

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βλάχος Στέφανος

Σημειώσεις - Ερωτήσεις - Ασκήσεις
Κεφαλαίων 1, 2, 7

Ανάπτυξη Εφαρμογών σε προγραμματιστικό περιβάλλον

Συνοπτικές Σημειώσεις

ΚΕΦΑΛΑΙΟ 1

Η έννοια του προβλήματος (σελ 3)

Με τον όρο πρόβλημα εννοείται μια κατάσταση η οποία χρήζει αντιμετώπισης (απαιτεί λύση), η δε λύση δεν είναι γνωστή ούτε προφανής

Το πρόβλημα του έτους 2000 (millennium bug) (σελ 4)

Η ανάγκη για την μεγαλύτερη δυνατή εξοικονόμηση χώρου, κατά την κατασκευή των πρώτων υπολογιστών, οδήγησε στην χρησιμοποίηση μόνο των δύο τελευταίων ψηφίων για την αναπαράσταση του έτους σε έναν υπολογιστή. Για παράδειγμα η ημερομηνία 5-Ιαν-1998, συμβολιζόταν ως εξής: 50198. Αυτό σημαίνει ότι μπορούν να αναπαρασταθούν μόνο 100 έτη (00 – 99). Αυτή η αναπαράσταση του έτους δεν μας δίνει καμία πληροφορία για τον αιώνα ή την χιλιετία. Έτσι τα ζεύγη ετών 1900 και 1800, 1900 και 2000, 1965 και 2065 κλπ, δεν διαχωρίζονται με αυτό τον τρόπο αναπαράστασης.

Το πρόβλημα ήταν ότι μέχρι την τελευταία μέρα του 1999 δεν ήταν γνωστό πόσοι υπολογιστές ανά τον κόσμο είχαν τροποποιηθεί κατάλληλα για να αναγνωρίσουν το έτος 2000 και πόσοι θα το θεωρούσαν σαν το έτος 1900. Αυτή η αδυναμία των υπολογιστών θα μπορούσε να προκαλέσει τρομερές ανωμαλίες σε μεγάλα υπολογιστικά συστήματα με τεράστιες οικονομικές και κοινωνικές επιπτώσεις.

Κατανόηση προβλήματος (σελ 5-7)

Το πρώτο βήμα για την επίλυση ενός προβλήματος είναι η σωστή κατανόησή του. Για να είναι όμως σίγουρη η σωστή κατανόηση του προβλήματος απαιτείται η **σωστή (σαφής) διατύπωσή του**. Η παρουσίαση (διατύπωση) του προβλήματος μπορεί να γίνει με οποιοδήποτε τρόπο αρκεί να είναι αντιληπτή τουλάχιστον από μια από τις πέντε αισθήσεις. Συνηθέστεροι είναι ο λόγος, γραπτός ή προφορικός. Αν η διατύπωση έχει άστοχη χρήση ορολογίας ή λανθασμένη σύνταξη είναι πολύ πιθανό να γίνει παρερμηνεία από αυτόν που θα κληθεί να το λύσει. Παρερμηνεία μπορεί να έχουμε και στην περίπτωση της ασάφειας ή της παράληψης κάποιων στοιχείων.

Ακόμα και αν η διατύπωση είναι πλήρης, θα πρέπει ο λύτης να είναι προσεκτικός και να αναλύσει όλες τις πτυχές και τις λεπτομέρειες για να εξασφαλίσει την **σωστή ερμηνεία** του προβλήματος.

Καθορισμός Απαιτήσεων (σελ 11-16)

Με τον όρο «**καθορισμός απαιτήσεων**» εννοούμε την σωστή καταγραφή των δεδομένων και των ζητούμενων ενός προβλήματος. Βασική προϋπόθεση για τον καθορισμό των απαιτήσεων είναι η **σωστή ερμηνεία** του προβλήματος, που προηγείται.

Ορισμοί (σελ 8)

Δεδομένο: Με τον όρο **δεδομένο** δηλώνεται οποιοδήποτε στοιχείο μπορεί να γίνει αντιληπτό από έναν τουλάχιστον παρατηρητή με μια από τις πέντε αισθήσεις του.

Πληροφορία: Με τον όρο **πληροφορία** αναφέρεται οποιοδήποτε γνωσιακό στοιχείο παρέχεται από επεξεργασία δεδομένων.

Επεξεργασία δεδομένων: Ο όρος **επεξεργασία δεδομένων** δηλώνει εκείνη τη διαδικασία κατά την οποία ένας «μηχανισμός» δέχεται δεδομένα, τα επεξεργάζεται σύμφωνα με έναν προκαθορισμένο τρόπο και αποδίδει πληροφορίες.

Δομή προβλήματος (σελ 8-11)

Με τον όρο **δομή** ενός προβλήματος αναφερόμαστε στα συστατικά του μέρη, στα επιμέρους τμήματα που το αποτελούν καθώς επίσης και τον τρόπο που αυτά συνδέονται μεταξύ τους.

Η διαδικασία της **ανάλυσης του προβλήματος** έγκειται στην εύρεση της δομής του προβλήματος. Τα **πλεονεκτήματα** της ανάλυσης είναι:

1. Το πρόβλημα αναλύεται σε απλούστερα που είναι ευκολότερο να λυθούν, άρα απλοποιείται και η επίλυση του αρχικού
2. Η όλη διαδικασία αμβλύνει την σκέψη και την αναλυτική ικανότητα του ατόμου.

Η δομή του προβλήματος που προκύπτει από την ανάλυση μπορεί να αναπαρασταθεί φραστικά ή γραφικά. Η συνηθέστερη γραφική αναπαράσταση είναι η λεγόμενη **διαγραμματική αναπαράσταση**. Η δημιουργία ενός τέτοιου διαγράμματος βοηθάει

1. στην καλύτερη κατανόηση του προβλήματος από τον λύτη.
2. στην σχεδίαση της ίδιας της λύσης.

Στάδια Επίλυσης Προβλήματος (σελ 16)

Συμπερασματικά από όλα τα παραπάνω διαφαίνεται ότι τα στάδια επίλυσης ενός προβλήματος είναι με τη σειρά που αναφέρονται:

1. **Η κατανόηση.** Περιλαμβάνει την κατανόηση του προβλήματος και τον καθορισμό των απαιτήσεων.
2. **Η ανάλυση.** Αφορά στην εύρεση της δομής του προβλήματος
3. **Η επίλυση.** Υλοποιείται η λύση μέσω της λύσης των επιμέρους προβλημάτων που προέκυψαν από την ανάλυση.

Κατηγορίες Προβλημάτων (σελ 16-18)

Τα προβλήματα χωρίζονται σε κατηγορίες βάση κάποιων χαρακτηριστικών.

- 1) **Βάση της δυνατότητας επίλυσης σε**
 - i. Επιλύσιμα
 - ii. Ανοικτά
 - iii. Άλυτα
- 2) **Βάση του βαθμού δόμησης σε**
 - i. Δομημένα
 - ii. Ημιδομημένα
 - iii. Αδόμητα
- 3) **Βάση του είδους της επίλυσης σε**
 - i. Απόφασης
 - ii. Υπολογιστικά
 - iii. Βελτιστοποίησης

Πρόβλημα και υπολογιστής (σελ 18-19)

Οι λόγοι για τους οποίους αναθέτουμε την επίλυση ενός προβλήματος σε υπολογιστές σχετίζονται με:

1. την πολυπλοκότητα των πράξεων
2. την επαναληπτικότητα των διαδικασιών
3. την ταχύτητα εκτέλεσης
4. το μεγάλο πλήθος των δεδομένων

Οι πράξεις που μπορεί να εκτελέσει ένας υπολογιστής είναι:

1. Η **πρόσθεση**. Όλες οι άλλες πράξεις υλοποιούνται με τον κατάλληλο συνδυασμό προσθέσεων
2. Η **σύγκριση**. Μας βοηθά στην εκτέλεση λογικών πράξεων και στην λήψη αποφάσεων
3. Η **μεταφορά δεδομένων**. Χρησιμοποιείται για να ληφθούν τα δεδομένα και να εξαχθούν τα αποτελέσματα που παράγονται από την εκτέλεση των επιθυμητών πράξεων.

Ερωτήσεις Κεφαλαίου 1

Ερωτήσεις Ανάπτυξης

1. Να δοθεί ο ορισμός του προβλήματος.(σελ 3)
2. Να περιγραφεί το πρόβλημα του 2000. (σελ 4)
3. Από ποιους παράγοντες εξαρτάται η κατανόηση ενός προβλήματος (σελ 5)
4. Ποια μπορεί να είναι η μορφή διατύπωσης ενός προβλήματος; (σελ 6)
5. Να δοθούν οι ορισμοί των όρων «δεδομένα», «επεξεργασία δεδομένων» και «πληροφορία». (σελ 8)
6. Τι εννοούμε με τον όρο «δομή προβλήματος»; (σελ 8)
7. Ποια είναι τα πλεονεκτήματα από την εύρεση της δομής ενός προβλήματος; (σελ 8)
8. Πώς μπορεί να αναπαρασταθεί η δομή ενός προβλήματος; (σελ 10)
9. Ποια είναι τα στάδια επίλυσης ενός προβλήματος; (σελ 16)
10. Με ποια γενικά χαρακτηριστικά μπορούμε να κατηγοριοποιήσουμε τα προβλήματα; (σελ 16-17)
11. Κατηγοριοποιήστε τα προβλήματα με βάση την δυνατότητα επίλυσης. (σελ 16)
12. Κατηγοριοποιήστε τα προβλήματα με βάση το βαθμό δόμησης (σελ 17)
13. Κατηγοριοποιήστε τα προβλήματα με βάση του είδους της επίλυσης. (σελ 17-18)
14. Με τι σχετίζονται οι λόγοι που αναθέτουμε την επίλυση προβλημάτων σε υπολογιστή; (σελ 19)
15. Ποιες βασικές λειτουργίες επιτελεί ένας υπολογιστής; (σελ 19)

Ερωτήσεις «Σωστό – Λάθος»

1. Πρόβλημα είναι μια μαθηματική κατάσταση που πρέπει να αντιμετωπιστεί.
2. Το πρόβλημα του 2000 προέρχεται από το γεγονός ότι οι υπολογιστές χρησιμοποιούσαν 2 ψηφία για την απεικόνιση του έτους.
3. Η επίλυση ενός προβλήματος δεν εξαρτάται άμεσα από την διατύπωσή του.
4. Η διατύπωση ενός προβλήματος μπορεί να γίνει με ποικίλους τρόπους.
5. Ο μοναδικός μηχανισμός επεξεργασίας είναι ο ανθρώπινος εγκέφαλος.
6. Για την παραγωγή πληροφοριών απαιτούνται δεδομένα.
7. Όλα προβλήματα επιλύονται με την βοήθεια υπολογιστή.
8. Η εύρεση της δομής ενός προβλήματος οξύνει την σκέψη του ανθρώπου.
9. Ένα πρόβλημα δεν μπορεί να αναλυθεί σε πολλά επιμέρους προβλήματα.
10. Ο καθορισμός των απαιτήσεων ενός προβλήματος προϋποθέτει τον προσδιορισμό των δεδομένων και την καταγραφή των ζητούμενων.
11. Τα στάδια επίλυσης ενός προβλήματος είναι κατά σειρά η κατανόηση, η ανάλυση και η επίλυση.
12. Ο υπολογιστής εκτελεί τον πολλαπλασιασμό με τη βοήθεια της πρόσθεσης.
13. Ανοικτά είναι τα προβλήματα για τα οποία έχει αποδειχθεί ότι δεν έχουν λύση.
14. Στα αδύμητα προβλήματα η λύση δεν προέρχεται από μια αυτοματοποιημένη διαδικασία.
15. Η επίλυση της πρωτοβάθμιας εξίσωσης είναι αδύμητο πρόβλημα.
16. Ένας από τους λόγους, που χρησιμοποιούμε τον υπολογιστή για την επίλυση προβλημάτων, είναι η δυνατότητά του να επεξεργάζεται μεγάλο πλήθος δεδομένων.
17. Οι υπολογιστές χρησιμοποιούνται για την επίλυση άλυτων προβλημάτων.
18. Οι υπολογιστές χρησιμοποιούνται και για την επίλυση ημιδομημένων προβλημάτων.

ΚΕΦΑΛΑΙΟ 2

Το όνομα Αλγόριθμος (σελ 25)

Ο άραβας μαθηματικός Abu Ja'far Mohammed ibn Musa **al Khowarizmi**, ήταν ο πρώτος που περιέγραψε **αυτοματοποιημένους** τρόπους για την επίλυση μαθηματικών εξισώσεων. Από το όνομά του έμεινε ο όρος «**αλγόριθμος**» που περιέγραφε αρχικά τη «**Συστηματική διαδικασία αριθμητικών χειρισμών**». Σήμερα χρησιμοποιούμε αυτοματοποιημένες διαδικασίες για την επίλυση διαφόρων προβλημάτων από τον υπολογιστή οι οποίες χαρακτηρίζονται με τον όρο **αλγόριθμος**.

Ορισμός Αλγορίθμου (σελ 25)

Αλγόριθμος είναι μια πεπερασμένη σειρά , αυστηρά καθορισμένων εντολών και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

Ο αλγόριθμος **δεν είναι η λύση**, δηλαδή η απάντηση, σε ένα πρόβλημα. Μας δίνει την **επίλυση** του προβλήματος, δηλαδή περιγράφει τα βήματα που πρέπει να ακολουθήσουμε για να φτάσουμε στη λύση. Ένας αλγόριθμος δεν είναι απαραίτητα μοναδικός για την επίλυση ενός προβλήματος. Μπορεί να υπάρχει και άλλη αλληλουχία εντολών για την επίλυση του ίδιου προβλήματος.

Κριτήρια Αλγορίθμων (σελ 25-26)

Κάθε αλγόριθμος πρέπει να ικανοποιεί τα ακόλουθα κριτήρια.

- 1) **Είσοδος(input)**. Καμία, μία ή και περισσότερες τιμές δεδομένων πρέπει να δίνονται ως εισοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων εντολών.
- 2) **Έξοδος(output)**. Ο αλγόριθμος πρέπει να δημιουργεί **τουλάχιστον μια** τιμή δεδομένων ως αποτέλεσμα προς τον χρήστη ή προς έναν άλλο αλγόριθμο.
- 3) **Καθοριστικότητα(definiteness)**. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμιά αμφιβολία για τον τρόπο εκτέλεσης.
- 4) **Περατότητα(finiteness)**. Ο αλγόριθμος πρέπει να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του.
- 5) **Αποτελεσματικότητα(effectiveness)**. Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή.

Σπουδαιότητα αλγορίθμων (σελ 27)

Οι αλγόριθμοι αποτελούν ένα από τα βασικά γνωστικά πεδία που μελετά η επιστήμη της πληροφορικής. Αυτή η μελέτη γίνεται από τις ακόλουθες σκοπιές:

- 1) **Υλικού** (Hardware)
- 2) **Γλωσσών** Προγραμματισμού (programming languages)
- 3) **Θεωρητική** (theoretical)
- 4) **Αναλυτική** (analytical)

Περιγραφή και αναπαράσταση αλγορίθμων (σελ 28)

Οι συνηθέστεροι τρόποι για την αναπαράσταση ενός αλγορίθμου είναι:

- 1) **Με ελεύθερο κείμενο.** Είναι ο πιο **αδόμητος** τρόπος παρουσίασης. Αυτό σημαίνει ότι δεν υπάρχει κάποια συγκεκριμένη οργάνωση στον τρόπο παρουσίασης και είναι εύκολο η περιγραφόμενη διαδικασία να παρερμηνευτεί. Επίσης είναι δύσκολη η μεταφορά του αλγορίθμου σε έναν υπολογιστή.
- 2) **Με φυσική γλώσσα σε βήματα.** Καλύτερη μέθοδος από το ελεύθερο κείμενο. Παρουσιάζονται όμως περιπτώσεις όπου τα βήματα (οι εντολές) δεν παρουσιάζονται με τον πλέον σαφή τρόπο.
- 3) **Με διαγραμματικές τεχνικές (διάγραμμα ροής).** Το διάγραμμα ροής είναι η επικρατέστερη από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί κατά καιρούς. Αρκετά πιο δομημένη μέθοδος από τις δυο προηγούμενες, αλλά δεν χρησιμοποιείται πολύ γιατί σε μεγάλους αλγορίθμους είναι αρκετά πολύπλοκη, και μεταφέρεται σχετικά δύσκολα σε υπολογιστή.
- 4) **Με κωδικοποίηση (ψευδοκώδικας - ψευδογλώσσα).** Είναι η πιο δομημένη αναπαράσταση από τις υπόλοιπες. Διατηρεί τις περισσότερες από τις αρχές των γλωσσών προγραμματισμού και αυτό την κάνει εύκολα μεταφέρσιμη σε υπολογιστή

Δομή ακολουθίας (σελ 30)

Η ακολουθιακή δομή εντολών (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου εντολών. Στην ακολουθιακή δομή εκτελούνται **όλες** οι εντολές **μια φορά** η κάθε μια και **δεν** χρειάζεται να πάρουμε υποπεριπτώσεις (εντολή **αν**) ή να επαναλάβουμε κάποιες εντολές (εντολές επανάληψης).

Σημείωση : Οι Δομές Επιλογής και Επανάληψης του 2^{ου} Κεφαλαίου θα αναλυθούν μαζί με το Κεφάλαιο 8.

ΚΕΦΑΛΑΙΟ 7

Τι είναι η γλώσσα προγραμματισμού

Η γλώσσα που χρησιμοποιούμε για να επικοινωνούμε με τους άλλους ανθρώπους έχει βασιστεί σε κάποιους κανόνες. Χωρίς αυτούς τους κανόνες δεν θα μπορούσαμε να είμαστε σαφείς και επομένως δεν θα μπορούσαμε να επικοινωνούμε. Αν θεωρήσουμε ότι επικοινωνία ανθρώπου και υπολογιστή είναι η καθοδήγηση του υπολογιστή για το πώς θα δουλέψει, τότε **γλώσσα προγραμματισμού είναι ένα σύνολο κανόνων (συντακτικών, γραμματικών και ορθογραφικών) που μας καθορίζει πως μπορούμε να περιγράψουμε σε ένα υπολογιστή, τι διαδικασίες θα εκτελέσει.**

Η γλώσσα προγραμματισμού ΓΛΩΣΣΑ του βιβλίου (σελ 147)

Η επιλογή μιας γλώσσας προγραμματισμού δεν είναι εύκολη υπόθεση. **Εξαρτάται από το είδος του προγράμματος, το διαθέσιμο εξοπλισμό και σαφώς τις γνώσεις και τις προτιμήσεις του προγραμματιστή.**

Επειδή οι περισσότερες γλώσσες προγραμματισμού έχουν πολλά κοινά στοιχεία δεν υπάρχει λόγος να διδαχθεί κάποια συγκεκριμένη. Σημασία έχει να φανούν οι ιδιαιτερότητες και οι περιορισμοί που θέτει μια γλώσσα προγραμματισμού στον προγραμματιστή. Γι' αυτό το λόγο καταφεύγουμε στη χρήση μιας **εικονικής γλώσσας προγραμματισμού**, η οποία **διατηρεί** τους περιορισμούς των γλωσσών προγραμματισμού αλλά έχει το δικό της ιδιαίτερο συντακτικό και λεξιλόγιο. Η ονομασία της είναι **ΓΛΩΣΣΑ** και θα την χρησιμοποιήσουμε για την αναπαράσταση των αλγορίθμων.

Αλφάβητο Γλώσσας (σελ 148)

Στη γλώσσα προγραμματισμού ΓΛΩΣΣΑ έχουμε το δικαίωμα να χρησιμοποιήσουμε γράμματα του ελληνικού και λατινικού αλφαβήτου, αριθμούς και σύμβολα. Συγκεκριμένα:

Γράμματα: A-Ω, α-ω, A-Z, a-z
Ψηφία: 0-9
Ειδικό χαρακτήρες: + - _ = () * & < > (κενός χαρακτήρας) ^ , .

Τύποι δεδομένων (σελ 148-149)

Στους αλγόριθμους και στα προγράμματα χρειάζεται πολλές φορές να χειριστούμε και να αναπαραστήσουμε κάποια δεδομένα. Για παράδειγμα μπορεί να χρειαστεί να αναπαραστήσουμε μια ποσότητα απόστασης (πραγματικός), έναν αριθμό μαθητών (ακέραιος), ένα ποσοστό (πραγματικός), μια φράση (χαρακτήρας) κ.α.. Ο υπολογιστής για εξοικονομήσει χώρο και για να χειριστεί ξεχωριστά τα διάφορα δεδομένα, υποστηρίζει διάφορους **τύπους δεδομένων**. Εμείς είμαστε υποχρεωμένοι όταν θέλουμε να αποθηκεύσουμε, να αλλάξουμε ή γενικά να χειριστούμε κάποια πληροφορία (ή δεδομένο) να προειδοποιήσουμε τον υπολογιστή για το είδος της πληροφορίας. Στη **ΓΛΩΣΣΑ** θεωρούμε ότι οι δυνατοί **τύποι δεδομένων** είναι:

Ακέραιος τύπος. Τον χρησιμοποιούμε για να υποδηλώσουμε στον υπολογιστή ότι κάποιο δεδομένο είναι αριθμός και μάλιστα ακέραιος, δηλαδή δεν έχει δεκαδικά ψηφία. Οι ακέραιοι μπορεί να είναι θετικοί αρνητικοί ή μηδέν.

Πραγματικός τύπος. Τον χρησιμοποιούμε για να υποδηλώσουμε στον υπολογιστή ότι κάποιο δεδομένο είναι αριθμός ο οποίος μπορεί να περιέχει και δεκαδικά ψηφία. Παραδείγματα πραγματικών αριθμών είναι οι αριθμοί 2.2, 3.14, 0.25 κ.α..

Χαρακτήρας. Χρησιμοποιείται όταν να αποθηκεύσουμε και να χειριστούμε έναν χαρακτήρα ή μια ακολουθία χαρακτήρων. Σε μια ποσότητα αυτού του τύπου μπορούμε να βάλουμε οποιοδήποτε χαρακτήρα του πληκτρολογίου. Για να δηλώσουμε ότι μια φράση είναι δεδομένο τύπου χαρακτήρας θα πρέπει να την βάλουμε μέσα σε εισαγωγικά, για παράδειγμα

‘καλημέρα κόσμε’. Επειδή σε μια ποσότητα αυτού του τύπου μπορούμε να βάλουμε τόσο αριθμούς όσο και γράμματα της αλφαβήτου, συχνά τις αναφέρουμε και σαν **αλφαριθμητικά**. Άλλες ονομασίες είναι **λεκτικά**, **συμβολοσειρές** ή **strings**. **Προσοχή**: η ακολουθία χαρακτήρων ‘20’ είναι τύπου χαρακτήρα ενώ η ακολουθία 20 (χωρίς εισαγωγικά) είναι ένας ακέραιος αριθμός.

Λογικός. Πολλές φορές θέλουμε απλά να αποθηκεύσουμε την πληροφορία για το αν ένα γεγονός ισχύει ή όχι. Επειδή σε αυτή τη περίπτωση έχουμε μόνο δύο τιμές και για να μην σπαταλήσουμε χώρο στον υπολογιστή για την αποθήκευση της πληροφορίας, χρησιμοποιούμε ένα ξεχωριστό τύπο δεδομένων που ονομάζεται **Λογικός τύπος δεδομένων**. Οι τιμές μιας τέτοιας ποσότητας είναι μόνο **αληθές** ή **ψευδές** και συνήθως τις ονομάζουμε **λογικές τιμές**.

Ονόματα – Ονοματολογία (σελ 150)

Όταν θέλουμε να δώσουμε όνομα σε μια ποσότητα δεδομένων σε ένα πρόγραμμα (και σε αλγόριθμο) πρέπει να τηρήσουμε κάποιους κανόνες. Συγκεκριμένα μπορούμε να χρησιμοποιήσουμε κεφαλαία ή μικρά γράμματα του ελληνικού ή λατινικού αλφαβήτου, ψηφία και την κάτω παύλα (‘ ’). Δεν επιτρέπεται κανένα άλλο σύμβολο. Δεν μπορούμε να έχουμε ένα όνομα που να αρχίζει από αριθμό. Δεν μπορούμε να χρησιμοποιήσουμε **δεσμευμένες λέξεις**. Τα ονόματα που ορίζονται από τον προγραμματιστή ονομάζονται **αναγνωριστικά**.

Όπως κάθε γλώσσα (ανθρώπινη) έτσι και οι γλώσσες προγραμματισμού έχουν ένα συγκεκριμένο **λεξιλόγιο**. Οι λέξεις μιας γλώσσας έχουν συγκεκριμένη σημασία και δεν μπορούμε να την παραβλέψουμε ή να την παραποιήσουμε. Οι λέξεις που μας ορίζει η ΓΛΩΣΣΑ εξ’ αρχής ονομάζονται **δεσμευμένες λέξεις** και δεν μπορούν να χρησιμοποιηθούν, όπως προαναφέραμε για τον ορισμό ενός δικού μας ονόματος.

Σταθερές (σελ 149)

Οι σταθερές (constants) είναι προκαθορισμένες τιμές που δεν μεταβάλλονται κατά την εκτέλεση του προγράμματος (ή του αλγορίθμου γενικότερα). Ανάλογα με το τι είδους είναι μια σταθερή ποσότητα, μπορούμε να έχουμε **ακέραιες**, **πραγματικές**, **αλφαριθμητικές** ή **λογικές σταθερές**. Για παράδειγμα οι παρακάτω ποσότητες είναι σταθερές διαφόρων τύπων: 2, 5.6, ‘καλημέρα’, ‘8.2 και 5.1’, αληθής, ‘αληθής’, -342 κ.λ.π.

Προσοχή: Η σταθερά αληθής είναι λογική τιμή, ενώ η σταθερά ‘αληθής είναι αλφαριθμητικό.

Συμβολικές Σταθερές (σελ 149-150)

Όπως σε κάθε γλώσσα προγραμματισμού έτσι και στην ΓΛΩΣΣΑ μπορούμε να χρησιμοποιήσουμε συμβολικά ονόματα (αναγνωριστικά) για να ονομάσουμε τις σταθερές μας. Τα ονόματα που θα χρησιμοποιήσουμε πρέπει να τηρούν του παραπάνω κανόνες ονοματολογίας. Για να δηλώσουμε σε ένα πρόγραμμα τι ονόματα θέλουμε για τις σταθερές μας (αν θέλουμε), πρέπει στο τμήμα των δηλώσεων (**βλέπε. 7.10 δομή προγράμματος σελ 157**) να γράψουμε τη δεσμευμένη λέξη **Σταθερές** ακολουθούμενη από τα ονόματα και τις τιμές που επιθυμούμε, όπως παρακάτω.

Σταθερές

<όνομα 1> = <τιμή 1>

<όνομα 2> = <τιμή 2>

.

.

.

<όνομα n> = <τιμή n>

Σταθερές

π = 3.14159

ΦΠΑ = 0.19

Ποσοστό = 0.25

Όνομα = ‘Κώστας’

Σημείωση: Η χρήση των **συμβολικών σταθερών** μπορεί να μοιάζει περιττή εφόσον οι ποσότητες που αναπαριστούν είναι σταθερές. Πολλές φορές όμως ενδέχεται να έχουμε μια σταθερά σε πολλά σημεία ενός προγράμματος. Αν την γράφουμε σε όλο το πρόγραμμα σαν σταθερά (και όχι σαν συμβολική σταθερά) και χρειαστεί να κάνουμε κάποια αλλαγή στην τιμή της, θα πρέπει να την αλλάξουμε όπου και αν εμφανίζεται μέσα στο πρόγραμμα. Αν όμως έχουμε φροντίσει να την ορίσουμε στο τμήμα δηλώσεων σαν συμβολική σταθερά, τότε οποιαδήποτε αλλαγή τιμής μπορεί να γίνει στην δήλωση και αυτομάτως να ισχύσει για όλο το πρόγραμμα.

Μεταβλητές (σελ 151-152)

Οι μεταβλητές είναι ποσότητες οι οποίες μπορούν να μεταβάλλονται κατά την διάρκεια εκτέλεσης ενός προγράμματος ή ενός αλγορίθμου γενικότερα. Επειδή οι ποσότητες αυτές είναι μεταβλητές, είμαστε υποχρεωμένοι να τους δώσουμε ένα **όνομα** με το οποίο μπορούμε να αναφερόμαστε σε αυτές, το οποίο υπόκειται στους περιορισμούς της ονοματολογίας που προαναφέραμε (σελ 150 βιβλίου). Το όνομα μιας μεταβλητής ονομάζεται και **αναγνωριστικό**. Στον προγραμματισμό οι μεταβλητές αντιστοιχούν σε κάποια θέση μνήμης στον υπολογιστή. Αν αλλάξουμε την τιμή της μεταβλητής αλλάζουμε το περιεχόμενο της θέσης μνήμης. Οι μεταβλητές μπορούν, όπως και οι σταθερές, να είναι **ακέραιες, πραγματικές, χαρακτήρα** ή **λογικές**. Για να χρησιμοποιήσουμε σε ένα πρόγραμμα κάποιες μεταβλητές είμαστε υποχρεωμένοι να τις «δηλώσουμε» στο τμήμα δηλώσεων του προγράμματος (**βλέπε. 7.10 δομή προγράμματος σελ 157**) όπως και τις συμβολικές σταθερές. Για παράδειγμα:

Μεταβλητές

<τύπος 1> : <ονόματα μεταβλητών>
<τύπος 2> : <ονόματα μεταβλητών>
.
.
.
<τύπος ν> : <ονόματα μεταβλητών>

Μεταβλητές

Πραγματικές : α, ποσοστό
Ακέραιες : β, κόστος
Λογικές: ισχύει
Χαρακτήρα : όνομα

Σημείωση: οι μόνοι τρόποι για να αλλάξουμε την τιμή μιας μεταβλητής είναι με την **εντολή εκχώρησης** και με την **εντολή ανάγνωσης(Διάβαση)** που παρουσιάζονται παρακάτω.

Τελεστές (σελ 152, 31, 46)

Οι τελεστές είναι ένα σύνολο συμβόλων κάθε ένα από τα οποία αντιπροσωπεύει μια πράξη. Οι πράξεις αυτές εκτελούνται μεταξύ σταθερών και μεταβλητών κατά την δική μας βούληση. Οι τελεστές χωρίζονται σε **αριθμητικούς, συγκριτικούς** και **λογικούς**. Αναλυτικότερα:

Αριθμητικοί : +, -, /, *, ^, div, mod

Συγκριτικοί : <, >, =, <>, >=, <=

Λογικοί : και (σύζευξη), ή (διάζευξη), όχι (άρνηση)

Συναρτήσεις (σελ 153)

Πολλές συναρτήσεις από τα μαθηματικά χρησιμοποιούνται συχνά στη ΓΛΩΣΣΑ. Οι συναρτήσεις είναι πολύπλοκες λειτουργίες που μας παρέχονται έτοιμες και μπορούμε να τις χρησιμοποιήσουμε σαν εντολές μέσα στο πρόγραμμά μας.

HM(X)	Υπολογισμός ημιτόνου
ΣΥΝ(X)	Υπολογισμός συνημιτόνου
ΕΦ(X)	Υπολογισμός εφαπτομένης
T_P(X)	Υπολογισμός τετραγωνικής ρίζας
ΛΟΓ(X)	Υπολογισμός φυσικού λογαρίθμου
E(X)	Υπολογισμός του e^x
A_M(X)	Ακέραιο μέρος του X
A_T(X)	Απόλυτη τιμή του X

Εκφράσεις (σελ 153-154, 31)

Οι εκφράσεις είναι παραστάσεις που αποτελούνται από τελεστές, σταθερές και μεταβλητές. Για να υπολογιστεί μια έκφραση πρέπει να αποδοθούν τιμές στις μεταβλητές και να γίνουν οι πράξεις με την κατάλληλη σειρά, όπως αυτή ορίζεται από την προτεραιότητα των πράξεων και τις παρενθέσεις που μπορεί να υπάρχουν. Μια έκφραση μπορεί να είναι από μια μόνο μεταβλητή ή σταθερά, μέχρι μια πολύπλοκη μαθηματική έκφραση.

Οι εκφράσεις χωρίζονται σε **αριθμητικές** και **λογικές**.

Οι **αριθμητικές** χρησιμοποιούν **αριθμητικούς τελεστές, αριθμητικές σταθερές και αριθμητικές μεταβλητές**.

π.χ. Αριθμητική Έκφραση: $(3 * \chi - 2) / (5 - \upsilon^3)$, όπου χ, υ αριθμητικές μεταβλητές.

Οι **λογικές** μπορούν να χρησιμοποιούν **λογικούς τελεστές, συγκριτικούς τελεστές, λογικές σταθερές και λογικές μεταβλητές**.

π.χ. Λογική Έκφραση: $\text{όχι}((3 > \gamma) \text{ και } (15 < \gamma))$, όπου γ λογική μεταβλητή.

Ιεραρχία αριθμητικών τελεστών (σελ 154)

Σε μια αριθμητική έκφραση η σειρά εκτέλεσης των πράξεων καθορίζεται από την παρακάτω ιεραρχία.

- 1) Δυνάμεις (^)
- 2) Πολλαπλασιασμός, διαίρεση, ακέραιο πηλίκο, ακέραιο υπόλοιπο (*, /, div, mod)
- 3) Πρόσθεση και αφαίρεση (+, -)

Δύο σημαντικοί κανόνες για την σειρά εκτέλεσης των πράξεων είναι:

- 1) υπολογίζονται πρώτα από όλα οι εκφράσεις μέσα στις παρενθέσεις
- 2) αν έχουμε να επιλέξουμε μεταξύ δύο ή παραπάνω τελεστών της ίδιας προτεραιότητας τους εκτελούμε όλους από τα αριστερά προς τα δεξιά.

Εντολή Εκχώρησης (σελ 154)

Η εντολή εκχώρησης χρησιμοποιείται για να μεταβάλλουμε την τιμή μιας μεταβλητής. Ορίζεται ως εξής: **<όνομα μεταβλητής> ← <έκφραση>**. Για να εκτελεστεί μια εντολή εκχώρησης θα πρέπει να υπολογιστεί η έκφραση στα δεξιά του τελεστή '←' και το αποτέλεσμα να εκχωρηθεί (αποθηκευτεί) στην μεταβλητή αριστερά.

Προσοχή: ο τύπος δεδομένων της μεταβλητής και ο τύπος της έκφρασης πρέπει να είναι ίδιοι. Δεν μπορούμε για παράδειγμα να εκχωρήσουμε σε μια ακέραιη μεταβλητή την τιμή μιας λογικής παράστασης. Για παράδειγμα:

A ← 134	(η A πρέπει να είναι ακέραιη)
B ← 12.24	(η B πρέπει να είναι πραγματική)
Γ ← αληθής	(η Γ πρέπει να είναι λογική)
Δ ← 'Καλημέρα'	(η Δ πρέπει να είναι τύπου χαρακτήρα)
A ← 2 * E - 5	(οι A και E πρέπει να είναι ακέραιες)

Εντολές Εισόδου Εξόδου (σελ 155-156)

Σε έναν αλγόριθμο ή πρόγραμμα που έχουμε δεδομένα και ζητούμενα, είναι προφανής η ανάγκη να βρούμε ένα τρόπο να δώσουμε τις τιμές των δεδομένων, έτσι ώστε να γίνει η εκτέλεση των βημάτων του αλγορίθμου, και να πάρουμε τα ζητούμενα σαν αποτέλεσμα της διαδικασίας. Η είσοδος των δεδομένων και η έξοδος των αποτελεσμάτων είναι ουσιαστικά ο μόνος τρόπος επαφής του αλγορίθμου ή του προγράμματος με τον «εξωτερικό κόσμο».

Για την είσοδο των δεδομένων σε έναν αλγόριθμο και σε ένα πρόγραμμα χρησιμοποιούμε την εντολή **Διάβασε**. Η εντολή συνοδεύεται με μια λίστα μεταβλητών, για τις οποίες θέλουμε να πάρουμε τιμές (συνήθως τα δεδομένα).

Σύνταξη

Διάβασε <ονόματα μεταβλητών χωρισμένα με κόμμα>

Παράδειγμα

Διάβασε α, ποσοστό, β, κόστος

Όταν εκτελούμε έναν αλγόριθμο ή ένα πρόγραμμα (μετά την συγγραφή τους) και συναντήσουμε την εντολή **Διάβασε**, διακόπτουμε την εκτέλεση και ζητούμε τόσες τιμές όσες είναι και οι μεταβλητές στην εντολή **Διάβασε**. Όταν πρόκειται για πρόγραμμα που εκτελείται στον υπολογιστή, τότε ο υπολογιστής σταματά την εκτέλεση του προγράμματος και περιμένει την πληκτρολόγηση των απαραίτητων τιμών. Μόλις δοθούν οι απαιτούμενες τιμές, τότε αυτές αποθηκεύονται στις μεταβλητές της εντολής, με τη σειρά που δόθηκαν.

Όταν, κατά την εκτέλεση ενός προγράμματος ή ενός αλγορίθμου, συναντήσουμε την εντολή **Γράψε** (Εμφάνισε, Εκτύπωσε), τότε εμφανίζουμε στον χρήστη τη λίστα των στοιχείων που ορίζεται στην εντολή. Αν πρόκειται για πρόγραμμα στον υπολογιστή, τα επιθυμητά αποτελέσματα εμφανίζονται στην οθόνη.

Με την εντολή **Γράψε** μπορούμε να εμφανίσουμε σταθερές τιμές (αριθμητικές, χαρακτήρα, λογικές), μεταβλητές ή και εκφράσεις.

Όταν συναντάται μια σταθερά στην εντολή **Γράψε**, τότε εμφανίζεται η σταθερά όπως παρουσιάζεται.

Όταν συναντάται το όνομα μιας μεταβλητής στην εντολή **Γράψε**, τότε εμφανίζεται η τιμή της μεταβλητής και όχι το όνομά της.

Όταν συναντάται μια έκφραση στην εντολή **Γράψε**, τότε εμφανίζεται η τιμή της, αφού πρώτα υπολογιστεί.

Σύνταξη

Γράψε <λίστα στοιχείων>

Παράδειγμα εκτύπωσης σταθερών

Γράψε 'καλημέρα κόσμε', 5, αληθής

Παράδειγμα εκτύπωσης μεταβλητών

Γράψε α, κόστος, όνομα

Παράδειγμα εκτύπωσης εκφράσεων

Γράψε α*β, (α<β)

Παράδειγμα εκτύπωσης διαφόρων στοιχείων

Γράψε 'το όνομά μου είναι ', όνομα, ' και είμαι', χ, ' ετών'

Δομή Προγράμματος (σελ 157)

Πρόγραμμα <όνομα προγράμματος> (ακολουθεί τους κανόνες ονοματολογίας)
Σταθερές (αν θέλουμε σταθερές)

<Δηλώσεις σταθερών>

Μεταβλητές (αν θέλουμε μεταβλητές)

<Δηλώσεις Μεταβλητών>

Αρχή

<εντολές>

Τέλος_Προγράμματος

Ερωτήσεις –Ασκήσεις κεφαλαίων 2 και 7

Ερωτήσεις Ανάπτυξης

1. Τι είναι αλγόριθμος και ποια τα κριτήρια που πρέπει να ικανοποιεί (σελ 25-26);
2. Από ποιες σκοπιές μελετά η πληροφορική τους αλγόριθμους (σελ 27);
3. Ποιοι οι τρόποι αναπαράστασης ενός αλγόριθμου (σελ 28);
4. Τι είναι διάγραμμα ροής, ποια τα γεωμετρικά σχήματα που χρησιμοποιούν και ποια η χρήση τους (σελ 28-29);
5. Τι είναι σταθερές και μεταβλητές και σε ποιους τύπους διακρίνονται (σελ 31, 149, 151);
6. Τι είναι οι συμβολικές σταθερές (σελ 149);
7. Ποια τα αποδεκτά ονόματα μεταβλητών, σταθερών, αλγορίθμων και προγραμμάτων (ονοματολογία) (σελ 150);
8. Τι είναι τελεστές και σε ποιες κατηγορίες χωρίζονται (σελ 31);
9. Ποιοι τελεστές όλων των κατηγοριών υπάρχουν (σελ 46, 152, 153);
10. Τι είναι έκφραση και τι αριθμητική έκφραση (σελ 31, 153);
11. Τι είναι εντολή εκχώρησης και ποια η λειτουργία της (σελ 155);
12. Ποιες εντολές εισόδου – εξόδου γνωρίζετε; Ποια η σύνταξή τους και ποια η λειτουργία τους (σελ 155-156);
13. Ποια η δομή ενός αλγόριθμου και ποια ενός προγράμματος;
14. Πότε χρησιμοποιείται η ακολουθιακή δομή και από ποιες εντολές αποτελείται;

Μετατρέψτε τις παρακάτω μαθηματικές εκφράσεις, σε μορφή αποδεκτή από τον ψευδοκώδικα και τις γλώσσες προγραμματισμού

1) $a+b$		
2) $a(b-1)$		
3) $(c-d)(a-b)$	5) $\frac{(a-1)}{4b}$	6) $\frac{a-1}{b-5}$
4) $\frac{c}{d}$	$\frac{3cd}{d+2}$	7) $\frac{a(5-d)}{cb}$

Σωστό – λάθος Α'

1. Ένας αλγόριθμος μπορεί να μην έχει είσοδο
2. Ένας αλγόριθμος μπορεί να μην έχει έξοδο
3. Η έννοια του αλγόριθμου αναφέρεται στα βήματα επίλυσης προβλημάτων πληροφορικής
4. Καθοριστικότητα στους αλγορίθμους σημαίνει ο αλγόριθμος να τερματίζει μετά από πεπερασμένο αριθμό βημάτων
5. Ο αλγόριθμος αποτελείται από πεπερασμένο σύνολο εντολών
6. Ο ποιο δομημένος τρόπος παρουσίασης ενός αλγορίθμου είναι με ελεύθερο κείμενο.
7. Η γλώσσα προγραμματισμού που χρησιμοποιούμε για την κωδικοποίηση ενός αλγορίθμου επηρεάζει την εκτέλεση του
8. Οι τιμές των μεταβλητών δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του αλγορίθμου
9. Στη δομή ακολουθίας υποχρεωτικά εκτελούνται όλες οι εντολές και μάλιστα η εκτέλεση είναι σειριακή
10. Η τιμή μιας μεταβλητής αλλάζει μόνο με εντολή εκχώρησης
11. Στο δεξί μέλος μιας εντολής εκχώρησης πρέπει να υπάρχει έκφραση
12. Οι παρενθέσεις μπορούν να μπουν σε μια έκφραση χωρίς να αλλοιώσουν το αποτέλεσμα της.
13. Οι λογικές τιμές μπορούν να πάρουν δύο διακριτές τιμές
14. Το διάγραμμα ροής αποτελεί διαγραμματικό τρόπο αναπαράστασης αλγορίθμων
15. Στα διαγράμματα ροής μια εντολή εκχώρησης εμφανίζεται σε πλάγιο παραλληλόγραμμο
16. Σε κάθε γεωμετρικό σχήμα ενός λογικού διαγράμματος αντιστοιχεί μια μόνο εντολή ή λειτουργία
17. Οι λέξεις Διάβασε και Δεδομένα είναι δεσμευμένες

18. Για να είναι σωστή η εκχώρηση $A \leftarrow$ "Αληθής" η μεταβλητή A πρέπει να είναι αλφαριθμητική
 19. Σ' ένα αλγόριθμο δεν χρειάζεται δήλωση μεταβλητών και σταθερών
 20. Σ' ένα διάγραμμα ροής ο κύκλος δηλώνει την αρχή και το τέλος του αλγόριθμου

Σωστό – λάθος Β'

Έστω οι δηλώσεις των μεταβλητών:

Μεταβλητές

Ακέραιος α, β, γ
Πραγματικός δ, ε, ζ
Χαρακτήρας η, θ

Εξετάστε τις παρακάτω εντολές και ξεχωρίστε τις σωστές από τις λανθασμένες βάση των παραπάνω δηλώσεων.

- | | | | |
|--------------------------|---|--------------------------|---------------------------------|
| <input type="checkbox"/> | $\alpha \leftarrow \beta * 3$ | <input type="checkbox"/> | $\theta \leftarrow \kappa$ |
| <input type="checkbox"/> | $\alpha \leftarrow \gamma - 2\beta$ | <input type="checkbox"/> | Διάβασε α, β, δ, η |
| <input type="checkbox"/> | $\alpha - 1 \leftarrow \beta + \gamma$ | <input type="checkbox"/> | Διάβασε (α-1), (β-γ) |
| <input type="checkbox"/> | $\beta \leftarrow * \delta$ | <input type="checkbox"/> | Διάβασε $\epsilon \leftarrow 2$ |
| <input type="checkbox"/> | $\alpha \leftarrow \alpha * \alpha + 2 * \alpha$ | <input type="checkbox"/> | Διάβασε η, 2, 15β |
| <input type="checkbox"/> | $\beta \leftarrow \alpha / (\gamma - 1)$ | <input type="checkbox"/> | Διάβασε θ, "καλημέρα" |
| <input type="checkbox"/> | $\epsilon \leftarrow \zeta + 3 * \alpha + 2 * \epsilon$ | <input type="checkbox"/> | Τύπωσε η, θ |
| <input type="checkbox"/> | $\delta \leftarrow 3 * \epsilon + \eta$ | <input type="checkbox"/> | Τύπωσε (α-1), (β-γ) |
| <input type="checkbox"/> | $\zeta \leftarrow 3 * \epsilon + \delta$ | <input type="checkbox"/> | Τύπωσε "η", "θ" |
| <input type="checkbox"/> | $\eta \leftarrow ' \kappa '$ | <input type="checkbox"/> | Τύπωσε α-β, "καλημέρα" |
| <input type="checkbox"/> | $\eta \leftarrow \theta$ | <input type="checkbox"/> | Τύπωσε $3 \leftarrow 2$ |

Ασκήσεις Αλγορίθμων

A) Τι θα εμφανίσουν οι παρακάτω αλγόριθμοι:

Αλγόριθμος A1

$\alpha \leftarrow 5$
 $\beta \leftarrow 10$
 $\alpha \leftarrow \alpha + \beta$
 Εμφάνισε "Νέα τιμή ", α
 Τέλος A1

Αλγόριθμος A2

$\alpha \leftarrow 5$
 Εμφάνισε α^2
 Τέλος A2

Αλγόριθμος A3

$\alpha \leftarrow 100$
 $\beta \leftarrow 200$
 Εμφάνισε " $\alpha * \beta =$ ", $\alpha * \beta$
 Τέλος A3

Αλγόριθμος A4

$\alpha \leftarrow 40$
 $\beta \leftarrow 20$
 $\alpha \leftarrow \alpha + \beta$
 $\beta \leftarrow \alpha + \beta$
 Εμφάνισε β, α
 Τέλος A4

Αλγόριθμος A5

$\alpha \leftarrow 4$
 $\beta \leftarrow (\alpha^2 \text{ div } 2) \text{ mod } 2$
 Εμφάνισε β
 Τέλος A5

Αλγόριθμος A6

$\alpha \leftarrow 2$
 $\beta \leftarrow \alpha + 2^3 - 10 + \alpha / 4$
 $\alpha \leftarrow -\alpha + \alpha^2$
 Εμφάνισε α, β
 Τέλος A6

Β) Τι θα εμφανίσουν οι παρακάτω αλγόριθμοι αν δοθεί σαν είσοδος το 8 και το 10

Αλγόριθμος A1	Αλγόριθμος A2	Αλγόριθμος A3
Διάβασε α	Διάβασε α	Διάβασε α
Διάβασε β	Διάβασε β	Διάβασε β
$\gamma \leftarrow \alpha * 5$	Εμφάνισε $2 * \alpha, "*" , \beta$	Εμφάνισε $\alpha, "*" , \beta, "=" , \alpha * \beta$
$\delta \leftarrow \gamma * 2$	Τέλος A2	Τέλος A3
$\epsilon \leftarrow \delta \bmod 2$		
Εμφάνισε α,β,γ,δ,ε		
Τέλος A1		

Γ) Τι θα εμφανίσουν οι ακόλουθοι αλγόριθμοι αν δοθούν σαν είσοδος οι τιμές 5, 2 και 3;

Αλγόριθμος A1	Αλγόριθμος A2	Αλγόριθμος A3
Διάβασε α	Διάβασε α	Διάβασε α
Διάβασε β	Διάβασε α	Διάβασε β
Διάβασε α	Διάβασε β	Διάβασε α
Εμφάνισε α,β	Εμφάνισε α,β	Εμφάνισε α
Τέλος A1	Τέλος A2	Τέλος A3

Δ) Ποια τα λάθη των ακόλουθων αλγόριθμων

Αλγόριθμος A1	Αλγόριθμος A2	Αλγόριθμος A3
Διάβασε α1	Διάβασε α1	Διάβασε α1
Διάβασε α2	$\alpha 1 \leftarrow \alpha 2 + \alpha 1$	Διάβασε α1
Διάβασε α	Εμφάνισε α1,α2	Εμφάνισε α1,α2
Εμφάνισε α1,α11	Τέλος A2	Τέλος A3
Τέλος A1		

Ασκήσεις Ακολουθιακής Δομής

1. Να γίνει αλγόριθμος, σε ψευδοκώδικα και διάγραμμα ροής, ο οποίος θα υπολογίζει το εμβαδόν ορθογωνίου με βάση B και ύψος Y .
2. Να γίνει αλγόριθμος, σε ψευδοκώδικα και διάγραμμα ροής, ο οποίος θα υπολογίζει το μέσο όρο τεσσάρων αριθμών, που θα δίνονται από το χρήστη.
3. Να γίνει αλγόριθμος, σε ψευδοκώδικα και διάγραμμα ροής, ο οποίος θα βρίσκει την τιμή της συνάρτησης $F(x)=6x^2+5x+4$, για μια τιμή που θα δίνει ο χρήστης.
4. Να γίνει αλγόριθμος, σε ψευδοκώδικα και διάγραμμα ροής, ο οποίος θα λύνει την εξίσωση $3x+6xy=15$, για μια δεδομένη τιμή του y .
5. Να γίνει αλγόριθμος, σε ψευδοκώδικα και διάγραμμα ροής, ο οποίος θα δέχεται (θα διαβάζει) ένα μήνυμα και θα το αναπαράγει στην έξοδο (θα το τυπώνει).
6. Σε ένα σχολείο οι ώρες διδασκαλίας μπορεί να διαφέρουν από μέρα σε μέρα. Να γίνει αλγόριθμος, ο οποίος θα διαβάζει τις ώρες διδασκαλίας για κάθε μια από τις πέντε εργάσιμες μέρες της εβδομάδας και να υπολογίζει των μέσο όρο των ωρών διδασκαλίας.
7. Ένας μανάβης αγοράζει τα μήλα με $x\text{€}$ το κιλό και τα πουλά για $y\text{€}$ το κιλό. Να γίνει αλγόριθμος, ο οποίος θα διαβάζει την τιμή αγοράς και την τιμή πώλησης των μήλων καθώς και τον αριθμό των κιλών που πωλήθηκαν. Ο αλγόριθμός θα επιστρέφει το κέρδος του μανάβη.
8. Σε μια εταιρεία, οι μηνιαίες αποδοχές ενός υπαλλήλου καθορίζονται από το βασικό μισθό και το επίδομα του. Ο βασικός μισθός ισούται με 400 Ευρώ προσαυξημένος κατά 20 Ευρώ επί τον αριθμό των ετών εργασίας του υπαλλήλου. Όλοι οι υπάλληλοι παίρνουν βασικό επίδομα 30 Ευρώ και για καθένα από τα παιδιά τους παίρνουν επιπλέον 15 Ευρώ. Να γραφεί αλγόριθμος που να υπολογίζει πόσα χρήματα θα πάρει τελικά ένα υπάλληλος σε ένα μήνα.
9. Μας δίνεται η επίδοση ενός δρομέα μεγάλων αποστάσεων σε ώρες, λεπτά και δευτερόλεπτα. Να γίνει αλγόριθμος που θα διαβάζει τις τρεις αυτές τιμές καθώς και το ονοματεπώνυμο του αθλητή. Ο αλγόριθμος θα εμφανίζει το ονοματεπώνυμο του αθλητή και δίπλα την επίδοσή του μεταφρασμένη σε δευτερόλεπτα.
10. Φτιάξτε έναν αλγόριθμο ο οποίος θα διαβάζει σε χιλιόμετρα, μέτρα και εκατοστά μια απόσταση και θα επιστρέφει την ισοδύναμη μόνο σε εκατοστά.

11. Φτιάξτε έναν αλγόριθμο ο οποίος θα διαβάσει έναν ακέραιο αριθμό και θα επιστρέφει τους τρεις αμέσως μεγαλύτερους ακεραίους. Αν για παράδειγμα δοθεί ο αριθμός 26, ο αλγόριθμος θα επιστρέφει τους αριθμούς 27, 28 και 29.
12. Να γίνει αλγόριθμος ο οποίος θα διαβάζει δυο αριθμούς α και β , θα τους αντιμεταθέτει και θα εκτυπώνει τις τιμές τους. Για παράδειγμα αν οι αριθμοί που θα διαβαστούν είναι αρχικά 5 και 8, για το α και β αντίστοιχα, μετά την εκτέλεση του αλγορίθμου οι τιμές που θα εκτυπωθούν θα πρέπει είναι 8 και 5.
13. Να γίνει αλγόριθμος ο οποίος θα διαβάζει τρεις αριθμούς α , β και γ , θα αυξάνει τον α κατά 1, θα αυξάνει τον β κατά 5, θα αυξάνει τον γ κατά $(\alpha+\beta)$ και θα εκτυπώνει τους τρεις αριθμούς
14. Ένα φωτοτυπείο χρεώνει για τα ασπρόμαυρα αντίτυπα 4 λεπτά και για τα έγχρωμα 20 λεπτά. Έστω ότι κάποιος επιθυμεί να φωτοτυπήσει ένα βιβλίο που περιέχει ασπρόμαυρες και έγχρωμες σελίδες. Να γίνει πρόγραμμα το οποίο θα διαβάζει τα δεδομένα και θα υπολογίζει πόσο τελικά θα πληρώσει σε Ευρώ.
15. Σε μια παρτίδα υπολογιστών που έλαβε ένας έμπορος, κάποιοι αποδείχθηκαν ελαττωματικοί. Να γίνει αλγόριθμος, ο οποίος θα διαβάζει πόσους υπολογιστές είχε η παρτίδα και πόσοι βγήκαν ελαττωματικοί. Ο αλγόριθμος θα επιστρέφει σαν αποτέλεσμα το ποσοστό των ελαττωματικών υπολογιστών.
16. Να αναπτυχθεί αλγόριθμος που να υπολογίζει το ποσοστό % των κοριτσιών και το ποσοστό % των αγοριών ενός τμήματος, όταν δίνεται ο αριθμός τους (π.χ. 10 κορίτσια και 10 αγόρια).
17. Ένας Η/Υ κοστίζει, χωρίς ΦΠΑ, $x\text{€}$. Να γίνει αλγόριθμος, ο οποίος θα διαβάζει την τιμή του Η/Υ και θα υπολογίζει πόσο θα είναι ο φόρος προστιθέμενης αξίας(ΦΠΑ) σε € (Θεωρούμε ότι ο ΦΠΑ είναι 19%).
18. Ένας Η/Υ κόστισε, μαζί με το ΦΠΑ, $x\text{€}$. Να γίνει αλγόριθμος, ο οποίος θα διαβάζει την τελική τιμή του Η/Υ και θα υπολογίζει πόσο ήταν ο φόρος προστιθέμενης αξίας(ΦΠΑ) σε € (Θεωρούμε ότι ο ΦΠΑ είναι 19%).
19. Ένας εργαζόμενος πήρε τα Χριστούγεννα αυξημένο μισθό, ίσο με $x\text{€}$. Να γίνει αλγόριθμος, ο οποίος θα διαβάζει το τελικό μισθό του εργαζομένου και θα υπολογίζει πόσα από αυτά τα χρήματα είναι το δώρο Χριστουγέννων (θεωρούμε ότι το δώρο είναι το 50% του βασικού μισθού).
20. Ένα μπλουζάκι έχει πριν τις εκπτώσεις $x\text{€}$ χωρίς ΦΠΑ. Στις εκπτώσεις το κατάστημα ρίχνει τις τιμές όλων των προϊόντων του κατά 20%. Να γίνει αλγόριθμος, ο οποίος θα

διαβάξει την αρχική τιμή της μπλούζας και θα υπολογίζει την τελική της τιμή κατά τη διάρκεια των εκπτώσεων με το ΦΠΑ (ΦΠΑ = 19%).

21. Ένας γονέας αποφάσισε να μοιράσει ένα ποσό στα παιδιά του σύμφωνα με την εξής λογική: το μεγαλύτερο παιδί θα πάρει το 40%, το δεύτερο παιδί το 35% και το τρίτο παιδί 25%. Να γραφεί αλγόριθμος που θα διαβάξει το ποσό και θα εμφανίζει τι τελικά θα πάρει κάθε παιδί.
22. Ένας ωρομίσθιος καθηγητής πληρώνεται με $x \text{ €}$ την κάθε ώρα διδασκαλίας. Το ΙΚΑ παίρνει το 10% από την συνολική μηνιαία αμοιβή του. Από τα υπόλοιπα χρήματα η εφορία παίρνει το 5%. Να γίνει αλγόριθμος, ο οποίος θα διαβάξει πόσες ώρες δίδαξε στην διάρκεια ενός μήνα και να επιστρέφει τα χρήματα που πήρε η εφορία.
23. Ένας καθηγητής επιθυμεί να φτιάξει αλγόριθμο ο οποίος θα διαβάξει το επώνυμο ενός μαθητή και τρεις βαθμούς. Ο πρώτος βαθμός είναι ο προφορικός βαθμός του μαθητή, ο δεύτερος είναι ο βαθμός του ωριαίου διαγωνίσματος και ο τρίτος βαθμός είναι ο βαθμός ενός τέστ. Να υπολογιστεί ο τελικός βαθμός του μαθητή αν ο προφορικός βαθμός μετρά 45%, ο βαθμός του διαγωνίσματος 40% και ο βαθμός του τέστ 15%. Στο τέλος να εμφανίζεται το μήνυμα: Ο βαθμός του/της ΕΠΙΘΕΤΟ είναι ΒΑΘΜΟΣ (π.χ. Ο βαθμός του/της Γιαννιώτη είναι 17).
24. Έστω ότι έχουμε να μοιράσουμε σε N μαθητές A τετράδια. Ο αριθμός των τετραδίων μπορεί να μη διαιρείται ακριβώς από το N . Να γίνει αλγόριθμος που θα διαβάξει τα A και N και θα εκτυπώνει το πλήθος των τετραδίων που θα πάρει ο κάθε μαθητής καθώς και το πόσα τετράδια θα περισσέψουν.
25. Δίνεται ένας ακέραιος αριθμός. Να γίνει αλγόριθμος ο οποίος θα βρίσκει και θα εκτυπώνει το τελευταίο ψηφίο (μικρότερης βαρύτητας) του αριθμού.
26. Να αναπτυχθεί πρόγραμμα που να διαβάξει έναν τριψήφιο αριθμό και να τυπώνει τα ψηφία του σε αντίστροφη σειρά.
27. Να γίνει αλγόριθμος ο οποίος θα δέχεται σαν είσοδο μια απόσταση σε cm και θα την επιστρέφει σε km , m και cm . Για παράδειγμα η απόσταση 152.346cm ισοδυναμεί με 1 km , 523 m και 46cm . Να τυπώνεται ανάλογο μήνυμα.
28. Θέλουμε να δώσουμε σε ένα φίλο μας ένα ποσό σε € , που δεν έχει υποδιαίρεση του 1€ . Να γίνει αλγόριθμος που θα διαβάξει το ποσό, θα υπολογίζει και θα εκτυπώνει τον ελάχιστο αριθμό χαρτονομισμάτων και νομισμάτων που μπορούμε να χρησιμοποιήσουμε. Θεωρούμε ότι μπορούμε να χρησιμοποιήσουμε χαρτονομίσματα και νομίσματα των 20€ , 10€ , 5€ , 2€ και 1€ .

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βλάχος Στέφανος

Σημειώσεις - Ερωτήσεις - Ασκήσεις
Κεφαλαίων 2, 8
(Δομή Επιλογής)

ΚΕΦΑΛΑΙΟ 2-8 (Δομή Επιλογής-Δομή Επανάληψης)

Λογικές Εκφράσεις (σελ 165-166)

Οι λογικές εκφράσεις μπορούν να περιλαμβάνουν

- λογικές σταθερές (αληθής – ψευδής)
- λογικές μεταβλητές
- αριθμητικές τιμές
- αριθμητικές μεταβλητές
- μεταβλητές χαρακτήρα
- σταθερές χαρακτήρα
- τελεστές σύγκρισης (>, <, >=, <=, <>, =)
- λογικές πράξεις (ή, και, όχι)

Οι αριθμητικές εκφράσεις έχουν σαν αποτέλεσμα αριθμητικές τιμές. Ομοίως και οι **λογικές εκφράσεις** έχουν σαν αποτέλεσμα **λογικές τιμές** (αληθής-ψευδής). Αν, για παράδειγμα, έχουμε την έκφραση ($5 > 2$), τότε αυτή μετά τον υπολογισμό (έλεγχο) της, θα πάρει την τιμή **αληθής**. Αν θέλαμε να ελέγξουμε αν μια αριθμητική μεταβλητή x , είναι μεγαλύτερη ή ίση του τέσσερα θα είχαμε την λογική έκφραση ($x >= 4$). Η τιμή αυτής της έκφρασης εξαρτάται από την τιμή του x . Αν το x είναι μικρότερο του 4, η έκφραση θα πάρει την τιμή **αληθής**, ενώ στην αντίθετη περίπτωση θα πάρει την τιμή **ψευδής**. Παρακάτω παρουσιάζονται κάποιες **απλές λογικές εκφράσεις** με τη χρήση **μόνο τελεστών σύγκρισης**.

(αληθής = ψευδής)	είναι ψευδής
(αληθής <> ψευδής)	είναι αληθής
($2 <= 2$)	είναι αληθής
($7 <> 7$)	είναι ψευδής

Για τη σύγκριση χαρακτήρων χρησιμοποιούμε την αλφαβητική σειρά των χαρακτήρων. Έτσι για παράδειγμα ο χαρακτήρας 'β' θεωρείται μικρότερος του 'δ'. Έτσι η έκφραση 'β' < 'δ' είναι αληθής. Στις κανονικές γλώσσες προγραμματισμού τα κεφαλαία θεωρούνται μικρότερα από τα αντίστοιχα μικρά γράμματα. Η σειρά τους είναι 'Α', 'Β', 'Γ', ..., 'α', 'β', 'γ' κ.λ.π. Έτσι οι εκφράσεις 'Α' < 'α', 'Δ' < 'δ' και 'Ω' < 'α' είναι όλες αληθείς.

Μπορούμε ακόμα να συγκρίνουμε αλφαριθμητικά. Αν το πρώτο γράμμα ενός αλφαριθμητικού είναι μικρότερο από το πρώτο γράμμα ενός άλλου τότε είναι μικρότερο. Η έκφραση ('αύριο' < 'βράδυ') είναι αληθής γιατί το 'α' (πρώτο γράμμα στη λέξη 'αύριο') είναι μικρότερο του 'β' (πρώτο γράμμα στη λέξη 'βράδυ'). Αν τα πρώτα γράμματα είναι ίδια τα παρακάμπτουμε και συγκρίνουμε τα επόμενα, μέχρι να βρεθούν δυο διαφορετικά και να βγάλουμε συμπέρασμα για τη σύγκριση ή μέχρις ότου η μια λέξη τελειώσει, οπότε είναι και μικρότερη. Έτσι προκύπτουν τα παρακάτω αποτελέσματα.

('β' > 'γ')	είναι ψευδής
('βροχή' = 'βροχή')	είναι αληθής
('βρέχω' < 'βροχή')	είναι αληθής (τα πρώτα διαφορετικά γράμματα είναι το 'ε' και το 'ο', από τα οποία το 'ε' είναι μικρότερο)
('έξω' > 'έξωση')	είναι ψευδής (όλα τα γράμματα της λέξης 'έξω' εμφανίζονται και στην λέξη 'έξωση', αλλά η λέξη 'έξω' τελείωσε, άρα είναι μικρότερη)

Οι **λογικές πράξεις** (ή, και, όχι) μπορούν να εφαρμοστούν μεταξύ λογικών εκφράσεων για τη δημιουργία πιο σύνθετων λογικών εκφράσεων.

Η λογική **διάζευξη** (ή), εφαρμόζεται μεταξύ δύο λογικών εκφράσεων και δίνει αποτέλεσμα 'αληθής' όταν τουλάχιστον ένα από τα ορίσματα είναι αληθές.

Η λογική **σύζευξη (και)**, εφαρμόζεται μεταξύ δύο λογικών εκφράσεων και δίνει αποτέλεσμα 'αληθής' όταν όλα τα ορίσματα είναι αληθή. Αν ένα τουλάχιστον είναι 'ψευδής', επιστρέφει την τιμή 'ψευδής'.

Η λογική **άρνηση (όχι)**, εφαρμόζεται σε μια λογική έκφραση και επιστρέφει το αντίθετο αποτέλεσμα από την είσοδο.

Ο παρακάτω πίνακας μας δείχνει ποια είναι τα αποτελέσματα των πράξεων σε όλους τους δυνατούς συνδυασμούς λογικών τιμών. Οι μεταβλητές X και Y θεωρούνται λογικές μεταβλητές. Προσοχή στην πράξη '**όχι**' η οποία παίρνει ένα μόνο όρισμα.

X	Y	Όχι(X)	X ή Y	X και Y
Ψευδής	Ψευδής	Αληθής	Ψευδής	Ψευδής
Ψευδής	Αληθής	Αληθής	Αληθής	Ψευδής
Αληθής	Ψευδής	Ψευδής	Αληθής	Ψευδής
Αληθής	Αληθής	Ψευδής	Αληθής	Αληθής

Οι λογικές πράξεις εφαρμόζονται όταν πρέπει να συνδυάσουμε δυο 'γεγονότα'. Ακολουθεί η περιγραφή κάποιων γεγονότων σε φυσική γλώσσα και η μεταφορά τους με τη μορφή λογικής έκφρασης.

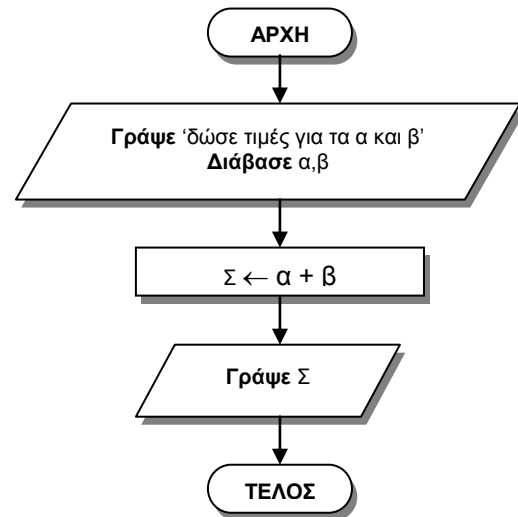
- ❑ Να ελέγξουμε εάν ένας αριθμός x είναι μεγαλύτερος από το 0 και μικρότερος του 10 $(x > 0)$ και $(x < 10)$
- ❑ Να ελέγξουμε αν ένας αριθμός x είναι μεγαλύτερος-ίσος του 5 ή μικρότερος-ίσος του -5 $(x \leq -5)$ ή $(x \geq 5)$
- ❑ Να ελέγξουμε αν δύο αριθμοί x και ψ είναι θετικοί (και οι δύο) $(x > 0)$ και $(\psi > 0)$
- ❑ Να ελέγξουμε αν οι αριθμοί x και ψ είναι και οι δυο μεγαλύτεροι από το δέκα και αν το άθροισμά τους είναι μεγαλύτερο από 30 (να συμβαίνουν όλα ταυτόχρονα) $(x > 10)$ και $(\psi > 10)$ και $((x + \psi) > 30)$
- ❑ Να ελέγξουμε αν, ο αριθμός x είναι μικρότερος του -2 ή μεγαλύτερος του 2 και ο αριθμός ψ είναι ίσος με 0 $((x < -2)$ ή $(x > 2))$ και $(\psi = 0)$

Προσοχή: Οι παρενθέσεις στις λογικές εκφράσεις έχουν την ίδια σπουδαιότητα με τις αριθμητικές εκφράσεις. Μπορούμε να αλλάξουμε τελείως την σημασία μιας λογικής έκφρασης αλλάζοντας τις παρενθέσεις. Στις λογικές παραστάσεις εκτελούνται πρώτα οι πράξεις 'όχι' μετά οι πράξεις 'και' και τέλος οι πράξεις 'ή', εφ' όσον δεν υπάρχουν παρενθέσεις.

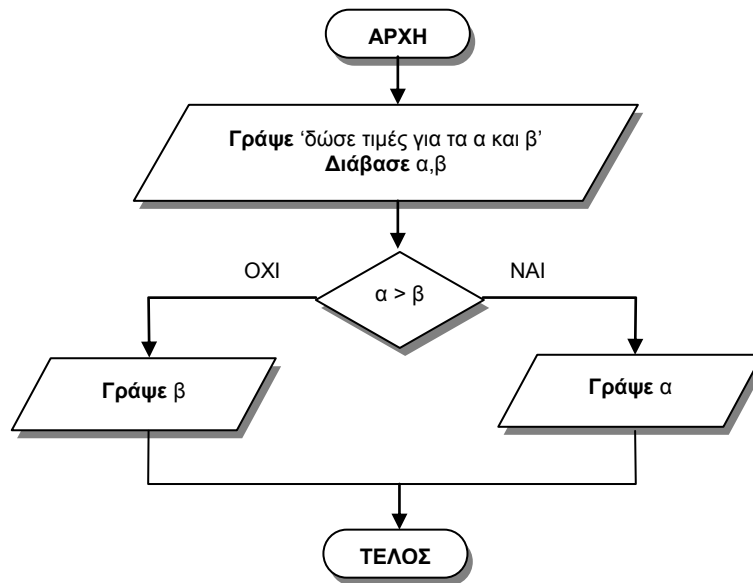
Απλή και Σύνθετη Επιλογή (σελ 32-35, 166-168)

Στις ασκήσεις (αλγορίθμους) ακολουθιακής δομής η διαδικασία επίλυσης αποτελείται από μια σειρά βημάτων, τα οποία σίγουρα θα εκτελεστούν. Δεν έχουμε την δυνατότητα (αλλά και την ανάγκη) να εκτελέσουμε διαφορετικές εντολές ανάλογα με τα δεδομένα. Για παράδειγμα στον αλγόριθμο εύρεσης του αθροίσματος δυο αριθμών α και β , είναι σίγουρο ότι δεν χρειάζεται να πάρουμε υποπεριπτώσεις και δεν χρειάζεται να αλλάξουμε τον τρόπο που θα αντιμετωπίσουμε το πρόβλημα ανάλογα με τις τιμές των α και β . Στο διάγραμμα ροής των ακολουθιακών αλγορίθμων είναι χαρακτηριστικό ότι από την έλλειψη '**αρχή**' έως την έλλειψη '**τέλος**' θα υπάρχει μόνο μια διαδρομή.

Πρόγραμμα άθροισμα
Μεταβλητές
 Ακέραιες α,β,Σ
Αρχή
 Γράψε 'δώσε τιμές για τα α και β'
 Διάβασε α,β
 $\Sigma \leftarrow \alpha + \beta$
 Γράψε Σ
Τέλος_προγράμματος



Υπάρχουν όμως περιπτώσεις ασκήσεων όπου δεν πρέπει να εκτελέσουμε τις ίδιες εντολές για κάθε τιμή δεδομένου που παίρνουμε. Αν για παράδειγμα μας ζητήσουν **να βρούμε τον μεγαλύτερο από δυο αριθμούς α και β**, δεν θα είμαστε σε θέση να πούμε, κατά τη σχεδίαση του αλγορίθμου, ποιος θα είναι μεγαλύτερος. Για την επίλυση μιας τέτοιας άσκησης θα είχαμε το παρακάτω διάγραμμα ροής.



Η δομή που έχει χρησιμοποιηθεί για την επίλυση αυτής της άσκησης ονομάζεται δομή επιλογής. Η δομή αυτή μας επιτρέπει να εκτελούμε διαφορετικές εντολές ανάλογα με τα δεδομένα του προγράμματος, όταν αυτό είναι απαραίτητο. Στο διάγραμμα ροής, σε αντίθεση με τα προηγούμενα διαγράμματα των ακολουθιακών ασκήσεων, έχουμε τη δυνατότητα να πάμε από την αρχή του αλγορίθμου στο τέλος του ακολουθώντας δυο διαφορετικές διαδρομές. Κάθε φορά που ο αλγόριθμος εκτελείται ακολουθούμε το ένα από τα δύο μονοπάτια και το άλλο μονοπάτι δεν εκτελείται καθόλου. Όταν γράφουμε έναν αλγόριθμο σε ψευδοκώδικα ή στη γλώσσα προγραμματισμού ΓΛΩΣΣΑ, χρησιμοποιούμε την εντολή **'Αν – Τότε - Αλλιώς - Τέλος Αν'** ή την εντολή **'Αν – Τότε – Τέλος Αν'**.

Η εντολή **'Αν – τότε – αλλιώς – τέλος αν'** ονομάζεται **σύνθετη επιλογή** και χρησιμοποιείται όταν έχουμε δυο ομάδες εντολών εκ των οποίων μόνο η μια θα πρέπει να εκτελεστεί, ανάλογα με το αποτέλεσμα μιας συνθήκης. Η σύνταξη της εντολής καθώς και ο προηγούμενος αλγόριθμος σε ΓΛΩΣΣΑ φαίνονται παρακάτω.

Αν <συνθήκη> τότε

<Ομάδα εντολών 1>

Αλλιώς

<Ομάδα εντολών 2>

Τέλος_Αν

Η συνθήκη είναι μια λογική έκφραση

**Πρόγραμμα μεγαλύτερος
Μεταβλητές**

Ακέραιες α,β

Αρχή

Γράψε 'δώσε τιμές για τα α και β'

Διάβασε α,β

Αν (α>β) τότε

Γράψε α

Αλλιώς

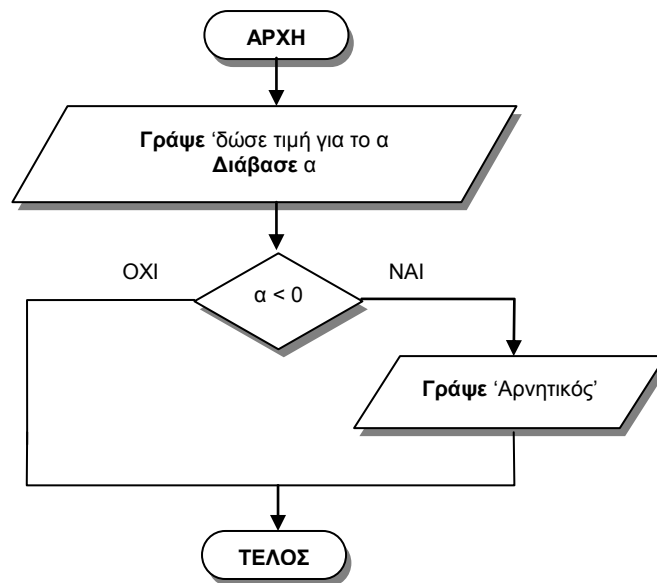
Γράψε β

Τέλος_Αν

Τέλος_προγράμματος

Κατά την εκτέλεση μιας εντολής επιλογής η λογική έκφραση της εντολής υπολογίζεται (παίρνει τιμή **αληθής** ή **ψευδής**). Αν η συνθήκη είναι **αληθής**, τότε εκτελείται η ομάδα εντολών 1, αλλιώς αν η συνθήκη είναι **ψευδής** εκτελείται η ομάδα εντολών 2

Υπάρχουν περιπτώσεις όπου η ομάδα εντολών 2 είναι κενή. Αυτό σημαίνει ότι θέλουμε να εκτελέσουμε κάποιες εντολές αν ισχύει μια συνθήκη, ειδικά δεν θέλουμε να εκτελεστεί τίποτα. Αν για παράδειγμα μας ζητήσουν **να φτιάξουμε πρόγραμμα το οποίο θα διαβάζει έναν αριθμό και θα τυπώνει το μήνυμα 'Αρνητικός'**, όταν ο αριθμός είναι **αρνητικός**, θα φτιάξουμε ένα διάγραμμα ροής όπως το παρακάτω.



Αν μεταφέρουμε τον αλγόριθμο αυτό σε πρόγραμμα, τότε θα χρησιμοποιήσουμε την άλλη μορφή την εντολής επιλογής, την **'Αν – τότε – Τέλος Αν'**, η οποία ονομάζεται **απλή επιλογή**. Σε αυτήν την εντολή έχουμε μια μόνο ομάδα εντολών, η οποία είτε θα εκτελεστεί, είτε όχι, ανάλογα με την τιμή της λογικής συνθήκης. Επειδή στην περίπτωση που η συνθήκη δεν ισχύει (**ψευδής**), δεν θέλουμε να εκτελεστεί καμία εντολή, δεν χρησιμοποιούμε καθόλου την πρόταση 'Αλλιώς'.

Αν <συνθήκη> τότε
<Ομάδα εντολών 1>
Τέλος_Αν

Ή

Αν <συνθήκη> τότε εντολή

Η συνθήκη είναι μια λογική έκφραση

Πρόγραμμα αρνητικός
Μεταβλητές

Ακέραιες α

Αρχή

Γράψε 'δώσε τιμή για το α'

Διάβασε α

Αν (α<0) τότε

Γράψε 'Αρνητικός'

Τέλος_Αν

Τέλος_προγράμματος

Εμφωλευμένη Επιλογή (σελ 37-39, 169-171)

Στις ομάδες εντολών που τοποθετούμε μέσα σε μια εντολή 'Αν', μπορούμε να βάλουμε οποιαδήποτε από τις γνωστές εντολές (Διάβασε, Γράψε, εκχώρησε (\leftarrow)), καθώς και την ίδια την εντολή 'Αν' ή τις εντολές πολλαπλής επιλογής και επανάληψης, που θα δούμε αργότερα. Μπορούμε να τοποθετήσουμε μια ή και περισσότερες εντολές επιλογής μέσα σε μια επιλογή για να φτιάξουμε πιο πολύπλοκες δομές. Μια τέτοια δομή ονομάζεται 'εμφωλευμένη επιλογή' ή 'εμφωλευμένη Αν'. Ακολουθεί η σύνταξη των εμφωλευμένων επιλογών.

Αν <συνθήκη> τότε
<εντολές>
Αν <Συνθήκη> τότε
<εντολές>

Αλλιώς
<εντολές>

Τέλος_αν
<εντολές>

Αλλιώς
<εντολές>
Αν <Συνθήκη> τότε
<εντολές>

Αλλιώς
<εντολές>

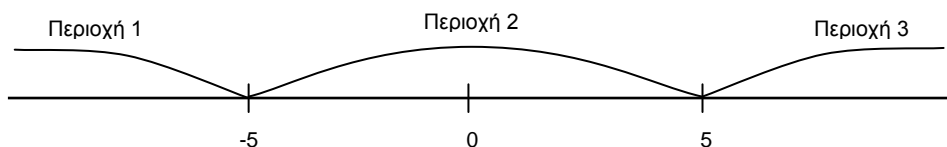
Τέλος_αν
<εντολές>

Τέλος_Αν

Ο συνδυασμός των εμφωλεύσεων γίνεται κατά βούληση και σε όσα επίπεδα θέλουμε. Η τοποθέτηση των εσωτερικότερων εντολών προς τα δεξιά βοηθά στο να αντιλαμβανόμαστε που αρχίζουν και που τελειώνουν οι διάφορες εντολές 'Αν'.

Ας θεωρήσουμε την παρακάτω άσκηση:

Να γίνει πρόγραμμα το οποίο θα διαβάζει έναν αριθμό και θα μας τυπώνει το κατάλληλο μήνυμα ανάλογα με την περιοχή του παρακάτω σχήματος, που βρίσκεται ο αριθμός ('μικρότερος του -5', 'μεταξύ -5 και 5', 'μεγαλύτερος το 5').



Αν αναλύσουμε το πρόβλημα θα δούμε ότι ο αριθμός, έστω Α, μπορεί να βρίσκεται σε ένα από τα τρία διαστήματα, οπότε έχουμε τις παρακάτω περιπτώσεις

- 1) το Α να είναι μικρότερο του -5 ($A < -5$).
- 2) το Α να είναι μεταξύ -5 και 5 ($(A \geq -5)$ και $(A \leq 5)$).

3) το A να είναι μεγαλύτερο του 5 ($A > 5$).

Χρησιμοποιώντας την **απλή εντολή 'Αν'** ('Αν-τότε-τέλος_αν'), μπορούμε να λύσουμε το πρόβλημα χρησιμοποιώντας τρεις ξεχωριστές 'Αν', ως εξής:

Πρόγραμμα Διαστήματα

Μεταβλητές

Ακέραιες A

Αρχή

Γράψε 'δώσε τιμή για το A'

Διάβασε A

Αν ($A < -5$) **τότε**

Γράψε 'Μικρότερος του -5'

Τέλος_Αν

Αν ($(A \geq -5)$ και $(A \leq 5)$) **τότε**

Γράψε 'Μεταξύ του 5 και του -5'

Τέλος_Αν

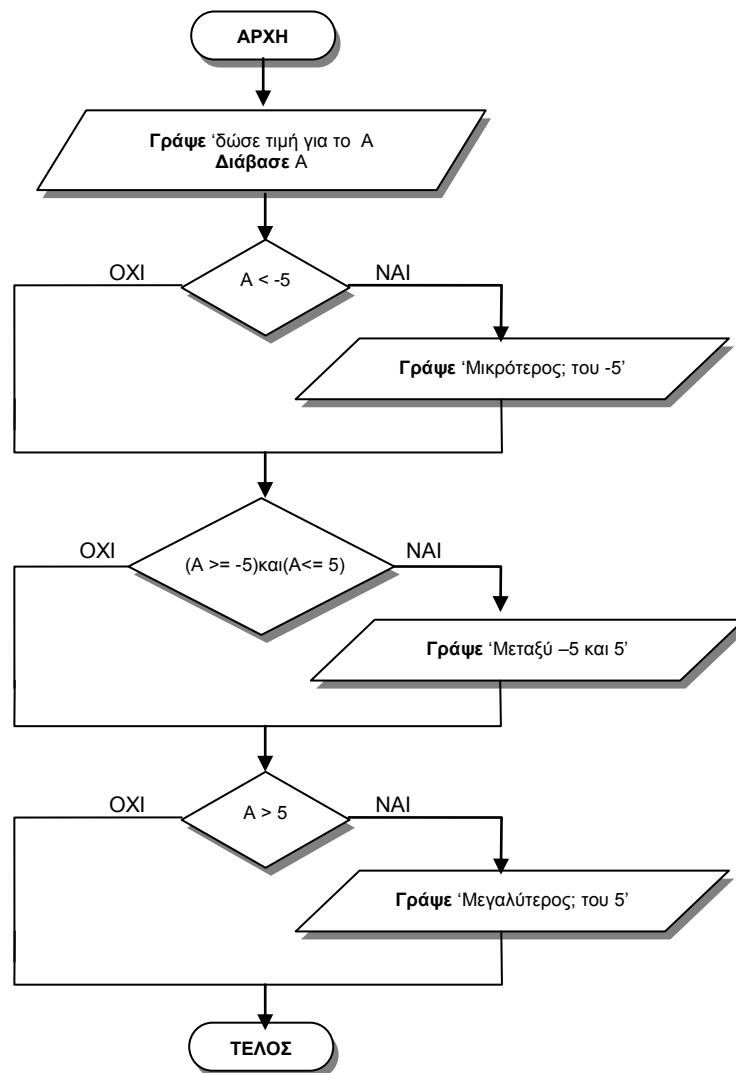
Αν ($A > 5$) **τότε**

Γράψε 'Μεγαλύτερος του 5'

Τέλος_Αν

Τέλος_προγράμματος

Το διάγραμμα ροής του παραπάνω προγράμματος παρουσιάζεται στο παρακάτω σχήμα.



Η ίδια άσκηση θα μπορούσε να αντιμετωπιστεί με **εμφωλευμένες επιλογές** ως εξής:

Πρόγραμμα Διαστήματα2

Μεταβλητές

Ακέραιες A

Αρχή

Γράψε 'δώσε τιμή για το A'

Διάβασε A

Αν ($A < -5$) τότε

Γράψε 'Μικρότερος του -5'

Αλλιώς

Αν ($A \leq 5$) τότε

Γράψε 'Μεταξύ του 5 και του -5'

Αλλιώς

Γράψε 'Μεγαλύτερος του 5'

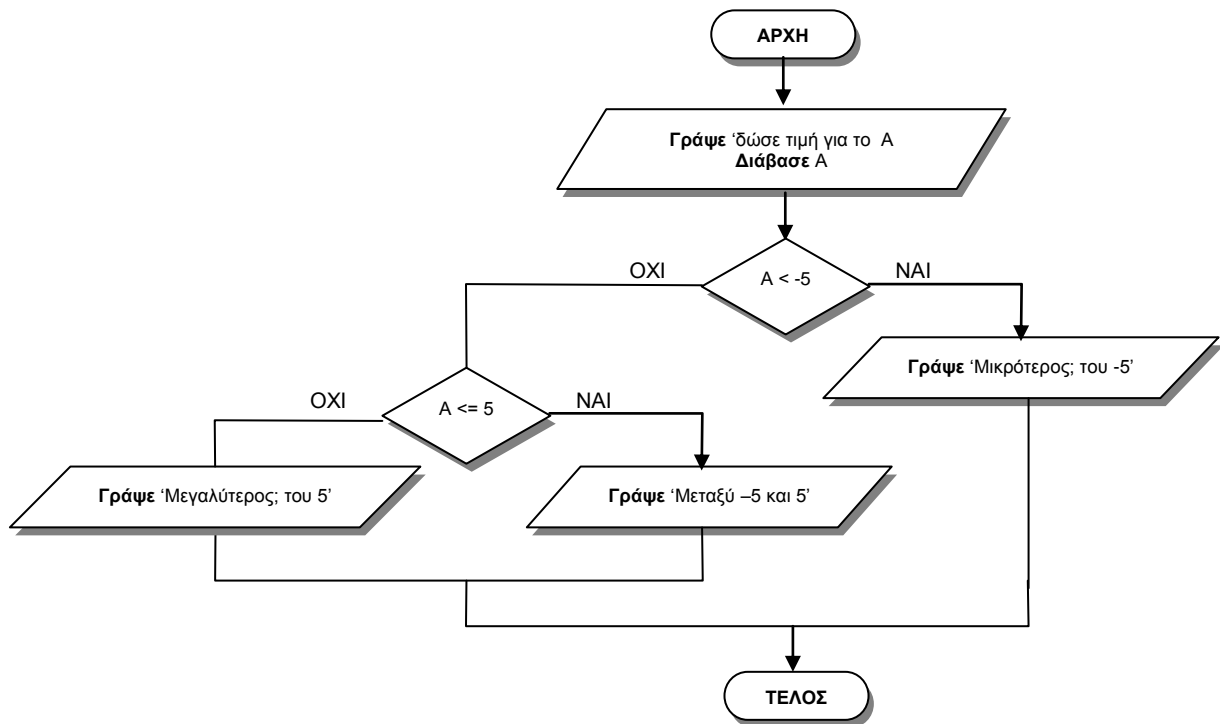
Τέλος_Αν

Τέλος_Αν

Τέλος_προγράμματος

Εσωτερική 'Αν'

Εξωτερική 'Αν'



Ενδιαφέρον παρουσιάζει το γεγονός ότι οι λογικές συνθήκες που χρησιμοποιούμε δεν είναι πλέον οι ίδιες.

Πολλαπλές Επιλογές (σελ 35-37, 171-173)

Οι δομές πολλαπλής επιλογής χρησιμοποιούνται όταν θέλουμε για διάφορες τιμές μιας μεταβλητής να εκτελέσουμε διαφορετικές εντολές. Οι δομές επιλογής που συντάσσονται με τις εντολές πολλαπλής επιλογής μπορούν να συνταχθούν και με τις εμφωλευμένες επιλογές. Επειδή όμως προκύπτουν πολύπλοκες δομές καταφεύγουμε στις πολλαπλές επιλογές.

Οι εντολές πολλαπλής επιλογής είναι η '**Αν-τότε-Αλλιώς_αν-Αλλιώς-Τέλος_Αν**' και η '**Επίλεξε**'. Οι δύο εντολές συντάσσονται ως εξής:

Αν <συνθήκη> **τότε**
 <Ομάδα εντολών 1>
Αλλιώς_Αν <συνθήκη> **τότε**
 <Ομάδα εντολών 2>
Αλλιώς_Αν <συνθήκη> **τότε**
 <Ομάδα εντολών 3>
Αλλιώς_Αν <συνθήκη> **τότε**
 <Ομάδα εντολών 4>
 .
 .
 .
Αλλιώς
 <Ομάδα εντολών ν>
Τέλος_Αν

Επίλεξε <έκφραση>
Περίπτωση <τιμές>
 <Ομάδα εντολών 1>
Περίπτωση <τιμές>
 <Ομάδα εντολών 2>
Περίπτωση <τιμές>
 <Ομάδα εντολών 3>
 .
 .
 .
Περίπτωση Αλλιώς
 <Ομάδα εντολών ν>
Τέλος_Επιλογών

Παράδειγμα: Έστω ότι ο βαθμός ενός μαθητή χαρακτηρίζεται βάση του παρακάτω πίνακα. Να γίνει πρόγραμμα το οποίο θα διαβάζει τον βαθμό του μαθητή και θα τυπώνει τον κατάλληλο χαρακτηρισμό.

0-9	10-15	16-18	19-20
Κακός	Μέτριος	Καλός	Άριστος

Πρόγραμμα Χαρακτηρισμός_Βαθμού
Μεταβλητές

Ακέραιες B

Αρχή

Γράψε 'δώσε το βαθμό του μαθητή'
Διάβασε B

Αν ($B > 0$) και ($B \leq 9$) **τότε**
Γράψε 'Κακός'

Αλλιώς_Αν ($B \leq 15$) **τότε**
Γράψε 'Μέτριος'

Αλλιώς_Αν ($B \leq 18$)
Γράψε 'Καλός'

Αλλιώς
Γράψε 'Άριστος'

Τέλος_Αν

Τέλος_προγράμματος

Παράδειγμα: Σε κάθε μέρα της εβδομάδας μπορούμε να αντιστοιχίσουμε έναν αριθμό, όπως στον παρακάτω πίνακα. Να γίνει πρόγραμμα το οποίο θα διαβάζει έναν αριθμό από το 1 έως το 7 και θα τυπώνει το όνομα της αντίστοιχης ημέρας της εβδομάδας.

1	2	3	4	5	6	7
Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή	Σάββατο	Κυριακή

Πρόγραμμα Ημέρες_Εβδομάδας
Μεταβλητές

Ακέραιες Η
Αρχή
Γράψε 'δώσε τον αριθμό της ημέρας'
Διάβασε Η
Επίλεξε Η
 Περίπτωση 1
 Γράψε 'Δευτέρα'
 Περίπτωση 2
 Γράψε 'Τρίτη'
 Περίπτωση 3
 Γράψε 'Τετάρτη'
 Περίπτωση 4
 Γράψε 'Πέμπτη'
 Περίπτωση 5
 Γράψε 'Παρασκευή'
 Περίπτωση 6
 Γράψε 'Σάββατο'
 Περίπτωση Αλλιώς
 Γράψε 'Κυριακή'
Τέλος_Επιλογών

Τέλος_προγράμματος

Η εντολή **Αλλιώς_Αν** μπορεί να χρησιμοποιηθεί είτε έχουμε διακριτές τιμές (δεύτερο παράδειγμα), είτε διαστήματα τιμών (πρώτο παράδειγμα). Οπότε θα μπορούσε να χρησιμοποιηθεί για την επίλυση και των δύο παραπάνω ασκήσεων. Αντίθετα η **επίλεξε** χρησιμοποιείται μόνο για διακριτές τιμές της έκφρασης που εξετάζεται.

Αλγόριθμος A3
 $\alpha \leftarrow 'A'$
 αν ($\alpha = 'Φ'$) τότε
 Εμφάνισε '5 Ευρώ'
 Αλλιώς_αν ($\alpha = 'Δ'$) τότε
 Εμφάνισε '2 Ευρώ'
 Αλλιώς
 Εμφάνισε '3 Ευρώ'
 Τέλος_αν
 Τέλος A3

Αλγόριθμος A4
 $\alpha \leftarrow$ ψευδής
 $\beta \leftarrow$ αλήθης
 αν (α και β) τότε
 Εμφάνισε α
 Αλλιώς
 Εμφάνισε β
 Τέλος_αν
 Τέλος A4

Αλγόριθμος A5
 $\alpha \leftarrow$ ψευδής
 $\beta \leftarrow$ αλήθης
 $\gamma \leftarrow$ αληθής
 αν ($(\alpha \text{ ή } \beta)$ και γ) τότε
 Εμφάνισε 'αληθής'
 Αλλιώς
 Εμφάνισε 'ψευδής'
 Τέλος_αν
 Τέλος A5

Αλγόριθμος A6
 $\alpha \leftarrow 10$
 $\beta \leftarrow 20$
 $\gamma \leftarrow 30$
 Αν ($\alpha^2 > \gamma$) και ($\beta < 0$) τότε
 Εμφάνισε 'περίπτωση 1'
 Αλλιώς_αν ($\alpha < \gamma$) και ($\beta - \gamma < 0$) τότε
 Εμφάνισε 'περίπτωση 2'
 Αλλιώς
 Εμφάνισε 'περίπτωση 3'
 Τέλος_αν
 Τέλος A6

Εκτέλεση Αλγορίθμων Β'

α) Να βρεθεί τι θα εμφανίσουν οι παρακάτω αλγόριθμοι όταν δοθούν οι τιμές 100 και 200 και τι τιμές θα έχουν οι μεταβλητές μετά την εκτέλεση.

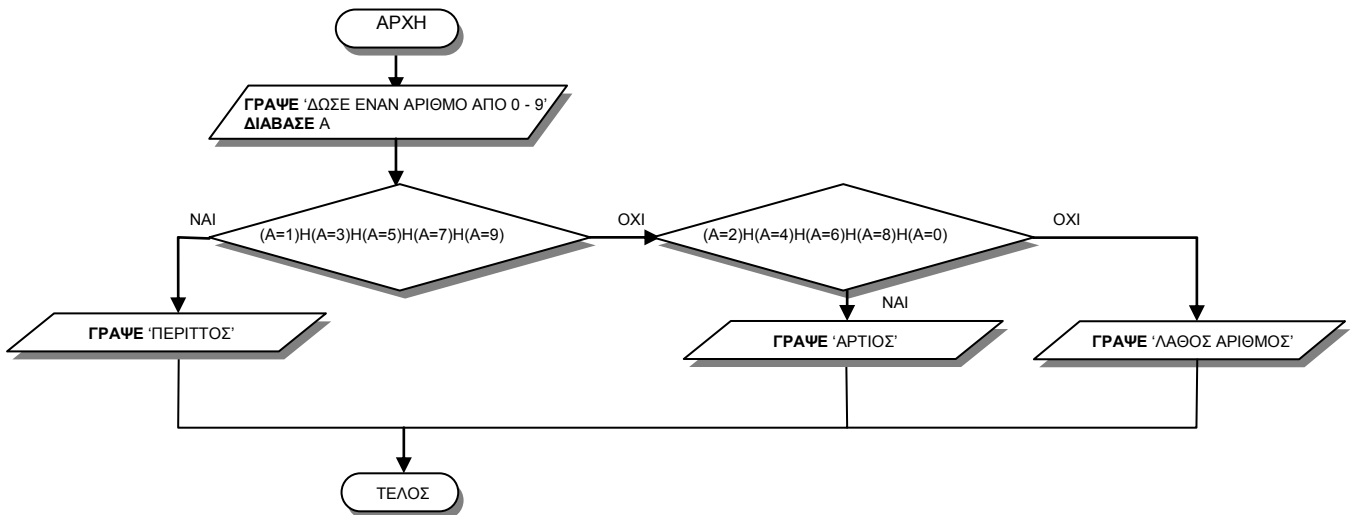
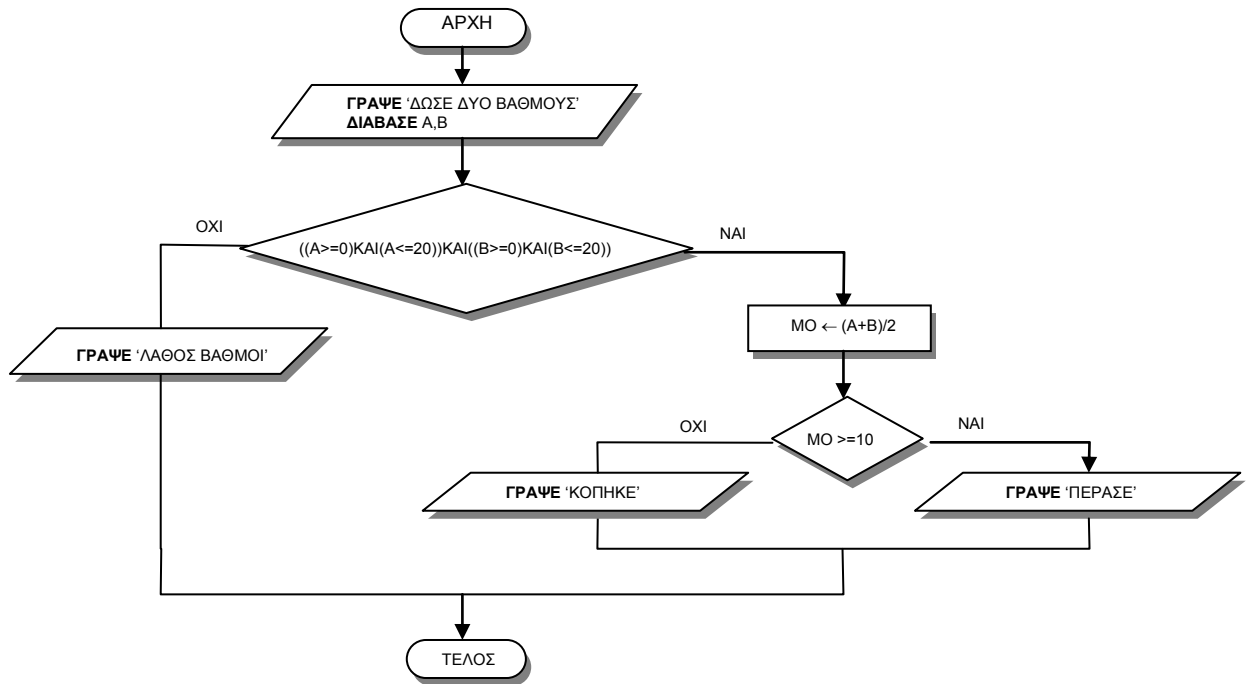
β) Να σχεδιάσετε τα αντίστοιχα διαγράμματα ροής.

Αλγόριθμος A1
 Διάβασε α, β
 $\beta \leftarrow \beta \text{ div } \alpha$
 αν ($\alpha = \beta$) τότε
 temp $\leftarrow \alpha$
 $\alpha \leftarrow \beta$
 $\beta \leftarrow$ temp
 $\gamma \leftarrow$ temp
 Αλλιώς
 $\beta \leftarrow \beta \text{ div } 2$
 $\alpha \leftarrow \alpha \text{ mod } 2$
 $\gamma \leftarrow 0$
 Τέλος_αν
 Επίλεξε β
 Περίπτωση 10
 Εμφάνισε β
 Περίπτωση 20
 Εμφάνισε $2 * \beta$
 Περίπτωση Αλλιώς
 Εμφάνισε $3 * \beta$
 Τέλος_Επιλογής
 Τέλος A1

Αλγόριθμος A2
 Διάβασε α, β
 $\gamma \leftarrow \beta \text{ mod } 10$
 Αν ($\gamma = 0$) τότε
 $\gamma \leftarrow 10$
 Τέλος_Αν
 Αν ($\alpha = 10$) τότε
 $\gamma \leftarrow \alpha \text{ mod } 10$
 Αλλιώς
 Αν ($\alpha < 10$) τότε
 $\gamma \leftarrow \alpha \text{ mod } 90$
 Αλλιώς
 $\gamma \leftarrow \alpha \text{ mod } 80$
 Τέλος_Αν
 Τέλος_Αν
 Εμφάνισε α, β, γ
 Τέλος A2

Μετατροπή Αλγορίθμων

Μετατρέψτε τα παρακάτω διαγράμματα ροής στους αντίστοιχους αλγορίθμους σε ΓΛΩΣΣΑ.



Ασκήσεις Δομής Επιλογής

1. Έστω το παρακάτω πρόγραμμα:

```
Πρόγραμμα Άσκ1
Μεταβλητές
    Ακέραιος: x,y
Αρχή
    Διάβασε x,y
    Αν  $x*x > y$  τότε
         $x \leftarrow x + y$ 
    τέλος_αν
    αν  $x*x > y$  τότε
         $x \leftarrow x + 4$ 
    αλλιώς
         $y \leftarrow y - x$ 
    τέλος_αν
    γράψε x,y
Τέλος_προγράμματος Άσκ1
```

α) Για κάθε βήμα του παραπάνω προγράμματος να γραφούν οι τιμές που παίρνουν οι μεταβλητές του για δυο διαφορετικές εκτελέσεις του, με εισόδους

- 1) 2 και 4
- 2) -5 και 20

β) Να γίνει το αντίστοιχο διάγραμμα ροής.

2. Να ξαναγράψετε το παρακάτω τμήμα αλγορίθμου ώστε να χρησιμοποιεί την δομή πολλαπλής επιλογής «**επίλεξε**». Να γίνει το διάγραμμα ροής του αλγορίθμου που θα προκύψει.

```
Διάβασε όνομα
Αν όνομα = 'Κάρολος' τότε
    Γράψε 'πέρασες'
Τέλος_αν
Αν όνομα = 'Γιάννης' τότε
    Γράψε 'έμεινες'
Τέλος_αν
Αν όνομα = 'Λεωνίδα' τότε
    Γράψε 'δώσε ξανα'
Τέλος_αν
```

3. Έστω το παρακάτω πρόγραμμα:

```
Πρόγραμμα Άσκ3
Μεταβλητές
    Πραγματικός: x
    Ακέραιος: y
Αρχή
    Διάβασε x,y
    Αν  $x > 0$  τότε
        Αν  $x*y > 0$  τότε
            Γράψε 'το y είναι θετικός'
        αλλιώς
            Γράψε 'το y είναι αρνητικός'
        τέλος_αν
    αλλιώς
        Γράψε 'το x δεν είναι θετικός'
    τέλος_αν
Τέλος_προγράμματος Άσκ3
```

α) Να βρείτε τι θα εμφανίσει το πρόγραμμα για τρεις διαφορετικές εκτελέσεις του, με εισόδους

- 1) 5 και -2
- 2) -4 και 11
- 3) 0 και 0

β) Να γίνει το αντίστοιχο διάγραμμα ροής.

4. Εάν οι μεταβλητές α, β και γ έχουν ως αρχική τιμή 2, ποιες από αυτές θα έχουν πάλι την τιμή 2 μετά την εκτέλεση του ακόλουθου τμήματος αλγορίθμου; (γ)

αν $a + b > \gamma$ **τότε**

$$a \leftarrow b + \gamma$$

αλλιώς

$$a \leftarrow \gamma - b$$

τέλος_αν

αν $a > b + \gamma$ **τότε**

$$b \leftarrow a - \gamma$$

αλλιώς

$$b \leftarrow a + \gamma$$

τέλος_αν

αν $a + b > b - \gamma$ **τότε**

$$\gamma \leftarrow a + b$$

αλλιώς

$$a \leftarrow b - \gamma$$

τέλος_αν

αν $a + b > \gamma$ **τότε**

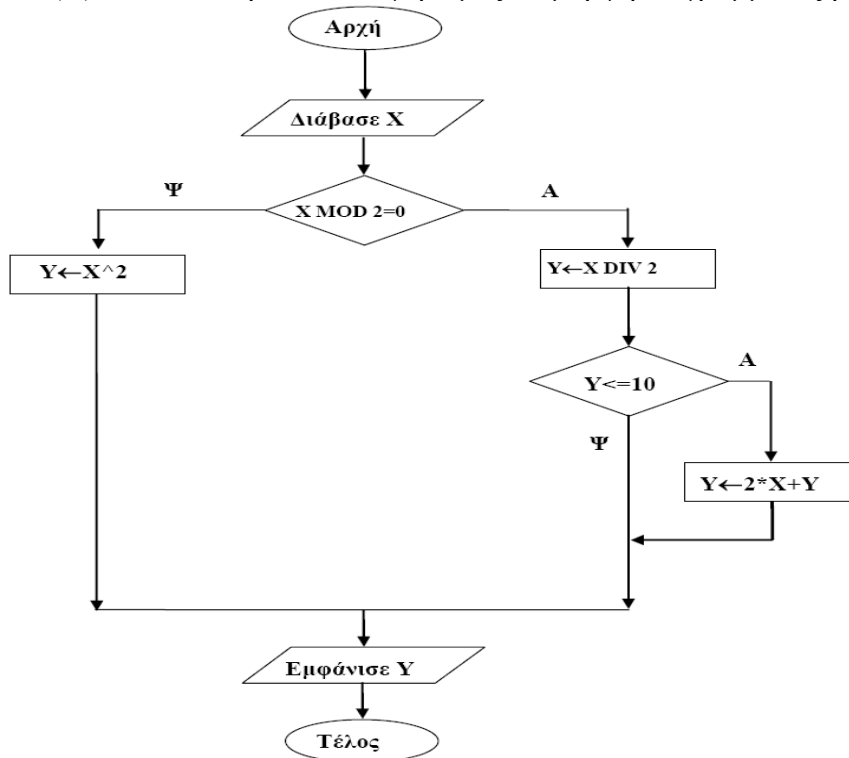
$$a \leftarrow b + \gamma$$

αλλιώς

$$\gamma \leftarrow b - a$$

τέλος_αν

5. (Π) Δίνεται ο παρακάτω αλγόριθμος σε μορφή διαγράμματος ροής:



α. Να κατασκευάσετε ισοδύναμο αλγόριθμο σε ψευδογλώσσα.

β. Να εκτελέσετε τον αλγόριθμο για κάθε μία από τις παρακάτω τιμές της μεταβλητής X. Να γράψετε στο τετράδιό σας την τιμή της μεταβλητής Y, όπως θα εμφανισθεί σε κάθε περίπτωση.

i. $X = 9$

ii. $X = 10$

iii. $X = 40$

6. (Π) Δίνεται το παρακάτω τμήμα αλγορίθμου σε φυσική γλώσσα κατά βήματα:

Βήμα 1: Αν $A > 0$ τότε πήγαινε στο **Βήμα 5**

Βήμα 2: Αν $A = 0$ τότε πήγαινε στο **Βήμα 7**

Βήμα 3: Τύπωσε “Αρνητικός”

Βήμα 4: Πήγαινε στο **Βήμα 8**

Βήμα 5: Τύπωσε “Θετικός”

Βήμα 6: Πήγαινε στο **Βήμα 8**

Βήμα 7: Τύπωσε “Μηδέν”

Βήμα 8: Τύπωσε “Τέλος”

1. Να σχεδιάσετε το ισοδύναμο διάγραμμα ροής.

2. Να κωδικοποιήσετε τον αλγόριθμο σε ψευδογλώσσα σύμφωνα με τις αρχές του δομημένου προγραμματισμού.

7. Να γίνει πρόγραμμα το οποίο θα βρίσκει το αποτέλεσμα της πράξης α/β για δυο αριθμούς που θα δίνει ο χρήστης.

8. Σε έναν κινηματογράφο μπορούμε να πληρώσουμε με δυο τρόπους.

α) με μηνιαία συνδρομή 30 € και 3 € ανά ταινία

β) χωρίς συνδρομή, με 10 € ανά ταινία

Να γίνει πρόγραμμα το οποίο θα διαβάζει το πλήθος των ταινιών που έχουμε σκοπό να δούμε στη διάρκεια ενός μήνα και να μας τυπώνει ένα μήνυμα για το ποιος τρόπος πληρωμής συμφέρει (π.χ. ‘Με συνδρομή’, ‘Χωρίς συνδρομή’).

9. Φτιάξτε ένα πρόγραμμα το οποίο υπολογίζει την τιμή της παρακάτω συνάρτησης όταν δίνεται η τιμή του X.

$$3/X, \text{ για } X > 0$$

$f(x) =$

$$5 \cdot X + 4, \text{ για } X \leq 0$$

10. Φτιάξτε ένα πρόγραμμα το οποίο υπολογίζει την τιμή της παρακάτω συνάρτησης όταν δίνεται η τιμή του X.

$$f(x) = \begin{cases} 3/X, & \text{για } X > 0 \\ 20, & \text{για } X = 0 \\ 5 \cdot X + 4, & \text{για } X < 0 \end{cases}$$

11. Να γίνει πρόγραμμα για τον υπολογισμό της παρακάτω παράστασης όταν σας δίνονται τα a και x.

$$Y = 4 \frac{a(3-x)}{x(x-1)}$$

12. Να γίνει πρόγραμμα για τον υπολογισμό της παρακάτω παράστασης όταν σας δίνεται το x.

$$Y = \begin{cases} \frac{2-x}{x-1} + \frac{x+1}{x}, x \geq 0 \\ \frac{3-x}{x+4} + \sqrt{3-x}, x < 0 \end{cases}$$

13. Να γραφεί πρόγραμμα το οποίο θα δέχεται έναν ακέραιο και θα εμφανίζει το διπλάσιό του αν ο αριθμός αυτός τελειώνει σε 3 ή σε 4, αλλιώς θα τυπώνει το τριπλάσιό του.
14. Να γίνει πρόγραμμα το οποίο θα διαβάζει έναν ακέραιο αριθμό και θα βρίσκει αν είναι άρτιος ή περιττός. Το πρόγραμμα θα τυπώνει το κατάλληλο μήνυμα.
15. Να γίνει πρόγραμμα το οποίο θα δέχεται δυο ακεραίους και ελέγχει αν ο ένας είναι ακέραιο πολλαπλάσιο του άλλου. Στο τέλος θα τυπώνει μόνο ένα 'ναι' ή ένα 'όχι'.
16. Για να θεωρηθεί ότι στην πόλη της Καρδίτσας είχαμε παγετό κατά τη διάρκεια της νύχτας, θα πρέπει σε τρία διαφορετικά σημεία της πόλης η θερμοκρασία να έπεσε κάτω από τους μηδέν βαθμούς Κελσίου. Να γίνει πρόγραμμα το οποίο θα διαβάζει τις τρεις θερμοκρασίες και θα τυπώνει το μήνυμα «παγετός», αν και οι τρεις είναι μικρότερες του μηδενός.
17. Ένας εργαζόμενος δουλεύει σε μια εταιρία ως ωρομίσθιος. Αν οι αποδοχές του κατά τη διάρκεια του έτους δεν ξεπεράσουν τις 5000 €, τότε δεν φορολογείται, αλλιώς πληρώνει φόρο 5% του συνολικού εισοδήματος. Να γίνει πρόγραμμα το οποίο θα διαβάζει τις ώρες που εργάστηκε σε ένα έτος, τα χρήματα που παίρνει για κάθε ώρα εργασίας και θα τυπώνει το ποσό του φόρου.
18. Φτιάξτε ένα πρόγραμμα το οποίο θα διαβάζει δυο αριθμούς, θα υπολογίζει τον μεγαλύτερο, θα τον αποθηκεύει σε μια μεταβλητή max και θα τυπώνει το περιεχόμενό της.
19. Φτιάξτε ένα πρόγραμμα το οποίο θα βρίσκει και θα τυπώνει το μεγαλύτερο από τρεις αριθμούς που θα διαβάζει.
20. Στην περιοχή της Θεσσαλίας, οι τέσσερις μεγαλύτερες πόλεις είχαν κατά τη διάρκεια μιας ημέρας τέσσερις ξεχωριστές θερμοκρασίες A, B, Γ, και Δ. Να γίνει πρόγραμμα που θα διαβάζει τις θερμοκρασίες και θα βρίσκει τη μικρότερη.
21. Να γίνει πρόγραμμα το οποίο θα δέχεται τρεις ακέραιους αριθμούς. Αν το άθροισμά τους είναι θετικό θα υπολογίζει και θα τυπώνει το μέσο όρο τους, αλλιώς θα βρίσκει και θα τυπώνει το μέγιστο από τους τρεις.
22. Στους προκριματικούς αγώνες στο άλμα εις μήκος στην Ολυμπιάδα, ένας αθλητής κάνει 3 αρχικές προσπάθειες και αν έχει επίδοση μεγαλύτερη από 7.50 μέτρα, τότε δικαιούται να συνεχίσει και να κάνει κι άλλες 3 ακόμα προσπάθειες. Να διαβασθούν οι 3 πρώτες προσπάθειες ενός αθλητή και να εκτυπωθεί ένα μήνυμα αν δικαιούται να συνεχίσει ή όχι. Στην περίπτωση που δικαιούται να διαβαστούν και ο υπόλοιπες τρεις προσπάθειες και να βρεθεί και να εκτυπωθεί η καλύτερη προσπάθεια του αθλητή.
23. Σε ένα πολυκατάστημα υπάρχουν οι παρακάτω τιμές για 4 διαφορετικά σαμπουάν. Να γράψετε αλγόριθμο που να υπολογίζει και να εμφανίζει το σαμπουάν με την λιγότερο συμφέρουσα τιμή.

Είδος	Τιμή	Ποσότητα
Σαμπουάν Α	5,2 €	500 ml
Σαμπουάν Β	7 €	750ml
Σαμπουάν Γ	18 €	1,5 lt
Σαμπουάν Δ	3,8 €	375 ml

24. Σε τρία διαφορετικά σημεία της Αθήνας καταγράφηκαν στις 12 το μεσημέρι οι θερμοκρασίες A, B, C
 Να αναπτύξετε αλγόριθμο που :
- 1) Να διαβάσει τις θερμοκρασίες A, B, C
 - 2) Να υπολογίζει και να εμφανίζει τη μέση τιμή των παραπάνω θερμοκρασιών
 - 3) Να εμφανίζει το μήνυμα 'ΚΑΥΣΩΝΑΣ' αν η μέση τιμή είναι μεγαλύτερη των 37 βαθμών Κελσίου
25. Να διαβασθούν δύο ακέραιοι αριθμοί και να υπολογισθεί η απόλυτη τιμή της διαφοράς τους χωρίς τη χρήση της συνάρτησης A_T.
26. Να διαβασθούν δύο ακέραιοι αριθμοί a και b, όπου $a > b$, και να βρεθεί αν ο b είναι διαιρέτης του a. Να τυπώνεται το αντίστοιχο μήνυμα (ναι ή όχι)
27. Να διαβασθεί η ώρα σε 24ωρη μορφή σαν ένας τετραψήφιος αριθμός, όπως για παράδειγμα 1452, να βρεθεί πρώτα αν παριστάνει σωστή ένδειξη χρόνου και αν ναι, να εμφανισθεί η ώρα στην αντίστοιχη 12ωρη μορφή της, για παράδειγμα 02:52 μμ.
28. Σε ένα πανεπιστημιακό μάθημα ο τελικός βαθμός του φοιτητή υπολογίζεται από το μέσο όρο των βαθμολογιών του στην γραπτή και την προφορική εξέταση. Αν σε μια από τις δυο εξετάσεις αποτύχει (βαθμός < 5) δεν περνά το μάθημα. Να γίνει πρόγραμμα το οποίο θα διαβάζει τις δυο βαθμολογίες και θα υπολογίζει τον τελικό βαθμό του φοιτητή αν έχει πετύχει, αλλιώς θα τυπώνει ότι απέτυχε.
29. Για να θεωρείται επιτυχών ένας υποψήφιος σ' έναν διαγωνισμό, θα πρέπει να εξετασθεί γραπτά σε δύο μαθήματα και να λάβει βαθμολογία τουλάχιστον 55 στο κάθε μάθημα αλλά και μέσο όρο από τα δύο μαθήματα τουλάχιστον 60. Να διαβασθούν οι βαθμοί του υποψηφίου στα δύο μαθήματα (κλίμακα από 0-100) και να βρεθεί αν είναι επιτυχών ή όχι.
30. Σύμφωνα με το Διατραπεζικό Σύστημα Συναλλαγών (ΔΙΑ.Σ.), μπορούν να γίνουν αναλήψεις από ένα μηχάνημα ATM μιας Τράπεζας αλλά με την κάρτα μιας άλλης Τράπεζας. Οι αναλήψεις αυτές χρεώνονται, όμως, με το 1% του ποσού της ανάληψης αλλά η χρέωση αυτή δεν πρέπει να είναι μικρότερη από 1 € και ούτε μεγαλύτερη από 3 €. Να καταχωρηθεί ένα αριθμητικό ποσό σε ευρώ και να βρεθεί η χρέωση που θα έχει ο πελάτης της Τράπεζας για την ανάληψη που θα κάνει, σύμφωνα με το ΔΙΑ.Σ.
31. Να διαβασθεί ο αριθμός των δικαιολογημένων και των αδικαιολόγητων απουσιών ενός μαθητή και να βρεθεί αν ο μαθητής προάγεται ή απορρίπτεται λόγω απουσιών αν είναι γνωστό ότι το όριο των δικαιολογημένων απουσιών είναι 50 και το όριο των αδικαιολόγητων απουσιών είναι 60.
32. Η ωριαία αμοιβή ενός εργαζομένου είναι 6 €. Όμως, αν οι ώρες εργασίας του είναι περισσότερες από 25, λαμβάνει και επιπλέον υπερωριακή αποζημίωση 3 € για κάθε επιπλέον ώρα. Να διαβασθούν οι ώρες εργασίας και να υπολογισθούν οι αποδοχές ενός εργαζομένου (κανονική αμοιβή, αμοιβή υπερωριών και συνολικές αποδοχές).
33. Να διαβασθούν τα μήκη των 4 πλευρών και μια γωνία ενός τετραπλεύρου και να υπολογισθεί το είδος του (τετράγωνο, ρόμβος, ορθογώνιο, παραλληλόγραμμο, τυχαίο).
34. Να διαβασθεί ένας ακέραιος αριθμός και να βρεθεί και να εκτυπωθεί ολογράφως το ακέραιο υπόλοιπο της διαίρεσής του με το 3.
35. Φτιάξτε ένα πρόγραμμα το οποίο θα διαβάζει από το χρήστη τρεις γωνίες και θα απαντά για το αν το τρίγωνο με αυτές τις γωνίες είναι σκαληνό, ισόπλευρο ή ισοσκελές. Αν οι γωνίες έχουν άθροισμα διάφορο των 180° , θα τυπώνει το μήνυμα 'λάθος δεδομένα'.

36. Δυο πωλητές μιας εταιρίας κάνουν σε ένα μήνα πωλήσεις αξίας X και Y € αντίστοιχα. Η εταιρία τους δίνει πριμ 100 € αν ξεπεράσουν τα 3000 € σε πωλήσεις. Αν και οι δυο ξεπεράσουν αυτό το όριο τότε θα μοιραστούν το πριμ. Κάντε ένα πρόγραμμα το οποίο θα διαβάζει την αξία των πωλήσεων για κάθε πωλητή και θα επιστρέφει τα πριμ τους (ακόμα και αν είναι 0).
37. Η Εφορία κάνει έκπτωση 10% στις πληρωμές που γίνονται μετρητοίς, αλλά η έκπτωση αυτή δεν μπορεί να ξεπερνάει τα 1.000 €. Να διαβασθεί το ποσό που καλείται να πληρώσει κάποιος στην Εφορία και να υπολογισθεί και να εκτυπωθεί η έκπτωση που θα του γίνει αν πληρώσει μετρητοίς καθώς και το ποσό που τελικά θα πληρώσει.
38. Για να έχει δικαίωμα μια οικογένεια με δύο παιδιά που σπουδάζουν να ζητήσει μετεγγραφή, θα πρέπει ο μέσος όρος των εισοδημάτων της την τελευταία τριετία να μην ξεπερνάει τα 35.000 €. Να διαβασθούν τα εισοδήματα των τριών τελευταίων ετών μιας οικογένειας που έχει δύο παιδιά που σπουδάζουν, να υπολογισθεί το μέσο ετήσιο εισόδημά της και να βρεθεί αν έχει το δικαίωμα ή όχι να ζητήσει μετεγγραφή.
39. Μια εταιρεία χορηγεί επίδομα σπουδών στους υπαλλήλους της με βάση τις γραμματικές τους γνώσεις και σαν ποσοστό του βασικού τους μισθού ως εξής : 5% για τους αποφοίτους γυμνασίου, 10% για τους αποφοίτους λυκείου και 20% για τους πτυχιούχους. Να διαβασθεί ο βασικός μισθός και ο κωδικός σπουδών (1, 2, 3) ενός υπαλλήλου και να υπολογισθεί το επίδομα σπουδών που θα του χορηγηθεί.
40. Το επίδομα παιδιών ενός υπαλλήλου υπολογίζεται βάσει του βασικού του μισθού ως εξής :
Για τα δύο πρώτα παιδιά είναι 4% για το κάθε παιδί.
Για το τρίτο παιδί είναι 6%.
Για κάθε επιπλέον παιδί μετά το τρίτο είναι 10%.
Να διαβασθεί ο βασικός μισθός ενός υπαλλήλου και ο αριθμός των παιδιών του και να υπολογισθεί το επίδομα παιδιών που δικαιούται.
41. Ένας πωλητής σε μια εταιρεία έχει σταθερές αποδοχές 500 € και αν φέρει εισπράξεις από 1.000 € έως και 2.000 €, λαμβάνει προμήθεια 5% επί των εισπράξεων, ενώ αν φέρει εισπράξεις παραπάνω από 2.000 €, λαμβάνει προμήθεια 8% επί των εισπράξεων. Να διαβασθούν οι εισπράξεις που έφερε στην εταιρεία ένας πωλητής και να βρεθούν οι συνολικές αποδοχές του.
42. (Π)Να αναπτύξετε αλγόριθμο ο οποίος υλοποιεί τη λειτουργία ενός αυτόματου τυποποιητή πορτοκαλιών που είναι η παρακάτω: Για κάθε πορτοκάλι που εισάγεται στον τυποποιητή, διαβάζεται η τιμή του βάρους του (B) και η διάμετρος του (Δ). Το πορτοκάλι κατατάσσεται ανάλογα με το βάρος και τη διάμετρό του ως εξής:

- 1) Αν $100 < B < 150$ και $8 < Δ < 10$, τότε τυπώνεται το μήνυμα "πρώτη διαλογή".
- 2) Αν $6 < Δ < 8$, τότε, ανεξαρτήτως βάρους, τυπώνεται το μήνυμα "δεύτερη διαλογή".
- 3) Σε κάθε άλλη περίπτωση τυπώνεται το μήνυμα "χυμοποίηση".

43. (Π)Ο Δείκτης Μάζας του ανθρώπινου Σώματος (ΔΜΣ) υπολογίζεται από το βάρος (B) σε χιλ. και το ύψος (Y) σε μέτρα με τον τύπο $ΔΜΣ=B/Y^2$. Ο ανωτέρω τύπος ισχύει για άτομα άνω των 18 ετών. Το άτομο ανάλογα με την τιμή του ΔΜΣ χαρακτηρίζεται σύμφωνα με τον παρακάτω πίνακα:

$ΔΜΣ < 18,5$	"αδύνατο άτομο"
$18,5 < ΔΜΣ < 25$	"κανονικό άτομο"
$25 < ΔΜΣ < 30$	"βαρύ άτομο"
$30 < ΔΜΣ$	"υπέρβαρο άτομο".

Να γράψετε αλγόριθμο σε ΓΛΩΣΣΑ ο οποίος:

- α. να διαβάζει την ηλικία, το βάρος και το ύψος του ατόμου

- β. εάν η ηλικία είναι μεγαλύτερη των 18 ετών, τότε
- 1) να υπολογίζει το ΔΜΣ
 - 2) να ελέγχει την τιμή του ΔΜΣ από τον ανωτέρω πίνακα και να εμφανίζει τον αντίστοιχο χαρακτηρισμό
- γ. εάν η ηλικία είναι μικρότερη ή ίση των 18 ετών, τότε να εμφανίζει το μήνυμα "δεν ισχύει ο δείκτης ΔΜΣ".
44. (Π)Σε τρεις διαφορετικούς αγώνες πρόκρισης για την Ολυμπιάδα του Σίδνεϋ στο άλμα εις μήκος ένας αθλητής πέτυχε τις επιδόσεις a, b, c. Να αναπτύξετε αλγόριθμο ο οποίος:
- α) να διαβάσει τις τιμές των επιδόσεων a, b, c.
 - β) να υπολογίζει και να εμφανίζει τη μέση τιμή των παραπάνω τιμών.
 - γ) να εμφανίζει το μήνυμα «ΠΡΟΚΡΙΘΗΚΕ», αν η παραπάνω μέση τιμή είναι μεγαλύτερη των 8 μέτρων.
45. Να γραφεί αλγόριθμος ο οποίος θα διαβάσει το γενικό μέσο όρο ενός μαθητή και το βαθμό στο μάθημα των μαθηματικών και της φυσικής και θα εμφανίζει το μήνυμα «Προάγεσαι» ή το μήνυμα «Δεν προάγεσαι». Ο μαθητής προάγεται ή απορρίπτεται βάσει των εξής κανόνων:
- 1) Ο μαθητής προάγεται αν ο γενικός μέσος όρος των μαθημάτων του είναι τουλάχιστον 9.5.
 - 2) Αν ο γενικός μέσος όρος είναι μεταξύ του 8.5 και του 9.5, μη συμπεριλαμβανομένου του πάνω ορίου, ο μαθητής προάγεται αν ο μέσος όρος που προκύπτει από τα μαθηματικά και τη φυσική είναι μεγαλύτερος από 10. Διαφορετικά απορρίπτεται.
 - 3) Αν ο γενικός μέσος όρος είναι μικρότερος του 8.5 ο μαθητής απορρίπτεται.
46. Σε έναν αγώνα τοξοβολίας κάθε αθλητής έχει 5 προσπάθειες. Η βαθμολογία του είναι ο μέσος όρος των επιμέρους βαθμολογιών. Έστω ότι κάθε προσπάθεια βαθμολογείτε από 1 έως 10 και ότι με το μηδέν υποδεικνύεται η βολή εκτός στόχου. Να γίνει πρόγραμμα που θα υπολογίζει την τελική βαθμολογία. Αν ο αθλητής χάσει πάνω από 3 προσπάθειες (πάρει 0) θα αποκλείεται.
47. Να γραφεί πρόγραμμα το οποίο θα διαβάσει δυο ακέραιες μεταβλητές και μια μεταβλητή χαρακτήρα. Αν η μεταβλητή χαρακτήρα πάρει τιμή '+', '*', '-', '/', τότε το πρόγραμμα θα κάνει την αντίστοιχη πράξη μεταξύ των δύο αριθμητικών τιμών. Αν η μεταβλητή χαρακτήρα έχει άλλη τιμή τότε θα τυπώνεται το μήνυμα 'λάθος τελεστής'. Προσοχή στην περίπτωση της διαίρεσης.
48. Μια εταιρία δίνει επίδομα στους υπαλλήλους της με βάση τον αριθμό των παιδιών που έχουν. Αν έχουν 1 παιδί, παίρνουν επίδομα ίσο με το 5% του βασικού μισθού, αν έχουν 2 παιδιά παίρνουν 10%, αν έχουν 3 παιδιά παίρνουν 15% και για πάνω από τρία παιδιά παίρνουν 20%. Να γίνει πρόγραμμα το οποίο θα διαβάσει το βασικό μισθό και τον αριθμό των παιδιών του υπαλλήλου και θα υπολογίζει το επίδομα που θα πάρει.
49. Να γίνει πρόγραμμα το οποίο θα διαβάσει 5 ακέραιους αριθμούς και θα βρίσκει το άθροισμα μόνο των θετικών.
50. Να γίνει πρόγραμμα που θα διαβάσει τις βαθμολογίες πέντε μαθητών σε ένα μάθημα και θα υπολογίζει και εκτυπώνει πόσοι έχουν βαθμολογία κάτω από 10.
51. Να γίνει πρόγραμμα το οποίο θα διαβάσει 5 ακεραίους και θα μετρά πόσοι από αυτούς είναι άρτιοι. Το πρόγραμμα θα τυπώνει το αποτέλεσμα.
52. Σε ένα πανεπιστημιακό μάθημα ένας φοιτητής έχει δικαίωμα να κάνει 20 απουσίες. Αν ξεπεράσει τις 20 απουσίες τότε κόβεται. Αν δεν υπερβεί το όριο των απουσιών τότε περνά το μάθημα αν ο μέσος όρος της γραπτής και της προφορικής του βαθμολογίας είναι τουλάχιστον 5. Να γίνει πρόγραμμα το οποίο:

- 1) α) θα διαβάξει τις απουσίες του φοιτητή και θα εκτυπώνει το μήνυμα **‘κόβεται από απουσίες’** αν έχει ξεπεράσει το όριο.
 - 2) β) αν δεν έχει ξεπεράσει το όριο, τότε θα διαβάξει την γραπτή και την προφορική του βαθμολογία, θα υπολογίζει το μέσο όρο τους και θα τυπώνει το μήνυμα **‘κόβεται λόγω βαθμολογίας’** αν ο μέσος όρος είναι κάτω από 5, αλλιώς θα τυπώνει το μήνυμα **‘Πέρασε’**.
53. Σε ένα μάθημα πανελληνίων εξετάσεων ο βαθμός ενός μαθητή υπολογίζεται από τον μέσο όρο των βαθμολογιών των δυο βαθμολογητών. Αν όμως η διαφορά στην βαθμολογία τους μεγαλύτερη των τριών μονάδων, τότε γίνεται αναβαθμολόγηση και ο βαθμός του μαθητή είναι η βαθμολογία του τρίτου βαθμολογητή. Να γίνει πρόγραμμα το οποίο:
- 1) α) θα διαβάξει τις 2 βαθμολογίες.
 - 2) β) θα ελέγχει αν χρειάζεται αναβαθμολόγηση και αν ναι θα διαβάξει και τρίτο βαθμό, αλλιώς θα υπολογίζει τον μέσο όρο των δύο πρώτων.
 - 3) γ) θα εκτυπώνει την τελική βαθμολογία του μαθητή
54. Σε ένα μάθημα δίνεται μεγαλύτερη βαρύτητα στην γραπτή βαθμολογία. Αυτό σημαίνει πως αν η γραπτή και η προφορική βαθμολογία ενός μαθητή διαφέρουν πάνω από τρεις μονάδες, τότε η προφορική βαθμολογία πρέπει να προσαρμοστεί στην γραπτή, έτσι ώστε η διαφορά τους να είναι τρεις μονάδες (π.χ. αν οι βαθμολογίες (γραπτός-προφορικός) είναι 12 και 19, τότε ο προφορικός θα γίνει 15, αν είναι 19 και 11, τότε ο προφορικός θα γίνει 16). Ο μέσος όρος, που είναι και η τελική βαθμολογία, υπολογίζεται αφού γίνει η ενδεχόμενη αναπροσαρμογή. Να γίνει πρόγραμμα το οποίο:
- 1) α) θα διαβάξει την προφορική και τη γραπτή βαθμολογία ενός μαθητή
 - 2) β) θα εξετάζει αν χρειάζεται αναπροσαρμογή του προφορικού βαθμού στο γραπτό και αν ναι θα την κάνει.
 - 3) γ) θα υπολογίζει και θα εκτυπώνει τον μέσο όρο.
55. (Π)Με το νέο σύστημα πληρωμής των διοδίων, οι οδηγοί των τροχοφόρων έχουν τη δυνατότητα να πληρώνουν το αντίτιμο των διοδίων με ειδική μαγνητική κάρτα. Υποθέστε ότι υπάρχει μηχανήμα το οποίο διαθέτει είσοδο για την κάρτα και φωτοκύτταρο. Το μηχανήμα διαβάζει από την κάρτα το υπόλοιπο των χρημάτων και το αποθηκεύει σε μία μεταβλητή Y και, με το φωτοκύτταρο, αναγνωρίζει τον τύπο του τροχοφόρου και το αποθηκεύει σε μία μεταβλητή T. Υπάρχουν τρεις τύποι τροχοφόρων: δίκυκλα (Δ), επιβατικά (Ε) και φορτηγά (Φ), με αντίτιμο διοδίων 1, 2 και 3 € αντίστοιχα.

Να αναπτύξετε αλγόριθμο, ο οποίος:

- 1) α. ελέγχει τον τύπο του τροχοφόρου και εκχωρεί στη μεταβλητή A το αντίτιμο των διοδίων, ανάλογα με τον τύπο του τροχοφόρου.
 - 2) β. ελέγχει την πληρωμή των διοδίων με τον παρακάτω τρόπο. Αν το υπόλοιπο της κάρτας επαρκεί για την πληρωμή του αντιτίμου των διοδίων, αφαιρεί το ποσό αυτό από την κάρτα. Αν η κάρτα δεν έχει υπόλοιπο, το μηχανήμα ειδοποιεί με μήνυμα για το ποσό που πρέπει να πληρωθεί. Αν το υπόλοιπο δεν επαρκεί, μηδενίζεται η κάρτα και δίνεται με μήνυμα το ποσό που απομένει να πληρωθεί.
56. (Π)Μια εταιρεία κινητής τηλεφωνίας ακολουθεί ανά μήνα την πολιτική τιμών που φαίνεται στον παρακάτω πίνακα:

Πάγιο 1500 δραχμές	
Χρόνος τηλεφωνημάτων (δευτερόλεπτα)	Χρονοχρέωση (δραχμές/δευτερόλεπτο)
1-500	1,5
501-800	0,9

801 και πάνω	0,5
--------------	-----

Να αναπτύξετε αλγόριθμο ο οποίος:

- 1) α) να διαβάσει τη χρονική διάρκεια των τηλεφωνημάτων ενός συνδρομητή σε διάστημα ενός μήνα
- 2) β) να υπολογίζει τη μηνιαία χρέωση του συνδρομητή
- 3) γ) να εμφανίζει (τυπώνει) τη λέξη «ΧΡΕΩΣΗ» και τη μηνιαία χρέωση του συνδρομητή.

57. Να γίνει αλγόριθμος που θα διαβάσει τις αστικές και υπεραστικές μονάδες και θα εμφανίζει τη χρέωση του συνδρομητή. Η χρέωση στους λογαριασμούς του ΟΤΕ υπολογίζεται από των παρακάτω πίνακα

Πάγιο €3
 Αστικές Μονάδες €0,1/μονάδα
 Υπεραστικές μονάδες (κλιμακωτή χρέωση)
 0 - 200 μονάδες €0.12/μονάδα
 201 - 400 μονάδες €0.09/μονάδα
 401 - €0,08/μονάδα

ΦΠΑ 18%

58. Ένας ραδιοφωνικός σταθμός χρεώνει τις διαφημίσεις των πελατών του με 200 € ανά δευτερόλεπτο αν η διαφήμιση έχει διάρκεια έως και 20 δευτερόλεπτα, με 160 € ανά δευτερόλεπτο για τα δευτερόλεπτα από 21 έως και 30 και με 120 € ανά δευτερόλεπτο για τα επιπλέον δευτερόλεπτα. Επίσης, παρέχει 10% έκπτωση αν γίνονται από 5 έως και 10 διαφημιστικές εκπομπές ανά εβδομάδα και 20% έκπτωση αν γίνονται περισσότερες από 10 διαφημιστικές εκπομπές ανά εβδομάδα. Να διαβασθεί η διάρκεια σε δευτερόλεπτα μιας διαφήμισης και ο αριθμός των προβολών της για μια εβδομάδα και να βρεθεί η χρέωση που προκύπτει.

59. Ένας σταθμός αυτοκινήτων (parking) χρεώνει ως εξής την παραμονή των αυτοκινήτων : η πρώτη ώρα παραμονής χρεώνεται προς 5 €, η κάθε επόμενη ώρα μετά την πρώτη χρεώνεται προς 2 €, ενώ αν το αυτοκίνητο έχει ολοκληρώσει 12 ώρες παραμονής στον σταθμό όλες οι ώρες θα χρεωθούν προς 1,5 € η καθεμία. Η χρέωση μιας ώρας γίνεται αρκεί να έχει περάσει και ένα λεπτό. Για παράδειγμα, για παραμονή 3 ώρες και 5 λεπτά, ο πελάτης χρεώνεται συνολικά 4 ώρες. Να γίνει αλγόριθμος που να διαβάσει τον συνολικό χρόνο παραμονής σε λεπτά ενός αυτοκινήτου στον σταθμό και να υπολογίζει τη χρέωσή του.

60. Τα αυτοκίνητα που νοικιάζει ένα γραφείο ενοικίασης χρεώνονται προς 0,04 € το χιλιόμετρο για τα πρώτα 100 χιλιόμετρα και προς 0,06 € το χιλιόμετρο για τα υπόλοιπα χιλιόμετρα συν πάγιο 6 €. Να διαβασθούν τα χιλιόμετρα που διήνυσε ένα αυτοκίνητο και να υπολογισθεί η συνολική χρέωσή του.

61. Ο φόρος που πληρώνει ένας εργαζόμενος εξαρτάται από το ετήσιο εισόδημά του και υπολογίζεται βάση του παρακάτω πίνακα.

Ετήσιο Εισόδημα	Φόρος
0-10.000 €	0%
10.001-15.000 €	5%
15.001-20.000 €	7.5%
20.001 και άνω	10%

Να γίνει πρόγραμμα το οποίο:

- 1) α) θα διαβάσει τα ονοματεπώνυμα και τα εισοδήματα δυο εργαζομένων
- 2) β) θα υπολογίζει το φόρο του καθ' ενός
- 3) γ) θα τυπώνει το φόρο και το ονοματεπώνυμο του εργαζομένου με το μεγαλύτερο φόρο.
- 4) δ) θα τυπώνει το μήνυμα '**Διαφορετική Κατηγορία**' αν οι δύο εργαζόμενοι ανήκουν σε διαφορετική κατηγορία φορολόγησης.

62. Ένα Internet Cafe χρεώνει κλιμακωτά τους πελάτες του βάση του παρακάτω πίνακα.

Χρόνος χρήσης (λεπτά)	Χρονοχρέωση (€/λεπτό)
1-60	0,02
61-90	0,015
91 και πάνω	0,01

- α) Να γίνει πρόγραμμα το οποίο θα διαβάζει το χρόνο χρήσης του internet σε ώρες και λεπτά.
- β) Θα μετατρέπει τον χρόνο σε λεπτά.
- γ) Θα υπολογίζει το κόστος βάση του παραπάνω πίνακα.
- δ) Θα ρωτά το χρήστη (εμφανίζοντας το κατάλληλο μήνυμα) αν πήρε καφέ. Θα διαβάζει την απάντηση του χρήστη σε μια μεταβλητή τύπου χαρακτήρα ('γ' για το ΝΑΙ και 'η' για το ΟΧΙ). Αν ο χρήστης απαντήσει 'γ', τότε θα χρεώνεται ακόμα 2 €.
- ε) Θα τυπώνεται το τελικό κόστος.

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βλάχος Στέφανος

Σημειώσεις - Ερωτήσεις - Ασκήσεις
Κεφαλαίων 2, 8
(Δομή Επανάληψης)

ΚΕΦΑΛΑΙΟ 2-8 (Δομή Επανάληψης)

Στην αλγοριθμική και στον προγραμματισμό είναι πολύ συχνά απαραίτητη η επαναληπτική δομή. Η βασική ιδέα της επαναληπτικής δομής είναι να μπορούμε να εκτελούμε ένα σύνολο εντολών πολλές φορές, χωρίς να είμαστε υποχρεωμένοι να τις ξαναγράψουμε και χωρίς να είμαστε υποχρεωμένοι εκ των προτέρων να προβλέψουμε πόσες φορές θα επαναληφθούν.

Εντολή 'Για' (43-44,178-180)

Ας θεωρήσουμε την παρακάτω άσκηση.

Να γίνει αλγόριθμος ο οποίος θα τυπώνει 100 φορές το μήνυμα 'Καλημέρα'

Ένας 'απλός' τρόπος είναι ο εξής:

Αλγόριθμος δοκιμή

Γράψε 'Καλημέρα'

Γράψε 'Καλημέρα'

Γράψε 'Καλημέρα'

Γράψε 'Καλημέρα'

 .

 .

(υποθέτουμε ότι η εντολή γράφεται 100 φορές)

 .

Γράψε 'Καλημέρα'

Τέλος_δοκιμή

Μια τέτοια επίλυση μπορεί αρχικά να φαίνεται εύκολη, παρά το γεγονός ότι πρέπει να γράψουμε την εντολή 100 φορές. Τι θα γινόταν αν έπρεπε να αλλάξουμε το πρόγραμμα, ώστε να γράφει τη φράση 1000 η και παραπάνω φορές ή αν έπρεπε να επαναλάβουμε περισσότερες από μια εντολές;

Η εντολή **'για'** μας βοηθά σε τέτοιες περιπτώσεις, **όπου γνωρίζουμε τον αριθμό των επαναλήψεων** να γράψουμε μικρά προγράμματα σε πολύ λίγο χρόνο. Η σύνταξη της εντολής είναι:

για μεταβλητή από τιμή1 μέχρι τιμή2 με βήμα τιμή3

<Ομάδα εντολών>

τέλος_επανάληψης

Για παράδειγμα ο παραπάνω αλγόριθμος θα μπορούσε να γίνει ως εξής:

Αλγόριθμος δοκιμή

Για i από 1 μέχρι 100

Γράψε 'Καλημέρα'

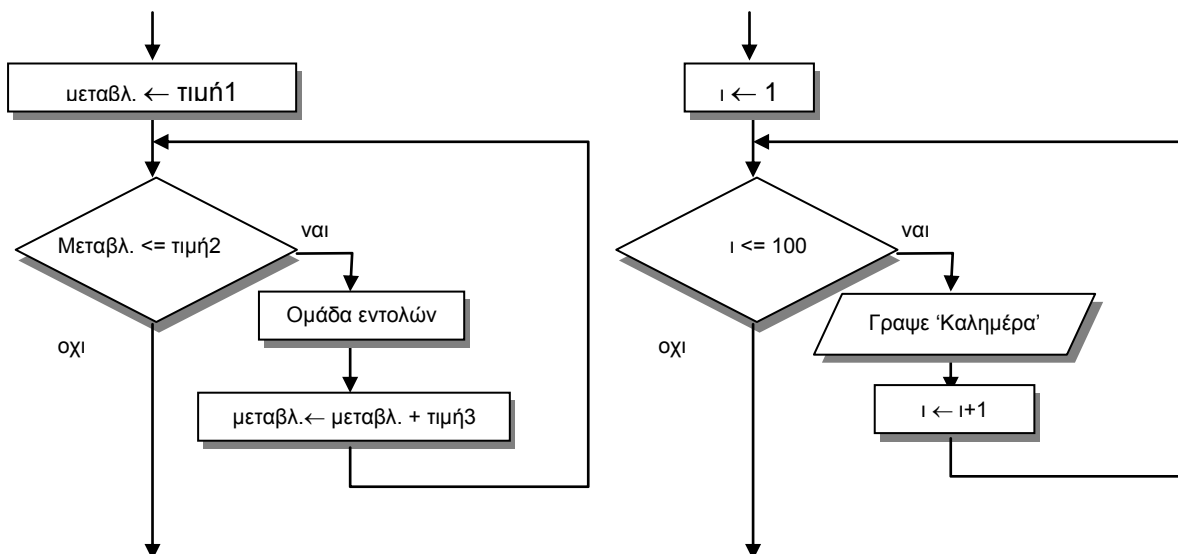
Τέλος_επανάληψης

Τέλος_δοκιμή

- Η εντολή **για** χρειάζεται για να δουλέψει μια μεταβλητή η οποία θα παίζει το ρόλο του μετρητή και θα μετράει πόσες επαναλήψεις έχουν γίνει. Ο μετρητής αυτός ξεκινά από τη **τιμή1** που ορίζουμε στην εντολή και αυξάνεται μετά από κάθε επανάληψη τόσο όσο ορίζεται από την **τιμή3** (βήμα). Όταν η τιμή της μεταβλητής ξεπεράσει το όριο **τιμή2** τότε η επανάληψη σταματά και συνεχίζουμε την εκτέλεση του αλγορίθμου στην επόμενη εντολή της **για**.

- Σε κάθε επανάληψη εκτελούνται όλες οι εντολές που βρίσκονται ανάμεσα από την **για** και το **τέλος_επανάληψης**.

Η γενική μορφή της εντολής και ο προηγούμενος αλγόριθμος σε διάγραμμα ροής έχουν την παρακάτω μορφή.



- Η εντολή ' $i \leftarrow 1$ ' και ' $i \leftarrow i+1$ ' δεν εμφανίζονται πουθενά όταν γράφουμε την εντολή σε κωδικοποίηση. Παρ' όλα αυτά οι δύο εντολές εκτελούνται κρυφά από την εντολή **για**. Αυτό πρέπει να το έχουμε υπ' όψιν μας ειδικά στην περίπτωση μετατροπής αλγορίθμου από κωδικοποίηση σε διάγραμμα ροής και αντίστροφα καθώς και στη περίπτωση που μας ζητάνε να εκτελέσουμε έναν αλγόριθμο που περιέχει την εντολή '**για**'.

- Στην περίπτωση που η πρόταση '**με βήμα** τιμή3' παραληφθεί υπονοείται ότι το βήμα αύξησης του μετρητή είναι 1. Το βήμα αύξησης συνήθως είναι θετικό και η τιμή1 είναι μικρότερη της τιμή2.

- Το βήμα αύξησης μπορεί να είναι και αρνητικό αρκεί $τιμή1 \geq τιμή2$.

- Η μεταβλητή που χρησιμοποιούμε στην **για** πρέπει να είναι ακέραια.

- Ο αριθμός των επαναλήψεων πρέπει να είναι γνωστός και να εκφράζεται είτε με μια σταθερά είτε με μια μεταβλητή της οποίας η τιμή είναι γνωστή (π.χ. τη διαβάζουμε πριν την επανάληψη). Αν για παράδειγμα στην προηγούμενη άσκηση ο αριθμός των επαναλήψεων θέλαμε να ορίζεται από το χρήστη θα κάναμε τα εξής:

Αλγόριθμος δοκιμή

Διάβασε N

Για i από 1 μέχρι N

Γράψε 'Καλημέρα' (το μήνυμα θα τυπωθεί N φορές)

Τέλος_επανάληψης

Τέλος_δοκιμή

Μπορεί τη στιγμή που σχεδιάζουμε τον αλγόριθμο να μη γνωρίζουμε το N, όμως θα πάρει μια συγκεκριμένη τιμή πριν από την εκτέλεση της επανάληψης και η εντολή για θα γνωρίζει πόσες φορές πρέπει να επαναληφθεί.

- Αν οι τιμές του μετρητή δεν μας ενδιαφέρουν, αλλά μας ενδιαφέρει μόνο να γίνει ένα σύνολο επαναλήψεων, τότε η αρχική και τελική τιμή του μετρητή μπορεί να είναι οποιαδήποτε. Δηλαδή αν θέλουμε δέκα επαναλήψεις, αντί να πούμε

Για i από 1 μέχρι 10 (δέκα επαναλήψεις)

Θα μπορούσαμε να πούμε

Για i από 11 μέχρι 20 ή **Για i από -1 μέχρι 0** ή **Για i από 5 μέχρι 14** και γενικότερα, να χρησιμοποιήσουμε οποιοδήποτε ζεύγος αρχικής και τελικής τιμής που διαφέρουν μεταξύ τους κατά 9 μονάδες

- Αν μας ενδιαφέρουν οι τιμές που θα πάρει ο μετρητής γιατί θέλουμε να τις χρησιμοποιήσουμε, τότε πρέπει να είμαστε προσεκτικοί στον ορισμό της αρχικής και τελικής τιμής καθώς και στο βήμα αλλαγής του μετρητή.

Έστω η παρακάτω άσκηση:

Να γίνει αλγόριθμος ο οποίος θα τυπώνει τους αριθμούς 1 έως 100.

Σε αυτή τη περίπτωση αντί να γράψουμε 100 φορές την εντολή **γράψε** ως εξής:

Γράψε 1

Γράψε 2

Γράψε 3

.

.

.

Γράψε 100

Θα μπορούσαμε να χρησιμοποιήσουμε την εντολή **για** ώστε να επαναλάβουμε αυτόματα την εντολή **γράψε** χωρίς κόπο.

Για i από 1 μέχρι 100 με βήμα 1

Γράψε i

Τέλος_επανάληψεων

Όπως φαίνεται στο παραπάνω τμήμα αλγορίθμου, σε κάθε επανάληψη τυπώνεται η τρέχουσα τιμή του i . Επειδή όμως το i παίρνει διαδοχικά τις τιμές 1 έως 100, θα τυπωθούν τελικά αυτές οι τιμές, όπως απαιτεί η εκφώνηση της άσκησης. Σε μια τέτοια περίπτωση δεν θα μπορούσαμε να βάλουμε άλλη αρχική και τελική τιμή για το i , διότι δεν θα μας βόλευε στην εκτύπωση των αποτελεσμάτων.

Παράδειγμα:

Να γίνει αλγόριθμος ο οποίος θα τυπώνει τους ζυγούς αριθμούς από το 1 μέχρι το N . Όπου N ένας αριθμός που θα δίνει ο χρήστης.

Λύση

Αλγόριθμος Ζυγοί

Γράψε 'Δώσε έναν ακέραιο αριθμό'

Διάβασε N

Για i από 2 μέχρι N με βήμα 2

Γράψε i

Τέλος_επανάληψης

Τέλος_Ζυγοί

Βάζοντας βήμα 2 στην εντολή **για** αναγκάζουμε το i να αυξάνει μετά από κάθε επανάληψη κατά δύο μονάδες. Έτσι ενώ η αρχική τιμή του i είναι 2, η επόμενη είναι η 4, η 6, η 8 κ.ο.κ. περνώντας μόνο από ζυγούς αριθμούς.

Εντολή 'όσο επανέλαβε' και 'μέχρις ότου'

Η εντολή **'για'** είναι χρήσιμη λόγω της απλότητάς της μόνο στις περιπτώσεις όπου ο αριθμός των επαναλήψεων είναι γνωστός. Η επανάληψη τερματίζει όταν ο μετρητής, που μετρά τις επαναλήψεις, ξεπεράσει το επιθυμητό όριο.

Σε περιπτώσεις όπου ο αριθμός των επαναλήψεων δεν είναι γνωστός εκ των προτέρων, αλλά είναι γνωστό το γεγονός (η συνθήκη) που θα τερματίσει την επανάληψη, χρησιμοποιούμε μια από τις δύο γενικότερες εντολές επανάληψης που είναι οι 'όσο επανέλαβε' και 'μέχρις_ότου'.

Όλες οι αλγόριθμοι που υλοποιούνται με τη χρήση της **για** μπορούν να λυθούν επίσης με την **όσο** ή την **μέχρις_ότου**, αλλά δεν συμβαίνει και το αντίστροφο.

Η Εντολή 'όσο επανέλαβε'(39-41,173-175)

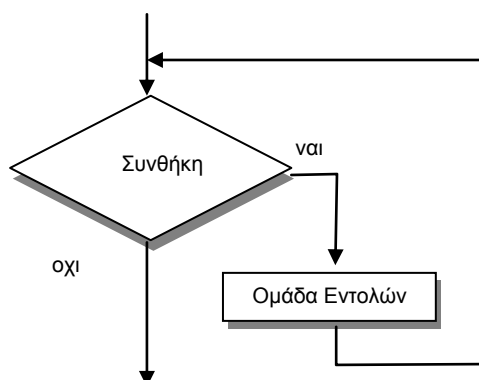
Σύνταξη

Όσο <συνθήκη> επανέλαβε

Ομάδα εντολών

Τέλος_επανάληψης

Διάγραμμα Ροής



Το σκεπτικό της **όσο** είναι: **Όσο** η **συνθήκη** της εντολής είναι **αληθής** επανέλαβε τις εντολές που βρίσκονται στο εσωτερικό της. Όταν η συνθήκη γίνει **ψευδής**, σταμάτα την επανάληψη.

Ο αλγόριθμος εκτύπωσης των αριθμών 1 έως 100 που είδαμε με την εντολή **για**, θα γινόταν με την **όσο** ως εξής:

Αλγόριθμος δοκιμή

$i \leftarrow 1$

όσο ($i \leq 100$) **επανάλαβε**

Γράψε i

$i \leftarrow i + 1$

Τέλος_επανάληψης

Τέλος_δοκιμή

- Η αρχικοποίηση του i σε 1 και η αύξηση του i κατά ένα μετά από κάθε επανάληψη, γίνοντουσαν αυτόματα στην εντολή **για**. Στην εντολή **όσο** είμαστε αναγκασμένοι να τοποθετήσουμε οι ίδιοι τις εντολές αυτές.

- Κατά την σύνταξη μιας εντολής **όσο**, πρέπει να είμαστε πολύ προσεκτικοί για να μην καταλήξουμε σε **ατέρμονο βρόγχο** (ασταμάτητο κύκλο). Αν γράψουμε λάθος συνθήκη ή δεν φροντίσουμε η συνθήκη μας να γίνεται κάποια στιγμή **ψευδής**, τότε η επανάληψη δεν πρόκειται να σταματήσει ποτέ και φυσικά δεν πρόκειται να πάρουμε αποτέλεσμα από τον αλγόριθμο. Αν στην παραπάνω άσκηση δεν βάζαμε την εντολή ' $i \leftarrow i + 1$ ' τότε η συνθήκη ($i \leq 100$) θα ήταν πάντα αληθής, αφού το i δεν θα άλλαζε και θα ήταν πάντα 1.

- Οι εντολές στο εσωτερικό της εντολής **όσο** δεν είναι υποχρεωτικό να εκτελεστούν. Αν κατά τον πρώτο έλεγχο της συνθήκης δούμε ότι είναι **ψευδής** τότε η επανάληψη θα τερματιστεί χωρίς να έχουμε εκτελέσει τις εντολές ούτε μια φορά.

Η Εντολή 'μέχρις ότου'(42,175-178)

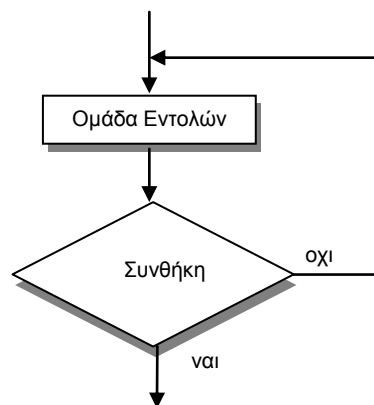
Σύνταξη

Αρχή_επανάληψης

Ομάδα εντολών

Μέχρις_ότου <Συνθήκη>

Διάγραμμα Ροής



Το σκεπτικό της **μέχρις_ότου** είναι: **επανάλαβε** τις εντολές **μέχρις ότου** η **συνθήκη** της εντολής να γίνει **αληθής**. Όταν η συνθήκη γίνει **αληθής**, σταμάτα την επανάληψη.

Ο αλγόριθμος εκτύπωσης των αριθμών 1 έως 100 που είδαμε με την εντολή **για**, θα γινόταν με την **μέχρις ότου** ως εξής:

Αλγόριθμος δοκιμή

$i \leftarrow 1$

αρχή_επανάληψης

Γράψε i

$i \leftarrow i + 1$

μέχρις_ότου ($i > 100$)

Τέλος_δοκιμή

- Η αρχικοποίηση του i σε 1 και η αύξηση του i κατά ένα μετά από κάθε επανάληψη, γίνοντουσαν αυτόματα στην εντολή **για**. Στην εντολή **μέχρις_ότου** είμαστε αναγκασμένοι να τοποθετήσουμε οι ίδιοι τις εντολές αυτές, όπως και στην **όσο**.

- Όπως και με την **όσο**, πρέπει να είμαστε ιδιαίτερα προσεκτικοί στη σύνταξη της εντολής, έτσι ώστε να μην καταλήξουμε σε **ατέρμονο βρόγχο** (ασταμάτητο κύκλο).

- Οι εντολές στο εσωτερικό της εντολής **μέχρις_ότου** θα εκτελεστούν οπωσδήποτε μια φορά, αφού ο έλεγχος για το τερματισμό της εντολής έπεται των εντολών.

Παράδειγμα:

Να γίνει αλγόριθμος ο οποίος θα διαβάζει ακέραιες τιμές μέχρις ότου δοθεί το 0 και θα μετρά πόσοι από αυτούς είναι θετικοί και πόσοι αρνητικοί.

Σε μια τέτοια περίπτωση δεν μπορούμε να προβλέψουμε πόσοι αριθμοί θα διαβαστούν γιατί αυτό εξαρτάται από την κρίση του χρήστη (πότε αυτός θα δώσει το 0), οπότε δεν είμαστε σε θέση να χρησιμοποιήσουμε τη **'για'**. Θα δούμε τον αλγόριθμο με την εντολή **όσο** και την εντολή **μέχρις_ότου**.

Αλγόριθμος Θετικοί_Αρνητικοί

$A \leftarrow 0$

$\Theta \leftarrow 0$

Διάβασε X

Όσο ($X \neq 0$) **επανάλαβε**

Αν ($X > 0$) **τότε**

$\Theta \leftarrow \Theta + 1$

Αλλιώς

$A \leftarrow A+1$
Τέλος_αν
Διάβασε X
Τέλος_επανάληψης

Γράψε 'Οι αρνητικοί αριθμοί είναι ',A
Γράψε 'Οι θετικοί αριθμοί είναι ', Θ
Τέλος_Θετικοί_Αρνητικοί

Αλγόριθμος Θετικοί_Αρνητικοί

$A \leftarrow 0$
 $\Theta \leftarrow 0$
Διάβασε X
Αρχή_επανάληψης
Αν (X>0) **τότε**
 $\Theta \leftarrow \Theta+1$
Αλλιώς
 $A \leftarrow A+1$
Τέλος_αν
Διάβασε X
Μέχρις_ότου (X=0)

Γράψε 'Οι αρνητικοί αριθμοί είναι ',A
Γράψε 'Οι θετικοί αριθμοί είναι ', Θ
Τέλος_Θετικοί_Αρνητικοί

Πολλαπλασιασμός αλά ρωσικά (σελ 45)

Στη σελίδα 45 του βιβλίου παρουσιάζεται ένας εναλλακτικός τρόπος πολλαπλασιασμού ο οποίος χρησιμοποιείται από τον υπολογιστή. Σύμφωνα με αυτή τη μέθοδο βάζουμε τους δύο αριθμούς που θέλουμε να πολλαπλασιάσουμε, έστω X και Y, τον έναν δίπλα στον άλλο. Διαδοχικά διπλασιάζουμε τον X, ενώ υποδιπλασιάζουμε τον Y, αγνοώντας τα δεκαδικά που μπορεί να προκύψουν, δηλαδή υπολογίζουμε το $Y \text{ div } 2$. Αν οι αριθμοί μας ήταν ο 15 και ο 25 τότε η στήλες που θα προέκυπταν από τις διαδοχικές αυτές πράξεις θα ήταν οι παρακάτω.

	X	Y	
	15	25	
$15*2=$	30	12	$= 25 \text{ div } 2$
$30*2=$	60	6	$= 12 \text{ div } 2$
$60*2=$	120	3	$= 6 \text{ div } 2$
$120*2=$	240	1	$= 3 \text{ div } 2$

Η διαδικασία διπλασιασμού και υποδιπλασιασμού σταματά όταν ο δεύτερος αριθμός (Y) γίνει ίσος με τη μονάδα. Το γινόμενο των αριθμών προκύπτει αν αθροίσουμε τις τιμές της πρώτης στήλης που έχουν στην αντίστοιχη θέση της δεύτερης στήλης περιττό αριθμό. Αυτές οι τιμές είναι οι:

	15	25
	120	3
	240	1
$15+120+240 =$	375	

Οπότε το γινόμενο των αριθμών είναι 375.

Έστω ότι μας ζητούσαν να υλοποιήσουμε έναν αλγόριθμο που θα εκτελεί τον πολλαπλασιασμό αλά ρωσικά τότε:

- Ο αλγόριθμος θα πρέπει να διαβάσει του δύο αριθμούς X και Y και να επιστρέφει το γινόμενό τους σε μια μεταβλητή, έστω Γ.
- Θα πρέπει επαναληπτικά να διπλασιάζει και να υποδιπλασιάζει τους αριθμούς X και Y, μέχρις ότου το Y γίνει 1.
- Αν μετά από έναν υποδιπλασιασμό του Y προκύψει περιττός αριθμός τότε η αντίστοιχη τιμή του X θα πρέπει να προστίθεται σε ένα άθροισμα (στο Γ).

Αλγόριθμος πολλαπλασιασμός_αλά_ρωσικά

Διάβασε X,Y

$\Gamma \leftarrow 0$

Όσο (Y>0) **επανάλαβε**

Αν (Y mod 2=1) **τότε**

$\Gamma \leftarrow \Gamma + X$

Τέλος_αν

$X \leftarrow X * 2$

$Y \leftarrow Y \text{ div } 2$

Τέλος_επανάληψης

Γράψε 'Το γινόμενο είναι ' ,Γ

Τέλος_πολλαπλασιασμός_αλά_ρωσικά

Ο ίδιος αλγόριθμος μπορεί να υλοποιηθεί και διαφορετικά χρησιμοποιώντας για παράδειγμα την εντολή '**μέχρις_ότου**' ως εξής:

Αλγόριθμος πολλαπλασιασμός_αλά_ρωσικά

Διάβασε X,Y

$\Gamma \leftarrow 0$

Αρχή_επανάληψης

Αν (Y mod 2=1) **τότε**

$\Gamma \leftarrow \Gamma + X$

Τέλος_αν
 $X \leftarrow X * 2$
 $Y \leftarrow Y \text{ div } 2$
Μέχρις_ότου ($Y=0$)

Γράψε 'Το γινόμενο είναι ',Γ
Τέλος_πολλαπλασιασμός_αλά_ρωσικά

Ολίσθηση (σελ 45)

Στους προγραμματιστές και στους χρήστες υπολογιστών οι αριθμοί παρουσιάζονται με την συνηθισμένη τους μορφή (ακέραιοι, πραγματικοί, θετικοί ή αρνητικοί). Για την αναπαράσταση, την αποθήκευση και την επεξεργασία των αριθμών από τον υπολογιστή χρησιμοποιείται, ανάλογα με τη περίπτωση, κάποια άλλη αναπαράσταση που δεν είναι ορατή σ' εμάς. Πάντα χρησιμοποιούνται τα δυαδικά ψηφία 0 και 1. Ο κατάλληλος συνδυασμός αυτών των ψηφίων μπορεί να οδηγήσει στην αναπαράσταση όλων των αριθμών. Για παράδειγμα ο αριθμός 25 στο δεκαδικό σύστημα αρίθμησης αναπαριστάται στον υπολογιστή με το δυαδικό σύστημα και είναι ο αριθμός 00011001 (στο δεκαδικό έχουμε $25=2$ δεκάδες και 5 μονάδες, ενώ στο δυαδικό έχουμε $25=1$ δεκαεξάδα και 1 οκτάδα και 1 μονάδα). Πάνω σε αριθμούς αυτής της μορφής μπορούμε να εκτελέσουμε μια πράξη που ονομάζεται **ολίσθηση**. Στην ολίσθηση μετακινούμε όλα τα ψηφία ενός αριθμού προς τη μια ή την άλλη πλευρά (ολίσθηση προς τα δεξιά ή ολίσθηση προς τα αριστερά). Επειδή όμως το πλήθος των ψηφίων που θα χρησιμοποιήσουμε είναι καθορισμένο (8), είμαστε αναγκασμένοι να «πετάξουμε» κάποια ψηφία και να αντικαταστήσουμε με 0. Έστω ο αριθμός 25 στη δυαδική του μορφή. Οι θέσεις που μπορούμε να χρησιμοποιήσουμε είναι 8.

0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Αν ολισθήσουμε τον αριθμό αριστερά κατά μια θέση τότε το αριστερότερο 0 θα χαθεί και η κενή θέση που θα προκύψει στα δεξιά θα συμπληρωθεί με ένα 0, ως εξής:

← εξαγωγή	0	<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	1	1	0	0	1	0	← εισαγωγή
0	0	1	1	0	0	1	0				

Στις 8 θέσεις έχει δημιουργηθεί τώρα ένας νέος αριθμός, που είναι ο αριθμός 50, που είναι διπλάσιος του αρχικού 25. **Άρα η ολίσθηση προς τα αριστερά κατά 1 θέση ισοδυναμεί με πολλαπλασιασμό του αριθμού επί 2.**

Αν στον αρχικό αριθμό εκτελούσαμε ολίσθηση προς τα δεξιά κατά μια θέση, τότε θα χάναμε το δεξιότερο 1 και στην κενή θέση στα αριστερά θα έμπαινε ένα καινούργιο 0 ως εξής:

εισαγωγή →	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	0	0	0	1	1	0	0	1	→ εξαγωγή
0	0	0	0	1	1	0	0				

Ο αριθμός που έχει σχηματιστεί μέσα στις οκτώ θέσεις είναι ο αριθμός 12, που ισοδυναμεί με το υποδιπλάσιο του αρχικού αριθμού 25, αγνοώντας το δεκαδικό μέρος. **Άρα η ολίσθηση προς τα δεξιά κατά μια θέση δίνει το ακέραιο πηλίκο της διαίρεσης του αριθμού με το 2 ($25 \text{ div } 2$).**

Πίνακες Παρακολούθησης Τιμών

Οι πίνακες παρακολούθησης είναι μια μέθοδος που μας επιτρέπει να εκτελούμε εντολή προς εντολή έναν αλγόριθμο(συνήθως σε ψευδοκώδικα). Στόχος αυτής της εκτέλεσης είναι η εξαγωγή αποτελεσμάτων για κάποια **συγκεκριμένα δεδομένα**. Πολλές φορές μπορούμε από τον πίνακα παρακολούθησης να βγάλουμε συμπεράσματα για την πληρότητα ενός αλγορίθμου και εφόσον χρειάζεται, να τον διορθώσουμε.

Η κατασκευή ενός πίνακα παρακολούθησης τιμών είναι απαραίτητη όταν μας ζητούν να βρούμε τα αποτελέσματα ενός αλγορίθμου(η ενός τμήματός του) μετά την εκτέλεσή του για συγκεκριμένα δεδομένα.

Πρώτη μας κίνηση για την παρακολούθηση είναι να αριθμήσουμε τις εντολές που εκτελούνται. Οι εκτελέσιμες εντολές είναι:

- 1) Η εντολή εισόδου "**Διάβασε**". Με την εντολή αυτή επιφέρεται αλλαγή στην τιμή μιας μεταβλητής.
- 2) Η εντολή εξόδου "**Γράψε**" ή "**Εμφάνισε**". Με την εντολή αυτή δεν γίνεται αλλαγή στις τιμές των μεταβλητών του αλγορίθμου.
- 3) Η εντολή απόδοσης τιμής (ή εντολή εκχώρησης) "<".
- 4) Από την **απλή** και **σύνθετη επιλογή** το μόνο τμήμα που αριθμείται είναι η αρχική πρόταση που περιλαμβάνει την λογική **συνθήκη(αν <λογική συνθήκη> τότε)**. Μας ενδιαφέρει η αρίθμηση της εντολής αυτής έτσι ώστε να αποφανθούμε από το αποτέλεσμα της συνθήκης για το ποια είναι η επόμενη προς εκτέλεση εντολή. Φυσικά αριθμούνται ανάλογα και οι εντολές που περιλαμβάνονται στις προτάσεις "**τότε**" και "**αλλιώς**".
- 5) Στις εντολές επανάληψης πρέπει να αριθμήσουμε και να παρακολουθήσουμε όλες εκείνες τις προτάσεις που περιλαμβάνουν **λογική συνθήκη**. Έτσι έχουμε:
 - 5.1) Στην εντολή "**όσο**" αριθμούμε μόνο την πρώτη πρόταση, δηλαδή την "**όσο <λογική συνθήκη> επανέλαβε**". Χειριζόμαστε την λογική συνθήκη όπως και στην επιλογή γιατί το αποτέλεσμά της καθορίζει το επόμενο βήμα εκτέλεσης(αν θα συνεχιστεί η επανάληψη ή θα τερματιστεί).
 - 5.2) Στην εντολή "**μέχρις ότου**" αριθμούμε την **πρόταση "μέχρις ότου <λογική συνθήκη> επανέλαβε"**. Το αποτέλεσμά της θα καθορίσει την έκβαση της επανάληψης.
 - 5.3) Στην εντολή "**για**" αριθμούμε την πρόταση "**για <μτ>από <ατ> μέχρι <ττ> με βήμα <μετ> κάνε**". Αυτή εντολή μπορεί να μην έχει φαινομενικά λογική συνθήκη. Λογική συνθήκη όμως υπάρχει και πρέπει να ελεγχθεί. Ομοίως στην εντολή αυτή υπάρχει κρυμμένη και η εντολή μεταβολής της μεταβλητής "μετρητή". Έτσι όταν φτάνουμε να εκτελέσουμε την εντολή "για" πρέπει να εκτελέσουμε την εντολή **μτ:=μτ+μετ** και να ελέγξουμε την συνθήκη **μτ<=ττ**. Αυτή είναι η λογική συνθήκη στην περίπτωση που **μετ>0**. Στην αντίθετη περίπτωση η λογική συνθήκη είναι **μτ>=ττ**. Όταν η εκάστοτε συνθήκη είναι αληθής συνεχίζουμε την επανάληψη. Όταν αυτές οι συνθήκες δεν ισχύουν σταματάμε την επανάληψη. Για παράδειγμα στην εντολή

για i από 1 μέχρι 10 με βήμα 2 κάνε
η λογική συνθήκη που πρέπει να ελέγξουμε είναι $i \leq 10$. Αν ισχύει θα συνεχίσουμε την επανάληψη, αν όχι σταματάμε. Αντίθετα στην εντολή

για i από 10 μέχρι 1 με βήμα -3 κάνε

η λογική συνθήκη είναι $i \geq 1$. Αν ισχύει τότε συνεχίζουμε την επανάληψη, αλλιώς σταματάμε.

Αφού αριθμήσουμε τις εκτελέσιμες εντολές(βήματα), πρέπει να καθορίσουμε ποιες από τις μεταβλητές του αλγορίθμου πρέπει να παρακολουθήσουμε. Συνήθως δεν χρειάζεται να παρακολουθήσουμε τα δεδομένα εφόσον αυτά δεν αλλάζουν τιμή κατά την εκτέλεση του αλγορίθμου. Παρόλα αυτά αφιερώνουμε μια στήλη και για αυτά έτσι ώστε να γνωρίζουμε τις τιμές τους. Αν μια μεταβλητή εισόδου, είναι ταυτόχρονα και μεταβλητή εξόδου τότε πρέπει να την παρακολουθήσουμε. Τα πράγματα περιπλέκονται όταν στο πρόβλημά μας χρησιμοποιείται πίνακας. Αν ο πίνακας είναι δεδομένος και δεν αλλάζουν τα περιεχόμενά του κατά την εκτέλεση του αλγορίθμου, δεν χρειάζεται να αφιερώσουμε χώρο στον πίνακα

παρακολούθησης για αυτόν. Αν όμως τα περιεχόμενα του πίνακα αλλάζουν τότε είμαστε υποχρεωμένοι να παρακολουθήσουμε αυτές αλλαγές, όπως για παράδειγμα σε πίνακα παρακολούθησης για τον αλγόριθμο ταξινόμησης φυσαλίδας.

Ιδιαίτερη προσοχή χρειάζεται για την παρακολούθηση των συνθηκών. Για αυτό το λόγο σε αλγορίθμους που περιέχουν συνθήκη (επιλογή ή επανάληψη), πρέπει να φτιάξουμε τρεις στήλες. Στην πρώτη θα μπαίνει η έκφραση της συνθήκης, στη δεύτερη το αποτέλεσμα της και στη τρίτη ο αριθμός του βήματος του αλγορίθμου στο οποίο πρέπει να πάμε για να συνεχίσουμε την εκτέλεση.

Επιπλέον μας χρειάζεται μια στήλη με τίτλο «Έξοδος» ή «Εκτύπωση» για την καταγραφή των εξόδων του αλγορίθμου, όπου θα καταγράφονται τα αποτελέσματα των εντολών **γράψε**.

Παράδειγμα:

Το πρώτο παράδειγμα που θα δούμε είναι ο αλγόριθμος υπολογισμού της πέμπτης δύναμης ενός αριθμού a (παράδειγμα σελ. 84). Ο αλγόριθμος μετά την αρίθμηση των εντολών γίνεται:

Αλγόριθμος Πέμπτη_Δύναμη

```

Αρχή
1       $\Pi \leftarrow 1$ 
2       $i \leftarrow 1$ 
3      όσο ( $i \leq 5$ ) επανάλαβε
4           $\Pi \leftarrow \Pi * a$ 
5           $i \leftarrow i + 1$ 
      Τέλος_επανάληψης
6      Γράψε  $\Pi$ 
Τέλος_πέμπτη_δύναμη
    
```

Η παρακολούθηση του αλγορίθμου απαιτεί την παρακολούθηση των μεταβλητών i και Π . Η μεταβλητή a δεν χρειάζεται παρακολούθηση αλλά την σημειώσαμε σε μια στήλη. Είναι σημαντικό να παρακολουθήσουμε και την λογική συνθήκη της εντολής «όσο» για να μπορέσουμε να αποφασίσουμε πότε θα σταματήσει η επανάληψη. Η παρακολούθηση του αλγορίθμου για $a=2$ φαίνεται στον παρακάτω πίνακα.

Βήμα Εκτέλεσης	Βήμα Αλγορίθμου	Μεταβλητές			Λογική Συνθήκη			Έξοδος
		a	i	Π	Έκφραση	Τιμή	Ε.Βήμα	
		2	Απ	Απ				
1	1		Απ	1				
2	2		1	1				
3	3		1	1	$1 \leq 5$	αληθής	4	
4	4		1	2				
5	5		2	2				
6	3		2	2	$2 \leq 5$	αληθής	4	
7	4		2	4				
8	5		3	4				
9	3		3	4	$3 \leq 5$	αληθής	4	
10	4		3	8				
11	5		4	8				
12	3		4	8	$4 \leq 5$	αληθής	4	
13	4		4	16				
14	5		5	16				
15	3		5	16	$5 \leq 5$	αληθής	4	
16	4		5	32				
17	5		6	32				
18	3		6	32	$6 \leq 5$	ψευδής	Τέλος	
19	6		6	32				32

Παράδειγμα:

Το επόμενο παράδειγμα που θα δούμε είναι η παρακολούθηση του μη αναδρομικού αλγορίθμου για την εύρεση του n -οστού όρου της ακολουθίας Fibonacci(σελ.128). Ο αλγόριθμος μετά την αρίθμηση των εντολών είναι:

Αλγόριθμος Fibonacci

Αρχή

1 $X \leftarrow 1$

2 $Y \leftarrow 0$

3 για i από 1 μέχρι n επανέλαβε ! η κρυμμένη λογική συνθήκη είναι ($i \leq n$)

4 $Z \leftarrow X$

5 $X \leftarrow X + Y$

6 $Y \leftarrow Z$

Τέλος_επανάλαβε

7 $\Phi \leftarrow X$

8 Γράψε Φ

Τέλος_fibonacci

Έστω ότι μας ζητούν να υπολογίσουμε τον 5^ο όρο της ακολουθίας Fibonacci. Στον πίνακα παρακολούθησης θα βάλουμε τις μεταβλητές n, i, X, Y, Z, Φ , ενώ θα πρέπει να παρακολουθούμε και την «κρυμμένη» λογική συνθήκη της εντολής «για», που είναι ($i \leq n$). Έτσι προκύπτει ο παρακάτω πίνακας.

Βήμα Εκτέλεσης	Βήμα Αλγορίθμου	Μεταβλητές						Λογική Συνθήκη			Έξοδος
		n	i	X	Y	Z	Φ	Έκφραση	Τιμή	Ε.Βήμα	
		5	Απ	Απ	Απ	Απ	Απ				
1	1		Απ	1	Απ	Απ	Απ				
2	2		Απ	1	0	Απ	Απ				
3	3		1	1	0	Απ	Απ	$1 \leq 5$	αληθής	4	
4	4		1	1	0	1	Απ				
5	5		1	1	0	1	Απ				
6	6		1	1	1	1	Απ				
7	3		2	1	1	1	Απ	$2 \leq 5$	αληθής	4	
8	4		2	1	1	1	Απ				
9	5		2	2	1	1	Απ				
10	6		2	2	1	1	Απ				
11	3		3	2	1	1	Απ	$3 \leq 5$	αληθής	4	
12	4		3	2	1	2	Απ				
13	5		3	3	1	2	Απ				
14	6		3	3	2	2	Απ				
15	3		4	3	2	2	Απ	$4 \leq 5$	αληθής	4	
16	4		4	3	2	3	Απ				
17	5		4	5	2	3	Απ				
18	6		4	5	3	3	Απ				
19	3		5	5	3	3	Απ	$5 \leq 5$	αληθής	4	
20	4		5	5	3	5	Απ				

21	5		5	8	3	5	Απ				
22	6		5	8	5	5	Απ				
23	3		6	8	5	5	Απ	6<=5	ψευδής	7	
24	7		6	8	5	5	8				
25	8		6	8	5	5	8				8

Ασκήσεις Επαναληπτικής Δομής

- 1) Φτιάξτε ένα πρόγραμμα που να τυπώνει του αριθμούς από 1 έως 10.
- 2) Φτιάξτε ένα πρόγραμμα που να τυπώνει τους αριθμούς από 1 έως n
- 3) Φτιάξτε ένα πρόγραμμα το οποίο:
 - α) θα διαβάσει n αριθμούς
 - β) θα τυπώνει τους ίδιους, τα τετράγωνά τους και τις ρίζες τους, αν είναι θετικοί.
- 4) Φτιάξτε ένα πρόγραμμα το οποίο θα βρίσκει το μέγιστο και τον ελάχιστο από n αριθμούς
- 5) Να γίνει πρόγραμμα το οποίο θα διαβάσει το τζίρο ενός εμπόρου για κάθε μια από τις 365 μέρες του έτους και θα βρίσκει τα συνολικά έσοδα.
- 6) Να γίνει πρόγραμμα το οποίο θα διαβάσει τους βαθμούς 150 μαθητών σε ένα μάθημα και θα υπολογίζει και θα τυπώνει τη μέση βαθμολογία τους.
- 7) Να γίνει πρόγραμμα το οποίο θα υπολογίζει το γινόμενο 20 αριθμών που θα δίνει ο χρήστης.
- 8) Να γραφτεί αλγόριθμος που να υπολογίζει το γινόμενο $P=1*2*3*...*n$ (παραγοντικό) για έναν δεδομένο αριθμό n .
- 9) Να γίνει πρόγραμμα το οποίο θα διαβάσει N θερμοκρασίες και βρίσκει και θα τυπώνει τη μικρότερη θερμοκρασία και ποια μέρα την είχε (αύξοντα αριθμό)
- 10) Να διαβασθούν οι βαθμοί και τα ονόματα 20 μαθητών και να βρεθεί ποιος είναι ο μεγαλύτερος βαθμός, ποιος μαθητής τον έχει και ποια θέση στην αρίθμηση έχει ο μαθητής.
- 11) Να διαβασθούν τα ονόματα και οι 6 προσπάθειες 30 αθλητών στο άλμα εις μήκος και να βρεθεί ο νικητής του αγώνα καθώς και η επίδοσή του.
- 12) Να διαβασθούν 100 ακέραιοι αριθμοί διαφορετικοί από το μηδέν και να βρεθεί το άθροισμα όσων απ' αυτούς είναι ζυγοί και το γινόμενο όσων απ' αυτούς διαιρούνται με το 3.
- 13) Να υπολογιστούν τα παρακάτω αθροίσματα για έναν αριθμό n
 - a) $S = 1^2+2^2+3^2+...+n^2$
 - b) $S = 1/2 + 1/4 + 1/6 + ... + 1/N$
 - c) $S = 1 + 1/3 + 1/5 + 1/7 + ... + 1/N$
 - d) $S = 1 + 1/4 + 1/7 + 1/10 + ... + 1/N$
 - e) $S = 1-2+3-4+5-...-(2n)+(2n+1)$
 - f) $S = 2^{1/4} + 4^{1/8} + 8^{1/16} + ... + n^{1/(2n)}$
 - g) $S = 1/1+1/2+1/4+1/8+1/16+1/32+1/n$
- 14) Η ακολουθία Fibonacci είναι ένα σύνολο αριθμών που συμβολίζονται με F_i , όπου i ο αύξων αριθμός του όρου ($5^{ος}$, $7^{ος}$, $10^{ος}$ κ.ο.κ.). Οι όροι της ακολουθίας δίνονται από τον παρακάτω τύπο
$$F_1=1$$
$$F_2=2$$
$$F_n=F_{n-1}+F_{n-2}$$

Να γίνει πρόγραμμα που να υπολογίζει τον n -οστό όρο της ακολουθίας (όπου n τιμή που θα δίνει ο χρήστης)

- 15) Να γίνει πρόγραμμα το οποίο θα υπολογίζει την τιμή της παρακάτω παράστασης όταν δίνεται το N

$$Y = \frac{1 + 2 + 3 + \dots + N}{1 * 3 * 5 * \dots * (2N + 1)} + N$$

- 16) Να γίνει πρόγραμμα το οποίο θα διαβάζει N βαθμολογίες και βρίσκει και θα τυπώνει το μέσο όρο μόνο των βαθμολογιών που είναι πάνω από 10.
- 17) Να γίνει πρόγραμμα το οποίο θα διαβάζει 50 θερμοκρασίες και θα βρίσκει και θα τυπώνει πόσες είναι θετικές ή 0 και πόσες αρνητικές.
- 18) Να γίνει πρόγραμμα το οποίο θα διαβάζει τα εισοδήματα 50 φορολογούμενων και θα βρίσκει και θα τυπώνει το φόρο του κάθε ένα βάση του παρακάτω πίνακα.

Εισόδημα	Φόρος
0 – 10.000	0%
10.001 – 15.000	10%
15.001 και πάνω	20%

- 19) Να γίνει πρόγραμμα το οποίο θα διαβάζει τις απουσίες και τα ονόματα N μαθητών. Θα τυπώνει για όλους τους μαθητές το όνομά τους και ένα μήνυμα για το αν πέρασε ή κόπηκε, ανάλογα με το αν οι απουσίες τους είναι πάνω από 50.
- 20) Να γίνει πρόγραμμα το οποίο θα διαβάζει τους μισθούς 100 εργαζομένων και θα επιστρέφει το πλήθος των εργαζομένων που έχουν μισθό ίσο ή μικρότερο των 1000 ευρώ και το πλήθος των εργαζομένων με μισθό πάνω των 1000 ευρώ.
- 21) Να γίνει πρόγραμμα που θα βρίσκει πόσοι αριθμοί από το 1 έως το 100 διαιρούνται με το 2 και το 3 αλλά όχι με το 4.
- 22) Ένα στάδιο έχει 30 σειρές με καθίσματα. Η κάτω σειρά έχει 500 θέσεις και η κάθε παραπάνω σειρά έχει 50 θέσεις περισσότερες από την προηγούμενή της. Να γίνει αλγόριθμος σε ΓΛΩΣΣΑ που θα υπολογίζει τις συνολικές θέσεις του σταδίου.
- 23) Να διαβασθούν το όνομα, η τάξη (a, b, c) και ο βαθμός 50 μαθητών ενός σχολείου και να βρεθούν για κάθε τάξη το σύνολο των μαθητών και ο μέσος όρος βαθμολογίας κάθε τάξης.
- 24) Να διαβαστεί ένας ακέραιος αριθμός και να βρεθούν οι διαιρέτες του.
- 25) Να διαβαστεί ένας ακέραιος αριθμός και να βρεθεί το πλήθος των διαιρετών του.
- 26) Η χρέωση των ΙΧ αυτοκινήτων που μετακινούνται μ' ένα οχηματαγωγό πλοίο είναι ανάλογη του μήκους τους ως εξής : έως και 2 μέτρα είναι 3 €, έως και 3 μέτρα είναι 6 € και για παραπάνω από 3 μέτρα είναι 9 €. Να διαβασθούν τα μήκη 10 αυτοκινήτων που μετακινήθηκαν μ' ένα οχηματαγωγό πλοίο και να υπολογισθεί το συνολικό ποσό είσπραξης.
- 27) Να διαβαστεί ένας ακέραιος αριθμός και να βρεθεί αν είναι πρώτος ή όχι.
- 28) Να βρεθούν και να εκτυπωθούν οι 50 πρώτοι “πρώτοι” αριθμοί.
- 29) Να διαβασθούν οι βαθμοί και το φύλο (1=αγόρι, 2=κορίτσι) από 20 μαθητές και να βρεθεί πόσα είναι τα αγόρια, πόσα είναι τα κορίτσια, ποιος είναι ο μέσος όρος των βαθμών των αγοριών, ποιος είναι ο μέσος όρος των βαθμών των κοριτσιών καθώς και ποιος είναι ο συνολικός μέσος όρος των βαθμών όλων των μαθητών.
- 30) Να διαβασθούν δύο ακέραιοι αριθμοί a και b, όπου $a > b$, και να βρεθεί ο μέγιστος κοινός διαιρέτης τους.

- 31) Να διαβασθούν δύο αριθμοί a και b , όπου $a < b$, καθώς και 100 τυχαίοι άλλοι αριθμοί και να βρεθεί το άθροισμα και το πλήθος των αριθμών που είναι μικρότεροι του a , μεγαλύτεροι του b και μεταξύ a και b .
- 32) Σε μια εταιρεία ο ιδιοκτήτης αποφάσισε με τον πρώτο μισθό που θα πάρουν οι υπάλληλοι της εταιρείας σε ευρώ (όταν καταργήθηκαν οι δραχμές), να τους δώσει αύξηση βάση των ετών που εργάζονται στην εταιρεία όπως φαίνεται παρακάτω
- Πάνω από 2 χρόνια 10%
 - Πάνω από 5 χρόνια 15%
 - Πάνω από 10 χρόνια 20%
 - Πάνω από 15 χρόνια 30%

Να γίνει αλγόριθμος που θα διαβάσει το μισθό σε δραχμές και το όνομα 30 υπαλλήλων και θα εμφανίζει το νέο μισθό σε δραχμές και σε ευρώ. Επίσης να εμφανίζει το όνομα του υπαλλήλου που παίρνει την μεγαλύτερη αύξηση.

- 33) Ένα videoclub νοικιάζει στους πελάτες του βιντεοκασέτες τύπου VHS (κωδικός=1) και ταινίες σε DVD (κωδικός=2). Η ελάχιστη χρέωση για κάθε βιντεοκασέτα είναι 2,5 € και για κάθε DVD 3 € και ο μέγιστος χρόνος που μπορεί να κρατήσει ο πελάτης μια βιντεοκασέτα είναι 3 ημέρες, ενώ ένα DVD 4 ημέρες, χωρίς να χρεωθεί με επιπλέον ποσό. Η καθυστέρηση χρεώνεται με 1,5 € για κάθε βιντεοκασέτα και με 1,8 € για κάθε DVD για κάθε ημέρα καθυστέρησης. Να διαβασθούν τα στοιχεία 20 ενοικιάσεων πελατών (ονοματεπώνυμο πελάτη, κωδικός ταινίας (1 ή 2), ημέρες παρακράτησης) και να βρεθούν και να εκτυπωθούν η αναλυτική χρέωση ανά πελάτη καθώς και η συνολική χρέωση και ο μέσος όρος χρέωσης ανά είδος ταινίας.
- 34) Μια τράπεζα δίνει επιτόκιο 3% σε κάθε πελάτη. Αυτό σημαίνει ότι μετά από κάθε χρόνο στο ποσό που έχει καταθέσει προστίθεται το 3% του ποσού (ποσό * 3/100). Ένας πελάτης έχειβάλει στην τράπεζα 1500 € και θέλει να βρει πόσο θα γίνει το ποσό του μετά από 10 χρόνια. Φτιάξτε ένα πρόγραμμα για να τον βοηθήσετε.
- 35) Σε ένα πρόγραμμα περιβαλλοντικής εκπαίδευσης συμμετέχουν 20 σχολεία. Στα πλαίσια αυτού του προγράμματος, εθελοντές μαθητές των σχολείων, που συμμετέχουν στο πρόγραμμα, μαζεύουν ποσότητες τριών υλικών (γυαλί, χαρτί και αλουμίνιο).
Να αναπτύξετε έναν αλγόριθμο, ο οποίος:
- να διαβάσει τις ποσότητες σε κιλά των παραπάνω υλικών που μάζεψαν οι μαθητές σε κάθε σχολείο
 - να υπολογίζει τη συνολική ποσότητα σε κιλά του κάθε υλικού που μάζεψαν οι μαθητές σε όλα τα σχολεία
 - αν η συνολική ποσότητα του χαρτιού που μαζεύτηκε από όλα τα σχολεία είναι λιγότερη των 1000 κιλών, να εμφανίζεται το μήνυμα «**Συγχαρητήρια**». Αν η ποσότητα είναι από 1000 κιλά και πάνω, αλλά λιγότερο από 2000, να εμφανίζεται το μήνυμα «**Δίνεται έπαινος**» και τέλος αν η ποσότητα είναι από 2000 κιλά και πάνω να εμφανίζεται το μήνυμα «**Δίνεται βραβείο**».
4. **Παρατήρηση:** Να θεωρήσετε ότι όλες οι ποσότητες είναι θετικοί αριθμοί.
- 36) Να γίνει πρόγραμμα το οποίο θα διαβάσει για 100 μαθητές τις βαθμολογίες τους σε 10 μαθήματα και θα υπολογίζει και θα τυπώνει το μέσο όρο του κάθε μαθητή.
- 37) Να γίνει αλγόριθμος που θα τυπώνει όλα τα ζευγάρια των ακεραίων X και Y τα οποία ικανοποιούν τη σχέση $3X+Y=8$. Θεωρείστε ότι τα X και Y κυμαίνονται από -100 έως 100 .
- 38) Να γραφεί αλγόριθμος ο οποίος διαβάσει έναν αριθμό σε δραχμές, τον μετατρέπει σε ευρώ και τον εμφανίζει. Η διαδικασία αυτή θα επαναλαμβάνεται μέχρι ο χρήστης να δώσει την τιμή 0. Στο τέλος, θα πρέπει να εμφανίζεται και το πλήθος των αριθμών που διάβασε και μετέτρεψε.
- 39) Να διαβασθεί ένας ακέραιος αριθμός A και να βρεθεί το πλήθος των ψηφίων του.

- 40) Να γραφεί αλγόριθμος ο οποίος θα διαβάσει το όνομα και το βαθμό στα μαθηματικά ενός αγνώστου πλήθους μαθητών. Στη συνέχεια να υπολογίζει:
- Το όνομα και το βαθμό του μαθητή με το μεγαλύτερο βαθμό
 - Το πλήθος των μαθητών που έχουν βαθμό κάτω από τη βάση (< 9.5)
 - Το μέσο όρο των βαθμών των μαθητών
- Ο αλγόριθμος θα τερματίζει όταν δοθεί ως είσοδος αρνητικός αριθμός ή μηδέν.
- 41) Στο υπεραστικό ΚΤΕΛ υπάρχουν πέντε κατηγορίες εισιτηρίων: Πολύτεκνο (Π), Αναπηρικό (Α), Στρατιωτικό (Σ), Φοιτητικό (Φ), και Κανονικό (Κ). Οι δυο πρώτες κατηγορίες πληρώνουν το 50% της αξίας του κανονικού εισιτηρίου. Η τρίτη και τέταρτη κατηγορία έχουν έκπτωση 25%, ενώ η τελευταία κατηγορία πληρώνει ολόκληρη την αξία του εισιτηρίου.
- Να γραφεί αλγόριθμος που για ένα λεωφορείο 50 θέσεων θα διαβάσει το αντίτιμο του ολόκληρου εισιτηρίου της διαδρομής που πάει καθένας από τους επιβάτες που επιβαίνουν σε ένα λεωφορείο και την κατηγορία κάθε επιβάτη και θα:
- θα εμφανίζει για κάθε επιβάτη τα χρήματα που πρέπει να πληρώσει,
 - θα εμφανίζει τα συνολικά χρήματα που θα έχουν εισπραχθεί,
- Ο αλγόριθμος θα τερματίζει όταν δοθεί ως κατηγορία το γράμμα Λ ή όταν το λεωφορείο γεμίσει.
- 42) Να διαβασθούν δύο ακέραιοι αριθμοί A και B, όπου $A > B$, και να βρεθεί το υπόλοιπο της ακέραιας διαίρεσης A/B με συνεχείς αφαιρέσεις και χωρίς να χρησιμοποιηθούν οι τελεστές mod και div.
- 43) Ένας έμπορος καταγράφει τα κέρδη κάθε ημέρας. Να γραφεί πρόγραμμα που να διαβάσει τα καθημερινά κέρδη του και να υπολογίζει μετά από πόσες μέρες έβγαλε 1000 €.
- 44) Ένας μετεωρολογικός σταθμός υπολογίζει κάθε μέρος το ύψος της βροχής που έχει πέσει. Για να βρει το μέσο ύψος βροχής ζητά να γραφτεί πρόγραμμα που να διαβάσει το καθημερινό ύψος βροχόπτωσης μέχρι να δοθεί αρνητική τιμή και να υπολογίζει το μέσο ύψος βροχόπτωσης.
- 45) Φτιάξτε ένα πρόγραμμα που να διαβάσει αριθμούς από το χρήστη μέχρι ο μέσος όρος των αριθμών που διάβασε να ξεπεράσει έναν δεδομένο αριθμό Π και να επιστρέφει το πλήθος των αριθμών που διάβασε.
- 46) Σε έναν κινηματογράφο μπορούμε να πληρώσουμε είτε με συνδρομή (30 εγγραφή και 3 δρχ. ανά ταινία) είτε χωρίς συνδρομή (7 δρχ. ανά ταινία). Βρείτε το ελάχιστο πλήθος των ταινιών X που πρέπει να δούμε σε ένα μήνα ώστε να μας συμφέρει η συνδρομή.
- 47) Σύμφωνα με εγκύκλιο του Υπουργείου Παιδείας, κάθε έτος, σε κάθε Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης συγκροτείται Επιτροπή Συγκέντρωσης Μηχανογραφικών Δελτίων των υποψηφίων για εισαγωγή στην τριτοβάθμια εκπαίδευση. Ο αριθμός των μελών της Επιτροπής είναι 6 άτομα αρχικά και αν ο αριθμός των υποψηφίων είναι μεγαλύτερος από 200 και μέχρι 270 προστίθεται ένα ακόμα άτομο, ενώ αν ο αριθμός των υποψηφίων είναι μεγαλύτερος από 270 και μέχρι 340 προστίθεται ένα ακόμα άτομο κοκ. Να διαβασθεί το πλήθος των υποψηφίων (Υ) σε μια Διεύθυνση Δευτεροβάθμιας Εκπαίδευσης και να βρεθεί και να εκτυπωθεί το πλήθος των μελών της Επιτροπής που θα συγκροτηθεί (ΑΜ).
- 48) Μια τράπεζα δίνει επιτόκιο 3% σε κάθε πελάτη. Αυτό σημαίνει ότι μετά από κάθε 6μηνο στο ποσό που έχει καταθέσει προστίθεται το 3% του ποσού ($\text{ποσό} * 3/100$). Ένας πελάτης έχει βάλει στην τράπεζα 1500 € και θέλει να βρει μετά από πόσα χρόνια το αρχικό του ποσό θα ξεπεράσει τα 3000 €.
- Να γίνει σχετικό πρόγραμμα που να υπολογίζει τα απαιτούμενα χρόνια για επιτόκιο 3%.

β) Να τροποποιήσετε το πρόγραμμα ώστε να υπολογίζει σε πόσα χρόνια το ποσό θα ξεπεράσει τα 3000 €, αν το επιτόκιο αλλάζει κάθε φορά κατά 0.2% (το πρώτο 6μηνο 3%, το δεύτερο 3.2%, το τρίτο 3.4%, 3.6%, 3.8% κ.ο.κ.)

49) Το στατιστικό τμήμα της μετεωρολογικής υπηρεσίας θέλει να δει τι μεταβολές θερμοκρασίας είχαμε κατά τη διάρκεια ενός έτους. Μετρώντας τις θερμοκρασίες μιας τοποθεσίας σε μια συγκεκριμένη ώρα της ημέρας για όλο το χρόνο, παρουσιάζεται από τη μια μέρα στην άλλη είτε αύξηση, είτε μείωση είτε στασιμότητα της θερμοκρασίας.

Να γίνει πρόγραμμα το οποίο:

α) θα διαβάζει τις θερμοκρασίες των 356 ημερών του έτους

β) θα μετρά πόσες φορές έχουμε αύξηση, πόσες μείωση και πόσες στασιμότητα της θερμοκρασίας από τη μια μέρα στην άλλη.

γ) θα υπολογίζει το μέγιστο διάστημα ημερών που έχουμε συνεχή αύξηση της θερμοκρασίας.

50) (Π) Ένας συλλέκτης γραμματοσήμων επισκέπτεται στο διαδίκτυο το αγαπημένο του ηλεκτρονικό κατάστημα φιλοτελισμού προκειμένου να αγοράσει γραμματόσημα. Προτίθεται να ξοδέψει μέχρι 1500 ευρώ.

Να αναπτύξετε αλγόριθμο ο οποίος:

α. Για κάθε γραμματόσημο, να διαβάζει την τιμή και την προέλευσή του (ελληνικό/ξένο) και να επιτρέπει την αγορά του, εφόσον η τιμή του δεν υπερβαίνει το διαθέσιμο υπόλοιπο χρημάτων. Διαφορετικά να τερματίζει τυπώνοντας το μήνυμα «ΤΕΛΟΣ ΑΓΟΡΩΝ».

ΣΗΜΕΙΩΣΗ: Δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου.

β. Να τυπώνει:

1. Το συνολικό ποσό που ξόδεψε ο συλλέκτης.

2. Το πλήθος των ελληνικών και το πλήθος των ξένων γραμματοσήμων που αγόρασε.

3. Το ποσό που περίσσεψε, εφόσον υπάρχει, διαφορετικά το μήνυμα «ΕΞΑΝΤΛΗΘΗΚΕ ΟΛΟ ΤΟ ΠΟΣΟ».

51) Δίνεται το παρακάτω τμήμα αλγορίθμου.

X <- 1

Όσο X < 5 επανέλαβε

A <- X + 2

B <- 3*A - 4

C <- B - A + 4

Αν A > B τότε

Αν A > C τότε

MAX <- A

Αλλιώς

MAX <- C

Τέλος_αν

Αλλιώς

Αν B > C τότε

MAX <- B

Αλλιώς

MAX <- C

Τέλος_αν

Τέλος_αν

Γράψε X,A,B,C,MAX

X <- X + 2

Τέλος_επανάληψης

A) Τι θα εμφανίζει ο αλγόριθμος σε κάθε επανάληψη;

B) κάντε το αντίστοιχο διάγραμμα ροής

52) Αν θεωρήσουμε ότι $A=8$, $B=5$, $\Gamma=4$ και $\Delta=1$, τι θα εμφανίσουν τα παρακάτω τμήματα αλγορίθμου

i	ii
Για M από 2 μέχρι 14 με βήμα Γ κάνε Γράψε M Τέλος_επανάληψης	$K \leftarrow 0$ $\Lambda \leftarrow 0$ Για μετρητής από 3 μέχρι 6 κάνε $K \leftarrow K + \Gamma - \Delta$ $\Lambda \leftarrow \Lambda * B$ Τέλος_επανάληψης Γράψε K, Λ
iii	iv
$K \leftarrow 1$ Για M από 1 μέχρι A με βήμα 2 κάνε $K \leftarrow K * M$ Γράψε M Τέλος_επανάληψης Γράψε K	$Z \leftarrow 5$ Για μετρητής από Γ μέχρι A κάνε $Z \leftarrow Z + \Delta$ Γράψε Z^2 Τέλος_επανάληψης

53) Τα απόμεινα τμήματα αλγορίθμων να γραφούν ξανά χρησιμοποιώντας την επαναληπτική δομή 'όσο ... επανέλαβε'

i	ii
$X \leftarrow 2$ Για Y από 1 μέχρι 5 κάνε $X \leftarrow X + 3$ Τέλος_επανάληψης	$X \leftarrow 2$ Για Y από -1 μέχρι 4 με βήμα 3 κάνε $X \leftarrow X - 2$ Τέλος_επανάληψης
iii	iv
$X \leftarrow 2$ Για Y από 5 μέχρι -5 με βήμα -2 κάνε $X \leftarrow X + 2$ Τέλος_επανάληψης	$X \leftarrow -2$ Για Y από 3 μέχρι 1 με βήμα -1 κάνε $X \leftarrow X - 1$ Τέλος_επανάληψης

54) Τα επόμενα τμήματα να γραφούν ξανά (εφόσον είναι δυνατόν) με την επαναληπτική δομή 'Για ... από ... μέχρι'

i	ii
$X \leftarrow 2$ $Y \leftarrow 3$ όσο $Y \leq 17$ επανέλαβε $X \leftarrow X + 2$ $Y \leftarrow Y + 1$ Τέλος_επανάληψης	$X \leftarrow 2$ $Y \leftarrow 0$ όσο $Y \leq 8$ επανέλαβε $Y \leftarrow Y + 1$ $X \leftarrow X + 2$ Τέλος_επανάληψης
iii	iv
$X \leftarrow 2$ $Y \leftarrow 0$ όσο $Y > 4$ επανέλαβε $Y \leftarrow Y + X$ $X \leftarrow X + 3$ Τέλος_επανάληψης	$W \leftarrow 0$ $X \leftarrow 2$ $Y \leftarrow 0$ όσο $Y < 6$ επανέλαβε $W \leftarrow W - 1$ $Y \leftarrow Y + 1$ $X \leftarrow 2 + W * Y$ Τέλος_επανάληψης
v	vi
$X \leftarrow 2$ $Y \leftarrow 7$ όσο $Y \leq 253$ επανέλαβε $Y \leftarrow Y + 8$ $X \leftarrow X + 1$ Τέλος_επανάληψης	$W \leftarrow 0$ $X \leftarrow 2$ $Y \leftarrow -3$ όσο $Y < 154$ επανέλαβε $W \leftarrow W + 2$ $Y \leftarrow Y + 0.5$ $X \leftarrow Y * (Y + 1) + W$ Τέλος_επανάληψης

- 55) Πόσες φορές θα εκτελεστεί η επανάληψη στο παρακάτω τμήμα αλγορίθμου και τι τιμές θα έχουν στο τέλος οι μεταβλητές αν δοθεί σαν είσοδος i) -2 και ii) 2
Επίσης να γίνει το διάγραμμα ροής του παρακάτω τμήματος αλγορίθμου.

```
Διάβασε X  
Y <- X/2 + 2  
Όσο X>0 και Y>X κάνε  
    X <- X+1  
    Y <- Y-1  
Τέλος_επανάληψης
```

- 56) Τι τιμές θα έχουν οι μεταβλητές μετά την εκτέλεση του παρακάτω τμήματος αλγορίθμου για είσοδο i) -1 ii) 7 .
Επίσης να γίνει το διάγραμμα ροής του παρακάτω τμήματος αλγορίθμου.

```
Διάβασε X  
Y <- -2+(X+1)/2  
Αρχή_επανάληψης  
    Αν X>Y τότε  
        Y <- Y-1  
        X <- X-2  
    Αλλιώς  
        Y <- Y-2  
        X <- X+1  
    Τέλος_αν  
Μέχρις_ότου (X>Y και Y<0)
```

- 57) Πόσες φορές θα εκτελεστεί η επανάληψη στο παρακάτω τμήμα αλγορίθμου και τι τιμές θα έχουν στο τέλος οι μεταβλητές αν δοθεί σαν είσοδος i) 2 και ii) -2

```
Διάβασε X  
E <- X  
Για Y από -1 έως X κάνε  
    Αν Y=1 τότε  
        E <- E + Y + 3  
    Αλλιώς  
        E <- E + Y  
    Τέλος_αν  
Τέλος_επανάληψης
```

- 58) Δίνεται ο παρακάτω αλγόριθμος

```
Αλγόριθμος Θέμα_2  
X <- 3  
Γιά i από 5 μέχρι 0 με βήμα -2  
    X <- X + 1  
    Αν (X > 4) τότε  
        Y <- X  
    Αλλιώς  
        Y <- -X  
    Τέλος Αν  
    Z <- X * Y  
Τέλος Για  
Τέλος Θέμα_2
```

Να βρείτε τις τιμές των μεταβλητών i, X, Y, Z σε όλες τις επαναλήψεις

59) Δίνεται ο παρακάτω αλγόριθμος

```
Αλγόριθμος Θέμα_2
Διάβασε X
Αν X mod 2 = 0 τότε
    Για i από 1 μέχρι 3
        X <- X + 3
    Τέλος_Για
Τέλος_αν
Αν X mod 2 <> 0 τότε
    Για i από 5 μέχρι 3 με_βήμα -1
        X <- X + 1
    Τέλος_για
Τέλος_αν
Τέλος Θέμα_2
```

- a) Να βρείτε την τιμή της μεταβλητής X μετά την εκτέλεση του για $X = 1$ και για $X = 2$
b) Να ξαναγραφεί ο παραπάνω αλγόριθμος χωρίς χρήση επαναληπτικών δομών

60) Να γίνει μετατροπή του παρακάτω τμήματος αλγορίθμου που χρησιμοποιεί την εντολή 'για', σε ισοδύναμα τμήματα που χρησιμοποιούν την 'όσο' και την 'μέχρις_ότου'.

```
D <- 2;
για X από 2 μέχρι 5 μεταβολή 2 κάνε
    A <- 10 * X;
    B <- 5 * X + 10;
    C <- A + B - (5 * X);
    D <- 3 * D - 5;
    Y <- A + B - C + D;
Τέλος_επανάληψης
Γράψε A mod B, C div D, Y * Y
```

61) Να γίνει εκτέλεση του παραπάνω αλγορίθμου και να βρεθούν οι τιμές των μεταβλητών A, B, C, D και Y μετά το τέλος κάθε επανάληψης καθώς και το τι θα τυπώσει ο αλγόριθμος.

62) Δίνεται το παρακάτω τμήμα αλγορίθμου

```
X <- 1
Όσο X < 5 επανάλαβε
    A <- X + 2
    B <- 3 * A - 4
    C <- B - A + 4
    Αν A > B τότε
        Αν A > C τότε
            MAX <- A
        Αλλιώς
            MAX <- C
        τέλος_αν
    αλλιώς
        Αν B > C τότε
            MAX <- B
        Αλλιώς
            MAX <- C
        Τέλος_αν
    Τέλος_αν
Εμφάνισε X, A, B, C, MAX
X <- X + 2
Τέλος_όσο
```

Ποιες είναι οι τιμές των μεταβλητών X, A, B, C, MAX που θα εμφανιστούν κατά την εκτέλεση του παραπάνω τμήματος αλγορίθμου;

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Βλάχος Στέφανος

Σημειώσεις - Ερωτήσεις - Ασκήσεις
Κεφαλαίων 3, 9
(Δομές Δεδομένων - Πίνακες)

Δεδομένα (σελ 53-54)

Σκοπός της χρήσης των αλγορίθμων είναι η εξαγωγή συμπερασμάτων και γνώσεων από την επεξεργασία των δεδομένων. Το παραγόμενο αποτέλεσμα ονομάζεται πληροφορία. Δε νοείται λοιπόν αλγόριθμος χωρίς κάποια δεδομένα που πρέπει να επεξεργαστεί. Στην παράγραφο 2.2 του βιβλίου (σελ 27), είχαμε δει πως η **Πληροφορική είναι η επιστήμη που μελετά τους αλγορίθμους από τις σκοπιές του Υλικού, των γλωσσών προγραμματισμού, την αναλυτική και τη θεωρητική**. Σε αυτό το σημείο θα πρέπει να επεκτείνουμε αυτή τη περιγραφή και να προσθέσουμε ότι:

Η πληροφορική μελετά επίσης τα δεδομένα από τις ακόλουθες σκοπιές:

Υλικού. Τα δεδομένα (κείμενο ή αριθμοί) μπορούν να αναπαρασταθούν στον υπολογιστή με διάφορες μεθόδους. Αυτοί οι μέθοδοι αναπαράστασης (representations) δεν έχουν επιλεγεί τυχαία, αλλά είναι αποτέλεσμα μελέτης, με σκοπό να κάνουν τα δεδομένα πιο εύχρηστα στον χειρισμό τους από τους αλγορίθμους. Κάποιες από τις αναπαραστάσεις αυτές είναι η δυαδική, το συμπλήρωμα ως προς 1, το συμπλήρωμα ως προς 2, ο κώδικας ASCII και ο κώδικας EBCDIC.

Γλωσσών Προγραμματισμού. Όπως έχουμε δει στη ΓΛΩΣΣΑ, υπάρχουν διάφοροι τύποι δεδομένων (ακέραιος, πραγματικός, χαρακτήρας, λογικός) με σκοπό να μπορούν να αναπαρασταθούν όλες οι πιθανές μορφές πληροφορίας. Κάθε γλώσσα προγραμματισμού παρέχει το δικό της σύνολο τύπων δεδομένων και η κάθε μια έχει τον δικό της τρόπο να αποθηκεύει και να χειρίζεται τις μεταβλητές που χρησιμοποιούμε. Αυτό έχει σαν αποτέλεσμα τη δημιουργία γλωσσών προγραμματισμού με ειδικευμένη χρήση, που προσφέρονται περισσότερο για την επίλυση συγκεκριμένων ειδών προβλημάτων.

Δομών Δεδομένων. Επειδή τις περισσότερες φορές καλούμαστε να μοντελοποιήσουμε και να αναπαραστήσουμε στον υπολογιστή πολύπλοκα δεδομένα της καθημερινής μας ζωής, δεν μας είναι αρκετοί οι τύποι δεδομένων που μας παρέχει μια γλώσσα προγραμματισμού. Ο συνδυασμός των βασικών τύπων δεδομένων για την δημιουργία πιο πολύπλοκων δομών έχει σαν αποτέλεσμα την δημιουργία δομών δεδομένων. Αν για παράδειγμα θέλουμε να αποθηκεύσουμε τα στοιχεία ενός ατόμου, δεν χρειάζεται να αποθηκεύσουμε 10 ξεχωριστές τιμές για το όνομα, το επίθετο την ηλικία το ύψος κ.α., αλλά μπορούμε να φτιάξουμε μια καινούργια δομή (εγγραφή) που θα περιλαμβάνει όλες τις παραπάνω τιμές. Ο συνδυασμός των εγγραφών πολλών ατόμων οδηγεί στην δημιουργία ενός αρχείου εγγραφών. Επειδή δεν υπάρχουν σαφείς κανόνες για την κατασκευή των δομών δεδομένων, η πληροφορική ασχολείται με την εύρεση των πιο αποτελεσματικών δομών.

Ανάλυσης Δεδομένων. Η συλλογή και η καταγραφή των δεδομένων είναι μόνο το πρώτο στάδιο στην αναπαράσταση μιας πραγματικής κατάστασης. Αν τα δεδομένα αυτά είναι ασύνδετα και δεν έχουν μεταξύ τους τις αλληλεξαρτήσεις που επιβάλλονται από τον πραγματικό κόσμο τότε είναι αδύνατο να έχουμε ένα αποτελεσματικό μοντέλο, τόσο ως προς την ενήμερωσή του όσο και ως προς την εξαγωγή χρήσιμων συμπερασμάτων. Με την αποτελεσματική ανάλυση των δεδομένων ασχολούνται οι τεχνολογίες των Βάσεων Δεδομένων, της Μοντελοποίησης Δεδομένων και της Αναπαράστασης Γνώσης.

Δομές Δεδομένων (σελ 54-55)

Ορισμός: Δομή Δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

Όπως αναφέραμε και παραπάνω η δομές δεδομένων είναι ένας σύνθετος τύπος δεδομένων που κατασκευάζεται από τον συνδυασμό απλών τύπων δεδομένων, με σκοπό την καλύτερη προσέγγιση και αναπαράσταση μιας πραγματικής κατάστασης. Η απλούστερη δομή που υπάρχει είναι η εγγραφή (η οποία δεν αποτελεί αντικείμενο μελέτης του βιβλίου). **Μια εγγραφή μπορεί να είναι μία μόνο μεταβλητή έως μια ομάδα διαφορετικών μεταβλητών που συνδυάζονται για να αναπαραστήσουμε ένα αντικείμενο.** Οι εγγραφές μπορούν να χρησιμοποιηθούν σαν δομικά στοιχεία για την κατασκευή πιο πολύπλοκων δομών και συνήθως χαρακτηρίζονται ως κόμβοι (nodes). Οι δομές που μπορούμε να κατασκευάσουμε είναι οι πίνακες, οι στοιβές, οι ουρές, τα δέντρα, οι γράφοι κ.α. Εμάς θα μας απασχολήσουν μόνο οι **πίνακες, οι ουρές και οι στοιβές.**

Οι βασικές λειτουργίες που θέλουμε να εκτελέσουμε πάνω στις δομές δεδομένων είναι οι ακόλουθες:

Προσπέλαση(access): κατά την οποία εντοπίζουμε έναν κόμβο με σκοπό να δούμε τα περιεχόμενά του ή να τα τροποποιήσουμε.

Εισαγωγή(insertion): κατά την οποία εισαγάγουμε στη δομή μας ένα καινούργιο κόμβο δεδομένων.

Διαγραφή(deletion): κατά την οποία διαγράφουμε έναν από τους κόμβους της δομής.

Αναζήτηση(searching): κατά την οποία αναζητούμε μέσα στη δομή κόμβους που πληρούν κάποια συγκεκριμένα κριτήρια.

Ταξινόμηση(sorting): κατά την οποία οι κόμβοι μιας δομής ταξινομούνται βάση μιας εκ των τιμών τους.

Αντιγραφή(copying): κατά την οποία δημιουργούμε ένα πλήρες ή μερικό αντίγραφο της δομής σε μια άλλη.

Συγχώνευση(merging): κατά την οποία δύο ή και περισσότερες δομές συνδυάζονται για την κατασκευή μιας καινούργιας .

Διαχωρισμός(separation): κατά την οποία μια δομή διαχωρίζεται σε μια ή περισσότερες δομές. Είναι η αντίστροφη διαδικασία της συγχώνευσης.

Η εφαρμογή των παραπάνω λειτουργιών πρέπει να είναι δυνατή σε κάθε δομή δεδομένων που κατασκευάζουμε. Ο τρόπος υλοποίησης της λειτουργίας διαφέρει από δομή σε δομή. Έτσι σε άλλες δομές είναι αποτελεσματική η εισαγωγή αλλά καθυστερεί η αναζήτηση, ενώ σε κάποιες άλλες μπορεί να συμβαίνει το αντίθετο. Αυτό μας δείχνει ότι καμιά δομή δεν είναι απαραίτητα καλύτερη από τις άλλες. Απλά η κάθε μια έχει τα χαρακτηριστικά της και εμείς επιλέγουμε την καλύτερη για τη δουλειά που πρέπει να κάνουμε.

Η επιστήμη της Πληροφορικής δεν μελετά ποτέ τους αλγορίθμους και τα δεδομένα ξεχωριστά. Η αποδοτικότητα ενός αλγορίθμου καθορίζεται σε πολύ μεγάλο βαθμό από την χρήση της κατάλληλης δομής δεδομένων και το σωστό χειρισμό της. Αυτή η ενοποίηση των όρων αλγόριθμος και δομές δεδομένων πρωτοεμφανίστηκε στο βιβλίο του **Wirth** το 1976 με τίτλο «**Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα**».

Πριν ξεκινήσουμε την περιγραφή των διαφόρων δομών θα κάνουμε ένα σημαντικό διαχωρισμό τους. Οι δομές χωρίζονται σε **στατικές** και **δυναμικές**.

Οι δομές δεδομένων αποθηκεύονται στη μνήμη των υπολογιστών. **Στατικές** είναι οι δομές που καταλαμβάνουν σταθερό χώρο στη μνήμη καθ' όλη τη διάρκεια της εκτέλεσης του προγράμματος που τις χρησιμοποιεί. Αυτό σημαίνει ότι αν δεν γνωρίζουμε εκ των προτέρων πόσα δεδομένα θα χρειαστεί να αποθηκεύσουμε, μπορεί να σπαταλήσουμε μεγάλο μέρος της μνήμης του υπολογιστή απλά δεσμεύοντας χώρο που πιθανών να χρειαστεί. Σαν αποτέλεσμα μπορεί να χρησιμοποιήσουμε τελικά πολύ λίγο από το χώρο που δεσμεύσαμε, ή μπορεί από κακή εκτίμηση να μην μας φτάσει. **Στατικές είναι και οι δομές στις οποίες θα αναφερθούμε.**

Αντίθετα οι **δυναμικές** προσαρμόζουν το μέγεθός τους και το χώρο που καταλαμβάνουν, στις αποθηκευτικές μας ανάγκες. Αν χρειαστεί να αποθηκεύσουμε καινούργια δεδομένα τότε η δομή επεκτείνεται, ενώ αν οι ανάγκες μας μειωθούν η δομή συρρικνώνεται απελευθερώνοντας μνήμη.

Πίνακες(σελ 56-59, 185-200)

Οι πίνακες είναι η βασικότερη **στατική** δομή δεδομένων, που υποστηρίζεται από όλες τις γλώσσες προγραμματισμού.

Ως γνωστό ή κάθε μια μεταβλητή που δηλώνουμε στην αρχή των προγραμμάτων μπορεί να αποθηκεύσει μόνο μια τιμή. Αν θέλουμε σε ένα πρόγραμμα να αποθηκεύσουμε πολλές τιμές του ίδιου τύπου θα πρέπει είτε να δηλώσουμε ισάριθμες μεταβλητές ή να χρησιμοποιήσουμε μια μεταβλητή η οποία θα χάνει τις παλαιότερες τιμές όταν θα της αναθέτουμε μια καινούργια. Ας θεωρήσουμε το παρακάτω παράδειγμα:

Να γίνει πρόγραμμα το οποίο θα διαβάζει τις βαθμολογίες 100 μαθητών σε ένα μάθημα και θα επιστρέφει το πλήθος των μαθητών που ξεπέρασαν το μέσο όρο.

Αν επιλέξουμε να λύσουμε το πρόβλημα χρησιμοποιώντας 100 διαφορετικές μεταβλητές (μια για κάθε βαθμό), θα λύσουμε μεν το πρόβλημα, αλλά το αποτέλεσμα θα είναι ένα μεγάλο πρόβλημα που θα είναι δύσκολο τόσο στη συγγραφή όσο και στην τροποποίηση.

Πρόγραμμα μαθητές

Πραγματικές β1, β2, β3, ... ,β100, Σ, ΜΟ

(αυτό βέβαια δεν μπορεί να γίνει στο προγραμματισμό θα ήμασταν αναγκασμένοι να γράψουμε όλες τις μεταβλητές)

ακέραιος πλήθος

Αρχή

Σ ← 0

Διάβασε β1

Σ ← Σ + β1

Διάβασε β2

Σ ← Σ + β2

Διάβασε β3

Σ ← Σ + β3

.

. (πάλι θα έπρεπε να γραφεί και για τις 100 μεταβλητές)

.

Διάβασε β100

Σ ← Σ + β100

ΜΟ ← Σ/100

Πλήθος ← 0

Αν (β1>ΜΟ) **τότε**

Πλήθος ← Πλήθος+1

Τέλος_αν

Αν (β2>ΜΟ) **τότε**

Πλήθος ← Πλήθος+1

Τέλος_αν

Αν (β3>ΜΟ) **τότε**

Πλήθος ← Πλήθος+1

Τέλος_αν

.

. (πάλι θα έπρεπε να γραφεί και για τις 100 μεταβλητές)

.

Αν (β100>ΜΟ) **τότε**

Πλήθος ← Πλήθος+1

Τέλος_αν

Τέλος

Φυσικά μια τέτοια λύση είναι απαράδεκτη ως προς το χρόνο, το μέγεθος και την δυνατότητα τροποποίησης.

Αντίθετα με το πίνακα έχουμε τη δυνατότητα να ομαδοποιήσουμε αυτές τις 100 τιμές κάτω από ένα κοινό όνομα. **Ο πίνακας λοιπόν είναι μια μεταβλητή που μπορεί να αποθηκεύσει πολλές τιμές του ίδιου τύπου. Το πλήθος των τιμών είναι το μέγεθος του**

πίνακα και είναι σταθερό καθ' όλη τη διάρκεια της εκτέλεσης του προγράμματος. Ακριβώς επειδή το μέγεθος του πίνακα δεν αλλάζει, ονομάζουμε τη δομή **στατική**.

Αν θέλαμε να χρησιμοποιήσουμε σε κάποιο πρόγραμμα ένα πίνακα που να μπορεί να αποθηκεύει 5 ακεραίους τότε θα δεσμεύαμε στη μνήμη του υπολογιστή 5 θέσεις ακεραίων που θα τους δίναμε ένα όνομα, έστω Π. Παρακάτω φαίνεται πως θα ήταν ο πίνακας στη μνήμη του υπολογιστή με κάποιες τυχαίες τιμές αποθηκευμένες.

Πίνακας Π

5	32	44	9	-9
---	----	----	---	----

Ας θεωρήσουμε ακόμα μια ακέραια μεταβλητή Α. Μπορείτε να εκτελέσετε την εντολή «Α ← 8» χωρίς να χρειαστεί να διευκρινίσετε σε ποια θέση μνήμης θα μπει το νούμερο 8. Αυτό συμβαίνει γιατί η μεταβλητή Α αναπαριστά μόνο μια θέση μνήμης και σε αυτήν θα αποθηκευτεί το 8. Στην περίπτωση όμως του πίνακα δεν μπορούμε να πούμε «Π ← 8», διότι ο πίνακας αναπαριστά 5 θέσεις μνήμης και δεν έχει γίνει σαφές σε ποια από αυτές θα αποθηκευτεί ο αριθμός. **Γενικά δεν μπορεί να γίνει ομαδική επεξεργασία των στοιχείων του πίνακα.** Άρα ποτέ το Π δεν μπορεί να εμφανιστεί σε μία έκφραση μόνο του, χωρίς προσδιορισμό της θέσης που θέλουμε να προσπελάσουμε. Για να λύσουμε αυτό το πρόβλημα αριθμούμε τις θέσεις του πίνακα και δίνουμε στη κάθε μια ένα αύξων αριθμό που τη χαρακτηρίζει μοναδικά. Αυτός ο αριθμός ονομάζεται **διατακτικός αριθμός** ή **δείκτης** της θέσης. Έτσι ο πίνακας Π θα μπορούσε να αναπαρασταθεί ως εξής:

Πίνακας Π

1	2	3	4	5
5	32	44	9	-9

Για να μπορέσετε λοιπόν να προσπελάσετε μια θέση του πίνακα πρέπει εκτός από το όνομά του να γράψετε και τον δείκτη της θέσης που επιθυμείτε, μέσα σε μια αγκύλη ή παρένθεση, ως εξής:

Π[1] ← 9
 Π[3] ← Π[1]+Π[2]
 Π[κ] ← Π[κ-1]+Π[κ-2]
 Π[κ-1] ← 7 κ.ο.κ.

Η χρήση μεταβλητών μέσα στις αγκύλες είναι επιτρεπτή εφ' όσον γνωρίζουμε τις τιμές των μεταβλητών κατά την εκτέλεση του αλγορίθμου και εφ' όσον αυτές οι τιμές δεν αναφέρονται σε ανύπαρκτες θέσεις. Για παράδειγμα για το πίνακα Π δεν μπορούμε να εκτελέσουμε την εντολή «Π[8] ← 10» διότι δεν υπάρχει 8^η θέση (ο πίνακας έχει μόνο 5). Με το ίδιο σκεπτικό μπορούμε να χρησιμοποιήσουμε μια εντολή Π[κ-2] μόνο αν η μεταβλητή κ κυμαίνεται από 3 έως 7, έτσι ώστε αφαιρώντας το 2 να προκύπτει μια έγκυρη τιμή μέσα στις αγκύλες (από 1 έως 5).

Πρέπει να τονίσουμε σε αυτό το σημείο ότι οι διατακτικοί αριθμοί των θέσεων χρησιμοποιούνται μόνο για τον προσδιορισμό των θέσεων και δεν έχουν καμία σχέση με το περιεχόμενό τους. Για να γίνει καλύτερα κατανοητό ας δούμε την αναπαράσταση ενός πίνακα με το όνομα ΧΑΡ που έχει 5 θέσεις τύπου χαρακτήρα.

Πίνακας ΧΑΡ

1	2	3	4	5
Κ	?	Ω	&	%

Για να χρησιμοποιήσετε ένα πίνακα σε έναν αλγόριθμο δεν χρειάζεται να τον δηλώσετε. Αντίθετα η δήλωση είναι απαραίτητη στα προγράμματα αλλά δεν διαφέρει πολύ από τη δήλωση απλών μεταβλητών. Αρκεί δίπλα από το όνομα της μεταβλητής να μπει σε αγκύλη το πλήθος των θέσεων που θέλουμε να έχει. Έτσι στο τμήμα δηλώσεων ενός

προγράμματος θα γράφαμε στο τμήμα δηλώσεων τα παρακάτω για τους πίνακες Π και ΧΑΡ, που είδαμε νωρίτερα.

```
.  
.   
Ακέραιες Π[5]  
Χαρακτήρα ΧΑΡ[5]  
.   
.
```

Οι πίνακες στους οποίους αναφερθήκαμε μέχρι τώρα ονομάζονται **μονοδιάστατοι** γιατί οι θέσεις μνήμης αντιμετωπίζονται σαν μια σειρά από θέσεις που μπορούν να προσδιοριστούν με τη χρήση ενός και μόνο **δείκτη**. Μπορούμε όμως να χρησιμοποιήσουμε και πίνακες περισσότερων διαστάσεων, οι οποίοι θα έχουν διαφορετικοί λογική διάταξη και θα χρειάζονται περισσότερους δείκτες για το προσδιορισμό μιας θέσης.

Ας θεωρήσουμε ξανά το προηγούμενο παράδειγμα:

Να γίνει πρόγραμμα το οποίο θα διαβάζει τις βαθμολογίες 100 μαθητών σε ένα μάθημα και θα επιστρέφει το πλήθος των μαθητών που ξεπέρασαν το μέσο όρο.

Ακολουθώντας το σκεπτικό της πρώτης λύσης και όσα γνωρίζουμε μέχρι τώρα θα μπορούσαμε να χρησιμοποιήσουμε αντί για 100 μεταβλητές μόνο ένα πίνακα 100 θέσεων. Η λύση της άσκησης θα ήταν η εξής:

```
Πρόγραμμα μαθητές  
Πραγματικές Β[100], Σ, ΜΟ  
ακέραιος πλήθος  
Αρχή  
    Σ ← 0  
    //Ανάγνωση του πίνακα  
    Διάβασε Β[1]  
    Διάβασε Β[2]  
    Διάβασε Β[3]  
    .  
    .  
    .  
    Διάβασε Β[100]  
    //Υπολογισμός μέσου όρου  
    Σ ← Σ + Β[1]  
    Σ ← Σ + Β[2]  
    Σ ← Σ + Β[3]  
    .  
    .  
    .  
    Σ ← Σ + Β[100]  
  
    ΜΟ ← Σ/100  
  
    //καταμέτρηση μαθητών με βαθμό μεγαλύτερο του μέσου όρου  
    Πλήθος ← 0  
    Αν (Β[1]>ΜΟ) τότε  
        Πλήθος ← Πλήθος+1  
    Τέλος_αν  
    Αν (Β[2]>ΜΟ) τότε  
        Πλήθος ← Πλήθος+1  
    Τέλος_αν  
    Αν (Β[3]>ΜΟ) τότε  
        Πλήθος ← Πλήθος+1  
    Τέλος_αν  
    .  
    .
```

```
·
  Αν (B[100]>ΜΟ) τότε
    Πλήθος←Πλήθος+1
  Τέλος_αν
```

Τέλος

Μια τέτοια λύση θα ήταν σωστή αν και δεν μπορούμε να πούμε ότι κερδίζουμε τίποτα ιδιαίτερο σε σχέση με τον πρώτο τρόπο λύσης, αφού καταλαμβάνουμε τον ίδιο χώρο στη μνήμη και το πρόγραμμα είναι το ίδιο μεγάλο και δύσχρηστο.

Η βασική διαφορά τους είναι πως το όνομα «β10» μιας μεταβλητής είναι ενιαίο και δεν μπορεί να διασπαστεί. Το 10 δεν θεωρείται αριθμός αλλά αδιάσπαστο τμήμα του ονόματος. **Δεν μπορούμε να πούμε ότι έχουμε την μεταβλητή «βί», όπου i παίρνει τιμές από 1 έως 100.** Αντίθετα κάτι τέτοιο συμβαίνει στους πίνακες. Στην ονομασία «B[10]» το 10 δεν είναι σταθερό κομμάτι του ονόματος αλλά ένας αριθμός που προσδιορίζει τον αύξων αριθμό της θέσης του πίνακα και επομένως μπορεί να αλλάξει. Μπορούμε να βάλουμε αντί για το «10» μια μεταβλητή η οποία θα αλλάζει τιμές, αρκεί οι τιμές που θα παίρνει να είναι έγκυρες (δηλαδή να υπάρχουν οι αντίστοιχες θέσεις του πίνακα). Με τη χρήση αυτής της ιδιότητας μπορούμε να θεωρήσουμε ότι οι 100 εντολές για την ανάγνωση των στοιχείων του πίνακα είναι ίδιες και το μόνο που αλλάζει είναι ο αριθμός της θέσης που διαβάζουμε. Αν λοιπόν τις βάλουμε σε μια εντολή επανάληψης χρησιμοποιώντας έναν μετρητή που θα παίρνει τιμές από 1 έως 100 θα έχουμε την ακόλουθη εντολή

```
Για i από 1 έως 100 κάνε
  Διάβασε B[i]
Τέλος_επανάληψης
```

Την ίδια ιδιότητα μπορούμε να τη χρησιμοποιήσουμε και στις άλλες εργασίες (υπολογισμός μέσου όρου, σύγκριση βαθμών), οπότε και προκύπτει ο παρακάτω αλγόριθμος

```
Πρόγραμμα μαθητές
Πραγματικές B[100], Σ, ΜΟ
ακέραιος πλήθος
Αρχή
```

```
  //Ανάγνωση του πίνακα
  Για i από 1 έως 100 κάνε
    Διάβασε B[i]
  Τέλος_επανάληψης
  //Υπολογισμός μέσου όρου
  Σ← 0
  Για i από 1 έως 100 κάνε
    Σ← Σ + B[i]
  Τέλος_επανάληψης
  ΜΟ ← Σ/100
```

```
  //καταμέτρηση μαθητών με βαθμό μεγαλύτερο του μέσου όρου
  Πλήθος ← 0
  Για i από 1 έως 100 κάνε
    Αν (B[i]>ΜΟ) τότε
      Πλήθος←Πλήθος+1
    Τέλος_αν
  Τέλος_επανάληψης
  Γράψε ΜΟ, Πλήθος
```

Τέλος

Αυτή η προσέγγιση όχι μόνο μας δίνει πιο σύντομους αλγόριθμους, αλλά μας δίνει και τη δυνατότητα να τους αλλάξουμε πολύ εύκολα, αλλάζοντας μόνο το μέγεθος του πίνακα και τις εντολές «για».

Για τη λύση αυτής της άσκησης είναι απαραίτητη η διατήρηση των τιμών που μας δίνει ο χρήστης στη μνήμη, ώστε να τις επεξεργαστούμε δύο φορές:

- A) μια φορά για τον υπολογισμό του μέσου όρου
- B) και μια φορά μετά τον υπολογισμό του μέσου όρου για να δούμε πόσοι μαθητές τον ξεπέρασαν.

Σε τέτοιες περιπτώσεις ασκήσεων που πρέπει να κρατήσουμε πολλές τιμές στη μνήμη και να τις επεξεργαστούμε πάνω από μια φορά, η χρήση του πίνακα είναι υποχρεωτική. Άλλες πάλι φορές μπορούμε να χρησιμοποιήσουμε τον πίνακα για να διευκολυνθούμε χωρίς να είναι αναγκαίοι.

Προβλήματα από τη χρήση πινάκων μπορούν να προκύψουν σε περιπτώσεις όπου δεν είμαστε σίγουροι από την αρχή για το μέγεθος του προβλήματος και το σύνολο των θέσεων που χρειαζόμαστε. Επειδή οι πίνακες είναι στατικές δομές θα πρέπει να κάνουμε μια εκτίμηση των αναγκών, αφού το μέγεθος του πίνακα ορίζεται στην αρχή και δεν μπορεί να αλλάξει. Έτσι μπορεί

A) να υπερεκτιμήσουμε τις ανάγκες μας και απλά να σπαταλήσουμε μνήμη (αφού θα έχει ήδη δεσμευτεί) ή να δεσμεύσουμε τόση πολύ μνήμη που να είναι τελικά αδύνατη η ολοκλήρωση της εκτέλεσης του προγράμματος. Άρα **οι πίνακες έχουν υψηλές απαιτήσεις σε μνήμη** και πρέπει να χρησιμοποιούνται με προσοχή

B) να υποεκτιμήσουμε τις ανάγκες μας και το αρχικό μέγεθος του πίνακα να μην είναι αρκετό για την αποθήκευση όλων των στοιχείων. Σε αυτή τη περίπτωση **περιορίζονται οι δυνατότητες του προγράμματος**.

Πολυδιάστατοι Πίνακες

Στους μονοδιάστατους πίνακες θεωρούμε ότι έχουμε μια σειρά από θέσεις μνήμης. Μπορείτε να διαχωρίσετε αυτές τις θέσεις χρησιμοποιώντας έναν ξεχωριστό αριθμό για κάθε μια, που ονομάζεται **δείκτης**. Υπάρχουν περιπτώσεις όπου τα δεδομένα που θέλουμε να αποθηκεύσουμε είναι του ίδιου τύπου αλλά έχουν και κάποια λογική οργάνωση. Για παράδειγμα

Να γίνει πίνακας στον οποίο θα αποθηκευτούν οι βαθμοί δέκα μαθητών σε πέντε διαφορετικά μαθήματα.

Με όσα γνωρίζουμε μέχρι τώρα για τους πίνακες θα μπορούσαμε να πούμε ότι θα χρησιμοποιήσουμε ένα μονοδιάστατο πίνακα 50 θέσεων (πέντε για κάθε μαθητή) και θα αποθηκεύσουμε σε αυτόν τους βαθμούς. Η προσέγγιση αυτή, αν και δεν είναι λάθος, δεν είναι βολική γιατί απαιτεί έναν πολύ ειδικό χειρισμό του πίνακα. Η δομή αυτού του πίνακα δεν μας θυμίζει καθόλου τη λογική οργάνωση των δεδομένων.

Προσπαθώντας να βελτιώσουμε την παραπάνω αντιμετώπιση θα μπορούσαμε να χρησιμοποιήσουμε δέκα πίνακες με πέντε θέσεις ο κάθε ένας. Και αυτή η προσέγγιση είναι σωστή αλλά μας αναγκάζει να χρησιμοποιήσουμε πολλές μεταβλητές τύπου πίνακα, κάτι που θα οδηγήσει σε πολύπλοκο πρόγραμμα.

Αυτό που θέλουμε είναι χρησιμοποιώντας μόνο ένα πίνακα να αποθηκεύσουμε όλα τα στοιχεία αλλά να είμαστε σε θέση να διαχωρίσουμε τα διαφορετικά του τμήματα (τις δέκα πεντάδες βαθμών). Αυτό το επιτυγχάνουμε με τη χρήση των δισδιάστατων πινάκων.

Ένας δισδιάστατος πίνακας μπορεί να θεωρηθεί σαν ένας **πίνακας πινάκων**. Ας θεωρήσουμε ότι έχουμε δέκα πίνακες, πέντε θέσεων ο κάθε ένας. Κάθε πίνακας έχει το δικό του όνομα και περιέχει πέντε ακέραιες τιμές.

	1	2	3	4	5
Μαθητής 1	15	16	18	19	15
Μαθητής 2	20	18	17	13	19
Μαθητής 3	19	13	17	18	15
Μαθητής 4	12	17	20	16	13
Μαθητής 5	13	14	19	17	18
Μαθητής 6	16	15	18	19	18
Μαθητής 7	17	17	18	19	18
Μαθητής 8	16	19	19	18	17
Μαθητής 9	15	14	17	18	17
Μαθητής 10	13	15	14	17	19

Οι δέκα πίνακες έχουν όμοια δομή, δηλαδή είναι του ίδιου τύπου. Γνωρίζουμε όμως ότι μπορούμε να φτιάξουμε πίνακα συνδυάζοντας μεταβλητές ίδιου τύπου. Με αυτό το σκεπτικό ενώνουμε του δέκα πίνακες σε ένα. Αυτόν τον νέο πίνακα τον ονομάζουμε **Βαθμολογίες**. Ο πίνακας **Βαθμολογίες** περιέχει δέκα πίνακες πέντε θέσεων ο κάθε ένας, τους οποίους θα πρέπει να αριθμήσουμε. Έτσι προκύπτει η παρακάτω δομή.

	1	2	3	4	5
1	15	16	18	19	15
2	20	18	17	13	19
3	19	13	17	18	15
4	12	17	20	16	13
5	13	14	19	17	18
6	16	15	18	19	18
7	17	17	18	19	18
8	16	19	19	18	17
9	15	14	17	18	17
10	13	15	14	17	19

Αυτός ο νέος πίνακας αποτελείται από δέκα γραμμές και πέντε στήλες. Η βασική διαφορά ενός τέτοιου πίνακα από ένα μονοδιάστατο είναι ότι για τον προσδιορισμό μιας θέσης του πίνακα δεν αρκεί ένας δείκτης, αφού πρέπει να προσδιορίσουμε τόσο τη γραμμή όσο και τη στήλη που βρίσκεται. Αντίθετα χρησιμοποιούμε δύο δείκτες π.χ Βαθμολογίες[6,3]. Το ποιο από τα δύο νούμερα θα αντιπροσωπεύει τις γραμμές και ποιο τις στήλες, δεν έχει καμία σημασία. Αυτό καθορίζεται από τη δήλωση του πίνακα. Αν για παράδειγμα ο πίνακας δηλωθεί ως «**Βαθμολογίες[10,5]**», τότε υπονοείται ότι το πρώτο νούμερο είναι οι γραμμές και το δεύτερο είναι οι στήλες. Αυτή η σύμβαση θα πρέπει να τηρείται σε όλη την έκταση του προγράμματος. Μπορούμε αν θέλουμε να δηλώσουμε τον πίνακα και ανάποδα, π.χ. «**Βαθμολογίες[5,10]**», αλλά θα πρέπει να έχουμε υπ' όψη ότι το πρώτο νούμερο αντιπροσωπεύει το μάθημα και το δεύτερο το μαθητή.

Έτσι το πρόβλημα που θέσαμε αρχικά αντιμετωπίζεται ως εξής:

Πρόγραμμα Βαθμοί
Πραγματικές Βαθμολογίες[10,5]
ακέραιος i,j
Αρχή
Για i από 1 έως 10 κάνε
Για j από 1 έως 5 κάνε

Διάβασε Βαθμολογίες[i,j]
Τέλος_επανάληψης
Τέλος_επανάληψης
Τέλος

Τυπικές Επεξεργασίες Πινάκων (σελ 198)

Οι βασικότερες εργασίες που γίνονται πάνω στους πίνακες είναι οι εξής:

- 1) Υπολογισμός αθροίσματος στοιχείων του πίνακα
 - 2) Εύρεση μεγίστου ή ελαχίστου στοιχείου.
 - 3) Ταξινόμηση στοιχείων πίνακα
 - 4) Αναζήτηση ενός στοιχείου στον πίνακα
 - 5) Συγχώνευση πινάκων
- 1) **Αθροίσματα.** Στην απλούστερη μορφή του αυτό το πρόβλημα έγκειται στην άθροιση όλων των στοιχείων του πίνακα, είτε αυτός είναι μονοδιάστατος, είτε πολυδιάστατος. Υπάρχει βέβαια και η πιθανότητα να ζητηθεί μερικό άθροισμα στοιχείων, που έχουν κάποια ιδιότητα. Στους δισδιάστατους πίνακες πολύ συχνά ζητείται το άθροισμα κατά στήλες ή κατά γραμμές.
 - 2) **Μέγιστος – Ελάχιστος.** Αυτοί οι αλγόριθμοι παρουσιάζουν ενδιαφέρον μόνο σε αταξινομητους πίνακες. Σε ταξινομημένους πίνακες ο μέγιστος και ο ελάχιστος μπορούν να βρεθούν εύκολα αφού βρίσκονται στην πρώτη και στην τελευταία θέση του πίνακα.
 - 3) **Ταξινόμηση.** Έχουν αναπτυχθεί κατά καιρούς διάφοροι αλγόριθμοι για την επίλυση αυτού του δύσκολου και απαιτητικού προβλήματος, όπως είναι η **γρήγορη ταξινόμηση**, η **ταξινόμηση ευθείας ανταλλαγής**, η **ταξινόμηση φυσαλίδας** κ.α. Στην ύλη του μαθήματος είναι μόνο η **ταξινόμηση φυσαλίδας**.
 - 4) **Αναζήτηση.** Και για το πρόβλημα της αναζήτησης έχουν αναπτυχθεί διάφοροι αλγόριθμοι. Σκοπός της αναζήτησης είναι να βρεθεί αν κάποιο συγκεκριμένο στοιχείο υπάρχει σε ένα πίνακα και αν υπάρχει σε ποια θέση βρίσκεται. Αν ο πίνακας είναι ταξινομημένος τότε χρησιμοποιείται η **δυναμική αναζήτηση**, ενώ σε αταξινομητους πίνακες χρησιμοποιείται η **σειριακή**.
 - 5) **Συγχώνευση.** Ο αλγόριθμος της συγχώνευσης πινάκων δεν αποτελεί αντικείμενο μελέτης. Σκοπός της συγχώνευσης είναι να παραχθεί, από δύο αρχικούς ταξινομημένους πίνακες μεγέθους M και N θέσεων, ένας νέος πίνακας μεγέθους (N+M), ο οποίος θα είναι επίσης ταξινομημένος.

Άλλες Στατικές Δομές Δεδομένων

Δύο πολύ βασικές δομές δεδομένων που χρησιμοποιούνται ευρύτατα είναι οι **ουρές** και οι **στοίβες**. Αυτές οι δύο δομές μπορούν να υλοποιηθούν και σαν **στατικές** και σαν **δυναμικές**. Οι **δυναμικές στοίβες** και **ουρές** δεν θα μας απασχολήσουν. Οι στατικές, χρησιμοποιούν για την υλοποίησή τους την δομή του πίνακα.

Σε μια δομή πίνακα θεωρούμε συνήθως ενεργές όλες τις διαθέσιμες θέσεις και είμαστε ελεύθεροι να προσπελάσουμε όποια θέση θέλουμε. Ο τρόπος και η σειρά προσπέλασης καθορίζονται συνήθως από το πρόβλημα που καλούμαστε να επιλύσουμε, αλλά συνήθως είμαστε ελεύθεροι να αυτοσχεδιάσουμε.

Στις ουρές και τις στοίβες αντίθετα, θεωρούμε ανά πάσα στιγμή ενεργό ένα μόνο τμήμα του πίνακα. Μπορούμε να επεξεργαστούμε τα στοιχεία που έχουμε αποθηκεύσει ή να προσθέσουμε καινούργια, αλλά πρέπει να τηρούμε κάποιους κανόνες.

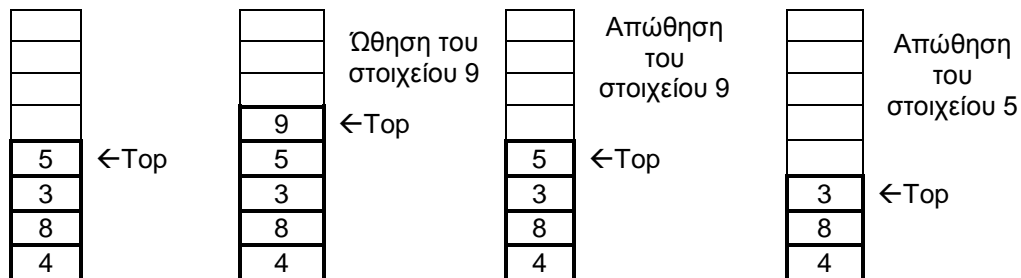
A) Τα καινούργια στοιχεία μπαίνουν στην αμέσως επόμενη ελεύθερη θέση (ανάλογα με τη δομή)

- B) Ένα στοιχείο που επεξεργαζόμαστε, αυτομάτως αφαιρείται από τη δομή
 Γ) Οι προσθέσεις και οι αφαιρέσεις στοιχείων γίνονται μόνο στα άκρα των δομών.

Στοίβα (σελ 59)

Η δομή της στοίβας δουλεύει με το σκεπτικό μιας στοίβας πιάτων. Σε μια στοίβα πιάτων το πιάτο που τοποθετήθηκε **πρώτο** βρίσκεται στον πάτο της στοίβας (**bottom**) και θα πλυθεί **τελευταίο**. Αντίθετα το πιάτο που τοποθετήθηκε **τελευταίο** στη κορυφή της στοίβας (**top**), θα πλυθεί **πρώτο**. Αυτός ο τρόπος λειτουργίας ονομάζεται **LIFO (Last-In-First-Out)**, δηλαδή **τελευταίο μέσα-πρώτο έξω**.

Για την υλοποίηση της στοίβας χρησιμοποιούμε ένα πίνακα. Δεν είναι απαραίτητο ο πίνακας να είναι γεμάτος. Ανά πάσα στιγμή είναι τοποθετημένα στον πίνακα κάποια στοιχεία. Το πρώτο που τοποθετήθηκε βρίσκεται στο κάτω μέρος της στοίβας. Το τελευταίο που εισήχθη στην στοίβα βρίσκεται στο πάνω μέρος και ονομάζεται **κορυφή (Top)**. Για να υποδηλώσουμε την κορυφή συνήθως χρησιμοποιούμε μια ακέραη μεταβλητή που ονομάζουμε **top** και η τιμή της δείχνει σε ποια θέση του πίνακα βρίσκεται η κορυφή της στοίβας.



Αν χρειαστεί να τοποθετήσουμε ένα νέο στοιχείο στη στοίβα, τότε θα το βάλουμε στην κορυφή και η μεταβλητή **top** θα αυξηθεί κατά ένα. Η διαδικασία της εισαγωγής νέου στοιχείου ονομάζονται **ώθηση (push)**. Αν πάλι χρειαστεί να επεξεργαστούμε ένα στοιχείο της στοίβας τότε είμαστε υποχρεωμένοι να πάρουμε από τη κορυφή. Το στοιχείο που επεξεργαζόμαστε αφαιρείται από τη δομή και η μεταβλητή **top** μειώνεται κατά ένα. Η διαδικασία της αφαίρεσης στοιχείου ονομάζεται **απώθηση (pop)**.

Στις στατικές στοίβες υπάρχει ένας περιορισμός λόγω του σταθερού μεγέθους του πίνακα. Αν ο πίνακας είναι για παράδειγμα 10 θέσεων, τότε είναι σίγουρο ότι η στοίβα δεν μπορεί να γίνει μεγαλύτερη. Αν σε μια ήδη γεμάτη στοίβα προσπαθήσουμε να προσθέσουμε ένα ακόμα στοιχείο τότε έχουμε **υπερχείλιση στοίβας**. Ένα τέτοιο φαινόμενο πρέπει να αποφεύγεται με τον κατάλληλο έλεγχο ώστε η ώθηση καινούργιου στοιχείου να επιτρέπεται μόνο αν η στοίβα δεν είναι γεμάτη ($top < 10$).

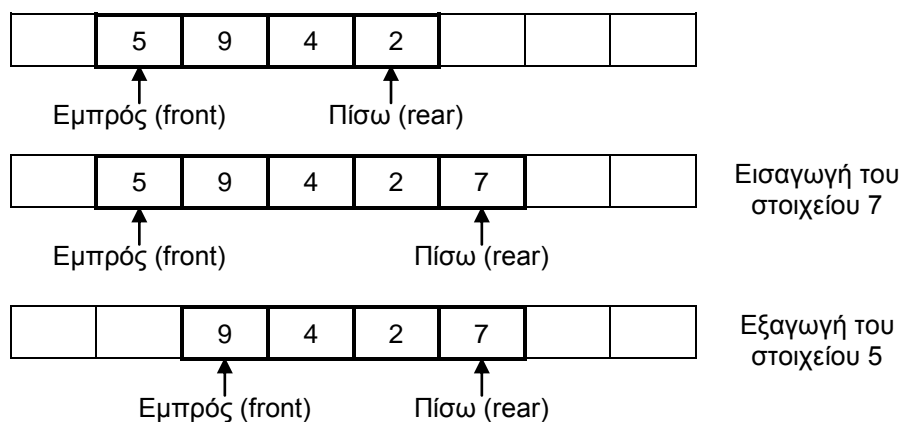
Στην αντίθετη περίπτωση, που η στοίβα είναι άδεια και προσπαθούμε να απωθήσουμε ένα στοιχείο, έχουμε **υποχείλιση στοίβας**. Για να είναι δυνατή η απώθηση στοιχείου θα πρέπει η στοίβα να περιέχει τουλάχιστον ένα στοιχείο ($top >= 1$).

Ουρά (σελ 61)

Η δομή της **ουράς** δουλεύει με το σκεπτικό των κλασικών ουρών που συναντάμε στην καθημερινή μας ζωή. Στην ουρά μιας υπηρεσίας, ο πελάτης που ήρθε **πρώτος** θα εξυπηρετηθεί **πρώτος**. Αντίθετα ο πελάτης που έφτασε **τελευταίος** θα εξυπηρετηθεί **τελευταίος**. Αυτός ο τρόπος λειτουργίας ονομάζεται **FIFO (First-In-First-Out)**, δηλαδή **πρώτο μέσα-πρώτο έξω**.

Για την υλοποίηση της ουράς χρησιμοποιούμε πάλι ένα πίνακα. Δεν είναι απαραίτητο ο πίνακας να είναι γεμάτος. Ανά πάσα στιγμή είναι τοποθετημένα στον πίνακα κάποια στοιχεία. Το πρώτο που τοποθετήθηκε βρίσκεται στο ένα άκρο της ουράς που ονομάζεται **εμπρός (front)**. Το τελευταίο που εισήχθη στην ουρά βρίσκεται στο άλλο άκρο που

ονομάζεται **πίσω (rear)**. Για να υποδηλώσουμε το εμπρός και το πίσω μέρος της ουράς συνήθως χρησιμοποιούμε δυο ακέραιες μεταβλητές που ονομάζουμε **front** και **rear**. Οι τιμές τους δείχνουν σε ποια θέση του πίνακα βρίσκονται τα άκρα της ουράς.



Αν χρειαστεί να τοποθετήσουμε ένα νέο στοιχείο στην ουρά, τότε θα το βάλουμε στο πίσω άκρο και η μεταβλητή **rear** θα αυξηθεί κατά ένα. Η διαδικασία της εισαγωγής νέου στοιχείου ονομάζονται **εισαγωγή (enqueue)**. Αν πάλι χρειαστεί να επεξεργαστούμε ένα στοιχείο της ουράς τότε είμαστε υποχρεωμένοι να το πάρουμε από το εμπρός άκρο. Το στοιχείο που επεξεργαζόμαστε αφαιρείται από τη δομή και η μεταβλητή **front** αυξάνεται κατά ένα. Η διαδικασία της αφαίρεσης στοιχείου ονομάζεται **εξαγωγή (dequeue)**.

Μια πολύ βασική διαφορά ανάμεσα στις δυο δομές είναι οι εργασίες που γίνονται στα άκρα τους. Στην **στοίβα** μόνο το ένα άκρο είναι ενεργό (η **κορυφή**) και όλες οι εργασίες γίνονται σε αυτό το άκρο (**ώθηση** και **απώθηση**). Στην **ουρά** και τα δύο άκρα είναι ενεργά, αλλά στο **εμπρός** γίνονται μόνο **εξαγωγές**, ενώ στο **πίσω** μόνο **εισαγωγές** στοιχείων.

Πίνακες

- 1) Έστω τρεις πίνακες που δηλώνονται ως εξής: Ακέραιος $A[4]$, $B[6]$, $\Gamma[4,6]$
Σημειώστε ένα Χ στα αριστερά των εντολών (ομάδων εντολών) που θεωρείτε λανθασμένες. Σε όποιες εντολές δεν προσδιορίζονται τα i , j θεωρήστε ότι οι τιμές τους είναι έγκυρες.
 - a) $A[5] \leftarrow \Gamma[1,6]$;
 - b) $B[2]+A[1] \leftarrow A[2]$;
 - c) $B[1] \leftarrow A[4] + \Gamma[4,6]$;
 - d) Για i από 1 μέχρι 6 κάνε
 $B[i] \leftarrow i$;
 $\Gamma[1, i] \leftarrow i*i$;
Τέλος_επανάληψης
 - e) $\Gamma[i, j] \leftarrow \Gamma[j, i]$;
 - f) $A[1] \leftarrow B[1]+2,5$;
 - g) Για i από 1 μέχρι 4 κάνε
 $A[i] \leftarrow A[i+2]$
Τέλος_επανάληψης
 - h) $A[3] \leftarrow 2*B[6] + B[6/2]$;
 - i) $A \leftarrow \Gamma[1,4]$;
 - j) $A[1] \leftarrow B[(\Gamma[2,4] \bmod 6)+1]$;
- 2) Φτιάξτε ένα πρόγραμμα το οποίο:
 - a) θα διαβάσει 1000 αριθμούς
 - b) θα τυπώνει τους ίδιους, τα τετράγωνα τους και τις ρίζες τους, αν είναι θετικοί.
 - c) θα βρίσκει το μέγιστο και τον ελάχιστο
- 3) Να διαβασθούν 100 αριθμοί και να υπολογισθεί και να εκτυπωθεί το άθροισμα των απολύτων τιμών τους.
- 4) Να καταχωρηθούν 50 ακέραιοι αριθμοί σ' έναν πίνακα $A[50]$ και να βρεθεί πόσοι απ' αυτούς είναι άρτιοι.
- 5) Εκατό διαφορετικοί φούρνοι σε όλη την Ελλάδα δίνουν τις δικές τους τιμές για το κιλό του ψωμιού. Να κάνετε πρόγραμμα το οποίο
 - a) θα διαβάσει τις τιμές αυτές και θα τις αποθηκεύει σε ένα πίνακα πραγματικών 100 θέσεων
 - b) θα υπολογίζει τη μέση τιμή του ψωμιού.
- 6) Να διαβασθούν 30 αριθμοί, να βρεθεί ο μέσος όρος τους καθώς και πόσοι απ' αυτούς είναι μεγαλύτεροι από τον μέσο όρο όλων των αριθμών.
- 7) Να διαβασθούν 20 αριθμοί και να βρεθεί ο μεγαλύτερος καθώς και σε ποια θέση βρίσκεται. Μετά να γίνει ανταλλαγή του με τον αριθμό που βρίσκεται στην πρώτη θέση.
- 8) Να γραφεί αλγόριθμος ο οποίος διαβάζει τις ηλικίες 100 ανθρώπων και τις καταχωρεί σε έναν μονοδιάστατο πίνακα A. Κατόπιν υπολογίζει και εκτυπώνει
 - a) Το μέσο όρο των ηλικιών
 - b) Τη μέγιστη ηλικία
 - c) Το πλήθος των ανθρώπων που είναι άνω των 50 ετών
- 9) Να γίνει πρόγραμμα το οποίο
 - a) θα διαβάσει 10 ακεραίους και θα του αποθηκεύει σε ένα πίνακα ακεραίων $A[10]$
 - b) θα εκτυπώνει τους αριθμούς με την αντίστροφη σειρά που διαβάστηκαν.
- 10) Να γίνει πρόγραμμα το οποίο
 - a) θα διαβάσει 10 ακεραίους και θα του αποθηκεύει σε ένα πίνακα ακεραίων $A[10]$
 - b) θα αντιστρέφει τα περιεχόμενα του πίνακα

Αρχικός πίνακας	15	8	22	43	21	9	7	6	3	11
--------------------	----	---	----	----	----	---	---	---	---	----

Τελικός πίνακας	11	3	6	7	9	21	43	22	8	15
--------------------	----	---	---	---	---	----	----	----	---	----

- 11) Αν υποθέσουμε ότι κάθε μήνας έχει τριάντα ημέρες τότε όλες οι ημέρες του χρόνου θα ήταν 360. Μας δίνονται οι 360 θερμοκρασίες για όλες τις ημέρες του έτους. Να γίνει πρόγραμμα το οποίο:
- θα διαβάζει τις τιμές και θα τις αποθηκεύει σε ένα πίνακα Θ[360]
 - θα υπολογίζει πόσες μέρες είχαμε αρνητική, πόσες θετική και πόσες μηδενική θερμοκρασία.
 - θα βρίσκει ποια μέρα είχαμε την μεγαλύτερη θερμοκρασία
 - θα βρίσκει σε ποιο μήνα ανήκει αυτή η μέρα.
- 12) Δίνεται ένας μονοδιάστατος πίνακας ο οποίος περιέχει τις βαθμολογίες 50 μαθητών σε κάποιο μάθημα. Κάποιος μαθητής θεωρείται ότι απέτυχε στο συγκεκριμένο μάθημα αν ο βαθμός του είναι μικρότερος του 9.5, ενώ κάποιος μαθητής θεωρείται ότι άριστευσε αν ο βαθμός του είναι μεγαλύτερος ή ίσος του 19. Να γραφεί αλγόριθμος ο οποίος θα διαβάζει τα στοιχεία ενός τέτοιου πίνακα και θα υπολογίζει το πλήθος των μαθητών που απέτυχαν και το πλήθος των μαθητών που άριστευσαν
- 13) (Π) Δίνεται μονοδιάστατος πίνακας Π, Ν στοιχείων, που είναι ακέραιοι αριθμοί. Να αναπτύξετε αλγόριθμο, ο οποίος να ταξινομεί με τη μέθοδο της φυσαλίδας τα στοιχεία του πίνακα Π.
- 14) Να διαβασθούν 100 αριθμοί σ' έναν πίνακα και να ταξινομηθούν κατ' αύξουσα σειρά σύμφωνα με τον αλγόριθμο της ταξινόμησης με επιλογή (selection sort). Σύμφωνα με τον αλγόριθμο αυτό, βρίσκουμε πρώτα το μικρότερο στοιχείο του πίνακα και το ανταλλάσσουμε με το στοιχείο της πρώτης θέσης, μετά βρίσκουμε το μικρότερο από τα υπόλοιπα στοιχεία του πίνακα και το ανταλλάσσουμε με το στοιχείο της δεύτερης θέσης κοκ.
- 15) Να γίνει πρόγραμμα που να διαβάζει ένα πίνακα πραγματικών Π[20] εξασφαλίζοντας ότι ο πίνακας είναι ταξινομημένος σε αύξουσα τάξη μεγέθους. Αν ο χρήστης δώσει ένα στοιχείο που καταπατά αυτό το κανόνα θα τυπώνεται το μήνυμα «λάθος αριθμός» και θα ζητά ξανά το λανθασμένο στοιχείο.
- 16) Να γίνει πρόγραμμα που να διαβάζει δυο ταξινομημένους πίνακες ακεραίων Α[10] και Β[10], όπως στην άσκηση 6. Στη συνέχεια το πρόγραμμα θα συγχωνεύει του δύο πίνακες σε ένα ταξινομημένο πίνακα ακεραίων Γ[20] και θα τυπώνει τα περιεχόμενά του.
- 17) Ένας μαθητής έχει κρατήσει τις απουσίες που έχει κάνει ανά μέρα, στις 30 μέρες ενός μήνα. Φτιάξτε ένα πρόγραμμα που:
- να διαβάζει τις απουσίες που έκανε ο μαθητής ανά μέρα και να τις αποθηκεύει σε ένα πίνακα 30 θέσεων.
 - Αν κάθε μέρα ο μαθητής έχει 7 ώρες μάθημα, να υπολογίζει πόσες ολόκληρες μέρες έλειψε.
- 18) Ένας μαθητής έχει κρατήσει τις απουσίες που έχει κάνει ανά μέρα, στις 30 μέρες ενός μήνα. Φτιάξτε ένα πρόγραμμα που:
- να διαβάζει τις απουσίες που έκανε ο μαθητής ανά μέρα και να τις αποθηκεύει σε ένα πίνακα 30 θέσεων.
 - να επιστρέφει τον ίδιο πίνακα μόνο που σε κάθε θέση θα είναι αποθηκευμένες οι απουσίες του μαθητή μέχρι και εκείνη τη μέρα. Για παράδειγμα η θέση 10 δεν θα περιέχει τις απουσίες της 10ης μέρας (όπως διαβάζονται στην αρχή) αλλά συνολικά τις απουσίες μέχρι και τη 10η μέρα από την αρχή του μήνα.
 - να επιστρέφει τη μέρα που ο μαθητής υπερέβη το όριο των απουσιών(εφόσον το υπερέβη), αν το όριο είναι 100 απουσίες.

- 19) Να καταχωρηθούν σ' έναν πίνακα ακεραίων 30 θέσεων οι μετρήσεις μόλυνσης της ατμόσφαιρας για 30 ημέρες και να βρεθούν και να εκτυπωθούν οι ημέρες εκείνες που η μόλυνση παρουσιάζει κορυφή, δηλ. είναι μεγαλύτερη από τη μόλυνση της προηγούμενης και της επόμενης ημέρας. Η μόλυνση μπορεί να πάρει τιμή από 0 έως 100.
- 20) Θέλουμε να κρατήσουμε τα στοιχεία 100 εργαζομένων μιας εταιρίας και να τα επεξεργαστούμε. Για το λόγο αυτό θα φτιάξετε ένα πρόγραμμα το οποίο:
- θα διαβάζει τα ονόματα και τα επίθετα των εργαζομένων και θα τα αποθηκεύει σε δυο πίνακες χαρακτήρων ΟΝΟΜΑΤΑ[100] και ΕΠΙΘΕΤΑ[100].
 - θα διαβάζει τους μισθούς των εργαζομένων και θα τους αποθηκεύει σε ένα πίνακα πραγματικών ΜΙΣΘΟΙ[100].
 - θα τυπώνει τα ονοματεπώνυμα των εργαζομένων που ο μισθός τους ξεπερνά ένα όριο Κ που δίνεται από το χρήστη.
 - θα διαβάζει από το χρήστη ένα επίθετο, θα αναζητά το επίθετο στον αντίστοιχο πίνακα και αν το βρίσκει θα επιστρέφει όλα τα στοιχεία του εργαζομένου.
- 21) Να διαβασθούν τα στοιχεία ενός δισδιάστατου πίνακα με 10 γραμμές και 20 στήλες και να βρεθεί το άθροισμά τους.
- 22) Να διαβασθούν τα στοιχεία ενός δισδιάστατου πίνακα με 10 γραμμές και 20 στήλες και να βρεθεί ο μεγαλύτερος αριθμός καθώς και σε ποια γραμμή και στήλη βρίσκεται.
- 23) Να διαβασθούν τα στοιχεία ενός δισδιάστατου πίνακα με 10 γραμμές και 20 στήλες και να βρεθεί το άθροισμα των στοιχείων κάθε γραμμής και να καταχωρηθεί σ' έναν άλλον πίνακα 10 θέσεων.
- 24) Να διαβασθούν τα στοιχεία ενός δισδιάστατου πίνακα με 10 γραμμές και 20 στήλες και να βρεθεί το πλήθος των θετικών και των αρνητικών αριθμών κάθε στήλης του πίνακα.
- 25) Να διαβασθούν σ' έναν πίνακα 6 X 4 οι πωλήσεις των 6 υποκαταστημάτων μιας εταιρείας για 4 εβδομάδες και να βρεθεί η μεγαλύτερη πώληση, ποιο υποκατάστημα την έκανε και σε ποια εβδομάδα.
- 26) Να φτιάξετε ένα πρόγραμμα που να διαβάζει 100 αριθμούς και να επιστρέφει σε ένα πίνακα 100 X 2 τους αριθμούς που διαβάζει και το αποτέλεσμα της συνάρτησης $Y=\chi(\chi+2)$, όπου χ οι αριθμοί που διαβάζουμε.
- 27) (Π) Δίνεται πίνακας Π δύο διαστάσεων, που τα στοιχεία του είναι ακέραιοι αριθμοί με Ν γραμμές και Μ στήλες. Να αναπτύξετε αλγόριθμο που να υπολογίζει το ελάχιστο στοιχείο του πίνακα.
- 28) Να διαβασθούν οι δύο πλευρές α και b από 20 ορθογώνια παραλληλόγραμμα και να βρεθεί ποιο είναι το μεγαλύτερο εμβαδόν και σε ποιο ορθογώνιο ανήκει.
- 29) Σε ένα σχολείο έχουμε κρατήσει έναν πίνακα με τα 9 μαθήματα της Γ τάξης που εξετάζονται και τους βαθμούς των μαθητών. Φτιάξτε ένα πρόγραμμα που να διαβάζει αυτά τα στοιχεία και να βρίσκει ποιος μαθητής έχει την χαμηλότερη βαθμολογία και σε ποιο μάθημα. Θεωρούμε ότι οι μαθητές και τα μαθήματα είναι αριθμημένα από το 1 έως το Ν και από το 1 έως το 9 αντίστοιχα. Το πρόγραμμα θα επιστρέφει τον αριθμό του μαθητή και του μαθήματος.
- 30) Να γραφεί αλγόριθμος που θα διαβάζει τους βαθμούς του απολυτηρίου για 200 μαθητές του Λυκείου. Όποιοι βαθμοί είναι μικρότεροι από 10 θα τους καταχωρεί στον πίνακα ΑΠΟΡ και όσοι είναι μεγαλύτεροι ή ίσοι του 10 στον πίνακα ΠΡΟΑΓ
- 31) Να καταχωρηθούν σ' έναν πίνακα δύο διαστάσεων 10 X 12 οι εισπράξεις των 10 καταστημάτων μιας αλυσίδας εστιατορίων για τους 12 μήνες ενός έτους και να υπολογισθεί και να καταχωρηθεί σ' έναν άλλον μονοδιάστατο πίνακα 10 θέσεων ο μέσος όρος των εισπράξεων του κάθε καταστήματος για τους 12 μήνες. Επίσης να βρεθεί ο

μικρότερος μέσος όρος των εισπράξεων καθώς και ποιο ή ποια καταστήματα τον σημείωσαν.

- 32) Μια εταιρία έχει 100 υποκαταστήματα σε διάφορες πόλεις. Σε ένα πίνακα ΕΣΟΔΑ μεγέθους 12×100 , αποθηκεύονται τα έσοδα κάθε υποκαταστήματος στους μήνες ενός έτους. Φτιάξτε ένα πρόγραμμα το οποίο:
- θα διαβάζει τα περιεχόμενα του πίνακα ΕΣΟΔΑ
 - θα υπολογίζει το μέσο εισόδημα κάθε υποκαταστήματος και θα αποθηκεύει τα αποτελέσματα σε ένα πίνακα 100 θέσεων.
 - θα βρίσκει ποιο υποκατάστημα έχει το μεγαλύτερο μέσο όρο πωλήσεων.
 - θα υπολογίζει ποιο μήνα η εταιρία είχε συνολικά (από όλα τα υποκαταστήματα) τα περισσότερα έσοδα.
- 33) Σε μια ομάδα μπάσκετ ο προπονητής αποφάσισε να κρατά στον ηλεκτρονικό υπολογιστή, πίνακα με τους πόντους που πέτυχαν οι 12 παίκτες που έχει στους 18 αγώνες της αγωνιστικής περιόδου ώστε να μπορεί να επεξεργάζεται τα δεδομένα αυτά. Ανέθεσε σε εσάς να αναπτύξετε τον αλγόριθμο που θέλει. Από τις απαιτήσεις του προπονητή από τον αλγόριθμο προκύπτει ότι ο αλγόριθμος που θα κάνετε θα πρέπει :
- Να διαβάζει το όνομα του κάθε παίκτη και τους πόντους που πέτυχε σε κάθε αγώνα στην προηγούμενη αγωνιστική περίοδο.
 - Να εμφανίζει το όνομα του παίκτη με τον μεγαλύτερο μέσο όρο πόντων
 - Να εμφανίζει το όνομα του κάθε παίκτη και τον μεγαλύτερο αριθμό πόντων που πέτυχε σε ένα παιχνίδι από όλη την αγωνιστική περίοδο.
- 34) Να διαβασθούν σ' έναν πίνακα 20×6 οι 6 προσπάθειες 20 αθλητών στο άλμα εις μήκος και να βρεθεί η καλύτερη επίδοση κάθε αθλητή καθώς και η καλύτερη επίδοση όλων των αθλητών.
- 35) Να διαβασθούν τα στοιχεία 100 μαθητών (όνομα, βαθμός, τάξη) από τρεις τάξεις (a, b, c) και να εκτυπωθούν ανά τάξη ξεχωριστά
- 36) Δίνεται ένας πίνακας λεκτικών(χαρακτήρες) 20 θέσεων. Στον πίνακα έχουμε αποθηκεύσει τα ονόματα των 20 μαθητών ενός τμήματος. Δίνεται επίσης ένας πίνακας ακεραίων 20×10 στον οποίο αναγράφονται οι βαθμοί των 20 μαθητών σε 10 μαθήματα. Φτιάξτε ένα πρόγραμμα το οποίο να βρίσκει το μέσο όρο του απουσιολόγου και να τυπώνει το όνομά του.
- 37) Παλινδρομικός καλείται ένας ακεραίος αριθμός του οποίου τα ψηφία όταν διαβάζονται από το τελευταίο προς το πρώτο, προκύπτει πάλι ο ίδιος αριθμός με τον αρχικό, όπως είναι για παράδειγμα ο 13531. Να διαβασθεί ένα ακεραίος αριθμός 10 ψηφίων το πολύ και να βρεθεί αν είναι παλινδρομικός.
- 38) Για 50 μαθητές να καταχωρηθούν τα ονόματά τους σ' έναν μονοδιάστατο πίνακα και οι βαθμοί τους σε 10 μαθήματα σ' έναν διδιάστατο πίνακα. Να υπολογισθεί ο μέσος όρος της βαθμολογίας του κάθε μαθητή σ' έναν ξεχωριστό μονοδιάστατο πίνακα καθώς και ο μέσος όρος της βαθμολογίας όλων των μαθητών. Επίσης, να εμφανισθούν τα ονόματα και οι μέσοι όροι των μαθητών που ο μέσος όρος τους ξεπερνάει τον γενικό μέσο όρο.
- 39) Να καταχωρηθούν σ' έναν πίνακα δύο διαστάσεων 10×30 οι θερμοκρασίες 10 πόλεων για τις 30 ημέρες ενός μήνα και να υπολογισθεί και να καταχωρηθεί σ' έναν άλλον μονοδιάστατο πίνακα 10 θέσεων ο μέσος όρος των θερμοκρασιών της κάθε πόλης για τις 30 ημέρες. Επίσης, να βρεθεί ο μεγαλύτερος μέσος όρος των θερμοκρασιών καθώς και ποια ή ποιες πόλεις την σημείωσαν.
- 40) Να καταχωρηθούν σ' έναν πίνακα ακεραίων δύο διαστάσεων οι μετρήσεις μόλυνσης της ατμόσφαιρας 10 πόλεων της Ελλάδας για 30 ημέρες και να βρεθούν τα εξής :
- Οι μέσες τιμές μόλυνσης για κάθε πόλη και να καταχωρηθούν σ' έναν καινούργιο πίνακα 10 θέσεων.
 - Ποια πόλη παρουσίασε τον μεγαλύτερο μέσο όρο μόλυνσης.
 - Ποια πόλη παρουσίασε τον μικρότερο μέσο όρο μόλυνσης.

- d) Ποια πόλη και ποια ημέρα παρουσίασε τη μεγαλύτερη μόλυνση.
 e) Ποια πόλη και ποια ημέρα παρουσίασε τη μικρότερη μόλυνση.
 Η μόλυνση μπορεί να πάρει τιμή από 0 έως 100.

41) Να διαβασθεί ένας 6ψήφιος ακέραιος αριθμός και να εμφανισθεί ο μικρότερος αριθμός που να έχει τα ίδια ακριβώς ψηφία. Για παράδειγμα, αν ο αρχικός αριθμός είναι ο 543968, ο τελικός αριθμός που θα προκύψει θα είναι ο 345689. (Υπόδειξη : με τη βοήθεια των τελεστών DIV και MOD μπορείτε να απομονώσετε τα ψηφία του αριθμού για να τα καταχωρήσετε σ' έναν πίνακα 6 θέσεων, τον οποίο θα πρέπει μετά να ταξινομήσετε κατ' αύξουσα σειρά).

42) Θεωρούμε τις πόλεις και τις μεταξύ τους αποστάσεις που φαίνονται στον παρακάτω πίνακα (Οι αποστάσεις είναι σε χιλιόμετρα και είναι εικονικές). Έστω ότι μας δίνεται ένας πίνακας ακεραίων Αποστάσεις[5,5] με τα στοιχεία του παραπάνω πίνακα και ένας πίνακας λεκτικών Ονόματα[5], με τις τιμές που φαίνονται παρακάτω.

	Αθήνα	Θεσ/νίκη	Καβάλα	Κόρινθος	Καρδίτσα
Αθήνα	0	600	900	100	300
Θεσ/νίκη	600	0	300	700	200
Καβάλα	900	300	0	1000	500
Κόρινθος	100	700	1000	0	400
Καρδίτσα	300	200	500	400	0

1	2	3	4	5
Αθήνα	Θεσ/νίκη	Καβάλα	Κόρινθος	Καρδίτσα

Φτιάξτε ένα πρόγραμμα που να τυπώνει, χρησιμοποιώντας μόνο τους πίνακες «Αποστάσεις» και «Ονόματα», όλους του δυνατούς συνδυασμούς πόλεων και τις αποστάσεις τους με τη μορφή του παρακάτω παραδείγματος.

Αθήνα Θεσσαλονίκη 600

Προσοχή: Να μην εμφανίζονται οι συνδυασμοί δυο φορές. Για παράδειγμα να μην εμφανίζεται και το Θεσσαλονίκη Αθήνα 600, και να μην τυπώνει του συνδυασμούς μιας πόλης με τον εαυτό της.

43) (Π)Μια αλυσίδα κινηματογράφων έχει δέκα αίθουσες. Τα ονόματα των αιθουσών καταχωρούνται σε ένα μονοδιάστατο πίνακα και οι μηνιαίες εισπράξεις κάθε αίθουσας για ένα έτος καταχωρούνται σε πίνακα δύο διαστάσεων. Να γράψετε αλγόριθμο ο οποίος:

- να διαβάζει τα ονόματα των αιθουσών
- να διαβάζει τις μηνιαίες εισπράξεις των αιθουσών αυτού του έτους
- να υπολογίζει τη μέση μηνιαία τιμή των εισπράξεων για κάθε αίθουσα
- να βρίσκει και να εμφανίζει τη μικρότερη μέση μηνιαία τιμή
- να βρίσκει και να εμφανίζει το όνομα ή τα ονόματα των αιθουσών που έχουν την ανωτέρω μικρότερη μέση μηνιαία τιμή.

44) (Π)Κατά τη διάρκεια Διεθνών Αγώνων Στίβου στον ακοντισμό έλαβαν μέρος δέκα (10) αθλητές. Κάθε αθλητής έκανε έξι (6) έγκυρες ρίψεις που καταχωρούνται ως επιδόσεις σε μέτρα. Να αναπτύξετε αλγόριθμο, ο οποίος:

- εισάγει σε πίνακα δύο διαστάσεων τις επιδόσεις όλων των αθλητών
- υπολογίζει και καταχωρεί σε μονοδιάστατο πίνακα την καλύτερη από τις επιδόσεις κάθε αθλητή
- ταξινομεί τις καλύτερες επιδόσεις των αθλητών που καταχωρήθηκαν στο μονοδιάστατο πίνακα
- βρίσκει την καλύτερη επίδοση του αθλητή που πήρε το χάλκινο μετάλλιο (τρίτη θέση).

- 45) (Π) Δίνεται πίνακας Π δύο διαστάσεων, που τα στοιχεία του είναι ακέραιοι αριθμοί με N γραμμές και M στήλες. Να αναπτύξετε αλγόριθμο που να υπολογίζει το ελάχιστο στοιχείο του πίνακα.
- 46) (Π) Μια δισκογραφική εταιρεία καταγράφει στοιχεία για ένα έτος για κάθε ένα από τα 20 CDs που κυκλοφόρησε. Τα στοιχεία αυτά είναι ο τίτλος του CD, ο τύπος της μουσικής που περιέχει και οι μηνιαίες του πωλήσεις (ποσά σε ευρώ) στη διάρκεια του έτους. Οι τύποι μουσικής είναι δύο: «ορχηστρική» και «φωνητική».
- Να αναπτυχθεί αλγόριθμος ο οποίος:
- α. Για κάθε ένα από τα 20 CDs, να διαβάζει τον τίτλο, τον τύπο της μουσικής και τις πωλήσεις του για κάθε μήνα, ελέγχοντας την έγκυρη καταχώριση του τύπου της μουσικής.
 - β. Να εμφανίζει τον τίτλο ή τους τίτλους των CDs με τις περισσότερες πωλήσεις τον 3^ο μήνα του έτους.
 - γ. Να εμφανίζει τους τίτλους των ορχηστρικών CDs με ετήσιο σύνολο πωλήσεων τουλάχιστον 5000 ευρώ.
 - δ. Να εμφανίζει πόσα από τα CDs είχαν σύνολο πωλήσεων στο δεύτερο εξάμηνο μεγαλύτερο απ' ό,τι στο πρώτο.
- 47) (Π) Μια εταιρεία αποθηκεύει είκοσι (20) προϊόντα σε δέκα (10) αποθήκες. Να γράψετε πρόγραμμα στη γλώσσα προγραμματισμού "ΓΛΩΣΣΑ", το οποίο:
- a) περιέχει τμήμα δήλωσης των μεταβλητών του προγράμματος
 - b) εισάγει σε μονοδιάστατο πίνακα τα ονόματα των είκοσι προϊόντων
 - c) εισάγει σε πίνακα δύο διαστάσεων $\Pi[20,10]$ την πληροφορία που αφορά στην παρουσία ενός προϊόντος σε μια αποθήκη (καταχωρούμε την τιμή 1 στην περίπτωση που υπάρχει το προϊόν στην αποθήκη και την τιμή 0, αν το προϊόν δεν υπάρχει στην αποθήκη).
 - d) υπολογίζει σε πόσες αποθήκες βρίσκεται το κάθε προϊόν
 - e) τυπώνει το όνομα κάθε προϊόντος και το πλήθος των αποθηκών στις οποίες υπάρχει το προϊόν.