

Βασικές έννοιες αλγορίθμων – Εντολές και στοιχεία της Γλώσσας

Αλγόριθμος : Είναι ένα μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων κι εκτελέσιμων σε πεπερασμένο χρόνο, που οδηγούν στην επίλυση ενός προβλήματος.

Τρόποι αναπαράστασης ενός αλγορίθμου

- 1) **Με ελεύθερο κείμενο :** Αποτελεί έναν αδόμητο τρόπο περιγραφής, που μπορεί να καταστρατηγήσει εύκολα τα κριτήρια του σωστού αλγορίθμου (ασάφειες, μη αποτελεσματικότητα κλπ)
- 2) **Με διαγραμματικές τεχνικές :** Αποτελεί έναν γραφικό τρόπο παρουσίασης των εντολών. Η πιο γνωστή τεχνική είναι το **διάγραμμα ροής** (flow chart) που παλαιότερα ήταν αρκετά δημοφιλής.
- 3) **Με φυσική γλώσσα κατά βήματα:** Χρειάζεται προσοχή διότι μπορεί να παραβιασθεί το κριτήριο της καθοριστικότητας. Δεν είναι ασφαλής τρόπος.
- 4) **Με κωδικοποίηση:** Στην περίπτωση αυτή, δημιουργούμε τις εντολές κατευθείαν σε μία γλώσσα προγραμματισμού ή σε μια ψευδογλώσσα.

Στα παρακάτω παραδείγματα χρησιμοποιούμε μία κωδικοποίηση σε μία (μη υπαρκτή) δομημένη γλώσσα προγραμματισμού, καλούμενη ως **ψευδογλώσσα**. Φτιάχνοντας τον αλγόριθμο με ψευδογλώσσα, μπορούμε, κατόπιν, να τον μετατρέψουμε χωρίς πολλές μετατροπές σε μία υπαρκτή γλώσσα.

Βασικά στοιχεία της ΓΛΩΣΣΑΣ.

1. Το αλφάβητο της ΓΛΩΣΣΑΣ

Το αλφάβητο αποτελείται από

- ◆ Γράμματα
- ◆ Κεφαλαία και μικρά ελληνικού αλφαβήτου (Α-Ω , α-ω)
- ◆ Κεφαλαία και μικρά αγγλικού αλφαβήτου (Α-Z, a-z)
- ◆ Ψηφία 0-9
- ◆ Ειδικούς χαρακτήρες: + - * / ^ = > < , . ! & [] () _ και ο κενός χαρακτήρας

Τύποι δεδομένων της ΓΛΩΣΣΑΣ

- ◆ **Ακέραιος.** Ο τύπος αυτός περιλαμβάνει τους ακέραιους (το σύνολο Z) που είναι γνωστοί από τα μαθηματικά. (δηλαδή θετικούς και αρνητικούς καθώς και το 0)
- ◆ **Πραγματικός.** Ο τύπος αυτός περιλαμβάνει τους πραγματικούς (το σύνολο R) που είναι γνωστοί από τα μαθηματικά.
(δηλαδή αριθμούς με δεκαδικό μέρος θετικούς και αρνητικούς καθώς και το 0)
- ◆ **Λογικός.** Ο τύπος αυτός δέχεται **μόνο** δύο τιμές **ΑΛΗΘΗΣ** και **ΨΕΥΔΗΣ**.
- ◆ **Χαρακτήρας.** Ο τύπος αυτός αναφέρεται τόσο σε ένα χαρακτήρα όσο και σε μία σειρά χαρακτήρων. Περιέχει οποιοδήποτε χαρακτήρα μπορεί να γραφεί στο πληκτρολόγιο. Οι χαρακτήρες πρέπει να βρίσκονται υποχρεωτικά ανάμεσα σε μονά εισαγωγικά. Επειδή μπορεί να περιέχει και αριθμούς ονομάζεται συχνά και αλφαριθμητικό
{Πχ 'α' 'αλλά%' '2a!@βγ..όλα-23' }

Σταθερές : Είναι ποσότητες που δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του αλγορίθμου.

Υπάρχουν 3 ειδών σταθερές:

- ◆ **Αριθμητικές** π.χ. 123, -8, 3.14
- ◆ **Αλφαριθμητικές (ή χαρακτήρες)** π.χ. "Γιάννης", "Μακεδονίας 12" (λέγονται και συμβολοσειρές και περικλείονται σε μονά εισαγωγικά)
- ◆ **Λογικές** π.χ. ΑΛΗΘΗΣ, ΨΕΥΔΗΣ.

Μεταβλητές : Είναι ποσότητες που μπορούν να μεταβάλλονται κατά τη διάρκεια εκτέλεσης του αλγορίθμου.

Κάθε μεταβλητή, αντιστοιχεί σε μια περιοχή της μνήμης του ΗΥ.

Η τιμή της μεταβλητής αποθηκεύεται σε αυτήν την περιοχή μνήμης.

Κατά την διάρκεια εκτέλεσης ενός αλγορίθμου η τιμή μιας μεταβλητής μπορεί να αλλάξει, **δεν αλλάζει όμως ο τύπος της**.

Στις μεταβλητές δίνουμε ένα όνομα (αναγνωριστικό) και αυτό χρησιμοποιούμε στον αλγόριθμο.

Ονόματα-Κανόνες δημιουργίας

Κάθε προγράμμα όπως και τα δεδομένα (μεταβλητές και σταθερές) που χρησιμοποιεί έχουν ένα όνομα με το οποίο αναφερόμαστε σε αυτό.

Τα ονόματα αυτά μπορούν να αποτελούνται από

- ◆ Γράμματα: Κεφαλαία και μικρά ελληνικού αλφαβήτου (Α-Ω , α-ω)Κεφαλαία και μικρά αγγλικού αλφαβήτου (Α-Z, a-z)
- ◆ Ψηφία: 0-9
- ◆ Ειδικούς χαρακτήρες: _ (κάτω παύλα (στα αγγλικά underscore))

Ενώ

- ◆ **Απαγορεύεται** η χρήση δεσμευμένων λέξεων ως ονόματα μεταβλητών (πχ γράψε, διάβασε)
- ◆ Τα ονόματα μεταβλητών **υποχρεωτικά αρχίζουν από γράμμα** (όχι ψηφίο ή underscore)

Μπορούμε να δώσουμε τιμή σε μια μεταβλητή με δύο τρόπους

- ◆ **α)** με μία **εντολή εκχώρησης τιμής**
π.χ. $X \leftarrow 123$ $\text{Name} \leftarrow \text{'Γιάννης'}$ ή
- ◆ **β)** με την **εντολή εισόδου Διάβασε** (οπότε έχουμε είσοδο δεδομένων από το πληκτρολόγιο)
π.χ. Διάβασε X Διάβασε Name



Τύποι μεταβλητών

Κι εδώ, ανάλογα με το είδος δεδομένου που εκχωρείται σε μια μεταβλητή, οι μεταβλητές διακρίνονται σε :

- ◆ Αριθμητικές π.χ. $X \leftarrow 123$
- ◆ Αλφαριθμητικές (ή χαρακτήρες) π.χ. $\text{Name} \leftarrow \text{'Γιάννης'}$
- ◆ Λογικές π.χ. $\text{Συνταξιούχος} \leftarrow \text{ΨΕΥΔΗΣ}$.

Επιπλέον, οι αριθμητικές διακρίνονται σε

- ◆ Ακέραιες , αν δέχονται ακέραια τιμή π.χ. 5 , -12
- ◆ Πραγματικές, αν δέχονται πραγματική τιμή π.χ. 3.14 , -4.56

Τελεστές. Είναι τα σύμβολα που χρησιμοποιούνται για τις διάφορες πράξεις. Οι τελεστές είναι 3 ειδών:

- ◆ Αριθμητικοί : + (πρόσθεση), - (αφαίρεση), * (πολλ/σμός), / (διαίρεση), ^ (ύψωση σε δύναμη), **div** (πηλίκιο ακέραιας διαίρεσης), **mod** (υπόλοιπο ακέραιας διαίρεσης).
Π.χ. $X \leftarrow 5 * 2 (=10)$,
 $X \leftarrow 6^2 (=36)$,

$X \leftarrow 5 \text{ div } 2$ (=2 πηλίκο),
 $X \leftarrow 5 \text{ mod } 2$ (=1 υπόλοιπο)

- ◆ Λογικοί : OXI, ΚΑΙ, Η . Π.χ. Έγκυρος $\leftarrow (X \geq 1)$ ΚΑΙ $(X \leq 20)$
- ◆ Συγκριτικοί : > , >= , < , <= , <> (διάφορο)

Ιεραρχία αριθμητικών τελεστών

Η ιεραρχία είναι:

- ◆ ^
- ◆ * / div mod
- ◆ + -

Προσοχή:

Αν συναντάμε παρενθέσεις προηγούνται οι τελεστές εντός των παρενθέσεων.

Όσοι τελεστές έχουν ίδια ιεραρχία εκτελούνται με την σειρά που τους συναντάμε από αριστερά προς τα δεξιά (όπως στα μαθηματικά).

Ιεραρχία λογικών τελεστών

- ◆ OXI
- ◆ ΚΑΙ
- ◆ Η

Συναρτήσεις της ΓΛΩΣΣΑΣ

Η γλώσσα περιέχει 8 εγγενείς συναρτήσεις

Οι κυριότερες είναι :

$HM(x)$, $SYN(x)$, $E\Phi(x)$, $T_P(x)$,
ημίτονο του x συνημίτονο του x εφαπτόμενη του x τετραγωνική ρίζα του x

$A_M(x)$, $A_T(x)$
ακέραιο μέρος του x απόλυτη τιμή του x

Εκφράσεις : Είναι οι παραστάσεις που αποτελούνται από σταθερές, μεταβλητές, τελεστές, παρενθέσεις. Μια έκφραση (στην απλούστερη μορφή της) μπορεί να αποτελείται από μια μόνο μεταβλητή ή μια σταθερά.

Παραδείγματα εκφράσεων: a , 100 , $2*a+b$, $3*(x^2 + T_P(\psi))$, $a > x + 2$ ΚΑΙ $b < \gamma$

Όταν σε μια έκφραση υπάρχουν πολλά είδη τελεστών, οι αριθμητικοί τελεστές προηγούνται από τους συγκριτικούς και τέλος ακολουθούν οι λογικοί.

Επίσης όπως και στα μαθηματικά, γίνονται πρώτα οι υπολογισμοί μέσα στις παρενθέσεις.

Είδη εκφράσεων

Οι εκφράσεις είναι 2 ειδών:

α) **λογικές εκφράσεις** είναι οι εκφράσεις που έχουν (αφού υπολογιστούν – όταν είναι σύνθετες-) ως αποτέλεσμα μια λογική τιμή.

β) **αριθμητικές εκφράσεις** είναι οι εκφράσεις που έχουν (αφού υπολογιστούν – όταν είναι σύνθετες-) ως αποτέλεσμα μια αριθμητική τιμή.

Βασικές εντολές :

- ο Εντολή εισόδου : ΔΙΑΒΑΣΕ . Π.χ. ΔΙΑΒΑΣΕ X . Στην μεταβλητή X εισάγεται μία τιμή.
- ο Εντολή εξόδου : ΓΡΑΨΕ Π.χ. ΓΡΑΨΕ X, Εμφανίζεται η τιμή της μεταβλητής X
ΓΡΑΨΕ 'Η τιμή του X είναι:', X Εμφανίζεται το κείμενο που περιέχεται στα εισαγωγικά και η τιμή της μεταβλητής X
ΓΡΑΨΕ 'Η τιμή του Ψ είναι:', $2*X+3$ Εμφανίζεται το κείμενο που περιέχεται στα εισαγωγικά και η τιμή της έκφρασης $2*X+3$

- ο Εντολή εκχώρησης τιμής : Μορφή : **Μεταβλητή** ← Έκφραση. Αριστερά βάζουμε το όνομα της μεταβλητής και δεξιά μία τιμή ή έκφραση Π.χ. $X \leftarrow (5 * 2) / 100$. Το αποτέλεσμα της έκφρασης $(5 * 2) / 100$ αφού πρώτα υπολογιστεί, εκχωρείται στην μεταβλητή.

Αλγοριθμικές δομές Εννοούμε τον τρόπο που θέτουμε τα βήματα (εντολές) στον αλγόριθμο ώστε να έχουμε σωστή και αυστηρή σύνταξη (δομημένη) και να πληρούνται τα χαρακτηριστικά του.

Οι αλγοριθμικές δομές είναι 3 :

Ακολουθιακή δομή: Οι εντολές εκτελούνται η μία μετά την άλλη.

Δομή Επιλογής : Μία ομάδα εντολών εκτελούνται ανάλογα με την τιμή μίας συνθήκης (απόφαση).

Δομή Επανάληψης : Μία ομάδα εντολών εκτελούνται συνεχώς επαναληπτικά ανάλογα με την τιμή μίας συνθήκης (επαναληπτικός βρόχος –Loop-).

Δομή Επιλογής

Υπάρχουν τρία είδη :

Η απλή

```

ΑΝ συνθήκη ΤΟΤΕ
    Εντολές
ΤΕΛΟΣ_ΑΝ
  
```

Η σύνθετη

```

ΑΝ συνθήκη ΤΟΤΕ
    Εντολές1
ΑΛΛΙΩΣ
    Εντολές2
ΤΕΛΟΣ_ΑΝ
  
```

Η πολλαπλή επιλογή

```

ΑΝ συνθήκη1 ΤΟΤΕ
    Εντολές1
ΑΛΛΙΩΣ_ΑΝ συνθήκη2 ΤΟΤΕ
    Εντολές2
ΑΛΛΙΩΣ_ΑΝ συνθήκη3 ΤΟΤΕ
    Εντολές3
    .....
ΑΛΛΙΩΣ
    ΕντολέςN
ΤΕΛΟΣ_ΑΝ
  
```

- Στο **πρώτο είδος**, οι εντολές εκτελούνται μόνο αν η συνθήκη είναι ΑΛΗΘΗΣ
- Στο **δεύτερο είδος**, αν η συνθήκη είναι ΑΛΗΘΗΣ τότε εκτελείται η ομάδα των Εντολών1 ενώ αν είναι ΨΕΥΔΗΣ εκτελείται η ομάδα των Εντολών2
- Στο **τρίτο είδος**, εκτελείται μόνο ΜΙΑ ΟΜΑΔΑ ΕΝΤΟΛΩΝ ανάλογα με το ποια συνθήκη αληθεύει.

Παράδειγμα πολλαπλής επιλογής: Να γραφεί ΠΡΟΓΡΑΜΜΑ που διαβάζει το βαθμό ενός μαθητή και τυπώνει το χαρακτηρισμό του (π.χ. αν διαβάσει το 15 να τυπώσει «Καλά»).

ΠΡΟΓΡΑΜΜΑ Χαρακτηρισμός_βαθμού
ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: ΒΑΘ *!ο βαθμός που θα διαβαστεί.*

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε βαθμολογία'

ΔΙΑΒΑΣΕ ΒΑΘ

ΑΝ ΒΑΘ<10 **ΤΟΤΕ**

ΓΡΑΨΕ "Κακώς"

ΑΛΛΙΩΣ_ΑΝ ΒΑΘ<12,5 **ΤΟΤΕ**

ΓΡΑΨΕ “Μέτρια”
ΑΛΛΙΩΣ_ΑΝ ΒΑΘ<15,5 **ΤΟΤΕ**
ΓΡΑΨΕ “Καλά”
ΑΛΛΙΩΣ_ΑΝ ΒΑΘ <18,5 **ΤΟΤΕ**
ΓΡΑΨΕ “Πολύ καλά”
ΑΛΛΙΩΣ
ΓΡΑΨΕ “Αριστα”
ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Χαρακτηρισμός_βαθμού

Εμφωλευμένες δομές ελέγχου

Πρόκειται για τις περιπτώσεις που έχουμε μία δομή Αν...τότε.. μέσα σε μία άλλη. Π.χ.

ΑΝ συνθήκη1 **ΤΟΤΕ**

Εντολές1

ΑΛΛΙΩΣ

ΑΝ συνθήκη2 **ΤΟΤΕ**

Εντολές2

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

} εμφωλευμένη

Επαναληπτική δομή

Υπάρχουν τρία είδη :

ΌΣΟ συνθήκη **ΕΠΑΝΑΛΑΒΕ**

Εντολές

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

Εντολές

ΜΕΧΡΙΣ_ΟΤΟΥ συνθήκη

1ο είδος

2ο είδος

ΓΙΑ μεταβλητή **ΑΠΟ** τιμή1 **ΜΕΧΡΙ** τιμή2 **ΜΕ ΒΗΜΑ** β

Εντολές

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

3ο είδος

Στο πρώτο είδος, Όσο <συνθήκη> επανάλαβε , οι εντολές εκτελούνται συνεχώς, όσο η συνθήκη είναι ΑΛΗΘΗΣ. Μόλις η συνθήκη γίνει ΨΕΥΔΗΣ τότε σταματάει η επανάληψη. Οι εντολές μπορεί να εκτελεστούν καμία, μία ή περισσότερες φορές. Αν η συνθήκη βρεθεί από την αρχή ψευδής, οι εντολές δεν εκτελούνται καμία φορά.

Η εντολή **Όσο..επανάλαβε...**: χρησιμοποιείται όταν δεν γνωρίζουμε τον αριθμό των επαναλήψεων (Μπορεί όμως να χρησιμοποιηθεί και όταν γνωρίζουμε τον αριθμό των επαναλήψεων).

Παράδειγμα 1 : (επαναληπτική είσοδος αγνώστου πλήθους στοιχείων)

Να γραφεί ΠΡΟΓΡΑΜΜΑ που διαβάζει το όνομα και το ποσό που έδωσε κάθε μαθητής σε έναν έρανο. Σταματάει όταν δοθεί για όνομα το !. Το πρόγραμμα υπολογίζει το σύνολο των χρημάτων που συγκεντρώθηκαν και τον μέσο όρο.

ΠΡΟΓΡΑΜΜΑ Άθροισμα_αριθμών

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: X, i !ο αριθμός που διαβάζεται, ο μετρητής
ΑΚΕΡΑΙΕΣ: ΑΘΡ !εδώ θα αποθηκεύεται το άθροισμα
ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ !εδώ θα αποθηκεύεται ο μέσος όρος
ΧΑΡΑΚΤΗΡΕΣ : ΟΝ ! Εδώ θα αποθηκευθεί το όνομα

ΑΡΧΗ

i ← 0 !Αρχικοποίηση ΜΕΤΡΗΤΗ
ΑΘΡ ← 0 !Αρχικοποίηση για το άθροισμα
ΔΙΑΒΑΣΕ ΟΝ !ο πρώτος διαβάζεται πριν μπει στην επανάληψη
ΟΣΟ ΟΝ <> '!' **ΕΠΑΝΑΛΑΒΕ**
 ΔΙΑΒΑΣΕ X
 i ← i+1 ! Αύξηση μετρητή
 ΑΘΡ ← ΑΘΡ + X !πρόσθεσέ τον στο άθροισμα
 ΔΙΑΒΑΣΕ ΟΝ !διάβασε τον επόμενο

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το άθροισμα είναι', ΑΘΡ

ΑΝ i <> 0 **ΤΟΤΕ**
 ΜΟ ← ΑΘΡ / i
 ΓΡΑΨΕ 'Ο μέσος όρος είναι ', ΜΟ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δεν δόθηκαν στοιχεία'
ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Άθροισμα_αριθμών

Παράδειγμα 2 με επαναληπτική δομή :

Να γραφεί ΠΡΟΓΡΑΜΜΑ που διαβάζει ΘΕΤΙΚΟΥΣ αριθμούς και υπολογίζει το μεγαλύτερο τους. Να σταματά όταν δοθεί ο αριθμός 0.

ΠΡΟΓΡΑΜΜΑ ΜΕΓΑΛΥΤΕΡΟΣ_ΑΡΙΘΜΟΣ

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: X !ο αριθμός που διαβάζεται
ΠΡΑΓΜΑΤΙΚΕΣ: MAX !εδώ θα αποθηκεύεται ο μέγιστος αριθμός
ΑΚΕΡΑΙΕΣ: i !ο μετρητής

ΑΡΧΗ

MAX ← -1 !αρχικοποιείται ο μέγιστος σε μια πολύ μικρή τιμή που η μεταβλητή δεν μπορεί να πάρει
i ← 0 !ο μετρητής αρχικοποιείται πριν ξεκινήσει η επανάληψη
ΔΙΑΒΑΣΕ X !ο πρώτος διαβάζεται πριν μπει στην επανάληψη

ΟΣΟ X <> 0 **ΕΠΑΝΑΛΑΒΕ**

i ← i + 1 *! αυξάνουμε τον μετρητή*

ΑΝ X > MAX **ΤΟΤΕ**

MAX ← X

ΘΕΣΗ_MAX ← i

ΤΕΛΟΣ_ΑΝ

ΔΙΑΒΑΣΕ X *! διάβασε τον επόμενο*

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Ο ΜΕΓΑΛΥΤΕΡΟΣ_ΑΡΙΘΜΟΣ είναι ο ', MAX, ' και είναι στη θέση ', ΘΕΣΗ_MAX

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΕΓΑΛΥΤΕΡΟΣ_ΑΡΙΘΜΟΣ

Να σημειωθεί ότι στις περισσότερες περιπτώσεις (και πάντα στο 3ο είδος) χρειαζόμαστε μία **μεταβλητή-μετρητή**. Ο μετρητής (συνήθως τον ονομάζουμε i) μετρά τον αριθμό των επαναλήψεων.

Στο **δεύτερο είδος**, **Αρχή_επανάληψης, ... Μέχρις_ότου**, εκτελείται πρώτα η ομάδα των εντολών και μετά ελέγχεται η συνθήκη. Συνεπώς οι **εντολές εκτελούνται τουλάχιστον μία φορά πάντα** (μία ή περισσότερες φορές).

Εδώ, η επανάληψη εκτελείται όσο η συνθήκη είναι ΨΕΥΔΗΣ. Μόλις η συνθήκη γίνει ΑΛΗΘΗΣ σταματάει η επανάληψη. (το αντίθετο με την Όσο ... επανάλαβε).

Έλεγχος ορθότητας

Να γραφεί τμήμα προγράμματος που διαβάζει το φύλο ενός μαθητή (Α για αγόρι, Κ για κορίτσι) και να γίνεται έλεγχος ορθότητας δεδομένων

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Φ *! Το φύλο που διαβάζεται*

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Παρακαλώ δώστε το φύλο. (Α για αγόρι, Κ για κορίτσι)'

ΔΙΑΒΑΣΕ Φ *! διάβασε το φύλο*

ΜΕΧΡΙΣ_ΌΤΟΥ Φ = 'Α' Ή Φ = 'Κ'

Το **τρίτο είδος**, **Για ... από μέχρι**, ενδείκνυται όταν γνωρίζουμε τον αριθμό των επαναλήψεων.

Χρησιμοποιεί απαραίτητα μετρητή.

Η μορφή αυτή ενσωματώνει :

- ❖ Την αρχικοποίηση του μετρητή.
- ❖ Τον έλεγχο της συνθήκης, αν φτάσαμε στην τελική τιμή.
- ❖ Την αύξηση / μείωση του μετρητή. Αν δεν καθορίσουμε βήμα τότε εννοείται βήμα 1.

Παράδειγμα: Να γραφεί αλγόριθμος που διαβάζει 10 ακέραιους αριθμούς και υπολογίζει το άθροισμά τους και το μέσο όρο τους.

ΠΡΟΓΡΑΜΜΑ Άθροισμα_αριθμών

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: X, i *! ο αριθμός που διαβάζεται, ο μετρητής*

ΑΚΕΡΑΙΕΣ: SUM *! εδώ θα αποθηκεύεται το άθροισμα*

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ *! εδώ θα αποθηκεύεται ο μέσος όρος*

ΑΡΧΗ

SUM ← 0 *!Αρχικοποίηση για το άθροισμα*

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΔΙΑΒΑΣΕ X *!κάνε εισαγωγή του αριθμού στη X*

SUM ← SUM + X *!πρόσθεσέ τον στο άθροισμα*

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ ‘Το άθροισμα είναι ‘, SUM

MO ← SUM / 10

ΓΡΑΨΕ ‘Ο μέσος όρος είναι ‘, MO

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Άθροισμα_αριθμών

Σύνδεση λογικών προτάσεων

Στις δομές ελέγχου και στις επαναληπτικές δομές χρησιμοποιούμε <συνθήκη> για να καθοδηγήσουμε τον Η/Υ προς ένα ή άλλο μονοπάτι εκτέλεσης εντολών. Η <συνθήκη> είναι μία λογική πρόταση που δίνει αποτέλεσμα ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ και μπορεί να είναι :

Απλή : π.χ. $i < 10$ ή

Σύνθετη : π.χ. $M \geq 18,5$ ΚΑΙ $M \leq 20$

Στην δεύτερη περίπτωση, χρησιμοποιούμε λογικές προτάσεις που συνδέονται μεταξύ τους με τους λογικούς τελεστές **ΚΑΙ** ή **Ή**. Επίσης, χρησιμοποιούμε και το **ΟΧΙ** (όχι) για την αντιστροφή της τιμής μίας πρότασης.

πρόταση A	πρόταση B	A Ή B	A ΚΑΙ B	ΟΧΙ A
ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ
ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ
ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ
ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ

Γενικά, στην περίπτωση του **Ή**, όταν μία πρόταση είναι αληθής (ΑΛΗΘΗΣ) τότε βγαίνει συνολικά αληθής. Στην περίπτωση του **ΚΑΙ**, όταν μία πρόταση είναι ψευδής (ΨΕΥΔΗΣ) τότε βγαίνει συνολικά ψευδής.

Δομή δεδομένων :

Είναι ένας τρόπος αποθήκευσης (οργάνωσης) δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.

Κάθε δομή αποτελείται από ένα σύνολο κόμβων.

Κάθε κόμβος περιέχει μία απλή τιμή (αριθμό, αλφαριθμητικό κλπ) ή μία πιο σύνθετη τιμή (π.χ. μία εγγραφή)

Κατηγορίες δομών δεδομένων

Στατικές

Το μέγεθός τους είναι καθορισμένο (σταθερό) και δεν μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος.

Οι κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.

Δυναμικές

Το μέγεθός τους δεν είναι καθορισμένο (δεν είναι σταθερό) και μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. Μπορούν να εισάγονται νέοι κόμβοι και να διαγράφονται υπάρχοντες.

Οι κόμβοι δεν αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.

Πίνακες

Χαρακτηριστικός εκπρόσωπος των **στατικών** δομών είναι οι πίνακες.

Γενικά ένας πίνακας είναι ένα σύνολο στοιχείων του **ιδίου τύπου** (δηλαδή ακέραιων, πραγματικών κλπ) τα οποία αναφέρονται με κοινό όνομα. (π.χ. Πίνακας Α) Το **μέγεθός** του πίνακα είναι καθορισμένο (**σταθερό**) και δεν μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος. Τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.

Πίνακας Π

12	-4	33	45	8
1	2	3	4	5

Πίνακας Α

'Γιάννης'	'Μαρία'	'Άννα'	'Νίκος'
1	2	3	4

Στο σχήμα οι αριθμοί κάτω από τις θέσεις του πίνακα δηλώνουν τον αριθμό θέσης του πίνακα.

Όταν ο πίνακας είναι μονοδιάστατος (δηλαδή πίνακας μίας γραμμής ή μία στήλης), τότε η αναφορά σε κάθε στοιχείο του πίνακα, δηλαδή στο περιεχόμενο μία θέσης του πίνακα, γίνεται από το όνομα του πίνακα, ακολουθούμενο από έναν δείκτη μέσα σε αγκύλες

Για παράδειγμα, με το Π [1] δηλώνουμε το περιεχόμενο της θέσης 1 του πίνακα Π. Οπότε μπορούμε να πούμε

Π [1] = 12 (Το πρώτο στοιχείο του πίνακα Π έχει τιμή 12)

Π [2] = -4 (Το δεύτερο στοιχείο του πίνακα Π έχει τιμή -4)

Π [4] = 45 (Το τέταρτο στοιχείο του πίνακα Π έχει τιμή 45) κ.λ.π.

Ομοίως για τον πίνακα Α έχουμε

Α [1] = 'Γιάννης' (Το πρώτο στοιχείο του πίνακα Α έχει τιμή 'Γιάννης')

Α [2] = 'Μαρία' (Το δεύτερο στοιχείο του πίνακα Α έχει τιμή 'Μαρία') κ.λ.π.

Τυπικές επεξεργασίες (λειτουργίες) σε έναν πίνακα.

- > **Διάβαση** των στοιχείων του πίνακα, δηλαδή εισαγωγή τιμών στις θέσεις του.
- > **Εκτύπωση** των στοιχείων του πίνακα
- > Υπολογισμός του **αθροίσματος** των στοιχείων του.
- > Υπολογισμός του **μέσου όρου** των στοιχείων του.
- > Εύρεση του **ελάχιστου ή μέγιστου** στοιχείου του.
- > **Αναζήτηση** ενός στοιχείου.
- > **Ταξινόμηση** του πίνακα.

> **Συγχώνευση** δύο πινάκων

Στους αλγόριθμους του πίνακα χρησιμοποιούμε, ως επί το πλείστον, τη δομή Για...από..μέχρι...

Αλγόριθμοι μονοδιάστατου πίνακα

Διάβαση στοιχείων (δηλ. εισαγωγή στοιχείων στις θέσεις του πίνακα) και εκτύπωση

ΠΡΟΓΡΑΜΜΑ Διάβαση_ Εκτύπωση_στοιχείων
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π [10] !10 θέσεις
ΑΚΕΡΑΙΕΣ: i !ο δείκτης θέσης

ΑΡΧΗ

! Διάβαση στοιχείων

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10
 ΔΙΑΒΑΣΕ Π [i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Εκτύπωση στοιχείων

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10
 ΓΡΑΨΕ Π [i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Διάβαση_ Εκτύπωση_στοιχείων

Εύρεση μέγιστου στοιχείου

ΠΡΟΓΡΑΜΜΑ Max_στοιχείων

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π [10] !10 θέσεις
ΑΚΕΡΑΙΕΣ: i !ο δείκτης θέσης
ΑΚΕΡΑΙΕΣ: Max

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10
 ΔΙΑΒΑΣΕ Π [i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Max ← Π [1] *!Θέτουμε σαν αρχική τιμή στο Max το πρώτο στοιχείο του πίνακα*
ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 10 *!Ξεκινάμε από τη 2^η θέση*
 ΑΝ Π [i] > Max **ΤΟΤΕ** *!αν το i στοιχείο είναι μεγαλύτερο από την τρέχουσα τιμή του Max τότε*
 Max ← Π [i] *!βάλε στο Max το i στοιχείο*

ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το μέγιστο στοιχείο του πίνακα είναι το ', Max
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Max_στοιχείων

Η εύρεση του ελάχιστου είναι παρόμοια. Αλλάζει η ανισότητα (δηλαδή ελέγχουμε Π[i] < Min).

Υπολογισμός αθροίσματος και μέσου όρου (ΜΟ) στοιχείων

ΠΡΟΓΡΑΜΜΑ ΜΟ_στοιχείων

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π [10] *!10 θέσεις*

ΑΚΕΡΑΙΕΣ: i *!ο δείκτης θέσης*

ΑΚΕΡΑΙΕΣ: ΑΘΡ

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΔΙΑΒΑΣΕ Π [i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΘΡ ← 0 *!αρχικοποίηση*

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΑΘΡ ← ΑΘΡ + Π [i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ ← ΑΘΡ / 10

ΓΡΑΨΕ “Ο Μέσος όρος είναι “, ΜΟ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΟ_στοιχείων

Αναζήτηση στοιχείου (θεωρούμε ότι αν υπάρχει, υπάρχει μια μόνο φορά).

Εδώ θα δούμε την **σειριακή μέθοδο**. Η λογική είναι η εξής : Σαρώνουμε τον πίνακα και εξετάζουμε σε κάθε θέση αν το στοιχείο της θέσης αυτής είναι ίσο με αυτό που ψάχνουμε. Αν

ναι, τότε σταματάμε.

Πότε χρησιμοποιούμε τη μέθοδο αυτή:

Όταν ο πίνακας είναι μη ταξινομημένος

Ο πίνακας είναι μικρού μεγέθους (π.χ. $N \leq 20$)

Η αναζήτηση να πραγματοποιείται σπάνια, διότι η μέθοδος αυτή είναι σχετικά αργή.

Θα χρειαστούμε μία **μεταβλητή βρέθηκε** λογικού τύπου, που θα γίνει ΑΛΗΘΗΣ αν το στοιχείο βρεθεί.

Επίσης, δεν γνωρίζουμε σε ποια θέση βρίσκεται. Μπορεί να είναι στην 1^η θέση αλλά μπορεί να είναι στη Νιοστή θέση. Θα χρησιμοποιήσουμε και μία **μεταβλητή θέση** που θα κρατήσει τη θέση όπου βρέθηκε.

ΠΡΟΓΡΑΜΜΑ Σειριακή_αναζήτηση

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π [10]

ΑΚΕΡΑΙΕΣ: i *!ο δείκτης θέσης*

ΑΚΕΡΑΙΕΣ: key *!αυτό που ψάχνουμε*

ΑΚΕΡΑΙΕΣ: θέση *!ο αριθμός θέσης όπου βρίσκεται το στοιχείο*
ΛΟΓΙΚΗ: βρέθηκε *!θα γίνει ΑΛΗΘΗΣ αν βρεθεί το στοιχείο*

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10
ΔΙΑΒΑΣΕ Π [i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

βρέθηκε ← ΨΕΥΔΗΣ *!αρχικοποίηση μεταβλητών*
θέση ← 0
i ← 1

ΌΣΟ (i <= 10) **ΚΑΙ** (βρέθηκε = ΨΕΥΔΗΣ) **ΕΠΑΝΑΛΑΒΕ** *!όσο δεν είμαστε στο τέλος του πίνακα
! και δεν βρέθηκε*

ΑΝ Π[i] = key **ΤΟΤΕ** *!αν το i στοιχείο ισούται με αυτό που ψάχνουμε τότε*
βρέθηκε ← ΑΛΗΘΗΣ *!βρέθηκε*
θέση ← i *!βάλε τον αριθμό θέσης στην μεταβλητή θέση*
ΑΛΛΙΩΣ *!αλλιώς*
i ← i + 1 *!προχώρα στην επόμενη θέση*

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ βρέθηκε = ΑΛΗΘΗΣ **ΤΟΤΕ**

ΓΡΑΨΕ 'Το στοιχείο βρέθηκε στη θέση ', θέση

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Το στοιχείο δεν βρέθηκε '

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ Σειριακή_αναζήτηση

Ταξινόμηση πίνακα.

Θα δούμε τη μέθοδο της φυσαλίδας (Bubble Sort) ή ευθείας ανταλλαγής. Βασίζεται στην αρχή της σύγκρισης γειτονικών στοιχείων του πίνακα και ανταλλαγής τους μέχρι να διαταχθούν όλα σε μία σειρά (αύξουσα ή φθίνουσα).

Η λογική είναι η εξής : Κάνουμε διαδοχικές σαρώσεις στον πίνακα. Σε κάθε σάρωση, το μικρότερο στοιχείο (για αύξουσα) μετακινείται προς την κορυφή του πίνακα. Αφού γίνουν όλες οι σαρώσεις θα έχει επιτευχθεί η ταξινόμηση.

ΠΡΟΓΡΑΜΜΑ Φυσαλίδα_Bubble_sort
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π [10]

ΑΚΕΡΑΙΕΣ: i , j , temp

*!i για τις διαδοχικές σαρώσεις από την αρχή προς το τέλος
! j για τη σάρωση που ελέγχει για το μικρότερο στοιχείο, ξεκινά από το τέλος
! temp βοηθητική μεταβλητή για την ανταλλαγή*

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
ΔΙΑΒΑΣΕ Π [i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 10 *!το i σαρώνει από τη 2^η θέση έως το τέλος*

ΓΙΑ j ΑΠΟ 10 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1 *!το j σαρώνει από το τέλος προς τη θέση i*

ΑΝ Π [$j-1$] > Π [j] ΤΟΤΕ *!αν το $j-1$ στοιχείο είναι μεγαλύτερο από το j στοιχείο τότε*

temp \leftarrow Π [$j-1$] *!κάνε αντιμετάθεση στοιχείων. Δηλαδή, το j στοιχείο θα πάει στη*

Π [$j-1$] \leftarrow Π [j] *!θέση του $j-1$ και το $j-1$ στη θέση του j . Επειδή δεν μπορεί να γίνει άμεσα*

Π [j] \leftarrow temp *!διότι θα χαθεί το στοιχείο $j-1$, προσωρινά τοποθετείται στη βοηθητική*

!μεταβλητή temp.

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Φυσαλίδα_Bubble_sort

Πλεονεκτήματα / Μειονεκτήματα χρήσης των πινάκων

Πλεονεκτήματα : Είναι ένας βολικός κι εύκολος τρόπος διαχείρισης δεδομένων του *ιδίου τύπου*

Μειονεκτήματα :

Απαιτούν μνήμη. Ο πίνακας δεσμεύει από την αρχή του προγράμματος αρκετές θέσεις μνήμης. Έτσι, αν χρησιμοποιούμε πολλούς πίνακες σε ένα πρόγραμμα το επιβαρύνουμε όσον αφορά τη μνήμη.

Περιορίζουμε τις δυνατότητες του προγράμματος. Επειδή ο πίνακας είναι στατική δομή, δεν μπορεί να αλλάξει το μέγεθός του κατά την εκτέλεση του προγράμματος.

Πότε είναι προτιμότερο να χρησιμοποιούμε του πίνακες; Όταν τα δεδομένα που εισάγουμε, απαιτείται να τα αποθηκεύουμε για να επαναχρησιμοποιηθούν (οπότε βρίσκονται στη μνήμη RAM καθ' όλη τη διάρκεια εκτέλεσης του προγράμματος).

Λυμένες ασκήσεις

Να γραφεί πρόγραμμα το οποίο:

α. να διαβάσει τα ονόματα 5 μαθητών και τον βαθμό τους στο ΑΕΠ

β. να βρίσκει και να εμφανίζει τον μεγαλύτερο βαθμό

γ. να βρίσκει και να εμφανίζει το όνομα ή τα ονόματα των μαθητών που έχουν τον μεγαλύτερο βαθμό.

δ. Να βρίσκει και να εμφανίζει τον μέσο όρο των μαθητών

ΠΡΟΓΡΑΜΜΑ μαθητές

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: ον[5]

ΑΚΕΡΑΙΕΣ: β[5], μαξ, ι

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ, αθρ

ΑΡΧΗ

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 5

ΓΡΑΨΕ 'Δώσε το όνομα και τον βαθμό του μαθητή ', ι

ΔΙΑΒΑΣΕ ον[ι], β[ι]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

μαξ ← β[1]

ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 5

ΑΝ μαξ < β[ι] ΤΟΤΕ

μαξ ← β[ι]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Ο μέγιστος βαθμός είναι ', μαξ

ΓΡΑΨΕ 'Οι μαθητές με τον μέγιστο βαθμό είναι : '

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 5

ΑΝ β[ι] = μαξ ΤΟΤΕ

ΓΡΑΨΕ ον[ι]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

αθρ ← 0

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 5

αθρ ← αθρ + β[ι]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ ← αθρ/5

ΓΡΑΨΕ ' Ο μέσος όρος της βαθμολογίας είναι: ', ΜΟ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Σε ένα πανεπιστημιακό τμήμα εισήχθησαν κατόπιν γενικών εξετάσεων 235 φοιτητές προερχόμενοι από την ΤΕΧΝΟΛΟΓΙΚΗ ή την ΘΕΤΙΚΗ κατεύθυνση.

Να αναπτύξετε αλγόριθμο, ο οποίος:

α. Για καθέναν από τους 235 φοιτητές διαβάσει: • το ονοματεπώνυμό του, • τα μόρια εισαγωγής του, • την κατεύθυνσή του, η οποία μπορεί να είναι «ΤΕΧΝΟΛΟΓΙΚΗ» ή «ΘΕΤΙΚΗ», ελέγχοντας την εγκυρότητα εισαγωγής της και να καταχωρεί τα δεδομένα αυτά σε τρεις πίνακες.

β. Να υπολογίζει και να εμφανίζει:

1. τον μέσο όρο των μορίων εισαγωγής των φοιτητών που προέρχονται από την ΤΕΧΝΟΛΟΓΙΚΗ κατεύθυνση.

2. το ποσοστό των φοιτητών, που προέρχονται από την ΤΕΧΝΟΛΟΓΙΚΗ κατεύθυνση.

3. την κατεύθυνση, από την οποία προέρχεται ο φοιτητής με τα περισσότερα μόρια εισαγωγής (να θεωρήσετε ότι δεν υπάρχει περίπτωση ισοβαθμίας).

ΠΡΟΓΡΑΜΜΑ Πανεπιστήμιο

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: ονομα[235], κατ[235]

ΑΚΕΡΑΙΕΣ: μορια[235], ι, συνολο, πλτ, μαξ, θ

ΠΡΑΓΜΑΤΙΚΕΣ: μο, ποσ

ΑΡΧΗ

σύνολο ← 0 !αρχικές τιμές

πλτ ← 0
μαξ ← -1

ΓΙΑ ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 235

ΓΡΑΨΕ "δώσε όνομα " , ι
ΔΙΑΒΑΣΕ ονομα[ι]
ΓΡΑΨΕ "δώσε μόρια "
ΔΙΑΒΑΣΕ μορια[ι]
ΓΡΑΨΕ "δώσε κατεύθυνση "

ΔΙΑΒΑΣΕ κατ[ι] ! έλεγχος εγκυρότητας

ΟΣΟ κατ[ι] <> "ΤΕΧΝΟΛΟΓΙΚΗ " και κατ[ι] <> "ΘΕΤΙΚΗ " **ΕΠΑΝΑΛΑΒΕ**
ΓΡΑΨΕ "λαθος κατεύθυνση. Ξαναδώστε "
ΔΙΑΒΑΣΕ κατ[ι]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ κατ[ι]="ΤΕΧΝΟΛΟΓΙΚΗ " **ΤΟΤΕ** ! υπολογισμός Μέσου όρου
συνολο ← συνολο +μορια[ι]
πλτ ← πλτ+1
ΤΕΛΟΣ_ΑΝ

ΑΝ μορια[ι] > μαξ **ΤΟΤΕ** ! εύρεση φοιτητή με τα περισσότερα μόρια
μαξ ← μορια[ι]
θ ← ι
ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ πλτ <> 0 **ΤΟΤΕ**
μο ← συνολο/πλτ
ΓΡΑΨΕ "ο μέσο όρος της τεχνολογικής είναι : ", μο
ΤΕΛΟΣ_ΑΝ

ποσ ← πλτ /235 * 100

ΓΡΑΨΕ "Το ποσοστό των φοιτητών της τεχνολογικής είναι : ", ποσ

ΓΡΑΨΕ "Η κατεύθυνση του φοιτητή με τα περισσότερα μόρια είναι ", κατ[θ]

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Να γραφεί πρόγραμμα το οποίο:

- Να διαβάζει τα ονόματα 15 μαθητών και τον βαθμό τους στο ΑΕΠ
- Να ταξινομεί τους μαθητές με φθίνουσα σειρά βαθμολογίας. (Δεν υπάρχει περίπτωση ισοβαθμίας)
- Να βρίσκει και να εμφανίζει τα ονόματα των 3 καλύτερων μαθητών (που έχουν δηλαδή τον μεγαλύτερο βαθμό).

ΠΡΟΓΡΑΜΜΑ μαθητές

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Όνομα [15] , τ2 !τ2 βοηθητική μεταβλητή για την ανταλλαγή των ονομάτων

ΑΚΕΡΑΙΕΣ: Βαθμοί [15], μαξ, ι, τ1, j !τ1 βοηθητική μεταβλητή για την ανταλλαγή των βαθμών

ΑΡΧΗ

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 15

ΓΡΑΨΕ 'Δώσε το όνομα και '

ΓΡΑΨΕ 'τον βαθμό του μαθητή ', i

ΔΙΑΒΑΣΕ Όνομα[i], Βαθμοί[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i **ΑΠΟ** 2 **ΜΕΧΡΙ** 15

ΓΙΑ j **ΑΠΟ** 15 **ΜΕΧΡΙ** i **ΜΕ_ΒΗΜΑ** -1

ΑΝ Βαθμοί[$j-1$] < Βαθμοί[j] **ΤΟΤΕ**

$\tau_1 \leftarrow$ Βαθμοί[$j-1$]

!Αντιμετάθεση βαθμών

Βαθμοί[$j-1$] \leftarrow Βαθμοί[j]

Βαθμοί[j] \leftarrow τ_1

$\tau_2 \leftarrow$ Όνομα[$j-1$]

!Αντιμετάθεση ονομάτων

Όνομα[$j-1$] \leftarrow Όνομα[j]

Όνομα[j] \leftarrow τ_2

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ "Οι 3 καλύτεροι μαθητές είναι"

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 3

ΓΡΑΨΕ Όνομα[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Ασκήσεις 9, 10, 11, 13 σελ. 46

Ασκήσεις 1, 2, 4, 5, 7 σελ. 75
