

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Τόμος 2ος

ΣΤΟΙΧΕΙΑ ΑΡΧΙΚΗΣ ΕΚΔΟΣΗΣ

Ομάδα Συγγραφής

ΑΘΗΝΑ ΒΑΚΑΛΗ, Λέκτωρ Πληροφορικής ΑΠΘ

ΗΛΙΑΣ ΓΙΑΝΝΟΠΟΥΛΟΣ, Μηχανικός Πληροφορικής

ΝΕΣΤΩΡ ΙΩΑΝΝΙΔΗΣ, Επίκουρος Καθηγητής Τμήματος
Πληροφορικής ΤΕΙ Αθήνας

ΧΡΗΣΤΟΣ ΚΟΙΛΙΑΣ, Επίκουρος Καθηγητής Τμήματος
Πληροφορικής ΤΕΙ Αθήνας

ΚΩΝ/ΝΟΣ ΜΑΛΑΜΑΣ, M.Sc. Πληροφορικής,
Σύμβουλος Επιχειρήσεων

ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΠΟΥΛΟΣ, Αναπληρωτής
Καθηγητής Τμήματος Πληροφορικής ΑΠΘ

ΠΑΝΑΓΙΩΤΗΣ ΠΟΛΙΤΗΣ, Δρ. Διδακτικής
Πληροφορικής, Καθηγητής ΠΕ 19

Υπεύθυνος για το Παιδαγωγικό Ινστιτούτο

ΓΕΩΡΓΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ, Σύμβουλος Π.Ι.
(κατά τη συγγραφή)

Επιτροπή Αξιολόγησης

ΚΩΝ/ΝΟΣ ΓΙΑΛΟΥΡΗΣ, Καθηγητής ΠΕ 19

ΕΥΑΓΓΕΛΙΑ ΓΟΥΛΗ, Καθηγήτρια ΠΕ 19

ΔΗΜΗΤΡΙΟΣ ΔΕΣΠΟΤΗΣ, Αναπληρωτής Καθηγητής
Πανεπιστημίου Πειραιώς

ΚΩΝ/ΝΟΣ ΖΑΧΑΡΗΣ, ΠΛΗΝΕΤ Καρδίτσας

ΑΛΕΞΑΝΔΡΟΣ ΣΙΔΕΡΙΔΗΣ, Καθηγητής Γεωπονικού
Πανεπιστημίου Αθηνών

Εικονογράφηση

ΑΓΓΕΛΟΣ ΑΓΙΟΣΤΡΑΤΙΤΗΣ

Ηλεκτρονική σελιδοποίηση

ANNA ΑΡΓΥΡΟΠΟΥΛΟΥ - ΧΡΗΣΤΟΣ ΠΙΓΚΑΣ

Εξώφυλλο

ΣΠΥΡΟΣ ΣΙΑΚΑΣ - ΝΙΚΟΣ ΓΕΩΡΓΙΟΥ

Επιμέλεια

**ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, Στουρνάρη 49Α, 106 82,
Αθήνα, Τηλ. 38.45.594**

Φορέας

**ΕΛΛΗΝΙΚΗ ΕΤΑΙΡΙΑ ΕΠΙΣΤΗΜΟΝΩΝ Η/Υ ΚΑΙ
ΠΛΗΡΟΦΟΡΙΚΗΣ (ΕΠΥ), Μαυρομιχάλη 16, Αθήνα,
τηλ.: 3645274, e.mail: epy@epy.gr**

Συντονιστές έργου

ΣΠ. ΜΠΑΚΟΓΙΑΝΝΗΣ, πρόεδρος Δ.Σ.

ΒΑΣ. ΜΙΧΑΛΑΚΟΠΟΥΛΟΣ, μέλος Δ.Σ.

Ενέργεια 1.1.α: «Προγράμματα Βιβλία»

Επιστημονικός Υπεύθυνος Ενέργειας

**Θεόδωρος Γ. Εξαρχάκος, Καθηγητής του Πανεπιστημίου
Αθηνών, Πρόεδρος του Παιδαγωγικού Ινστιτούτου**

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

Επιστημονικός Υπεύθυνος Έργου

**Γιάννης Σαλβαράς, Επίκουρος Καθηγητής του
Δημοκρίτειου Πανεπιστημίου Θράκης**

Καλλιτεχνικός Υπεύθυνος Έργου

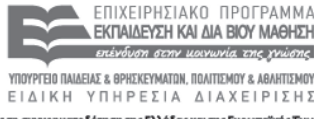
**Σπύρος Ι. Παπασπύρου, Καθηγητής Εφαρμογών του
ΤΕΙ Ηπείρου**

ΣΤΟΙΧΕΙΑ ΕΠΑΝΕΚΔΟΣΗΣ

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



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
Επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

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

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ**

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

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

Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον

Τετράδιο Μαθητή

Γ' Γενικού Λυκείου

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

Τόμος 2ος

**ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»**

Η αξιολόγηση, η κρίση των προσαρμογών και η επιστημονική επιμέλεια του προσαρμοσμένου βιβλίου πραγματοποιείται από τη Μονάδα Ειδικής Αγωγής του Ινστιτούτου Εκπαιδευτικής Πολιτικής.

Η προσαρμογή του βιβλίου για μαθητές με μειωμένη όραση από το ΙΤΥΕ – ΔΙΟΦΑΝΤΟΣ πραγματοποιείται με βάση τις προδιαγραφές που έχουν αναπτυχθεί από ειδικούς εμπειρογνώμονες για το ΙΕΠ.

**ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΒΙΒΛΙΟΥ
ΓΙΑ ΜΑΘΗΤΕΣ ΜΕ ΜΕΙΩΜΕΝΗ ΟΡΑΣΗ**

ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ

Κεφάλαιο 6

Εισαγωγή στον προγραμματισμό

6.1. Προσδοκώμενα αποτελέσματα

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

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

6.2. Δραστηριότητες - ασκήσεις

ΔΤ1.

Ο παρακάτω αλγόριθμος αποτελεί τμήμα μη δομημένου προγράμματος.

Να γράψεις αλγόριθμο σχεδιασμένο με τις αρχές του δομημένου προγραμματισμού, που να εκτελεί τις ίδιες λειτουργίες.

ΑΡΧΗ

ΟΣΟ συνθήκη1 ΕΠΑΝΑΛΑΒΕ

 Εντολή 2

 ΑΝ συνθήκη3 ΤΟΤΕ

 Εντολή4

 Πήγαινε στο Τέλος

 ΑΛΛΙΩΣ

 Εντολή5

 ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ

ΔΤ2.

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

Το πρόβλημα αυτό περιλαμβάνει τη συγκέντρωση

των απαντήσεων, την επεξεργασία τους και την παρουσίαση σε μορφή πίνακα των αποτελεσμάτων για όλες τις χώρες της Ευρωπαϊκής Ένωσης.

ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1.

Στο προγραμματιστικό περιβάλλον να εκτελέσετε ένα έτοιμο πρόγραμμα.

Να κάνετε αλλαγές με το συντάκτη σε μία εντολή, να το μεταγλωττίσετε, να σημειώσετε τα λάθη, να τα διορθώσετε (ουσιαστικά να επαναφέρετε το πρόγραμμα στην αρχική του μορφή) και τελικά να πάρετε τα αποτελέσματα.

ΣΤΟ ΣΠΙΤΙ

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1.

Να καταγράψεις σε έναν κατάλογο τα ονόματα και κυρίως τα βασικά χαρακτηριστικά των γλωσσών προγραμματισμού. Συγκεκριμένα για κάθε γλώσσα προγραμματισμού να καταγράψεις: το όνομά της, τις διάφορες εκδόσεις της, το είδος των δραστηριοτήτων για την οποία χρησιμοποιείται, τη διάδοσή της σε προσωπικούς υπολογιστές, και το είδος του προγραμματισμού

που υποστηρίζει, για παράδειγμα αντικειμενοστραφή, οπτικό, οδηγούμενο από τα γεγονότα κ.λπ.

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

ΔΣ2.

Ο παρακάτω αλγόριθμος αποτελεί τμήμα μη δομημένου προγράμματος.

Να γράψεις αλγόριθμο σχεδιασμένο με τις αρχές του δομημένου προγραμματισμού, που να εκτελεί τις ίδιες λειτουργίες.

ΑΡΧΗ

ΑΝ συνθήκη1 ΤΟΤΕ

Εντολή1

ΑΝ συνθήκη2 ΤΟΤΕ

Εντολή2

Εντολή3

Πήγαινε στην Εντολή5

ΤΕΛΟΣ_ΑΝ

Εντολή4

Εντολή5

Πήγαινε στην Αρχή

ΤΕΛΟΣ_ΑΝ

Εντολή6

ΤΕΛΟΣ

6.3. Τεστ αυτοαξιολόγησης ●

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

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

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

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

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

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

- B) Μεταγλωττιστής
- Γ) Διερμηνευτής
- Δ) Συνδέτης

9. Η Pascal είναι μία γλώσσα:

- A) Μηχανής
- B) Υψηλού επιπέδου
- Γ) Συμβολική
- Δ) 4ης γενιάς

10. Ο μεταγλωττιστής επισημαίνει:

- A) Όλα τα λάθη του προγράμματος
- B) Μόνο τα λογικά λάθη του προγράμματος
- Γ) Μόνο τα συντακτικά λάθη του προγράμματος
- Δ) Μόνο τα λάθη που προέρχονται από αναγραμματισμό των εντολών

11. Ο δομημένος προγραμματισμός είναι:

- A) Μία γενική μεθοδολογία ανάπτυξης προγραμμάτων
- B) Ένας τρόπος προγραμματισμού που εφαρμόζεται μόνο από τη γλώσσα Pascal
- Γ) Η εξέλιξη του τμηματικού προγραμματισμού
- Δ) Ένας τρόπος να εξαλείψουμε τις εντολές GOTO από ένα πρόγραμμα

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

12. Ποια από τα παρακάτω είναι χαρακτηριστικά

ενός δομημένου προγράμματος:

A) Δομικό στοιχείο είναι τα αντικείμενα.

B) Έχει μία είσοδο και μία έξοδο.

Γ) Χρησιμοποιεί τις τρεις δομές: της ακολουθίας, της επιλογής και της επανάληψης.

Δ) Μπορεί να εκμεταλλευτεί τους παράλληλους υπολογιστές.

13. Κάθε φυσική γλώσσα προσδιορίζεται από:

A) Το αλφάβητό της

B) Το λεξιλόγιό της

Γ) Τη γραμματική της

Δ) Τη σημασιολογία της

14. Ποιες από τις παρακάτω γλώσσες χρησιμοποιούνται για ανάπτυξη εφαρμογών τεχνητής νοημοσύνης:

A) LISP

B) FORTRAN

Γ) COBOL

Δ) PROLOG

E) JAVA



Κεφάλαιο

7

Βασικές Έννοιες Προγραμματισμού

7.1. Προσδοκώμενα αποτελέσματα

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

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

7.2. Επιπέδον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

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

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

v_0 : αρχική ταχύτητα

y_0 : το αρχικό ύψος από τη γη

g : η επιτάχυνση της βαρύτητας

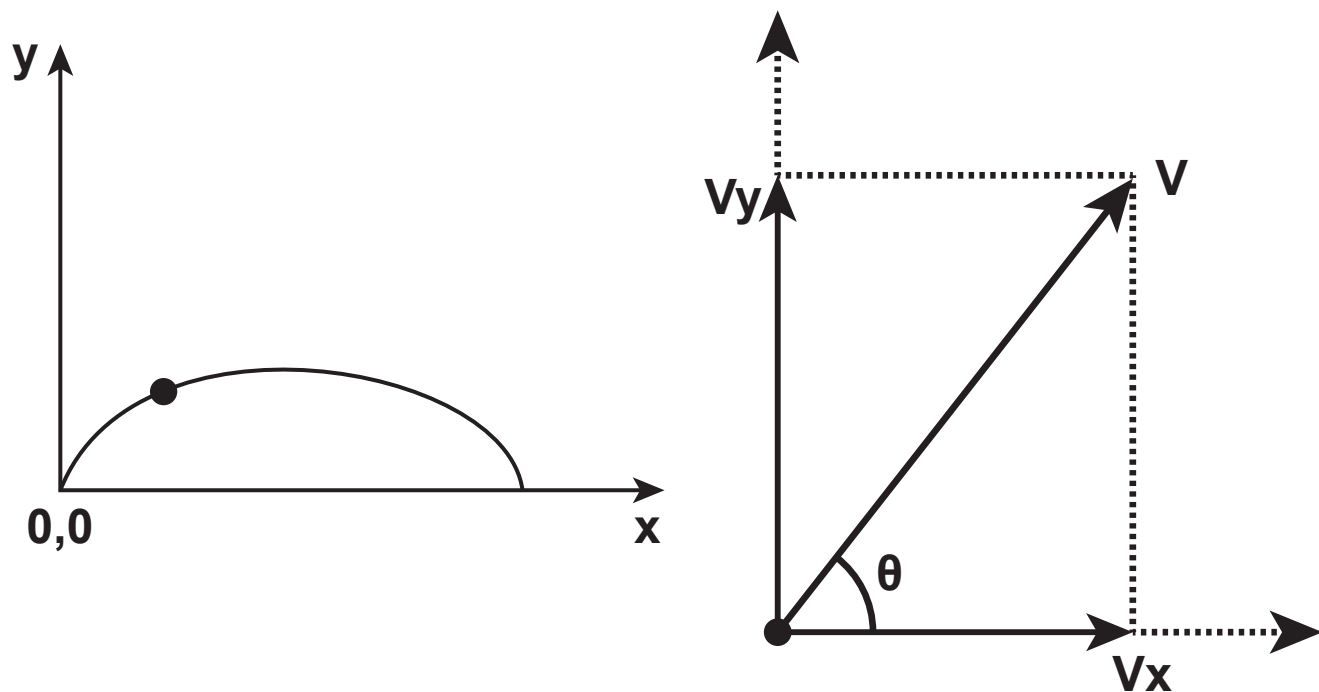
θ : η γωνία βολής

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

x_0 : η αρχική οριζόντια θέση

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

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



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



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

Περιβάλλον προγραμματισμού ΓΛΩΣΣΑ

ΠΡΟΓΡΑΜΜΑ Τροχιά_μπάλας

! Πρόγραμμα υπολογισμού θέσης μπάλας

! Θεωρούμε την αντίσταση του αέρα αμελητέα και αγνοούμε την καμπυλότητα της γης

ΣΤΑΘΕΡΕΣ

$G = 9.81$

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: $V_0, V_{X0}, V_{Y0}, \Theta$, Βεληνεκές

ΑΡΧΗ

! Αρχικές τιμές

$V_0 \leftarrow 20$

! Εισαγωγή δεδομένων

ΓΡΑΨΕ 'Δώσε τη γωνία'

ΔΙΑΒΑΣΕ Θ

! Υπολογισμοί

$V_{X0} \leftarrow V_0 * \Sigma\Upsilon\Nu(\Theta)$

$V_{Y0} \leftarrow V_0 * \text{HM}(\Theta)$

Βεληνεκές $\leftarrow 2 * V_{X0} * V_{Y0} / G$

! Εμφάνιση αποτελεσμάτων

ΓΡΑΨΕ 'Το βεληνεκές για γωνία', Θ , ' είναι:', Βεληνεκές

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Περιβάλλον προγραμματισμού PASCAL

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

    writeln('ΓΩΝΙΑ',A:3,' ΒΕΛΗΝΕΚΕΣ:',range:7:2);
end.
```

Περιβάλλον προγραμματισμού BASIC

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

```
vel = 2 * vx0 * vy0 / g
PRINT "Βεληνεκές ="; vel
END
```

ΠΑΡΑΔΕΙΓΜΑ 2

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

Συγκεκριμένα δίνεται από τον τύπο:

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

a: Μηνιαία δόση

p: Αρχική τιμή αυτοκινήτου

d: Ποσό προκαταβολής

i: Μηνιαίο επιτόκιο

m: Περίοδος αποπληρωμής σε μήνες

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

ΠΡΟΓΡΑΜΜΑ Υπολογισμός_δόσεων

!Υπολογισμός δόσεων αγοράς αυτοκινήτου

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Τιμή, Προκαταβολή, Περίοδος, Υπόλοιπο

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

!Τιμή: Αρχική τιμή αυτοκινήτου σε ευρώ

!Προκαταβολή: Ποσό προκαταβολής σε ευρώ

!Δόση: Μηνιαία δόση σε ευρώ

!Ετήσιο_επιτόκιο: ετήσιο επιτόκιο σε ποσοστό %

!Επιτόκιο: Μηνιαίο επιτόκιο σε δεκαδικό αριθμό

!Περίοδος: Περίοδος αποπληρωμής σε μήνες

!Τελικό_ποσό: Συνολικό ποσό όλων των δόσεων

!Ποσοστό: Ποσοστό διαφοράς τιμής δόσεων με αρχική τιμή

ΑΡΧΗ

ΓΡΑΨΕ `Δώσε την αρχική τιμή: `

ΔΙΑΒΑΣΕ Τιμή

ΓΡΑΨΕ `Δώσε την προκαταβολή: `

ΔΙΑΒΑΣΕ Προκαταβολή

ΓΡΑΨΕ `Δώσε επιτόκιο: `

ΔΙΑΒΑΣΕ Ετήσιο_επιτόκιο

ΓΡΑΨΕ `Δώσε περίοδο σε μήνες: `

ΔΙΑΒΑΣΕ Περίοδος

Επιτόκιο \leftarrow (Ετήσιο_επιτόκιο/12)/100

Υπόλοιπο \leftarrow Τιμή-Προκαταβολή

Κλάσμα \leftarrow $(1+\text{Επιτόκιο})^{\text{Περίοδος}}/((1+\text{Επιτόκιο})^{\text{Περίοδος}}-1)$

Δόση \leftarrow $\text{Επιτόκιο} * \text{Υπόλοιπο} * \text{Κλάσμα}$

Τελικό_ποσό \leftarrow $\text{Δόση} * \text{Περίοδος} + \text{Προκαταβολή}$

Ποσοστό \leftarrow $\text{Τελικό_ποσό} / \text{Τιμή} * 100$

ΓΡΑΨΕ ` Η μηνιαία δόση είναι: `, Δόση

ΓΡΑΨΕ ` Το συνολικό ποσό είναι: `, Τελικό_ποσό,

ΓΡΑΨΕ ` η αύξηση από το αρχικό είναι `, Ποσοστό ,
`%`

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ



7.3. Συμβουλές - υποδείξεις

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

- Τα προγράμματά σου πρέπει να είναι **απλά** και **κατανοητά**. Όχι μόνο για τους άλλους που θα χρειαστεί κάποια στιγμή να τα διαβάσουν και να τα καταλάβουν, αλλά και για σένα τον ίδιο που μετά από κάποιο καιρό θα επανέλθεις σε παλαιότερό σου πρόγραμμα για να το τροποποιήσεις ή να το επεκτείνεις.
- να χρησιμοποιείς ονόματα σταθερών και μεταβλητών που να υπονοούν τη χρήση τους.
- να γράφεις σχόλια μέσα στο πρόγραμμά σου, ειδικά σε εκείνα τα σημεία του που υπάρχουν σχετικές δυσκολίες στην κατανόηση.
- η χρήση κενών γραμμών διευκολύνει στην ανάγνωση του προγράμματος και οριοθετεί τις ενότητες του.
- η χρησιμοποίηση σταθερών σε διευκολύνει σε πιθανές επόμενες αλλαγές και σε προστατεύει από ενδεχόμενες αθέλητες τροποποιήσεις.

- Θα πρέπει να αποδίδεις αρχικές τιμές στις μεταβλητές που χρησιμοποιείς στο πρόγραμμα. Σε πολλά προγραμματιστικά περιβάλλοντα η ίδια η γλώσσα φροντίζει έτσι ώστε να αποδίδει αυτόματα αρχική τιμή ίση με μηδέν στις μεταβλητές. Η δυνατότητα αυτή όμως πολλές φορές μπορεί να οδηγήσει σε λάθος αποτελέσματα. Επιπλέον η απόδοση αρχικών τιμών βοηθάει στην καλύτερη κατανόηση του προγράμματος και στην ευκολότερη συντήρησή του.
- Να αποφεύγεις να χρησιμοποιείς μεγάλους υπολογισμούς. Η διάσπαση ενός υπολογισμού σε απλούστερους διευκολύνει τους άλλους στην κατανόηση του προγράμματος και σένα στην αποφυγή λαθών.

ΣΤΗΝ ΤΑΞΗ

7.4. Δραστηριότητες - ασκήσεις

ΔΤ1.

Να μετατρέψετε σε κώδικα προγράμματος τις παρακάτω παραστάσεις:

1. Η περίοδος γραμμικής αρμονικής ταλάντωσης είναι:

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

2. Η κινητική ενέργεια ενός σώματος είναι:

$$E_{\text{κιν}} = \frac{1}{2}mv^2$$

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

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

4. Η μία λύση εξίσωσης Β' βαθμού είναι:

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

ΔΤ2.

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

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

ΔΤ3.

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

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

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

όπου τ είναι η ημιπερίμετρος του τριγώνου, δηλαδή

$$= \frac{\alpha + \beta + \gamma}{2}.$$

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

ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1.

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

ΔΕ2.

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

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

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

ΔΕ3.

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

ΣΤΟ ΣΠΙΤΙ

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1.

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

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

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

ΔΣ2.

Η περίοδος ενός εκκρεμούς δίνεται από τον τύπο:

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

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

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

ΔΣ3.

Για να υπολογίσουμε τη ροή του αίματος στον ανθρώπινο οργανισμό χρησιμοποιούμε τον τύπο ροής

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

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

7.5. Τεστ αυτοαξιολόγησης

1.

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

1. A) Δήλωση μεταβλητών
B) Δήλωση σταθερών
Γ) Επικεφαλίδα προγράμματος
Δ) Εντολή εισόδου ΔΙΑΒΑΣΕ
2. A) ΓΡΑΨΕ 'Η συνολική τιμή είναι', Τιμή
B) ΔΙΑΒΑΣΕ N
Γ) Κόστος $\leftarrow N * 100$
Δ) Τιμή \leftarrow Κόστος + Κόστος * 0.18

2.

Συμπλήρωσε τα κενά με τη σωστή λέξη που λείπει:

3. Τα στοιχεία προγράμματος των οποίων η τιμή μπορεί να μεταβληθεί κατά τη διάρκεια εκτέλεσης ενός προγράμματος ονομάζονται _____ .

4. Η τελευταία εντολή κάθε προγράμματος είναι _____ .

3.

Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

5. Η δήλωση των μεταβλητών που χρησιμοποιούνται σε ένα πρόγραμμα είναι υποχρεωτική.
6. Το σύμβολο της εντολής εκχώρησης είναι το ίσον =.

7. Κατά τον υπολογισμό μιας αριθμητικής παράστασης, πρώτα εκτελείται ο πολλαπλασιασμός και στη συνέχεια η πρόσθεση.
8. Οι λογικές μεταβλητές δέχονται μόνο δύο τιμές.

4.

Διάλεξε ένα μεταξύ των προτεινόμενων:

9. Ποιες από τις παρακάτω εντολές δίνουν σαν αποτέλεσμα εκτέλεσης το μήνυμα:

Η τιμή είναι 100

A) Τιμή \leftarrow 100

ΓΡΑΨΕ 'Η τιμή είναι' 100

B) ΓΡΑΨΕ 'Η τιμή είναι', Τιμή

Γ) Τιμή \leftarrow 100

ΓΡΑΨΕ 'Η τιμή είναι', 100

Δ) Τιμή \leftarrow 100

ΓΡΑΨΕ 'Η τιμή είναι', Τιμή

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

A) 35

B) 134

Γ) 22

Δ) 148

11. Τι θα τυπώσουν οι παρακάτω εντολές:

$A \leftarrow 100$

$X \leftarrow (2 + \text{T_P}(A) * 3/10)^2 - (A + 50)/5$

ΓΡΑΨΕ X

A) 22

B) -5

Γ) 10

Δ) 25

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

A) Πλήθος +1 ← Πλήθος

B) Πλήθος ← Πλήθος+1

Γ) Πλήθος ← +1

Δ) Πλήθος = Πλήθος+1

5.

Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

13. Τα είδη μεταβλητών που υποστηρίζει η ΓΛΩΣΣΑ είναι:

A) ακέραιες

B) πραγματικές

Γ) μιγαδικές

Δ) χαρακτήρες

E) ημερομηνίες

Z) λογικές

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

A) A

B) Στοιχείο1

Γ) 1Στοιχείο

Δ) Φύλο μαθητή

E) Τιμή-σε-\$

Z) ΤΑΧΥΤΗΤΑ



Επιλογή και Επανάληψη

8.1. Προσδοκώμενα αποτελέσματα

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

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

αποδοτικότερου προγράμματος.

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

8.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

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

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

Τα όρια αυτά που ισχύουν για την περιοχή της Αθήνας για δύο από τους πλέον συχνά εμφανιζόμενους

ρύπους O_3 και NO_2 παρουσιάζονται στον παρακάτω πίνακα.

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

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

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

**ΠΡΟΓΡΑΜΜΑ Ρύττοι
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: NO₂, O₃**

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε την τιμή του Διοξειδίου του αζώτου'
ΔΙΑΒΑΣΕ NO₂

ΓΡΑΨΕ 'Δώσε την τιμή του Όζοντος'
ΔΙΑΒΑΣΕ O₃

ΑΝ NO₂ > 700 **Ή** O₃ > 500 **ΤΟΤΕ**

ΓΡΑΨΕ 'ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ
ΑΠΑΓΟΡΕΥΣΗ ΚΥΚΛΟΦΟΡΙΑΣ'

ΑΛΛΙΩΣ_ΑΝ NO₂ > 500 **Ή** O₃ > 300 **ΤΟΤΕ**

ΓΡΑΨΕ 'ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ'

ΑΛΛΙΩΣ_ΑΝ NO₂ > 400 **Ή** O₃ > 250 **ΤΟΤΕ**


ΓΡΑΨΕ 'ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ'

ΑΛΛΙΩΣ

ΓΡΑΨΕ ' Ρύττοι μέσα στα όρια'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ



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

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

Περιβάλλον προγραμματισμού PASCAL

```
program rypoi;
```

```
var
```

```
    no2,o3:real;
```

```
begin
```

```
write ('ΔΩΣΕ ΤΗΝ ΤΙΜΗ ΤΟΥ ΔΙΟΞΕΙΔΙΟΥ:');readln (no2);
```

```
write ('ΔΩΣΕ ΤΗΝ ΤΙΜΗ ΤΟΥ ΟΖΟΝΤΟΣ:'); readln (o3);
```

```
if (no2 > 700) or (o3 > 500) then
```

```
    write ('ΠΑΡΑ ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΑΠΑΓΟΡΕΥΣΗ  
                                                ΚΥΚΛΟΦΟΡΙΑΣ')
```

```
else if (no2 > 500) or (o3 > 300) then
```

```
    write ('ΠΟΛΥ ΥΨΗΛΟΙ ΡΥΠΟΙ ΕΚΤΑΚΤΑ ΜΕΤΡΑ')
```

```
else if (no2 > 400) or (o3 > 250) then
```

```
    write ('ΠΡΟΣΟΧΗ ΥΨΗΛΟΙ ΡΥΠΟΙ')
```

```
else
```

```
    write ('Ρύποι μέσα στα όρια')
```

```
endif
```

```
end.
```

Περιβάλλον προγραμματισμού BASIC

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

ΠΑΡΑΔΕΙΓΜΑ 2

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

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

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

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

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

ΠΡΟΓΡΑΜΜΑ ΛΟΓΑΡΙΑΣΜΟΣ_ΝΕΡΟΥ ΣΤΑΘΕΡΕΣ

ΦΠΑ=0.18

ΤΙΜΗ1=0,34

ΤΙΜΗ2=0,52

ΤΙΜΗ3=1,51

ΤΙΜΗ4=2,11

ΤΙΜΗ5=2,64

ΠΑΓΙΟ=1,47

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Μετρητής

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

ΧΑΡΑΚΤΗΡΕΣ: Όνομα, Επώνυμο

ΑΡΧΗ

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

ΔΙΑΒΑΣΕ Μετρητής

ΟΣΟ Μετρητής <> 0 **ΕΠΑΝΑΛΑΒΕ**

ΓΡΑΨΕ 'Δώσε το Ονοματεπώνυμο'

ΔΙΑΒΑΣΕ Επώνυμο, Όνομα

ΓΡΑΨΕ 'Δώσε την Κατανάλωση'

ΔΙΑΒΑΣΕ Κατανάλωση

ΕΠΙΛΕΞΕ Κατανάλωση

ΠΕΡΙΠΤΩΣΗ <= 15

Αξία ← Κατανάλωση*ΤΙΜΗ1

ΠΕΡΙΠΤΩΣΗ <= 60 **ΤΟΤΕ**

Αξία ← 15*ΤΙΜΗ1 +
+(Κατανάλωση-15)*ΤΙΜΗ2

ΠΕΡΙΠΤΩΣΗ ≤ 81 ΤΟΤΕ

Αξία ← 15*ΤΙΜΗ1+45*ΤΙΜΗ2 +
+(Κατανάλωση-60)*ΤΙΜΗ3

ΠΕΡΙΠΤΩΣΗ ≤ 105 ΤΟΤΕ

Αξία ← 15*ΤΙΜΗ1+45*ΤΙΜΗ2 + 21*ΤΙΜΗ3 +
+(Κατανάλωση-81)*ΤΙΜΗ4

ΠΕΡΙΠΤΩΣΗ_ΑΛΛΙΩΣ

Αξία ← 15*ΤΙΜΗ1+45*ΤΙΜΗ2+21*ΤΙΜΗ3+
+24*ΤΙΜΗ4+(Κατανάλωση-105)*ΤΙΜΗ5

ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

Αποχέτευση ← Αξία*0.4

Άλλα ← Αξία*0.01

Τιμή ← Αξία+Αποχέτευση+Άλλα+ ΠΑΓΙΟ

Αξία_ΦΠΑ ← Τιμή* ΦΠΑ

Τελική_τιμή ← Τιμή + Αξία_ΦΠΑ

ΓΡΑΨΕ 'Ο λογαριασμός του', Επώνυμο,' είναι ',
Τελική_τιμή

ΓΡΑΨΕ 'Αξία νερού:', Αξία

ΓΡΑΨΕ 'άλλα:', Αποχέτευση+ Άλλα, 'ΦΠΑ:',
Αξία_ΦΠΑ

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

ΔΙΑΒΑΣΕ Μετρητής

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΑΡΑΔΕΙΓΜΑ 3

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

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

Περιβάλλον προγραμματισμού ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑ Τροχιά_μπάλας2

ΣΤΑΘΕΡΕΣ

$$G = 9.81$$

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: V0, VX0, VY0, Θ, Βεληνεκές

ΑΡΧΗ

ΓΙΑ Θ ΑΠΟ 20 ΜΕΧΡΙ 80 ΜΕ_ΒΗΜΑ 10

ΓΡΑΨΕ 'Γωνία:', Θ

ΓΙΑ V0 ΑΠΟ 10 ΜΕΧΡΙ 40 ΜΕ_ΒΗΜΑ 10

$$VX0 \leftarrow V0 * \Sigma Y N(\Theta)$$

$$VY0 \leftarrow V0 * \text{ΗΜ}(\Theta)$$

$$\text{Βεληνεκές} \leftarrow 2 * VX0 * VY0 / G$$

ΓΡΑΨΕ 'Ταχύτητα:', V0, 'Βεληνεκές:',
Βεληνεκές

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Περιβάλλον προγραμματισμού PASCAL

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

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

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

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

Περιβάλλον προγραμματισμού BASIC

` Τροχιά μπάλλας 2

$g=9.81$

FOR a = 20 TO 80 STEP 10

akt = a * ATN(1) * 4 / 180

PRINT "Γωνία ";akt

FOR v0 = 10 TO 40 STEP 10

vx0 = v0 * COS(akt)

vy0 = v0 * SIN(akt)

vel = 2 * vx0 * vy0 / g

PRINT "ΜΕ ΑΡΧΙΚΗ ΤΑΧΥΤΗΤΑ";v0

PRINT "ΒΕΛΗΝΕΚΕΣ = "; vel

NEXT v0

NEXT a

END



Η συνάρτηση ATN επιστρέφει το τόξο εφαπτομένης.
Άρα $ATN(1)=\pi/4$, αφού $\varepsilon\varphi(\pi/4)=1$.



8.3. Συμβουλές - υποδείξεις

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

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

- Πριν χρησιμοποιήσεις εμφωλευμένα AN, σκέψου μήπως το ίδιο πρόγραμμα μπορεί να υλοποιηθεί απλούστερα με σύνθετες λογικές εκφράσεις, την εντολή AN-ΑΛΛΙΩΣ_AN ή κάποια άλλη εντολή επιλογής που πιθανόν να προσφέρει το υπολογιστικό περιβάλλον που χρησιμοποιείς.
- Οι μεταβλητές που ελέγχουν την επανάληψη του βρόχου ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ πρέπει υποχρεωτικά να αλλάζουν τιμή μέσα στο σώμα του βρόχου, αλλιώς ή δεν εκτελείται ποτέ ή συνηθέστερα δεν σταματάει η εκτέλεσή του (ατέρμων βρόχος).
- Οι επαναλήψεις που υλοποιούνται με την εντολή ΟΣΟ, μπορεί να μην εκτελεστούν ούτε μία φορά, αφού ο έλεγχος γίνεται στην είσοδο του βρόχου, αντίθετα οι επαναλήψεις ΜΕΧΡΙΣ_ΟΤΟΥ θα πραγματοποιηθούν τουλάχιστον μία φορά.
- Η εντολή ΓΙΑ χρησιμοποιείται μόνο για προκαθορισμένο αριθμό επαναλήψεων. Αν λοιπόν ξέρεις τον αριθμό των επαναλήψεων ή μπορείς να τον υπολογίσεις, τότε να χρησιμοποιείς την εντολή ΓΙΑ.
- Ποτέ μη χρησιμοποιείς εντολές που αλλάζουν την αρχική τιμή, την τελική τιμή, το βήμα ή τη μεταβλητή που ελέγχει την επανάληψη μέσα σε ένα βρόχο ΓΙΑ. Αν και μερικές γλώσσες προγραμματισμού επιτρέπουν αυτές τις αλλαγές, να τις αποφεύγεις, γιατί οδηγούν σε προγράμματα δυσνόητα και συνήθως λανθασμένα.

8.4. Δραστηριότητες - ασκήσεις

ΔΤ1.

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

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

ΔΤ2.

Να γράψεις τις εντολές για τα παρακάτω:

- A) Αν η Βαθμολογία (ΒΑΘΜΟΣ) είναι μεγαλύτερη από το Μέσο όρο (ΜΟ), τότε να τυπώνει “Πολύ καλά”, αν είναι ίση ή μικρότερη του Μέσου όρου μέχρι και 2 μονάδες, να τυπώνει “Καλά”, και, όταν είναι μικρότερη του Μέσου όρου περισσότερο από 2 μονάδες, να τυπώνει “Μέτρια”.
- B) Αν το τμήμα (ΤΜΗΜΑ) είναι Γ1 και η βαθμολογία (ΒΑΘΜΟΣ) είναι μεγαλύτερη από 15, τότε να τυπώνει το επώνυμο (ΕΠΩΝΥΜΟ).
- Γ) Αν η απάντηση (ΑΠΑΝΤΗΣΗ) δεν είναι Ν ή ν ή Ο ή ο, τότε να τυπώνει το μήνυμα “Λάθος απάντηση...”.

Δ) Αν ο αριθμός X είναι αρνητικός ή το $HM(X)=0$, τότε να τυπώνεται το μήνυμα “Λάθος δεδομένα...”, αλλιώς να υπολογίζεται η παράσταση $(X^2+5*X)/(T_P(X)*HM(X))$.

ΔΤ3.

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

ΔΤ4.

Έστω το παρακάτω τμήμα προγράμματος:

K ← 0

ΓΙΑ I ΑΠΟ 0 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 5

A ← I³

K ← K+A

ΓΡΑΨΕ I, A

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ K

Πόσες φορές θα εκτελεστεί ο βρόχος;

Ποια η λειτουργία των εντολών;

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

ΔΤ5.

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

A)

ΔΙΑΒΑΣΕ Μισθός

ΟΣΟ Μισθός <>0 ΕΠΑΝΑΛΑΒΕ

Άθροισμα ← 0

ΑΝ Μισθός > Μέγιστος ΤΟΤΕ

Μέγιστος ← Μισθός

ΤΕΛΟΣ_ΑΝ

ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ

Ελάχιστος ← Μισθός

ΤΕΛΟΣ_ΑΝ

Άθροισμα ← Άθροισμα+Μισθός

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

B)

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

Άθροισμα ← 0

ΑΝ Μισθός > Μέγιστος ΤΟΤΕ

Μέγιστος ← Μισθός

ΤΕΛΟΣ_ΑΝ

ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ

Ελάχιστος ← Μισθός

ΤΕΛΟΣ_ΑΝ

Άθροισμα ← Άθροισμα+Μισθός
ΔΙΑΒΑΣΕ Μισθός
ΜΕΧΡΙΣ_ΟΤΟΥ Μισθός<>0

Γ)

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 100

Άθροισμα ← 0

ΔΙΑΒΑΣΕ Μισθός

ΑΝ Μισθός > Μέγιστος ΤΟΤΕ

Μέγιστος ← Μισθός

ΤΕΛΟΣ_ΑΝ

ΑΝ Μισθός < Ελάχιστος ΤΟΤΕ

Ελάχιστος ← Μισθός

ΤΕΛΟΣ_ΑΝ

Άθροισμα ← Άθροισμα+Μισθός

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

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

ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1.

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

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

Το πρόγραμμα να γραφεί με τους ακόλουθους τρόπους:

- Με εντολές **ΑΝ ... ΤΟΤΕ**
- Με εντολές **ΑΝ ... ΤΟΤΕ ... ΑΛΛΙΩΣ_ΑΝ**
- Με εμφωλευμένα **ΑΝ**
- Με την εντολή **ΕΠΙΛΕΞΕ**

ΔΕ2.

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

ΔΕ3.

Να γραφεί πρόγραμμα το οποίο θα εκτελεί κάποια από τις βασικές πράξεις πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση ανάμεσα σε δύο ακέραιους αριθμούς και θα εμφανίζει το αποτέλεσμα στην οθόνη.

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

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

Δώσε επιλογή:—

ΔΕ4.

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

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

Το πρόγραμμα να εκτελεστεί με δεδομένα τις πραγματικές τιμές ρύπων που μετρήθηκαν τη χθεσινή ημέρα. Οι τιμές αυτές δίδονται από το τμήμα ποιότητας της

ατμόσφαιρας του ΥΠΕΧΩΔΕ και βρίσκονται στη διεύθυνση: www.minenv.gr

ΔΕ5.

* *

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

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

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

Σε παραλληλία

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

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

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

ΣΤΟ ΣΠΙΤΙ

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1.

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

**ΚΛΙΜΑΚΑ ΥΠΟΛΟΓΙΣΜΟΥ ΦΟΡΟΥ ΕΙΣΟΔΗΜΑΤΟΣ ΦΥΣΙΚΩΝ ΠΡΟΣΩΠΩΝ ΟΙΚΟΝ.
ΕΤΟΥΣ 1999**

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

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

Να γραφεί πρόγραμμα το οποίο:

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

ΔΣ2.

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

ΔΣ3.

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

ΔΣ4.

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

ΔΣ5.*******

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

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

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

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

Ποια μπορεί να είναι τα κριτήρια για διακοπή των επαναλήψεων;

Υπόδειξη: Να μετατρέψεις αρχικά τη γωνία x σε ακτίνια.

ΔΣ6.******

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

8.5. Τεστ αυτοαξιολόγησης

1.

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

1.

- A) ΓΡΑΨΕ 'Δεν υπάρχει ρίζα'
- B) AN A>0 ΤΟΤΕ
- Γ) ΤΕΛΟΣ_ΑΝ
- Δ) ΑΛΛΙΩΣ
- E) Ρίζα<-T_P(A)

2.

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

2.

Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

- 3. Οι εντολές που βρίσκονται σε ένα βρόχο ΟΣΟ ΕΠΑΝΑΛΑΒΕ εκτελούνται τουλάχιστον μία φορά.
- 4. Η τιμή του βήματος στην εντολή ΓΙΑ είναι υποχρεωτική να αναγράφεται.
- 5. Κάθε εντολή AN πρέπει να έχει την αντίστοιχη εντολή ΤΕΛΟΣ_ΑΝ.
- 6. Κάθε βρόχος που υλοποιείται με την εντολή ΟΣΟ ΕΠΑΝΑΛΑΒΕ μπορεί να γραφεί και με

χρήση της εντολής ΓΙΑ.

7. Αν το A έχει την τιμή 5 και το B την τιμή 6, τότε η λογική έκφραση $A > 5 \text{ Ή } A < 3 \text{ ΚΑΙ } B > 5$ είναι ψευδής.

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

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

A)

Άθροισμα $\leftarrow 0$

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100

 Άθροισμα \leftarrow Άθροισμα+I

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

B)

Άθροισμα $\leftarrow 0$

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2

 Άθροισμα \leftarrow Άθροισμα+I

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Γ)

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2

 Άθροισμα $\leftarrow 0$

 Άθροισμα \leftarrow Άθροισμα+I

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Δ)

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ_ΒΗΜΑ 2

 Άθροισμα $\leftarrow I$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

9. Τι θα εκτυπώσει το παρακάτω τμήμα προγράμματος:

$A \leftarrow 0$

ΓΙΑ I ΑΠΟ 10 ΜΕΧΡΙ 20 ΜΕ_ΒΗΜΑ 10

$A \leftarrow A + I^2$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ A

A) 0

B) 100

Γ) 500

Δ) 400

10. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη:

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

$A \leftarrow 0$

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5

$A \leftarrow A - 1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΕΧΡΙΣ_ΟΤΟΥ $A = 0$

A) 10

B) 0

Γ) 5

Δ) Άπειρες

11. Δίνονται οι παρακάτω εντολές:

$A \leftarrow 1$

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2

$A \leftarrow A * I$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

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

A)
A ← 1
I ← 1
ΟΣΟ I ≤ 10 ΕΠΑΝΑΛΑΒΕ
 I ← I+2
 A ← A*I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

B)
A ← 1
I ← 1
ΟΣΟ I ≤ 10 ΕΠΑΝΑΛΑΒΕ
 A ← A*I
 I ← I+2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Γ)
A ← 1
I ← 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
 A ← A*I
 I ← I+2
ΜΕΧΡΙΣ_ΟΤΟΥ I < 10

Δ)
A ← 1
I ← 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
 A ← A*I
 I ← I+2
ΜΕΧΡΙΣ_ΟΤΟΥ I = 10

12. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη:

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 2 ΜΕ_ΒΗΜΑ 3

ΓΡΑΨΕ 'Μήνυμα'

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

A) 2

B) 0

Γ) 1

Δ) Άπειρες

13. Ποια η λειτουργία του παρακάτω τμήματος προγράμματος:

B ← 10

ΔΙΑΒΑΣΕ A

B ← A

ΑΝ A < 0 ΤΟΤΕ

 B ← -A

ΤΕΛΟΣ_ΑΝ

A ← 0

ΓΡΑΨΕ B

A) Τυπώνει τον αριθμό που διάβασε.

B) Τυπώνει την απόλυτη τιμή του αριθμού που διάβασε.

Γ) Τυπώνει πάντα την τιμή 0.

Δ) Τυπώνει πάντα την τιμή 10.

Κεφάλαιο 9

Πίνακες

9.1. Προσδοκώμενα αποτελέσματα

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

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

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

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

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

9.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

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

ΠΡΟΓΡΑΜΜΑ Αεροπορικές_εταιρείες

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:N,I, Εισπράξεις[50], Σύνολο

ΠΡΑΓΜΑΤΙΚΕΣ:ΜΟ

ΧΑΡΑΚΤΗΡΕΣ:Εταιρεία[50]

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

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

ΔΙΑΒΑΣΕ N

ΜΕΧΡΙΣ_ΟΤΟΥ N<=50

Σύνολο ← 0


```

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΓΡΑΨΕ 'Δώσε αεροπορική εταιρεία ...'
    ΔΙΑΒΑΣΕ Εταιρεία[I]
    ΓΡΑΨΕ 'Δώσε εισπράξεις ...'
    ΔΙΑΒΑΣΕ Εισπράξεις [I]
    Σύνολο ← Σύνολο+Εισπράξεις[I]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΟ ← Σύνολο/Ν
ΓΡΑΨΕ 'Μεγαλύτερες από το μέσο όρο'
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Ν
    ΑΝ Εισπράξεις[I]> ΜΟ ΤΟΤΕ
        ΓΡΑΨΕ Εταιρεία[I]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

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

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

Περιβάλλον προγραμματισμού PASCAL

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

Περιβάλλον προγραμματισμού Basic

```
` Αεροπορικές εταιρίες
DIM company$(50), E(50)
DO
  INPUT "Αριθμός εταιριών:", n
```

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

ΠΑΡΑΔΕΙΓΜΑ 2

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

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

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

ΠΡΟΓΡΑΜΜΑ Αυτοκίνητα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Θόρυβος [4,5], I,J, Ταχύτητα[5],
Άθροισμα, Συν_Άθροισμα

ΧΑΡΑΚΤΗΡΕΣ: Μοντέλο[4]

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, Συν_ΜΟ

!Εισαγωγή δεδομένων

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 5

ΓΡΑΨΕ `Δώσε ταχύτητα..`

ΔΙΑΒΑΣΕ Ταχύτητα[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4

ΓΡΑΨΕ `Δώσε μοντέλο..`

ΔΙΑΒΑΣΕ Μοντέλο[I]

ΓΡΑΨΕ `Δώσε επίπεδα θορύβου..`

ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5

ΔΙΑΒΑΣΕ Θόρυβος[I,J]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Υπολογισμοί μέσω των τιμών

Συν_Άθροισμα \leftarrow 0

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4

Άθροισμα \leftarrow 0

ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5

Άθροισμα \leftarrow Άθροισμα+Θόρυβος[I,J]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Συν_Άθροισμα \leftarrow Συν_Άθροισμα+Άθροισμα

ΜΟ \leftarrow Άθροισμα/5

ΓΡΑΨΕ Μοντέλο[I], `:`, ΜΟ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 5

Άθροισμα \leftarrow 0

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 4

Άθροισμα \leftarrow Άθροισμα+Θόρυβος[I,J]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ \leftarrow Άθροισμα/4

ΓΡΑΨΕ Ταχύτητα[J] `:', ΜΟ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Συν_ΜΟ ← Συν_Άθροισμα/20
ΓΡΑΨΕ `Συνολικό μέσο επίπεδο θορύβου:', Συν_ΜΟ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΠΑΡΑΔΕΙΓΜΑ 3

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

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

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

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

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

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

ΠΡΟΓΡΑΜΜΑ Συγχώνευση ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[100], B[100], Γ[200], I, J, K, N, M, Λ
! A και B αρχικοί πίνακες
! Γ τελικός πίνακας

ΑΡΧΗ

! Διάβασε τα δεδομένα

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

ΔΙΑΒΑΣΕ N

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N

ΔΙΑΒΑΣΕ A[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

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

ΔΙΑΒΑΣΕ M

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ M

ΔΙΑΒΑΣΕ B[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Συγχώνευση πινάκων

! I είναι ο δείκτης για τον πίνακα A

! J είναι ο δείκτης για τον πίνακα B

! K είναι ο δείκτης για τον πίνακα Γ

I ← 1

J ← 1

K ← 1

ΟΣΟ $I \leq N$ ΚΑΙ $J \leq M$ ΕΠΑΝΑΛΑΒΕ

! Όσο και τα δύο έχουν στοιχεία

ΑΝ $A[I] < B[J]$ ΤΟΤΕ

$\Gamma[K] \leftarrow A[I]$

$K \leftarrow K+1$

$I \leftarrow I+1$

ΑΛΛΙΩΣ

$\Gamma[K] \leftarrow B[J]$

$K \leftarrow K+1$

$J \leftarrow J + 1$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Μεταφορά των υπολοίπων στοιχείων του A ή του B

ΑΝ $I > N$ ΤΟΤΕ

ΓΙΑ Λ ΑΠΟ K ΜΕΧΡΙ $N+M$

$\Gamma[\Lambda] \leftarrow B[J]$

$J \leftarrow J + 1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΛΙΩΣ

ΓΙΑ Λ ΑΠΟ K ΜΕΧΡΙ $N+M$

$\Gamma[\Lambda] \leftarrow A[I]$

$I \leftarrow I+1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Εκτύπωση τελικού πίνακα

ΓΙΑ Λ ΑΠΟ 1 ΜΕΧΡΙ $N+M$

ΓΡΑΨΕ $\Gamma[\Lambda]$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Γλώσσα προγραμματισμού PASCAL

program merge;

var

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

{A και B αρχικοί ταξινομημένοι πίνακες C τελικός πίνακας}

begin

write ('Δώσε τη διάσταση του πίνακα A (n) ');
readln(n);

for i:=1 to n **do**

readln(a[i]);

write ('Δώσε τη διάσταση του πίνακα B (m) ');
readln (m);

for i:=1 to m **do**

readln (b[i]);

i:=1; j:=1; k:=1;

while (i<=n) **and** (j<=m) **do**

if (a[i]<b[j]) **then**

begin

c[k]:=a[i];k:=k+1;i:=i+1;

end

else

```

    begin
        c[k]:=b[j]; k:=k+1; j:=j+1;
    end;
if i>n then
    for l:=k to n+m do
        begin
            c[l]:=b[j]; j:=j+1;
        end
    else
        for l:=k to n+m do
            begin
                c[l]:=a[i]; i:=i+1;
            end;
        for l:=1 to n+m
            write (c[l]);
        end.

```

Περιβάλλον προγραμματισμού Basic

` Merging

```
DIM a(100), b(100), c(200)
```

```
READ n
```

```
FOR i = 1 TO n: READ a(i): NEXT i
```

```
DATA 5
```

```
DATA 2,7,12,18,26
```

```
READ m
```

```
FOR i = 1 TO m: READ b(i): NEXT i
```

```
DATA 5
```

```
DATA 1,6,10,15,25
```

```
i = 1: j = 1: k = 1
```

```

WHILE i <= n AND j <= m
  IF a(i) < b(j) THEN
    c(k) = a(i): k = k + 1: i = i + 1
  ELSE
    c(k) = b(j): j = j + 1: k = k + 1
  END IF
WEND
IF i > n THEN
  FOR r = k TO n + m
    c(r) = b(j): j = j + 1
  NEXT r
ELSE
  FOR r = k TO n + m
    c(r) = a(i): i = i + 1
  NEXT r
END IF
FOR i = 1 TO m + n
  PRINT c(i)
NEXT i
END

```



9.3. Συμβουλές - υποδείξεις

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

- Εξέτασε αν πραγματικά χρειάζεται πίνακας για την επίλυση του προβλήματος. Αν δεν είναι απαραίτητος, μην τον χρησιμοποιείς. Να έχεις πάντα στο νου σου ότι οι πίνακες ξοδεύουν μεγάλα ποσά μνήμης.
- Για να αποφύγεις τα πλέον κοινά λάθη στη χρήση των πινάκων να προσέχεις πάντα:
 - Να δίνεις αρχικές τιμές σε όλους τους πίνακες.
 - Μην ξεπερνάς τα όρια του πίνακά σου. Το πιο συνηθισμένο λάθος στη χρήση των πινάκων είναι η προσπάθεια ανάγνωσης ή εκχώρησης τιμής έξω από τα όρια του πίνακα.
 - Η επεξεργασία γίνεται στα στοιχεία του πίνακα. Άρα σε όλες τις εντολές πρέπει να εμφανίζονται τα στοιχεία του πίνακα και όχι το όνομα του ίδιου του πίνακα.
 - Όλα τα στοιχεία του πίνακα έχουν τον ίδιο τύπο, για παράδειγμα όλα είναι ακέραια ή όλα είναι χαρακτήρες όπως ορίστηκαν στο τμήμα δηλώσεων.
 - Στην ταξινόμηση ή την αναζήτηση σε έναν πίνακα να χρησιμοποιείς πάντα τη μέθοδο που είναι πιο κατάλληλη.

9.4. Δραστηριότητες - ασκήσεις

ΔΤ1.

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

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

ΔΤ2.

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

ΔΤ3.

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

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

ΔΤ4.

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

ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1.

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

ΔΕ2.

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

ΔΕ3.

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

ΔΕ4.

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

ΣΤΟ ΣΠΙΤΙ

Στο τετράδιό σας αντιμετωπίστε τα παρακάτω προβλήματα:

ΔΣ1.

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

ΔΣ2.

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

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

Πρόσθεση: $c_{ij} = a_{ij} + b_{ij}$

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

ΔΣ3.

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

ΔΣ4.

* *

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

ΔΣ5.

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

ΔΣ6.

* * *

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

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



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

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

9.5. Τεστ αυτοαξιολόγησης

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

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

Άθροισμα ← 0

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N

 Άθροισμα ← Άθροισμα+A[I,I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

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

7. Η χρήση των πινάκων σε ένα πρόγραμμα αυξάνει την απαιτούμενη μνήμη.

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

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

A)

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

B)

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

Γ)

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

Δ)

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

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

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 8

 Άθροισμα ← 0

 ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 10

 Άθροισμα ← Άθροισμα+A[I,J]

 ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

 ΜΟ ← Άθροισμα/10

 ΓΡΑΨΕ ΜΟ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

A) Γράφει το μέσο όρο των στοιχείων του πίνακα.

B) Γράφει το μέσο όρο των στοιχείων κάθε γραμμής.

Γ) Γράφει το μέσο όρο των στοιχείων κάθε στήλης.

Δ) Γράφει το μέσο όρο της τελευταίας γραμμής.

10. Ποιο είναι το αποτέλεσμα των παρακάτω εντολών:

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10

 A[I] ← 10+I

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΣΥΝ ← 0

ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2

 ΣΥΝ ← ΣΥΝ+A[Κ]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ΣΥΝ

- A) 75
- B) 155
- Γ) 50
- Δ) 125

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

11. Τυπικές επεξεργασίες σε έναν πίνακα είναι:

- A) Ταξινόμηση
- B) Πρόσθεση στοιχείων
- Γ) Πολλαπλασιασμός στοιχείων
- Δ) Συγχώνευση
- E) Αναζήτηση

12. Η επιλογή του καλύτερου αλγορίθμου ταξινόμησης εξαρτάται από:

- A) Τον τύπο δεδομένων που έχει ο πίνακας
- B) Τη διάσταση του πίνακα
- Γ) Το πλήθος των στοιχείων του πίνακα
- Δ) Την αρχική διάταξη των στοιχείων

● Κεφάλαιο 10

Υποπρογράμματα

10.1. Προσδοκώμενα αποτελέσματα

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

Για να αναλύσεις σωστά ένα σύνθετο πρόγραμμα σε υποπρογράμματα, πρέπει αρχικά να αποφασίζεις πότε θα χρησιμοποιήσεις συναρτήσεις και διαδικασίες, να γνωρίζεις τη δομή καθώς και τα βασικά χαρακτηριστικά αυτών των υποπρογραμμάτων. Κάθε γλώσσα προγραμματισμού έχει ελαφρώς διαφορετικό τρόπο με τον οποίο αντιμετωπίζει τα υποπρογράμματα και ειδικά τον τρόπο με τον οποίο χειρίζεται τις παραμέτρους. Στο βιβλίο σου παρουσιάστηκαν θεωρητικά οι αρχές επικοινωνίας των υποπρογραμμάτων με τη χρήση των παραμέτρων,

στο εργαστήριο θα γνωρίσεις το συγκεκριμένο τρόπο που το δικό σου προγραμματιστικό περιβάλλον υλοποιεί αυτές τις αρχές και πώς χρησιμοποιεί τις παραμέτρους.

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

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

10.2. Επιπλέον παραδείγματα

ΠΑΡΑΔΕΙΓΜΑ 1

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

Το πρόγραμμα χρησιμοποιεί τις εξής διαδικασίες και συναρτήσεις:

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

για τον υπολογισμό της τυπικής απόκλισης.

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

Υπολογισμός_Διαμέσου: Πραγματική συνάρτηση η οποία υπολογίζει τη διάμεσο τιμή.

ΠΡΟΓΡΑΜΜΑ Στατιστική **ΜΕΤΑΒΛΗΤΕΣ**

ΑΚΕΡΑΙΕΣ: I, Πλήθος, Στοιχεία[100], Μέγιστο,
Άθροισμα, Άθροισμα_2, Βοηθητική
ΠΡΑΓΜΑΤΙΚΕΣ: MO, Τυπ_Απόκλιση, Διάμεσος

ΑΡΧΗ

! Εισαγωγή στοιχείων

ΓΡΑΨΕ 'Δώσε το πλήθος των αριθμών (μέγιστο 100)'

ΔΙΑΒΑΣΕ Πλήθος

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ Πλήθος

ΓΡΑΨΕ 'Δώσε έναν αριθμό'

ΔΙΑΒΑΣΕ Στοιχεία[I]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΚΑΛΕΣΕ Υπολόγισε_MO_ΤυπΑπ(Στοιχεία, Πλήθος,
MO, Τυπ_Απόκλιση)

ΚΑΛΕΣΕ Ταξινόμηση(Στοιχεία, Πλήθος)

Διάμεσος ← Υπολογισμός_Διαμέσου(Στοιχεία, Πλήθος)

! Εκτύπωση αποτελεσμάτων

ΓΡΑΨΕ 'ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΤΟΥΣ ', Πλήθος,
'ΑΡΙΘΜΟΥΣ'

ΓΡΑΨΕ 'ΜΕΣΟΣ ΟΡΟΣ:', MO

ΓΡΑΨΕ 'ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ:', Τυπ_Απόκλιση

ΓΡΑΨΕ 'ΔΙΑΜΕΣΟΣ:', Διάμεσος
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Στατιστική

ΔΙΑΔΙΚΑΣΙΑ Υπολόγισε_ΜΟ_ΤυπΑπ(Πίνακας, N, ΜΟ,
ΤυπΑποκλ)

! Υπολογισμός μέσου όρου
! Υπολογισμός τυπικής απόκλισης

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Πίνακας[100], N, I, Άθροισμα, Άθροισμα_2
ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, ΤυπΑποκλ

ΑΡΧΗ

Άθροισμα ← 0

Άθροισμα_2 ← 0

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N

Άθροισμα ← Άθροισμα+ Πίνακας[I]

Άθροισμα_2 ← Άθροισμα_2+Πίνακας[I]^2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ ← Άθροισμα/N

ΤυπΑποκλ ← $T_P(\text{Άθροισμα_2} / N - \text{ΜΟ}^2)$

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Υπολόγισε_ΜΟ_ΤυπΑπ

ΔΙΑΔΙΚΑΣΙΑ Ταξινόμησε(Πίνακας, N)

! Ταξινόμηση των στοιχείων του πίνακα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: I, N1, T, Βοηθητική, Πίνακας[100], N

ΑΡΧΗ

N1 ← N

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

T ← 0

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ N1-1

ΑΝ Πίνακας[I] > Πίνακας[I+1] **ΤΟΤΕ**
 Βοηθητική ← Πίνακας[I]
 Πίνακας[I] ← Πίνακας[I+1]
 Πίνακας[I+1] ← Βοηθητική
 T ← I
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
N1 ← T
ΜΕΧΡΙΣ_ΟΤΟΥ T=0
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Ταξινόμηση

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

ΑΡΧΗ

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

Προγραμματιστικό περιβάλλον Pascal

```
program Statistiki;
```

```
type list=array[1..100] of integer;
```

```
var
```

```
    avg, st_dev:real;
```

```
    i,median,n:integer;
```

```
    a:list;
```

```
{-----}
```

```
procedure sort(var a:list;n:integer);
```

```
{Ταξινόμηση των στοιχείων του πίνακα A}
```

```
var
```

```
    i,n1,t,temp:integer;
```

```
begin
```

```
    n1:=n;
```

```
    repeat
```

```
        t:=0;
```

```
        for i:=1 to n1-1 do
```

```
            if a[i]>a[i+1] then
```

```
                begin
```

```
                    temp:=a[i];
```

```
                    a[i]:=a[i+1];
```

```
                    a[i+1]:=temp;
```

```
                    t:=i;
```

```
                end;
```

```
            n1:=t;
```

```
        until t=0;
```

```
end;
```

```
{-----}
```

```
procedure ave_stdev(a:list;n:integer;
```

```
    var average,stddv:real);
```

{υπολογισμός μέσου όρου και τυπικής απόκλισης}

var

i:integer;
sum, sum_2:real;

begin

sum:=0; sum_2:=0;
for i:=1 to n do
begin
sum:=sum+a[i]; sum_2:=sum_2+sqr(a[i]);
end;
average:=sum/n;
stddev:=sqrt (sum_2/n – sqr(average)) ;

end;

{————— -}

function med(a:list; n:integer):real;

{υπολογισμός της διαμέσου τιμής}

begin

if n mod 2=0 then
med:=(a[n div 2]+a[(n div 2)+1])/2
else
med:=a[(n+1) div 2];
end;

begin {κυρίως πρόγραμμα}

write('ΔΩΣΕ ΤΟΝ ΠΛΗΘΟΣ ΤΩΝ ΑΡΙΘΜΩΝ (<100):');

readln (n);

for i:=1 to n do

begin

write ('ΔΩΣΕ ΤΟΝ ', i:3,'ο ΑΡΙΘΜΟ :');

readln (a[i]);

end;

```

ave_stddev (a,n,avg,st_dev);
qsort(a,1,n);
median:= med(a,n);
writeln ('Μέση τιμή :', avg, 'Τυπική απόκλιση :',
st_dev);

writeln('Διάμεσος τιμή:', median);
end.

```

Προγραμματιστικό περιβάλλον Basic

```

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

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

CLS
PRINT "*** Αποτελέσματα ***"
PRINT " _____ "
PRINT "mx="; mx, "sx="; sx
PRINT "median="; med
END
SUB MeanAndStdDev (z(), m, s)

```

` Υπολογισμός μέσης τιμής και τυπικής απόκλισης

s1 = 0: s2 = 0

n = UBOUND(z)

FOR i = 1 TO n

s1 = s1 + z(i)

s2 = s2 + z(i) * z(i)

NEXT i

m = s1 / n

s = SQR(s2 / n - m * m)

END SUB

FUNCTION Median (x(), n)

IF n MOD 2 = 0 THEN

Median = (x(n / 2) + x (n/2 + 1)) / 2

ELSE

Median = x((n + 1) / 2)

END IF

END FUNCTION

SUB Sort (x(), n) STATIC

k = n

DO

t = 0

FOR i = 1 TO k - 1

IF x(i) > x(i + 1) THEN

SWAP x(i), x(i + 1)

t = i

END IF

NEXT i

k = t

LOOP UNTIL t = 0

END SUB

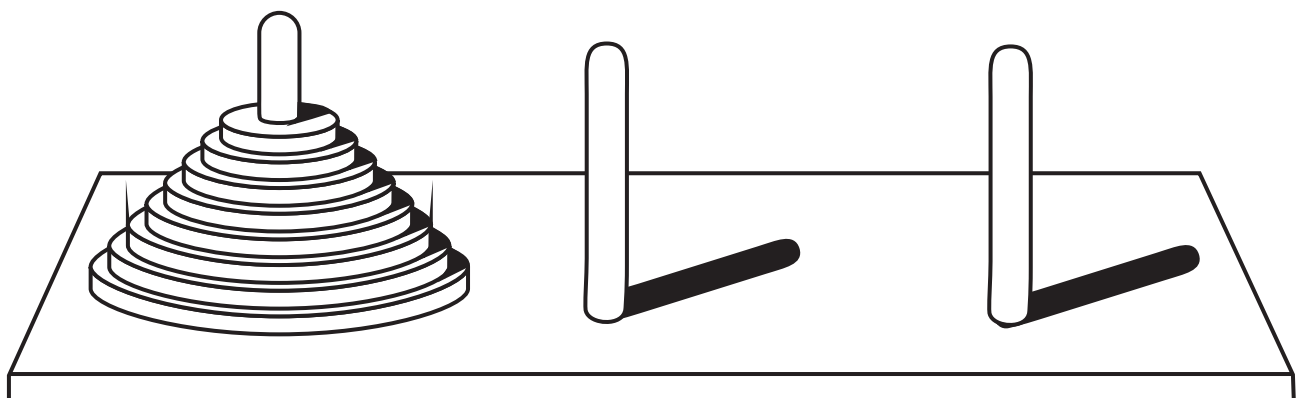
ΠΑΡΑΔΕΙΓΜΑ 2

* * *

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

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

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



- Όταν ένας δίσκος μεταφέρεται, πρέπει να τοποθετηθεί σε έναν από τους τρεις στύλους.
- Μόνο ένας δίσκος μπορεί να μεταφερθεί κάθε φορά και πρέπει να βρίσκεται στην κορυφή του στύλου.
- Ένας μεγαλύτερος δίσκος δεν πρέπει να τοποθετηθεί πάνω από ένα μικρότερο.

Ιστορικό σημείωμα

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

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

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

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

- Αν υπάρχουν δύο δίσκοι, τότε χρειάζονται τρεις απλές κινήσεις:

Ο πρώτος δίσκος από το Στύλο1 μεταφέρεται στο Στύλο2.

Ο δεύτερος δίσκος από το Στύλο2 μεταφέρεται στο Στύλο3.

Ο δίσκος από το Στύλο2 μεταφέρεται στο Στύλο3.

Υποθέτουμε ότι η λύση υπάρχει για $N-1$ δίσκους,

τότε για N δίσκους η λύση δίνεται αναδρομικά:

- Οι $N-1$ δίσκοι μεταφέρονται από το Στύλο1 στο Στύλο2, χρησιμοποιώντας το Στύλο3 ως βοηθητικό.
- Ο τελευταίος δίσκος, που είναι ο τελευταίος άρα και ο μεγαλύτερος, μεταφέρεται από το Στύλο1 στο Στύλο3.
- Οι $N-1$ δίσκοι μεταφέρονται από το Στύλο2 στο Στύλο3, χρησιμοποιώντας το Στύλο1 ως βοηθητικό.

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

ΠΡΟΓΡΑΜΜΑ Πύργοι_του_Ανόι

ΣΤΑΘΕΡΕΣ

Στύλος1='Α'

Στύλος2='Β'

Στύλος3='Γ'

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:N

ΑΡΧΗ

ΓΡΑΨΕ 'δώσε τον αριθμό των δίσκων'

ΔΙΑΒΑΣΕ N

ΚΑΛΕΣΕ Μετακίνησε(N, Στύλος1, Στύλος2, Στύλος3)

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ Μετακίνησε(N,ΣτύλοςA, ΣτύλοςB,ΣτύλοςΓ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:N

ΧΑΡΑΚΤΗΡΕΣ: ΣτύλοςΑ, ΣτύλοςΒ,ΣτύλοςΓ

ΑΡΧΗ

ΑΝ N=1 ΤΟΤΕ
ΓΡΑΨΕ `Μετακίνησε από τον', ΣτύλοςΑ, `στον',
ΣτύλοςΓ

ΑΛΛΙΩΣ
ΚΑΛΕΣΕ Μετακίνησε(N-1, ΣτύλοςΑ, ΣτύλοςΓ
ΣτύλοςΒ)
ΓΡΑΨΕ `Μετακίνησε από τον', ΣτύλοςΑ, `στον',
ΣτύλοςΓ
ΚΑΛΕΣΕ Μετακίνησε(N-1, ΣτύλοςΒ, ΣτύλοςΑ,
ΣτύλοςΓ)

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Προγραμματιστικό περιβάλλον Pascal

```

program anoi;
const
    st1='Α';
    st2='Β';
    st3='Γ';
var
    n:integer;

procedure move(n:integer;sta,stb,stc:char);
begin
    if n=1 then
    begin
        writeln(`ΜΕΤΑΚΙΝΗΣΕ ΑΠΟ `,sta,` ΣΤΟ `,stc)
    end
    else
    begin
        move(n-1,sta,stc,stb);
    end
end

```

```

        writeln ('ΜΕΤΑΚΙΝΗΣΕ ΑΠΟ ', sta, ' ΣΤΟ ',stc);
        move (n-1, stb, sta, stc)
    end;
end;

begin
    write ('ΑΡΙΘΜΟΣ ΔΙΣΚΩΝ :'); readln(n);
    move (n, st1, st2, st3);
end.

```

Προγραμματιστικό περιβάλλον Basic

```

DECLARE SUB Move (n!, a$, b$, c$)
` Anoi
INPUT "N=", n
CALL Move(n, "A", "B", "C")
END

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

```

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

Μετακίνησε από τον Α στον Β

Μετακίνησε από τον Α στον Γ

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

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

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

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

Για τους 64 δίσκους που απασχολεί τους μοναχούς χρειάζονται $1,845 \times 10^{19}$, δηλαδή ένας αριθμός με 20 ψηφία. Πόσο μεγάλος είναι ένας τέτοιος αριθμός; Αν υποθέσουμε ότι εκτελούμε χίλιες κινήσεις το δευτερόλεπτο

(όσες περίπου μπορεί να εκτελέσει ένας γρήγορος υπολογιστής), τότε χρειάζονται περίπου μισό εκατομμύριο χρόνια!!!

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

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



10.3. Συμβουλές - υποδείξεις

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

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

τους αλγορίθμους για το κάθε υποπρόγραμμα και στη συνέχεια να γράφεις το πρόγραμμα.

- Να μελετάς αν ένα υποπρόγραμμα πρέπει να υλοποιηθεί με διαδικασία ή μπορεί να υλοποιηθεί με συνάρτηση.
- Εξέτασε αν κάποια υποπρογράμματα τα οποία έχεις ήδη γράψει ή υπάρχουν σε έτοιμες βιβλιοθήκες προγραμμάτων μπορούν να χρησιμοποιηθούν. Θα γλιτώσεις χρόνο και κόπο.
- Κάθε υποπρόγραμμα να προσπαθείς να είναι όσο το δυνατόν πιο ανεξάρτητο από τα άλλα. Αυτό σε προφυλάσσει από λάθη στο πρόγραμμά σου και σου επιτρέπει τη χρήση του σε άλλα προγράμματα αργότερα.

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

- Να ορίζεις τον τύπο της συνάρτησης. Οι συναρτήσεις παράγουν μόνο ένα αποτέλεσμα συγκεκριμένου τύπου, ακεραίου, πραγματικού κ.λπ. που πρέπει να ορίζεται.
- Να μην υπάρχουν λάθη στην αντιστοίχιση τυπικών και πραγματικών παραμέτρων. Πρόσεξε ότι οι λίστες πρέπει να περιέχουν τον ίδιο αριθμό παραμέτρων και κάθε τυπική παράμετρος με την αντίστοιχη πραγματική πρέπει να είναι του ιδίου τύπου.

10.4. Δραστηριότητες - ασκήσεις

ΔΤ1.

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

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

ΔΤ2.

Να γράψεις τα υποπρογράμματα που υλοποιούν τα παρακάτω:

- A) Να διαβάζει έναν αριθμό και να επιστρέφει το τετράγωνό του.
- B) Να δέχεται δύο αριθμούς και να επιστρέφει το μικρότερο από δύο αριθμούς.
- Γ) Να δέχεται την τιμή ενός προϊόντος και να υπολογίζει

και να τυπώνει την αξία του ΦΠΑ.

Δ) Να ελέγχει αν ένας αριθμός είναι άρτιος.

ΔΤ3.

Να σημειώσεις, στο τετράδιό σου, όλα τα βήματα για τον υπολογισμό του $4!$, τόσο με τη χρήση επαναληπτικής διαδικασίας όσο και με τη χρήση αναδρομικής, σύμφωνα με τα προγράμματα που δίνονται στο βιβλίο σου.

ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ

Στο προγραμματιστικό περιβάλλον του εργαστηρίου του σχολείου σας:

ΔΕ1.

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

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

ΔΕ2.

Να εκτελέσεις το πρόγραμμα του παραδείγματος 1.

ΔΕ3.

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

Για τους μιγαδικούς αριθμούς $a+βi$ και $γ+δi$ έχουμε:

$$(a+βi)+(γ+δi) = (a+γ)+(β+δ)i$$

$$(\alpha + \beta i) - (\gamma + \delta i) = (\alpha - \gamma) + (\beta - \delta)i$$

$$(\alpha + \beta i) \cdot (\gamma + \delta i) = (\alpha\gamma - \beta\delta) + (\alpha\delta + \beta\gamma)i$$

$$\frac{\alpha + \beta i}{\gamma + \delta i} = \left(\frac{\alpha\gamma + \beta\delta}{\gamma^2 + \delta^2} \right) + \left(\frac{\beta\gamma - \alpha\delta}{\gamma^2 + \delta^2} \right)i$$



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

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

ΔΕ4.

Να ξαναγράψεις το πρόγραμμα της ΔΕ1 χρησιμοποιώντας αναδρομικές συναρτήσεις.

Σύγκρινε τα δύο προγράμματα.

ΣΤΟ ΣΠΙΤΙ

ΔΣ1.

Να γράψεις ένα πρόγραμμα το οποίο διαβάζει την τιμή βιβλίων σε ΕΥΡΩ και μετατρέπει τις τιμές τους σε Δραχμές, Γερμανικά Μάρκα, Γαλλικά Φράγκα και Ιταλικές Λιρέτες. Να χρησιμοποιήσεις για τις μετατροπές τις τρέχουσες ισοτιμίες των νομισμάτων.

ΔΣ2.

Να ξαναγράψεις την άσκηση ΔΣ6 του κεφαλαίου 9, τα αποτελέσματα των αγώνων ομίλου του EuroBasket, χρησιμοποιώντας διαδικασίες και συναρτήσεις.

ΔΣ3.

* *

Να επεκτείνεις το παράδειγμα 1, ώστε να υπολογίζει την επικρατούσα τιμή, δηλαδή την τιμή που εμφανίζεται περισσότερες φορές.

ΔΣ4.

Να γράψεις το πρόγραμμα ΔΕ5 του κεφαλαίου 8 που υπολογίζει τη συνολική χωρητικότητα πυκνωτών και τη συνολική αντίσταση αντιστάσεων με τη χρήση υποπρογραμμάτων.

ΔΣ5.

Να γραφεί πρόγραμμα το οποίο να προσθέτει δύο κλάσματα. Το πρόγραμμα δέχεται τέσσερις ακεραίους αριθμούς, τους παρονομαστές και τους αριθμητές των δύο κλασμάτων, υπολογίζει και εκτυπώνει τον αριθμητή και τον παρονομαστή του αποτελέσματος.

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



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

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

ΔΣ6.

**

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

10.5. Τεστ αυτοαξιολόγησης ●

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

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

2. Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος:

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

3. Διάλεξε ένα μεταξύ των προτεινόμενων:

8. Όταν μία μεταβλητή ισχύει μόνο στο υποπρόγραμμα που ορίστηκε ονομάζεται:
A) Τοπική
B) Καθολική

Γ) Παράμετρος

Δ) Τυπική

9. Τι θα τυπώσουν οι παρακάτω εντολές:

.....

A ← 5

B ← 10

Γ ← 0

ΚΑΛΕΣΕ Διαδ1(A, B)

ΓΡΑΨΕ A,B,Γ

.....

ΔΙΑΔΙΚΑΣΙΑ Διαδ1(Γ,Δ)

.....

ΑΡΧΗ

Γ ← Γ-Δ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Διαδ1

A) 5,10,0 B) 5,10,-5

Γ) -5,10,0 Δ) -5,10,-5

10. Τι θα τυπώσουν οι παρακάτω εντολές:

.....

A ← 5

B ← 10

ΚΑΛΕΣΕ Διαδ1(B, A)

ΓΡΑΨΕ A,B

.....

ΔΙΑΔΙΚΑΣΙΑ Διαδ1(A,B)

.....

ΑΡΧΗ

ΓΡΑΨΕ A,B

$A \leftarrow A - B$

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ Διαδ1

A) 5,10

5,10

B) 10,5

5,5

Γ) 5,10

-5,10

Δ) 10,5

5,10

4. Διάλεξε όλα όσα χρειάζονται μεταξύ των προτεινόμενων:

11. Ο ορισμός κάθε αναδρομικού υποπρογράμματος περιλαμβάνει:

A) Αναδρομική σχέση

B) Τιμή βάσης

Γ) Υπολογισμό παραγοντικού

Δ) Επαναληπτική διαδικασία

E) Καθολικές μεταβλητές

12. Μερικά από τα πλεονεκτήματα του τμηματικού προγραμματισμού είναι:

A) Λιγότερος χρόνος για την ανάπτυξη του προγράμματος

B) Ευκολότερη διόρθωση

Γ) Ταχύτητα κατά την εκτέλεση

Δ) Χρήση αναδρομικών διαδικασιών



Απαντήσεις στα τεστ αυτοαξιολόγησης

ΚΕΦΑΛΑΙΟ 6

1. πηγαίο
2. ιεραρχική
3. οπτικό, οδηγούμενο από τα γεγονότα

2. 4. Σωστό
5. Λάθος
6. Λάθος
7. Λάθος

3. 8. Α
9. Β
10. Γ
11. Α

4. 12. Β, Γ
13. Α, Β, Γ, Δ
14. Α, Δ

ΚΕΦΑΛΑΙΟ 7

1. 1. Γ, Β, Α, Δ
2. Β, Γ, Δ, Α

2. 3. μεταβλητές
4. ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ όνομα προγράμματος

3. 5. Σωστό

6. Λάθος

7. Σωστό

8. Σωστό

4. 9. Δ

10. Β

11. Α

12. Β

5. 13. Α, Β, Δ, Ζ

14. Α, Β, Ζ

ΚΕΦΑΛΑΙΟ 8

1. 1. Β, Ε, Δ, Α, Γ

2. Γ, Δ, Β, Α

2. 3. Λάθος

4. Λάθος

5. Σωστό

6. Λάθος

7. Σωστό

3. 8. Β

9. Γ

10. Δ

11. Β

12. Γ

13. Β

ΚΕΦΑΛΑΙΟ 9

- 1.** 1. μονοδιάστατοι
2. στατική
3. διαγωνίου
- 2.** 4. Λάθος
5. Σωστό
6. Λάθος
7. Σωστό
- 3.** 8. Γ
9. Β
10. Α
- 4.** 11. Α, Δ, Ε
12. Γ, Δ

ΚΕΦΑΛΑΙΟ 10

- 1.** 1. τυπικών
2. διαδικασίες, συναρτήσεις
3. απεριόριστη
- 2.** 4. Λάθος
5. Σωστό
6. Λάθος
7. Λάθος
- 3.** 8. Α
9. Γ
10. Β
- 4.** 11. Α, Β
12. Α, Β



Περιεχόμενα

6. Εισαγωγή στον Προγραμματισμό	5
7. Βασικές Έννοιες Προγραμματισμού	13
8. Επιλογή και Επανάληψη	31
9. Πίνακες	61
10. Υποπρογράμματα	83
Απαντήσεις στα τεστ αυτοαξιολόγησης	109

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

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