αποδοτικότητα.
13. Η περιγραφή της μεθόδου Διαίρει και Βασίλευε για τη σχεδίαση αλγορίθμων μπορεί να αποδοθεί με τα επόμενα βήματα:
 - A) Δίνεται το όνομα του χρήστη του αλγορίθμου.
 - B) Δίνεται για επίλυση ένα στιγμιότυπο ενός προβλήματος.
 - Γ) Το συνολικό μέγεθος του προβλήματος διαιρείται διά 2.
 - Δ) Υποδιαίρεση του στιγμιότυπου του προβλήματος σε υποστιγμιότυπα του ίδιου προβλήματος.
 - E) Δίνεται ανεξάρτητη λύση σε κάθε υπο-στιγμιότυπο.
 - Z) Συνδυάζονται όλες οι μερικές λύσεις που βρέθηκαν για τα υπο-στιγμιότυπα, έτσι ώστε να δοθεί η συνολική λύση του προβλήματος.
 - H) Εκτυπώνονται κάθε φορά πολλά είδη λύσεων.

Ανάλυση Αλγορίθμων

5.1. Προσδοκώμενα αποτελέσματα

Όταν θα έχεις ολοκληρώσει τη μελέτη αυτού του κεφαλαίου, θα έχεις κατανοήσει τις τεχνικές ανάλυσης των αλγορίθμων. Θα μπορείς να μετράς την επίδοση των αλγορίθμων με βάση την αποδοτικότητά τους. Θα είσαι σε θέση χρησιμοποιώντας τις κατάλληλες μεθόδους να ελέγχεις τη ορθότητά τους. Ή ακόμα λαμβάνοντας υπόψη σου κάποια κριτήρια, να επιλέγεις τον προτιμότερο αλγόριθμο για το πρόβλημα που καλείσαι να αντιμετωπίσεις. Τέλος, θα έχεις κατανοήσει την έννοια της πολυπλοκότητας των αλγορίθμων.

5.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

Έστω ότι έχουμε το παρακάτω πρόγραμμα υλοποίησης ενός αλγορίθμου:

```
Αλγόριθμος Ελεγχος_εκτέλεσης  
a ← 1  
b ← 2  
Για i από 1 μέχρι 100  
    a ← i  
    b ← a * i
```

Τέλος_επανάληψης
 Εκτύπωσε a
 Εκτύπωσε b
 Τέλος_Ελεγχος_εκτέλεσης

Να υπολογισθεί η επίδοσή του με βάση τον αριθμό των πράξεων που θα εκτελεστούν.

Εντολή αλγορίθμου	Αριθμός πράξεων
ανάθεση τιμών στα a και b	2
Βρόχος επανάληψης	
αρχική τιμή i	1
έλεγχος i	101
αύξηση i	100
ανάθεση τιμών στο a	100
ανάθεση τιμών στο b (2x100)	200
Εκτύπωση a,b	2
ΣΥΝΟΛΟ	506



Με δεδομένο ότι ο βρόχος του προγράμματος θα εκτελεσθεί 100 φορές προκύπτει η παραπάνω ανάλυση, η οποία αποτελεί εκτίμηση και του χρόνου εκτέλεσης του προγράμματος αλγορίθμου. Είναι χρήσιμο εδώ να καταγραφεί το μέγεθος του προβλήματος και να εκφρασθεί το σύνολο των κριτηρίων επίδοσης σε σχέση με αυτό.

ΠΑΡΑΔΕΙΓΜΑ 2

Έστω ότι έχουμε τον παρακάτω αλγόριθμο ανάγνωσης και άμεσης εκτύπωσης των στοιχείων ενός δισδιάστατου πίνακα A:

```

Αλγόριθμος Ανάγνωση_Εκτύπωση_Πίνακα
Δεδομένα // n //
Για i από 1 μέχρι n
  Για j από 1 μέχρι n
    Διάβασε A[i, j]
    Εκτύπωσε A[i, j]
  Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος Ανάγνωση_Εκτύπωση_Πίνακα
  
```

Να υπολογισθεί ο χρόνος εκτέλεσης του αλγορίθμου αυτού και να σχολιασθεί η βαρύτητα των πράξεων επανάληψης σε σχέση με την απόφαση για την πολυπλοκότητα των αλγορίθμων.



Από ό,τι παρατηρούμε υπάρχουν δύο βρόχοι επανάληψης (ένας βρόχος για κάθε διάσταση του πίνακα). Για κάθε στιγμή επανάληψης μέσα στο εσωτερικό των δύο βρόχων γίνονται δύο απλές πράξεις (ανάγνωση και εκτύπωση) μοναδιαίου κόστους η καθεμία. Επομένως η πολυπλοκότητα του παραπάνω αλγορίθμου θα εκφράζεται με $n * n * 2$, δηλαδή ο αλγόριθμος είναι τετραγωνικός. Είναι φανερό ότι οι βρόχοι επανάληψης είναι εκείνοι που καθορίζουν την επιβάρυνση στο κόστος εκτέλεσης του αλγορίθμου.

ΠΑΡΑΔΕΙΓΜΑ 3. Ανάλυση αλγορίθμου ταξινόμησης

Έστω ότι έχουμε την παρακάτω απλή μορφή για τον αλγόριθμο ταξινόμησης με ευθεία ανταλλαγή (bubblesort):

```

Αλγόριθμος Ευθεία_Ανταλλαγή
Δεδομένα // A //
Για i από 1 μέχρι n-1
    Για j από 1 μέχρι n-1
        Αν A[j+1] < A[j] τότε
            Αντιμετάθεσε A[j+1], A[j]
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
Αποτελέσματα // A //
Τέλος Ευθεία_Ανταλλαγή

```

Έστω ότι έχουμε τον πίνακα A με τα παρακάτω στοιχεία:

1	2	3	4
7	4	3	8

Να παρακολουθήσετε την πορεία του αλγορίθμου με καταγραφή της επίδοσής του που θα εκφράζεται από τον αριθμό των πράξεων που πρέπει να εκτελεσθούν.

Επίδοση Αλγορίθμου

Για να εκφρασθεί η απόδοση του αλγορίθμου, χρειάζεται να μετρηθεί ο αριθμός των συγκρίσεων και ο αριθμός των ανταλλαγών που θα πραγματοποιηθούν για την ταξινόμηση. Στη συνέχεια παρουσιάζονται αναλυτικά αυτοί οι υπολογισμοί:

Αριθμός Συγκρίσεων

$i \leftarrow 1$ (1ο Πέρασμα)

1	2	3	4	
7	4	3	8	Σύγκριση 1ου και 2ου
4	7	3	8	Σύγκριση 2ου και 3ου
4	3	7	8	Σύγκριση 3ου και 4ου

$i \leftarrow 2$ (2ο Πέρασμα)

1	2	3	4	
4	3	7	8	Σύγκριση 1ου και 2ου
3	4	7	8	Σύγκριση 2ου και 3ου
3	4	7	8	Σύγκριση 3ου και 4ου

$i \leftarrow 3$ (3ο Πέρασμα)

1	2	3	4	
3	4	7	8	Σύγκριση 1ου και 2ου
3	4	7	8	Σύγκριση 2ου και 3ου
3	4	7	8	Σύγκριση 3ου και 4ου

Όπως φαίνεται από την παρακολούθηση της πορείας του αλγορίθμου, η εκτέλεσή του περιλαμβάνει 3 συνεχόμενα «περάσματα» πάνω από τα στοιχεία του πίνακα και σε κάθε πέρασμα συγκρίνονται ζευγάρια στοιχείων. Από ό,τι φαίνεται και παραπάνω γίνονται συνολικά 9 πράξεις σύγκρισης, αφού σε κάθε πέρασμα γίνονται 3 συγκρίσεις και όπως αναφέρθηκε υπάρχουν 3 περάσματα. Ο πίνακας του παραδείγματός μας έχει $n=4$ θέσεις. Αν γενικεύσουμε την καταγραφή των πράξεων της σύγκρισης, παρατηρούμε ότι θα χρειαστούν $(n-1)*(n-1)$ **συγκρίσεις** για έναν πίνακα n θέσεων.

Αριθμός Ανταλλαγών

Έστω τώρα ότι θέλουμε να καταγράψουμε τον αριθμό των ανταλλαγών που γίνονται για τις ανάγκες της ταξινόμησης. Για να υπάρξει κριτήριο για την τυποποίηση της επίδοσης του αλγορίθμου, θα πρέπει να μετρηθεί ο αριθμός των ανταλλαγών που γίνονται στη χειρότερη περίπτωση. Επομένως η χειρότερη περίπτωση για έναν πίνακα 4 θέσεων (που είναι και το μέγεθος του προβλήματός μας) είναι τα στοιχεία του να βρίσκονται ταξινομημένα κατά φθίνουσα τάξη έτσι ώστε όλα να πρέπει να αλλάξουν θέση. Έστω λοιπόν η πιο απλή περίπτωση αυτής της κατάστασης:

$i \leftarrow 1$ (1ο Πέρασμα)

4	3	2	1	1 ανταλλαγή
3	4	2	1	1 ανταλλαγή
3	2	4	1	1 ανταλλαγή

$i \leftarrow 2$ (2ο Πέρασμα)

3	2	1	4	1 ανταλλαγή
2	3	1	4	1 ανταλλαγή
2	1	3	4	0 ανταλλαγή

$i \leftarrow 3$ (3ο Πέρασμα)

2	1	3	4	1 ανταλλαγή
1	2	3	4	0 ανταλλαγή
1	2	3	4	0 ανταλλαγή

Επομένως θα γίνουν συνολικά $3+2+1=6$ ανταλλαγές για την ταξινόμηση 4 στοιχείων που είναι τοποθετημένα σε αντίθετη της κανονικής τάξης (χειρότερη περίπτωση). Αν γενικεύσουμε το παράδειγμα για πίνακα n θέσεων θα χρειασθούν στη χειρότερη περίπτωση $(n-1)+(n-2)+\dots+1$ ανταλλαγές στοιχείων.

Η πολυπλοκότητα του παραπάνω αλγορίθμου υπολογίζεται από το άθροισμα του «κόστους» των συγκρίσεων και των ανταλλαγών που είναι:

$$(n-1) + (n-2) + \dots + 1 + (n-1) \cdot (n-1) = \frac{3}{2}n^2 - \frac{5}{2}n + 1$$

Είναι φανερό ότι η τάξη της πολυπλοκότητας του αλγορίθμου υπολογίζεται από το μεγαλύτερο όρο του πολυωνύμου, ο οποίος είναι το n^2 , και έτσι προκύπτει ότι ο παραπάνω αλγόριθμος έχει τετραγωνική πολυπλοκότητα.

ΠΑΡΑΔΕΙΓΜΑ 4. Εύρεση του i -οστού αριθμού Fibonacci

Στο κεφάλαιο του Βιβλίου σου περί αναδρομής συζητήσαμε τα πλεονεκτήματα και μειονεκτήματα των επαναληπτικών και των αναδρομικών συναρτήσεων. Μάλιστα καταλήξαμε στο συμπέρασμα ότι μία επαναληπτική διαδικασία πρέπει να προτιμάται έναντι μίας ισοδύναμης αναδρομικής όταν το βασικό μας κριτήριο είναι ο χρόνος εκτέλεσης/απόκρισης. Στο σημείο αυτό επανερχόμαστε ώστε να εξηγηθεί ότι συχνά μία αναδρομική πρέπει να αποφεύγεται και για έναν άλλο σημαντικό λόγο.

Οι επόμενοι δύο αλγόριθμοι υπολογίζουν τον i -οστό αριθμό Fibonacci. Ο πρώτος είναι επαναληπτικός, ενώ ο δεύτερος είναι αναδρομικός.


```

Αλγόριθμος Fibonacci
Δεδομένα // n //
Αν  $n \leq 1$  τότε
    Fib  $\leftarrow$  n
αλλιώς
    j  $\leftarrow$  0
    k  $\leftarrow$  1
    Για i από 1 μέχρι n
        j  $\leftarrow$  j+k
        k  $\leftarrow$  j-k
    Τέλος_επανάληψης
    Fib  $\leftarrow$  j
Τέλος_Αν
Αποτελέσματα // Fib //
Τέλος Fibonacci

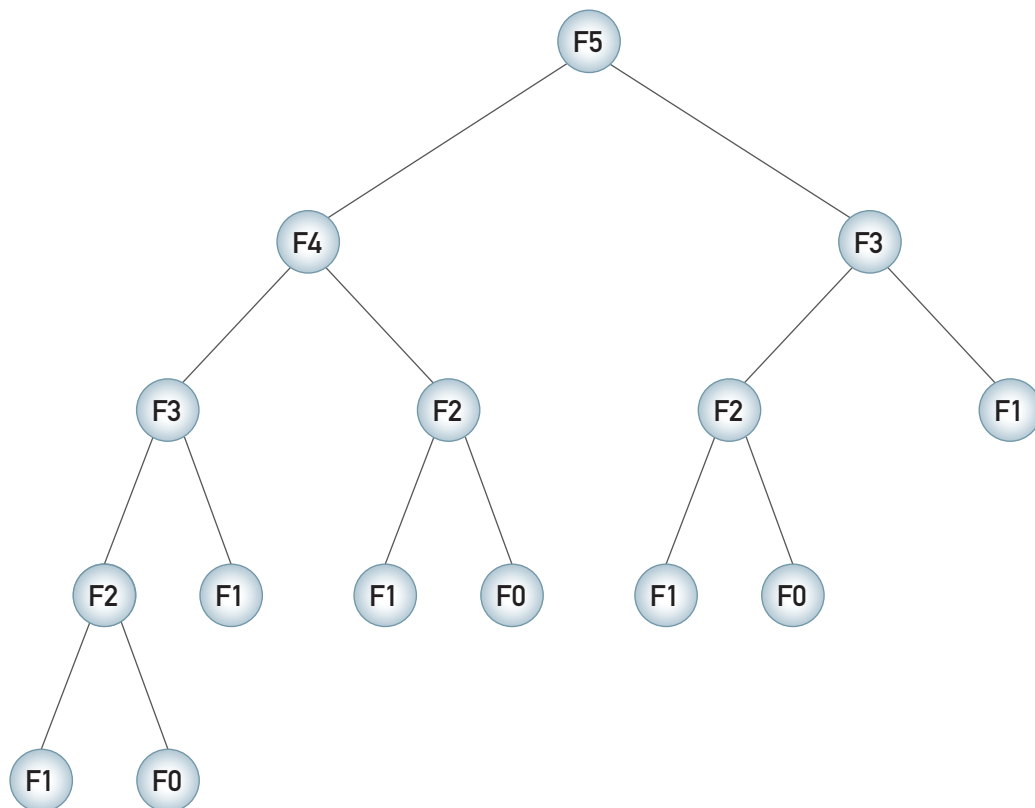
```

```

Αλγόριθμος Fibonacci
Δεδομένα // n //
Αν  $n \leq 1$  τότε
    Fib  $\leftarrow$  n
αλλιώς
    Fib  $\leftarrow$  Fibonacci(n-1) +
        Fibonacci(n-2)
Τέλος_Αν
Αποτελέσματα // Fib //
Τέλος Fibonacci

```

Η επαναληπτική διαδικασία, λοιπόν, βασίζεται σε ένα και μόνο βρόχο, άρα η πολυπλοκότητα της μεθόδου είναι τάξης $O(n)$. Η δεύτερη διαδικασία υπολογίζει τις ίδιες τιμές πολλές φορές. Για παράδειγμα, κατά τον υπολογισμό του F_5 γίνονται οι κλήσεις Fibonacci (4) και Fibonacci (3). Όμως κατά τον υπολογισμό του Fibonacci (4) καλείται για δεύτερη φορά η Fibonacci (3). Με το σκεπτικό αυτό η Fibonacci (2) θα κληθεί τρεις φορές, η Fibonacci (1) θα κληθεί πέντε φορές και η Fibonacci (0) θα κληθεί τρεις φορές. Έτσι συμπερασματικά προκύπτει ότι η πολυπλοκότητα της αναδρομικής μεθόδου είναι τάξης $O(F_n)$.



Σχ. 5.1. Κλήσεις για τον υπολογισμό του F_5

5.3. Δραστηριότητες - ασκήσεις

ΔΤ1. Να συζητηθεί η επίδοση των παρακάτω κομματιών αλγορίθμων και να καταγραφεί για κάθε περίπτωση και η αντίστοιχη πολυπλοκότητα. Για να βρείτε την πολυπλοκότητα, θα πρέπει να μετρήσετε τον αριθμό των πράξεων στη χειρότερη περίπτωση:

1.

```
Για i από 1 μέχρι n - 1 με βήμα 2
  a ← 2*i
Τέλος_επανάληψης
```

2.

```
Για i από 1 μέχρι n
  Για j από 1 μέχρι n
    a ← 2*i + j
  Τέλος_επανάληψης
Τέλος_επανάληψης
```

3.

```
Για i από 1 μέχρι n με βήμα 2
  Για j από 1 μέχρι n
    a ← 2*i + j
  Τέλος_επανάληψης
Τέλος_επανάληψης
```

ΔΤ2. Έστω ότι ένας πίνακας κρατά τα ποσά που έχουν δώσει οι μαθητές της τάξης σας για την ενίσχυση του παιδικού χωριού SOS της περιοχής σας. Να δώσετε έναν αλγόριθμο για τον υπολογισμό του συνολικού ποσού που θα διατεθεί και να σχολιάσετε την πολυπλοκότητά του.

ΔΤ3. Να σχολιασθεί και να παρακολουθήσετε βήμα-βήμα τον ακόλουθο επαναληπτικό αλγόριθμο υπολογισμού των αριθμών Fibonacci. Ποια είναι η πολυπλοκότητα του ακόλουθου αλγορίθμου;

Αλγόριθμος Fibonacci

```
i ← 1
j ← 0
k ← 0
l ← 1
```

Επανάλαβε όσο $n > 0$

Αν $n \bmod 2 = 1$ **τότε** $m \leftarrow j * l$

$j \leftarrow i * l + j * k + m$

$i \leftarrow i * k + m$

$m \leftarrow \text{Ρίζα}(l)$!Ρίζα είναι η συνάρτηση τετραγωνικής ρίζας'

$h \leftarrow 2 * k * l + m$

$k \leftarrow \text{Ρίζα}(k) + m$

$n \leftarrow n \text{ DIV } 2$

Τέλος_επανάληψης

Fib ← j

Αποτελέσματα Fib

Τέλος Fibonacci

ΔΤ4. Σε μία αποθήκη ταινιών κινηματογράφου υπάρχει αρχειοθέτηση των ταινιών με βάση τη χρονιά που παρουσιάστηκε κάθε ταινία. Ένας σύλλογος ενδιαφέρεται να κάνει ένα αφιέρωμα στις ταινίες της δεκαετίας του 1960 που διαθέτει η αποθήκη. Να προτείνετε έναν αλγόριθμο αναζήτησης των ταινιών αυτών και να σχολιάσετε την επίδοση και την πολυπλοκότητά του.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Να βρεις την πολυπλοκότητα των παρακάτω κομματιών αλγορίθμων:

1.
 Όσο $i < 100$ επανάλαβε
 $a \leftarrow 2 * i$
 Τέλος_επανάληψης

2.
 Για i από 1 μέχρι $n - 1$
 $a \leftarrow 2 * i$
 Τέλος_επανάληψης

3.
 Για i από 1 μέχρι $n - 1$ με βήμα 3
 $a \leftarrow 2 * i$
 Τέλος_επανάληψης

ΔΣ2. Έστω ότι έχουμε τον παρακάτω αλγόριθμο:

Αλγόριθμος Ελεγχος_επίδοσης
 $x \leftarrow 10$
 $c \leftarrow 20$
 Για i από 100 μέχρι 10 με_βήμα -10
 $x \leftarrow i$
 $y \leftarrow 2 * x + i$
 Τέλος_επανάληψης
Εκτύπωσε x
Εκτύπωσε y
Τέλος Ελεγχος_επίδοσης

Να υπολογισθεί η επίδοσή του με βάση τον αριθμό των πράξεων που θα εκτελεστούν.

ΔΣ3. Να παρακολουθήσεις την πορεία του αλγορίθμου της Δυναμικής αναζήτησης που δόθηκε στο προηγούμενο κεφάλαιο (Τετράδιο Μαθητή) για έναν πίνακα 10 θέσεων. Να καταγράψεις τον αριθμό των πράξεων που γίνονται σε κάθε βήμα και να καταλήξεις σε ένα συμπέρασμα για την πολυπλοκότητά του.

ΔΣ4. Έστω ότι έχεις τον παρακάτω αλγόριθμο:

```
Αλγόριθμος Ευθεία_Ανταλλαγή2
Δεδομένα // A //
Για i από 1 μέχρι n-1
    Για j από 1 μέχρι n-1
        Αν A[j+1] < A[j] τότε
            Αντιμετάθεσε A[j+1], A[j]
        Τέλος_αν
    Τέλος_επανάληψης
Εκτύπωσε A[j]
Τέλος_επανάληψης
Αποτελέσματα // A //
Τέλος Ευθεία_Ανταλλαγή2
```

Να σχολιάσεις την πολυπλοκότητα του αλγορίθμου. Χρησιμοποίησε το Παράδειγμα 2 και να ελέγξεις αν ο παραπάνω αλγόριθμος είναι ίδιας ή διαφορετικής πολυπλοκότητας από τον αλγόριθμο του Παραδείγματος 2.

ΔΣ5. Μπορείς να προτείνεις κάποιο διαφορετικό αλγόριθμο για την ανεύρεση των ταινιών κινηματογράφου από την αποθήκη που περιγράφηκε παραπάνω στις δραστηριότητες για την τάξη (ΔΤ4); Πώς θα σχολίαζες την πολυπλοκότητα των δύο διαφορετικών αλγορίθμων;

5.4. Τεστ αυτοαξιολόγησης

1. Δίνονται οι παρακάτω ομάδες προτάσεων. Σε καθεμία από αυτές, να κάνετε τις απαραίτητες διορθώσεις ώστε να ισχύουν οι προτάσεις:

1. Η χειρότερη περίπτωση ενός αλγορίθμου αφορά στο ελάχιστο κόστος εκτέλεσης του αλγορίθμου, κόστος που μετράται σε υπολογιστικούς πόρους.
2. Τα δεδομένα συνιστούν το μέγεθος της πολυπλοκότητας ενός αλγορίθμου.
3. Ο απλούστερος τρόπος μέτρησης της επίδοσης ενός αλγορίθμου είναι ο *εμπειρικός* ή αλλιώς ο λεγόμενος *εκ των προτέρων* που υλοποιείται και εφαρμόζεται σε ένα σύνολο δεδομένων ώστε να υπολογισθεί ο απαιτούμενος χρόνος *επεξεργασίας* και η πολυπλοκότητα.

2. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

4. Τα δεδομένα συνιστούν το μέγεθος της _____ ενός αλγορίθμου.
5. Ο συμβολισμός $O(n^2)$ εκφράζει την _____ πολυπλοκότητα.
6. Ο συμβολισμός $O(\log n)$ εκφράζει τη _____ πολυπλοκότητα.
7. Ως _____ ορίζονται οι αλγόριθμοι που δε δίνουν την καλύτερη λύση, αλλά προτιμώνται για λόγους ταχύτητας.

3. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

8. Αν η πολυπλοκότητα ενός αλγορίθμου είναι $f(n)$, τότε λέγεται ότι ο αλγόριθμος είναι τάξης $O(g(n))$ αν υπάρχουν δύο θετικοί ακέραιοι c και n_0 , έτσι ώστε για κάθε $n \geq n_0$ να ισχύει: $|f(n)| > c |g(n)|$
9. Ο συμβολισμός $O(n^2)$ εκφράζει την κυβική πολυπλοκότητα και πρέπει να χρησιμοποιείται μόνο για προβλήματα μεγάλου μεγέθους.
10. Ο συμβολισμός $O(n)$ εκφράζει τη γραμμική πολυπλοκότητα η οποία είναι η καλύτερη επίδοση για έναν αλγόριθμο που πρέπει να εξετάσει ή να δώσει στην έξοδο n στοιχεία.
11. Ευριστικοί λέγονται οι αλγόριθμοι που δεν είναι τυποποιημένοι και δεν ανήκουν στις γνωστές οικογένειες αλγορίθμων.

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

12. Μία βασική πράξη μπορεί να είναι:
 - A) βρόχος επανάληψης
 - B) ανάθεση τιμής
 - Γ) σύγκριση μεταξύ δύο μεταβλητών
 - Δ) συνθήκη Αν..αλλιώς
 - E) οποιαδήποτε αριθμητική πράξη μεταξύ δύο μεταβλητών
13. Ο χρόνος εκτέλεσης κάθε αλγορίθμου εξαρτάται από ένα σύνολο παραγόντων που περιλαμβάνουν τα εξής:
 - A) Τύπος ηλεκτρονικού υπολογιστή που θα εκτελέσει το πρόγραμμα του αλγορίθμου
 - B) Χρονική στιγμή εκτέλεσης του αλγορίθμου
 - Γ) Συνθήκες ανταγωνισμού με άλλους αλγορίθμους
 - Δ) Γλώσσα προγραμματισμού που θα χρησιμοποιηθεί
 - E) Δομή προγράμματος και δομές δεδομένων που χρησιμοποιεί
 - Z) Αριθμός εντολών του αλγορίθμου
 - H) Χρόνος για πρόσβαση στο δίσκο και στις ενέργειες εισόδου-εξόδου
 - Θ) Είδος συστήματος, ενός χρήστη ή πολλαπλών χρηστών

Κεφάλαιο 6

Εισαγωγή στον Προγραμματισμό

6.1. Προσδοκώμενα αποτελέσματα

Στο κεφάλαιο αυτό θα γνωρίσεις τις βασικές έννοιες και τις βασικές αρχές των τεχνικών και των μεθόδων που θα χρησιμοποιήσεις στη συνέχεια για την ανάπτυξη των προγραμμάτων σου. Θα γνωρίσεις τα χαρακτηριστικά των κυριότερων γλωσσών προγραμματισμού και τις τεχνικές της ιεραρχικής σχεδίασης και του τμηματικού προγραμματισμού. Ιδιαίτερο βάρος πρέπει να δώσεις στα χαρακτηριστικά και κυρίως στα πλεονεκτήματα του δομημένου προγραμματισμού.

Πρέπει να είσαι σε θέση να περιγράφεις όλη τη διαδικασία δημιουργίας και εκτέλεσης ενός προγράμματος και τέλος να εκτελέσεις ένα απλό έτοιμο πρόγραμμα στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σου.

6.2. Δραστηριότητες - ασκήσεις

- ΔΤ1.** Ο παρακάτω αλγόριθμος αποτελεί τμήμα μη δομημένου προγράμματος. Να γράψεις αλγόριθμο σχεδιασμένο με τις αρχές του δομημένου προγραμματισμού, που να εκτελεί τις ίδιες λειτουργίες.

```
ΑΡΧΗ
ΟΣΟ συνθήκη1 ΕΠΑΝΑΛΑΒΕ
    Εντολή 2
ΑΝ συνθήκη3 ΤΟΤΕ
    Εντολή4
    Πήγαινε στο Τέλος
ΑΛΛΙΩΣ
    Εντολή5
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ
```

- ΔΤ2.** Σχεδιάσε ένα διάγραμμα που να δείχνει τις διάφορες λειτουργίες και τις σχέσεις ανάμεσα σε αυτές τις λειτουργίες για το πρόγραμμα που συγκεντρώνει και επεξεργάζεται τα δεδομένα του προβλήματος της δραστηριότητας ΔΤ1 του πρώτου κεφαλαίου του τετραδίου σου, την έρευνα της *SOS Ρατσισμός*.

Το πρόβλημα αυτό περιλαμβάνει τη συγκέντρωση των απαντήσεων, την επεξεργασία τους και την παρουσίαση σε μορφή πίνακα των αποτελεσμάτων για όλες τις χώρες της Ευρωπαϊκής Ένωσης.

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

- ΔΕ1.** Στο προγραμματιστικό περιβάλλον να εκτελέσετε ένα έτοιμο πρόγραμμα. Να κάνετε αλλαγές με το συντάκτη σε μία εντολή, να το μεταγλωττίσετε, να σημειώσετε τα λάθη, να τα διορθώσετε (ουσιαστικά να επαναφέρετε το πρόγραμμα στην αρχική του μορφή) και τελικά να πάρετε τα αποτελέσματα.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

- ΔΣ1.** Να καταγράψεις σε έναν κατάλογο τα ονόματα και κυρίως τα βασικά χαρακτηριστικά των γλωσσών προγραμματισμού. Συγκεκριμένα για κάθε γλώσσα προγραμματισμού να καταγράφεις: το όνομά της, τις διάφορες εκδόσεις της, το είδος των δραστηριοτήτων για την οποία χρησιμοποιείται, τη διάδοσή της σε προσωπικούς υπολογιστές, και το είδος του προγραμματισμού που υποστηρίζει, για παράδειγμα αντικειμενοστραφή, οπτικό, οδηγούμενο από τα γεγονότα κ.λπ.

Για τη δημιουργία του καταλόγου να ανατρέξεις στα βιβλία σου και σε όποια άλλη πηγή έχεις διαθέσιμη.

- ΔΣ2.** Ο παρακάτω αλγόριθμος αποτελεί τμήμα μη δομημένου προγράμματος.
Να γράψεις αλγόριθμο σχεδιασμένο με τις αρχές του δομημένου προγραμματισμού,
που να εκτελεί τις ίδιες λειτουργίες.

```
ΑΡΧΗ
ΑΝ συνθήκη1 ΤΟΤΕ
    Εντολή1
    ΑΝ συνθήκη2 ΤΟΤΕ
        Εντολή2
        Εντολή3
        Πήγαινε στην Εντολή5
    ΤΕΛΟΣ_ΑΝ
    Εντολή4
    Εντολή5
    Πήγαινε στην Αρχή
ΤΕΛΟΣ_ΑΝ
Εντολή6
ΤΕΛΟΣ
```

6.3. Τεστ αυτοαξιολόγησης

1. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

1. Ο μεταγωγτιστής μεταγωγτίζει το _____ πρόγραμμα σε αντικείμενο πρόγραμμα.
2. Ο τμηματικός προγραμματισμός υλοποιεί την _____ σχεδίαση του προγράμματος.
3. Οι γλώσσες που υλοποιούν τον _____ και τον _____ διευκολύνουν την ανάπτυξη εφαρμογών σε γραφικά περιβάλλοντα.

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

4. Η Visual Basic είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού.
5. Οι εντολές στις συμβολικές γλώσσες αποτελούνται από ακολουθίες 0 και 1.
6. Ο δομημένος προγραμματισμός εξασφαλίζει τη δημιουργία σωστών προγραμμάτων.
7. Οι γλώσσες 4ης γενιάς είναι κατάλληλες για ανάπτυξη γενικών εφαρμογών.

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

8. Οι εντολές ενός προγράμματος γράφονται σε ένα πρόγραμμα που ονομάζεται:
Α) Συντάκτης
Β) Μεταγωγτιστής

- Γ) Διερμηνευτής
 - Δ) Συνδέτης
9. Η Pascal είναι μία γλώσσα:
- A) Μηχανής
 - B) Υψηλού επιπέδου
 - Γ) Συμβολική
 - Δ) 4ης γενιάς
10. Ο μεταγλωττιστής επισημαίνει:
- A) Όλα τα λάθη του προγράμματος
 - B) Μόνο τα λογικά λάθη του προγράμματος
 - Γ) Μόνο τα συντακτικά λάθη του προγράμματος
 - Δ) Μόνο τα λάθη που προέρχονται από αναγραμματισμό των εντολών
11. Ο δομημένος προγραμματισμός είναι:
- A) Μία γενική μεθοδολογία ανάπτυξης προγραμμάτων
 - B) Ένας τρόπος προγραμματισμού που εφαρμόζεται μόνο από τη γλώσσα Pascal
 - Γ) Η εξέλιξη του τμηματικού προγραμματισμού
 - Δ) Ένας τρόπος να εξαλείψουμε τις εντολές GOTO από ένα πρόγραμμα

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

12. Ποια από τα παρακάτω είναι χαρακτηριστικά ενός δομημένου προγράμματος:
- A) Δομικό στοιχείο είναι τα αντικείμενα.
 - B) Έχει μία είσοδο και μία έξοδο.
 - Γ) Χρησιμοποιεί τις τρεις δομές: της ακολουθίας, της επιλογής και της επανάληψης.
 - Δ) Μπορεί να εκμεταλλευτεί τους παράλληλους υπολογιστές.
13. Κάθε φυσική γλώσσα προσδιορίζεται από:
- A) Το αλφάβητό της
 - B) Το λεξιλόγιό της
 - Γ) Τη γραμματική της
 - Δ) Τη σημασιολογία της
14. Ποιες από τις παρακάτω γλώσσες χρησιμοποιούνται για ανάπτυξη εφαρμογών τεχνητής νοημοσύνης:
- A) LISP
 - B) FORTRAN
 - Γ) COBOL
 - Δ) PROLOG
 - E) JAVA

Βασικές Έννοιες Προγραμματισμού

7.1. Προσδοκώμενα αποτελέσματα

Το κεφάλαιο αυτό ουσιαστικά αποτελεί την πρώτη σου επαφή με προγραμματιστικό περιβάλλον. Παρουσιάζονται τα βασικά στοιχεία προγραμματισμού και σταδιακά δομείται η προγραμματιστική τακτική. Ανεξάρτητα από το πραγματικό περιβάλλον προγραμματισμού που στη συνέχεια θα δουλέψεις, θεμελιώδεις κανόνες, αρχές και έννοιες που πρωτοπαρουσιάζονται σε αυτό το κεφάλαιο, όπως οι έννοιες της μεταβλητής και της σταθεράς, η εκχώρηση τιμής, είναι αναγκαίο να γίνουν κτήμα σου. Η σωστή αντίληψη και εικόνα που θα πρέπει να σχηματίσεις θα σε βοηθήσει να ασχοληθείς με μεγαλύτερη ευκολία με άλλα προγραμματιστικά περιβάλλοντα.

Οι λυμένες ασκήσεις του κεφαλαίου παρουσιάζονται αρχικά στο περιβάλλον της ιδεατής γλώσσας προγραμματισμού ΓΛΩΣΣΑ και στη συνέχεια κάποιες από αυτές παρουσιάζονται στα πραγματικά προγραμματιστικά περιβάλλοντα Basic και Pascal.

7.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

Μία μπάλα η οποία εκτοξεύεται στον αέρα ακολουθεί μία παραβολική τροχιά μέχρι να πέσει πάλι στη γη. Αν θεωρήσουμε την αντίσταση του αέρα αμελητέα, και αν αγνοήσουμε την καμψυλότητα της γης, τότε το ύψος της μπάλας σε κάθε χρονική στιγμή δίνεται από τον τύπο:

$$Y(t) = y_0 + v_{y0}t + \frac{1}{2}gt^2$$

v_0 : αρχική ταχύτητα

y_0 : το αρχικό ύψος από τη γη

g : η επιτάχυνση της βαρύτητας

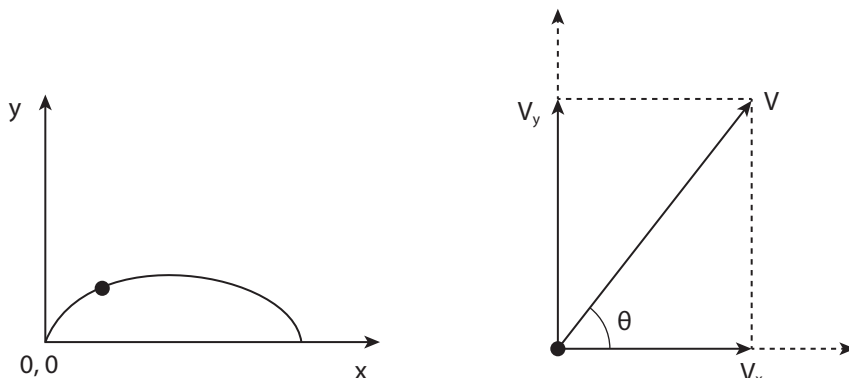
θ : η γωνία βολής

v_{y0} : η προβολή της αρχικής ταχύτητας στον άξονα των y , $v_{y0} = v_0 \eta\mu\theta$ ενώ η οριζόντια απόσταση είναι $x(t) = x_0 + v_{x0}t$

x_0 : η αρχική οριζόντια θέση

v_{x0} : η προβολή της αρχικής ταχύτητας στον άξονα των x , $v_{x0} = v_0 \sigma\upsilon\nu\theta$

Αν θεωρήσουμε ότι η μπάλα εκτοξεύεται από το σημείο 0,0, τότε, όπως φαίνεται από τους τύπους παραπάνω, θα πέσει στη γη σε απόσταση $x = 2v_{x0}v_{y0}/g$, το οποίο ονομάζεται βεληνεκές.



Να γραφεί πρόγραμμα που να υπολογίζει την απόσταση x στην οποία η μπάλα θα πέσει στη γη (το βεληνεκές). Θεωρήστε ότι η μπάλα ξεκινάει από το σημείο με συντεταγμένες 0,0 και με αρχική ταχύτητα 20 m/sec. Η γωνία θ δίνεται από το πληκτρολόγιο κατά την ώρα της εκτέλεσης.



Για τον υπολογισμό των τριγωνομετρικών συναρτήσεων ημίτονο και συνημίτονο η γωνία βολής θ στο περιβάλλον της ΓΛΩΣΣΑΣ εκφράζεται σε μοίρες. Στα πραγματικά προγραμματιστικά περιβάλλοντα θα πρέπει να προσέξεις αν υπάρχει δυνατότητα έκφρασης της γωνίας σε μοίρες, ή όπως συμβαίνει συνήθως, υποχρεωτικά θα πρέπει η γωνία να εκφράζεται σε ακτίνια. Υπενθυμίζεται ότι **2π ακτίνια = 360 μοίρες**.

Περιβάλλον προγραμματισμού ΓΛΩΣΣΑ

```
ΠΡΟΓΡΑΜΜΑ Τροχιά_μπάλας
! Πρόγραμμα υπολογισμού θέσης μπάλας
! Θεωρούμε την αντίσταση του αέρα αμελητέα και αγνοούμε την καμπυλό-
  τητα της γης

ΣΤΑΘΕΡΕΣ
G = 9.81

ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ:V0,VX0,VY0,Θ, Βεληνεκές

ΑΡΧΗ
! Αρχικές τιμές

V0 <- 20

! Εισαγωγή δεδομένων
ΓΡΑΨΕ 'Δώσε τη γωνία'
ΔΙΑΒΑΣΕ Θ

! Υπολογισμοί
VX0 <- V0*ΣΥΝ(Θ)
VY0 <- V0*ΗΜ(Θ)
Βεληνεκές <- 2*VX0*VY0/G

! Εμφάνιση αποτελεσμάτων
ΓΡΑΨΕ 'Το βεληνεκές για γωνία ',Θ,' είναι:', Βεληνεκές

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Περιβάλλον προγραμματισμού PASCAL

```
program ball;
const
  g=9.81;
  pi=3.14;
var
  a:integer;
  range,akt, v0,vx0,vy0:real;
begin
  {αρχικές τιμές} v0:=20;
  {εισαγωγή δεδομένων}
  write('ΔΩΣΕ ΤΗ ΓΩΝΙΑ :');
  readln(a);
  {μετατροπή της γωνίας σε ακτίνια}
  akt:=a*pi/180;
  vx0:=v0*cos(akt);
  vy0:=v0*sin(akt);
  range:=2*vx0*vy0/g;
  writeln('ΓΩΝΙΑ',A:3,' ΒΕΛΗΝΕΚΕΣ:',range:7:2);
end.
```

Περιβάλλον προγραμματισμού BASIC

```
`τροχιά μπάλας
g = 9.81:pi=3.14
v0 = 20
INPUT "Γωνία=", a
a = a * pi / 180
vx0 = v0 * COS(a)
vy0 = v0 * SIN(a)
vel = 2 * vx0 * vy0 / g
PRINT "Βεληνεκές ="; vel
END
```

ΠΑΡΑΔΕΙΓΜΑ 2

Η αγορά ενός αυτοκινήτου πολύ συχνά γίνεται με δόσεις. Ο υπολογισμός της δόσης εξαρτάται από την τιμή του αυτοκινήτου, την προκαταβολή, το επιτόκιο και τέλος την περίοδο αποπληρωμής.

Συγκεκριμένα δίνεται από τον τύπο:

$$a = i(p - d) \frac{(1+i)^m}{(1+i)^m - 1}$$

a: Μηνιαία δόση

p: Αρχική τιμή αυτοκινήτου

d: Ποσό προκαταβολής

i: Μηνιαίο επιτόκιο

m: Περίοδος αποπληρωμής σε μήνες

Να γραφεί πρόγραμμα το οποίο να διαβάζει την τιμή του αυτοκινήτου, το ποσό της προκαταβολής, το επιτόκιο και την περίοδο αποπληρωμής και στη συνέχεια να υπολογίζει το ποσό της κάθε δόσης καθώς και το ποσοστό επιβάρυνσης της τιμής του αυτοκινήτου από την αρχική του αξία.

ΠΡΟΓΡΑΜΜΑ Υπολογισμός_δόσεων

!Υπολογισμός δόσεων αγοράς αυτοκινήτου

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:Τιμή, Προκαταβολή, Περίοδος, Υπόλοιπο

ΠΡΑΓΜΑΤΙΚΕΣ:Ετήσιο_επιτόκιο,Επιτόκιο,Τελικό_ποσό, Δόση,Κλάσμα,Ποσοστό

!Τιμή:Αρχική τιμή αυτοκινήτου σε ευρώ

!Προκαταβολή:Ποσό προκαταβολής σε ευρώ

!Δόση: Μηνιαία δόση σε ευρώ

!Ετήσιο_επιτόκιο:ετήσιο επιτόκιο σε ποσοστό %

!Επιτόκιο:Μηνιαίο επιτόκιο σε δεκαδικό αριθμό

!Περίοδος: Περίοδος αποπληρωμής σε μήνες

!Τελικό_ποσό: Συνολικό ποσό όλων των δόσεων

!Ποσοστό:Ποσοστό διαφοράς τιμής δόσεων με αρχική τιμή

ΑΡΧΗ

ΓΡΑΨΕ `Δώσε την αρχική τιμή: `

ΔΙΑΒΑΣΕ Τιμή

ΓΡΑΨΕ `Δώσε την προκαταβολή: `

```

ΔΙΑΒΑΣΕ Προκαταβολή
ΓΡΑΨΕ 'Δώσε επιτόκιο: '
ΔΙΑΒΑΣΕ Ετήσιο_επιτόκιο
ΓΡΑΨΕ 'Δώσε περίοδο σε μήνες:'
ΔΙΑΒΑΣΕ Περίοδος
Επιτόκιο <- (Ετήσιο_επιτόκιο/12)/100
Υπόλοιπο <- Τιμή-Προκαταβολή
Κλάσμα <- (1+Επιτόκιο)^Περίοδος/((1+Επιτόκιο)^Περίοδος-1)
Δόση <- Επιτόκιο*Υπόλοιπο*Κλάσμα
Τελικό_ποσό <- Δόση*Περίοδος+Προκαταβολή
Ποσοστό <- Τελικό_ποσό/Τιμή*100
ΓΡΑΨΕ ' Η μηνιαία δόση είναι:', Δόση
ΓΡΑΨΕ ' Το συνολικό ποσό είναι:', Τελικό_ποσό,
ΓΡΑΨΕ ' η αύξηση από το αρχικό είναι ', Ποσοστό, '%'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```



7.3. Συμβουλές - υποδείξεις

Τώρα που αρχίζεις να γράφεις προγράμματα, καλό και χρήσιμο θα ήταν να μάθεις να ακολουθείς κάποιους κανόνες και κάποιες γενικές αρχές, έτσι ώστε η συγγραφή, η κατανόηση και η τροποποίηση των προγραμμάτων σου να γίνεται εύκολα και γρήγορα. Τη σημασία όλων αυτών θα τη νιώθεις όλο και περισσότερο όσο τα προγράμματα γίνονται περισσότερο σύνθετα και πολύπλοκα.

Τα προγράμματά σου πρέπει να είναι **απλά** και **κατανοητά**. Όχι μόνο για τους άλλους που θα χρειαστεί κάποια στιγμή να τα διαβάσουν και να τα καταλάβουν, αλλά και για σένα τον ίδιο που μετά από κάποιο καιρό θα επανέλθεις σε παλαιότερό σου πρόγραμμα για να το τροποποιήσεις ή να το επεκτείνεις.

να χρησιμοποιείς ονόματα σταθερών και μεταβλητών που να υπονοούν τη χρήση τους.

να γράφεις σχόλια μέσα στο πρόγραμμά σου, ειδικά σε εκείνα τα σημεία του που υπάρχουν σχετικές δυσκολίες στην κατανόηση.

η χρήση κενών γραμμών διευκολύνει στην ανάγνωση του προγράμματος και οριοθετεί τις ενότητές του.

η χρησιμοποίηση σταθερών σε διευκολύνει σε πιθανές επόμενες αλλαγές και σε προστατεύει από ενδεχόμενες αθέλητες τροποποιήσεις.

Θα πρέπει να αποδίδεις αρχικές τιμές στις μεταβλητές που χρησιμοποιείς στο πρόγραμμα. Σε πολλά προγραμματιστικά περιβάλλοντα η ίδια η γλώσσα φροντίζει έτσι ώστε να αποδίδει αυτόματα αρχική τιμή ίση με μηδέν στις μεταβλητές. Η δυνατότητα αυτή όμως πολλές φορές μπορεί να οδηγήσει σε λάθος αποτελέσματα. Επιπλέον η απόδοση αρχικών τιμών βοηθάει στην καλύτερη κατανόηση του προγράμματος και στην ευκολότερη συντήρησή του.

Να αποφεύγεις να χρησιμοποιείς μεγάλους υπολογισμούς. Η διάσπαση ενός υπολογισμού σε απλούστερους διευκολύνει τους άλλους στην κατανόηση του προγράμματος και σένα στην αποφυγή λαθών.

7.4. Δραστηριότητες - ασκήσεις

ΔΤ1. Να μετατρέψετε σε κώδικα προγράμματος τις παρακάτω παραστάσεις:

1. Η περίοδος γραμμικής αρμονικής ταλάντωσης είναι:

$$T = 2\pi\sqrt{\frac{m}{D}}$$

2. Η κινητική ενέργεια ενός σώματος είναι:

$$E_{\text{κιν}} = \frac{1}{2}mv^2$$

3. Συνισταμένη δύο δυνάμεων που ενεργούν στο ίδιο σημείο και σχηματίζουν γωνία φ δίνεται από τον τύπο:

$$F = \sqrt{F_1^2 + F_2^2 + 2F_1F_2\cos\varphi}$$

4. Η μία λύση εξίσωσης Β' βαθμού είναι:

$$\frac{-\beta + \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}$$

ΔΤ2. Τι τύπου μεταβλητές πρέπει να χρησιμοποιήσετε για τα παρακάτω στοιχεία του μαθητολόγιου του σχολείου μας; Γράψτε το αντίστοιχο τμήμα δηλώσεων.

1. Το όνομα ενός μαθητή.
2. Ο αριθμός μαθητολογίου του μαθητή.
3. Τη βαθμολογία του μαθητή.
4. Το τηλέφωνο ενός μαθητή.
5. Τη διεύθυνση ενός μαθητή.
6. Το φύλο ενός μαθητή (πώς μπορεί να οριστεί με χρήση λογικής μεταβλητής;)

ΔΤ3. Γράψτε το πρόγραμμα για το παρακάτω πρόβλημα και στη συνέχεια πραγματοποιήστε εικονική εκτέλεσή του έτσι ώστε να βεβαιωθείτε ότι λειτουργεί σωστά.

Δίδονται οι πλευρές ενός τριγώνου και υπολογίζεται το εμβαδόν του τριγώνου με τον τύπο του Ήρωνα:

$$E = \sqrt{\tau(\tau - \alpha)(\tau - \beta)(\tau - \gamma)}$$

όπου τ είναι η ημιπερίμετρος του τριγώνου, δηλαδή $\tau = \frac{\alpha + \beta + \gamma}{2}$.

Προβληματιστείτε πάνω στο ερώτημα "Μπορεί ο υπολογισμός αυτός να γίνεται για κάθε τριάδα αριθμών". Προσπαθήστε να δικαιολογήσετε την απάντησή σας όσο καλύτερα μπορείτε.

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1. Γράψτε τον κώδικα και εκτελέστε το πρόγραμμα για το παράδειγμα 2, “Υπολογισμός δόσεων αγοράς αυτοκινήτου”. Εκτελέστε το πρόγραμμα για διάφορες τιμές προκαταβολής και περιόδους αποπληρωμής.

ΔΕ2. Η απόσταση μεταξύ δύο σημείων (x_1, y_1) και (x_2, y_2) ενός Καρτεσιανού συστήματος συντεταγμένων υπολογίζεται από τον τύπο:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Γράψτε πρόγραμμα το οποίο να υπολογίζει και να εκτυπώνει την απόσταση δύο σημείων των οποίων οι συντεταγμένες δίνονται από το χρήστη.

ΔΕ3. Να γράψετε πρόγραμμα το οποίο διαβάζει το ονοματεπώνυμο, την τάξη και τη βαθμολογία σε τρία μαθήματα ενός μαθητή και υπολογίζει το μέσο όρο του σε αυτά τα μαθήματα. Στη συνέχεια εκτυπώνει το όνομα του μαθητή, το τμήμα του και το μέσο όρο.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Η μετατροπή της θερμοκρασίας από βαθμούς Celsius σε Fahrenheit δίνεται από τον τύπο:

$$F = \frac{9}{5}C + 32$$

Να γραφεί πρόγραμμα το οποίο να διαβάζει τη θερμοκρασία σε βαθμούς Celsius και να την υπολογίζει και να την τυπώνει σε βαθμούς Fahrenheit.

ΔΣ2. Η περίοδος ενός εκκρεμούς δίνεται από τον τύπο:

$$T = 2\pi \sqrt{\frac{L}{g}}$$

όπου L το μήκος του εκκρεμούς και g η επιτάχυνση της βαρύτητας.

Γράψτε πρόγραμμα το οποίο να υπολογίζει την περίοδο του εκκρεμούς. Το μήκος του εκκρεμούς θα δίνεται από το χρήστη κατά την εκτέλεση του προγράμματος.

ΔΣ3. Για να υπολογίσουμε τη ροή του αίματος στον ανθρώπινο οργανισμό χρησιμοποιούμε τον τύπο ροής υγρών σε σωλίνες. Για παράδειγμα, η ροή του αίματος στην αορτή (τη βασική αρτηρία που μεταφέρει αίμα σε όλα τα όργανα εκτός από τους πνεύμονες) υπολογίζεται από τον τύπο $POH = 5500\pi r^4$, όπου r η ακτίνα της αορτής. Μία υγιής αορτή έχει διάμετρο περίπου 0,02m. Η μείωση της διαμέτρου (στένωση) της αορτής προκαλεί σοβαρά καρδιαγγειακά νοσήματα αφού οποιαδήποτε στένωση προκαλεί πολύ μεγάλη μείωση της ροής αίματος. Για παράδειγμα, στένωση κατά 33% της αορτής προκαλεί μείωση κατά 80% της ροής του αίματος, με πολύ σοβαρές επιπλοκές στην υγεία του ανθρώπου.

Να γράψετε πρόγραμμα το οποίο να υπολογίζει τη ροή του αίματος σε μία φυσιολογική αορτή (με ακτίνα 0.01m) και την ποσοστιαία μεταβολή της ροής που επέρχεται με μείωση της ακτίνας της αορτής κατά 10%, 33% και 50%.

7.5. Τεστ αυτοαξιολόγησης

1. Δίνονται οι παρακάτω ομάδες προτάσεων. Σε καθεμία από αυτές, να βάλετε τις προτάσεις στη σωστή σειρά με την οποία θα πρέπει να γράφονται σε ένα πρόγραμμα:

- | | |
|---|-----------------------------------|
| 1. Α) Δήλωση μεταβλητών | Γ) Επικεφαλίδα προγράμματος |
| Β) Δήλωση σταθερών | Δ) Εντολή εισόδου ΔΙΑΒΑΣΕ |
| 2. Α) ΓΡΑΨΕ 'Η συνολική τιμή είναι', Τιμή | Γ) Κόστος <- N * 100 |
| Β) ΔΙΑΒΑΣΕ N | Δ) Τιμή <- Κόστος + Κόστος * 0.18 |

2. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

3. Τα στοιχεία προγράμματος των οποίων η τιμή μπορεί να μεταβληθεί κατά τη διάρκεια εκτέλεσης ενός προγράμματος ονομάζονται _____.
4. Η τελευταία εντολή κάθε προγράμματος είναι _____.

3. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

5. Η δήλωση των μεταβλητών που χρησιμοποιούνται σε ένα πρόγραμμα είναι υποχρεωτική.
6. Το σύμβολο της εντολής εκχώρησης είναι το ίσον =.
7. Κατά τον υπολογισμό μιας αριθμητικής παράστασης, πρώτα εκτελείται ο πολλαπλασιασμός και στη συνέχεια η πρόσθεση.
8. Οι λογικές μεταβλητές δέχονται μόνο δύο τιμές.

4. Διάλεξε ένα μεταξύ των προτεινόμενων:

9. Ποιες από τις παρακάτω εντολές δίνουν σαν αποτέλεσμα εκτέλεσης το μήνυμα: Η τιμή είναι 100

- | | |
|-------------------------------|----------------------------|
| A) Τιμή <- 100 | Γ) Τιμή <- 100 |
| ΓΡΑΨΕ 'Η τιμή είναι' 100 | ΓΡΑΨΕ 'Η τιμή είναι', 100 |
| B) ΓΡΑΨΕ 'Η τιμή είναι', Τιμή | Δ) Τιμή <- 100 |
| | ΓΡΑΨΕ 'Η τιμή είναι', Τιμή |

10. Μετά την εκτέλεση της εντολής $Y \leftarrow 5 * (X - 3) + X^3 - 2 + Z$ ποια είναι η τιμή της μεταβλητής Y, αν $X = 5$ και $Z = 1$:

- A) 35 B) 134 Γ) 22 Δ) 148

11. Τι θα τυπώσουν οι παρακάτω εντολές:

- A <- 100
 $X \leftarrow (2 + T_P(A) * 3 / 10)^2 - (A + 50) / 5$
ΓΡΑΨΕ X
- A) 22 B) -5 Γ) 10 Δ) 25

12. Σε ένα πρόγραμμα έχουμε μία μεταβλητή Πλήθος την οποία θέλουμε να την αυξήσουμε κατά μία μονάδα. Ποια από τις εντολές έχει σαν αποτέλεσμα την αύξηση αυτή:

- A) $\text{Πλήθος} + 1 \leftarrow \text{Πλήθος}$ Γ) $\text{Πλήθος} \leftarrow +1$
B) $\text{Πλήθος} \leftarrow \text{Πλήθος} + 1$ Δ) $\text{Πλήθος} = \text{Πλήθος} + 1$

5. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

13. Τα είδη μεταβλητών που υποστηρίζει η ΓΛΩΣΣΑ είναι:

- A) ακέραιες Γ) μιγαδικές Ε) ημερομηνίες
B) πραγματικές Δ) χαρακτήρες Ζ) λογικές

14. Ποια από τα παρακάτω είναι δεκτά σαν ονόματα σταθερών:

- A) A Γ) 1Στοιχείο Ε) Τιμή-σε-\$
B) Στοιχείο1 Δ) Φύλο μαθητή Ζ) ΤΑΧΥΤΗΤΑ

Επιλογή και Επανάληψη

8.1. Προσδοκώμενα αποτελέσματα

Σε αυτό το κεφάλαιο θα γνωρίσεις τις σημαντικότερες εντολές που θα χρησιμοποιείς σε όλα τα προγράμματά σου. Οι εντολές αυτές εκφράζουν τις βασικές δομές του δομημένου προγραμματισμού: τη δομή της επιλογής και τη δομή της επανάληψης ή ανακύκλωσης όπως συχνά θα την ακούσεις να λέγεται. Η επιλογή υλοποιείται με την εντολή **ΑΝ** και τις διάφορες μορφές της καθώς και με την εντολή **ΕΠΙΛΕΞΕ**, ενώ η επανάληψη με τις εντολές **ΟΣΟ_ΕΠΑΝΑΛΑΒΕ** και **ΜΕΧΡΙΣ_ΟΤΟΥ** καθώς και με την εντολή **ΓΙΑ**. Η σωστή γνώση της χρήσης αυτών των εντολών και η γνώση των διαφορών που παρουσιάζουν σου επιτρέπουν να επιλέγεις την καταλληλότερη για κάθε συγκεκριμένο πρόγραμμα.

Ο σκοπός σου δεν είναι να γράψεις απλά ένα πρόγραμμα το οποίο επιλύει το πρόβλημα, αλλά να χρησιμοποιήσεις τις εντολές που επιτρέπουν τη σύνταξη του πιο απλού, σύντομου, κατανοητού και τελικά αποδοτικότερου προγράμματος.

Οι λυμένες ασκήσεις του κεφαλαίου αυτού, όπως και του προηγούμενου, παρουσιάζονται στο περιβάλλον της ιδεατής γλώσσας προγραμματισμού ΓΛΩΣΣΑ και μερικές από αυτές παρουσιάζονται στα πραγματικά προγραμματιστικά περιβάλλοντα Basic και Pascal.

8.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

Για τη μέτρηση της ποιότητας της ατμόσφαιρας στην Αθήνα, όπως και σε κάθε μεγάλη πόλη που έχει πρόβλημα μόλυνσης της ατμόσφαιρας, μετρούνται συνεχώς τα επίπεδα συγκεκριμένων βλαβερών συστατικών της, που είναι γνωστά ως ρύποι. Οι ρύποι αυτοί είναι το διοξείδιο του αζώτου (NO_2), το μονοξείδιο του άνθρακα (CO), το διοξείδιο του θείου (SO_2), το όζον (O_3) και ο καπνός.

Για τον περιορισμό της ρύπανσης σε περιπτώσεις που σημειώνεται σημαντική αύξηση των τιμών των ρύπων χρησιμοποιούνται τα όρια εκτάκτων μέτρων.

Τα όρια αυτά που ισχύουν για την περιοχή της Αθήνας για δύο από τους πλέον συχνά εμφανιζόμενους ρύπους O_3 και NO_2 παρουσιάζονται στον παρακάτω πίνακα.

Ρύπος	Στάδιο Προειδοποίησης	Στάδιο λήψης μέτρων Α! βαθμίδας	Στάδιο λήψης μέτρων Β! βαθμίδας
NO_2 ($\mu\text{g}/\text{m}^3$)	400	500	700
O_3 ($\mu\text{g}/\text{m}^3$)	250	300	500

Να γραφεί πρόγραμμα το οποίο διαβάζει τις τιμές του NO_2 και του O_3 και να τυπώνει το αντίστοιχο μήνυμα σύμφωνα με τον παρακάτω πίνακα.

Κάτω από το στάδιο προειδοποίησης	Στάδιο Προειδοποίησης	Στάδιο λήψης μέτρων Α! βαθμίδας	Στάδιο λήψης μέτρων Β! βαθμίδας
ΡΥΠΟΙ ΜΕΣΑ ΣΤΑ ΟΡΙΑ	ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ	ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ	ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΑΠΑΓΟΡΕΥΣΗ ΚΥΚΛΟΦΟΡΙΑΣ

ΠΡΟΓΡΑΜΜΑ Ρύποι

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: NO_2 , O_3

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε την τιμή του Διοξειδίου του αζώτου'

ΔΙΑΒΑΣΕ NO_2

ΓΡΑΨΕ 'Δώσε την τιμή του Όζοντος'

ΔΙΑΒΑΣΕ O_3

ΑΝ $\text{NO}_2 > 700$ **Η** $\text{O}_3 > 500$ **ΤΟΤΕ**

ΓΡΑΨΕ 'ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΑΠΑΓΟΡΕΥΣΗ ΚΥΚΛΟΦΟΡΙΑΣ'

ΑΛΛΙΩΣ_ΑΝ $\text{NO}_2 > 500$ **Η** $\text{O}_3 > 300$ **ΤΟΤΕ**

ΓΡΑΨΕ 'ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ'

ΑΛΛΙΩΣ_ΑΝ $\text{NO}_2 > 400$ **Η** $\text{O}_3 > 250$ **ΤΟΤΕ**

ΓΡΑΨΕ 'ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Ρύποι μέσα στα όρια'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ



Η επιλογή του επιπέδου των ρύπων μπορεί να γίνει με πολλούς άλλους τρόπους. Μπορούν να χρησιμοποιηθούν εμφωλευμένα AN ή δύο διαφορετικές εντολές AN-ΑΛΛΙΩΣ_AN, ένα AN για το όζον και ένα δεύτερο για το διοξείδιο ή ακόμη και δύο εντολές ΕΠΙΛΕΞΕ.

Ο καλύτερος τρόπος λύσης εξαρτάται από το πρόβλημα και τα ζητούμενα αποτελέσματα. Η λύση που δόθηκε είναι η πιο σύντομη και η πιο απλή για το συγκεκριμένο πρόβλημα.

Περιβάλλον προγραμματισμού PASCAL

```
program rypoi;
var
    no2,o3:real;

begin
write ('ΔΩΣΕ ΤΗΝ ΤΙΜΗ ΤΟΥ ΔΙΟΞΕΙΔΙΟΥ:'); readln (no2);
write ('ΔΩΣΕ ΤΗΝ ΤΙΜΗ ΤΟΥ ΟΖΟΝΤΟΣ:'); readln (o3);
if (no2 > 700) or (o3 > 500) then
    write ('ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΑΠΑΓΟΡΕΥΣΗ ΚΥΚΛΟΦΟΡΙΑΣ')
else if (no2 > 500) or (o3 > 300) then
    write ('ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ')
else if (no2 > 400) or (o3 > 250) then
    write ('ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ')
else
    write ('Ρύποι μέσα στα όρια')
endif
end.
```

Περιβάλλον προγραμματισμού BASIC

```
' rypoi
INPUT "NO2=", NO2
INPUT "O3=", O3
IF NO2 > 700 OR O3 > 500 THEN
    PRINT "ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΑΠΑΓΟΡΕΥΣΗ ΚΥΚΛΟΦΟΡΙΑΣ"
ELSEIF NO2 > 500 OR O3 > 300 THEN
    PRINT "ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ2"
ELSEIF NO2 > 400 OR O3 > 250 THEN
    PRINT "ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ"
ELSE
    PRINT "Ρύποι μέσα στα όρια"
END IF
END
```

ΠΑΡΑΔΕΙΓΜΑ 2

Ο λογαριασμός του νερού είναι τριμηνιαίος και υπολογίζεται με βάση την κατανάλωση νερού. Η αξία του νερού υπολογίζεται από τον παρακάτω πίνακα.

Κατανάλωση/μήνα σε κυβικά μέτρα	Τιμή σε ευρώ
0-15	0,34
16-60	0,52
61-81	1,51
82-105	2,11
>105	2,64

Στην αξία του νερού προστίθεται το πάγιο (έστω 1,47€), η αποχέτευση 40% της αξίας του νερού, άλλες επιβαρύνσεις 1% καθώς και το ΦΠΑ που είναι 18% στο σύνολο του λογαριασμού.

Να γραφεί πρόγραμμα που διαβάζει το ονοματεπώνυμο του καταναλωτή, τον αριθμό του μετρητή νερού, την κατανάλωση (ανά τρίμηνο) και να υπολογίζει και να τυπώνει τα ποσά του λογαριασμού.

Η διαδικασία επαναλαμβάνεται συνεχώς για διάφορους καταναλωτές και τερματίζεται με την είσοδο του 0 ως αριθμού μετρητή.

ΠΡΟΓΡΑΜΜΑ ΛΟΓΑΡΙΑΣΜΟΣ_ΝΕΡΟΥ

ΣΤΑΘΕΡΕΣ

ΦΠΑ=0.18
ΤΙΜΗ1=0,34
ΤΙΜΗ2=0,52
ΤΙΜΗ3=1,51
ΤΙΜΗ4=2,11
ΤΙΜΗ5=2,64
ΠΑΓΙΟ=1,47

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Μετρητής

ΠΡΑΓΜΑΤΙΚΕΣ: Κατανάλωση, Τιμή, Αποχέτευση, Άλλα, Αξία_ΦΠΑ, Αξία,
& Τελική_Τιμή

ΧΑΡΑΚΤΗΡΕΣ: Όνομα, Επώνυμο

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε Αριθμό του μετρητή (0 για τέλος)'

ΔΙΑΒΑΣΕ Μετρητής

ΟΣΟ Μετρητής <> 0 **ΕΠΑΝΑΛΑΒΕ**

ΓΡΑΨΕ 'Δώσε το Ονοματεπώνυμο'

ΔΙΑΒΑΣΕ Επώνυμο, Όνομα

ΓΡΑΨΕ 'Δώσε την Κατανάλωση'

ΔΙΑΒΑΣΕ Κατανάλωση

ΕΠΙΛΕΞΕ Κατανάλωση

ΠΕΡΙΠΤΩΣΗ <= 15

```

Αξία <- Κατανάλωση*ΤΙΜΗ1
ΠΕΡΙΠΤΩΣΗ <= 60 ΤΟΤΕ
Αξία <- 15*ΤΙΜΗ1 + (Κατανάλωση-15)*ΤΙΜΗ2
ΠΕΡΙΠΤΩΣΗ <= 81 ΤΟΤΕ
Αξία <- 15*ΤΙΜΗ1 + 45*ΤΙΜΗ2 + (Κατανάλωση-60)*ΤΙΜΗ3
ΠΕΡΙΠΤΩΣΗ <= 105 ΤΟΤΕ
Αξία <- 15*ΤΙΜΗ1 + 45*ΤΙΜΗ2 + 21*ΤΙΜΗ3 + (Κατανάλωση-81)*ΤΙΜΗ4
ΠΕΡΙΠΤΩΣΗ_ΑΛΛΙΩΣ
Αξία <- 15*ΤΙΜΗ1 + 45*ΤΙΜΗ2 + 21*ΤΙΜΗ3 + 24*ΤΙΜΗ4 + (Κατανάλωση-105)*ΤΙΜΗ5

ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
Αποχέτευση <- Αξία*0.4
Άλλα <- Αξία*0.01
Τιμή <- Αξία + Αποχέτευση + Άλλα + ΠΑΓΙΟ
Αξία_ΦΠΑ <- Τιμή* ΦΠΑ
Τελική_τιμή <- Τιμή + Αξία_ΦΠΑ
ΓΡΑΨΕ 'Ο λογαριασμός του', Επώνυμο, ' είναι ', Τελική_τιμή
ΓΡΑΨΕ 'Αξία νερού:', Αξία
ΓΡΑΨΕ 'άλλα:', Αποχέτευση + Άλλα, 'ΦΠΑ:', Αξία_ΦΠΑ
ΓΡΑΨΕ 'Δώσε Αριθμό του επόμενου μετρητή (0 για τέλος)'
ΔΙΑΒΑΣΕ Μετρητής
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΠΑΡΑΔΕΙΓΜΑ 3

Στο παράδειγμα 1 του προηγούμενου κεφαλαίου με την μπάλα που εκτοξεύεται στον αέρα η γωνία βολής μπορεί να μεταβάλλεται από 20 έως 80 μοίρες σε βήματα των 10 μοιρών. Επίσης η αρχική ταχύτητα μπορεί να μεταβάλλεται από 10 μ/sec έως 40 μ/sec σε βήματα των 10 μ/sec.

Να γραφεί πρόγραμμα που να υπολογίζει την οριζόντια απόσταση (το βεληνεκές) για κάθε συνδυασμό γωνίας και αρχικής ταχύτητας.

Περιβάλλον προγραμματισμού ΓΛΩΣΣΑ

```

ΠΡΟΓΡΑΜΜΑ Τροχιά_μπάλας2

```

```

ΣΤΑΘΕΡΕΣ

```

```

    G = 9.81

```

```

ΜΕΤΑΒΛΗΤΕΣ

```

```

    ΠΡΑΓΜΑΤΙΚΕΣ: V0, VX0, VY0, Θ, Βεληνεκές

```

```

ΑΡΧΗ

```

```

    ΓΙΑ Θ ΑΠΟ 20 ΜΕΧΡΙ 80 ΜΕ_ΒΗΜΑ 10

```

```

        ΓΡΑΨΕ 'Γωνία:', Θ

```

```

        ΓΙΑ V0 ΑΠΟ 10 ΜΕΧΡΙ 40 ΜΕ_ΒΗΜΑ 10

```

```

            VX0 <- V0*ΣΥΝ(Θ)

```

```

            VY0 <- V0*ΗΜ(Θ)

```

```

            Βεληνεκές <- 2*VX0*VY0/G

```



```

        ΓΡΑΨΕ 'Ταχύτητα:',v0,'Βεληνεκές:',Βεληνεκές
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Περιβάλλον προγραμματισμού PASCAL

```

program ball_2;
const
    g=9.81;
    pi=3.14;
var
    v01,a:integer;
    range,akt,v,v0,vx0,vy0:real;
begin
    for a:=2 to 8 do
        begin
            {μετατροπή της γωνίας σε ακτίνια}
            akt:=a*10*pi/180;
            writeln ('ΓΩΝΙΑ :',a*10:5);
            for v01:=1 to 4 do
                begin
                    v0:=v01*10;
                    vx0:=v0*cos(akt);
                    vy0:=v0*sin(akt);
                    range:=2*v0*vy0/g;
                    writeln('ΜΕ ΑΡΧΙΚΗ ΤΑΧΥΤΗΤΑ:',v0:5,'ΒΕΛΗΝΕΚΕΣ:',range:7:2);
                end;
            end;
        end.

```

Επειδή η Pascal δεν επιτρέπει τον καθορισμό του βήματος για τις επαναλήψεις που υλοποιούνται με την εντολή **For**, το βήμα είναι πάντα 1 ή -1, πρέπει να χρησιμοποιηθεί κάποιο τέχνασμα.

Έτσι αντί η γωνία βολής να μεταβάλλεται από 10 έως 80 με βήματα των 10 μοιρών, η μεταβλητή *a* μεταβάλλεται από 2 έως 8, αυξανόμενο κατά μονάδα σε κάθε επανάληψη και στη συνέχεια πολλαπλασιάζεται με 10 στη μετατροπή σε ακτίνια: $akt:=a*10*pi/180$.

Αντίστοιχα για την αρχική ταχύτητα η μεταβλητή *v01* παίρνει τιμές από 1 ως 4 και στη συνέχεια πολλαπλασιάζεται με το 10 για να δώσει την αρχική ταχύτητα, $v0:=v01*10$.

Περιβάλλον προγραμματισμού BASIC

```

' Τροχιά μπάλλας 2
g=9.81
FOR a = 20 TO 80 STEP 10
    akt = a * ATN(1) * 4 / 180
    PRINT "Γωνία ";akt
    FOR v0 = 10 TO 40 STEP 10
        vx0 = v0 * COS(akt)
        vy0 = v0 * SIN(akt)

```

```

    vel = 2 * vx0 * vy0 / g
    PRINT "ΜΕ ΑΡΧΙΚΗ ΤΑΧΥΤΗΤΑ";v0
    PRINT "ΒΕΛΗΝΕΚΕΣ = "; vel
NEXT v0
NEXT a
END

```



Η συνάρτηση ATN επι-
στρέφει το τόξο εφαπτο-
μένης. Άρα $ATN(1)=\pi/4$,
αφού $\epsilon\phi(\pi/4)=1$.



8.3. Συμβουλές - υποδείξεις

Εφόσον όπως έχουμε αναφέρει πολλές φορές κάθε πρόγραμμα μπορεί να υλοποιηθεί με τη χρήση των τριών δομών της ακολουθίας, της επιλογής και της επανάληψης, αν μάθεις να χρησιμοποιείς σωστά τις εντολές επιλογής και επανάληψης, μπορείς να υλοποιήσεις σχεδόν οποιονδήποτε αλγόριθμο. Στην πραγματικότητα όμως μόνο η εξάσκηση και η πείρα θα σου εξασφαλίσουν τη δυνατότητα να συντάσσεις εύκολα και γρήγορα σωστά προγράμματα. Οι παρακάτω συμβουλές θα σε βοηθήσουν στη συγγραφή σωστών προγραμμάτων αποφεύγοντας μερικά από τα πιο συνηθισμένα λάθη που παρουσιάζονται.

Όταν χρησιμοποιείς σύνθετες λογικές εκφράσεις, να προσέχεις την ιεραρχία των τελεστών. Είναι καλύτερο να χρησιμοποιείς πάντα παρενθέσεις, έστω και αν δεν είναι απαραίτητο, σε προφυλάσσει από πιθανά λάθη και αβλεψίες, ενώ ταυτόχρονα κάνει το πρόγραμμα πιο εύκολο στην κατανόησή του.

Πριν χρησιμοποιήσεις εμφωλευμένα ΑΝ, σκέψου μήπως το ίδιο πρόγραμμα μπορεί να υλοποιηθεί απλούστερα με σύνθετες λογικές εκφράσεις, την εντολή ΑΝ-ΑΛΛΙΩΣ_ΑΝ ή κάποια άλλη εντολή επιλογής που πιθανόν να προσφέρει το υπολογιστικό περιβάλλον που χρησιμοποιείς.

Οι μεταβλητές που ελέγχουν την επανάληψη του βρόχου ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ πρέπει υποχρεωτικά να αλλάζουν τιμή μέσα στο σώμα του βρόχου, αλλιώς ή δεν εκτελείται ποτέ ή συνηθέστερα δεν σταματάει η εκτέλεσή του (ατέρμων βρόχος).

Οι επαναλήψεις που υλοποιούνται με την εντολή ΟΣΟ, μπορεί να μην εκτελεστούν ούτε μία φορά, αφού ο έλεγχος γίνεται στην είσοδο του βρόχου, αντίθετα οι επαναλήψεις ΜΕΧΡΙΣ_ΟΤΟΥ θα πραγματοποιηθούν τουλάχιστον μία φορά.

Η εντολή ΓΙΑ χρησιμοποιείται μόνο για προκαθορισμένο αριθμό επαναλήψεων. Αν λοιπόν ξέρεις τον αριθμό των επαναλήψεων ή μπορείς να τον υπολογίσεις, τότε να χρησιμοποιείς την εντολή ΓΙΑ.

Ποτέ μη χρησιμοποιείς εντολές που αλλάζουν την αρχική τιμή, την τελική τιμή, το βήμα ή τη μεταβλητή που ελέγχει την επανάληψη μέσα σε ένα βρόχο ΓΙΑ. Αν και μερικές γλώσσες προγραμματισμού επιτρέπουν αυτές τις αλλαγές, να τις αποφεύγεις, γιατί οδηγούν σε προγράμματα δυσνόητα και συνήθως λανθασμένα.

8.4. Δραστηριότητες - ασκήσεις

ΔΤ1. Αν η μεταβλητή A έχει την τιμή 10, η μεταβλητή B έχει την τιμή 5 και η μεταβλητή Γ έχει την τιμή 3, ποιες από τις παρακάτω εκφράσεις είναι αληθείς και ποιες ψευδείς.

- A) ΟΧΙ ($A > B$)
- B) $A > B$ ΚΑΙ $A < \Gamma$ Η $\Gamma \leq B$
- Γ) $A > B$ ΚΑΙ ($A < \Gamma$ Η $\Gamma \leq B$)
- Δ) $A = B$ Η $(\Gamma - B) < 0$
- Ε) ($A > B$ ΚΑΙ $\Gamma < B$) Η ($B < \Gamma$ ΚΑΙ $A < \Gamma$)

ΔΤ2. Να γράψεις τις εντολές για τα παρακάτω:

- A) Αν η Βαθμολογία (ΒΑΘΜΟΣ) είναι μεγαλύτερη από το Μέσο όρο (ΜΟ), τότε να τυπώνει "Πολύ καλά", αν είναι ίση ή μικρότερη του Μέσου όρου μέχρι και 2 μονάδες, να τυπώνει "Καλά", και, όταν είναι μικρότερη του Μέσου όρου περισσότερο από 2 μονάδες, να τυπώνει "Μέτρια".
- B) Αν το τμήμα (ΤΜΗΜΑ) είναι Γ1 και η βαθμολογία (ΒΑΘΜΟΣ) είναι μεγαλύτερη από 15, τότε να τυπώνει το επώνυμο (ΕΠΩΝΥΜΟ).
- Γ) Αν η απάντηση (ΑΠΑΝΤΗΣΗ) δεν είναι Ν ή ν ή Ο ή ο, τότε να τυπώνει το μήνυμα "Λάθος απάντηση...".
- Δ) Αν ο αριθμός X είναι αρνητικός ή το $HM(X)=0$, τότε να τυπώνεται το μήνυμα "Λάθος δεδομένα...", αλλιώς να υπολογίζεται η παράσταση $(X^2 + 5 * X) / (T_P(X) * HM(X))$.

ΔΤ3. Τι αλλαγές πρέπει να γίνουν στο πρόγραμμα του παραδείγματος 1 ώστε να τυπώνει και ποιος από τους δύο ρύπους υπερέβη τα όρια λήψης μέτρων.

ΔΤ4. Έστω το παρακάτω τμήμα προγράμματος:

```
K <- 0
ΓΙΑ I ΑΠΟ 0 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 5
    A <- I^3
    K <- K+A
    ΓΡΑΨΕ I, A
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ K
```

Πόσες φορές θα εκτελεστεί ο βρόχος;

Ποια η λειτουργία των εντολών;

Γράψτε τις παραπάνω εντολές χρησιμοποιώντας την εντολή επανάληψης ΟΣΟ και την εντολή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ. Ποιον από τους τρεις τρόπους προτιμάς και γιατί;

ΔΤ5. Διάβασε προσεκτικά τα παρακάτω τμήματα προγράμματος. Ποια είναι τα λάθη; Διόρθωσέ τα, ώστε να λειτουργούν σωστά.

A)

```
ΔΙΑΒΑΣΕ Μισθός
ΟΣΟ Μισθός <>0 ΕΠΑΝΑΛΑΒΕ
    Αθροισμα <- 0
    ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Αθροισμα <- Αθροισμα+Μισθός
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    Αθροισμα <- 0
    ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Αθροισμα <- Αθροισμα+Μισθός
    ΔΙΑΒΑΣΕ Μισθός
ΜΕΧΡΙΣ_ΟΤΟΥ Μισθός<>0
```

Γ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    Αθροισμα <- 0
    ΔΙΑΒΑΣΕ Μισθός
    ΑΝ Μισθός > Μέγιστος ΤΟΤΕ
        Μέγιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ
        Ελάχιστος <- Μισθός
    ΤΕΛΟΣ_ΑΝ
    Αθροισμα <- Αθροισμα+Μισθός
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Εκτέλεσε εικονικά τις εντολές στο χαρτί και σημείωνε τα αποτελέσματα που προκύπτουν. Με αυτόν τον τρόπο θα δεις τα λάθη και στη συνέχεια θα κάνεις τις διορθώσεις.

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1. Να γραφεί πρόγραμμα που να διαβάζει το βαθμό ενός μαθητή και να υπολογίζει την αντίστοιχη αξιολόγησή του με βάση το βαθμό του και σύμφωνα με τον παρακάτω πίνακα:

17,5 – 20	Άριστα
15,5 – 17,4	Πολύ καλά
13,5 – 15,4	Καλά
9,5 – 13,4	Μέτρια
0 – 9,4	Απορρίπτεται

Το πρόγραμμα να γραφεί με τους ακόλουθους τρόπους:

- Με εντολές AN ... TOTE
- Με εντολές AN ... TOTE ... ΑΛΛΙΩΣ_ΑΝ
- Με εμφωλευμένα AN
- Με την εντολή ΕΠΙΛΕΞΕ

ΔΕ2. Στο κεφάλαιο 2 του βιβλίου σου παρουσιάστηκε και συζητήθηκε αναλυτικά ο Πολλαπλασιασμός αλά Ρωσικά. Να γράψεις πρόγραμμα που να υλοποιεί τον αλγόριθμο αυτό. Το πρόγραμμα να εκτελεστεί για διάφορα ζεύγη τιμών.

ΔΕ3. Να γραφεί πρόγραμμα το οποίο θα εκτελεί κάποια από τις βασικές πράξεις πρόσθεσης, αφαίρεσης, πολλαπλασιασμού και διαίρεσης ανάμεσα σε δύο ακέραιους αριθμούς και θα εμφανίζει το αποτέλεσμα στην οθόνη.

Το πρόγραμμα θα ελέγχεται από το παρακάτω μενού επιλογής και θα σταματάει όταν ο χρήστης επιλέξει από το μενού την επιλογή έξοδο.

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

Δώσε επιλογή:___

ΔΕ4. Να επεκτείνεις το παράδειγμα 1, τον υπολογισμό της ατμοσφαιρικής ρύπανσης, έτσι ώστε να παίρνει 6 τιμές ανά ώρα από 5 διαφορετικούς σταθμούς μέτρησης για τους δύο ρύπους. Το πρόγραμμα

- να υπολογίζει τη μέση τιμή κάθε ρύπου ανά ώρα και ανά σταθμό
- να βρίσκει τη μέγιστη μέση τιμή για κάθε ρύπο
- να ελέγχει τις μέγιστες αυτές τιμές με τα όρια που δόθηκαν

Το πρόγραμμα να εκτελεστεί με δεδομένα τις πραγματικές τιμές ρύπων που μετρήθηκαν τη χθεσινή ημέρα. Οι τιμές αυτές δίδονται από το τμήμα ποιότητας της ατμόσφαιρας του ΥΠΕΧΩΔΕ και βρίσκονται στη διεύθυνση: www.minenv.gr

ΔΕ5. Να γραφεί πρόγραμμα το οποίο να υπολογίζει τη συνολική χωρητικότητα πυκνωτών και τη συνολική αντίσταση αντιστάσεων. Η συνολική αντίσταση R και η συνολική χωρητικότητα C δίνεται από τους τύπους:

Σε σειρά

$$R = R_1 + R_2 + R_3 + \dots$$

$$C = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \dots$$

Σε παράλληλη

$$C = C_1 + C_2 + C_3 + \dots$$

$$R = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots$$

Το πρόγραμμα θα ελέγχεται από μενού επιλογής και θα τερματίζεται όταν ο χρήστης επιλέξει έξοδο.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Η φορολογία εισοδήματος φυσικών προσώπων υπολογίζεται από τις αρμόδιες υπηρεσίες του Υπουργείου των Οικονομικών κλιμακωτά, με τη βοήθεια του παρακάτω πίνακα.

ΚΛΙΜΑΚΑ ΥΠΟΛΟΓΙΣΜΟΥ ΦΟΡΟΥ ΕΙΣΟΔΗΜΑΤΟΣ ΦΥΣΙΚΩΝ ΠΡΟΣΩΠΩΝ
ΟΙΚΟΝ. ΕΤΟΥΣ 1999

Κλιμάκιο εισοδήματος	Φορολογικός συντελεστής	Φόρος κλιμακίου	Σύνολο	
			εισοδήματος	φόρου
1.055.000	0	0	1.055.000	0
1.582.500	5	79.125	2.637.500	79.125
1.582.500	15	237.375	4.220.000	316.500
3.165.000	30	949.500	7.385.000	1.266.000
8.440.000	40	3.376.000	15.825.000	4.642.000
Υπερβάλλον	45			

Για κάθε φορολογούμενο δίνονται τα εξής στοιχεία: αριθμός φορολογικού μητρώου (ΑΦΜ), όνομα φορολογούμενου, φορολογητέο εισόδημα.

Να γραφεί πρόγραμμα το οποίο:

Να διαβάζει τα στοιχεία των φορολογουμένων, να υπολογίζει και να τυπώνει το φόρο που τους αντιστοιχεί. Το πρόγραμμα θα διαβάζει τα στοιχεία πολλών φορολογουμένων και θα τελειώνει όταν διαβάζει για ΑΦΜ τον αριθμό 0.

ΔΣ2. Να γραφεί πρόγραμμα που να υπολογίζει τις ρίζες της δευτεροβάθμιας εξίσωσης $ax^2 + bx + \gamma = 0$. Αν δεν υπάρχουν πραγματικές ρίζες, να εκτυπώνει αντίστοιχο μήνυμα.

ΔΣ3. Να γραφεί πρόγραμμα το οποίο διαβάζει το όνομα ενός μαθητή, τους βαθμούς του σε τρία μαθήματα και υπολογίζει και τυπώνει το μέσο όρο. Το πρόγραμμα να σταματάει, όταν για όνομα δοθεί το κενό.

ΔΣ4. Να γράψετε πρόγραμμα που να υπολογίζει τη συνάρτηση $y(x) = x^2 - 3x + 2$ για όλες τις τιμές του x από -1 έως 3 σε βήματα του 0.1 .

ΔΣ5. Ένας τρόπος υπολογισμού των τριγωνομετρικών συναρτήσεων που χρησιμοποιείται συχνά από τους υπολογιστές είναι με τον υπολογισμό των παρακάτω σειρών:

$$\eta\mu x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$\sigma\upsilon\nu x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

Να γράψεις πρόγραμμα το οποίο να διαβάζει τη γωνία x σε μοίρες και να υπολογίζει το ημίτονο και το συνημίτονό της σύμφωνα με τους παραπάνω τύπους.

Ποια μπορεί να είναι τα κριτήρια για διακοπή των επαναλήψεων;

Υπόδειξη: Να μετατρέψεις αρχικά τη γωνία x σε ακτίνια.

ΔΣ6. Να γραφεί ένα πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τον αναλύει σε γινόμενο πρώτων παραγόντων.

8.5. Τεστ αυτοαξιολόγησης

1. Δίνονται οι παρακάτω ομάδες εντολών. Σε καθεμία από αυτές, να βάλετε τις εντολές στη σωστή σειρά με την οποία θα πρέπει να γράφονται σε ένα πρόγραμμα.

1.

- A) ΓΡΑΨΕ 'Δεν υπάρχει ρίζα'
- B) ΑΝ $A > 0$ ΤΟΤΕ
- Γ) ΤΕΛΟΣ_ΑΝ
- Δ) ΑΛΛΙΩΣ
- Ε) $\text{Ρίζα} < -T_P(A)$

2.

- A) ΜΕΧΡΙΣ_ΟΤΟΥ (Απάντηση='Ν' Ή Απάντηση='ν')
- B) ΔΙΑΒΑΣΕ Απάντηση
- Γ) ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
- Δ) ΓΡΑΨΕ 'Δώσε απάντηση :'

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

- 3. Οι εντολές που βρίσκονται σε ένα βρόχο ΟΣΟ ΕΠΑΝΑΛΑΒΕ εκτελούνται τουλάχιστον μία φορά.
- 4. Η τιμή του βήματος στην εντολή ΓΙΑ είναι υποχρεωτική να αναγράφεται.

5. Κάθε εντολή ΑΝ πρέπει να έχει την αντίστοιχη εντολή ΤΕΛΟΣ_ΑΝ.
6. Κάθε βρόχος που υλοποιείται με την εντολή ΟΣΟ ΕΠΑΝΑΛΑΒΕ μπορεί να γραφεί και με χρήση της εντολής ΓΙΑ.
7. Αν το Α έχει την τιμή 5 και το Β την τιμή 6, τότε η λογική έκφραση $A > 5 \vee A < 3 \wedge B > 5$ είναι ψευδής.

3.

Διάλεξε ένα μεταξύ των προτεινόμενων:

8. Ποιο από τα παρακάτω υπολογίζει το άθροισμα των περιττών αριθμών που υπάρχουν στους 100 πρώτους ακέραιους.

A)

```
Αθροισμα <- 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100
    Αθροισμα <- Αθροισμα+Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
Αθροισμα <- 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
    Αθροισμα <- Αθροισμα+Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
    Αθροισμα <- 0
    Αθροισμα <- Αθροισμα+Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2
    Αθροισμα <- Ι
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

9. Τι θα εκτυπώσει το παρακάτω τμήμα προγράμματος:

```
A <- 0
ΓΙΑ Ι ΑΠΟ 10 ΜΕΧΡΙ 20 ΜΕ_ΒΗΜΑ 10
    A <- A+I^2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ A
```

A) 0 B) 100 Γ) 500 Δ) 400

10. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη:

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    A <- 0
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
        A <- A-1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΕΧΡΙΣ_ΟΤΟΥ A=0
```

A) 10 B) 0 Γ) 5 Δ) Άπειρες

11. Δίνονται οι παρακάτω εντολές:

```
A <- 1
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2
    A <- A*I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Ποιες από τις επόμενες ομάδες εντολών δίνουν στο A την ίδια τιμή:

A)

```
A <- 1
I <- 1
ΟΣΟ I <= 10 ΕΠΑΝΑΛΑΒΕ
    I <- I+2
    A <- A*I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
A <- 1
I <- 1
ΟΣΟ I <=10 ΕΠΑΝΑΛΑΒΕ
    A <- A*I
    I <- I+2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ)

```
A <- 1
I <- 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    A <- A*I
    I <- I+2
ΜΕΧΡΙΣ_ΟΤΟΥ I<10
```

Δ)

```
A <- 1
I <- 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    A <- A*I
    I <- I+2
ΜΕΧΡΙΣ_ΟΤΟΥ I=10
```

12. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη:

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ 3
    ΓΡΑΨΕ 'Μήνυμα'
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

A) 2 B) 0 Γ) 1 Δ) Άπειρες

13. Ποια η λειτουργία του παρακάτω τμήματος προγράμματος:

```
B <- 10
ΔΙΑΒΑΣΕ A
B <- A
ΑΝ A < 0 ΤΟΤΕ
    B <- -A
ΤΕΛΟΣ_ΑΝ
A <- 0
ΓΡΑΨΕ B
```

- A) Τυπώνει τον αριθμό που διάβασε.
B) Τυπώνει την απόλυτη τιμή του αριθμού που διάβασε.
Γ) Τυπώνει πάντα την τιμή 0.
Δ) Τυπώνει πάντα την τιμή 10.

Πίνακες

9.1. Προσδοκώμενα αποτελέσματα

Στο κεφάλαιο αυτό θα μάθεις να χρησιμοποιείς στα προγράμματά σου τους πίνακες για την αποθήκευση μεγάλου αριθμού δεδομένων ιδίου τύπου.

Αρχικά πρέπει να αποφασίζεις αν η χρήση της δομής του πίνακα σε βοηθάει στην υλοποίηση του προγράμματός σου.

Στη συνέχεια πρέπει να επιλέγεις το είδος του πίνακα που χρειάζεται και να μπορείς να τον ορίσεις σωστά, αλλά και να χειριστείς σωστά τα στοιχεία του. Συγκεκριμένα πρέπει να μπορείς να εισάγεις, να επεξεργάζεσαι και να τυπώνεις τα στοιχεία ενός πίνακα τόσο μονοδιάστατου όσο και δισδιάστατου.

Οι επεξεργασίες που απαιτούνται σε έναν πίνακα είναι συνήθως η αναζήτηση, η ταξινόμηση και η συγχώνευση. Μερικούς από τους αλγορίθμους για τις βασικές αυτές επεξεργασίες τις γνώρισες στο κεφάλαιο 3 και 4, εδώ θα έχεις την ευκαιρία να τους υλοποιήσεις σε προγραμματιστικό περιβάλλον.

Οι λυμένες ασκήσεις του κεφαλαίου αυτού, όπως και των προηγούμενων, παρουσιάζονται στο περιβάλλον της ιδεατής γλώσσας προγραμματισμού ΓΛΩΣΣΑ και μερικές από αυτές παρουσιάζονται στα πραγματικά προγραμματιστικά περιβάλλοντα Basic και Pascal.

9.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

Να γραφεί πρόγραμμα το οποίο διαβάζει τα ονόματα 50 αεροπορικών εταιρειών και τις αντίστοιχες εισπράξεις τους. Να τυπώνει τα ονόματα των εταιρειών που έχουν εισπράξεις περισσότερες από το μέσο όρο.

ΠΡΟΓΡΑΜΜΑ Αεροπορικές_εταιρείες

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: N, I, Εισπράξεις[50], Σύνολο

ΠΡΑΓΜΑΤΙΚΕΣ: MO

ΧΑΡΑΚΤΗΡΕΣ: Εταιρεία[50]

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Αριθμός εταιρειών.. (μικρότερο από 50)'

ΔΙΑΒΑΣΕ N

ΜΕΧΡΙΣ_ΟΤΟΥ N<=50

Σύνολο <- 0

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΓΡΑΨΕ 'Δώσε αεροπορική εταιρεία ...'

ΔΙΑΒΑΣΕ Εταιρεία[I]

ΓΡΑΨΕ 'Δώσε εισπράξεις ...'

ΔΙΑΒΑΣΕ Εισπράξεις [I]

Σύνολο <- Σύνολο+Εισπράξεις[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

MO <- Σύνολο/N

ΓΡΑΨΕ 'Μεγαλύτερες από το μέσο όρο'

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΑΝ Εισπράξεις[I]> MO **ΤΟΤΕ**

ΓΡΑΨΕ Εταιρεία[I]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Οι παραπάνω πίνακες λέγονται παράλληλοι. Δύο ή περισσότεροι πίνακες λέγονται παράλληλοι, αν σε αυτούς έχουμε αποθηκεύσει τα χαρακτηριστικά οντοτήτων με τέτοιο τρόπο ώστε τα δεδομένα κάθε οντότητας να βρίσκονται σε στοιχεία με την ίδια τιμή δείκτη.

Στο παραπάνω παράδειγμα οι πίνακες Εισπράξεις και Εταιρεία είναι παράλληλοι αφού τα στοιχεία που αναφέρονται σε κάθε γραμμή τους, δηλαδή το όνομα και οι εισπράξεις, αφορούν την ίδια εταιρεία.

Περιβάλλον προγραμματισμού PASCAL

```
program air_co;
var
  n,i,sum:integer;
  ave:real;
  tickets:array[1..50] of integer;
  company:array[1..50] of string;
begin
  repeat
    write('ΑΡΙΘΜΟΣ ΕΤΑΙΡΕΙΩΝ :');readln(n);
  until (n<=50);
  sum:=0;
  for i:=1 to n do
  begin
    write('ΕΤΑΙΡΕΙΑ :'); readln(company[i]);
    write('ΕΙΣΠΡΑΞΕΙΣ :');readln(tickets[i]);
    sum:=sum+tickets[i];
  end;
  ave:=sum/n;
  for i:=1 to n do
  begin
    if tickets[i] > ave then
      writeln (company[i]);
    end;
  end.
end.
```

Περιβάλλον προγραμματισμού Basic

```
` Αεροπορικές εταιρίες
DIM company$(50), E(50)
DO
  INPUT "Αριθμός εταιριών:", n
LOOP UNTIL n <= 50
sum = 0
FOR i = 1 TO n
  PRINT "Εταιρία; i; " :";
  INPUT "", company$(i)
  PRINT "Εισπράξεις=";
  INPUT "", E(i)
  sum = sum + E(i)
NEXT i
MO = sum / n
PRINT "Μεγαλύτερες από το μέσο όρο"
PRINT "=====
FOR i = 1 TO n
  IF E(i) > MO THEN PRINT company$(i)
NEXT i
END
```

ΠΑΡΑΔΕΙΓΜΑ 2

Μία εταιρεία κατασκευής αυτοκινήτων έχει μετρήσεις από το επίπεδο θορύβου όλων των μοντέλων της (σε decibel -dB). Οι μετρήσεις γίνονται για διαφορετικές ταχύτητες και δίνονται από τον παρακάτω πίνακα:

Μοντέλο	Ταχύτητα (km/h)				
	40	60	80	100	120
GX	88	90	93	105	112
LX	75	78	81	89	95
Gti	80	85	90	96	101
SX	68	78	85	102	105

Να γραφεί πρόγραμμα το οποίο θα υπολογίζει και θα τυπώνει το μέσο επίπεδο θορύβου για κάθε μοντέλο, το μέσο επίπεδο θορύβου για κάθε ταχύτητα και το συνολικό μέσο επίπεδο θορύβου όλων των αυτοκινήτων.

ΠΡΟΓΡΑΜΜΑ Αυτοκίνητα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Θόρυβος[4, 5], I, J, Ταχύτητα[5], Αθροισμα, Συν_Αθροισμα

ΧΑΡΑΚΤΗΡΕΣ: Μοντέλο[4]

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, Συν_ΜΟ

!Εισαγωγή δεδομένων

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5
  ΓΡΑΨΕ 'Δώσε ταχύτητα.. '
  ΔΙΑΒΑΣΕ Ταχύτητα[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4
  ΓΡΑΨΕ 'Δώσε μοντέλο.. '
  ΔΙΑΒΑΣΕ Μοντέλο[I]
  ΓΡΑΨΕ 'Δώσε επίπεδα θορύβου.. '
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
    ΔΙΑΒΑΣΕ Θόρυβος[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

! Υπολογισμοί μέσων τιμών

```
Συν_Αθροισμα <- 0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4
  Αθροισμα <- 0
  ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
    Αθροισμα <- Αθροισμα+Θόρυβος[I, J]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Συν_Αθροισμα <- Συν_Αθροισμα+Αθροισμα
ΜΟ <- Αθροισμα/5
  ΓΡΑΨΕ Μοντέλο[I], ' : ', ΜΟ
```

```

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5
    Αθροισμα <- 0
    ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4
        Αθροισμα <- Αθροισμα+Θόρυβος[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΜΟ <- Αθροισμα/4
    ΓΡΑΨΕ Ταχύτητα[J] ' :', ΜΟ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    Συν_ΜΟ <- Συν_Αθροισμα/20
    ΓΡΑΨΕ 'Συνολικό μέσο επίπεδο θορύβου :', Συν_ΜΟ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΠΑΡΑΔΕΙΓΜΑ 3

Δίνονται δύο ταξινομημένοι κατά αύξουσα σειρά μονοδιάστατοι πίνακες, ακεραίων αριθμών. Να γραφεί πρόγραμμα το οποίο να συγχωνεύει τους δύο πίνακες σε έναν τρίτο ο οποίος να είναι επίσης ταξινομημένος κατά αύξουσα σειρά. Οι δύο αρχικοί πίνακες δεν μπορούν να περιέχουν περισσότερα από 100 στοιχεία ο καθένας.

Η συγχώνευση είναι μία βασική λειτουργία των πινάκων και γενικότερα των δομών δεδομένων. Στη συνέχεια δίνεται ένας πολύ απλός αλγόριθμος συγχώνευσης δύο ταξινομημένων πινάκων σε έναν τρίτο ταξινομημένο πίνακα.

Θεωρείται ότι στην είσοδο του αλγορίθμου συγχώνευσης δίνονται δύο ταξινομημένοι, κατά αύξουσα σειρά, πίνακες Α και Β, μεγέθους Ν και Μ στοιχείων αντίστοιχα, ενώ στην έξοδο προκύπτει ένας τρίτος πίνακας Γ με Ν+Μ ταξινομημένα στοιχεία επίσης κατά αύξουσα σειρά.

Στο πρόγραμμα Συγχώνευση που ακολουθεί οι μεταβλητές i, j και k είναι δείκτες για την κίνηση μέσα στους πίνακες Α, Β και Γ. Η μέθοδος προχωρεί ως εξής:

Το μικρότερο στοιχείο από τους πίνακες Α και Β τοποθετείται στον πίνακα Γ με ταυτόχρονη αύξηση του αντίστοιχου δείκτη. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου τελειώσουν τα στοιχεία του ενός πίνακα.

Στη συνέχεια τα υπόλοιπα στοιχεία του άλλου πίνακα μεταφέρονται στον πίνακα Γ.

ΠΡΟΓΡΑΜΜΑ Συγχώνευση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α[100], Β[100], Γ[200], I, J, K, N, M, Λ

! Α και Β αρχικοί πίνακες

! Γ τελικός πίνακας

ΑΡΧΗ

! Διάβασε τα δεδομένα

ΓΡΑΨΕ 'Δώσε το πλήθος των στοιχείων του πίνακα Α (<100)'

ΔΙΑΒΑΣΕ N

ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** N

ΔΙΑΒΑΣΕ A[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

ΓΡΑΨΕ 'Δώσε το πλήθος των στοιχείων του πίνακα Β(<100)'
ΔΙΑΒΑΣΕ Μ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Μ
    ΔΙΑΒΑΣΕ Β[Ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Συγχώνευση πινάκων

! Ι είναι ο δείκτης για τον πίνακα Α
! J είναι ο δείκτης για τον πίνακα Β
! Κ είναι ο δείκτης για τον πίνακα Γ
Ι <- 1
J <- 1
Κ <- 1
ΟΣΟ Ι <= Ν ΚΑΙ J <= Μ ΕΠΑΝΑΛΑΒΕ
! Όσο και τα δύο έχουν στοιχεία
    ΑΝ Α[Ι] < Β[J] ΤΟΤΕ
        Γ[Κ] <- Α[Ι]
        Κ <- Κ+1
        Ι <- Ι+1
    ΑΛΛΙΩΣ
        Γ[Κ] <- Β[J]
        Κ <- Κ+1
        J <- J +1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Μεταφορά των υπολοίπων στοιχείων του Α ή του Β
ΑΝ Ι > Ν ΤΟΤΕ
    ΓΙΑ Λ ΑΠΟ Κ ΜΕΧΡΙ Ν+Μ
        Γ[Λ] <- Β[J]
        J <- J +1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ
    ΓΙΑ Λ ΑΠΟ Κ ΜΕΧΡΙ Ν+Μ
        Γ[Λ] <- Α[Ι]
        Ι <- Ι+1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Εκτύπωση τελικού πίνακα

ΓΙΑ Λ ΑΠΟ 1 ΜΕΧΡΙ Ν+Μ
    ΓΡΑΨΕ Γ[Λ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Γλώσσα προγραμματισμού PASCAL

```
program merge;

var
  i,j,k,l,n,m: integer;
  a,b: array[1..100] of integer;
  c: array[1..200] of integer;

{A και B αρχικοί ταξινομημένοι πίνακες C τελικός πίνακας}

begin
  write ('Δώσε τη διάσταση του πίνακα A (n) '); readln(n);
  for i:=1 to n do
    readln(a[i]);
  write ('Δώσε τη διάσταση του πίνακα B (m) '); readln (m);
  for i:=1 to m do
    readln (b[i]);
  i:=1; j:=1; k:=1;
  while (i<=n) and (j<=m) do
    if (a[i]<b[j]) then
      begin
        c[k]:=a[i];k:=k+1;i:=i+1;
      end
    else
      begin
        c[k]:=b[j]; k:=k+1; j:=j+1;
      end;
  if i>n then
    for l:=k to n+m do
      begin
        c[l]:=b[j]; j:=j+1;
      end
    else
      for l:=k to n+m do
        begin
          c[l]:=a[i]; i:=i+1;
        end;
    for l:=1 to n+m
      write (c[l]);
end.
```

Περιβάλλον προγραμματισμού Basic

```
' Merging
DIM a(100), b(100), c(200)
READ n
FOR i = 1 TO n: READ a(i): NEXT i
DATA 5
DATA 2,7,12,18,26
```



```

READ m
FOR i = 1 TO m: READ b(i): NEXT i
DATA 5
DATA 1,6,10,15,25
i = 1: j = 1: k = 1
WHILE i <= n AND j <= m
    IF a(i) < b(j) THEN
        c(k) = a(i): k = k + 1: i = i + 1
    ELSE
        c(k) = b(j): j = j + 1: k = k + 1
    END IF
WEND
IF i > n THEN
    FOR r = k TO n + m
        c(r) = b(j): j = j + 1
    NEXT r
ELSE
    FOR r = k TO n + m
        c(r) = a(i): i = i + 1
    NEXT r
END IF
FOR i = 1 TO m + n
    PRINT c(i)
NEXT
END

```



9.3. Συμβουλές - υποδείξεις

Η χρήση των πινάκων είναι ένας βολικός τρόπος για την αποθήκευση μεγάλου αριθμού δεδομένων ιδίου τύπου. Συνήθως οι νέοι προγραμματιστές χρησιμοποιούν πίνακες ακόμη και όταν η χρήση τους δεν είναι απαραίτητη.

Εξέτασε αν πραγματικά χρειάζεται πίνακας για την επίλυση του προβλήματος. Αν δεν είναι απαραίτητος, μην τον χρησιμοποιείς. Να έχεις πάντα στο νου σου ότι οι πίνακες ξοδεύουν μεγάλα ποσά μνήμης.

Για να αποφύγεις τα πλέον κοινά λάθη στη χρήση των πινάκων να προσέχεις πάντα:

- Να δίνεις αρχικές τιμές σε όλους τους πίνακες.
- Μην ξεπερνάς τα όρια του πίνακά σου. Το πιο συνηθισμένο λάθος στη χρήση των πινάκων είναι η προσπάθεια ανάγνωσης ή εκχώρησης τιμής έξω από τα όρια του πίνακα.
- Η επεξεργασία γίνεται στα στοιχεία του πίνακα. Άρα σε όλες τις εντολές πρέπει να εμφανίζονται τα στοιχεία του πίνακα και όχι το όνομα του ίδιου του πίνακα.
- Όλα τα στοιχεία του πίνακα έχουν τον ίδιο τύπο, για παράδειγμα όλα είναι ακέραια ή όλα είναι χαρακτήρες όπως ορίστηκαν στο τμήμα δηλώσεων.
- Στην ταξινόμηση ή την αναζήτηση σε έναν πίνακα να χρησιμοποιείς πάντα τη μέθοδο που είναι πιο κατάλληλη.

9.4. Δραστηριότητες - ασκήσεις

ΔΤ1. Να γράψετε τις δηλώσεις των παρακάτω πινάκων, καθώς και τις εντολές με τις οποίες εκχωρούνται οι τιμές σε αυτά.

- A) Πίνακας 5 στοιχείων που κάθε στοιχείο έχει την τιμή του δείκτη του.
- B) Πίνακας που θα περιέχει τα ψηφία.
- Γ) Πίνακας που περιέχει τα ονόματα των συμμαθητών σου.
- Δ) Πίνακας με 10 στοιχεία, πρώτο στοιχείο τον αριθμό 500 και κάθε επόμενο στοιχείο να είναι το μισό του προηγούμενου, δηλαδή το δεύτερο 250, το τρίτο 125 κ.ο.κ.

ΔΤ2. Έχουμε δύο πίνακες, ο ένας με τα μοντέλα των υπολογιστών και ο δεύτερος με τις τιμές τους. Να γράψετε τις εντολές που βρίσκουν και τυπώνουν το φθηνότερο μοντέλο καθώς και το ακριβότερο.

ΔΤ3. Να γράψετε τις εντολές που δίνουν τις ακόλουθες τιμές σε έναν πίνακα ακεραίων A.

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

ΔΤ4. Να γραφούν οι εντολές που ανταλλάσσουν τα στοιχεία της τρίτης και της έκτης στήλης σε έναν πίνακα ακεραίων 5x6.

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1. Να γράψετε ένα πρόγραμμα το οποίο να διαβάζει τον αριθμό των τερμάτων που σημειώθηκαν στους αγώνες ποδοσφαίρου μίας αγωνιστικής της Α κατηγορίας (9 τιμές), να υπολογίζει το μέσο αριθμό τερμάτων καθώς και το εύρος των τερμάτων (δηλαδή τη διαφορά της μεγαλύτερης από τη μικρότερη τιμή).

ΔΕ2. Να γράψετε το πρόγραμμα του παραδείγματος 2 (επίπεδα θορύβου αυτοκινήτων) και να το εκτελέσετε για τις τιμές που δίνονται στον πίνακα του παραδείγματος. Το πρόγραμμά σας να τυπώνει τον πίνακα με τα επίπεδα θορύβου για κάθε μοντέλο.

ΔΕ3. Να γράψετε την άσκηση ΔΕ4 (ρύπανση ατμόσφαιρας) του προηγούμενου κεφαλαίου χρησιμοποιώντας πίνακες για την αποθήκευση των τιμών καθώς και των ονομάτων των σταθμών μέτρησης.

ΔΕ4. Να γράψετε πρόγραμμα το οποίο να ταξινομεί τα μοντέλα αυτοκινήτων του παραδείγματος 2, κατά αύξουσα σειρά του μέσου επιπέδου θορύβου κάθε μοντέλου.

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1. Να συμπληρώσετε το παράδειγμα 1 (εισπράξεις αεροπορικών εταιρειών), ώστε να τυπώνει και αυτές που έχουν εισπράξεις κάτω από το μέσο όρο, να βρίσκει και να τυπώνει την εταιρεία με τις λιγότερες και με τις περισσότερες εισπράξεις.

ΔΣ2. Να γραφεί πρόγραμμα το οποίο να δέχεται δύο τετραγωνικούς δισδιάστατους πίνακες και να υπολογίζει το άθροισμα και το γινόμενό τους.

Υπόδειξη: Αν a και b είναι οι αρχικοί πίνακες και c ο τελικός, τότε ισχύει:

Πρόσθεση:
$$c_{ij} = a_{ij} + b_{ij}$$

Πολ/σμός:
$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}$$

ΔΣ3. Να γραφεί πρόγραμμα που να υπολογίζει το άθροισμα των κυρίων διαγωνίων τετραγωνικού πίνακα $N \times N$.

ΔΣ4. Να γραφεί ένα πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό d και μία βάση μετατροπής b , όπου $2 \leq b \leq 16$ και να μετατρέπει τον αριθμό d σε σύστημα αρίθμησης με βάση b .

ΔΣ5. Δίνεται ένας πίνακας A που περιέχει N τυχαίους ακραίους αριθμούς. Να γραφεί πρόγραμμα το οποίο να διαβάζει έναν αριθμό και να ελέγχει αν ο αριθμός υπάρχει στον πίνακα. Για την αναζήτηση να χρησιμοποιηθεί ο αλγόριθμος της σειριακής αναζήτησης που παρουσιάστηκε στο κεφάλαιο 3.

ΔΣ6. Δίνονται οι πίνακες $\Sigma 1(K,K)$ και $\Pi 1(K,K)$ που περιέχουν τα αποτελέσματα των αγώνων ομίλου του EuroBasket. Ο πίνακας $\Sigma 1$ περιέχει τα αποτελέσματα των αγώνων (N (νίκη) ή H (ήττα)), ενώ ο πίνακας $\Pi 1$ τη διαφορά πόντων για κάθε αγώνα.

Να γραφεί πρόγραμμα το οποίο θα βρίσκει και θα εκτυπώνει την τελική βαθμολογία του ομίλου. Σε περίπτωση ισοβαθμίας προηγείται η ομάδα που έχει την καλύτερη διαφορά πόντων από τις ισόβαθμές της.



Τα στοιχεία της κύριας διαγωνίου δεν περιέχουν καμία πληροφορία (καμία ομάδα δεν παίζει με τον εαυτό της!).

Ο πίνακας περιέχει στοιχεία μόνο κάτω ή πάνω από τη διαγώνιο του, είναι δηλαδή τριγωνικός (κάθε ομάδα παίζει μόνο μία φορά με κάθε αντίπαλο).

9.5. Τεστ αυτοαξιολόγησης

1. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

1. Οι πίνακες οι οποίοι έχουν τα στοιχεία τους σε μία στήλη ονομάζονται _____.
2. Οι πίνακες είναι μία _____ δομή δεδομένων.
3. Το αποτέλεσμα από τις παρακάτω εντολές είναι ο υπολογισμός του αθροίσματος του _____ πίνακα A

```
Αθροισμα <- 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ N
    Αθροισμα <- Αθροισμα+A[I,I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

4. Οι πίνακες πρέπει να χρησιμοποιούνται πάντα όταν αυτό είναι δυνατό.
5. Η δήλωση των πινάκων που χρησιμοποιούνται σε ένα πρόγραμμα είναι υποχρεωτική.
6. Για την ταξινόμηση ενός πίνακα 100 στοιχείων μπορεί να χρησιμοποιηθεί μόνο μία μέθοδος.
7. Η χρήση των πινάκων σε ένα πρόγραμμα αυξάνει την απαιτούμενη μνήμη.

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

8. Ποιες από τις παρακάτω εντολές τυπώνουν όλα τα στοιχεία ενός δισδιάστατου πίνακα Π 2x2:

A)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 2
    ΓΡΑΨΕ Π[I,I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 2
    ΓΡΑΨΕ Π[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Γ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 2
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 2
        ΓΡΑΨΕ Π[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Δ)

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 2
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 2
        ΓΡΑΨΕ Π
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

9. Ποιο το αποτέλεσμα των παρακάτω εντολών στον πίνακα A 8x10:

```
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 8
    Αθροισμα <- 0
    ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10
        Αθροισμα <- Αθροισμα+A[I,J]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ <- Αθροισμα/10
ΓΡΑΨΕ ΜΟ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

- Α) Γράφει το μέσο όρο των στοιχείων του πίνακα.
- Β) Γράφει το μέσο όρο των στοιχείων κάθε γραμμής.
- Γ) Γράφει το μέσο όρο των στοιχείων κάθε στήλης.
- Δ) Γράφει το μέσο όρο της τελευταίας γραμμής.

10. Ποιο είναι το αποτέλεσμα των παρακάτω εντολών:

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10
    Α[I] <- 10+I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΣΥΝ <- 0
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2
    ΣΥΝ <- ΣΥΝ+Α[Κ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΣΥΝ
```

- Α) 75 Β) 155 Γ) 50 Δ) 125

4. *Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:*

11. Τυπικές επεξεργασίες σε έναν πίνακα είναι:
- Α) Ταξινόμηση
 - Β) Πρόσθεση στοιχείων
 - Γ) Πολλαπλασιασμός στοιχείων
 - Δ) Συγχώνευση
 - Ε) Αναζήτηση
12. Η επιλογή του καλύτερου αλγορίθμου ταξινόμησης εξαρτάται από:
- Α) Τον τύπο δεδομένων που έχει ο πίνακας
 - Β) Τη διάσταση του πίνακα
 - Γ) Το πλήθος των στοιχείων του πίνακα
 - Δ) Την αρχική διάταξη των στοιχείων

Υποπρογράμματα

10.1. Προσδοκώμενα αποτελέσματα

Με το κεφάλαιο αυτό κλείνει η ενότητα του προγραμματισμού με χρήση δομημένης γλώσσας προγραμματισμού. Το κεφάλαιο ασχολείται με την έννοια τμηματικού προγραμματισμού, πώς δηλαδή αναλύεται το πρόγραμμα σε **υποπρογράμματα** και τον τρόπο με τον οποίο η ΓΛΩΣΣΑ χειρίζεται τα υποπρογράμματα.

Για να αναλύσεις σωστά ένα σύνθετο πρόγραμμα σε υποπρογράμματα, πρέπει αρχικά να αποφασίζεις πότε θα χρησιμοποιήσεις **συναρτήσεις** και **διαδικασίες**, να γνωρίζεις τη δομή καθώς και τα βασικά χαρακτηριστικά αυτών των υποπρογραμμάτων. Κάθε γλώσσα προγραμματισμού έχει ελαφρώς διαφορετικό τρόπο με τον οποίο αντιμετωπίζει τα υποπρογράμματα και ειδικά τον τρόπο με τον οποίο χειρίζεται τις παραμέτρους. Στο βιβλίο σου παρουσιάστηκαν θεωρητικά οι αρχές επικοινωνίας των υποπρογραμμάτων με τη χρήση των παραμέτρων, στο εργαστήριο θα γνωρίσεις το συγκεκριμένο τρόπο που το δικό σου προγραμματιστικό περιβάλλον υλοποιεί αυτές τις αρχές και πώς χρησιμοποιεί τις παραμέτρους.

Τέλος σε αυτό το κεφάλαιο θα γνωρίσεις τον τρόπο που υλοποιείται η αναδρομή, τα πλεονεκτήματα από τη σύνταξη αναδρομικών προγραμμάτων αλλά και τα μειονεκτήματά της σε σχέση με τα επαναληπτικά προγράμματα.

Οι λυμένες ασκήσεις του κεφαλαίου αυτού παρουσιάζονται στο περιβάλλον της ιδεατής γλώσσας προγραμματισμού ΓΛΩΣΣΑ και επίσης στα πραγματικά προγραμματιστικά περιβάλλοντα Basic και Pascal.

10.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

Πολλά από τα προγράμματα που αναπτύχθηκαν στα προηγούμενα κεφάλαια μπορούν να γραφούν καλύτερα με τη χρήση υποπρογραμμάτων. Εδώ θα δούμε το πρόγραμμα που υπολογίζει τα βασικά στατιστικά μεγέθη, τη μέση τιμή, την τυπική απόκλιση και τη διάμεσο τιμή που παρουσιάστηκε στο βιβλίο σου στο κεφάλαιο 9.

Το πρόγραμμα χρησιμοποιεί τις εξής διαδικασίες και συναρτήσεις:

Υπολόγισε_ΜΟ_ΤυπΑπ: Υπολογίζει τη μέση τιμή και την τυπική απόκλιση ακεραίων αριθμών. Το τμήμα αυτό θα μπορούσε να υλοποιηθεί και με δύο συναρτήσεις, μία για τον υπολογισμό της μέσης τιμής και μίας δεύτερης για τον υπολογισμό της τυπικής απόκλισης.

Ταξινόμηση: Η διαδικασία αυτή ταξινομεί τα στοιχεία του πίνακα χρησιμοποιώντας μία παραλλαγή του αλγορίθμου που παρουσιάστηκε στο βιβλίο σου.

Υπολογισμός_Διαμέσου: Πραγματική συνάρτηση η οποία υπολογίζει τη διάμεσο τιμή.

ΠΡΟΓΡΑΜΜΑ Στατιστική

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: I, Πλήθος, Στοιχεία[100], Μέγιστο,
Αθροισμα, Αθροισμα_2, Βοηθητική
ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, Τυπ_Απόκλιση, Διάμεσος

ΑΡΧΗ

! Εισαγωγή στοιχείων
ΓΡΑΨΕ 'Δώσε το πλήθος των αριθμών (μέγιστο 100)'
ΔΙΑΒΑΣΕ Πλήθος
ΓΙΑ I **ΑΠΟ** 1 **ΜΕΧΡΙ** Πλήθος
 ΓΡΑΨΕ 'Δώσε έναν αριθμό'
 ΔΙΑΒΑΣΕ Στοιχεία[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΚΑΛΕΣΕ Υπολόγισε_ΜΟ_ΤυπΑπ(Στοιχεία, Πλήθος, ΜΟ, Τυπ_Απόκλιση)
ΚΑΛΕΣΕ Ταξινόμηση(Στοιχεία, Πλήθος)
Διάμεσος <- Υπολογισμός_Διαμέσου(Στοιχεία, Πλήθος)
! Εκτύπωση αποτελεσμάτων
ΓΡΑΨΕ 'ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΤΟΥΣ ', Πλήθος, 'ΑΡΙΘΜΟΥΣ'
ΓΡΑΨΕ 'ΜΕΣΟΣ ΟΡΟΣ:', ΜΟ
ΓΡΑΨΕ 'ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ:', Τυπ_Απόκλιση
ΓΡΑΨΕ 'ΔΙΑΜΕΣΟΣ:', Διάμεσος

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Στατιστική

ΔΙΑΔΙΚΑΣΙΑ Υπολόγισε_ΜΟ_ΤυπΑπ(Πίνακας, N, ΜΟ, ΤυπΑποκλ)

! Υπολογισμός μέσου όρου
! Υπολογισμός τυπικής απόκλισης

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Πίνακας[100], N, I, Αθροισμα, Αθροισμα_2
ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, ΤυπΑποκλ

ΑΡΧΗ

```

Αθροισμα <- 0
Αθροισμα_2 <- 0
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N
    Αθροισμα <- Αθροισμα+ Πίνακας[I]
    Αθροισμα_2 <- Αθροισμα_2+ Πίνακας[I]^2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ <- Αθροισμα/N
ΤυπΑποκλ <- T_P(Αθροισμα_2 / N - ΜΟ^2)
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Υπολόγισε_ΜΟ_ΤυπΑπ

ΔΙΑΔΙΚΑΣΙΑ Ταξινομήσε(Πίνακας, N)
!Ταξινομήση των στοιχείων του πίνακα
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: I, N1, T, Βοηθητική, Πίνακας[100], N
ΑΡΧΗ
    N1 <- N
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        T <- 0
        ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N1-1
            ΑΝ Πίνακας[I] > Πίνακας[I+1] ΤΟΤΕ
                Βοηθητική <- Πίνακας[I]
                Πίνακας[I] <- Πίνακας[I+1]
                Πίνακας[I+1] <- Βοηθητική
                T <- I
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
        N1 <- T
    ΜΕΧΡΙΣ_ΟΤΟΥ T=0
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Ταξινομήσε

ΣΥΝΑΡΤΗΣΗ Υπολογισμός_Διαμέσου(A, N): ΠΡΑΓΜΑΤΙΚΗ
ΑΚΕΡΑΙΕΣ: A[100], N

ΑΡΧΗ
    ΑΝ N MOD 2 = 0 ΤΟΤΕ
        Υπολογισμός_Διαμέσου <- (A[N/2]+A[N/2+1])/2
    ΑΛΛΙΩΣ
        Υπολογισμός_Διαμέσου <- A[(N+1)/2]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ Υπολογισμός_Διαμέσου

```

Προγραμματιστικό περιβάλλον Pascal

```

program Statistiki;

type list=array[1..100] of integer;
var
    avg, st_dev:real;
    i,median,n:integer;

```



```

    a:list;
{-----}
procedure sort(var a:list;n:integer);
{Ταξινόμηση των στοιχείων του πίνακα A}
var
    i,n1,t,temp:integer;
begin
    n1:=n;
    repeat
        t:=0;
        for i:=1 to n1-1 do
            if a[i]>a[i+1] then
                begin
                    temp:=a[i];
                    a[i]:=a[i+1];
                    a[i+1]:=temp;
                    t:=i;
                end;
            n1:=t;
        until t=0;
    end;
{-----}
procedure ave_stdev(a:list;n:integer; var average,stddv:real);
{υπολογισμός μέσου όρου και τυπικής απόκλισης}
var
    i:integer;
    sum, sum_2:real;

begin
    sum:=0; sum_2:=0;
    for i:=1 to n do
        begin
            sum:=sum+a[i]; sum_2:=sum_2+sqr(a[i]);
        end;
    average:=sum/n;
    stddv:=sqrt (sum_2/n - sqr(average)) ;
end;
{-----}
function med(a:list; n:integer):real;
{υπολογισμός της διαμέσου τιμής}
begin
    if n mod 2=0 then
        med:=(a[n div 2]+a[(n div 2)+1])/2
    else
        med:=a[(n+1) div 2];
    end;
begin {κυρίως πρόγραμμα}
    write ('ΔΩΣΕ ΤΟΝ ΠΛΗΘΟΣ ΤΩΝ ΑΡΙΘΜΩΝ (<100):');
    readln (n);

```

```

for i:=1 to n do
begin
    write ('ΔΩΣΕ ΤΟΝ ', i:3,'ο ΑΡΙΘΜΟ :');readln (a[i]);
end;
ave_stdev (a,n,avg,st_dev);
qsort(a,1,n);
median:= med(a,n);
writeln ('Μέση τιμή :', avg, 'Τυπική απόκλιση :',st_dev);
writeln('Διάμεσος τιμή :',median);
end.

```

Προγραμματιστικό περιβάλλον Basic

```

DECLARE FUNCTION Median! (x!(), n!)
DECLARE SUB Sort (k!(), n!)
DECLARE SUB MeanAndStdDev (z!(), m!, s!)
` Στατιστική
DIM x(100)
DATA 7
DATA 1,5,7,12,9,13,6
READ n
FOR i = 1 TO n: READ x(i): NEXT i
,

CLS
CALL MeanAndStdDev(x(), mx, sx)
CALL Sort(x(), n)
med = Median(x(), n)
,

CLS
PRINT "**** Αποτελέσματα ****"
PRINT "-----"
PRINT "mx="; mx, "sx="; sx
PRINT "median="; med
END

SUB MeanAndStdDev (z(), m, s)
` Υπολογισμός μέσης τιμής και τυπικής απόκλισης
s1 = 0: s2 = 0
n = UBOUND(z)
FOR i = 1 TO n
    s1 = s1 + z(i)
    s2 = s2 + z(i) * z(i)
NEXT i
m = s1 / n
s = SQR(s2 / n - m * m)
END SUB

FUNCTION Median (x(), n)
IF n MOD 2 = 0 THEN
    Median = (x(n / 2) + x (n/2 + 1 ) ) /2

```

```

ELSE
    Median = x((n + 1) / 2)
END IF
END FUNCTION

SUB Sort (x(), n) STATIC
k = n
DO
    t = 0
    FOR i = 1 TO k - 1
        IF x(i) > x(i + 1) THEN
            SWAP x(i), x(i + 1)
            t = i
        END IF
    NEXT i
    k = t
LOOP UNTIL t = 0
END SUB

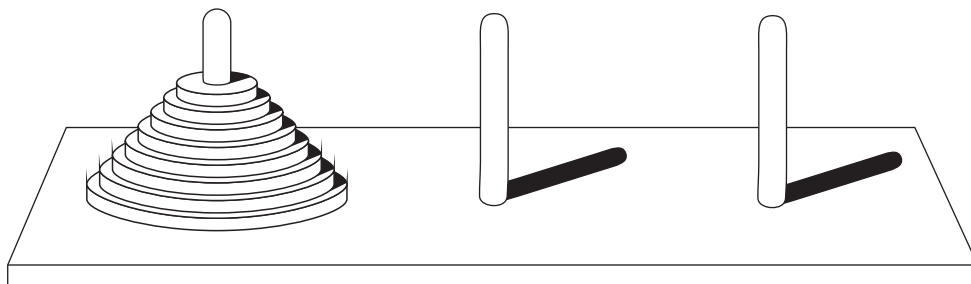
```

ΠΑΡΑΔΕΙΓΜΑ 2

Ένα χαρακτηριστικό πρόβλημα το οποίο λύνεται εύκολα με τη χρήση αναδρομής, ενώ είναι πολύ δύσκολο με επαναληπτική διαδικασία, είναι οι πύργοι του **Ανόι**.

Στο πρόβλημα των πύργων του Ανόι υπάρχουν τρεις στύλοι και στον πρώτο από αυτούς βρίσκονται περασμένοι δίσκοι διαφορετικής διαμέτρου, έτσι ώστε οι διάμετροι των δίσκων να μικραίνουν από κάτω προς τα πάνω.

Όλοι οι δίσκοι, που βρίσκονται στον πρώτο στύλο, πρέπει να μεταφερθούν στο τρίτο ακολουθώντας τους εξής κανόνες:



Όταν ένας δίσκος μεταφέρεται, πρέπει να τοποθετηθεί σε έναν από τους τρεις στύλους. Μόνο ένας δίσκος μπορεί να μεταφερθεί κάθε φορά και πρέπει να βρίσκεται στην κορυφή του στύλου.

Ένας μεγαλύτερος δίσκος δεν πρέπει να τοποθετηθεί πάνω από ένα μικρότερο.

Ιστορικό σημείωμα

Σύμφωνα με το μύθο το πρόβλημα δόθηκε στους μοναχούς του ιερού ναού του Μπενάρες. Στους μοναχούς δόθηκε μια χρυσή βάση με τρεις χρυσές βελόνες και εξήντα τέσσερις μικροί χρυσοί δίσκοι. Όταν οι μοναχοί κατορθώσουν να λύσουν το πρόβλημα, δηλαδή να μεταφέρουν τους εξήντα τέσσερις δίσκους από την πρώτη βελόνα στην τρίτη ακολουθώντας τους τρεις κανόνες που αναφέρθηκαν, τότε θα έρθει η συντέλεια του Κόσμου (όπως θα δούμε στη συνέχεια, δεν διατρέχουμε κανένα κίνδυνο).

Το παιχνίδι είναι σχετικά εύκολο να λυθεί για μικρό αριθμό δίσκων, τρεις-τέσσερις, αλλά δυσκολεύει εξαιρετικά όσο ο αριθμός των δίσκων αυξάνεται.

Η γενική διατύπωση της λύσης όμως με χρήση αναδρομικής διαδικασίας είναι αρκετά απλή και περιγράφεται από τα παρακάτω βήματα:

Αν υπάρχει μόνο ένας δίσκος, τότε μεταφέρεται από το Στύλο1 στο Στύλο3. Το πρόβλημα λοιπόν έχει λύση για $N = 1$.

Αν υπάρχουν δύο δίσκοι, τότε χρειάζονται τρεις απλές κινήσεις:

Ο πρώτος δίσκος από το Στύλο1 μεταφέρεται στο Στύλο2.

Ο δεύτερος δίσκος από το Στύλο2 μεταφέρεται στο Στύλο3.

Ο δίσκος από το Στύλο2 μεταφέρεται στο Στύλο3.

Υποθέτουμε ότι η λύση υπάρχει για $N-1$ δίσκους, τότε για N δίσκους η λύση δίνεται αναδρομικά:

Οι $N-1$ δίσκοι μεταφέρονται από το Στύλο1 στο Στύλο2, χρησιμοποιώντας το Στύλο3 ως βοηθητικό.

Ο τελευταίος δίσκος, που είναι ο τελευταίος άρα και ο μεγαλύτερος, μεταφέρεται από το Στύλο1 στο Στύλο3.

Οι $N-1$ δίσκοι μεταφέρονται από το Στύλο2 στο Στύλο3, χρησιμοποιώντας το Στύλο1 ως βοηθητικό.

Το πρόγραμμα που λύνει τους Πύργους του Ανόι είναι το ακόλουθο:

ΠΡΟΓΡΑΜΜΑ Πύργοι_του_Ανόι

ΣΤΑΘΕΡΕΣ

Στύλος1='Α'

Στύλος2='Β'

Στύλος3='Γ'

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:N

ΑΡΧΗ

ΓΡΑΨΕ 'δώσε τον αριθμό των δίσκων'

ΔΙΑΒΑΣΕ N

ΚΑΛΕΣΕ Μετακίνηση(N, Στύλος1, Στύλος2, Στύλος3)

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Μετακίνηση (N, ΣτύλοςΑ, ΣτύλοςΒ, ΣτύλοςΓ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: N

ΧΑΡΑΚΤΗΡΕΣ: ΣτύλοςΑ, ΣτύλοςΒ, ΣτύλοςΓ

ΑΡΧΗ

ΑΝ N=1 **ΤΟΤΕ**

ΓΡΑΨΕ 'Μετακίνηση από τον', ΣτύλοςΑ, ' στον' , ΣτύλοςΓ

ΑΛΛΙΩΣ

ΚΑΛΕΣΕ Μετακίνηση (N-1, ΣτύλοςΑ, ΣτύλοςΓ, ΣτύλοςΒ)

ΓΡΑΨΕ 'Μετακίνηση από τον', ΣτύλοςΑ, ' στον' , ΣτύλοςΓ

ΚΑΛΕΣΕ Μετακίνηση (N-1, ΣτύλοςΒ, ΣτύλοςΑ, ΣτύλοςΓ)

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Προγραμματιστικό περιβάλλον Pascal

programanoi;

const

st1='A';

st2='B';

st3='Γ';

var

n:**integer**;

procedure move (n:**integer**;sta, stb, stc:**char**);

begin

if n=1 **then**

begin

writeln('METAKINHSE APO ',sta,' ΣΤΟ ',stc)

end

else

begin

move(n-1, sta, stc, stb);

writeln ('METAKINHSE APO ', sta, ' ΣΤΟ ',stc);

move (n-1, stb, sta, stc)

end;

end;

begin

write ('ΑΡΙΘΜΟΣ ΔΙΣΚΩΝ :'); **readln**(n);

move (n, st1, st2, st3);

end.

Προγραμματιστικό περιβάλλον Basic

DECLARE SUB Move (n!, a\$, b\$, c\$)

' Anoi

INPUT "N=", n

CALL Move (n, "A", "B", "C")

END

```

SUB Move (n, a$, b$, c$)
IF n = 1 THEN
    PRINT "Move "; a$; " to "; c$
ELSE
    CALL Move(n - 1, a$, c$, b$)
    PRINT "Move "; a$; " to "; c$
    CALL Move(n - 1, b$, a$, c$)
END IF
END SUB

```

Η εκτέλεση του προγράμματος για 4 δίσκους δίνει τα εξής αποτελέσματα:

```

Μετακίνησε από τον Α στον Β
Μετακίνησε από τον Α στον Γ
Μετακίνησε από τον Β στον Γ
Μετακίνησε από τον Α στον Β
Μετακίνησε από τον Γ στον Α
Μετακίνησε από τον Γ στον Β
Μετακίνησε από τον Α στον Β
Μετακίνησε από τον Α στον Γ
Μετακίνησε από τον Β στον Γ
Μετακίνησε από τον Β στον Α
Μετακίνησε από τον Γ στον Α
Μετακίνησε από τον Β στον Γ
Μετακίνησε από τον Α στον Β
Μετακίνησε από τον Α στον Γ
Μετακίνησε από τον Β στον Γ

```

Η λύση που δόθηκε θεωρητικά επιλύει το πρόβλημα για οποιονδήποτε αριθμό δίσκων. Ας μελετήσουμε όμως τον αριθμό των κινήσεων που απαιτείται.

Όπως είδαμε, χρειάζονται 15 κινήσεις για την επίλυση του προβλήματος με 4 δίσκους. Για 5 δίσκους οι κινήσεις που χρειάζονται είναι 31.

Γενικά για N δίσκους η λύση δίνεται μετά από $2^N - 1$ κινήσεις. Αυτό σημαίνει ότι για 10 δίσκους χρειάζονται 1023 κινήσεις και για 20 δίσκους οι κινήσεις ξεπερνούν το εκατομμύριο.

Για τους 64 δίσκους που απασχολεί τους μοναχούς χρειάζονται $1,845 \times 10^{19}$, δηλαδή ένας αριθμός με 20 ψηφία. Πόσο μεγάλος είναι ένας τέτοιος αριθμός; Αν υποθέσουμε ότι εκτελούμε χίλιες κινήσεις το δευτερόλεπτο (όσες περίπου μπορεί να εκτελέσει ένας γρήγορος υπολογιστής), τότε χρειάζονται περίπου μισό εκατομμύριο χρόνια!!!

Η επίλυση του προβλήματος του πύργου του Ανόι αποτελεί χαρακτηριστική περίπτωση αλγορίθμων εκθετικής πολυπλοκότητας ($O(2^N)$).

Οι αλγόριθμοι της μορφής αυτής όπως αναφέρθηκε στο βιβλίο σου στο κεφάλαιο 5 "Ανάλυση Αλγορίθμων" είναι ουσιαστικά ακατάλληλοι για την πρακτική επίλυση προβλημάτων και χρήσιμοι μόνο για θεωρητικά προβλήματα, αφού κάθε αύξηση του N κατά μία μονάδα διπλασιάζει τον απαιτούμενο χρόνο εκτέλεσης.



10.3. Συμβουλές - υποδείξεις

Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και τις δικές της αρχές για τη σύνταξη και χρήση των υποπρογραμμάτων και πρέπει να μελετήσεις προσεκτικά πώς υλοποιούνται τα υποπρογράμματα στο προγραμματιστικό περιβάλλον που χρησιμοποιείς. Υπάρχουν όμως κάποιοι γενικοί κανόνες που πρέπει να ακολουθείς.

Πριν ξεκινήσεις να γράφεις το πρόγραμμά σου, να μελετήσεις πώς το πρόγραμμα μπορεί να αναλυθεί σε επιμέρους τμήματα και να αποφασίσεις για τα αντίστοιχα υποπρογράμματα. Χρήσιμο είναι να κάνεις ένα διάγραμμα που θα δείχνει την ιεραρχία ανάμεσα στα υποπρογράμματα. Τότε να αναπτύσσεις τους αλγορίθμους για το κάθε υποπρόγραμμα και στη συνέχεια να γράφεις το πρόγραμμα.

Να μελετάς αν ένα υποπρόγραμμα πρέπει να υλοποιηθεί με διαδικασία ή μπορεί να υλοποιηθεί με συνάρτηση.

Εξέτασε αν κάποια υποπρογράμματα τα οποία έχεις ήδη γράψει ή υπάρχουν σε έτοιμες βιβλιοθήκες προγραμμάτων μπορούν να χρησιμοποιηθούν. Θα γλιτώσεις χρόνο και κόπο.

Κάθε υποπρόγραμμα να προσπαθείς να είναι όσο το δυνατόν πιο ανεξάρτητο από τα άλλα. Αυτό σε προφυλάσσει από λάθη στο πρόγραμμά σου και σου επιτρέπει τη χρήση του σε άλλα προγράμματα αργότερα.

Για να αποφύγεις τα πλέον κοινά λάθη, να προσέχεις ιδιαίτερα:

Να ορίζεις τον τύπο της συνάρτησης. Οι συναρτήσεις παράγουν μόνο ένα αποτέλεσμα συγκεκριμένου τύπου, ακεραίου, πραγματικού κ.λπ. που πρέπει να ορίζεται.

Να μην υπάρχουν λάθη στην αντιστοίχιση τυπικών και πραγματικών παραμέτρων. Πρόσεξε ότι οι λίστες πρέπει να περιέχουν τον ίδιο αριθμό παραμέτρων και κάθε τυπική παράμετρος με την αντίστοιχη πραγματική πρέπει να είναι του ίδιου τύπου.

10.4. Δραστηριότητες - ασκήσεις

ΔΤ1. Τι είδους υποπρόγραμμα, διαδικασία ή συνάρτηση πρέπει να χρησιμοποιήσεις για τα παρακάτω:

- A) Εισαγωγή τριών δεδομένων.
- B) Εισαγωγή ενός δεδομένου.
- Γ) Υπολογισμός του μικρότερου από πέντε ακεραίου.
- Δ) Υπολογισμός των δύο μικρότερων από πέντε ακεραίους.
- E) Έλεγχος αν δύο αριθμοί είναι ίσοι.
- Z) Να ταξινομεί, και να επιστρέφει ταξινομημένους, πέντε αριθμούς.
- H) Έλεγχος αν ένας χαρακτήρας είναι φωνήεν ή σύμφωνο.

- ΔΤ2.** Να γράψεις τα υποπρογράμματα που υλοποιούν τα παρακάτω:
- A) Να διαβάζει έναν αριθμό και να επιστρέφει το τετράγωνό του.
 - B) Να δέχεται δύο αριθμούς και να επιστρέφει το μικρότερο από δύο αριθμούς.
 - Γ) Να δέχεται την τιμή ενός προϊόντος και να υπολογίζει και να τυπώνει την αξία του ΦΠΑ.
 - Δ) Να ελέγχει αν ένας αριθμός είναι άρτιος.
- ΔΤ3.** Να σημειώσεις, στο τετράδιό σου, όλα τα βήματα για τον υπολογισμό του $4!$, τόσο με τη χρήση επαναληπτικής διαδικασίας όσο και με τη χρήση αναδρομικής, σύμφωνα με τα προγράμματα που δίνονται στο βιβλίο σου.

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

- ΔΕ1.** Να γράψεις πρόγραμμα το οποίο θα διαβάζει δύο αριθμούς, θα υπολογίζει το Μέγιστο Κοινό Διαιρέτη (ΜΚΔ) και το Ελάχιστο Κοινό Πολλαπλάσιο (ΕΚΠ) και τέλος θα τυπώνει τα αποτελέσματα.

Υπόδειξη: Για δύο αριθμούς x, y ισχύει: $x \cdot y = \text{ΜΚΔ}(x, y) \cdot \text{ΕΚΠ}(x, y)$

- ΔΕ2.** Να εκτελέσεις το πρόγραμμα του παραδείγματος 1.

- ΔΕ3.** Να γράψεις πρόγραμμα το οποίο να εκτελεί τις τέσσερις πράξεις σε μιγαδικούς αριθμούς.

Για τους μιγαδικούς αριθμούς $a+bi$ και $\gamma+di$ έχουμε:

$$(a+bi) + (\gamma+di) = (a+\gamma) + (b+d)i$$

$$(a+bi) - (\gamma+di) = (a-\gamma) + (b-d)i$$

$$(a+bi) \cdot (\gamma+di) = (a\gamma - b\delta) + (a\delta + b\gamma)i$$

$$\frac{a+bi}{\gamma+di} = \left(\frac{a\gamma + b\delta}{\gamma^2 + \delta^2} \right) + \left(\frac{b\gamma - a\delta}{\gamma^2 + \delta^2} \right) i$$



Το πρόγραμμα θα οδηγείται από μενού επιλογής όπου ο χρήστης θα επιλέγει το είδος της πράξης.

Στην περίπτωση της διαίρεσης το γ και το δ πρέπει να είναι διάφορα του 0.

- ΔΕ4.** Να ξαναγράψεις το πρόγραμμα της ΔΕ1 χρησιμοποιώντας αναδρομικές συναρτήσεις.

Σύγκρινε τα δύο προγράμματα.

- ΔΣ1.** Να γράψεις ένα πρόγραμμα το οποίο διαβάζει την τιμή βιβλίων σε ΕΥΡΩ και μετατρέπει τις τιμές τους σε Δραχμές, Γερμανικά Μάρκα, Γαλλικά Φράγκα και Ιταλικές Λιρέτες. Να χρησιμοποιήσεις για τις μετατροπές τις τρέχουσες ισοτιμίες των νομισμάτων.
- ΔΣ2.** Να ξαναγράψεις την άσκηση ΔΣ6 του κεφαλαίου 9, τα αποτελέσματα των αγώνων ομίλου του EuroBasket, χρησιμοποιώντας διαδικασίες και συναρτήσεις.
- ΔΣ3.** Να επεκτείνεις το παράδειγμα 1, ώστε να υπολογίζει την επικρατούσα τιμή, δηλαδή την τιμή που εμφανίζεται περισσότερες φορές.
- ΔΣ4.** Να γράψεις το πρόγραμμα ΔΕ5 του κεφαλαίου 8 που υπολογίζει τη συνολική χωρητικότητα πυκνωτών και τη συνολική αντίσταση αντιστάσεων με τη χρήση υποπρογραμμάτων.
- ΔΣ5.** Να γραφεί πρόγραμμα το οποίο να προσθέτει δύο κλάσματα. Το πρόγραμμα δέχεται τέσσερις ακέραιους αριθμούς, τους παρονομαστές και τους αριθμητές των δύο κλασμάτων, υπολογίζει και εκτυπώνει τον αριθμητή και τον παρονομαστή του αποτελέσματος.

$$A/B + \Gamma/\Delta = E/Z$$



Υπόδειξη: Ενώ το πρόβλημα αρχικά φαίνεται απλό, η υλοποίησή του είναι αρκετά πολύπλοκη. Αρχικά πρέπει να απλοποιηθούν τα κλάσματα, στη συνέχεια να γίνουν ομώνυμα, να προστεθούν οι αριθμητές και τέλος να απλοποιηθεί το αποτέλεσμα.

Οι διαδικασίες αυτές απαιτούν τον υπολογισμό του ΜΚΔ (για την απλοποίηση) και του ΕΚΠ για τη μετατροπή των κλασμάτων σε ομώνυμα. Να χρησιμοποιήσετε τις συναρτήσεις της άσκησης ΔΕ1.

- ΔΣ6.** Δίνεται η εξίσωση $e^x - 2x - 1 = 0$. Να γραφεί πρόγραμμα το οποίο να βρίσκει μια ρίζα της εξίσωσης αυτής στο διάστημα $[1,2]$ με τη μέθοδο της διχοτόμησης, όπως περιγράφηκε στην παράγραφο 4.3 του βιβλίου σου.

10.5. Τεστ αυτοαξιολόγησης

1. Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

1. Η λίστα των παραμέτρων που υπάρχει στη δήλωση μίας διαδικασίας ονομάζεται λίστα _____ παραμέτρων.
2. Τα δύο είδη υποπρογραμμάτων είναι οι _____ και οι _____.
3. Οι μεταβλητές οι οποίες ισχύουν σε όλα τα υποπρογράμματα ενός προγράμματος και όχι μόνο σε αυτό που ορίστηκε λέγεται ότι έχουν _____ εμβέλεια.

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

4. Μια διαδικασία και μια συνάρτηση μπορούν να εκτελούν ακριβώς τις ίδιες λειτουργίες.
5. Το πλήθος των τυπικών και των πραγματικών παραμέτρων πρέπει να είναι ίδιο.
6. Οι αναδρομικές διαδικασίες είναι πάντοτε προτιμότερες από τις αντίστοιχες επαναληπτικές.
7. Η ενεργοποίηση μίας συνάρτησης γίνεται με την εντολή ΚΑΛΕΣΕ.

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

8. Όταν μία μεταβλητή ισχύει μόνο στο υποπρόγραμμα που ορίστηκε ονομάζεται:
Α) Τοπική Β) Καθολική Γ) Παράμετρος Δ) Τυπική
9. Τι θα τυπώσουν οι παρακάτω εντολές:

```
.....  
A <- 5  
B <- 10  
Γ <- 0  
ΚΑΛΕΣΕ ΔΙΑΔ1 (A, B)  
ΓΡΑΨΕ A, B, Γ  
.....  
ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ1 (Γ, Δ)  
.....  
ΑΡΧΗ  
Γ <- Γ-Δ  
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ ΔΙΑΔ1
```

- Α) 5,10,0 Β) 5,10,-5 Γ) -5,10,0 Δ) -5,10,-5

10. Τι θα τυπώσουν οι παρακάτω εντολές:

```
.....  
A <- 5  
B <- 10  
ΚΑΛΕΣΕ ΔΙΑΔ1 (B, A)  
ΓΡΑΨΕ A, B  
.....  
ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ1 (A, B)
```

.....

ΑΡΧΗ

ΓΡΑΨΕ Α, Β

A <- A-B

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Διαδ1

A) 5,10

B) 10,5

Γ) 5,10

Δ) 10,5

5,10

5,5

-5,10

5,10

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

11. Ο ορισμός κάθε αναδρομικού υποπρογράμματος περιλαμβάνει:

- A) Αναδρομική σχέση
- B) Τιμή βάσης
- Γ) Υπολογισμό παραγοντικού
- Δ) Επαναληπτική διαδικασία
- Ε) Καθολικές μεταβλητές

12. Μερικά από τα πλεονεκτήματα του τμηματικού προγραμματισμού είναι:

- A) Λιγότερος χρόνος για την ανάπτυξη του προγράμματος
- B) Ευκολότερη διόρθωση
- Γ) Ταχύτητα κατά την εκτέλεση
- Δ) Χρήση αναδρομικών διαδικασιών

Απαντήσεις στα τεστ αυτοαξιολόγησης

ΚΕΦΑΛΑΙΟ 1

- | | | | |
|-----------|--|-----------|-----------|
| 1. | 1. Κατανόηση, ανάλυση, επίλυση | 3. | 8. ΛΑΘΟΣ |
| | 2. Είσοδος, έλεγχος, επεξεργασία, έξοδος | | 9. ΣΩΣΤΟ |
| 2. | 3. κατανόηση | | 10. ΣΩΣΤΟ |
| | 4. πληροφορία | | 11. ΛΑΘΟΣ |
| | 5. διατύπωση | | 12. ΣΩΣΤΟ |
| | 6. δομή | | |
| | 7. απαιτήσεων | | |

ΚΕΦΑΛΑΙΟ 2

- | | | | |
|-----------|---|-----------|---------------------|
| 1. | 1. Η αναπαράσταση αλγορίθμου με ελεύθερο κείμενο αποτελεί μη δομημένο τρόπο παρουσίασης αλγορίθμου. | 2. | 4. Ακολουθιακή |
| | 2. Τα διαγράμματα ροής αποτελούν ένα γραφικό τρόπο παρουσίασης ενός αλγορίθμου με χρήση συγκεκριμένων τύπων γραφικών για κάθε ενέργεια. | | 5. Επιλογής |
| | 3. Η κωδικοποίηση ενός αλγορίθμου γίνεται με ένα πρόγραμμα που όταν εκτελεσθεί θα δίνει πάντοτε τα ίδια αποτελέσματα με τον αλγόριθμο. | | 6. Υλοποίηση |
| | | 3. | 7. Διαγράμματα ροής |
| | | | 8. Εμφωλευμένες |
| | | 4. | 9. ΣΩΣΤΟ |
| | | | 10. ΛΑΘΟΣ |
| | | | 11. ΛΑΘΟΣ |
| | | | 12. ΣΩΣΤΟ |
| | | 5. | 13. Α, Β, Γ |
| | | | 14. Α, Δ, Ε |
| | | | 15. Α, Β |
| | | | 16. Β, Δ |
| | | | 17. Α, Β, Δ |

ΚΕΦΑΛΑΙΟ 3

- | | |
|--|---|
| <p>1. 1. Δομή Δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίσταται επεξεργασία από ένα σύνολο λειτουργιών, που καλούνται από το υπόλοιπο πρόγραμμα.</p> <p>2. Οι δυναμικές δομές δεδομένων δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης, αλλά στηρίζονται στην τεχνική της λεγόμενης δυναμικής παραχώρησης μνήμης.</p> <p>3. Οι πίνακες χρησιμεύουν για την αποθήκευση και διαχείριση δύο σπουδαίων δομών, της στοίβας και της ουράς.</p> | <p>2. 4. Αντιγραφή
5. Συγχώνευση
6. Διαχωρισμός
7. Ώθηση, απώθηση
8. Εισαγωγή, εξαγωγή
9. ταξινόμηση</p> <p>3. 10. ΣΩΣΤΟ
11. ΛΑΘΟΣ
12. ΛΑΘΟΣ
13. ΣΩΣΤΟ</p> <p>4. 14. Α, Γ, Δ, Ε, Η
15. Α, Γ, Ε</p> <p>5. 16. Α, Γ</p> |
|--|---|

ΚΕΦΑΛΑΙΟ 4

- | | |
|---|--|
| <p>1. 1. Οι μέθοδοι λύσης ενός προβλήματος που προκύπτουν από την ανάλυσή του οδηγούν στη σχεδίαση ενός αλγορίθμου που συνιστά την ακολουθία βημάτων, που πρέπει να ακολουθηθούν, για να επιλυθεί το πρόβλημα.</p> <p>2. Κατά την επίλυση ενός προβλήματος, υπάρχει σύγκριση των χαρακτηριστικών και των ιδιοτήτων των τεχνικών που μπορούν να αποτελέσουν πρόταση λύσης του προβλήματος. Το αποτέλεσμα της σύγκρισης των διαφορετικών τεχνικών είναι η επιλογή της καταλληλότερης τεχνικής για την επίλυση του συγκεκριμένου προβλήματος.</p> | <p>3. Η μέθοδος του Δυναμικού Προγραμματισμού για τη σχεδίαση αλγορίθμων χρησιμοποιείται κυρίως για την επίλυση προβλημάτων βελτιστοποίησης, δηλαδή χρησιμοποιείται κυρίως όταν χρειάζεται να βρεθεί το ελάχιστο ή το μέγιστο κάποιου μεγέθους.</p> <p>2. 4. Άπληστη
5. Οπισθοδρόμησης
6. Δυαδική αναζήτηση
7. Ταξινόμηση φυσαλίδας, ταξινόμηση με επιλογή</p> <p>3. 8. ΣΩΣΤΟ
9. ΛΑΘΟΣ
10. ΛΑΘΟΣ</p> <p>4. 11. Α, Δ, Ε
12. Β, Γ, Ε
13. Β, Δ, Ε, Ζ</p> |
|---|--|

ΚΕΦΑΛΑΙΟ 5

- | | | | |
|-----------|---|-----------|--|
| 1. | 1. Η χειρότερη περίπτωση ενός αλγορίθμου αφορά στο μέγιστο κόστος εκτέλεσης του αλγορίθμου, κόστος που μετράται σε υπολογιστικούς πόρους. | 2. | 4. εισόδου
5. τετραγωνική
6. λογαριθμική
7. αιτιοκρατικοί |
| | 2. Τα δεδομένα συνιστούν το μέγεθος της εισόδου ενός αλγορίθμου. | 3. | 8. ΛΑΘΟΣ
9. ΛΑΘΟΣ
10. ΣΩΣΤΟ
11. ΣΩΣΤΟ |
| | 3. Ο απλούστερος τρόπος μέτρησης της επίδοσης ενός αλγορίθμου είναι ο <i>εμπειρικός</i> ή αλλιώς ο λεγόμενος <i>εκ των υστέρων</i> που εφαρμόζεται σε ένα σύνολο δεδομένων, ώστε να υπολογισθεί ο απαιτούμενος χρόνος επεξεργασίας και η χωρητικότητα μνήμης. | 4. | 12. Β, Γ, Ε
13. Α, Δ, Ε, Η, Θ |

ΚΕΦΑΛΑΙΟ 6

- | | | | |
|-----------|--|-----------|--|
| 1. | 1. πηγαίο
2. ιεραρχική
3. οπτικό, οδηγούμενο από τα γεγονότα | 3. | 8. Α
9. Β
10. Γ
11. Α |
| 2. | 4. ΣΩΣΤΟ
5. ΛΑΘΟΣ
6. ΛΑΘΟΣ
7. ΛΑΘΟΣ | 4. | 12. Β, Γ
13. Α, Β, Γ, Δ
14. Α, Δ |

ΚΕΦΑΛΑΙΟ 7

- | | | | |
|-----------|---|-----------|---------------------------------|
| 1. | 1. Γ, Β, Α, Δ
2. Β, Γ, Δ, Α | 4. | 9. Δ
10. Β
11. Α
12. Β |
| 2. | 3. μεταβλητές
4. ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
<i>όνομα προγράμματος</i> | 5. | 13. Α, Β, Δ, Ζ
14. Α, Β, Ζ |
| 3. | 5. ΣΩΣΤΟ
6. ΛΑΘΟΣ
7. ΣΩΣΤΟ
8. ΣΩΣΤΟ | | |

ΚΕΦΑΛΑΙΟ 8

- 1.** 1. Β, Ε, Δ, Α, Γ
2. Γ, Δ, Β, Α
- 2.** 3. ΛΑΘΟΣ
4. ΛΑΘΟΣ
5. ΣΩΣΤΟ
6. ΛΑΘΟΣ
7. ΣΩΣΤΟ

- 3.** 8. Β
9. Γ
10. Δ
11. Β
12. Γ
13. Β

ΚΕΦΑΛΑΙΟ 9

- 1.** 1. μονοδιάστατοι
2. στατική
3. *διαγωνίου*
- 2.** 4. ΛΑΘΟΣ
5. ΣΩΣΤΟ
6. ΛΑΘΟΣ
7. ΣΩΣΤΟ

- 3.** 8. Γ
9. Β
10. Α
- 4.** 11. Α, Δ, Ε
12. Γ, Δ

ΚΕΦΑΛΑΙΟ 10

- 1.** 1. τυπικών
2. διαδικασίες, συναρτήσεις
3. απεριόριστη
- 2.** 4. ΛΑΘΟΣ
5. ΣΩΣΤΟ
6. ΛΑΘΟΣ
7. ΛΑΘΟΣ

- 3.** 8. Α
9. Γ
10. Β
- 4.** 11. Α, Β
12. Α, Β

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

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

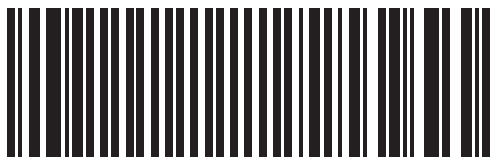
11010011 11110101 11100011 11110111 11100001
11110001 11100111 11110100 11011110 11110001
11101001 11100001 00101110 00100000 11000010
11110001 11011110 11101010 11100101 11110010
00100000 11110100 11101111 00100000 11101010
11110001 11110101 11101100 11011101 11101101
11101111 00100000 11101100 11011110 11101101
11110101 11101100 11100001 00101110 00100000
11000011 11101001 11100001 00100000 11100010
11101111 11011110 11101000 11100101 11101001
11100001 00100000 11110011 11110100 11100001
00100000 11101100 11100001 11101000 11011110
11101100 11100001 11110100 11011100 00100000
11110011 11101111 11110101 00100000 11110000
11011110 11100011 11100001 11101001 11101101
11100101 00100000 11110011 11110100 11101111
00100000 01110111 01110111 01110111 00101110
01110011 01110100 01110101 01100100 01111001
00110100 01100101 01111000 01100001 01101101
01110011 00101110 01100111 01110010

Κωδικός Βιβλίου: 0-22-0204

ISBN 978-960-06-2447-2



ΙΝΣΤΙΤΟΥΤΟ
ΤΕΧΝΟΛΟΓΙΑΣ
ΥΠΟΛΟΓΙΣΤΩΝ & ΕΚΔΟΣΕΩΝ



(01) 000000 0 22 0204 4