



Δομημένος Προγραμματισμός Γ' ΕΠΑ.Λ

Απαντήσεις των Ερωτήσεων
& Λύσεις των Ασκήσεων

Περιέχονται επίσης:

- Ενδεικτικός προγραμματισμός ύλης
- Υλοποίηση δραστηριοτήτων σε Pascal
v.1.0 (σχ. έτος 2009-10)

Για τη δημιουργία αυτού του βιβλίου συνεργάστηκαν οι:

Λιακέας Γεώργιος	Σχολικός Σύμβουλος Πληροφορικής Μεσσηνίας & Λακωνίας
Μητρόπουλος Γεώργιος	καθ. Πληροφορικής – 2 ^{ου} ΕΠΑ.Λ Καλαμάτας
Νόκας Γεώργιος	καθ. Πληροφορικής – 4 ^{ου} ΕΠΑ.Λ Καλαμάτας
Μελιγαλιώτης Παν.	καθ. Πληροφορικής – 1 ^{ου} ΕΠΑ.Λ Μεσσήνης
Βουδούρης Ευάγγελος	καθ. Πληροφορικής – 1 ^{ου} ΕΠΑ.Λ Κυπαρισσίας
Νάνος Ιωάννης	καθ. Πληροφορικής - 1 ^{ου} ΕΠΑ.Λ Σπάρτης
Βλαχάκης Νικόλαος	καθ. Πληροφορικής – 1 ^{ου} ΕΠΑ.Λ Μολάων
Πατσουράκος Δημήτριος	καθ. Πληροφορικής – 1 ^{ου} ΕΠΑ.Λ Γυθείου
Λυροπούλου Βιβή	καθ. Πληροφορικής – 1 ^{ου} ΕΠΑ.Λ Έλους Λακωνίας
Γκίνη Αγγελική	καθ. Πληροφορικής – 1 ^{ου} ΕΠΑ.Λ Βοιών

Για την εκτέλεση των ασκήσεων της Pascal, χρησιμοποιήθηκε το λογισμικό **PAME** (Pascal Made Easy) το οποίο έχει δημιουργήσει ο καθηγητής ΠΕ19 κ. **Γ. Σαμαρτζίδης** του ΕΠΑ.Λ Βοιών.

<http://users.lak.sch.gr/samartzidis/pame/>



«Είναι σχεδόν βέβαιο ότι το Σύμπαν – η γη, οι γαλαξίες και τα νεφελώματα, τα δημιουργήματα της ζωής – υπάρχει από καιρό και ίσως θα υπάρχει για πάντα. Αλλά αν ο παρατηρητής, δηλαδή ο άνθρωπος, παύσει να υπάρχει, η γένεση ή η εξέλιξη του Σύμπαντος δεν αφορά περίπου κανέναν ή, ακριβέστερα, δεν θα περιγράφεται μήτε θα θαυμάζεται από κανέναν. Θα πρόκειται για την ακραία, την ολική μοναξιά.»

(Από το βιβλίο του Γ. Γραμματικάκη **Η κόμη της Βερενίκης**)

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Γ'ΕΠΑ.Λ

3 ώρες Θεωρία - 2 ώρες Εργαστήριο

Ενδεικτικός προγραμματισμός ύλης

Θεωρία		Εργαστήριο	
Εβδομάδα		Εβδομάδα	
1η	Κεφ.1, 1/2 Κεφ.2	1η	Κεφάλαιο 8 - Η γλώσσα Pascal
2η	1/2 Κεφ.2	2η	Κεφάλαιο 8 - Η γλώσσα Pascal
3η	Κεφ.3, §.4.1	3η	Κεφάλαιο 8 - Η γλώσσα Pascal
4η	§ 4.2	4η	Κεφάλαιο 9 - Βασικές Εντολές
5η	§. 4.3	5η	Κεφάλαιο 9 - Βασικές Εντολές
6η	§ 4.4.1 - Ακολουθία	6η	Κεφάλαιο 9 - Βασικές Εντολές
7η	§ 4.4.2 - Επιλογή	7η	Κεφάλαιο 9 - Βασικές Εντολές
8η	§ 4.4.2 - Επιλογή	8η	§ 10.1 - IF
9η	§ 4.4.3 - Επανάληψη	9η	§ 10.1 - IF
10η	§ 4.4.3 - Επανάληψη	10η	§ 10.2 - CASE
11η	§ 4.4.3 - Επανάληψη	11η	§ 10.2 - CASE
12η	Ασκήσεις § 4.4	12η	§ 11.1, 11.2 - WHILE
13η	Ασκήσεις § 4.4	13η	§ 11.2 - WHILE
14η	Ασκήσεις § 4.4	14η	§ 11.2 - WHILE
15η	Ασκήσεις § 4.4	15η	§ 11.3 - REPEAT UNTIL
16η	Ασκήσεις § 4.4	16η	§ 11.3 - REPEAT UNTIL
17η	Ασκήσεις § 4.4	17η	§ 11.3 - REPEAT UNTIL
18η	Ασκήσεις § 4.4	18η	§ 11.4 - FOR
19η	Ασκήσεις § 4.4	19η	§ 11.4 - FOR
20η	Κεφάλαιο 5	20η	§ 12.1.1 - Διαδικασίες οριζόμενες από το χρήστη
21η	Κεφάλαιο 5	21η	§ 12.1.1 - Διαδικασίες οριζόμενες από το χρήστη
22η	Κεφάλαιο 6	22η	§ 12.2.1 - Συναρτήσεις οριζόμενες από το χρήστη
23η	Κεφάλαιο 6	23η	§ 12.2.1 - Συναρτήσεις οριζόμενες από το χρήστη
24η	Κεφάλαιο 7	24η	§ 12.2.1 - Συναρτήσεις οριζόμενες από το χρήστη
25η	Κεφάλαιο 7	25η	§ 13.1 Κατηγορίες τύπων δεδομένων
26η	Κεφάλαιο 7	26η	§ 13.2 Συναρτήσεις τακτικών τύπων

Κεφάλαιο 1

ΚΑΤΑΝΟΗΣΗ ΚΑΘΟΡΙΣΜΟΣ ΚΑΙ ΔΟΜΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Ερωτήσεις

1. Τι καλείται κατανόηση προβλήματος;

Κατανόηση του προβλήματος είναι η εργασία κατά την οποία μελετάμε προσεκτικά το περιεχόμενο του προβλήματος ώστε να **διακρίνουμε το συσχετισμό των δεδομένων με τα ζητούμενά του**, αποσαφηνίζοντας ακόμα και παρερμηνείες οι οποίες προκύπτουν από τη διατύπωσή του.

2. Πότε επιβάλλεται να χωρίσουμε ένα πρόβλημα σε άλλα απλούστερα;

Επιβάλλεται να χωρίσουμε ένα πρόβλημα σε άλλα απλούστερα, όταν είναι μεγάλο ή όταν απαιτεί πολύπλοκο τρόπο επίλυσης.

3. Τι καλείται δομή προβλήματος;

Δομή προβλήματος ονομάζονται τα απλούστερα προβλήματα από τα οποία αποτελείται καθώς και ο τρόπος με τον οποίον συνδέονται, έτσι ώστε η επίλυσή τους να συνθέτει τη λύση του αρχικού προβλήματος.¹

4. Τι καλείται καθορισμός προβλήματος;

Καθορισμός προβλήματος είναι ο προσδιορισμός των δεδομένων και των ζητούμενων του προβλήματος. Ο καθορισμός του προβλήματος ακολουθεί την εργασία της ανάλυσης του προβλήματος σε απλούστερα.²

5. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

α. Ένα πρόβλημα το χωρίζουμε σε άλλα απλούστερα όταν είναι **μεγάλο** ή όταν έχει **πολύπλοκο** τρόπο επίλυσης.

β. Η επίλυση ενός προβλήματος προϋποθέτει την **κατανόησή** του.

γ. Τα προβλήματα δεν παραμένουν **στατικά** με το πέρασμα του χρόνου αλλά τα στοιχεία τους μεταβάλλονται **δυναμικά**.

δ. Μετά την κατανόηση του προβλήματος και διάκρισή του σε απλούστερα, δεν απομένει παρά να προσδιορίσουμε τα **δεδομένα** του προβλήματος καθώς και τα **ζητούμενα**.

6. Να σημειώσετε με Σωστό – Λάθος τις παρακάτω προτάσεις:

α. Η διοργάνωση μιας εκδρομής αποτελεί πρόβλημα που απαιτεί επίλυση. (Σ)

β. Η σαφήνεια στη διατύπωση ενός προβλήματος είναι δεδομένη. (Λ)

γ. Κάθε πρόβλημα μπορεί να επιλυθεί με τη χρήση υπολογιστή (Λ)

¹ Σημ. Χρειάζεται καλύτερη διατύπωση για τον όρο «δομή προβλήματος», τόσο στη σελ. 5 όσο και στη σελίδα 6 στην Ανακεφαλαίωση.

² Σημ. Στην Ανακεφαλαίωση εισάγεται η έννοια «Καθορισμός απαιτήσεων προβλήματος» η οποία εξετάζεται στο Κεφάλαιο 2.

Ασκήσεις

1. Δίνεται ο αριθμός των νικών και των ηττών μιας ομάδας καλαθοσφαίρισης σε μια αγωνιστική περίοδο. Η ομάδα βαθμολογείται για κάθε νίκη με δύο βαθμούς και για κάθε ήττα με έναν. Να υπολογίσετε τη συνολική βαθμολογία της ομάδας και να καθορίσετε τον πίνακα δεδομένων και ζητούμενων και των σχέσεων που τα συνδέει.

Λύση

Δεδομένα:	
	Όνομα ομάδας
	Έδρα
	Αριθμός νικών
	Αριθμός ηττών
	Βαθμοί νίκης
	Βαθμοί ήττας

Ζητούμενα	
	Συνολική βαθμολογία της ομάδας

Σχέσεις που συνδέουν δεδομένα και ζητούμενα	
	$\text{Συνολική βαθμολογία} = \text{αριθμός νικών} * \text{βαθμοί νίκης} + \text{αριθμός ηττών} * \text{βαθμοί ήττας}$

2. Δίνονται το ονοματεπώνυμο, η διεύθυνση, το τηλέφωνο και το πλήθος των Kwh ρεύματος που κατανάλωσε ένας πελάτης της ΔΕΗ. Η ηλεκτρική εταιρεία κοστολογεί τα πρώτα 100 Kwh³ προς 0,15€ το ένα και τα υπόλοιπα προς 0,2€. Ζητείται να υπολογιστεί το ποσό που θα καταβάλει ο καταναλωτής. Να καθορίσετε τον πίνακα δεδομένων και ζητούμενων και των σχέσεων που τα συνδέουν.

Λύση

³ Το Kwh είναι η συνήθης μονάδα μέτρησης της ηλεκτρικής ενέργειας. Στο βιβλίο αναφέρεται επανειλημμένα η μονάδα μέτρησης Kw, η οποία όμως είναι μονάδα ισχύος.

Δεδομένα:	
	Όνοματεπώνυμο καταναλωτή
	Διεύθυνση
	Τηλέφωνο
	Πλήθος Kwh κατανάλωσης
	Τιμή1/ Kwh για κατανάλωση ≤ 100 Kwh
	Τιμή2/ Kwh για κατανάλωση >100 Kwh

Ζητούμενα	
	Πληρωτέο ποσό

Συνθήκη	
	Η Τιμή2/ Kwh χρεώνεται εφόσον η κατανάλωση ξεπερνά τα 100 Kwh και για τον αριθμό των Kwh, τα οποία είναι πάνω από τα 100.

Σχέσεις που συνδέουν δεδομένα και ζητούμενα	
	Αριθμός Kwh με Τιμή1 = Ο αριθμός των Kwh κατανάλωσης που είναι μικρότερος ή ίσος του 100
	Αριθμός Kwh με Τιμή2 = Πλήθος Kwh κατανάλωσης – 100
	Πληρωτέο ποσό = Αριθμός Kwh με Τιμή1 * Τιμή1/ Kwh + Αριθμός Kwh με Τιμή2 * Τιμή2/ Kwh

3. Δίνονται οι δικαιολογημένες και οι αδικαιολόγητες απουσίες ενός μαθητή με δεδομένο ότι το όριο των δικαιολογημένων απουσιών είναι 50 και το όριο των αδικαιολόγητων 60. Να ελέγξετε αν ο μαθητής προάγεται ή απορρίπτεται λόγω απουσιών. Να καθορίσετε πίνακα δεδομένων και ζητουμένων και των σχέσεων που τα συνδέουν.

Λύση

Δεδομένα:	
	Όνοματεπώνυμο μαθητή
	Διεύθυνση
	Τηλέφωνο
	Αριθμός δικαιολογημένων απουσιών
	Αριθμός αδικαιολόγητων απουσιών

Ζητούμενα	
	Προαγωγή ή απόρριψη του μαθητή

Συνθήκη	
	Οι δικαιολογημένες απουσίες πρέπει να είναι λιγότερες ή ίσες με 50
	Οι αδικαιολόγητες απουσίες πρέπει να είναι λιγότερες ή ίσες με 60

Σχέσεις που συνδέουν δεδομένα και ζητούμενα	
	Προάγεται ⁴ : δικαιολογημένες απ. +αδικαιολόγητες απ. ≤ 110
	Απορρίπτεται: Το άθροισμα των δικαιολ. απ. + αδικ.απ. >110

⁴Η απάντηση που δίνεται είναι στο «πνεύμα» του βιβλίου (δείτε και σελ. 24,28). **Καλύτερα θα ήταν να έμπαιναν οι σχέσεις:** 1) Προάγεται: Οι δικαιολογημένες απουσίες είναι ≤ 50 ΚΑΙ οι αδικαιολόγητες απουσίες είναι ≤ 60 . 2) Απορρίπτεται: Οι δικαιολογημένες απουσίες είναι ≤ 50 ΚΑΙ οι αδικαιολόγητες είναι > 60 .

Κεφάλαιο 2

ΚΑΘΟΡΙΣΜΟΣ ΑΠΑΙΤΗΣΕΩΝ

Ερωτήσεις

1. Πότε εξυπηρετεί η λύση ενός προβλήματος με τη βοήθεια υπολογιστή⁵;

Η χρήση υπολογιστή για την λύση ενός προβλήματος εξυπηρετεί όταν:

- Έχουμε μεγάλο πλήθος δεδομένων να επεξεργαστούμε.
- Τα αποτελέσματα που παράγονται είναι πολλά.
- Η μέθοδος επίλυσης είναι πολύπλοκη για τον άνθρωπο.
- Η μέθοδος επίλυσης επαναλαμβάνεται πολλές φορές.

2. Τι ονομάζουμε δεδομένα ενός προβλήματος;

Δεδομένα ενός προβλήματος ονομάζεται το **σύνολο** εκείνων των **στοιχείων**, τα οποία είναι **απαραίτητα για την επίλυση του προβλήματος**.

3. Τι καλείται καθορισμός απαιτήσεων ενός προβλήματος⁶;

Καθορισμός των απαιτήσεων του προβλήματος είναι η εργασία η οποία περιλαμβάνει τη σαφή **καταγραφή των δεδομένων**, την αναλυτική **διατύπωση του προβλήματος**, τη λεπτομερειακή **περιγραφή των ζητούμενων** και τη **διερεύνηση των αποτελεσμάτων**.

4. Ποιοι είναι οι παράγοντες που επηρεάζουν τον καθορισμό των απαιτήσεων για την επίλυση ενός προβλήματος;

Οι **παράγοντες** που επηρεάζουν τον καθορισμό των απαιτήσεων ενός προβλήματος είναι:

- Η **επιλογή δεδομένων** που έχουν άμεση σχέση με το πρόβλημά μας.
- Ο έλεγχος **ορθότητας των δεδομένων** του προβλήματος.
- Ο έλεγχος **πληρότητας των δεδομένων** σε σχέση με το ζητούμενο αποτέλεσμα του προβλήματος.
- Ο καθορισμός των **ζητούμενων** του προβλήματος.

5. Γιατί η καταχώριση των δεδομένων ενός προβλήματος στον υπολογιστή δεν αποτελεί μέρος του καθορισμού των απαιτήσεων ενός προβλήματος;

Η καταχώριση των δεδομένων⁷ του προβλήματος (σ.σ. αν γίνεται σωστά ή όχι) δεν έχει σχέση με τον ορισμό (καταγραφή) των δεδομένων που είναι εργασία η οποία γίνεται κατά τον καθορισμό των απαιτήσεων του προβλήματος. [Για παράδειγμα, στο πρόβλημα της εξαγωγής του ΜΟ των μαθημάτων ενός μαθητή Γυμνασίου, ΟΡΙΖΟΥΜΕ ότι οι βαθμοί πρέπει να βρίσκονται στην κλίμακα από 1 μέχρι 20. Αυτό είναι διαφορετικό πράγμα από τη σωστή καταχώριση της βαθμολογίας].

6. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

α. Εκτός από την κατανόηση του προβλήματος εξίσου σημαντικός είναι και ο **ακριβής προσδιορισμός** των δεδομένων.

β. Σημαντικό είναι να ελέγχουμε την **ορθότητα** των δεδομένων σε σχέση με το επιδιωκόμενο αποτέλεσμα, καθώς και την **πληρότητα** των δεδομένων σε σχέση με τα απαραίτητα για την επίλυση του προβλήματος.

⁵ Δεν αναφέρεται κάτι πάνω σε αυτό στο Κεφάλαιο 2 του βιβλίου.

⁶ Στην Ανακεφαλαίωση του Κεφαλαίου 2 περιγράφεται ο «καθορισμός απαιτήσεων του προβλήματος» (η ίδια έννοια έχει αναφερθεί και στην Ανακεφαλαίωση του Κεφαλαίου 1) με τρόπο παρόμοιο με εκείνον της έννοιας «καθορισμός προβλήματος» στη λεζάντα της σ.6. Πρόκειται προφανώς(;) για την ίδια έννοια.

⁷ Data Entry.

7. Να σημειώσετε με Σ- Λ τις παρακάτω προτάσεις:

- α. Ο έλεγχος της ορθότητας των δεδομένων αποτελεί μέρος της κατανόησης του προβλήματος (Λ – βλέπε Ανακεφαλαίωση Κεφαλαίου 1 – στάδια επίλυσης προβλήματος).
- β. Οι πληροφορίες που προέρχονται από την επεξεργασία των δεδομένων ενός προβλήματος μπορούν να χρησιμοποιηθούν ως δεδομένα για το ίδιο πρόβλημα (Σ –βλέπε πίνακα, σελ.10 και σελ. 11 σειρές 2-3).
- γ. Ο έλεγχος της ορθής καταχώρησης δεδομένων ενός προβλήματος στον υπολογιστή , αποτελεί μέρος του καθορισμού των απαιτήσεων του προβλήματος; (Λ – βλέπε σελ. 11-3^η παράγραφο).

Ασκήσεις

Στα παρακάτω προβλήματα να καθορίσετε τα δεδομένα τους και να προτείνετε εφόσον είναι δυνατόν, ελέγχους για την ορθότητα και την πληρότητά τους:

1. Δίνεται ο αριθμός των νικών μιας ομάδας καλαθοσφαίρισης σε μια αγωνιστική περίοδο, καθώς και ο αριθμός των ηττών της. Η ομάδα βαθμολογείται για κάθε νίκη με δύο βαθμούς και για κάθε ήττα με έναν. Να υπολογιστεί η συνολική βαθμολογία της ομάδας.

Λύση

Δεδομένα:	
	Όνομα ομάδας
	Έδρα
	Αριθμός νικών
	Αριθμός ηττών
	Βαθμοί νίκης
	Βαθμοί ήττας
	Συνολικός αριθμός αγώνων

Έλεγχος ορθότητας:

- Ο βαθμός σε κάθε αγώνα επιτρέπεται να είναι είτε 2 (νίκη) είτε 1 (ήττα).
- Ο αριθμός νικών επιτρέπεται να είναι από 0 μέχρι το συνολικό αριθμό αγώνων του πρωταθλήματος.
- Ο αριθμός ηττών επιτρέπεται να είναι από 0 μέχρι το συνολικό αριθμό αγώνων του πρωταθλήματος.
- Το άθροισμα του αριθμού νικών και του αριθμού ηττών πρέπει να είναι ίσο με το συνολικό αριθμό αγώνων του πρωταθλήματος.

Έλεγχος πληρότητας:

- Πρέπει να υπάρχει στα δεδομένα η Έδρα της κάθε ομάδας ώστε να διακρίνονται ομάδες με τυχόν ίδιο όνομα (π.χ. Ολυμπιακός Πειραιώς και Ολυμπιακός Βόλου).
- Πρέπει να υπάρχει στα δεδομένα ο συνολικός αριθμός αγώνων του πρωταθλήματος ώστε να μπορεί να γίνεται έλεγχος για αγώνες οι οποίοι έχουν αναβληθεί και εκκρεμεί η διεξαγωγή τους.

2. Δίνονται το ονοματεπώνυμο, η διεύθυνση, το τηλέφωνο και το πλήθος των Kwh ρεύματος που κατανάλωσε ένας πελάτης της ΔΕΗ. Η ηλεκτρική εταιρεία κοστολογεί τα πρώτα 100 Kwh προς 0,15€ το ένα και τα υπόλοιπα προς 0,2€. Ζητείται να υπολογιστεί το ποσό που θα καταβάλει ο καταναλωτής. Να καθορίσετε τον πίνακα δεδομένων και ζητούμενων και των σχέσεων που τα συνδέουν.

Λύση

Δεδομένα:	
	Αριθμός Μετρητή ΔΕΗ
	Ονοματεπώνυμο καταναλωτή
	Διεύθυνση
	Τηλέφωνο
	Περίοδος κατανάλωσης
	Πλήθος Kwh κατανάλωσης
	Τιμή1/ Kwh για κατανάλωση \leq 100 Kwh
	Τιμή2/ Kwh για κατανάλωση $>$ 100 Kwh

Έλεγχος ορθότητας:

- Οι χρεώσεις για τα Kwh κατανάλωσης επιτρέπεται να είναι είτε 0,15€ είτε 0,2€.
- Οι χρεώσεις με 0,2€ θα γίνονται μόνο για τα Kwh, τα οποία θα υπερβαίνουν τα 100.

Έλεγχος πληρότητας:

- Πρέπει να υπάρχει στα δεδομένα ο Αριθμός Μετρητή του ρολογιού της ΔΕΗ ώστε να υπάρχει διάκριση μεταξύ καταναλωτών με το ίδιο όνομα και την ίδια διεύθυνση.
- Πρέπει να υπάρχει στα δεδομένα η Περίοδος κατανάλωσης ώστε να υπάρχει αντιστοιχία Περιόδου κατανάλωσης – ηλεκτρικής ενέργειας που καταναλώθηκε.

3. Δίνονται οι δικαιολογημένες και οι αδικαιολόγητες απουσίες ενός μαθητή. Να ελέγξετε αν ο μαθητής προάγεται ή απορρίπτεται λόγω απουσιών γνωρίζοντας ότι το όριο των δικαιολογημένων απουσιών είναι 50 και το όριο των αδικαιολόγητων 60.

Λύση

Δεδομένα:	
	A.M. μαθητή
	Επώνυμο
	Όνομα
	Πατρώνυμο
	Διεύθυνση
	Τηλέφωνο
	Συνολικός αριθμός απουσιών
	Αριθμός δικαιολογημένων απουσιών

Έλεγχος ορθότητας:

- Ο αριθμός των δικαιολογημένων απουσιών επιτρέπεται να είναι από 0 έως 50.

- Το άθροισμα του αριθμού των δικαιολογημένων και του αριθμού των αδικαιολόγητων απουσιών πρέπει να είναι ίσο με το συνολικό αριθμό των απουσιών.

Έλεγχος πληρότητας:

- Πρέπει να υπάρχει στα δεδομένα ο Αριθμός Μητρώου του μαθητή ώστε τα στοιχεία του να καταχωρούνται με μοναδικό τρόπο.
- Πρέπει να υπάρχει στα δεδομένα το Πατρώνυμο του μαθητή ώστε να είναι δυνατή η καταχώρηση απουσιών σε περίπτωση που κάποιοι μαθητές έχουν το ίδιο όνομα και επώνυμο.
- Πρέπει να υπάρχει στα δεδομένα ο Συνολικός αριθμός των απουσιών ώστε να είναι δυνατός ο υπολογισμός των αδικαιολόγητων απουσιών σε σχέση με τις δικαιολογημένες.

4. Δίνεται το μήκος και το πλάτος μιας διδακτικής αίθουσας, να υπολογίσετε το εμβαδόν της.

Λύση

Δεδομένα:	
	Αριθμός Διδακτικής Αίθουσας
	Όροφος
	Μήκος
	Πλάτος

Ζητούμενα	
	Εμβαδόν αίθουσας

Συνθήκη	
	Το εμβαδόν υπολογίζεται εφόσον και οι δύο διαστάσεις έχουν δοθεί στις ίδιες μονάδες μήκους (π.χ. μέτρα)

Σχέσεις που συνδέουν δεδομένα και ζητούμενα	
	$\text{Εμβαδόν} = \text{μήκος} * \text{πλάτος}$

Έλεγχος ορθότητας:

- Και οι δύο διαστάσεις πρέπει να εκφράζονται με τις ίδιες μονάδες.

Έλεγχος πληρότητας:

- Πρέπει να καταχωρούνται στα δεδομένα ο αριθμός της αίθουσας καθώς και ο όροφος στον οποίο βρίσκεται ώστε να μην υπάρχουν αμφιβολίες σε ποια αίθουσα αντιστοιχεί το εμβαδόν που υπολογίστηκε.

Κεφάλαιο 3

ΑΛΓΟΡΙΘΜΟΙ – ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Ερωτήσεις

1. Τι καλείται αλγόριθμος;

Αλγόριθμος καλείται η διαδοχική σειρά πεπερασμένου πλήθους λογικών βημάτων τα οποία εκτελούνται σε πεπερασμένο χρόνο και τα οποία απαιτούνται για την επίλυση ενός προβλήματος.

2. Ποια είναι τα χαρακτηριστικά ενός καλού αλγορίθμου;

Τα χαρακτηριστικά ενός καλού αλγορίθμου είναι:

- Να περιγράφει με **ακρίβεια τα δεδομένα** που ο ίδιος θα χρησιμοποιήσει.
- Κάθε βήμα του να είναι τελείως **καθορισμένο** και να φανερώνει τι ακριβώς κάνει.
- Να δίνει με **ακρίβεια τα αποτελέσματα** που παράγονται από το κάθε βήμα του.
- Να **ολοκληρώνει πάντοτε σωστά τη σκέψη επίλυσης** του προβλήματος ύστερα από έναν **πεπερασμένο** αριθμό βημάτων, που το καθένα εκτελείται σε **πεπερασμένο χρόνο**.

3. Να συμπληρώσετε τα κενά με τις λέξεις που λείπουν.

α. Απαραίτητο και σπουδαίο είναι η **καταγραφή της σκέψης** για την επίλυση ενός προβλήματος.

β. Η διαδοχική σειρά **πεπερασμένου πλήθους** λογικών βημάτων που απαιτούνται για την επίλυση ενός προβλήματος καλείται αλγόριθμος.

Κεφάλαιο 4

ΑΝΑΠΤΥΞΗ ΑΛΓΟΡΙΘΜΩΝ

Ερωτήσεις

1. Τι καλείται ψευδοκώδικας;

Ψευδοκώδικας καλείται η καταγραφή του αλγορίθμου με απλές **λεκτικές** προστακτικές **προτάσεις** και **σύμβολα** τα οποία προέρχονται από διαδοσμένες γλώσσες προγραμματισμού.

2. Τι είναι το λογικό διάγραμμα;

Λογικό διάγραμμα είναι η μέθοδος αναπαράστασης του αλγορίθμου με τη χρήση απλών **γεωμετρικών σχημάτων** τα οποία υποστηρίζονται με **απλές λέξεις** για την αναπαράσταση συγκεκριμένων λειτουργιών του αλγορίθμου.

3. Για ποιο λόγο είναι απαραίτητη η τυποποίηση του αλγορίθμου;

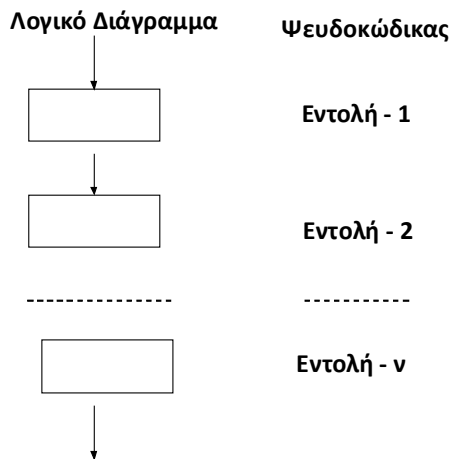
Η τυποποίηση του αλγορίθμου επίλυσης ενός προβλήματος είναι απαραίτητη ώστε να υπάρχει μια **κοινή γλώσσα έκφρασης και επικοινωνίας**, όσο το δυνατόν περισσότερων ανθρώπων που ασχολούνται με την επίλυση προβλημάτων με τη χρήση υπολογιστή. Έτσι ο αλγόριθμος είναι δυνατόν να είναι **κατανοητός και από άλλους**, οι οποίοι θα τον μελετήσουν, αλλά και από τον ίδιο το **δημιουργό του** όταν μετά από κάποιο καιρό θελήσει να θυμηθεί ή να τροποποιήσει το αλγόριθμο, τον οποίο κατασκεύασε.

4. Ποιες είναι οι βασικές αλγοριθμικές δομές;

Οι βασικές αλγοριθμικές δομές είναι η ακολουθία, η επιλογή και η επανάληψη.

5. Ποια είναι η δομή της ακολουθίας;

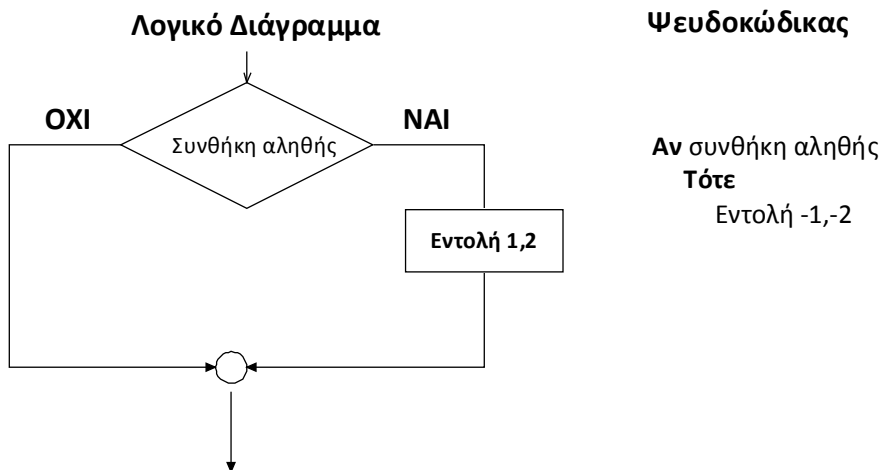
Στην ακολουθία, οι εντολές οι οποίες βρίσκονται σε διαδοχή, εκτελούνται με τη σειρά με την οποία είναι γραμμένες.



6. Ποια είναι η δομή της επιλογής;

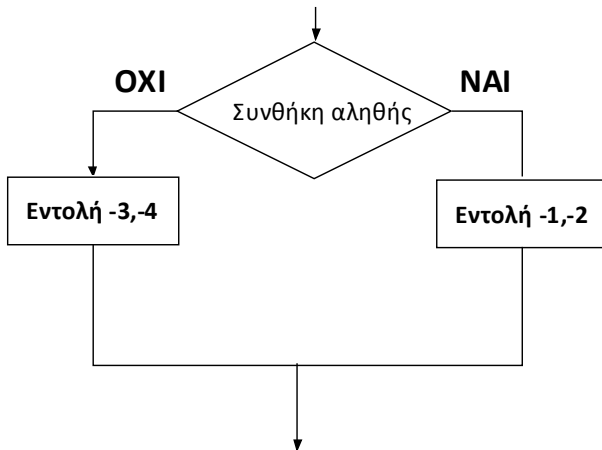
Με τη δομή επιλογής παρέχεται η δυνατότητα εκτέλεσης μιας ή περισσότερων εντολών **ανάλογα με το αποτέλεσμα ελέγχου μιας συνθήκης**.

Απλή Επιλογή



Διπλή Επιλογή

Λογικό Διάγραμμα

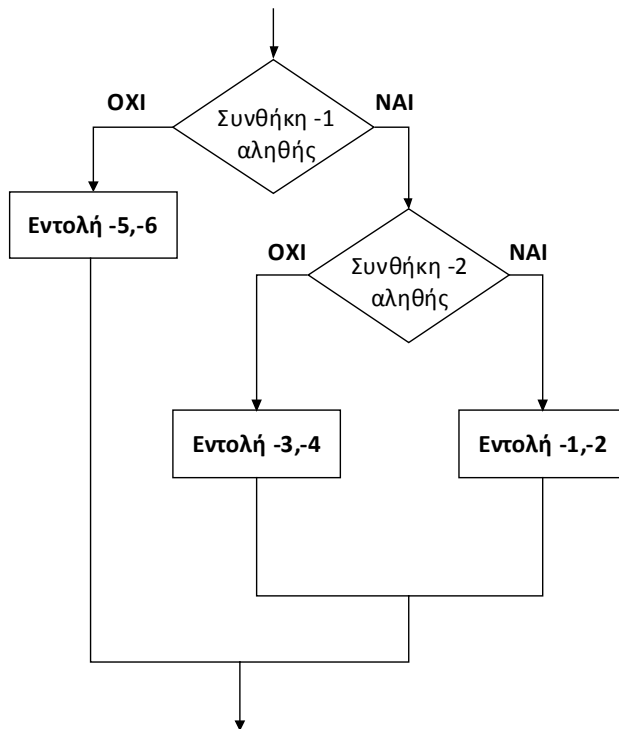


Ψευδοκώδικας

Αν συνθήκη αληθής
Τότε
 Εντολή -1,-2
Αλλιώς
 Εντολή -3,-4

Επιλογή μέσα σε άλλη επιλογή

Λογικό Διάγραμμα



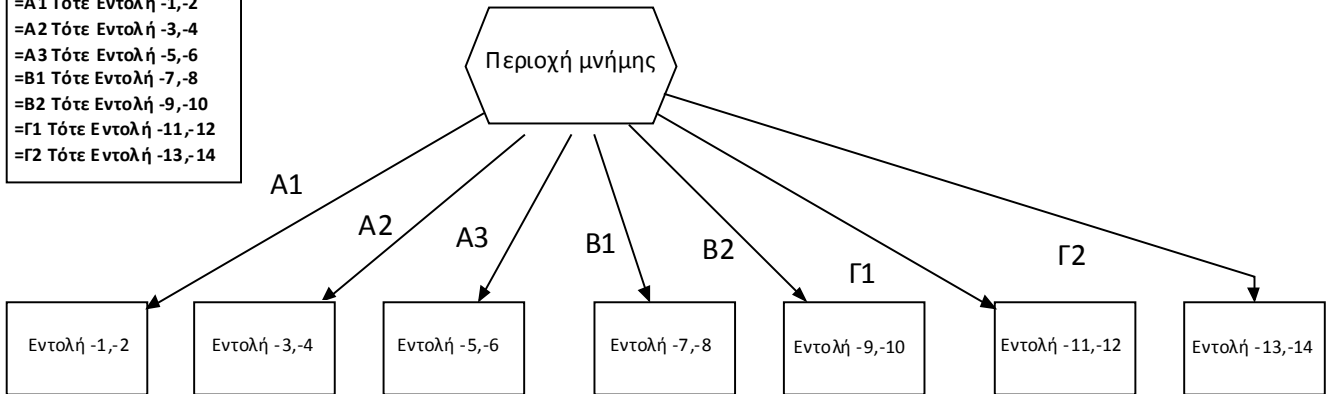
Ψευδοκώδικας

Αν συνθήκη -1 αληθής
Τότε
 Αν συνθήκη -2 αληθής
 Τότε
 Εντολή -1,-2
 Αλλιώς
 Εντολή -3,-4
Αλλιώς
 Εντολή -5,-6

Πολλαπλή Επιλογή

Ψευδοκώδικας

Αν περιοχή μνήμης
 =A1 Τότε Εντολή -1,-2
 =A2 Τότε Εντολή -3,-4
 =A3 Τότε Εντολή -5,-6
 =B1 Τότε Εντολή -7,-8
 =B2 Τότε Εντολή -9,-10
 =Γ1 Τότε Εντολή -11,-12
 =Γ2 Τότε Εντολή -13,-14

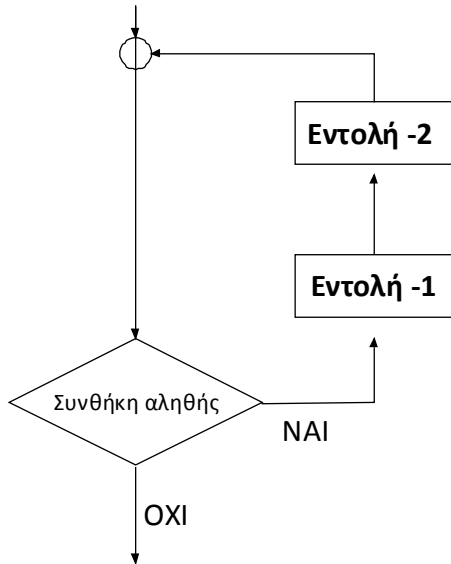


7. Να γράψετε σε ψευδοκώδικα και λογικό διάγραμμα τη δομή της επανάληψης.

Υπάρχουν τρεις μορφές δομής επανάληψης, οι δύο είναι με συνθήκη ενώ η τρίτη είναι προκαθορισμένη επανάληψη.

Επανάληψη με συνθήκη I

Λογικό Διάγραμμα



Ψευδοκώδικας

Εφόσον (συνθήκη αληθής) **Επανάλαβε**

Αρχή

Εντολή -1

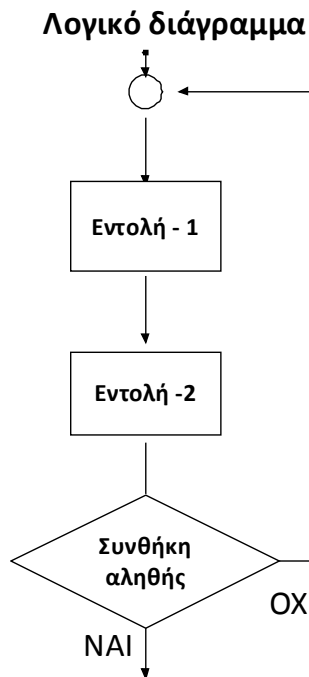
Εντολή -2

.....

Εντολή -n

Τέλος

Επανάληψη με συνθήκη II



Ψευδοκώδικας

Επανάλαβε

Εντολή -1

Εντολή -2

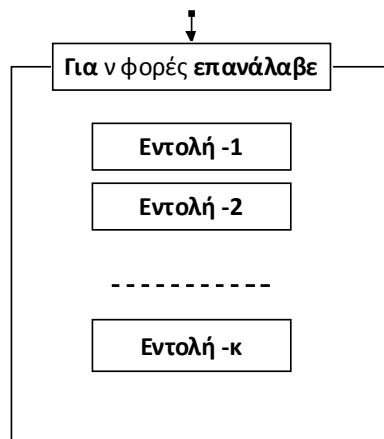
.....

Εντολή -n

μέχρι η συνθήκη να είναι αληθής

Προκαθορισμένη επανάληψη

Λογικό διάγραμμα



Ψευδοκώδικας

Για n φορές επανάλαβε

Αρχή

Εντολή -1

Εντολή -2

.....

Εντολή -κ

Τέλος

8. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

α. Είναι απαραίτητο να γίνει μια τυποποίηση του τρόπου που θα αναπαρίστανται οι αλγόριθμοι.

β. Ο ψευδοκώδικας διαδόθηκε περισσότερο, όταν και οι γλώσσες προγραμματισμού έγιναν απλούστερες και απέκτησαν έναν τρόπο έκφρασης, πολύ κοντά στον καθημερινό τρόπο επικοινωνίας μας.

γ. Με την εντολή Επανάλαβε ... μέχρι έχουμε εκτέλεση τουλάχιστον μία φορά, των εντολών που περιέχει, έστω και αν η συνθήκη είναι αληθής.

9. Να σημειώσετε με Σ-Λ τις παρακάτω προτάσεις:

- α. Η καταγραφή του αλγορίθμου είναι απαραίτητη βοήθεια για την επίλυση ενός προβλήματος. (Σ- προφανώς).
- β. Ο ψευδοκώδικας χρησιμοποιεί σχήματα για την αναπαράσταση του αλγορίθμου (Λ).
- γ. Στη δομή επανάληψης **επανάλαβε εφόσον** εκτελούνται τουλάχιστον μια φορά οι εντολές που περιέχονται. (Λ)

Ασκήσεις

1. Δίνονται το ονοματεπώνυμο του κάθε μαθητή μιας τάξης και η βαθμολογία του στο μάθημα της Πληροφορικής και θέλουμε να υπολογίσουμε το μέσο όρο της βαθμολογίας των μαθητών της τάξης στο μάθημα αυτό. Να παραστήσετε με ψευδοκώδικα τον αλγόριθμο του προβλήματος.

Λύση

Αρχή

Διάβασε τα δεδομένα: πλήθος_μαθητών

Άθροισμα = 0

Για πλήθος_μαθητών φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: Επώνυμο_μαθητή

Όνομα_μαθητή

Βαθμός_μαθητή

Άθροισμα=Άθροισμα+Βαθμός_Μαθητή

Τέλος

Αν πλήθος_μαθητών <>0

Τότε

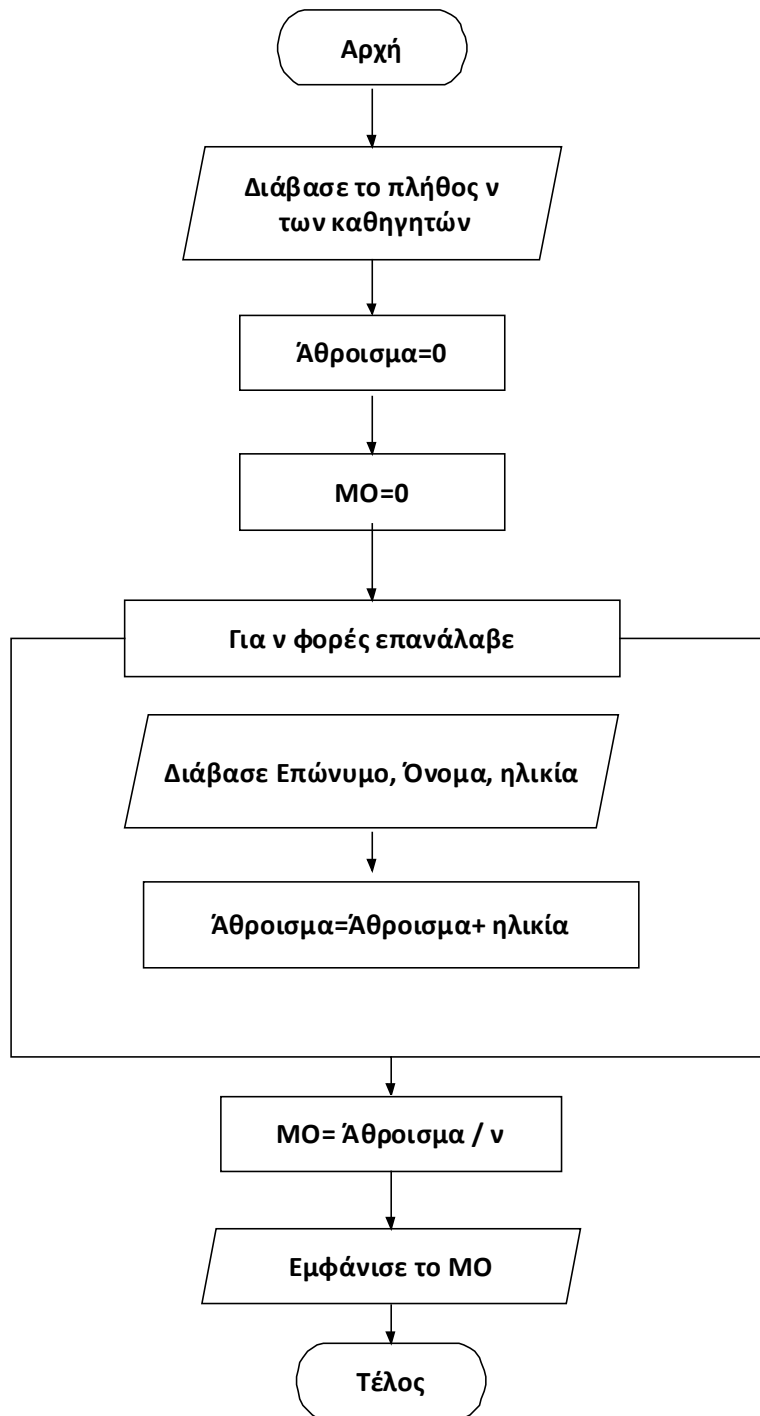
ΜΟ=Άθροισμα / πλήθος_μαθητών

Εμφάνισε το ΜΟ

Τέλος

2. Δίνεται η ηλικία καθενός από τους καθηγητές του σχολείου μας και ζητείται να υπολογιστεί ο μέσος όρος της ηλικίας τους. Να παραστήσετε με λογικό διάγραμμα τον αλγόριθμο του προβλήματος.

Λύση



3. Δίνονται το ονοματεπώνυμο κάθε μαθητή μιας τάξης και η βαθμολογία του στο μάθημα της Πληροφορικής. Ζητείται να υπολογίσουμε το πλήθος των μαθητών της τάξης που έχουν βαθμολογία σε αυτό το μάθημα μεγαλύτερη από 18. Να παραστήσετε με ψευδοκώδικα τον αλγόριθμο του προβλήματος.

Λύση

Αρχή

Διάβασε τα δεδομένα: πλήθος_μαθητών

Για πλήθος_μαθητών φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: Επώνυμο_μαθητή

Όνομα_μαθητή

Βαθμός_μαθητή

Αν Βαθμός_μαθητή > 18

Τότε

μετρητής=μετρητής+1

Τέλος

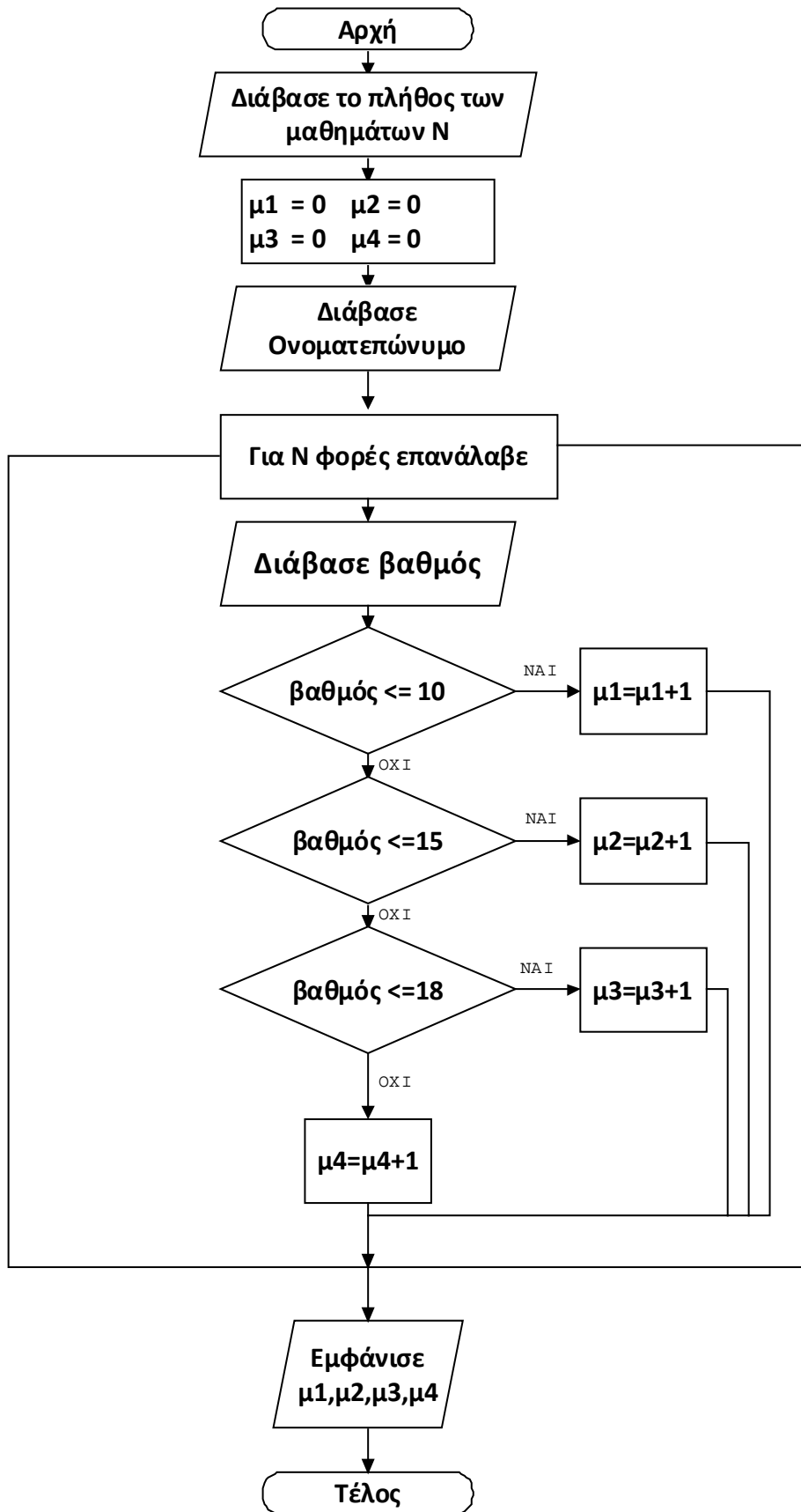
Εμφάνισε μετρητής

Τέλος

4. Δίνονται οι βαθμοί του τριμήνου ενός μαθητή σε όλα τα μαθήματά του. Να υπολογίσετε το πλήθος των μαθημάτων του που έχουν βαθμολογία έως 10, το πλήθος των μαθημάτων , που έχουν βαθμολογία από 11-15, από 16-18 και από 19 έως 20. Να παραστήσετε με λογικό διάγραμμα τον αλγόριθμο του προβλήματος.

Λύση

- Αφού η άσκηση δε ζητά έλεγχο, θεωρούμε ότι οι βαθμοί καταχωρούνται σωστά και βρίσκονται στην κλίμακα από 1 έως 20.
- Η μεταβλητή N κρατά το πλήθος των μαθημάτων.
- Οι μετρητές μ1,μ2,μ3,μ4 κρατούν τον αριθμό των μαθημάτων που είναι μικρότερα ή ίσα με 10, από 11 έως και 15, 16 έως και 18, 19 έως και 20 αντίστοιχα.



5. Στον Πανελλήνιο Διαγωνισμό Πληροφορικής που γίνεται κάθε χρόνο μπορούν να συμμετάσχουν μαθητές από κάθε τάξη του Γυμνασίου, των ΕΠΑ.Λ και του ΓΕΛ. Αν μας δίνεται η τάξη και ο τύπος του σχολείου που φοιτά ο μαθητής, να υπολογιστεί το πλήθος των μαθητών που συμμετέχουν από κάθε τάξη. Να παραστήσετε με ψευδοκώδικα τον αλγόριθμο του προβλήματος.

Λύση

(Αφού ζητείται ο αριθμός από κάθε τάξη θα πρέπει να χρησιμοποιήσουμε εννέα μεταβλητές μια και δεν έχουμε δυνατότητα χρησιμοποιήσουμε πίνακα. Οι μεταβλητές είναι οι $\gamma_1, \gamma_2, \gamma_3$ για τις τρεις τάξεις του Γυμνασίου, $\lambda_1, \lambda_2, \lambda_3$ για τις τρεις τάξεις του ΓΕΛ, $\epsilon_1, \epsilon_2, \epsilon_3$ για τις τρεις τάξεις του ΕΠΑ.Λ. Το πλήθος μαθητών παριστάνουμε με n . Επίσης, δε ζητείται από την άσκηση να γίνει έλεγχος καταχώρησης στοιχείων).

Αρχή

$\gamma_1, \gamma_2, \gamma_3, \lambda_1, \lambda_2, \lambda_3, \epsilon_1, \epsilon_2, \epsilon_3 = 0$

Διάβασε τα δεδομένα: n

Για n φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: Ονοματεπώνυμο_μαθητή

Σχολείο

Τάξη

Αν Σχολείο = Γυμνάσιο

Τότε

Αν Τάξη = Α

Τότε

$\gamma_1 = \gamma_1 + 1$

Αλλιώς

Αν Τάξη = Β

Τότε

$\gamma_2 = \gamma_2 + 1$

αλλιώς

$\gamma_3 = \gamma_3 + 1$

Αλλιώς

Αν Σχολείο = ΓΕΛ

Τότε

Αν Τάξη = Α

Τότε

$\lambda_1 = \lambda_1 + 1$

Αλλιώς

Αν Τάξη = Β

Τότε

$\lambda_2 = \lambda_2 + 1$

Αλλιώς

$\lambda_3 = \lambda_3 + 1$

Αλλιώς

Αν Τάξη = A

Τότε

$$\varepsilon_1 = \varepsilon_1 + 1$$

Αλλιώς

Αν Τάξη = B

Τότε

$$\varepsilon_2 = \varepsilon_2 + 2$$

Αλλιώς

$$\varepsilon_3 = \varepsilon_3 + 1$$

Τέλος

Εμφάνισε $\gamma_1, \gamma_2, \gamma_3, \lambda_1, \lambda_2, \lambda_3, \varepsilon_1, \varepsilon_2, \varepsilon_3$

Τέλος

6. Σε μια εταιρεία γίνονται κρατήσεις, από το μισθό του υπαλλήλου υπέρ του συνδικαλιστικού του σωματείου, ανάλογα με το ύψος του μισθού του. Για μισθό έως 440€ η κράτηση είναι 1%, έως 735€ 1.5%, έως 880€ 2%, έως 1135€ 2.5% και για πάνω από 1135€ 3.5%. Δίνονται για κάθε υπάλληλο το ονοματεπώνυμό του και ο μισθός του. Να υπολογιστεί το ποσό κράτησης που αντιστοιχεί στον κάθε υπάλληλο. Να παραστήσετε με ψευδοκώδικα τον αλγόριθμο του προβλήματος. [Έχει γίνει προσαρμογή των μισθών από δραχμές σε ευρώ σε σχέση με τα δεδομένα που υπάρχουν στο βιβλίο].

Λύση

[Δεν διευκρινίζεται αν οι κρατήσεις γίνονται στο σύνολο του μισθού ή σε τμήματά του. Π.χ στη 2^η περίπτωση, η κράτηση 1.5% θα γίνει στο σύνολο του μισθού 735 ή στα πρώτα 440€ θα κρατηθεί 1% και στα υπόλοιπα 1.5%; Θα θεωρήσουμε το δεύτερο.]

Αρχή

Διάβασε τα δεδομένα: πλήθος_υπαλλήλων

Για πλήθος_υπαλλήλων φορές **επανάλαβε**

Διάβασε τα δεδομένα: ονοματεπώνυμο

μισθός

Αν μισθός <= 440

Τότε

$$\text{κρατήσεις} = 0.01 * \text{μισθός}$$

Αλλιώς

Αν μισθός <= 735

Τότε

$$\text{κρατήσεις} = 440 * 0.01 + 0.015 * (\text{μισθός} - 440)$$

Αλλιώς

Αν μισθός <= 880

Τότε

$$\text{κρατήσεις} = 440 * 0.01 + (735 - 440) * 0.015 + 0.02 * (\text{μισθός} - 735)$$

Αλλιώς

Αν μισθός <=1135

Τότε

κρατήσεις = $440 * 0.01 + (735 - 440) * 0.015 + (880 - 735) * 0.02 + 0.025 * (\text{μισθός} - 880)$

Αλλιώς

κρατήσεις = $440 * 0.01 + (735 - 440) * 0.015 + (880 - 735) * 0.02 + (1135 - 880) * 0.025 + 0.035 * (\text{μισθός} - 1135)$

Εμφάνισε ονοματεπώνυμο

Εμφάνισε κρατήσεις

Τέλος

Τέλος

7. Δίνονται οι ψήφοι που έλαβε ο κάθε υποψήφιος μαθητής για το 15-μελές συμβούλιο του σχολείου. Ζητείται να υπολογιστεί η επί τοις εκατό αντιστοιχία τους⁸. Να αναπαραστήσετε με ψευδοκώδικα τον αλγόριθμο του προβλήματος.

Λύση

[Η μεταβλητή **πλήθος_υποψηφίων** αντιστοιχεί στο πλήθος των υποψηφίων. Η μεταβλητή **ψήφοι** αντιστοιχεί στον αριθμό των ψήφων, που έλαβε ο καθένας. Η μεταβλητή **αριθμός_έγκυρων ψηφοδελτίων** αντιστοιχεί στον αριθμό των έγκυρων ψηφοδελτίων και θεωρείται θετικός ακέραιος. Η μεταβλητή **ποσοστό** κρατά το ποσοστό των ψήφων που αντιστοιχεί σε κάθε υποψήφιο.]

Αρχή

Διάβασε τα δεδομένα: πλήθος_υποψηφίων

αριθμός_έγκυρων ψηφοδελτίων

Για πλήθος_υποψηφίων φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: ονοματεπώνυμο_υποψηφίου

ψήφοι

ποσοστό = $\text{ψήφοι} * 100 / \text{αριθμός_έγκυρων_ψηφοδελτίων}$

Εμφάνισε ονοματεπώνυμο_υποψηφίου, ποσοστό

Τέλος

Τέλος

8. Δίνεται ο τύπος ενός οχήματος (μηχανή, ΙΧ επιβατηγό, ΙΧ φορτηγό, ΔΧ επιβατηγό, ΔΧ φορτηγό) και το μήκος του. Η χρέωση (τιμές σε €) σε ένα οχηματαγωγό πλοίο είναι ανάλογη του τύπου του οχήματος και του μήκους του ως εξής:

⁸ Η διατύπωση της άσκησης είναι προβληματική. Θα πρέπει να δίνεται ο αριθμός των έγκυρων ψηφοδελτίων. Ο αριθμός των μαθητών που ψήφισαν έγκυρα δεν μπορεί να προσδιοριστεί από το άθροισμα των ψήφων των υποψηφίων διότι στο ψηφοδέλτιο μπορεί να μπουν πολλοί σταυροί ή κανένας.

ΜΗΚΟΣ σε μέτρα	Μηχανή	ΙΧ επιβατηγό	ΔΧ επιβατηγό	ΙΧ φορτηγό	ΔΧ φορτηγό
Έως 2	1.5	3	6	9	12
3	1.5	6	9	12	15
5		9	12	15	18
10				18	21
περισσότερα					24

Δίνονται οι αριθμοί των οχημάτων της κάθε κατηγορίας, που επιβιβάσθηκαν στο οχηματαγωγό πλοίο. Να υπολογιστεί το συνολικό ποσό είσπραξης. Σε όποια κατηγορία δεν αντιστοιχεί ποσό, σημαίνει ότι δεν μπορεί να υπάρχει και αντίστοιχος τύπος οχήματος. Να παραστήσετε με ψευδοκώδικα τον αλγόριθμο του προβλήματος.

Λύση

[πλήθος_μ το πλήθος των μηχανών, πλήθος_ΙΧε το πλήθος των ΙΧ επιβατηγών, πλήθος_ΔΧε το πλήθος των ΔΧ επιβατηγών, πλήθος_ΙΧφ το πλήθος των ΙΧ φορτηγών, πλήθος_ΔΧφ το πλήθος των ΔΧ φορτηγών, σπ το συνολικό ποσό είσπραξης]

[Δε γίνεται έλεγχος για την ακρίβεια καταχώρησης των στοιχείων αφού δε ζητείται από την άσκηση]

Αρχή

Διάβασε τα δεδομένα: πλήθος_μ, πλήθος_ΙΧε, πλήθος_ΔΧε, πλήθος_ΙΧφ, πλήθος_ΔΧφ
σπ=0

Για πλήθος_μ φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: μήκος
σπ=σπ+1.5

Τέλος

Για πλήθος_ΙΧε φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: μήκος
Αν μήκος ≤ 2

Τότε

σπ=σπ+3

Αλλιώς

Αν μήκος ≤ 3

Τότε

σπ=σπ+6

Αλλιώς

σπ=σπ+9

Τέλος

Για πλήθος_ΔΧε φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: μήκος
Αν μήκος ≤ 2

Τότε

σπ=σπ+6

Αλλιώς

Αν μήκος ≤ 3

Τότε

$\sigma\pi = \sigma\pi + 9$

Αλλιώς

$\sigma\pi = \sigma\pi + 12$

Τέλος

Για πλήθος_ΙΧφ φορές επανάλαβε

Αρχή

Διάβασε τα δεδομένα: μήκος

Αν μήκος ≤ 2

Τότε

$\sigma\pi = \sigma\pi + 9$

Αλλιώς

Αν μήκος ≤ 3

Τότε

$\sigma\pi = \sigma\pi + 12$

Αλλιώς

Αν μήκος ≤ 5

Τότε

$\sigma\pi = \sigma\pi + 15$

Αλλιώς

$\sigma\pi = \sigma\pi + 18$

Τέλος

Για πλήθος_ΔΧφ φορές επανάλαβε

Αρχή

Διάβασε τα δεδομένα: μήκος

Αν μήκος ≤ 2

Τότε

$\sigma\pi = \sigma\pi + 12$

Αλλιώς

Αν μήκος ≤ 3

Τότε

$\sigma\pi = \sigma\pi + 15$

Αλλιώς

Αν μήκος ≤ 5

Τότε

$\sigma\pi = \sigma\pi + 18$

Αλλιώς

Αν μήκος ≤ 10

Τότε

$\sigma\pi = \sigma\pi + 21$

Αλλιώς

$\sigma\pi = \sigma\pi + 24$

Τέλος

Εμφάνισε $\sigma\pi$

Τέλος

Παρατήρηση

Σε περίπτωση που θέλουμε να εξετάζεται και η περίπτωση όπου κάποιο όχημα ξεπερνά το μήκος, το οποίο δίνεται στον πίνακα και να εμφανίζεται ανάλογο μήνυμα, μπορούμε να τροποποιήσουμε τον αλγόριθμο όπως παρακάτω:

Αρχή

Διάβασε τα δεδομένα: πλήθος_μ, πλήθος_ΙΧε, πλήθος_ΔΧε, πλήθος_ΙΧφ, πλήθος_ΔΧφ
σπ=0

Για πλήθος_μ φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: μήκος

Αν μήκος ≤ 3

σπ=σπ+1.5

Αλλιώς

Εμφάνισε το μήνυμα «Δεν υπάρχει τέτοιος τύπος μηχανής»

Τέλος

Για πλήθος_ΙΧε φορές **επανάλαβε**

Αρχή

Διάβασε τα δεδομένα: μήκος

Αν μήκος ≤ 2

Τότε

σπ=σπ+3

Αλλιώς

Αν μήκος ≤ 3

Τότε

σπ=σπ+6

Αλλιώς

Αν μήκος ≤ 5

σπ=σπ+9

Αλλιώς

Εμφάνισε το μήνυμα «Δεν υπάρχει τέτοιος τύπος ΙΧε»

Τέλος

.....

Συνεχίζουμε με τον ίδιο τρόπο και για το υπόλοιπο τμήμα του αλγορίθμου.

Κεφάλαιο 5

ΣΤΟΙΧΕΙΑ ΔΟΜΗΜΗΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Ερωτήσεις

1. Τι καλείται τμηματικός προγραμματισμός;

Τμηματικός προγραμματισμός ονομάζεται η εργασία που ακολουθεί την ιεραρχική σχεδίαση και είναι η υλοποίηση του προγράμματος μέσα από ανεξάρτητες λογικές ενότητες – τμήματα. Τα μεμονωμένα τμήματα πρέπει να είναι όσο το δυνατόν μικρότερα ώστε να είναι εύκολο να διορθωθούν.

2. Τι καλείται ιεραρχικός προγραμματισμός;

Ιεραρχικός προγραμματισμός καλείται η τεχνική σχεδίασης αλγορίθμου κατά την οποία το πρόβλημα **διασπάται** σε μια σειρά από **απλούστερα προβλήματα**, τα οποία όταν επιλυθούν, οδηγούν στην επίλυση του αρχικού προβλήματος. Η σχεδίαση του προγράμματος πρέπει να προχωρά από **πάνω προς τα κάτω**. Στα ανώτερα επίπεδα ξεκινάμε από τα **γενικά** και προχωρούμε σε κατώτερα επίπεδα **προσθέτοντας λεπτομέρειες**. Η ανάλυση σε κατώτερα επίπεδα σταματά όταν η επεξεργασία περιέχει τόσες λεπτομέρειες ώστε να μπορεί να κωδικοποιηθεί σε μια γλώσσα προγραμματισμού.

3. Τι καλείται δομημένος προγραμματισμός;

Δομημένος προγραμματισμός καλείται η τεχνική σχεδίασης αλγορίθμου και του αντίστοιχου προγράμματος, η οποία χρησιμοποιεί τις αρχές του **ιεραρχικού** και του **τμηματικού προγραμματισμού**. Στον δομημένο προγραμματισμό ο **αλγόριθμος** αποτελείται από **ανεξάρτητα τμήματα** με βάση ένα προκαθορισμένο σχέδιο. Έτσι, με τον ίδιο τρόπο θα αναπτυχθεί αμέσως μετά και η κωδικοποίηση του αλγορίθμου σε γλώσσα προγραμματισμού δημιουργώντας τελικά το πρόγραμμα. Ο δομημένος προγραμματισμός χρησιμοποιεί μόνο τις βασικές αλγοριθμικές δομές της **ακολουθίας**, της **επιλογής** και της **επανάληψης** ενώ αποφεύγει τη χρήση της εντολής GO TO. Ένα δομημένο πρόγραμμα διαβάζεται και συντηρείται εύκολα.

4. Ποια είναι τα πλεονεκτήματα του δομημένου προγραμματισμού;

Τα κύρια πλεονεκτήματα του δομημένου προγραμματισμού είναι:

- Διευκόλυνση στην ανάπτυξη αλγορίθμων **κατά τμήματα**.
- Ευκολία και ταχύτητα στην **κωδικοποίηση**.
- Καλύτερη **ποιότητα** προγραμμάτων.
- Ευκολία στις **διορθώσεις** και τη συντήρηση.
- **Τεκμηρίωση** που περιέχεται σχεδόν εξ ολοκλήρου στο ίδιο το πρόγραμμα.

5. Τι καλείται διάγραμμα HIPO;

Το διάγραμμα HIPO⁹ (Hierarchy Input Process Output = Ιεραρχία Εισόδου – Επεξεργασίας – Εξόδου) είναι μια τεχνική υλοποίησης του Δομημένου Προγραμματισμού και των αρχών του Ιεραρχικού και του Τμηματικού προγραμματισμού με τη

⁹ Η απόδοση του όρου HIPO (Hierarchical Input Output Processing) που δίνεται στις σελίδες 39 και 41 χρειάζεται διόρθωση σε (Hierarchy plus Input Process Output).

χρήση **διαγραμμάτων**. Ένα διάγραμμα ΗΙΡΟ αποτελείται από **τρία βασικά τμήματα**, τα οποία περιγράφουν την είσοδο των δεδομένων, την επεξεργασία και τα αποτελέσματα που απαιτεί ένας αλγόριθμος και το αντίστοιχο πρόγραμμα. Στα διαγράμματα αυτά εμφανίζονται τα διάφορα **επίπεδα ιεραρχίας** και τα τμήματα του αλγορίθμου – προγράμματος με **αριθμό σύμφωνα με τη σειρά εκτέλεσής τους**. Στο ανώτερο επίπεδο υπάρχει μια περιγραφή της όλης διαδικασίας, στο αμέσως επόμενο επίπεδο διακρίνονται σε τμήματα οι τρεις βασικές λειτουργίες (είσοδος, επεξεργασία, έξοδος), οι οποίες μπορεί να αναλύονται περαιτέρω σε άλλα υποτμήματα.

6. Ποια είναι τα τρία βασικά τμήματα ενός διαγράμματος ΗΙΡΟ;

Τα τρία βασικά τμήματα ενός διαγράμματος ΗΙΡΟ είναι αυτά που περιγράφουν την είσοδο των δεδομένων, την επεξεργασία και τα αποτελέσματα που απαιτεί ένας αλγόριθμος.

7. Ποιες είναι οι βασικές αρχές που χρησιμοποιεί ο δομημένος προγραμματισμός;

Ο δομημένος προγραμματισμός χρησιμοποιεί τις βασικές αρχές του **ιεραρχικού** και του **τμηματικού προγραμματισμού**.

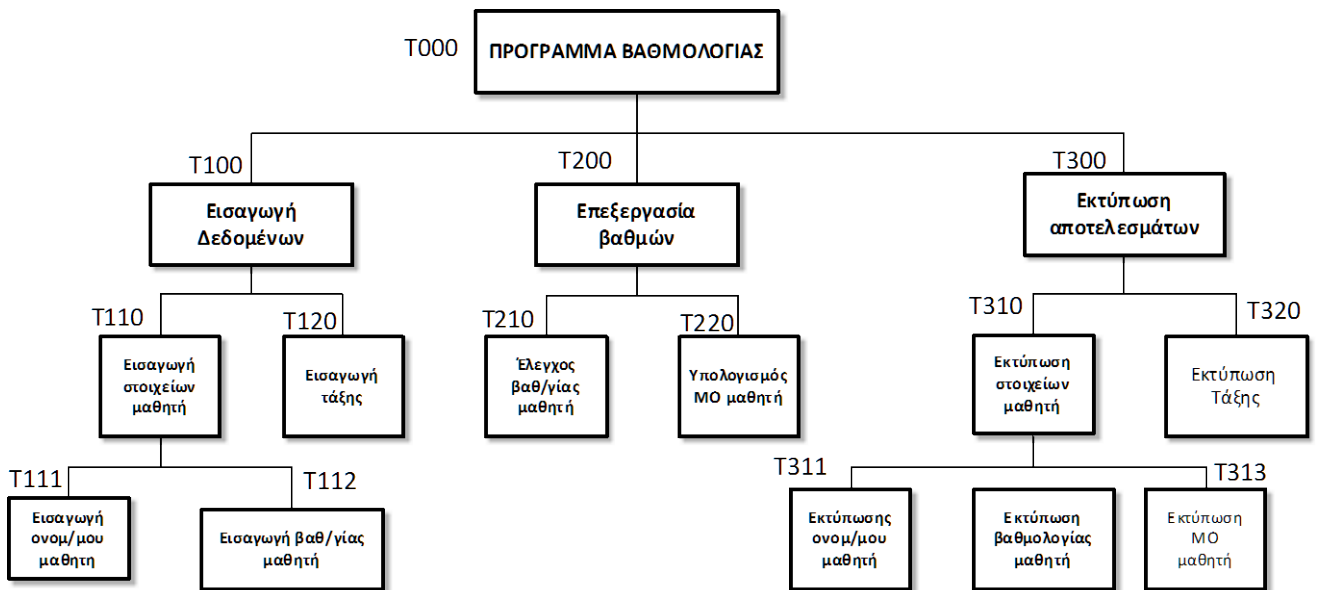
8. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

- α. Ο δομημένος προγραμματισμός προϋποθέτει την **ανάπτυξη** του αλγορίθμου.
- β. Ο δομημένος προγραμματισμός αποτελείται από **ανεξάρτητα** μέρη με βάση ένα προκαθορισμένο σχέδιο.
- γ. Ένα δομημένο πρόγραμμα είναι πολύ **πιο εύκολο** στην κατανόηση από προγράμματα που είναι γραμμένα με διαφορετικό τρόπο.
- δ. Ένα δομημένο πρόγραμμα διαβάζεται εύκολα και επειδή αποτελείται από σαφώς καθορισμένα **και ανεξάρτητα** τμήματα, έχει εύκολη και τη **συντήρησή** του.
- ε. Οι βασικές αλγοριθμικές δομές δεν είναι αρκετές για την κωδικοποίηση ενός **δομημένου** αλγορίθμου, αλλά πρέπει να αναπτυχθούν χρησιμοποιώντας συγκεκριμένες **τεχνικές**.
- στ. Ένα διάγραμμα ΗΙΡΟ αποτελείται από τρία βασικά τμήματα, τα οποία περιγράφουν την **είσοδο** των δεδομένων, την **επεξεργασία** και τα **αποτελέσματα (έξοδο)**.

Ασκήσεις

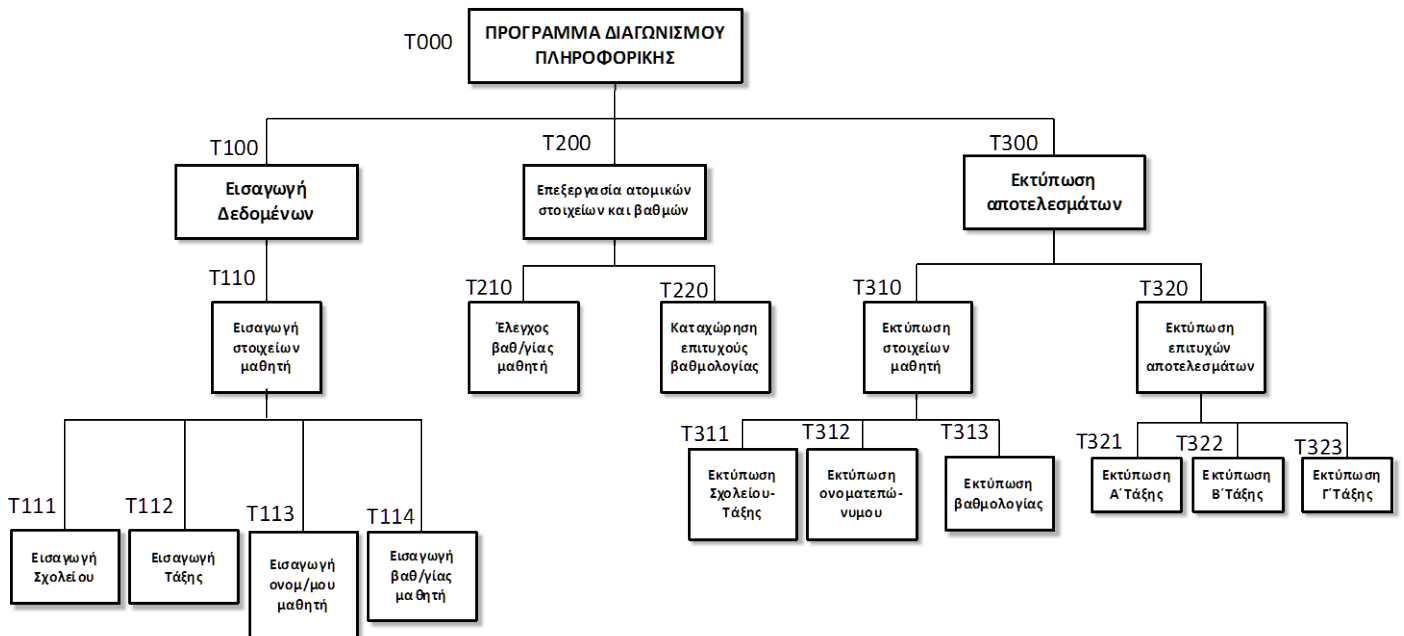
1. Δίνονται το ονοματεπώνυμο κάθε μαθητή μιας τάξης και η βαθμολογία του σε κάθε μάθημά του και θέλουμε να υπολογίσουμε το μέσο όρο της βαθμολογίας όλων των μαθητών της τάξης. Να σχεδιάσετε το διάγραμμα ΗΙΡΟ του προβλήματος.

Λύση



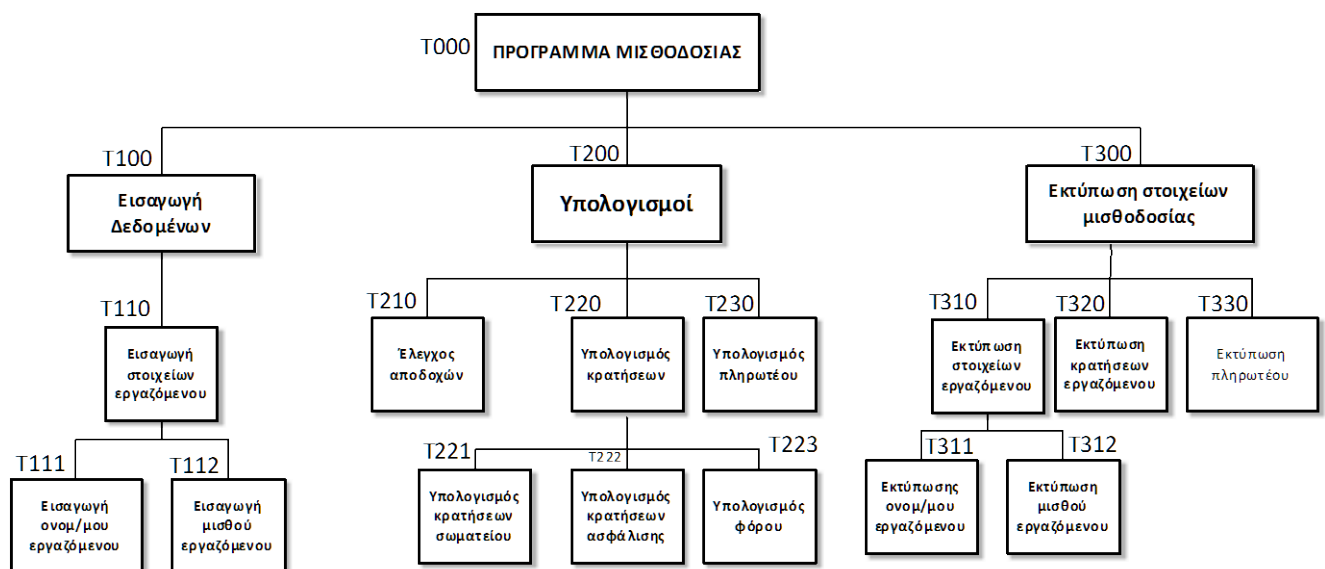
2. Στον Πανελλήνιο Διαγωνισμό Πληροφορικής που γίνεται κάθε χρόνο μπορούν να συμμετάσχουν μαθητές από κάθε τάξη του Γυμνασίου, των ΕΠΑ.Λ και του ΓΕΛ. Αν μας δίνεται η τάξη και ο τύπος του σχολείου που φοιτά ο μαθητής, να υπολογιστεί το πλήθος των μαθητών που συμμετέχουν από κάθε τάξη και ο συνολικός αριθμός των επιτυχόντων. Να σχεδιάσετε το διάγραμμα HIPO του προβλήματος.

Λύση



3. Σε μια εταιρεία γίνονται κρατήσεις, από το μισθό του υπαλλήλου υπέρ του συνδικαλιστικού του σωματείου, ανάλογα με το ύψος του μισθού του. Για μισθό έως 440€ η κράτηση είναι 1%, έως 735€ 1.5%, έως 880€ 2%, έως 1135€ 2.5% και για πάνω από 1135 3.5%. Δίνονται για κάθε υπάλληλο το ονοματεπώνυμό του και ο μισθός του. Επίσης γίνεται ασφαλιστική κράτηση ίση με το 15% επί του μισθού του καθενός και κράτηση φόρου που είναι ίση με 15% επί του φορολογητέου ποσού. (φορολογητέο ποσό = μισθός – (συνδ. Κράτηση + ασφ. Κράτηση). Αν μας δίνεται για κάθε υπάλληλο το ονοματεπώνυμό του και ο μισθός του να υπολογιστούν το ποσό των κρατήσεων που αντιστοιχεί στον κάθε υπάλληλο και το τελικό πληρωτέο ποσό που του αντιστοιχεί. Να σχεδιάσετε το διάγραμμα HIPO του προβλήματος. [Οι μισθοί έχουν προσαρμοστεί από δραχμές σε ευρώ]

Λύση

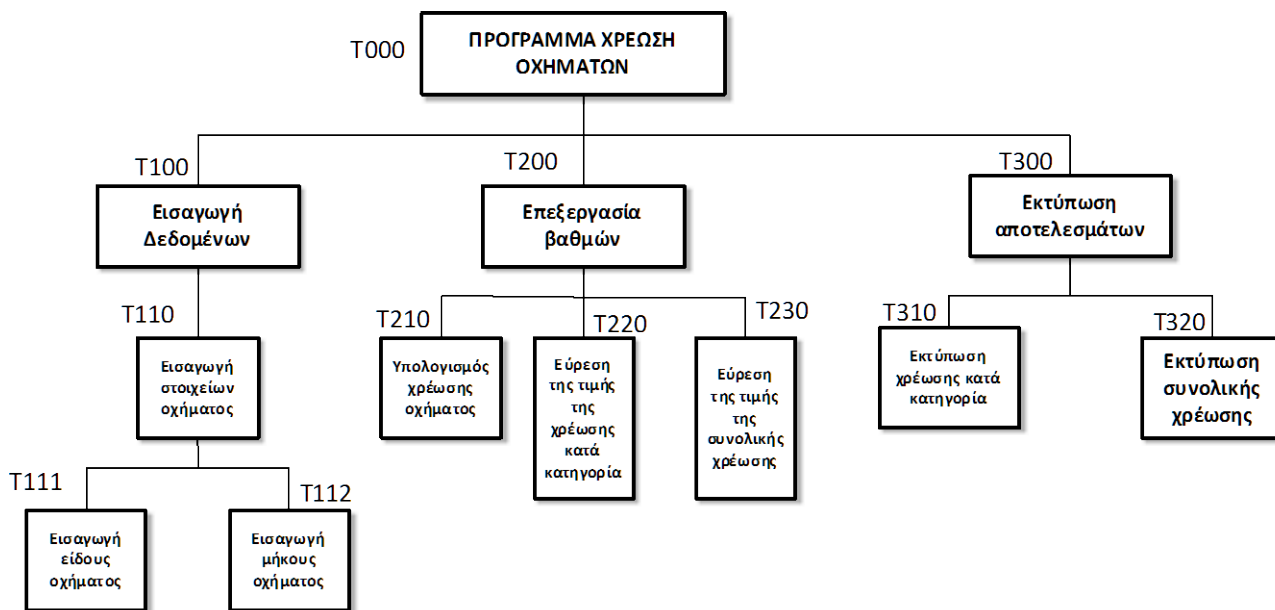


4. Δίνεται ο τύπος ενός οχήματος (μηχανή, ΙΧ επιβατηγό, ΙΧ φορτηγό, ΔΧ επιβατηγό, ΔΧ φορτηγό) και το μήκος του. Η χρέωση (τιμές σε €) σε ένα οχηματαγωγό πλοίο είναι ανάλογη του τύπου του οχήματος και του μήκους του ως εξής:

ΜΗΚΟΣ σε μέτρα	Μηχανή	ΙΧ επιβατηγό	ΔΧ επιβατηγό	ΙΧ φορτηγό	ΔΧ φορτηγό
Έως 2	1.5	3	6	9	12
3	1.5	6	9	12	15
5		9	12	15	18
10				18	21
περισσότερα					24

Δίνονται οι αριθμοί των οχημάτων της κάθε κατηγορίας, που επιβιβάστηκαν στο οχηματαγωγό πλοίο. Να υπολογιστεί το συνολικό ποσό είσπραξης. Σε όποια κατηγορία δεν αντιστοιχεί ποσό, σημαίνει ότι δεν μπορεί να υπάρχει και αντίστοιχος τύπος οχήματος. Να σχεδιάσετε το διάγραμμα HIPO του προβλήματος.

Λύση



Κεφάλαιο 6

ΕΛΕΓΧΟΣ ΑΛΓΟΡΙΘΜΩΝ

Ερωτήσεις

1. Τι καλείται δοκιμή αλγορίθμου;

Δοκιμή αλγορίθμου λέμε τον έλεγχο του αλγορίθμου για την ορθότητα των αποτελεσμάτων του

2. Ποια είναι τα απαραίτητα βήματα για τον έλεγχο ενός αλγορίθμου;

Για να ελέγξουμε έναν αλγόριθμο εκτελούμε τα παρακάτω βήματα:

- Καταγράφουμε **εικονικά δεδομένα** σύμφωνα με τις απαιτήσεις του προβλήματος.
- Εξάγουμε τα αποτελέσματα σύμφωνα με τα βήματα του αλγορίθμου με **χειρογραφικό** τρόπο.
- **Κωδικοποιούμε** τον αλγόριθμο σε μία γλώσσα προγραμματισμού.
- Εκτελούμε το πρόγραμμα **σε υπολογιστή** και καταγράφουμε τα αποτελέσματα.
- **Συγκρίνουμε** τα χειρογραφικά και τα αποτελέσματα από την εκτέλεση του προγράμματος.

3. Τι καλείται εκσφαλμάτωση;

Εκσφαλμάτωση (debugging) είναι η διαδικασία εύρεσης των λογικών λαθών ενός αλγορίθμου.

4. Ποια η μεθοδολογία ανεύρεσης λογικών λαθών;

- Διαπίστωση του **είδους** του λάθους.
- Ανεύρεση του **ανεξάρτητου τμήματος** του αλγορίθμου που εκτελεί τη λανθασμένη λειτουργία.
- **Ανεύρεση του λάθους** μέσα σε αυτό το ανεξάρτητο τμήμα.

5. Τι καλείται έλεγχος δεδομένων;

Έλεγχος δεδομένων καλείται η διαδικασία επιλογής δεδομένων με τέτοιο τρόπο έτσι ώστε αυτά να αντιστοιχούν και σε ένα διαφορετικό παράδειγμα καλύπτοντας κατ' αυτό τον τρόπο κάθε δυνατή περίπτωση πραγματικών δεδομένων.

6. Τι καλείται αξιολόγηση αλγορίθμου;

Αξιολόγηση ενός αλγορίθμου ονομάζεται η διαδικασία της αποτίμησής¹⁰ του λαμβάνοντας υπόψη συγκεκριμένες παραμέτρους.

7. Ποια τα βασικά κριτήρια αξιολόγησης;

Τα βασικά κριτήρια αξιολόγησης ενός αλγορίθμου είναι:

- Η δυνατότητα που παρέχει για ελαχιστοποίηση του κόστους ελέγχου της ορθότητάς του.
- Η δυνατότητα που παρέχει για εύκολη συντήρηση και ευκολία για αλλαγές και επεκτάσεις.
- Η δυνατότητα που παρέχει για εύκολη ανάπτυξη σε γλώσσα προγραμματισμού , ανάλογα με την υφή του.

8. Για ποιο λόγο σήμερα μπορούμε να έχουμε ευχερέστερο έλεγχο της ορθότητας ενός αλγορίθμου;

Σήμερα, ο έλεγχος της ορθότητας ενός αλγορίθμου είναι ευχερής διότι α) είναι **προσιτή η χρήση του υπολογιστή** (από άποψη τόσο του υλικού όσο και του λογισμικού) σε αντίθεση με τα παλαιότερα χρόνια και β) οι γλώσσες προγραμματισμού έχουν αναπτυχθεί πάρα πολύ, ώστε να είναι **εύχρηστες** τόσο, όσο ο ψευδοκώδικας.

9. Να συμπληρώσετε τα κενά με τις λέξεις που λείπουν:

α. Μετά την αποτύπωση του αλγορίθμου ενός προβλήματος ακολουθεί η διαδικασία της **κωδικοποίησης** και της **εκτέλεσης** του προγράμματος.

β. Για τον έλεγχο του αλγορίθμου που σχεδιάσαμε, πρέπει να δημιουργήσουμε **δεδομένα** και με χειρογραφικό τρόπο να εξαγάγουμε τα αποτελέσματα σύμφωνα με τον αλγόριθμο.

γ. Σήμερα μπορούμε με **ευκολία** να δοκιμάζουμε την ορθότητα ενός αλγορίθμου, αφού τον κωδικοποιήσουμε σε γλώσσα προγραμματισμού.

δ. Πρωταρχικός παράγων επιτυχίας της δοκιμής του αλγορίθμου μας είναι η **επιλογή** των **δεδομένων** του προβλήματος.

ε. Η αξιολόγηση ενός αλγορίθμου εξαρτάται άμεσα και από τη **γλώσσα προγραμματισμού** στην οποία θα υλοποιηθεί.

στ. Η αξιολόγηση βέβαια ενός αλγορίθμου **εξαρτάται** και από την επίδοσή του σε χρόνο εκτέλεσης του προγράμματος.

ζ. Η αξιολόγηση ενός αλγορίθμου εξαρτάται και από την απασχόληση της **μνήμης** ή άλλων αντίστοιχων **πόρων** του συστήματος.

Ασκήσεις

- 1. Δίνονται το ονοματεπώνυμο κάθε μαθητή μιας τάξης και η βαθμολογία του σε κάθε μάθημά του και θέλουμε να υπολογίσουμε το μέσο όρο της βαθμολογίας όλων των μαθητών της τάξης. Να καταγράψετε εικονικά δεδομένα σύμφωνα με τις απαιτήσεις του προβλήματος και με χειρογραφικό τρόπο να εξάγετε τα αποτελέσματα.**

Λύση

¹⁰ Το βιβλίο στη σελ. 45 για να ορίσει την έννοια της «Αξιολόγησης αλγορίθμου» χρησιμοποιεί την ίδια τη λέξη «αξιολόγηση» μέσα στον ορισμό.

Σημειώνουμε τα ονόματα και αναλυτική βαθμολογία πέντε μαθητών της τάξης και δοκιμάζουμε χειρογραφικά να εξαγάγουμε το αποτέλεσμα. [Κατασκευάζουμε πρόγραμμα σε Pascal και δίνουμε τα ίδια δεδομένα. Τα αποτελέσματα πρέπει να συμφωνούν].

2. Στον Πανελλήνιο Διαγωνισμό Πληροφορικής που γίνεται κάθε χρόνο μπορούν να συμμετάσχουν μαθητές από κάθε τάξη του Γυμνασίου, των ΕΠΑ.Λ και του ΓΕΛ. Αν μας δίνεται η τάξη και ο τύπος του σχολείου που φοιτά ο μαθητής, να υπολογιστεί το πλήθος των μαθητών που συμμετέχουν από κάθε τάξη και ο συνολικός αριθμός των επιτυχόντων. Να καταγράψετε εικονικά δεδομένα σύμφωνα με τις απαιτήσεις του προβλήματος και με χειρογραφικό τρόπο να εξάγετε τα αποτελέσματα.

Λύση

Σημειώνουμε τα ονόματα και το βαθμό επτά μαθητών Γυμνασίου (2 της Α, 3 της Β, 2 της Γ από τους οποίους οι επιτυχόντες είναι ένας από κάθε τάξη), οκτώ μαθητών ΓΕΛ (3 της Α, 1 της Β, 4 της Γ, από τους οποίους οι επιτυχόντες να είναι δύο από κάθε τάξη) και εννέα μαθητών ΕΠΑ.Λ (3 της Α, 3 της Β, 3 της Γ, από τους οποίους οι επιτυχόντες όλοι της Α, δύο της Β και κανένα από τη Γ). [Κατασκευάζουμε πρόγραμμα σε Pascal και δίνουμε τα ίδια δεδομένα. Τα αποτελέσματα πρέπει να συμφωνούν].

3. Σε μια εταιρεία γίνονται κρατήσεις, από το μισθό του υπαλλήλου υπέρ του συνδικαλιστικού του σωματείου, ανάλογα με το ύψος του μισθού του. Για μισθό έως 440€ η κράτηση είναι 1%, έως 735€ 1.5%, έως 880€ 2%, έως 1135€ 2.5% και για πάνω από 1135 3.5%. Δίνονται για κάθε υπάλληλο το ονοματεπώνυμό του και ο μισθός του. Επίσης γίνεται ασφαλιστική κράτηση ίση με το 15% επί του μισθού του καθενός και κράτηση φόρου που είναι ίση με 15% επί του φορολογητέου ποσού. (φορολογητέο ποσό =μισθός – (συνδ. Κράτηση + ασφ. Κράτηση). Αν μας δίνεται για κάθε υπάλληλο το ονοματεπώνυμό του και ο μισθός του να υπολογιστούν το ποσό των κρατήσεων που αντιστοιχεί στον κάθε υπάλληλο και το τελικό πληρωτέο ποσό που του αντιστοιχεί. Να καταγράψετε εικονικά δεδομένα σύμφωνα με τις απαιτήσεις του προβλήματος και με χειρογραφικό τρόπο να εξάγετε τα αποτελέσματα. [Οι μισθοί έχουν προσαρμοστεί οι μισθοί από δραχμές σε ευρώ]

Λύση

Σημειώνουμε τα στοιχεία και το μισθό για υπάλληλο α) μέχρι 440€ β) μέχρι 880€ γ)1135€ δ)2000€ και υπολογίζουμε χειρογραφικά τις κρατήσεις καθώς και το καθαρό ποσό. [Κατασκευάζουμε πρόγραμμα σε Pascal και δίνουμε τα ίδια δεδομένα. Τα αποτελέσματα πρέπει να συμφωνούν].

4. Δίνεται ο τύπος ενός οχήματος (μηχανή, ΙΧ επιβατηγό, ΙΧ φορτηγό, ΔΧ επιβατηγό, ΔΧ φορτηγό) και το μήκος του. Η χρέωση (τιμές σε €) σε ένα οχηματαγωγό πλοίο είναι ανάλογη του τύπου του οχήματος και του μήκους του ως εξής:

ΜΗΚΟΣ σε μέτρα	Μηχανή	ΙΧ επιβατηγό	ΔΧ επιβατηγό	ΙΧ φορτηγό	ΔΧ φορτηγό
Έως 2	1.5	3	6	9	12
3	1.5	6	9	12	15
5		9	12	15	18
10				18	21
περισσότερα					24

Δίνονται οι αριθμοί των οχημάτων της κάθε κατηγορίας, που επιβιβάσθηκαν στο οχηματαγωγό πλοίο. Να υπολογιστεί το συνολικό ποσό είσπραξης. Σε όποια κατηγορία δεν αντιστοιχεί ποσό, σημαίνει ότι δεν μπορεί να υπάρχει και αντίστοιχος τύπος οχήματος. Να καταγράψετε εικονικά δεδομένα σύμφωνα με τις απαιτήσεις του προβλήματος και με χειρογραφικό τρόπο να εξάγετε τα αποτελέσματα.

Λύση

Σημειώνουμε τα στοιχεία α) για μηχανή έως 2μ β) για μηχανή έως 3μ γ) επιβατηγό γ1) ΙΧ έως 3μ γ2) ΔΧ έως 2μ γ3) ΔΧ έως 5 μ δ) για φορτηγό δ1) ΙΧ έως 3μ δ2) ΙΧ έως 5μ δ3) ΔΧ έως 3μ δ4) ΔΧ έως 5μ δ5) έως 10μ ε δ6) 15 μ και υπολογίζουμε τα συνολικά ποσά κατά κατηγορία καθώς και συνολικό ποσό είσπραξης. [Κατασκευάζουμε πρόγραμμα σε Pascal και δίνουμε τα ίδια δεδομένα. Τα αποτελέσματα πρέπει να συμφωνούν].

Κεφάλαιο 7

ΕΙΔΗ, ΤΕΧΝΙΚΕΣ ΚΑΙ ΠΕΡΙΒΑΛΛΟΝΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Ερωτήσεις

1. Να αναφέρετε διαφορές μεταξύ γλωσσών μηχανής και γλωσσών χαμηλού επιπέδου.

- Οι γλώσσες μηχανής κωδικοποιούν τις εντολές τους με ομάδες 0 και 1 σταθερού ή μεταβλητού πλάτους ενώ οι γλώσσες χαμηλού επιπέδου με συντομογραφίες λέξεων της αγγλικής γλώσσας.
- Ένα πρόγραμμα σε γλώσσα μηχανής εκτελείται απευθείας από την ΚΜΕ του υπολογιστή ενώ ένα πρόγραμμα γλώσσας χαμηλού επιπέδου απαιτεί μεταφραστικό πρόγραμμα για να μετατραπεί σε γλώσσα μηχανής.

2. Να αναφέρετε διαφορές μεταξύ γλωσσών υψηλού επιπέδου και γλωσσών χαμηλού επιπέδου.

- Ένα πρόγραμμα σε γλώσσα υψηλού επιπέδου δεν είναι εξαρτημένο από τον υπολογιστή ενώ ένα πρόγραμμα σε γλώσσα χαμηλού επιπέδου είναι.
- Μια γλώσσα υψηλού επιπέδου βασίζεται σε λέξεις κλειδιά της αγγλικής γλώσσας ενώ μια γλώσσα χαμηλού επιπέδου όχι.
- Μια εντολή σε γλώσσα υψηλού επιπέδου μεταφράζεται σε πολλές εντολές του κώδικα μηχανής κατά τη διάρκεια της μετάφρασης ενώ μια εντολή σε γλώσσα χαμηλού επιπέδου μεταφράζεται σε μία μόνο εντολή γλώσσας μηχανής.

3. Πώς μπορούμε να χαρακτηρίσουμε τις γλώσσες υψηλού επιπέδου ανάλογα με τις λειτουργικές τους ιδιότητες;

Σε γλώσσες για

- α: Επιστημονικές εφαρμογές (Fortran, C, APL).
- β: Εμπορικές εφαρμογές (Cobol, RPG).
- γ: Εκπαιδευτικές εφαρμογές (Basic, Logo).
- δ: Ειδικές¹¹ εφαρμογές (Lisp, Prolog).
- ε: Επιστημονικές – εμπορικές εφαρμογές (PL-1, PASCAL).

4. Τι γνωρίζετε για τις γλώσσες 4ης γενιάς;

- Είναι γλώσσες υψηλής παραγωγικότητας και μπορούν να τις χρησιμοποιήσουν και άτομα που δεν έχουν γνώσεις προγραμματισμού.
- Διαθέτουν δυναμικές εντολές για τη δημιουργία αρχείων, τη δημιουργία και ενημέρωση οθονών για την εκμετάλλευση των αρχείων και τη δημιουργία εκτυπώσεων. Κάθε μία εντολή γλώσσας 4^{ης} γενιάς αντιστοιχεί σε ολόκληρο πρόγραμμα μιας κλασσικής γλώσσας προγραμματισμού.
- Είναι γλώσσες διαλογικές.
- Η υλοποίηση πληροφοριακών συστημάτων είναι ταχύτερη με χρήση γλωσσών 4^{ης} γενιάς.

5. Τι γνωρίζετε για τον διαδικασιακό προγραμματισμό;

Στο Διαδικασιακό προγραμματισμό, το πρόγραμμα είναι γραμμένο σε κάποια γλώσσα εξαρτημένη από τη διαδικασία, η οποία είναι χαρακτηριστικό των γλωσσών 3^{ης} γενιάς.

¹¹ Τεχνητής νοημοσύνης

Σχολικός Σύμβουλος Πληροφορικής και καθηγητές Πληροφορικής ΕΠΑ.Λ Μεσσηνίας & Λακωνίας

(COBOL, FORTRAN, PASCAL). Μια γλώσσα που είναι εξαρτημένη από τη διαδικασία **δίνει έμφαση στον προγραμματισμό των υπολογιστικών και λογικών διαδικασιών που απαιτούνται για την επίλυση του προβλήματος**. Προβλήματα που επιλύονται με διαδικασιακό προγραμματισμό είναι για λογιστικές, εμπορικές, διοικητικές εφαρμογές, εφαρμογές μεταφορών κτλ.

Στο διαδικασιακό προγραμματισμό, το **πρόγραμμα** είναι μια διαδικασία η οποία εκτελεί **ένα προς ένα τα βήματα του αλγόριθμου επίλυσης του προβλήματος**. Οι εντολές εκτελούνται **διαδοχικά** εκτός αν υπάρχουν **συνθήκες** οπότε κάποιες εντολές παραλείπονται και δημιουργούνται **διακλαδώσεις** ή κάποιες εντολές **επαναλαμβάνονται**.

Ο διαδικασιακός προγραμματισμός, με τις γλώσσες 3^{ης} γενιάς έχει απομακρύνει τον προγραμματισμό από το επίπεδο της μηχανής και τον έχει φέρει προς το **επίπεδο του ανθρώπου**.

6. Να δώσετε μερικά παραδείγματα γλωσσών που είναι εξαρτημένες από τη διαδικασία.

COBOL, FORTRAN, PASCAL και επίσης οι Ada, Modula 2, Concurrent C.

7. Ποιες είναι οι βασικές αλγοριθμικές δομές και ποιες οι τεχνικές σχεδίασης στο δομημένο προγραμματισμό;

Στον δομημένο προγραμματισμό, οι βασικές αλγοριθμικές δομές είναι οι δομές της **διαδοχής**, της απλής **επιλογής** και της **επανάληψης**.

Στον δομημένο προγραμματισμό χρησιμοποιούνται οι τεχνικές του **ιεραρχικού** και του **τμηματικού** προγραμματισμού.

8. Πού χρησιμοποιούνται τα διαγράμματα HIPO;

Τα διαγράμματα HIPO χρησιμοποιούνται στον δομημένο προγραμματισμό για την αναπαράσταση των τεχνικών του ιεραρχικού σχεδιασμού και του τμηματικού προγραμματισμού. Στα διαγράμματα αυτά εμφανίζονται τα διάφορα επίπεδα ιεραρχίας και τα διάφορα τμήματα προγράμματος με αριθμό σύμφωνα με τη σειρά εκτέλεσής τους.

9. Πού οφείλει την καθιέρωσή του ο παράλληλος προγραμματισμός;

Ο Παράλληλος Προγραμματισμός οφείλει την καθιέρωσή του στην **ανάπτυξη του υλικού**. Η εμφάνιση της αρχιτεκτονικής των **πολλών επεξεργαστών** οι οποίοι χρησιμοποιούν **κοινή μνήμη** είχε ως συνέπεια την ανάπτυξη των παράλληλων αλγορίθμων οι οποίοι και επέβαλαν τον Παράλληλο Προγραμματισμό. Αντιπροσωπευτική γλώσσα για παράλληλη επεξεργασία είναι η γλώσσα Occam.

10. Τι είναι ο αντικειμενοστρεφής προγραμματισμός;

Ο αντικειμενοστρεφής προγραμματισμός είναι μία τεχνική στην οποία υπάρχει ενσωμάτωση των δεδομένων και του τρόπου χειρισμού τους μέσα από την έννοια του **αντικειμένου**.

Τα **χαρακτηριστικά** ενός αντικειμένου είναι τα **δεδομένα από τα οποία αποτελείται** και οι **μέθοδοι** του αντικειμένου είναι οι ενέργειες οι οποίες σχετίζονται με την επεξεργασία των δεδομένων και **καθορίζουν τη συμπεριφορά του αντικειμένου** στο πρόγραμμα. Οι μέθοδοι μπορεί να είναι **διαδικασίες** ή **συναρτήσεις** του χρήστη στο κυρίως πρόγραμμα.

Σε ένα αντικειμενοστρεφές πρόγραμμα, μπορούν να δημιουργούνται **κλάσεις ιεραρχικά δομημένες**. Με την ιεραρχική δόμηση των κλάσεων, οι **υποκλάσεις κληρονομούν τα χαρακτηριστικά και τις ιδιότητες των κλάσεων** από τις οποίες προήλθαν, έχοντας τη δυνατότητα να προστεθούν σε αυτές (δηλ. στις υποκλάσεις) **νέες ιδιότητες** και χαρακτηριστικά.

11. Ποια είναι τα χαρακτηριστικά και ποιες οι μέθοδοι στον αντικειμενοστρεφή προγραμματισμό;

Τα **χαρακτηριστικά** ενός αντικειμένου είναι τα **δεδομένα από τα οποία αποτελείται** και οι **μέθοδοι** του αντικειμένου είναι οι ενέργειες οι οποίες σχετίζονται με την επεξεργασία των δεδομένων και **καθορίζουν τη συμπεριφορά του αντικειμένου** στο πρόγραμμα. Οι μέθοδοι μπορεί να είναι **διαδικασίες** ή **συναρτήσεις** του χρήστη στο κυρίως πρόγραμμα.

12. Τι είναι ο Συναρτησιακός Προγραμματισμός;

Ο Συναρτησιακός Προγραμματισμός **στηρίζεται στην έννοια της συνάρτησης**, όπως την έχουμε γνωρίσει από τα μαθηματικά. Η συνάρτηση δίνει τη δυνατότητα απεικόνισης ενός πεδίου ορισμού σε ένα πεδίο τιμών. Στο Συναρτησιακό προγραμματισμό οι **εντολές** και οι **δομές ελέγχου είναι συναρτήσεις** οι οποίες έχουν ως ορίσματα δεδομένα ή άλλες συναρτήσεις.

13. Αναφέρετε παραδείγματα γλωσσών Συναρτησιακού Προγραμματισμού.

Παραδείγματα γλωσσών Συναρτησιακού Προγραμματισμού είναι οι γλώσσες FORTH και LOGO.

14. Ποια η διαφορά του Λογικού από το Διαδικασιακό προγραμματισμό;

Στον Λογικό προγραμματισμό, στο πρόγραμμα **δεν περιγράφεται ο αλγόριθμος** επίλυσης του προβλήματος αλλά **η γνώση** που σχετίζεται με το πρόβλημα. *Ο Λογικός Προγραμματισμός σε αντίθεση με το Διαδικασιακό, στηρίζεται στη γνώση.* Η γνώση περιγράφεται υπό μορφή **γεγονότων** και **κανόνων** ενώ **ορίζεται μία πρόταση – στόχος προς απόδειξη**. Για την απόδειξη της πρότασης αυτής το πρόγραμμα εφαρμόζει τους κανόνες του πάνω στα γεγονότα.

15. Να εξηγήσετε τους όρους πηγαίο πρόγραμμα, αντικείμενο πρόγραμμα και μεταγλωττιστής.

Πηγαίο πρόγραμμα (source) ονομάζεται το πρόγραμμα το οποίο είναι γραμμένο σε μια γλώσσα προγραμματισμού¹². Ο **μεταγλωττιστής** είναι ένα ειδικό λογισμικό (πρόγραμμα) το οποίο μετατρέπει το πηγαίο πρόγραμμα σε **αντικείμενο** (object) πρόγραμμα. Το αντικείμενο πρόγραμμα περνά στη συνέχεια από κάποιες διεργασίες δίνοντας το πρόγραμμα σε κώδικα (**γλώσσα**) **μηχανής** το οποίο μπορεί άμεσα να εκτελέσει ο υπολογιστής¹³.

16. Να αναφέρετε ομοιότητες και διαφορές μεταξύ compiler και interpreter.

Ομοιότητες:

- Τόσο ο compiler όσο και ο interpreter είναι μεταφραστικά προγράμματα τα οποία μετατρέπουν ένα πρόγραμμα από γλώσσα προγραμματισμού¹⁴ σε γλώσσα μηχανής ώστε να μπορεί να εκτελεστεί από τον υπολογιστή.

Διαφορές:

- Ο compiler κάνει ανάλυση(συντακτική και γραμματική) ολόκληρου του πηγαίου προγράμματος και το μετατρέπει σε αντικείμενο πρόγραμμα μόνο εφόσον δεν υπάρχουν λάθη. Αντίθετα, ο interpreter δεν ελέγχει τη σύνταξη ολόκληρου του πηγαίου προγράμματος πριν από την εκτέλεση.
- Με τη χρήση compiler και εφόσον γίνει επιτυχής μετάφραση, το αντικείμενο πρόγραμμα μπορεί να χρησιμοποιηθεί όσες φορές θέλουμε χωρίς να προηγηθεί μετά-

¹² υψηλού επιπέδου.

¹³ Η ΚΜΕ του υπολογιστή.

¹⁴ υψηλού επιπέδου

φραση. Αντίθετα, με τη χρήση interpreter, πάντα θα πρέπει να επαναλαμβάνεται η διαδικασία μετάφρασης – ελέγχου κάθε φορά που εκτελείται το πηγαίο πρόγραμμα¹⁵. Ο συντακτικός έλεγχος και η μετάφραση γίνεται σε κάθε μία από τις εντολές του πηγαίου προγράμματος. Εφόσον δεν υπάρχουν λάθη, παράγεται η αντίστοιχη εντολή σε κώδικα μηχανής, εκτελείται και συνεχίζεται η μετάφραση της επόμενης εντολής.

- Ο έλεγχος και η διόρθωση των λαθών ενός προγράμματος με τη χρήση interpreter είναι πιο εύκολη από ότι η διόρθωση με τη χρήση compiler, αφού στην περίπτωση του interpreter το πρόγραμμα σταματά στο σημείο του λάθους, ο προγραμματιστής το πληροφορείται, το διορθώνει και συνεχίζει την εκτέλεση του προγράμματος.
- Η εκτέλεση ενός προγράμματος με interpreter είναι πολύ πιο αργή σε σύγκριση με την εκτέλεση του ίδιου προγράμματος, μεταφρασμένου με compiler.

17. Να εξηγήσετε τη διαδικασία γραφής –εκτέλεσης προγράμματος.

Το **πηγαίο πρόγραμμα** γράφεται σε κάποια γλώσσα προγραμματισμού με τη βοήθεια ενός απλού προγράμματος επεξεργασίας κειμένου (που λέγεται **εκδότης – διορθωτής** (editor)).

Ο **μεταγλωττιστής**¹⁶ ελέγχει το πηγαίο πρόγραμμα για ορθογραφικά ή συντακτικά λάθη. Σε περίπτωση που διαπιστώνονται λάθη, αυτά διορθώνονται και το πηγαίο πρόγραμμα επανελέγχεται από το μεταγλωττιστή. Η διαδικασία αυτή επαναλαμβάνεται μέχρι τα λάθη να μηδενιστούν. Στην περίπτωση αυτή, η μεταγλώττιση του προγράμματος είναι επιτυχής και παράγεται το **αντικείμενο πρόγραμμα**.

Το αντικείμενο πρόγραμμα δεν είναι κατ' ανάγκη εκτελέσιμο από τον υπολογιστή αλλά πρέπει να **συνδεθεί** με **υποπρογράμματα της βιβλιοθήκης της γλώσσας** ή του προγραμματιστή με τη βοήθεια ενός άλλου προγράμματος του **συνδέτη** (linker) ώστε τελικά να παραχθεί εκτελέσιμος κώδικας.

[διαδικασία: **γραφή** → **μεταγλώττιση** → **σύνδεση** → **εκτέλεση**].

Σε άλλες περιπτώσεις αντί του συνδέτη χρησιμοποιείται ένα άλλο πρόγραμμα, ο **φορτωτής** (loader) ο οποίος συνδέει το αντικείμενο πρόγραμμα με υποπρογράμματα της βιβλιοθήκης της γλώσσας ή του προγραμματιστή και φορτώνει στη μνήμη για εκτέλεση του κώδικα μηχανής που προκύπτει.

[διαδικασία: **γραφή** → **μεταγλώττιση** → **φόρτωση**].

Στην περίπτωση που το μεταφραστικό πρόγραμμα είναι διερμηνέας, η μεταγλώττιση, η σύνδεση και η εκτέλεση γίνονται εντολή προς εντολή κατά τη σειρά εκτέλεσής τους.

18. Ποια είναι τα σύγχρονα προγραμματιστικά περιβάλλοντα;

Τα σύγχρονα προγραμματιστικά περιβάλλοντα εμπεριέχουν σε ένα **ολοκληρωμένο περιβάλλον**, πολλά από τα προγράμματα ανάπτυξης, όπως διορθωτή (editor), μεταγλωττιστή (compiler), εκσφαλματωτή (debugger) και άλλα στα οποία μπορούμε να έχουμε πρόσβαση μέσα από κατάλληλα μενού. Επιπλέον, ένα ολοκληρωμένο περιβάλλον ανάπτυξης μπορεί να διαθέτει και εργαλεία σχεδίασης με τη βοήθεια γραφικών οπότε χαρακτηρίζεται ως **γραφικό περιβάλλον**.

19. Να συμπληρώσετε τα κενά από τη λέξη που λείπει:

- α. ανάπτυξης προγράμματος
- β. κωδικοποίηση
- γ. ακολουθία, μεταβλητού

¹⁵ Το πηγαίο πρόγραμμα πρέπει να υπάρχει στον υπολογιστή και να εκτελείται σε συνδυασμό με τον interpreter.

¹⁶ compiler

- δ. συντομογραφίες λέξεων, διευθύνσεις, συμβολικές, χαμηλού
- ε. επιπέδου, αγγλικής, γραμματική, συντακτικό, προτάσεις, εντολές
- στ. 4^{ης}, εκτέλεσης, διαλογικές, εντολών

20. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

- α. διαδικασία, 3ης γενιάς
- β. αλγοριθμικές, διαδοχή, επιλογή, επανάληψη, ιεραρχικού, τμηματικού, ΗΙΡΟ, Ιε-
ράρχηση, επεξεργασίας.
- γ. πολλών, κοινή, παράλληλων, προγραμματισμού.
- δ. χειρισμού, αντικειμένου, χαρακτηριστικά, επεξεργασία, αντικειμένου, διαδικασίες.
- ε. μεταφράζει, προγραμματισμού, μηχανής, πηγαίο, αντικείμενο.

Κεφάλαιο 8

Η ΓΛΩΣΣΑ PASCAL

Ερωτήσεις

- 1. Να αναφέρετε μερικά από τα ιδιαίτερα χαρακτηριστικά της Pascal.**
 - Δίνει τη δυνατότητα στον προγραμματιστή να δημιουργεί **δικούς του τύπους δεδομένων**.
 - Χρήση **μεταβλητών τύπου δείκτη** και δυνατότητα δυναμικής διαχείρισης της κεντρικής μνήμης.
 - Η **σύνθετη εντολή** (compound statement), δηλαδή η χρήση μιας σειράς εντολών ως μία εντολή.
- 2. Ποιο είναι το αλφάβητο της Pascal;**

Το αλφάβητο της Pascal αποτελείται από βασικά σύμβολα, όπως τα γράμματα του ελληνολατινικού αλφαβήτου, τα αριθμητικά ψηφία (0-9) και τα ειδικά σύμβολα, όπως +, -, *, /, \, ., ; κτλ.
- 3. Ποια είναι τα ονόματα-ταυτότητες και σε τι χρησιμεύουν;**

Τα ονόματα-ταυτότητες (identifiers) είναι ονόματα, τα οποία αναφέρονται σε σταθερές, μεταβλητές, τύπους δεδομένων, διαδικασίες, συναρτήσεις κλπ. Ένα όνομα –ταυτότητα αποτελείται από μια σειρά χαρακτήρων (γράμματα λατινικού αλφαβήτου, αριθμοί ή `_`), πρέπει να αρχίζει πάντοτε με γράμμα και δεν πρέπει να περιέχει κενά.
- 4. Σε τι χρησιμεύει το Συντακτικό (Syntax) της γλώσσας;**

Το Συντακτικό είναι ένα σύνολο τυπικών κανόνων οι οποίοι προσδιορίζουν πώς γράφονται έγκυρες εντολές σε μια γλώσσα προγραμματισμού.
- 5. Σε τι χρησιμεύει η Σημασιολογία (Semantics);**

Η Σημασιολογία είναι ένα σύνολο κανόνων οι οποίοι προσδιορίζουν τη σημασία των εντολών που γράφονται σε μια γλώσσα προγραμματισμού.
- 6. Ποιοι είναι οι απλοί ή στοιχειώδεις τύποι δεδομένων;**

Οι απλοί ή στοιχειώδεις τύποι δεδομένων είναι: ο ακέραιος, ο πραγματικός, ο λογικός και ο χαρακτήρας.
- 7. Πότε χρησιμοποιείται ο πραγματικός τύπος;**

Ο πραγματικός τύπος χρησιμοποιείται εκεί που οι αριθμητικές τιμές δεν είναι ακέραιοι αριθμοί ή οι αναμενόμενες τιμές τους είναι εκτός των ορίων του ακέραιου τύπου.
- 8. Ποιες είναι οι πράξεις που μπορεί να γίνουν με μεταβλητές ή εκφράσεις λογικού τύπου;**

Οι πράξεις που μπορεί να γίνουν με μεταβλητές ή εκφράσεις λογικού τύπου είναι η σύζευξη, η διάζευξη, η αποκλειστική διάζευξη και η άρνηση και γίνονται με τη χρήση των τελεστών **and**, **or**, **xor** και **not** αντίστοιχα.
- 9. Να εξηγήσετε τη σημασία των: 2000, '2000', etos, 'etos' στη γλώσσα Pascal.**

2000 → ακέραιο δεδομένο
'2000' → δεδομένο αλφαριθμητικού τύπου (string)
etos → όνομα-ταυτότητα (identifier)
'etos' → δεδομένο αλφαριθμητικού τύπου (string)
- 10. Ποια είναι η δομή ενός προγράμματος Pascal;**

Ένα πρόγραμμα Pascal αποτελείται από την **Επικεφαλίδα** (αρχίζει με τη λέξη **program** ακολουθούμενη από το όνομα του προγράμματος και μια λίστα με τα ονόματα των αρ-

χείων τα οποία μπορεί να είναι προαιρετικά μονάδες εισόδου / εξόδου ή τα αρχεία που θα χρησιμοποιήσει κατά την εκτέλεσή του, το τμήμα **Δηλώσεων** (declarations) που ακολουθούν τη σειρά: Δηλώσεις σταθερών (constants), τύπων (εφόσον ορίζονται νέοι τύποι), μεταβλητών και ακολουθεί το **Κύριο Πρόγραμμα** το οποίο ξεκινά με τη λέξη **begin** και τελειώνει με την **end**.

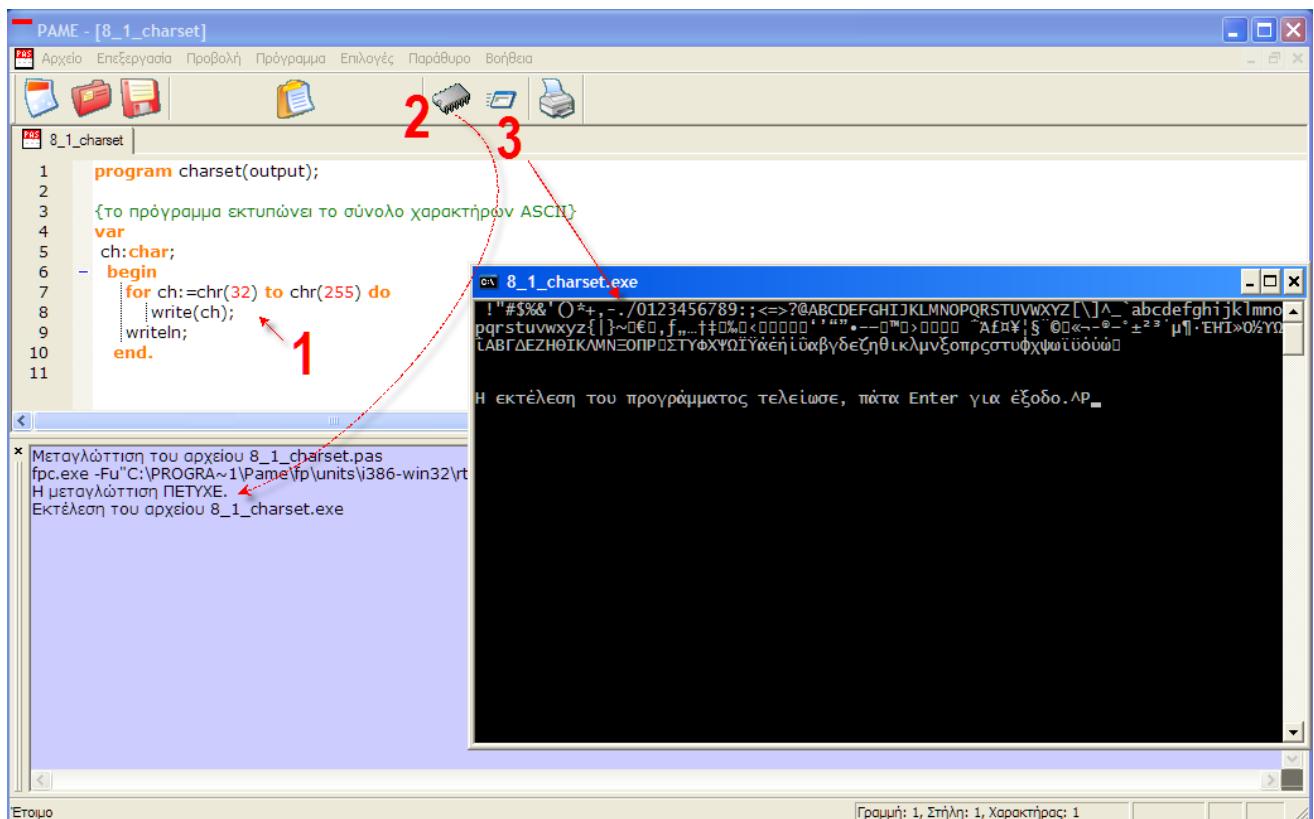
Ασκήσεις

1. Να γράψετε με τον editor, να μεταφράσετε και να εκτελέσετε το παρακάτω πρόγραμμα χρησιμοποιώντας τις κατάλληλες εντολές.

```
program charset(output);  
{το πρόγραμμα εκτυπώνει το σύνολο των χαρακτήρων ASCII }  
var  
    ch:char;  
begin  
    for ch:=chr(32) to chr(255) do  
        write(ch); writeln  
end.
```

Λύση

Χρησιμοποιούμε το ελεύθερο λογισμικό PAME (Pascal Made Easy) το οποίο έχει δημιουργήσει ο καθηγητής ΠΕ19 κ. Γ. Σαμαρτζίδης. <http://users.lak.sch.gr/samartzidis/pame/>
Στην επόμενη εικόνα φαίνεται ο κώδικας (1), η εργασία της μεταγλώττισης της (2) και το αποτέλεσμα της εκτέλεσης (3).



2. Να γράψετε με τον editor, να μεταφράσετε και να εκτελέσετε το παρακάτω πρόγραμμα χρησιμοποιώντας τις κατάλληλες εντολές.

```
program table (output);
```

```
{το πρόγραμμα τυπώνει τα τετράγωνα και τους κύβους των αριθμών 1 έως και 10 }
```

```
var
```

```
  i:integer;
```

```
begin
```

```
  writeln('Number':10,'square':10,'cube':10);
```

```
  for i:=1 to 10 do
```

```
    writeln(i:10,sqr(i):10,i*sqr(i):10)
```

```
end.
```

Λύση

The screenshot shows a Pascal IDE window titled "PAME - [8_2_table]". The source code is as follows:

```
1 program table(output);
2 {Το πρόγραμμα τυπώνει τα τετράγωνα και τους κύβους των αριθμών}
3 var i:integer;
4 begin
5   writeln('Αριθμός':10,'τετράγωνο':10,'κύβος':10);
6   for i:=1 to 10 do
7     |writeln(i:10,sqr(i):10,i*sqr(i):10);
8
9 end.
```

The execution output window, titled "8_2_table.exe", displays the following table:

Αριθμός	τετράγωνο	Κύβος
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Below the table, the output window displays the message: "Η εκτέλεση του προγράμματος τελείωσε, πάτα Enter για έξοδο."

Κεφάλαιο 9

ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ

Ερωτήσεις

1. Να εξηγήσετε πως λειτουργούν οι διαδικασίες `read(parameter,parameter,...)` και `readln(parameter,parameter,...)`. Ποιες οι ομοιότητες και ποιες οι διαφορές μεταξύ τους;

Ομοιότητες

Τόσο η διαδικασία `read` όσο και η `readln` ακολουθείται από μία ή περισσότερες παραμέτρους(μεταβλητές) που χωρίζονται με κόμμα και περιέχονται σε μια παρένθεση. Όταν εκτελείται η διαδικασία, τα δεδομένα διαβάζονται και αποδίδονται στις μεταβλητές κατά τη σειρά κατά την οποία είναι γραμμένες στην παρένθεση. Τα αριθμητικά δεδομένα πρέπει να χωρίζονται με κενό ενώ τα δεδομένα χαρακτήρων όχι.

Διαφορές

Η διαδικασία `readln` μόλις βρει όλες τις τιμές που αντιστοιχούν στα ορίσματά της τοποθετείται στην επόμενη γραμμή ενώ η `read` μόλις βρει όλα τα ορίσματα που περιμένει παραμένει στην ίδια γραμμή. Η επόμενη πρόταση `read` θα διαβάσει από την ίδια γραμμή.

Η `readln` χωρίς παραμέτρους έχει ως αποτέλεσμα να περνά ο έλεγχος στην επόμενη γραμμή.

2. Να εξηγήσετε πως λειτουργούν οι διαδικασίες `write(parameter,parameter,...)` και `writeln(parameter,parameter,...)`. Ποιες οι ομοιότητες και ποιες οι διαφορές μεταξύ τους;

Ομοιότητες

Τόσο η `write` όσο και η `writeln` δέχονται ως ορίσματα σταθερές, μεταβλητές ή εκφράσεις. Η τιμή κάθε παραμέτρου κατά σειρά, τυπώνεται στη γραμμή από αριστερά προς τα δεξιά.

Διαφορές

Η `write` γράφει τις πληροφορίες που περιέχονται στα ορίσματά της χωρίς να προχωρά σε αλλαγή γραμμής. Αντίθετα η `writeln` αφού γράψει τις προσδιοριζόμενες πληροφορίες έχει ως αποτέλεσμα η επόμενη εντολή εισόδου / εξόδου να εκτελείται στην αρχή της επόμενης γραμμής. Η `writeln` χωρίς παραμέτρους, έχει ως αποτέλεσμα να περνά ο έλεγχος στην επόμενη γραμμή.

3. Να εξηγήσετε πως λειτουργούν οι διαδικασίες `write(x:m)` και `writeln(x:m:n)`. Ποιες οι ομοιότητες και ποιες οι διαφορές μεταξύ τους;

Η εντολή `write(x:m)`, όπου x ακέραιη έκφραση, τυπώνει την τιμή της μεταβλητής x από τα δεξιά ενός πεδίου εύρους m . Αν το εύρος είναι μεγαλύτερο από το μήκος του πεδίου, εμφανίζονται κενά στα αριστερά της τιμής x ενώ αν το εύρος είναι μεγαλύτερο της τιμής αγνοείται και η τιμή εκτυπώνεται με το ελάχιστο δυνατό εύρος. Μετά την εκτέλεση της εντολής, ο έλεγχος παραμένει στην ίδια γραμμή.

Η εντολή `writeln(x:m:n)` λειτουργεί κατά τον ίδιο τρόπο με την εντολή `write(x:m)` με τη διαφορά ότι για ορίσματα x πραγματικού τύπου το n προσδιορίζει το πλήθος των δεκαδικών ψηφίων που θα εκτυπωθούν. Επίσης, με την εντολή `writeln` ο έλεγχος περνά στην επόμενη γραμμή.

4. Ποιο θα είναι το αποτέλεσμα της εντολής `write(3.14159265:10:3)`;

Το αποτέλεσμα θα είναι η εκτύπωση του αριθμού 3.142 με πέντε κενά αριστερά του. Αυτό, διότι έχουμε στρογγυλοποίηση στο 3^ο δεκαδικό ψηφίο (n=3) και αφού ο αριθμός (μαζί με την υποδιαστολή) καταλαμβάνει πέντε θέσεις (m=10), θα υπάρχουν αριστερά του άλλες πέντε κενές θέσεις.

5. Ποια είναι τα αποτελέσματα των παρακάτω εντολών, αν αυτές εκτελούνται διαδοχικά `x:=2; x:=2* x +4; y:=(4*x-1)*(5*x-4)`;

Το αποτέλεσμα είναι το μεν x να έχει τελική τιμή 8 και το y 1116.

6. Να συμπληρώσετε τα κενά στον πίνακα που ακολουθεί.

	Εντολή	Δεδομένα	Περιεχόμενα μετά την εκτέλεση
1	<code>read(x)</code>	122	X= <u>122</u>
2	<code>read(a, b, c)</code>	133 244 355	a= <u>133</u> b= <u>244</u> c= <u>355</u>
3	<code>read(d, e)</code>	166 277 3888	d= <u>166</u> e= <u>277</u>
4	<code>read(f, g)</code>	11.99 21 31 41 51	f= <u>11.99</u> g= <u>21</u>
5	<code>read(x, a, d)</code>	10 20 30	x= <u>10</u> a= <u>20</u> d = <u>30</u>

7. Να συμπληρώσετε τα κενά στον πίνακα που ακολουθεί.

Περιεχόμενα παραμέτρων	Εντολή	Αποτελέσματα
i=2	<code>writeln(i)</code> <code>writeln('i=', i)</code> <code>writeln('Error=',i)</code> <code>writeln('Error Message')</code>	<u>2</u> <u>i=2</u> <u>Error=2</u> <u>Error Message</u>
la ='name=' ty ='Tttt'	<code>writeln(ty, ' ',la)</code> <code>writeln(ty, la)</code> <code>writeln('Language=', la)</code>	<u>Tttt... name=</u> <u>Ttttname=</u> <u>Language=name=</u>

Ασκήσεις

1. Να γράψετε πρόγραμμα που να τυπώνει:

στη γραμμή 1 «αυτό είναι το πρώτο μου πρόγραμμα» ,

στη γραμμή 2 το όνομα σας,

στη γραμμή 3 ημερομηνία.

Λύση

```
program ennea_1;
begin
  writeln('Αυτό είναι το πρώτο μου πρόγραμμα');
  writeln('Γιώργος');
  writeln('16-11-2009')
end.
```

2. Να γράψετε πρόγραμμα στο οποίο να γίνεται εισαγωγή των δεδομένων μήκος, πλάτος, ύψος έτσι, ώστε να υπολογίζονται και να τυπώνονται τα παρακάτω αποτελέσματα:

το εμβαδόν της βάσης είναιτετρ.εκ.
το εμβαδόν της έδρας με διαστάσεις ..,..
είναιτετρ.εκ.
το εμβαδόν της έδρας με διαστάσεις ..,..
είναι τετρ.εκ.
ο όγκος του παραλληλεπιπέδου είναι κυβ. εκ.

Λύση

```
program ennea_2;
var
  mikos,platos,ypsos,vasi,edra1, edra2,ogkos:real;
begin
  writeln('Δώσε το μήκος');
  readln(mikos);
  writeln('Δώσε το πλάτος');
  readln(platos);
  writeln('Δώσε το ύψος');
  readln(ypsos);
  vasi:=mikos*platos;
  edra1:=platos*ypsos;
  edra2:=ypsos*mikos;
  ogkos:=mikos*platos*ypsos;
  writeln('Το εμβαδόν της έδρας με διαστάσεις', mikos :6:2,
',', platos:6:2,' είναι: ',vasi:8:2,' τετρ. εκατοστά');
  writeln('Το εμβαδόν της έδρας με διαστάσεις', platos:6:2,
',', ypsos:6:2,' είναι: ', edra1:8:2,' τετρ. εκατοστά');
  writeln('Το εμβαδόν της έδρας με διαστάσεις', ypsos:6:2,
',', mikos:6:2,' είναι: ', edra2:8:2,' τετρ. εκατοστά');
  writeln('Ο όγκος του παραλληλεπιπέδου είναι', ogkos:2, ' κυβικά
εκατοστά');
end.
```

3. Να γράψετε πρόγραμμα το οποίο να διαβάζει έναν αριθμό και να τυπώνει το διπλάσιο και το τριπλάσιο του. Να γίνει η ίδια διαδικασία για τους 2 επόμενους απ' αυτόν αριθμούς. Η μορφή της εκτύπωσης να είναι σύμφωνα με το παρακάτω υπόδειγμα:

8 16 24

9 18 27

10 20 30

Λύση

```
program ennea_3;
var
  number:integer;
begin
  writeln('Δώστε τον ακέραιο αριθμό');
  readln(number);
  writeln('Αριθμός':10, 'Διπλάσιο':10, 'Τριπλάσιο':10);
  writeln(number:10, 2*number:10, 3*number:10);
  number:=number+1;
end.
```

```

    writeln(number:10, 2*number:10, 3*number:10);
    number:=number+1;
    writeln(number:10, 2*number:10, 3*number:10);
end.

```

4. Να γράψετε πρόγραμμα στο οποίο να γίνεται εισαγωγή των ψήφων τριών ατόμων για το καθένα χωριστά. Να βρεθεί ο μέσος όρος των ψήφων και να τυπωθεί η διαφορά των ψήφων καθενός από το μέσο όρο. Οι αριθμοί κατά την εκτύπωση να συνοδεύονται από τις κατάλληλες επεξηγήσεις σύμφωνα με το υπόδειγμα.

Η μορφή της εκτύπωσης θα είναι:

Ο Γιώργος πήρε ψήφους

Ο Χάρης πήρε ψήφους

Ο Χρίστος πήρεψήφους

ο μέσος όρος των ψήφων είναι ψήφοι

Η διαφορά από το μέσο όρο είναι:

για το Γιώργο ψήφοι

για το Χάρη ψήφοι

για το Χρίστο ψήφοι

Λύση

```

program ennea_4;
var
    ps1,ps2,ps3:integer;
    mo,diafora:real;
begin
    writeln('Ψήφοι Γιώργου ');
    readln(ps1);
    writeln('Ψήφοι Χάρη');
    read(ps2);
    writeln('Ψήφοι Χρίστου');
    read(ps3);
    writeln('Ο Γιώργος πήρε ',ps1, ' ψήφους');
    writeln('Ο Χάρης πήρε ',ps2, ' ψήφους');
    writeln('Ο Χρίστος πήρε ',ps3, ' ψήφους');
    mo:=(ps1+ps2+ps3)/3;
    writeln('Ο μέσος όρος των ψήφων είναι ', mo:10:2, ' ψήφοι');
    writeln('Η διαφορά από το μέσο όρο είναι:');
    diafora:=mo-ps1;
    writeln('για το Γιώργο ' ,diafora:10:2,' ψήφοι');
    diafora:=mo-ps2;
    writeln('για το Χάρη ',diafora:10:2,' ψήφοι');
    diafora:=mo-ps3;
    writeln('για το Χρίστο' ,diafora:10:2,' ψήφοι');
end.

```

5. Να γράψετε πρόγραμμα που να διαβάζει το μήκος της ακτίνας ενός κύκλου και να τυπώνει τη διάμετρο, το μήκος και το εμβαδόν αυτού του κύκλου.

αποτελέσματα:

για κύκλο ακτίνας **** εκ.

η διάμετρος είναι ***** εκ.

το μήκος της περιφέρειας είναι ***** εκ.

το εμβαδόν του κύκλου είναι ***** τετρ.εκ

Λύση

```
program ennea_pente;
const pi=3.14;
var r,diam,perifereia,emvadon:real;
begin
  writeln('Δώστε την ακτίνα');
  readln(r);
  diam:=2*r;
  perifereia:=2*pi*r;
  emvadon:=pi*r*r;
  writeln('Για κύκλο ακτίνας ',r:10:2,' εκ.');
```

```
writeln('Η διάμετρος είναι ',diam:10:2,' εκ.');
```

```
writeln('Το μήκος της περιφέρειας είναι ',perifereia:10:2,' εκ.');
```

```
writeln('Το εμβαδόν είναι ',emvadon:10:2,' τετρ.εκ.')
```

```
end.
```

6. Να γράψετε πρόγραμμα που να διαβάζει ένα τριψήφιο ακέραιο αριθμό και να τον τυπώνει ανάστροφα. Π.χ. αν ο αριθμός είναι 123, ο ανάστροφος είναι 321.

Λύση

[Σκεπτικό: Απομονώνουμε τα ψηφία του αριθμού διαιρώντας διαδοχικά με 100 και 10].

```
program ennea_6;
var a,a1,a2,a3:integer;
begin
  writeln('Γράψε έναν τριψήφιο αριθμό');
  readln(a);
  a1:=a div 100;
  a:=a mod 100;
  a2:=a div 10;
  a3:= a mod 10;
  write(a3,a2,a1)
end.
```

Λύση β'

[Με τη χρήση μιας επιπλέον μεταβλητής, μπορεί να συντεθεί ο ακέραιος με την αντίστροφη σειρά των ψηφίων και στη συνέχεια να εκτυπωθεί].

```
program ennea_6_1;
var a,a1,a2,a3,k:integer;
begin
  writeln('Γράψε έναν τριψήφιο αριθμό');
  readln(a);
```

```

a1:=a div 100;
a:=a mod 100;
a2:=a div 10;
a3:= a mod 10;
k:=s3*100+a2*10+a1;
write (k);
end.

```

Παρατήρηση

[Αν στην εκφώνηση δεν αναφερόταν ρητά ότι το δεδομένο εισόδου πρέπει να είναι ακέραιος αριθμός, θα μπορούσαν να διαβαστούν τα ψηφία ως χαρακτήρες και στη συνέχεια να εκτυπωθούν με την αντίστροφη σειρά].

```

program ennea_6;
var char1, char2, char3:char;
begin
  writeln('Γράψε έναν τριψήφιο αριθμό');
  read(char1);
  read(char2);
  read(char3);
  writeln();
  write(char3, char2, char1)
end.

```

7. Να γράψετε το ίδιο πρόγραμμα για έναν τετραψήφιο αριθμό.

Λύση

```

program ennea_7;
var a, a1, a2, a3, a4:integer;
begin
  writeln('Γράψε έναν τετραψήφιο αριθμό');
  readln(a);
  a1:=a div 1000;
  a:=a mod 1000;
  a2:=a div 100;
  a:=a mod 100;
  a3:= a div 10;
  a4:= a mod 10;
  writeln(a4, a3, a2, a1)
end.

```

8. Να γράψετε πρόγραμμα που να διαβάζει έναν πραγματικό αριθμό και να τυπώνει τις 5 πρώτες δυνάμεις του και το άθροισμα των 3 πρώτων δυνάμεων του.

Τα αποτελέσματα να έχουν τη μορφή:

αριθμός :

1η δύναμη του αριθμού :

2η δύναμη του αριθμού :

3η δύναμη του αριθμού :

4η δύναμη του αριθμού :

5η δύναμη του αριθμού :

άθροισμα των δυνάμεων 1,2,3 :

Λύση

```
program ennea_8;
var number, sum, dyn: real;
begin
  sum:=0;
  writeln('Δώστε έναν πραγματικό αριθμό');
  readln(number);
  writeln('Η 1η δύναμη του αριθμού: ', number:10:2);
  sum:=number;
  dyn:=number*number;
  writeln('Η 2η δύναμη του αριθμού: ', dyn:10:2);
  sum:=sum+dyn;
  dyn:=number*dyn;
  writeln('Η 3η δύναμη του αριθμού: ', dyn:10:2);
  sum:=sum+dyn;
  dyn:=number*dyn;
  writeln('Η 4η δύναμη του αριθμού: ', dyn:10:2);
  dyn:=number*dyn;
  writeln('Η 5η δύναμη του αριθμού: ', dyn:10:2);
  write('Το άθροισμα των τριών πρώτων δυνάμεων είναι:');
  writeln(sum:10:2)
end.
```

9. Να γράψετε πρόγραμμα που να διαβάζει θερμοκρασία σε βαθμούς Fahrenheit και να τη μετατρέπει σε βαθμούς Κελσίου. Ο τύπος μετατροπής είναι:

$$\frac{f-32}{9} = \frac{c}{5}$$

Λύση

[Λύνοντας ως προς c, ο τύπος γίνεται: $c = \frac{5}{9} * (f - 32)$].

```
program ennea_9;
var f, c: real;
begin
  writeln('Δώστε τη θερμοκρασία σε Φαρενάιτ');
  readln(f);
  c:=5*(f-32)/9;
  writeln('Η θερμοκρασία σε βαθμούς Κελσίου είναι:');
  writeln(c:10:2);
end.
```

10. Να γράψετε πρόγραμμα που να διαβάζει τους συντελεστές α, β, της πρωτοβάθμιας εξίσωσης $ax+b=0$ και να υπολογίζει τη ρίζα της. (Υποτίθεται ότι α, β είναι διάφορα του μηδενός).

Λύση

```
program ennea_10;
```

```

var
  a,b:integer;
  x:real;
begin
  writeln('Δώστε το α');
  readln(a);
  writeln('Δώστε το β');
  readln(b);
  x:=-b/a;
  writeln('Η ρίζα χ της εξίσωσης είναι:');
  writeln(x:10:2)
end.

```

11. Να γράψετε πρόγραμμα που να διαβάζει ημερομηνία με τη μορφή μέρα/μήνας/έτος και να την τυπώνει με τη μορφή έτος/μήνας/μέρα αποτελέσματα:
 διαβάστηκε η ημερομηνία ../../..
 που μετατρέπεται στην ../../..

Λύση

```

program ennea_10;
var
  s1,s2,s3:String;
begin
  writeln('Δώστε ημέρα');
  readln(s1);
  writeln('Δώστε το μήνα');
  readln(s2);
  writeln('Δώστε το έτος');
  readln(s3);
  writeln(s1,'/',s2,'/',s3);
  writeln(s3,'/',s2,'/',s1);
end.

```

Κεφάλαιο 10

ΕΝΤΟΛΕΣ ΕΠΙΛΟΓΗΣ ΚΑΙ ΑΠΟΦΑΣΕΩΝ

Ερωτήσεις

1. Πώς μπορεί να αλλάξει η σειρά εκτέλεσης των εντολών;

Η σειρά εκτέλεσης των εντολών μπορεί να αλλάξει με τη χρήση εντολών ελέγχου και επιλογής.

2. Πώς επιτυγχάνεται η εκτέλεση ορισμένων εντολών υπό συνθήκες;

Μια εντολή ελέγχου, περιέχει μια λογική έκφραση. Όταν η τιμή της λογικής έκφρασης είναι αληθής εκτελείται μία συγκεκριμένη εντολή (ή μια συγκεκριμένη ομάδα εντολών). Αν η τιμή της λογικής έκφρασης είναι ψευδής τότε εκτελείται μια άλλη εντολή (ή μια άλλη ομάδα εντολών).

3. Να εξηγήσετε την παρακάτω εντολή και να την τροποποιήσετε ώστε να περιλαμβάνει και την περίπτωση εκείνη στην οποία ο αριθμός είναι περιττός. Να δώσετε παραδείγματα και τα αποτελέσματα της εκτέλεσης των εντολών.

```
If arithm mod 2 = 0  
Then writeln('ο αριθμός είναι άρτιος');
```

Αν ο το περιεχόμενο¹⁷ της μεταβλητής `arithm` διαιρείται ακριβώς με το 2, τότε εμφανίζεται το μήνυμα «Ο αριθμός είναι άρτιος». Ο κώδικας πρέπει να συμπληρωθεί με την περίπτωση στην οποία το περιεχόμενο της μεταβλητής δεν διαιρείται ακριβώς με το δύο οπότε πρέπει να βγαίνει το μήνυμα ότι «Ο αριθμός είναι περιττός».

```
if arithm mod 2 = 0  
then  
    writeln('ο αριθμός είναι άρτιος')  
else  
    writeln('ο αριθμός είναι περιττός')
```

4. Να εξηγήσετε την παρακάτω εντολή και να δώσετε παραδείγματα και τα αποτελέσματα της εκτέλεσης της εντολής.

```
if diaretis <>0  
then pililko:=diareteos div diaretis  
else writeln('διαίρεση με μηδέν δεν είναι επιτρεπτή');
```

Στην εντολή ελέγχου υπάρχει η συνθήκη `diaretis<>0`, όπου εξετάζεται αν το περιεχόμενο της μεταβλητής `diaretis` είναι διάφορο του μηδενός. Αν αυτό ισχύει, είναι δηλαδή η συνθήκη αληθής τότε εκτελείται η εντολή μετά το `then`, όπου υπολογίζεται το ακέραιο πηλίκο της διαίρεσης των μεταβλητών `diareteos` και `diaretis` και το αποτέλεσμα εκχωρείται στη μεταβλητή `pililko`. Αν η συνθήκη δεν ισχύει εκτελείται η εντολή μετά το `else` και εμφανίζεται το μήνυμα «διαίρεση με το μηδέν δεν είναι επιτρεπτή».

5. Να χρησιμοποιήσετε την εντολή if για τον έλεγχο της ηλικίας ενός ψηφοφόρου¹⁸ και να δώσετε το κατάλληλο μήνυμα ανάλογα με την περίπτωση για ηλικία > 65, για ηλικία <18 και για $18 \leq \text{ηλικία} \leq 65$.

```
program erwtisi10_5;
```

¹⁷ (ακέραιο)

¹⁸ Η άσκηση δε ζητά να γίνεται έλεγχος έγκυρης ηλικίας εισόδου.


```

var ilikia:integer;
begin
  writeln ('Δώστε την ηλικία του ψηφοφόρου');
  readln(ilikia);
  if ilikia<18
  then
    writeln ('Δεν έχετε δικαίωμα ψήφου')
  else
    if ilikia <=65
    then
      writeln ('Πρέπει να ψηφίσετε')
    else
      writeln ('Δεν είστε υποχρεωμένος να ψηφίσετε');
    end.
  end.

```

6. Να χρησιμοποιήσετε την εντολή case για τον έλεγχο της ηλικίας ενός ψηφοφόρου και να δώσετε το κατάλληλο μήνυμα ανάλογα με την περίπτωση, για ηλικία > 65, για ηλικία <18 και για $18 \leq \text{ηλικία} \leq 65$.

Απάντηση 1

```

program erwtisi10_6;
var ilikia:integer;
begin
  writeln ('Δώστε την ηλικία του ψηφοφόρου');
  readln(ilikia);
  case ilikia of
    0 .. 17: writeln ('Δεν έχετε δικαίωμα ψήφου');
    18 .. 65: writeln ('Πρέπει να ψηφίσετε');
    else writeln('Δεν είστε υποχρεωμένος να ψηφίσετε');
  end
end.

```

Απάντηση2

Επειδή ο τελεστής «..» δεν περιλαμβάνεται στη θεωρία του σχολικού βιβλίου, μια δεύτερη (λίγο πιο πολύπλοκη) απάντηση θα ήταν:

```

program erwtisi10_6_1;
var ilikia:integer;
begin
  writeln ('Δώστε την ηλικία του ψηφοφόρου');
  readln(ilikia);
  case ilikia < 18 of
    true: writeln ('Δεν έχετε δικαίωμα ψήφου');
    false:
      begin
        case ((ilikia>=18) and (ilikia<=65)) of
          true: writeln('Πρέπει να ψηφίσετε!');
          false:writeln('Δεν είστε υποχρεωμένος να ψηφίσετε');
        end;
      end;
  end
end.

```

7. Να συγκρίνετε την εντολή `case` με την `if` και να δώσετε δύο παραδείγματα χρήσης της εντολής `case`.

Η εντολή `case` είναι χρήσιμη γιατί βοηθά τη δόμηση ενός προγράμματος και το καθιστά πιο ευανάγνωστο αντικαθιστώντας τα πολλά `if`.

Παραδείγματα

Την εντολή `case` μπορούμε να χρησιμοποιήσουμε στην περίπτωση ενός μενού με πολλαπλές επιλογές (σελ. 95) ή στην περίπτωση που έχουμε εκχώρηση τιμής σε διαφορετικές μεταβλητές ανάλογα με την τιμή ενός ακεραίου (σελ. 95).

8. Να γράψετε την επόμενη κωδικοποίηση χρησιμοποιώντας `case` αντί της `if`.

```
if n=5
    then tm5:=tm5+1
    else if n=15
        then tm15:=tm15+1
        else if n=25
            then tm25:=tm25+1;
```

Με τη χρήση της εντολής `case`, η παραπάνω εντολή `if` γίνεται:

```
case n of
    5: tm5:=tm5+1;
    15:tm15:=tm15+1;
    25:tm25:=tm25+1;
end;
```

9. Να γράψετε μία εντολή `case` η οποία ανάλογα με το χαρακτηρισμό της βαθμολογίας τυπώνει το αντίστοιχο μήνυμα. Αν ο χαρακτηρισμός=1 τύπωσε άριστα, 2 λίαν καλώς, 3 καλώς, 4 σχεδόν καλώς, 5 απορρίπτεται, αλλιώς λάθος χαρακτηρισμός. Να επαναλάβετε τα προηγούμενα με την εντολή `if`.

A) Με τη χρήση της εντολής `case` έχουμε:

```
program test;
var vathmos:integer;

begin
    writeln ('Δώστε τη βαθμολογία');
    readln(vathmos);
    case vathmos of
        1: writeln ('Άριστα');
        2: writeln ('Πολύ Καλά');
        3: writeln ('Καλά');
        4: writeln ('Σχεδόν καλά');
        5: writeln ('Απορρίπτεται');
    else
        writeln ('Λάθος βαθμός');
    end;
end.
```

B) Με τη χρήση της εντολής `if` έχουμε:

```

program test;
var vathmos:integer;
begin
  writeln ('Δώστε τη βαθμολογία');
  readln(vathmos);
  if vathmos=1
  then
    writeln ('Άριστα')
  else
    if vathmos=2
    then
      writeln('Πολύ Καλά')
    else
      if vathmos=3
      then
        writeln(' Καλά')
      else
        if vathmos=4
        then
          writeln('Σχεδόν καλά')
        else
          if vathmos=5
          then
            writeln('Απορρίπτεται')
          else
            writeln ('Λάθος βαθμός');

end.

```

10. Να συμπληρώσετε με Σωστό - Λάθος τις παρακάτω προτάσεις:

- i. Η έκφραση της εντολής case μπορεί να δώσει τιμές τύπου ακέραιου, πραγματικού ή λογικού. (Λ, μπορεί να είναι οποιουδήποτε βαθμωτού τύπου εκτός του πραγματικού)
- ii. Οι τιμές που παίρνουν οι ετικέτες στην εντολή case μπορεί να έχουν οποιαδήποτε σειρά, αλλά οι ετικέτες δεν επαναλαμβάνονται. (Σ)
- iii. Όλες οι δυνατές τιμές της έκφρασης σε μια εντολή case πρέπει να περιλαμβάνονται στις λίστες τιμών των ετικετών. (Λ, Δεν είναι ανάγκη να περιλαμβάνονται όλες)

Ασκήσεις

1. Να γράψετε πρόγραμμα που να ελέγχει το ποσό ανάληψης ενός καταθέτη σε μία Τράπεζα. Σε περίπτωση που η ανάληψη είναι μεγαλύτερη από τις καταθέσεις τυπώνει απαγορευτικό μήνυμα, αλλιώς τυπώνει το ποσό ανάληψης και το υπόλοιπο των καταθέσεων.

Λύση

```

program deka_1;
{Το syn_kat αντιστοιχεί στο σύνολο των καταθέσεων
 ενώ η μεταβλητή analipsi στην ανάληψη που γίνεται κάθε
 φορά. Στο syn_kat δίνεται η αυθαίρετη τιμή 100000.

```

Κανονικά, θα πρέπει να υπάρχει επικοινωνία με τη Β.Δ. της τράπεζας για να αναγνωστεί το υπόλοιπο των καταθέσεων}.

```
var syn_kat, analipsi: longint;  
begin  
  syn_kat := 100000;  
  writeln('Ποσό ανάληψης');  
  readln(analipsi);  
  if analipsi > syn_kat  
  then  
    writeln('Ποσό ανάληψης μεγαλύτερο του ποσού κατάθεσης')  
  else  
    begin  
      syn_kat := syn_kat - analipsi;  
      writeln('Ανάληψη= ', analipsi);  
      writeln('Υπόλοιπο ποσού Κατάθεσης=', syn_kat);  
    end  
  end.  
end.
```

2. Να γράψετε πρόγραμμα που να διαβάζει 4 βαθμούς, να βρίσκει το μέσο όρο τους και, αν αυτός είναι μεγαλύτερος από 15, τυπώνει ΕΠΙΤΥΧΩΝ.

Λύση

```
program deka_2;  
var  
  b1, b2, b3, b4: integer;  
  mo: real;  
begin  
  writeln('Δώστε τους τέσσερις βαθμούς για τα μαθήματα  
χωρισμένους με κενά');  
  readln(b1, b2, b3, b4);  
  mo := (b1 + b2 + b3 + b4) / 4;  
  if mo > 15  
  then  
    writeln('Επιτυχών!');  
  end.  
end.
```

3. Η ωριαία αμοιβή εργαζομένου είναι 6€. Αν οι ώρες εργασίας είναι περισσότερες από 18, παίρνει επιπλέον υπερωριακή αποζημίωση 3€ για κάθε υπερωριακή ώρα. Να γράψετε πρόγραμμα που να διαβάζει τις ώρες εργασίας και να υπολογίζει τις αποδοχές του εργαζομένου. Τα αποτελέσματα να ακολουθήσουν το υπόδειγμα:

κανονική αμοιβή

αμοιβή υπερωριών

συνολικές αποδοχές

Υπόδειξη: Να γίνουν δύο τουλάχιστον εκτελέσεις του προγράμματος, μία με υπερωρίες και μία χωρίς.

Λύση

```
program deka_3;
const
    poso_wras=6;
    poso_yperwrias=3;
    orio=18;
var
    (*ar_wrwn=αριθμός ωρών, kan_am=κανονική αμοιβή,
    yper_am=αμοιβή υπερωριών*)
    ar_wrwn,kan_am,yper_am,synolo:integer;

begin
    yper_am:=0;
    writeln('Δώστε τον αριθμό των ωρών');
    readln(ar_wrwn);
    if ar_wrwn<=orio
    then
        kan_am:=ar_wrwn*poso_wras
    else
        begin
            kan_am:=orio*poso_wras;
            yper_am:=(ar_wrwn-orio)*poso_yperwrias;
        end;
    synolo:=kan_am+yper_am;
    writeln('κανονική αμοιβή= ', kan_am);
    writeln('αμοιβή υπερ/ριών= ', yper_am);
    writeln('-----');
    writeln('συνολικές αποδοχές= ', synolo)
end.
```

4. Να μετατρέψετε τα προηγούμενα προγράμματα if — - then —-else κάνοντας χρήση της εντολής case.

Λύση

```
1)
program deka_4_1;
{Το syn_kat αντιστοιχεί στο σύνολο των καταθέσεων
 ενώ η μεταβλητή analipsi στην ανάληψη που γίνεται κάθε
 φορά}
var syn_kat,analipsi:longint;
begin
    syn_kat:=100000;
    writeln('Ποσό ανάληψης');
    readln(analipsi);

    case analipsi of
        0 .. 100000 :
            begin
                syn_kat:=syn_kat-analipsi;
```

```

        writeln('Ανάληψη= ', analipsi);
        writeln('Υπόλοιπο ποσού Κατάθεσης=', syn_kat);
    end;
else
    writeln('Ποσό ανάληψης μεγαλύτερο του ποσού κατάθεσης');
end
end.

```

2)

```

program deka_4_2;
var
    b1,b2,b3,b4:integer;
    mo:real;
begin
    mo:=0.0;
    writeln('Δώστε τους τέσσερις βαθμούς για τα μαθήματα
χωρισμένους με κενά');
    readln(b1,b2,b3,b4);
    mo:=(b1+b2+b3+b4)/4;
    case mo>15.0 of
        true:writeln ('Επιτυχών');
    end
end.

```

3)

```

program deka_4_3;
const
    poso_wras=6;
    poso_yperwrias=3;
    orio=18;
var
    (*ar_wrwn=αριθμός ωρών, kan_am=κανονική αμοιβή, yper_am=αμοιβή
υπερωριών*)

    ar_wrwn,kan_am,yper_am,synolo:integer;

begin
    kan_am:=0;
    yper_am:=0;
    writeln('Δώστε τον αριθμό των ωρών');
    readln(ar_wrwn);
    case ar_wrwn of
        1 .. 18: kan_am:=ar_wrwn*poso_wras;
    else
        begin
            kan_am:=orio*poso_wras;
            yper_am:=(ar_wrwn-orio)*poso_yperwrias;
        end;
    end;
    synolo:=kan_am+yper_am;
    writeln('κανονική αμοιβή= ', kan_am);
    writeln('αμοιβή υπερ/ριών= ', yper_am);

```

```
writeln('-----');
writeln('συνολικές αποδοχές= ', synolo);
```

end.

5. Να γράψετε πρόγραμμα που να διαβάζει το αρχικό ενός ονόματος και να τυπώνει το αντίστοιχο όνομα (ονόματα: Πέτρος, Χριστίνα, Νίκος, Άννα, Γιώργος).

Λύση

```
program deka_5;
var ch:char;
begin
  writeln('Δώστε το αρχικό');
  readln(ch);
  case ch of
    'Π': writeln('Πέτρος');
    'Χ': writeln('Χριστίνα');
    'Ν': writeln('Νίκος');
    'Α': writeln('Άννα');
    'Γ': writeln('Γιώργος');
  end;
end.
```

6. Να γράψετε πρόγραμμα που να εμφανίζει menu με τις 4 πράξεις:

1. πρόσθεση
2. αφαίρεση
3. πολλαπλασιασμός
4. διαίρεση,

να διαβάζει τον κωδικό της πράξης και τους δύο αριθμούς και να εμφανίζει το αποτέλεσμα.(Διαίρεση με μηδέν δε γίνεται).

Λύση

```
program deka_6;
var choice:integer;
    n1,n2,apotelesma:real;
    flag:boolean;
{Σε περίπτωση επιλογής πράξης έξω από τα όρια 1-4 το flag
γίνεται false}
begin
  flag:=true;
  writeln ('1.Πρόσθεση');
  writeln ('2.Αφαίρεση');
  writeln ('3.Πολλαπλασιασμός');
  writeln ('4.Διαίρεση');
  writeln ('Δώστε την πράξη επιλέγοντας 1 έως 4');
  readln(choice);
  writeln ('Δώστε τους αριθμούς, πατώντας σε κάθε έναν en-
ter');
  readln(n1,n2);
  case choice of
    1: apotelesma:=n1+n2;
    2: apotelesma:=n1-n2;
```

```

3: apotelesma:=n1*n2;
4:
   begin
     if n2 <> 0 then
       apotelesma := n1/n2
     else
       writeln('Αδύνατη πράξη')
     end
   else
     begin
       writeln('Μη επιτρεπτή επιλογή');
       flag:=false;
     end
   end;
if flag=true
then
  writeln ('Το αποτέλεσμα είναι ',apotelesma:10:2);
end.

```

7. Να γράψετε πρόγραμμα που να διαβάζει δύο ηλικίες και να δίνει τη διαφορά τους. Η εισαγωγή στοιχείων και τα αποτελέσματα να είναι σύμφωνα με το υπόδειγμα:

Δεδομένα : 9 6

Αποτελέσματα:

Ο Νίκος είναι 9 ετών

Ο Γιώργος είναι 6 ετών

Ο Νίκος είναι 3 έτη μεγαλύτερος απ' το Γιώργο

Να γίνουν άλλες δύο εκτελέσεις με δεδομένα 6, 9 και 9, 9.

Λύση

```

program deka_7;
var n1,n2,diafora:integer;
begin
  writeln ('Δώστε τις δύο ηλικίες');
  readln(n1,n2);
  diafora:=n1-n2;
  if diafora > 0
  then
    writeln('Ο Γιώργος είναι ', diafora,' χρόνια μεγαλύτερος από τον
Νίκο')
  else
    if diafora < 0
    then
      begin
        diafora:=-diafora;
        writeln('Ο Νίκος είναι ', diafora,' χρόνια μεγαλύτερος
από τον Γιώργο');
      end
    else
      writeln('Ο Γιώργος είναι συνομήλικος με τον Νίκο ')
    end
  end
end.

```


end.

8. Για να ψηφίσει ένας πολίτης, πρέπει να είναι τουλάχιστον 18 ετών. Αν όμως είναι άνω των 70, δεν υποχρεώνεται να ψηφίσει. Να γράψετε πρόγραμμα που να διαβάζει την ηλικία και να δίνει το κατάλληλο μήνυμα.

Υπόδειξη¹⁹: Να χρησιμοποιήσετε το δυαδικό²⁰ τελεστή and. Να γίνουν 3 διαφορετικές εκτελέσεις με αποτελέσματα:

Είναι χρονών υποχρεώνεται να ψηφίσει.

Είναιχρονών δεν υποχρεώνεται να ψηφίσει.

Είναι χρονών δεν μπορεί να ψηφίσει.

Λύση

```
program deka_8;
var ilikia:integer;
begin
  writeln('Δώστε την ηλικία');
  readln(ilikia);
  if ilikia < 18
  then
    writeln('Είναι ', ilikia, ' χρόνων. Δεν μπορεί να
ψηφίσει!')
  else
    if (ilikia>=18) and (ilikia <=70)
    then
      writeln('Είναι ', ilikia, ' χρόνων. Είναι
υποχρεωμένος να ψηφίσει.')
    else
      writeln('Είναι ', ilikia, ' χρόνων. Δεν
είναι υποχρεωμένος να ψηφίσει.');
```

end.

Δραστηριότητες

Να κωδικοποιήσετε σε γλώσσα Pascal τους αλγορίθμους των ασκήσεων 3,4 και 5 του ΚΕ-ΦΑΛΑΙΟΥ 4 ΑΝΑΠΤΥΞΗ ΑΛΓΟΡΙΘΜΩΝ

Άσκηση 3 ΚΕΦ.4

```
program drastiriotita_kef10_3;
const orio=18;
  {on_ep --> ονοματεπώνυμο μαθητή, vathmos --> ο βαθμός,
  plithos --> το πλήθος των μαθητών με βαθμό άνω των 18}
var on_ep:string;
    vathmos, plithos:integer;
```

¹⁹ Θα ήταν προτιμότερο να λυθεί με τον τρόπο που εξετάζεται στην ερώτηση 6.

²⁰ Προφανώς εννοεί «λογικό».

```

begin
  plithos:=0;
  repeat
    writeln('Δώσε το ονοματεπώνυμο (0 για τέλος)');
    readln(on_ep);
    if on_ep<> '0'
      then
        begin
          {έλεγχος για έγκυρο βαθμό}
          repeat
            writeln('Δώσε τον βαθμό');
            readln(vathmos);
          until (vathmos>0) and (vathmos<=20);
          end;
        if vathmos>18
          then
            plithos:=plithos+1;
        until on_ep='0';
    writeln(' πλήθος των μαθητών με βαθμό άνω των 18=',plithos);
  end.

```

Άσκηση 4 ΚΕΦ.4

```

program drastiriotita_kef10_4;
{vathmos --> 0 βαθμός του μαθητή, n -->το πλήθος των μαθημάτων}
{m1_10,m11_15,m16_18,m_19_20, οι μετρητές για τις αντίστοιχες
περιοχές βαθμών}

var vathmos,n,m1_10,m11_15,m16_18,m19_20,i:integer;
begin
  m1_10:=0;
  m11_15:=0;
  m16_18:=0;
  m19_20:=0;
  writeln('Δώστε τον αριθμό των μαθημάτων ');
  readln(n);
  for i:=1 to n do
    begin
      repeat
        writeln('Δώστε το βαθμό για το ', i, 'ο μάθημα');
        readln(vathmos);
      until (vathmos>0) and (vathmos<=20);
      case vathmos of
        1..10:m1_10:=m1_10+1;
        11..15:m11_15:=m11_15+1;
        16..18:m16_18:=m16_18+1;
        19..20:m19_20:=m19_20+1;
      end;
    end;
  end;

```

```

        writeln('Ο αριθμός των μαθημάτων με βαθμό έως και 10
είναι', m1_10);
        writeln('Ο αριθμός των μαθημάτων με βαθμό 11 έως και 15
είναι', m11_15);
        writeln('Ο αριθμός των μαθημάτων με βαθμό 16 έως και 18
είναι', m16_18);
        writeln('Ο αριθμός των μαθημάτων με βαθμό 19 έως και 120
είναι', m19_20)
end.

```

Άσκηση 5 ΚΕΦ.4

```

program drastiriotita_kef10_5;
{g1,g2,g3,l1,l2,l3,e1,e2,e3 --> Οι μετρητές για τις Α,Β,Γ Γυ-
μνασίου, ΓΕΛ, ΕΠΑ.Λ αντίστοιχα}
var g1,g2,g3,l1,l2,l3,e1,e2,e3,n,i:integer;
    taksi:char;
    onoma,sxoleio:string;

begin
    g1:=0;g2:=0;g3:=0; l1:=0;l2:=0;l3:=0; e1:=0;e2:=0;e3:=0;
    writeln('Γράψε το πλήθος των μαθητών');
    readln(n);
    for i:=1 to n do
        begin
            writeln ('Διάβασε όνομα',i);
            readln(onoma);
            writeln('Διάβασε το Σχολείο');
            readln(sxoleio);
            writeln('Διάβασε την Τάξη');
            readln(taksi);
            if sxoleio='ΓΥΜΝΑΣΙΟ'
            then
                begin
                    case taksi of
                        'A':g1:=g1+1;
                        'B':g2:=g2+1;
                        'Γ':g3:=g3+1;
                    end;
                end
            else
                if sxoleio='ΓΕΛ'
                then
                    begin
                        case taksi of
                            'A':l1:=l1+1;
                            'B':l2:=l2+1;
                            'Γ':l3:=l3+1;
                        end;
                    end
                end
            end
        end
    end
end

```

```

else if sxoleio= 'ΕΠΑΛ'
  then
    begin
      case taksi of
        'A':e1:=e1+1;
        'B':e2:=e2+1;
        'Γ':e3:=e3+1;
      end;
    end
  else
    writeln('Μη έγκυρος τύπος σχολείου');
end;
writeln('Γυμνάσιο, A, B, Γ αντίστοιχα =',g1,',',g2,',',g3);
writeln('Γενικό Λυκ., A, B, Γ αντίστοιχα =',l1,',',l2,',',l3);
writeln('ΕΠΑ.Λ., A, B, Γ αντίστοιχα =',e1,',',e2,',',e3);
end.

```

Κεφάλαιο 11

ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ

Ερωτήσεις

1. **Να δώσετε τη μορφή της εντολής επανάληψης while, να εξηγήσετε τη λειτουργία της και να δώσετε δυο δικά σας παραδείγματα.**

`while` λογική έκφραση `do` εντολή

Κάθε φορά, ακόμα και την πρώτη, **πριν εκτελεστεί η εντολή** που αποτελεί το σώμα της επανάληψης (αυτή που βρίσκεται μετά το `do`) ελέγχεται η **λογική έκφραση** η οποία ακολουθεί το `while`. Στην περίπτωση που η λογική έκφραση δίνει αποτέλεσμα `true`, η εντολή μετά το `do` εκτελείται ενώ στην περίπτωση που η λογική έκφραση δίνει ως αποτέλεσμα `false` η εντολή μετά το `do` δεν εκτελείται και ο έλεγχος περνά στην επόμενη εντολή του προγράμματος. Στην περίπτωση που μετά το `do` αντί για μία, έχουμε ομάδα εντολών, αυτές τοποθετούνται μεταξύ των λέξεων `begin – end`. Η εντολή `while` χρησιμοποιείται για μη προκαθορισμένο αριθμό επαναλήψεων.

Παραδείγματα. Δείτε στις σελίδες 100, 101 του βιβλίου

2. **Να δώσετε τη μορφή της εντολής επανάληψης repeat until, να εξηγήσετε τη λειτουργία της και να δώσετε δυο δικά σας παραδείγματα.**

```
repeat
  εντολή_1 [ ;
  εντολή_2 ;
  .....
  εντολή_v ]
until (λογική έκφραση) ;
```

Η εντολή `repeat` χρησιμοποιείται για μη προκαθορισμένο αριθμό επαναλήψεων. Ο έλεγχος για την επανάληψη γίνεται στο τέλος του κύκλου των εντολών που επαναλαμβάνονται. Το χαρακτηριστικό της επανάληψης `repeat` είναι ότι η εντολή / ες που περιέχονται, εκτελούνται τουλάχιστον μια φορά.

Παραδείγματα. Δείτε στη σελίδα 102 του βιβλίου.

3. **Να δώσετε τη μορφή της εντολής επανάληψης for, να εξηγήσετε τη λειτουργία της και να δώσετε δυο δικά σας παραδείγματα.**

```
for μεταβλητή:=αρχική_τιμή to τελική_τιμή do
  Εντολή ;
```

Η εντολή που ακολουθεί το `do` μπορεί να είναι απλή ή σύνθετη²¹ και εκτελείται μέχρις ότου η μεταβλητή ελέγχου αποκτήσει τιμή ίση με την τελική τιμή. Αν η αρχική τιμή είναι μεγαλύτερη από την τελική, οι εντολές που ακολουθούν το `do` δεν εκτελούνται. Η εντολή `for` είναι η εντολή προκαθορισμένης επανάληψης διότι ο αριθμός των επαναλήψεων είναι εκ των προτέρων γνωστός.

Αν η μεταβλητή ελέγχου αρχίζει από τη μεγαλύτερη τιμή μέχρι τη μικρότερη, τότε η μορφή της εντολής είναι η παρακάτω:

```
for μεταβλητή:=αρχική_τιμή downto τελική_τιμή do
  Εντολή ;
```

Παραδείγματα. Δείτε στη σελίδα 104 του βιβλίου.

²¹ Σύνθετη εντολή είναι η εντολή που περιλαμβάνεται ανάμεσα στις λέξεις `begin-end` (δείτε σελ. 94)

4. Να συγκρίνετε τις εντολές επανάληψης και να αναφέρετε ποιες είναι οι ομοιότητες και ποιες οι διαφορές τους.

Ομοιότητες: while, repeat, for

Και οι τρεις είναι εντολές επανάληψης. Επαναλαμβάνουν δηλαδή μια εντολή ή μια ομάδα εντολών όσες φορές απαιτείται.

Διαφορές: while, repeat, for

Οι εντολές **while** και **repeat** χρησιμοποιούνται για μη προκαθορισμένο αριθμό επαναλήψεων. Σε περίπτωση που ο αριθμός των επαναλήψεων είναι προκαθορισμένος, είναι πιο βολικό να χρησιμοποιείται η **for**.

Διαφορά while, repeat

Η εντολή ή οι εντολές που περιλαμβάνονται στην **while** – ανάλογα με τη συνθήκη – μπορεί να μην εκτελεστούν ούτε μία φορά ενώ η εντολή / εντολές που περιλαμβάνονται στην **repeat** θα εκτελεστούν τουλάχιστον μία φορά²².

Εάν οι εντολές στο σώμα επανάληψης είναι πάνω από μία, στην επανάληψη **while** θα πρέπει να αποτελέσουν σύνθετη εντολή μπαίνοντας ανάμεσα σε **begin – end** ενώ στην εντολή **repeat** αρκεί να μπουν ανάμεσα στο **repeat** και στο **until**.

Διαφορά for, while, repeat

Ο μετρητής στην εντολή **for** μπορεί να αυξάνει (ή να μειώνεται) κατά 1. Αν χρειάζεται αύξηση ή μείωση με άλλη τιμή τότε χρησιμοποιούμε το **while** (ή το **repeat**).

5. Να δώσετε τα αποτελέσματα από την εκτέλεση των παρακάτω εντολών

```
arithmos:=1;
while arithmos < 11 do
begin
arithmos:=arithmos + 1;
write(arithmos:5);
end;
```

Το αποτέλεσμα θα είναι να τυπωθούν οι αριθμοί από 2 μέχρι και 11. «Εισερχόμενη» στο **while** η μεταβλητή **arithmos**, αυξάνει κατά 1, άρα η εκτύπωση ξεκινά από το 2. Η τελευταία τιμή που κάνει τη συνθήκη **arithmos < 11** αληθή είναι η τιμή 10 η οποία μπαίνοντας στο βρόχο αυξάνει κατά 1 άρα η τελική τιμή που θα εκτυπωθεί είναι το 11.

6. Να αλλάξετε τη σειρά των εντολών της προηγούμενης ερώτησης ώστε το αποτέλεσμα να είναι η εκτύπωση των αριθμών 1 έως και 10.

Αρκεί να αλλάξει η σειρά των εντολών στη σύνθετη εντολή.

```
arithmos:=1;
while arithmos < 11 do
begin
write(arithmos:5);
arithmos:=arithmos + 1;
end;
```

7. Να χρησιμοποιήσετε τις εντολές επανάληψης repeat until και for ώστε να έχετε τα ίδια αποτελέσματα με την ερώτηση 6.

Με την εντολή **repeat – until**

²² Η εντολή **while** χρησιμοποιεί τη λογική έκφραση για να συνεχίσει την επανάληψη ενώ η **repeat / until** για να τη σταματήσει.

```

program erwtisi_11_7;
var
  arithmos:integer;
begin
  arithmos:=1;
  repeat
    write (arithmos:5);
    arithmos:=arithmos+1;
  until arithmos>10
end.

```

Με την εντολή for

```

program erwtisi_11_7_1;
var
  arithmos:integer;
begin
  for arithmos:=1 to 10 do
    write (arithmos:5);
  end.

```

8. Πόσες επαναλήψεις²³ θα έχουμε από την εκτέλεση των παρακάτω εντολών;

```

telos:= false;
while not telos do
  begin
    arithmos:=arithmos + 2 ;
    if arithmos > 100
      then
        telos :=true;
  end;

```

Υποθέτουμε ότι η τιμή της μεταβλητής arithmos πριν την εκτέλεση του βρόχου είναι 1. Σε κάθε επανάληψη, η μεταβλητή αυξάνει κατά δύο (1, 3, 5, κτλ). Οι επαναλήψεις σταματούν όταν η τιμή της λογικής έκφρασης not telos είναι false δηλαδή όταν η τιμή της telos είναι true. Αυτό συμβαίνει όταν η μεταβλητή arithmos γίνει πάνω από 100 (101 στην υπόθεσή μας) άρα δηλαδή μετά από 50 επαναλήψεις.

9. Να χρησιμοποιήσετε τις εντολές επανάληψης repeat until και for, ώστε να έχετε τον ίδιο αριθμό επαναλήψεων με την ερώτηση 8.

Με τη χρήση της repeat – until

```

program erwtisi_11_9_1;
var
  arithmos:integer;
  telos:boolean;
begin
  {Υποθέτουμε ότι η μεταβλητή arithmos έχει την την τιμή 1}
  arithmos:=1;
  telos:=false;
  repeat
    arithmos:=arithmos+2;
    if arithmos > 100

```

²³ Θα πρέπει να γνωρίζουμε την τιμή της μεταβλητής arithmos πριν ξεκινήσει η εκτέλεση του βρόχου.

```

        then
            telos:=true;
        until telos=true;
    end.

```

Με τη χρήση της for

```

program erwtisi_11_9_2;
var
    arithmos, epanal:integer;
begin
    {Υποθέτουμε ότι ο αριθμός έχει την την τιμή 1}
    arithmos:=1;
    for epanal:=1 to 50 do
        arithmos:=arithmos+2;
    end.

```

10. Να δώσετε τα αποτελέσματα από την εκτέλεση των παρακάτω εντολών και να εξηγήσετε τη διαφορά που υπάρχει στη χρήση των μεταβλητών athroisma και metritis.

```

athroisma:=0;
metritis:=0;
while metritis < 10 do
begin
    athroisma:=athroisma + metritis;
    writeln(metritis:10,athroisma:10);
    metritis:= metritis + 1;
end;

```

Να τροποποιήσετε τον προηγούμενο κώδικα ώστε, να δίνει τα ίδια αποτελέσματα με τη χρήση των εντολών repeat until και for.

Η μεταβλητή metritis αυξάνει κατά ένα, μετρά δηλαδή από 0 μέχρι και 10. (Όταν η τιμή της γίνει 10, η συνθήκη στο while γίνεται ψευδής και σταματά η εκτέλεση του βρόχου). Η μεταβλητή athroisma ξεκινά από το μηδέν και αυξάνει (συσσωρεύει τιμές) προσθέτοντας την εκάστοτε τιμή της μεταβλητής metritis. Η τελική τιμή της μεταβλητής athroisma είναι 45.

Με τη χρήση της εντολής repeat – until ο κώδικας γίνεται:

```

program erwtisi_11_10_1;
var
    athroisma,metritis:integer;
begin
    athroisma:=0;
    metritis:=0;
    repeat
        athroisma:=athroisma+metritis;
        writeln(metritis:10, athroisma:10);
        metritis:=metritis+1;
    until metritis > 9;
end.

```

Με τη χρήση της εντολής for, ο κώδικας γίνεται:

Σχολικός Σύμβουλος Πληροφορικής και καθηγητές Πληροφορικής ΕΠΑ.Λ Μεσσηνίας & Λακωνίας


```

program erwtisi_11_10_2;
var
  athroisma,metritis:integer;
begin
  athroisma:=0;
  for metritis:=1 to 9 do
    begin
      athroisma:=athroisma+metritis;
      writeln(metritis:10, athroisma:10);
    end;
  end.

```

Ασκήσεις

1. Να γράψετε ένα πρόγραμμα που να υπολογίζει τη μικρότερη δύναμη του 2 που είναι μεγαλύτερη από ένα δεδομένο αριθμό και να τυπώνει ποια είναι η δύναμη αυτή.

Τα αποτελέσματα να δοθούν με τη μορφή:

Η μικρότερη δύναμη του 2

μεγαλύτερη από τον αριθμό

είναι

δηλ. είναι το 2 στην δύναμη

Υπόδειξη: Η άσκηση να λυθεί με δύο τρόπους (με χρήση WHILE και REPEAT-UNTIL).

Λύση α' - με τη χρήση του while

```

program endeka_1;
var ekthetis,dynami,arithmos:integer;

begin
  ekthetis:=1;
  writeln('Δώσε τον αριθμό');
  readln(arithmos);
  dynami:=2;

  while dynami <= arithmos do
    begin
      dynami := 2 * dynami;
      ekthetis := ekthetis+1;
    end;

  writeln ('Η μικρότερη δύναμη του 2 μεγαλύτερη από τον
αριθμό είναι ', ekthetis);
  writeln ('Δηλαδή το 2 εις την ', ekthetis);
end.

```

Λύση β' - με τη χρήση του repeat – until

```

program endeka_1_1;
var ekthetis,dynami,arithmos:integer;
begin
  ekthetis:=1;
  writeln('Δώσε τον αριθμό');
  readln(arithmos);
  dynami:=2;
  repeat
    dynami := 2 * dynami;
    ekthetis := ekthetis+1;
  until dynami > arithmos;
  writeln ('Η μικρότερη δύναμη του 2');
  writeln (' μεγαλύτερη από τον αριθμό ',arithmos);
  writeln (' είναι ', ekthetis);
  writeln ('Δηλαδή το 2 εις την ', ekthetis)
end.

```

2. Να γράψετε πρόγραμμα που να υπολογίζει το μικρότερο αριθμό Fibonacci²⁴ που δεν είναι μικρότερος από ένα προκαθορισμένο όριο²⁵. Κάθε αριθμός της σειράς Fibonacci, εκτός από τους δύο πρώτους που είναι 0 και 1, σχηματίζεται από το άθροισμα των δύο προηγούμενων του.

Υποδείξεις:

Να χρησιμοποιηθεί η εντολή while.

Τα αποτελέσματα να δοθούν με τη μορφή:

αποτελέσματα:

ο μικρότερος αριθμός Fibonacci

που δεν είναι μικρότερος από

είναι ο αριθμός

Λύση

```

program endeka_2;
var arith1,arith2,arith3,orio:integer;
begin
  writeln('Δώστε τον αριθμό - όριο');
  readln(orio);

  arith1:=0;
  arith2:=1;
  arith3:=arith1+arith2;

  while arith3 < orio do
    begin
      arith1:=arith2;
      arith2:=arith3;
      arith3:=arith1+arith2;
    end;
end;

```

²⁴ Fibonacci. Ιταλός μαθηματικός που έζησε κατά τους Μεσαιωνικούς χρόνους. Θεωρείται ίσως ο πιο ταλαντούχος μαθηματικός της εποχής του.

²⁵ Ο αριθμός-όριο πρέπει να είναι μεγαλύτερος του 2.

```

writeln ('Ο μικρότερος αριθμός Fibonacci που δεν είναι
μικρότερος από τον ', orio);
writeln ('είναι ο ', arith3)
end.

```

3. Να γράψετε πρόγραμμα που να διαβάζει:

- α. το πλήθος των εργατών μίας εταιρείας
- β. πόσες είναι οι υποχρεωτικές ώρες εργασίας (κάθε επιπλέον ώρα είναι υπερωρία)
- γ. την ωριαία αμοιβή
- δ. την επιπλέον αμοιβή για κάθε υπερωριακή ώρα
- ε. τις ώρες εργασίας κάθε εργάτη και να υπολογίζει την αποζημίωση κάθε εργάτη.

Υποδείξεις: Να χρησιμοποιηθεί η εντολή while. Τα αποτελέσματα να δοθούν με τη μορφή:

πλήθος εργατών: — -
 υποχρεωτικές ώρες εργασίας: —-
 ωριαία αποζημίωση : —-
 πρόσθετη υπερωριακή αποζημ: —-

αποδοχές εργατών:— -
 Η εκτύπωση να γίνει με τη μορφή:

α/α	ώρες εργ.	αποζημ.	ώρες υπερ.	υπερ. αποζ.	σύνολο
1					
2					
...		

Να δοθεί δυνατότητα να σταματήσει η επανάληψη, σε περίπτωση που θα δοθεί αριθμός αρνητικός για ώρες εργασίας ενός εργάτη.

Λύση

```

program endeka_3;
var
plithos_ergatwn, ypoxr_wres, yperwries, kan_amoivi, yper_amoivi,
wres_ergasias, apozimiwsi, yper_apozimiwsi, synolo, i: integer;
begin
  i:=1;
  wres_ergasias:=0;
  writeln ('Πλήθος εργατών;');
  readln (plithos_ergatwn);
  writeln ('Υποχρεωτικές ώρες;');
  readln(ypoxr_wres);
  writeln ('Ωριαία αμοιβή;');
  readln ( kan_amoivi);
  writeln ('Υπερωριακή αμοιβή;');
  readln(yper_amoivi);

```

```

while (i<= plithos_ergatwn) and (wres_ergasias>=0) do
  begin
    {Αρχικοποίηση μεταβλητών}
    apozimiwsi:=0;
    yper_apozimiwsi:=0;
    yperwries:=0;
    yper_apozimiwsi:=0;
    writeln ('Εργαζόμενος ',i,' Ώρες εργασίας');
    readln( wres_ergasias);
    if wres_ergasias >= 0
      then
        begin
          if wres_ergasias <= ypoxr_wres
            then
              apozimiwsi:= wres_ergasias*kan_amoivi
            else
              begin
                apozimiwsi:= ypoxr_wres*kan_amoivi;
                yperwries:=wres_ergasias-ypoxr_wres;
                yper_apozimiwsi:=(yperwries)*yper_amoivi;
              end;
            synolo:= apozimiwsi+yper_apozimiwsi;
            writeln(i,' ',wres_ergasias,'
',apozimiwsi,' ',yperwries,' ',yper_apozimiwsi,' ',synolo);
          end;
          i:=i+1;
        end
      end.

```

4. Να τροποποιηθεί η προηγούμενη άσκηση, ώστε το πρόγραμμα να λειτουργεί και με την εντολή repeat.

Λύση

Με τη χρήση της επανάληψης repeat-until, το πρόγραμμα γίνεται:

```

program endeka_4;
var
plithos_ergatwn, ypoxr_wres, yperwries, kan_amoivi, yper_amoivi, wres_ergasias, apozimiwsi, yper_apozimiwsi, synolo, i: integer;

begin
  i:=0;
  wres_ergasias:=0;
  writeln ('Πλήθος εργατών');
  readln (plithos_ergatwn);
  writeln ('Υποχρεωτικές ώρες');
  readln(ypoxr_wres);
  writeln ('Ωριαία αμοιβή');
  readln ( kan_amoivi);
  writeln ('Υπερωριακή αμοιβή');
  readln(yper_amoivi);

```

```

repeat
  {Αρχικοποίηση μεταβλητών}
  apozimiwsi:=0;
  yper_apozimiwsi:=0;
  yperwries:=0;
  yper_apozimiwsi:=0;
  i:=i+1;
  writeln ('Εργαζόμενος ',i,' Ωρες εργασίας');
  readln( wres_ergasias);
  if wres_ergasias>=0
    then
      begin
        if wres_ergasias <= ypoxr_wres
          then
            apozimiwsi:= wres_ergasias*kan_amoivi
          else
            begin
              apozimiwsi:= ypoxr_wres*kan_amoivi;
              yperwries:=wres_ergasias-ypoxr_wres;
              yper_apozimiwsi:=(yperwries)*yper_amoivi;
            end;
            synolo:= apozimiwsi+yper_apozimiwsi;
            writeln(i,' ',wres_ergasias,' ',apozimiwsi,'
',yperwries,' ',yper_apozimiwsi,' ',synolo);
          end;

        until (i> plithos_ergatwn) or (wres_ergasias<0)
      end.

```

5. Να γράψετε πρόγραμμα που να διαβάζει το πλήθος των μαθητών και το βαθμό του καθενός και να τυπώνει ένα χαρακτηρισμό ανάλογα με το βαθμό του.

(9 < βαθμός <= 12 μέτρια
 12 < βαθμός <=15 καλά
 15 < βαθμός <= 18 πολύ καλά
 18 < βαθμός <= 20 άριστα)

Να χρησιμοποιήσετε την εντολή repeat και δυαδικούς²⁶ τελεστές (and...).

Στο τέλος να τυπωθεί το πλήθος κάθε κατηγορίας, δηλ. άριστα, πολύ καλά κλπ. καθώς και το σύνολο όλων των μαθητών.

Λύση²⁷

```

program endeka_5;

```

```

var

```

{Οι μεταβλητές metria,kala,polykala,arista κρατούν τον αριθμό των μαθητών με τους αντίστοιχους βαθμούς. Στην περίπτωση που ο βαθμός είναι κάτω από εννέα η αντίστοιχη εντολή writeln παραμένει κενή γιατί δε ζητείται να γραφτεί κάτι από την εκφώνηση της άσκησης }

```

  plithos_mathitwn,vathmos,metria,kala,polykala,arista,i:integer;

```

²⁶ λογικούς

²⁷ Ο κώδικας είναι σε μικρότερο μέγεθος γραμματοσειράς επειδή χρησιμοποιούνται εσοχές και κάποιες εντολές εκτείνονται σε αρκετά μεγάλο μήκος.

```

begin
    metria:=0;
    kala:=0;
    polykala:=0;
    arista:=0;

    writeln ('Πλήθος μαθητών;');
    readln (plithos_mathitwn);
    i:=1;
    repeat
        repeat
            writeln ('Μαθητής ',i,' Βαθμός;');
            readln( vathmos);
        until (vathmos>0) and (vathmos<=20);
        if vathmos <=9
            then
                writeln()
            else
                if (vathmos > 9 ) and ( vathmos <=12)
                    then
                        begin
                            writeln('Μέτρια');
                            metria:=metria+1;
                        end
                    else
                        if vathmos <=15
                            then
                                begin
                                    writeln('Καλά');
                                    kala:=kala+1;
                                end
                            else
                                if vathmos <=18
                                    then
                                        begin
                                            writeln('Πολύ Καλά');
                                            polykala:=polykala+1;
                                        end
                                    else
                                        if vathmos <=20
                                            then
                                                begin
                                                    writeln('Άριστα');
                                                    arista:=arista+1;
                                                end;
                                        i:=i+1;
                                    until i> plithos_mathitwn;
                                writeln ('Μέτρια= ',metria, 'Καλά = ', kala,'Πολύ καλά=
',polykala,'Άριστα=',arista)
                            end.

```

6. Να γράψετε πρόγραμμα που να διαβάζει τις 4 πλευρές και μία γωνία ενός τετραπλεύρου και να υπολογίζει το είδος του (τετράγωνο, ρόμβο, ορθογώνιο,

παραλληλόγραμμο, τυχαίο). Να χρησιμοποιήσετε την εντολή `repeat` και δυαδικούς²⁸ τελεστές.

Λύση

Χρήσιμα θεωρήματα από τη Γεωμετρία:

«Ένα τετράπλευρο είναι παραλληλόγραμμο, όταν έχει τις απέναντι πλευρές του ίσες».

«Ένα παραλληλόγραμμο, το οποίο έχει μία γωνία ορθή, είναι ορθογώνιο».

«Ένα παραλληλόγραμμο, το οποίο έχει δύο διαδοχικές πλευρές ίσες και μία γωνία ορθή είναι τετράγωνο».

«Ένα παραλληλόγραμμο, το οποίο έχει δύο διαδοχικές πλευρές ίσες και μία γωνία $\neq 90^\circ$ είναι ρόμβος ».

```
program endeka_6;
var p1,p2,p3,p4,angle,i:integer;
begin
  i:=1;
  repeat
    writeln ('Δώσε την ', i, ' πλευρά:');
    case i of
      1:readln(p1);
      2:readln(p2);
      3:readln(p3);
      4:readln(p4);
    end;
    i:=i+1;
  until i>4;

  writeln ('Δώστε μια γωνία του τετραπλεύρου');
  readln (angle);
  if (p1=p3) and (p2=p4)
  then
    if angle=90
    then
      if p1=p2
      then
        writeln('Τετράγωνο')
      else
        writeln('Ορθογώνιο')
      else
        if p1=p2
        then
          if (angle=60) or (angle=120)
          then
            writeln('Ρόμβος')
          else
            writeln('Αφού οι πλευρές του
είναι ίσες, είναι ρόμβος. Δεν δώσατε όμως σωστή γωνία!')
          else
            writeln('Παραλληλόγραμμο')
        else
          writeln('Το τετράπλευρο έχει ακαθόριστο σχήμα');
        end.
end.
```

²⁸ λογικούς

Το σκεπτικό της λύσης:

Εξετάζουμε τις απέναντι πλευρές του τετραπλεύρου (p_1 με p_3 και p_2 με p_4).

Αν είναι ίσες, το τετράπλευρο είναι τουλάχιστον παραλληλόγραμμο αλλιώς το σχήμα του είναι ακαθόριστο.

Στην περίπτωση που οι απέναντι πλευρές είναι ίσες εξετάζουμε τη γωνία, η οποία μας δίνεται.

Αν είναι ορθή, τότε αν οι διαδοχικές πλευρές του είναι ίσες ($p_1=p_2$) τότε είναι τετράγωνο αλλιώς είναι ορθογώνιο.

Αν η γωνία δεν είναι ορθή, τότε εξετάζουμε και πάλι τις διαδοχικές πλευρές του ($p_1=p_2$);

Αν ισχύει η ισότητα, τότε το τετράπλευρο είναι ρόμβος. Προσοχή όμως! Η γωνία των δεδομένων πρέπει να είναι 60° ή 120° αλλιώς πρόκειται για λάθος δεδομένα.

Αν $p_1 \neq p_2$ τότε το τετράπλευρο είναι απλώς παραλληλόγραμμο.

Δραστηριότητες

Να κωδικοποιήσετε σε γλώσσα Pascal τους αλγορίθμους των ασκήσεων 2, 3, 4 του ΚΕΦΑΛΑΙΟΥ 5 ΣΤΟΙΧΕΙΑ ΔΟΜΗΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Άσκηση 2 ΚΕΦ.5

Η άσκηση 2 του ΚΕΦ. 5 είναι παρόμοια με την άσκηση 5 του ΚΕΦ. 4 με τη διαφορά α) ότι στην άσκηση 2 ΚΕΦ. 5 ζητείται να εισάγονται αυτοί που συμμετείχαν επιτυχώς από κάθε τάξη ενώ στην άσκηση 5 του ΚΕΦ. 4 αυτοί που απλώς συμμετείχαν. (Χρησιμοποιούμε τις ίδιες μεταβλητές οι οποίες έχουν άλλη σημασία) β) Στην άσκηση 2 του ΚΕΦ. 5 ζητείται επιπλέον ο συνολικός αριθμός των επιτυχόντων. Θα χρησιμοποιήσουμε μία επιπλέον μεταβλητή ακέραίου τύπου π.χ. τη `synolo` η οποία στο τέλος θα κρατήσει το άθροισμα των μεταβλητών $g_1+g_2+g_3+l_1+l_2+l_3+e_1+e_2+e_3$.

Άσκηση 3 ΚΕΦ.5

```
program drastiriotita11_3;
var onoma:string;
    misthos,kratiseis,katharo:real;
begin
    {Αρχικοποιήσεις}
    onoma:='';misthos:=0;kratiseis:=0;katharo:=0;
    write('Όνοματεπώνυμο= ');
    readln(onoma);
    repeat
        write('Μισθός= ');
        readln(misthos);
    until misthos>0;
    if misthos<=440
    then
        kratiseis:=0.01*misthos
    else
        if misthos<=735
        then
            kratiseis:=4.4+0.015*(misthos-440)
```



```

        else
            if misthos<=880
                then
                    kratiseis:=8.825+0.02*(misthos-735)
                else
                    if misthos<=1135
                        then
                            kratiseis:=11.725+0.025*(misthos-880)
                        else
                            kratiseis:=18.1+0.035*(misthos-1135);
                    writeln('Όνοματεπώνυμο ', onoma);
                    writeln('Κρατήσεις= ', kratiseis:10:3)
            end.

```

Άσκηση 4 ΚΕΦ.5

```

program drastiriotital1_4;
var
plithos_m,plithos_IXe,plithos_DXe,plithos_IXf,plithos_DXf,i,
mikos:integer;
    synolo:real;

begin
    plithos_m:=0;plithos_IXe:=0;
    plithos_DXe:=0;plithos_IXf:=0;plithos_DXf:=0;synolo:=0;
    write('πλήθος μηχανών= ');
    readln( plithos_m);
    write('πλήθος IXε= ');
    readln( plithos_IXe);
    write('πλήθος ΔXε= ');
    readln( plithos_DXe);
    write('πλήθος IXφ= ');
    readln( plithos_IXf);
    write('πλήθος ΔXφ= ');
    readln( plithos_DXf);
    {Υποθέτουμε ότι για κάθε κατηγορία υπάρχει τουλάχιστον 1
όχημα}

    for i:=1 to plithos_m do
        synolo:=synolo+1.5;

    for i:=1 to plithos_IXe do
        begin
            repeat
                writeln('Δώσε το μήκος του IX επιβατηγού');
                readln(mikos);
            until (mikos>0) and (mikos<=5);

            if (mikos<=2)
                then

```

```

        synolo:=synolo+3
    else
        if (mikos<=3)
            then
                synolo:=synolo+6
            else
                if (mikos<=5)
                    then
                        synolo:=synolo+9;
end;

for i:=1 to plithos_DXe do
begin
repeat
    writeln('Δώσε το μήκος του ΔΧ επιβατηγού');
    readln(mikos);
until (mikos>0) and (mikos<=5);

if (mikos<=2)
then
    synolo:=synolo+6
else
    if (mikos<=3)
        then
            synolo:=synolo+9
        else
            if (mikos<=5)
                then
                    synolo:=synolo+12;
end;

for i:=1 to plithos_IXf do
begin
repeat
    writeln('Δώσε το μήκος του ΙΧ φορτηγό');
    readln(mikos);
until (mikos>0) and (mikos<=10);

if (mikos<=2)
then
    synolo:=synolo+9
else
    if (mikos<=3)
        then
            synolo:=synolo+12
        else
            if (mikos<=5)
                then
                    synolo:=synolo+15
            else

```

```

                                synolo:=synolo+18;
                                end;
for i:=1 to plithos_DXf do
begin
repeat
    writeln('Δώσε το μήκος του ΔΧ φορτηγό');
    readln(mikos);
until (mikos>0);

    if (mikos<=2)
    then
        synolo:=synolo+12
    else
        if (mikos<=3)
        then
            synolo:=synolo+15
        else
            if (mikos<=5)
            then
                synolo:=synolo+18
            else
                if (mikos<=10)
                then
                    synolo:=synolo+21
                else
                    synolo:=synolo+24;

                                end;
writeln('Το συνολικό ποσό χρέωσης είναι = ', synolo:10:2);

end.

```

Κεφάλαιο 12

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

Ερωτήσεις

1. Να συγκρίνετε τη μορφή της διαδικασίας και του προγράμματος.

Η πρώτη γραμμή (επικεφαλίδα) ενός προγράμματος **ονομάζει** το πρόγραμμα και **προσδιορίζει τα αρχεία**, από όπου παίρνουμε τα **δεδομένα**, και τα **αρχεία**, όπου γράφονται τα **αποτελέσματα**.

Η πρώτη γραμμή μιας διαδικασίας **ονομάζει** τη διαδικασία και **προσδιορίζει τις μεταβλητές** (παραμέτρους) που χρησιμοποιούνται για **είσοδο των δεδομένων** και **έξοδο των αποτελεσμάτων**.

2. Τί είναι και σε τι χρησιμεύει μια διαδικασία;

Μια διαδικασία είναι ένα **υποπρόγραμμα** και έχει την **ίδια μορφή** με ένα πρόγραμμα, με τη διαφορά ότι αντί της λέξης **program** στην πρώτη γραμμή, έχουμε τη λέξη **procedure** και μετά το τελευταίο end έχουμε «;» αντί «.».

3. Ποιες είναι οι τυπικές και ποιες οι πραγματικές παράμετροι μιας διαδικασίας;

Οι **τυπικές παράμετροι**(formal) είναι οι μεταβλητές οι οποίες τοποθετούνται στην **αρχή μιας διαδικασίας** για την **είσοδο** των δεδομένων και την **έξοδο** των αποτελεσμάτων. Οι **πραγματικές** (actual) παράμετροι είναι εκείνες που **αντικαθιστούν** τις τυπικές κατά την κλήση της διαδικασίας. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που έχει την ίδια σειρά.

4. Πώς εκτελείται μία διαδικασία;

Για να εκτελεστεί μια διαδικασία πρέπει να γράψουμε το **όνομά** της, ακολουθούμενο από τη **λίστα των πραγματικών παραμέτρων** (που τυχόν περιέχει). Οι παράμετροι τοποθετούνται εντός παρενθέσεως και αν είναι περισσότερες από μία, χωρίζονται με κόμμα. Κατά την κλήση της διαδικασίας (μέσα από το κυρίως πρόγραμμα ή μέσα από μια άλλη διαδικασία), ο **έλεγχος** μεταφέρεται στην είσοδό της και στη συνέχεια εκτελούνται οι εντολές που βρίσκονται μέσα σε αυτή. Όταν τελειώσουν, ο έλεγχος ξαναγυρίζει στο σημείο από όπου έγινε η κλήση της διαδικασίας και στην εντολή η οποία βρίσκεται στη συνέχεια.

5. Ποιες είναι οι παράμετροι τιμών και ποιες οι παράμετροι μεταβλητών σε μία διαδικασία; Να δώσετε παραδείγματα από τα προγράμματα διαδικασιών του βιβλίου σας.

Οι **παράμετροι τιμών μεταφέρουν δεδομένα** μέσα σε μια διαδικασία (λέγονται και παράμετροι εισόδου). Μια παράμετρος τιμής ορίζει τοπικά το όνομα μια περιοχής τυπικής παραμέτρου. Η περιοχή αυτή παίρνει τιμή από την πραγματική παράμετρο, όταν καλείται η διαδικασία. Οι πραγματικές παράμετροι αντιστοιχούν **μία προς μία** με τις τυπικές παραμέτρους.

Σε πολλές περιπτώσεις χρειάζεται να εφοδιάσουμε τη διαδικασία και με παράμετρο η οποία **να μεταφέρει μια τιμή στη διαδικασία** η δε διαδικασία να αλλάζει την τιμή της παραμέτρου και η **αλλαγή αυτή να μεταφέρεται στην πραγματική παράμετρο**. Τον ρόλο αυτό παίζει η **παράμετρος μεταβλητής**. Στην περίπτωση αυτή η τυπική παράμετρος παίρνει τιμή από την πραγματική παράμετρο αλλά οποιαδήποτε αλλαγή της τιμής της τυπικής παραμέτρου έχει ως αποτέλεσμα αλλαγή και της τιμής της πραγματικής παραμέτρου. Οι παράμετροι μεταβλητών μεταφέρουν τιμές **έξω από τη διαδικασία** και για το

λόγο αυτό ονομάζονται **παράμετροι εξόδου**. Παράδειγμα διαδικασίας με παραμέτρους τιμών και παραμέτρους μεταβλητών είναι η διαδικασία *γροlogismos* στη σελ. 113.

6. Τί είναι συνάρτηση οριζόμενη από το χρήστη και ποια είναι η διαφορά της από τη διαδικασία;

Η συνάρτηση είναι ένα **υποπρόγραμμα** η οποία δίνει ως αποτέλεσμα (**επιστρέφει**) μια **τιμή** η οποία μπορεί να έχει ένα οποιοδήποτε τύπο από τους επιτρεπτούς της Pascal. Αυτή είναι και η βασική της διαφορά από τη διαδικασία η οποία δεν επιστρέφει τιμή. Η μορφή της συνάρτησης είναι παρόμοια με αυτή της διαδικασίας με τη διαφορά ότι αντί της λέξης *procedure* έχουμε τη λέξη **function**. Η πρώτη γραμμή της συνάρτησης καθορίζει το όνομά της και ορίζει μεταβλητές (τυπικές παραμέτρους) για την εισαγωγή των δεδομένων. Στην πρώτη γραμμή επίσης δίνεται και ο τύπος της συνάρτησης (*integer*, *real*, κτλ).

7. Πώς καλείται μία συνάρτηση;

Κατ' αρχήν για μπορέσει να κληθεί μια συνάρτηση πρέπει πρώτα να **δηλωθεί** στο τμήμα δηλώσεων του προγράμματος. Κάθε συνάρτηση **καλείται με το όνομά της** και αν υπάρχουν παράμετροι, οι πραγματικές παράμετροι αντικαθιστούν τις τυπικές παραμέτρους. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που έχει την ίδια σειρά.

8. Να δώσετε δικά σας παραδείγματα κλήσεων συναρτήσεων από τα προγράμματα συναρτήσεων του βιβλίου σας ορίζοντας δικά σας δεδομένα και να προσδιορίσετε τα αποτελέσματα.

Αν ορίσουμε μια συνάρτηση για το μέσο όρο στο τμήμα δηλώσεων όπως φαίνεται παρακάτω:

```
function Avg(X, Y, Z : real) : real;
begin
    Avg=(X+Y+Z)/3;
end;
```

στη συνέχεια μπορούμε να την καλέσουμε μέσα από μια εντολή της γλώσσας, π.χ.

```
writeln(Avg(56, 43.5, 45*0.5):10:2);
```

9. Να γράψετε μια συνάρτηση για το ημιάθροισμα δύο αριθμών.

```
function half_sum(n1,n2:real):real;
begin
    half_sum:=(n1+n2)/2;
end;
```

10. Να γράψετε μια συνάρτηση για το γινόμενο δύο αριθμών.

```
function product(n1,n2:real):real;
begin
    product:=n1*n2;
end;
```

Ασκήσεις

1. Να γράψετε πρόγραμμα που να υπολογίζει τον τελικό βαθμό ενός μαθητή σ' ένα μάθημα από την προφορική βαθμολογία των τριών τριμήνων και από το βαθμό των γραπτών.

Υποδείξεις

α. Ο υπολογισμός του βαθμού γίνεται με βάση τον τύπο:

$$\text{Τελικός Βαθμός} = \frac{A. \text{τριμ} + B. \text{τριμ} + G. \text{τριμ} + 2 * \text{Γραπτός Βαθμός}}{5}$$

β. Να χρησιμοποιήσετε μία διαδικασία για την εισαγωγή δεδομένων, μία για την εύρεση του μέσου όρου και μια για την εκτύπωση των αποτελεσμάτων.

γ. Να χρησιμοποιηθούν παράμετροι τιμών καθώς και παράμετροι μεταβλητών (var) στις διαδικασίες.

Λύση

```
program dwdeka_1;
```

```
var v1,v2,v3,v4:integer;
```

```
mo:real; {η mo χρησιμοποιείται ως καθολική μεταβλητή}
```

```
{Οι b1,b2,b3,b4 είναι παράμετροι μεταβλητών. Θα μπορούσαν να χρησιμοποιηθούν τα ίδια γράμματα και στις επόμενες διαδικασίες αλλά για διδακτικούς λόγους χρησιμοποιούνται διαφορετικά}
```

```
procedure eisagwgi(var b1,b2,b3,b4:integer);
```

```
begin
```

```
repeat
```

```
writeln('Δώστε το βαθμό Α'τριμ., Β'τριμ., Γ'τριμ.,  
Γραπτών με κενά ανάμεσα');
```

```
readln(b1,b2,b3,b4);
```

```
{έλεγχος για έγκυρους βαθμούς}
```

```
until ((b1>0) and (b1<=20)) and ((b2>0) and  
(b2<=20))and ((b3>0) and (b3<=20))and((b4>0) and  
(b4<=20));
```

```
end;
```

```
procedure ypologismos(var c1,c2,c3,c4:integer);
```

```
begin
```

```
mo:=(c1+c2+c3+2*c4)/5
```

```
end;
```

```
{Οι d1,d2,d3,d4 είναι παράμετροι τιμών}
```

```
procedure apotelesmata(d1,d2,d3,d4:integer;m:real);
```

```
begin
```

```
writeln ('Βαθμοί ');
```

```
writeln(d1:10, d2:10,d3:10,d4:10);
```

```
writeln ('Μέσος Όρος');
```

```

        writeln(m:10:2);
end;

begin
    eisagwgi (v1,v2,v3,v4);
    ypologismos(v1,v2,v3,v4);
    apotelesmata(v1,v2,v3,v4,mo)
end.

```

2. Να γράψετε πρόγραμμα το οποίο να διαβάζει δύο αριθμούς και το είδος της πράξης που θα εκτελέσει (πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση). Να χρησιμοποιήσετε διαδικασίες (χωρίς παραμέτρους) για την εισαγωγή των δεδομένων, την επιλογή της πράξης, τον υπολογισμό και την εκτύπωση του αποτελέσματος. Να χρησιμοποιήσετε ένα μηχανισμό ελέγχου, ώστε οι διαδικασίες να επαναλαμβάνονται όσες φορές θέλουμε.

Λύση

```

program dwdeka_2;
var n1,n2, apotelesma:real;
    ch,reply:char;
    flag:boolean;
procedure eisagwgi;
begin
    writeln('Δώστε δύο αριθμούς χωρισμένους με κενά ');
    readln(n1, n2);
    writeln('Δώστε το σύμβολο της πράξης (+, -, *, /)');
    readln(ch);
end;
procedure ypologismos;
begin
    case ch of
        '+' : apotelesma:=n1+n2;
        '-' : apotelesma:=n1-n2;
        '*' : apotelesma:=n1*n2;
        '/' : begin
                if n2<>0
                    then
                        apotelesma:=n1/n2
                    else
                        flag:=false;
                end;
            else
                flag:=false;
            end;
end;
end;
procedure ektypwsi;
begin
    if flag=true
        then
            writeln (n1:10:2,ch:2,n2:10:2,'=',apotelesma:10:2)
        else
            writeln ('Αδύνατη πράξη');
end;

```

```

end;
begin
  repeat
    flag:=true;
    eisagwgi;
    ypologismos;
    ektypwsi;
    writeln('-----');
-----');
    writeln ('Πατήστε ελληνικό "Ο" ή "ο" για να σταματή-
σετε, οτιδήποτε άλλο για να συνεχίσετε');
    readln(reply);
    until (reply='Ο') or (reply='ο');
end.

```

3. Να γραφεί πρόγραμμα που :

- α. Διαβάζει το εμβαδόν τεσσάρων διαμερισμάτων, του καθενός χωριστά και τη συνολική δαπάνη πετρελαίου,
- β. Υπολογίζει το ποσοστό συμμετοχής (= εμβαδόν διαμερίσματος /συνολικό εμβαδόν) και την αντίστοιχη χρέωση για κάθε διαμέρισμα,
- γ. Τυπώνει τα αποτελέσματα σύμφωνα με το υπόδειγμα που ακολουθεί:

Υποδείξεις

- α. Να γραφούν διαδικασίες για την εισαγωγή των δεδομένων (εμβαδόν, κόστος πετρελαίου), για τον υπολογισμό και την εκτύπωση των δεδομένων
- β. Να τυπωθούν τα αποτελέσματα σύμφωνα με το παρακάτω υπόδειγμα:

αξία πετρελαίου διαμέρισμα	εμβαδόν	300 € ποσοστά	χρέωση πετρελαίου
Διαμέρισμα1	150	0.349	104.65
Διαμέρισμα2	100	0.233	69.77
Διαμέρισμα3	80	0.186	55.81
Διαμέρισμα4	100	0.233	69.77
	430	1.000	300.00

Λύση

```

program dwdeka_3;
{em1 έως em4 τα εμβαδά, syn_em το συνολικό εμβαδόν, ps1 έως
ps5 τα ποσοστά συμμετοχής, kat_pet η κατανάλωση πετρελαίου}
var em1,em2, em3, em4,
syn_em,ps1,ps2,ps3,ps4,x1,x2,x3,x4,kat_pet:real;
i:integer;

procedure eisagwgi(var e1,e2,e3,e4,kp:real);
begin
  writeln('Δώστε το εμβαδόν του 1ου διαμερίσματος');
  readln(e1);
  writeln('Δώστε το εμβαδόν του 2ου διαμερίσματος');
  readln(e2);

```



```

        writeln('Δώστε το εμβαδόν του 3ου διαμερίσματος');
        readln(e3);
        writeln('Δώστε το εμβαδόν του 4ου διαμερίσματος');
        readln(e4);
        writeln('Δώστε τη συνολική δαπάνη του πετρελαίου');
        readln(kp);
    end;

procedure ypologismos(var e1,e2,e3,e4,kp:real);
begin
    syn_em:=e1+e2+e3+e4;
    ps1:=e1/syn_em;
    x1:=ps1*kp;
    ps2:=e2/syn_em;
    x2:=ps2*kp;
    ps3:=e3/syn_em;
    x3:=ps3*kp;
    ps4:=e4/syn_em;
    x4:=ps4*kp;
end;

procedure ektypwsi;
begin
    writeln ('Αξία πετρελαίου           ',kat_pet,'€');
    for i:=1 to 60 do
        write('-');
    writeln();
    writeln ('διαμέρισμα':15, 'εμβαδόν':15, 'ποσοστά':15,
'χρέωση':15);
    writeln('διαμέρισμα1':15, em1:15:1,ps1:15:3, x1:15:2);
    writeln('διαμέρισμα2':15, em2:15:1,ps2:15:3, x2:15:2);
    writeln('διαμέρισμα3':15, em3:15:1,ps3:15:3, x3:15:2);
    writeln('διαμέρισμα4':15, em4:15:1,ps4:15:3, x4:15:2);
    for i:=1 to 60 do
        write('-');
    writeln();

writeln(syn_em:30:1,(ps1+ps2+ps3+ps4):15:1,(x1+x2+x3+x4):15:2);
    end;

begin
    {Αρχικοποίηση μεταβλητών}

em1:=0;em2:=0;em3:=0;em4:=0;syn_em:=0;ps1:=0;ps2:=0;ps3:=0;ps4
:=0;x1:=0;x2:=0;x3:=0;x4:=0;kat_pet:=0;
    eisagwgi(em1,em2,em3,em4,kat_pet);
    ypologismos(em1,em2,em3,em4,kat_pet);
    ektypwsi;
end.

```

4. Για τον υπολογισμό φόρου έστω ότι ισχύει ο παρακάτω πίνακας:

Μισθός			Ποσοστό φόρου
0€	έως	300€	12%
301€	>>	600€	18%
601€	>>	900€	25%
901€	>>	1200€	35%
1201€	>>	1500€	50%
1501€	>>	και άνω	60%

Να γραφεί πρόγραμμα το οποίο να χρησιμοποιεί τις διαδικασίες:

α. εισαγωγής δεδομένων (ονοματεπώνυμο, αποδοχές),

β. υπολογισμού.(φόρου, πληρωτέου ποσού),

γ. εκτύπωσης των αποτελεσμάτων σύμφωνα με το υπόδειγμα:

ΕΠΩΝΥΜΟ ΔΗΜΗΤΡΙΑΔΗΣ	ΟΝΟΜΑ ΝΙΚΟΛΑΟΣ	ΑΠΟΔΟΧΕΣ 750	ΦΟΡΟΣ 127.5	ΚΑΘ. ΠΛΗΡ. 622.5
------------------------	-------------------	-----------------	----------------	---------------------

Υποδείξεις:

α. Ο φόρος να υπολογιστεί κλιμακωτά, πχ. για το ποσό των 750€ έχουμε: $300€ \times 12\% + 300€ \times 18\% + 150€ \times 25\% = 127.5€$

β. Οι διαδικασίες επαναλαμβάνονται, μέχρις ότου δοθεί τέλος δεδομένων (αποδοχές=0).

Λύση

```

program dwdeka_4;
var apodoxes, foros, katharo: real;
    onoma, epwnymo: String;

procedure eisagwgi(var epw, ono: String; var ap: real);
begin
    writeln('Δώστε το Επώνυμο');
    readln(epw);
    writeln('Δώστε το Όνομα');
    readln(ono);
    writeln('Δώστε τις αποδοχές');
    readln(ap);
end;

procedure ypologismos(var ap: real);
begin
    if ap <= 300
    then
        foros := ap * 0.12
    else
        if ap <= 600
        then
            foros := 300 * 0.12 + (ap - 300) * 0.18
        else
            if ap <= 900

```

```

        then
            foros:=300*0.12+300*0.18+(ap-600)*0.25
        else
            if ap<=1200
                then
                    foros:=300*0.12+300*0.18+300*0.25+(ap-900)*0.35
                else
                    if ap<=1500
                        then
                            foros:=300*0.12+300*0.18+300*0.25+300*0.35+(ap-1200)*0.50
                        else
                            if ap>1500
                                then
                                    foros:=300*0.12+300*0.18+300*0.25+300*0.35+300*0.5+(ap-
                                    1500)*0.60;
                                    katharo:=ap-foros;
                                end;
                            end;
                    end;
            end;

procedure ektypwsi;
begin
    writeln    ('Επώνυμο':15,      'Όνομα':15,      'Αποδοχές':10,
    'Φόρος':10, 'Καθαρό':10);
    writeln
    (epwnymo:15,onoma:15,apodoxes:10:2,foros:10:2,katharo:10:2)

end;

begin
    epwnymo:='';
    onoma:='';
    apodoxes:=0;
    foros:=0;
    katharo:=0;
    repeat
        eisagwgi(epwnymo,onoma,apodoxes);
        ypologismos(apodoxes);
        ektypwsi;
    Until apodoxes=0;
end.

```

5. Να γράψετε πρόγραμμα που να διαβάζει δυο ακέραιους και θετικούς αριθμούς και να καλεί μία συνάρτηση για τον υπολογισμό του μέγιστου κοινού διαιρέτη²⁹. Η εργασία επαναλαμβάνεται όσο οι αριθμοί είναι και οι δύο θετικοί. Αν και οι δύο αριθμοί είναι μηδέν, η εργασία σταματάει και τυπώνεται κατάλληλο μήνυμα.

Υποδείξεις:

α. Να χρησιμοποιηθεί η εντολή `while` και παράμετροι για τη συνάρτηση, β. Τα αποτελέσματα παρουσιάζονται με τη μορφή:

²⁹ τους

οι αριθμοί και έχουν

μέγιστο κοινό διαιρέτη

ή τέλος εργασίας

Λύση

```
program dwdeka_5;
var ar1,ar2,apotelesma:integer;

{αλγόριθμος του Ευκλείδη για το ΜΚΔ}
function mkd(a:integer;b:integer):integer;
begin
  while (a>0) and (b>0) do
    if a>=b
      then
        a:= a mod b
      else
        b:=b mod a;
    mkd:=a+b;
  end;

{κυρίως πρόγραμμα}
begin
writeln('Δώστε δύο θετικούς ακέραιους χωρισμένους με κενό');
readln (ar1,ar2);
while (ar1<>0) and (ar2<>0) do
  begin
    apotelesma:=mkd(ar1,ar2);
    writeln('Οι αριθμοί ',ar1, ' και ', ar2, ' έχουν ΜΚΔ το ',
apotelesma);
    writeln('Δώστε δύο θετικούς ακέραιους χωρισμένους με κενό');
    readln (ar1,ar2);
  end
end.
```

6. Να γράψετε πρόγραμμα που να διαβάζει έναν ακέραιο αριθμό και να καλεί μία συνάρτηση για τον υπολογισμό της τρίτης δύναμης του αριθμού αυτού. Η εργασία συνεχίζεται, αν ο αριθμός είναι διάφορος του μηδενός, αλλιώς η εργασία σταματάει και δίνεται και κατάλληλο μήνυμα.

Υποδείξεις:

α.: Να χρησιμοποιηθεί η εντολή repeat

β. Τα αποτελέσματα παρουσιάζονται με τη μορφή:

ο κύβος του αριθμού *****.**** είναι *********

ή ο αριθμός είναι μηδέν

τέλος εργασίας.

Λύση

```
program dwdeka_6;
var ar1,apotelesma:integer;

function dynami(a:integer):integer;
begin
  dynami:=a*sqr(a);
end;
```

```

begin
  repeat
    writeln('Δώστε έναν ακέραιο αριθμό');
    readln (ar1);
    apotelesma:=dynami(ar1);
    if ar1 <>0
      then
        writeln('Η τρίτη δύναμη του αριθμού ',ar1, ' είναι: ', apotelesma)
      else
        writeln('Τέλος του προγράμματος');
    until ar1=0;

  end.

```

7. Να γράψετε πρόγραμμα που χρησιμοποιώντας κατάλληλες συναρτήσεις να υπολογίζει και να τυπώνει το μήκος του κύκλου και το εμβαδόν του κυκλικού δίσκου ακτίνας R. Η διαδικασία να επαναλαμβάνεται για τιμές της ακτίνας από 10cm έως 1.5 m με βήμα 10 cm.

Λύση

```

program dwdeka_7;
var aktina:real;
function mikos(var r:real):real;
begin
  mikos:=2*pi*r;
end;

```

```

function emvadon(var r:real):real;
begin
  emvadon:=pi*sqr(r);
end;

```

{κυρίως πρόγραμμα}

```

begin
  writeln('Έναρξη προγράμματος');
  aktina:=0.0;
  while aktina <= 1.5 do
    begin
      aktina:=aktina+0.1;
      writeln('Για ακτίνα=',aktina:6:2,'--', ' Σπερ. =',
        ',mikos(aktina):6:2,'--', 'Εκυκλ.δίσκου=', emvadon(aktina):6:2);
    end
  end.

```

8. Να γράψετε πρόγραμμα που να διαβάζει το ύψος και τη βάση ενός τριγώνου και να καλεί μία συνάρτηση που να υπολογίζει το εμβαδόν του. Να γίνει εκτύπωση των αποτελεσμάτων σύμφωνα με το υπόδειγμα:

βάση
 ύψος
 Εμβαδόν

Λύση

```

program dwdeka_8;
var

```

```

vasi, ypsos:real;

function emvadon(var v,y:real):real;
begin
    emvadon:=0.5*v*y;
end;
{κυρίως πρόγραμμα}
begin
    writeln('Δώσε τη βάση του τριγώνου');
    readln(vasi);
    writeln('Δώσε το ύψος του τριγώνου');
    readln(ypsos);
    writeln('βάση = ', vasi:10:2);
    writeln('ύψος = ', ypsos:10:2);
    writeln('Εμβαδόν=', emvadon(vasi,ypsos):10:2)
end.

```

9. Να γράψετε πρόγραμμα που να διαβάζει:

α. ένα τόξο (ή γωνία) σε μοίρες, να μετατρέπει την τιμή αυτή σε ακτίνια καλώντας μία συνάρτηση και να τυπώνει στην οθόνη το αποτέλεσμα.

β. ένα τόξο (ή γωνία) σε ακτίνια, να μετατρέπει την τιμή αυτή σε μοίρες καλώντας μία συνάρτηση και να τυπώνει στην οθόνη το αποτέλεσμα. Υπόδειξη:

Ο τύπος μετατροπής είναι:

$$\frac{\text{ακτίνια}}{\pi} = \frac{\text{μοίρες}}{180}$$

όπου $\pi=3.14$

Λύση

```

program dwdeka_9;
var
    angle:real;
    epilogi:char;

function se_aktinia(m:real):real;
begin
    se_aktinia:=m*pi/180;
end;

function se_moires(a:real):real;
begin
    se_moires:=a*180/pi;
end;

begin
    writeln('Μετατροπή');
    writeln('1. Από μοίρες σε ακτίνια');
    writeln('2. Από ακτίνια σε μοίρες');
    writeln('Πατήστε 1 ή 2');
    readln(epilogi);
    case epilogi of
        '1':begin

```

```

        writeln('Δώστε τη γωνία σε μοίρες');
        readln(angle);
        writeln ('Η αντίστοιχη γωνία σε ακτίνια είναι:',
se_aktinia(angle):10:2);
        end;
    '2':begin
        writeln('Δώστε τη γωνία σε ακτίνια');
        readln(angle);
        writeln ('Η αντίστοιχη γωνία σε μοίρες είναι:',
se_moires(angle):10:2);
        end;
    else
        writeln ('Λάθος επιλογή!');
    end
end.

```

10. Να γράψετε πρόγραμμα που να διαβάζει έναν αριθμό n , να καλεί μια συνάρτηση για τον υπολογισμό του $n!$ (όπου $n!=1*2*3*...*n$) και να τυπώνει το αποτέλεσμα στην οθόνη. Η διαδικασία να επαναλαμβάνεται για 10 αριθμούς με τη χρησιμοποίηση της εντολής FOR.

Λύση

```

program dwdeka_10;
var ar,i:integer;
function paragontiko(var n:integer):longint;
var i:integer;
    product:longint;

begin
    {Η μεταβλητή product χρησιμοποιείται για λόγους "κοσμητικούς",
    θα μπορούσε να χρησιμοποιηθεί κατευθείαν το paragontiko}

    if n<=1 then paragontiko:=1
    else
        begin
            product:=1;
            for i:=2 to n do
                product:=product*i;
            paragontiko:=product;
        end;
    end;
    {κύριο πρόγραμμα}
begin
    for i:=1 to 10 do
        begin
            writeln ('Δώστε έναν ακέραιο');
            readln(ar);
            writeln('Το παραγοντικό του είναι:', paragontiko(ar));
        end
    end.

```

11. Ο υπολογισμός της τιμής του ημίτονου ενός τόξου χ (σε ακτίνια) δίνεται από το άθροισμα της σειράς:

$$\eta\mu(\chi) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Να γράψετε πρόγραμμα που να διαβάζει την τιμή του τόξου σε μοίρες και να υπολογίζει το ημίτονο του τόξου αυτού

Υποδείξεις:

α. Η διαδικασία σταματά όταν η διαφορά δυο διαδοχικών υπολογισθέντων τιμών της συνάρτησης είναι μικρότερη από 0.00001.

β. Να χρησιμοποιήσετε τις δυο προηγούμενες συναρτήσεις για τον υπολογισμό του παραγοντικού και της μετατροπής ενός τόξου από μοίρες σε ακτίνια.

Λύση

```
program dwdeka_11;
var i,paragontiko_p,b:integer;
{x--> η τιμή της γωνίας σε μοίρες, athroisma -->κρατά το
άθροισμα για τον υπολογισμό του ημιτόνου,
athroisma_p --> κρατά το παλιό άθροισμα ώστε να υπολογίζεται η
diaf, oros--> ο όρος που προστίθεται κάθε φορά στο άθροισμα}

x,a, athroisma,athroisma_p,diaf,oros:real;

function se_aktinia(m:real):real;
begin
    se_aktinia:=m*pi/180;
end;
{Σε μεγάλες τιμές της γωνίας π.χ. 180, 360 κτλ, βγαίνει
runtime error γιατί η τιμή του παραγοντικού υπερβαίνει τα όρια
του longint. Αυτό μπορεί να αντιμετωπιστεί δηλώνοντας τόσο τη
συνάρτηση paragontiko, όσο και τη μεταβλητή product ως real}

function paragontiko(var n:integer):longint;
var
    i:integer;
    product:longint;
begin
    if n<=1 then paragontiko:=1
    else
        begin
            product:=1;
            for i:=2 to n do
                product:=product*i;
            paragontiko:=product;
        end;
    end;
end;

function dynami(v:real;k:integer):real;
var
    l:integer;
    product:real;
```



```

begin
  if k=0
    then
      dynami:=1
    else
      begin
        product:=1;
        for l:=1 to k do
          product:=product*v;
          dynami:=product;
        end;
      end;
end;

{κυρίως πρόγραμμα}
begin
  writeln('Δώσε την τιμή του τόξου σε μοίρες');
  readln(x);
  a:=-1.0;
  (*μετατροπή των μοιρών σε ακτίνια *)
  x:=se_aktinia(x);
  writeln('Η γωνία σε ακτίνια είναι:', x:10:3);

  (*Αρχικοποίηση*)
  athroisma:= 0;
  athroisma_p:=0;
  i:=0;
  repeat
    paragontiko(i);
    b:=2*i+1;
    oros:=dynami(a,i)*dynami(x,b)/paragontiko(b);
    athroisma_p:=athroisma;
    athroisma:=athroisma+oros;
    i:=i+1;
    diaf:=abs(athroisma-athroisma_p);
  until diaf < 0.00001;
  writeln ('Το ημίτονο της γωνίας είναι=', athroisma:10:9);
end.

```

Κεφάλαιο 13

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

1. Να εξηγήσετε τι είναι τύπος δεδομένων.

Είναι μια **τυπική περιγραφή** των **τιμών** που μπορούν να αποθηκευτούν σε μια **σταθερά** ή **μεταβλητή** καθώς και των **πράξεων** που μπορεί να εκτελεστούν με τις τιμές αυτές.

2. Να δώσετε την περιγραφή του **Ακέραιου (Integer)** τύπου δεδομένων.

Ο Ακέραιος τύπος δεδομένων περιλαμβάνει ένα **σύνολο ακέραιων τιμών** στο διάστημα από $-\text{MaxInt}^{30}$ έως MaxInt , όπου MaxInt^{31} είναι ο μεγαλύτερος ακέραιος που μπορεί να αναπαρασταθεί στον υπολογιστή. Στην πραγματικότητα το σύνολο αυτό είναι ένα **υποσύνολο των ακεραίων** που εξετάζουμε στα Μαθηματικά. Οι **επιτρεπτές πράξεις** ορίζονται με τα σύμβολα $_, /, *, \text{mod}, \text{div}, :=$ καθώς και με τους σχεσιακούς τελεστές ($<, =, >$ κτλ).

3. Να δώσετε την περιγραφή του **Πραγματικού (Real)** τύπου δεδομένων.

Ο Πραγματικός (real) τύπος δεδομένων στην Pascal, περιλαμβάνει ένα **υποσύνολο των πραγματικών αριθμών** των Μαθηματικών. Το εύρος του υποσυνόλου αυτού καθώς και η ακρίβεια των πραγματικών αριθμών σε δεκαδικά ψηφία ποικίλλει ανάλογα με τον τύπο του υπολογιστή και το μεταφραστικό πρόγραμμα. Οι **επιτρεπτές πράξεις** είναι **ίδιες** με αυτές των ακεραίων **εκτός από τις mod και div**.

4. Να δώσετε την περιγραφή του **Λογικού (Boolean)** τύπου δεδομένων.

Ο Λογικός (Boolean) τύπος, είναι ένας τύπος δεδομένων του οποίου οι τιμές προέρχονται από ένα σύνολο που περιέχει **δύο τιμές** την **true** (σωστό) και την **false** (λάθος). **Επιτρεπτές πράξεις** για αυτόν τον τύπο δεδομένων είναι οι **and, or, not, :=** καθώς και οι **σχεσιακοί τελεστές** ($<, =, >$ κτλ).

5. Να δώσετε την περιγραφή του τύπου **δεδομένων χαρακτήρα (Char)**.

Ο τύπος δεδομένων χαρακτήρα περιλαμβάνει το **σύνολο των χαρακτήρων** που διαθέτει ο υπολογιστής ανάλογα με τον τύπο του. Οι **επιτρεπτές πράξεις** για αυτόν τον τύπο δεδομένου γίνονται με τους **σχεσιακούς τελεστές** και το **:=**.

6. Ποιες είναι οι κοινές ιδιότητες των τεσσάρων βασικών τύπων; Πώς ονομάζονται γενικά οι τύποι με αυτές τις ιδιότητες;

Οι τέσσερις βασικοί τύποι δεδομένων έχουν δύο **κοινές ιδιότητες**:

1. Αποτελούνται από **αδιαίρετα ή ατομικά** στοιχεία.
2. Είναι **διατεταγμένοι**³².

Τύποι με αυτές τις ιδιότητες ονομάζονται **μονόμετροι ή βαθμωτοί** (scalar data types).

7. Ποιοι τύποι δεδομένων ονομάζονται τακτικοί; Ποιοι από τους βασικούς τύπους είναι και ποιοι δεν είναι τακτικοί; Να αιτιολογήσετε την απάντησή σας.

Τακτικοί (ordinal) ονομάζονται εκείνοι οι τύποι δεδομένων, όπου όλες οι τιμές τους – εκτός από την πρώτη – έχουν **μία μοναδική προηγούμενη τιμή** και όλες οι τιμές τους – εκτός από την τελευταία – έχουν **μία μοναδική επόμενη τιμή**. Από τους τέσσερις βασι-

³⁰ Σελ. 71 Από $-\text{MaxInt}-1$ έως MaxInt . Εύρος διαστήματος από -32768 έως 32767

³¹ Η τιμή της MaxInt είναι 32767

³² Όταν λέμε «διατεταγμένοι», εννοούμε ότι μόνο μία από τις σχέσεις $<, >, =$ αληθεύει για κάθε ζεύγος τιμών.

κούς τύπους δεδομένων, **τακτικοί είναι** οι τρεις (**integer**, **boolean**³³, **char**) ενώ **δεν είναι τακτικός ο real**. Αυτό, γιατί σε ένα δεδομένο τύπου **real** αν προστεθεί ένα δεκαδικό ψηφίο, ο προηγούμενος και ο επόμενος αλλάζουν. Π.χ. 56.98=56.980, όμως προηγούμενος του πρώτου είναι ο 56.97 και επόμενος ο 56.99 ενώ του δεύτερου προηγούμενος ο 56.979 και επόμενος ο 56.981.

8. Να αναφέρετε τις συναρτήσεις τακτικών τύπων και να εξηγήσετε με παραδείγματα τη λειτουργία τους.

Η Pascal διαθέτει 4 συναρτήσεις για τους τακτικούς τύπους:

- **chr(x):integer):char**
Επιστρέφει τον χαρακτήρα του κώδικα ASCII, ο οποίος αντιστοιχεί στον ακέραιο αριθμό x. Το όρισμα x παίρνει τιμές από 0 έως και 255.
π.χ. `writeln(chr(80))` εκτυπώνει το P.
- **ord(x):integer**
Επιστρέφει τη θέση του x στη διάταξη του τακτικού (ordinal) τύπου. Η ord είναι αντίστροφη της chr.
 - Αν ο x είναι χαρακτήρας, επιστρέφει τον ακέραιο που φανερώνει την θέση του χαρακτήρα στην ακολουθία χαρακτήρων του υπολογιστή.
π.χ. `writeln(ord('P'))` εκτυπώνει 80.
 - Αν ο x είναι ακέραιος επιστρέφει τον ίδιο ακέραιο.
π.χ. `writeln(ord(230))` εκτυπώνει 230.
 - Αν ο x είναι λογικός τύπος, επιστρέφει 0 αν η τιμή είναι false και 1 αν η τιμή είναι true.
π.χ. `writeln(ord(true))` εκτυπώνει 1.
- **pred(x):ordinal type**
Επιστρέφει το προηγούμενο του ορίσματος. Το όρισμα x και η τιμή που επιστρέφει είναι τακτικού (ordinal) τύπου.
π.χ. `writeln(pred(0))` εκτυπώνει - 1.
`writeln(pred('D'))` εκτυπώνει C
- **succ(x):ordinal type**
Επιστρέφει το επόμενο του ορίσματος. Το όρισμα x και η τιμή που επιστρέφει είναι τακτικού τύπου.
Π.χ. `writeln(succ('C'))` επιστρέφει το D που είναι ο επόμενος του C στον κώδικα ASCII.

9. Πώς διακρίνονται οι απλοί από τους σύνθετους τύπους δεδομένων; Να δώσετε παραδείγματα απλών και σύνθετων τύπων δεδομένων.

Στους **απλούς τύπους** δεδομένων οι τιμές είναι **ατομικές** (atomic). Όταν λέμε ότι η τιμή είναι ατομική εννοούμε ότι **δεν αποτελείται από μέρη τα οποία μπορούν να προσπελαστούν ανεξάρτητα**. Παραδείγματα απλών τύπων δεδομένων είναι οι βαθμωτοί τύποι (Ακέραιος, Πραγματικός, Λογικός, Χαρακτήρας). Αντίθετα, στους **σύνθετους τύπους** δεδομένων, μια μεταβλητή μπορεί να δεχθεί ως τιμή μια **ενότητα τιμών**. Παραδείγματα σύνθετων τύπων είναι τα σύνολα, οι εγγραφές, οι πίνακες και τα αρχεία.

10. Πώς διακρίνονται οι σύνθετοι τύποι δεδομένων με δόμηση, από τους σύνθετους τύπους χωρίς δόμηση; Να δώσετε παραδείγματα σύνθετων τύπων με δόμηση και σύνθετων τύπων χωρίς δόμηση.

³³ Στο διμελές σύνολο των τιμών των λογικών μεταβλητών, πρώτο είναι το στοιχείο false (αντιστοιχεί στο 0) και δεύτερο το true (αντιστοιχεί στο 1).

Ο σύνθετος τύπος δεδομένων με δόμηση, είναι μια οργανωμένη συλλογή από συστατικά, όπου υπάρχει μια σχέση μεταξύ των μερών της συλλογής. Η σχέση αυτή επιτρέπει την πρόσβαση σε μεμονωμένα μέρη της συλλογής αλλά και το χειρισμό της συλλογής ως μιας ενότητας. Στην Pascal οι σύνθετοι τύποι δεδομένων με δόμηση είναι οι Πίνακες και τα Αρχεία. Ένας σύνθετος τύπος δεδομένων χωρίς δόμηση περιλαμβάνει δεδομένα που οι τιμές τους δεν έχουν καμία σχέση πέραν του ότι είναι μέλη της ίδιας ενότητας. Στην Pascal, σύνθετοι τύποι δεδομένων χωρίς δόμηση είναι τα Σύνολα και οι Εγγραφές.

11. Ποιες είναι οι δομές δεδομένων; Πώς διακρίνονται οι στατικές από τις δυναμικές δομές δεδομένων; Να δώσετε παραδείγματα στατικών δομών δεδομένων και δυναμικών δομών δεδομένων.

Δομές δεδομένων ονομάζονται οι σύνθετοι τύποι δεδομένων. Οι δομές δεδομένων διακρίνονται σε στατικές και σε δυναμικές. Στις στατικές δομές δεδομένων το πλήθος των στοιχείων είναι σταθερό και καθορίζεται από τον ορισμό του τύπου στο τμήμα δηλώσεων του προγράμματος. Παράδειγμα στατικής δομής δεδομένων είναι ο Πίνακας. Στις δυναμικές δομές δεδομένων, το πλήθος των στοιχείων της δομής καθορίζεται κατά τη διάρκεια της εκτέλεσης του προγράμματος και πιθανόν να αυξομειώνεται. Παραδείγματα ³⁴δυναμικών δομών δεδομένων είναι οι συνδεδεμένες λίστες, η στοίβα, η ουρά, τα δυαδικά δένδρα κτλ.

³⁴ Στην εξεταστέα ύλη δεν αναφέρονται παραδείγματα δυναμικών δομών δεδομένων.
Σχολικός Σύμβουλος Πληροφορικής και καθηγητές Πληροφορικής ΕΠΑ.Λ Μεσσηνίας & Λακωνίας