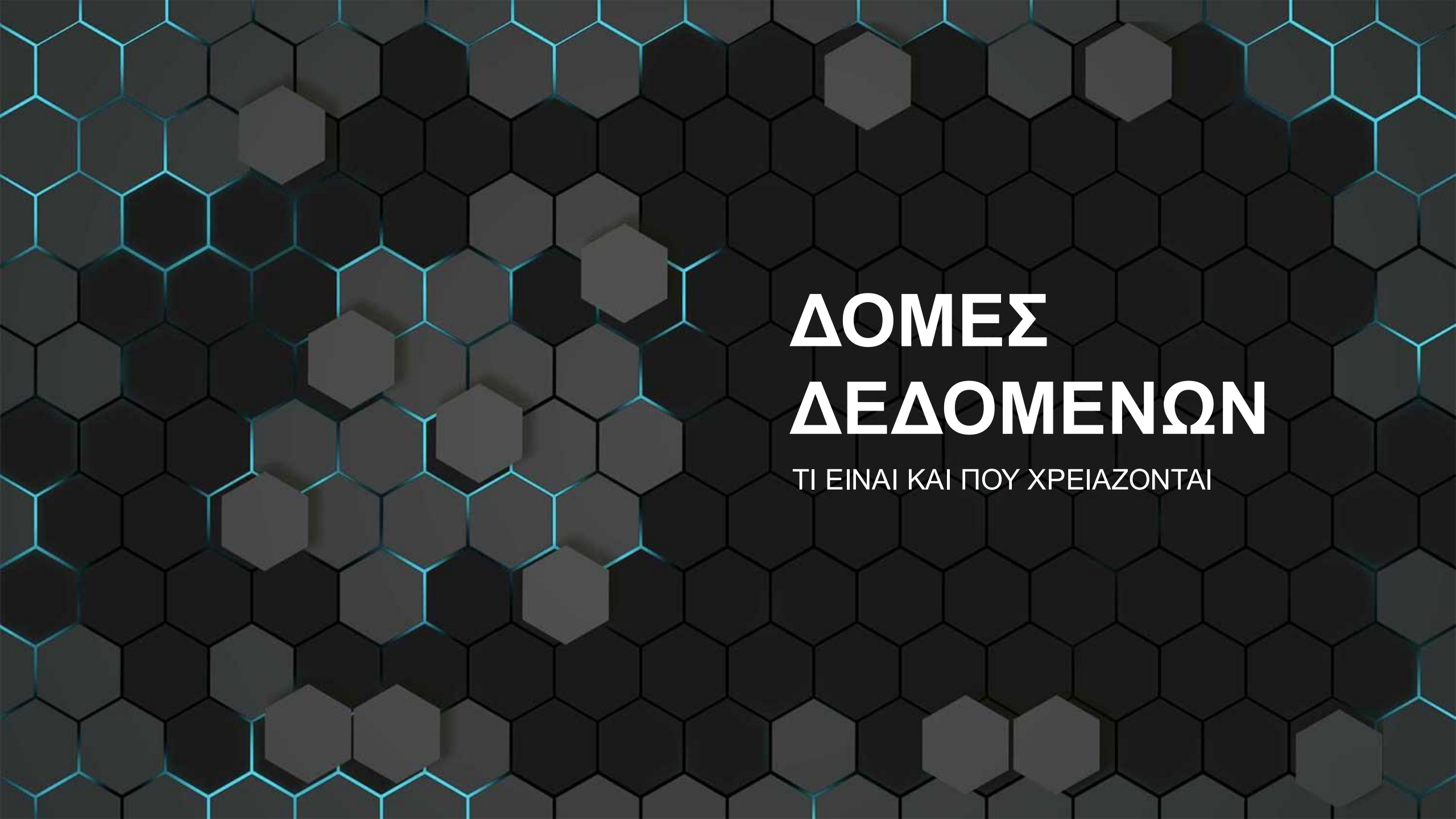


ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

ΤΟΜΠΟΥΣΟΓΛΟΥ ΑΛΕΞΑΝΔΡΟΣ
tobousoglou.a@zois.edu.gr



ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΤΙ ΕΙΝΑΙ ΚΑΙ ΠΟΥ ΧΡΕΙΑΖΟΝΤΑΙ

ΒΑΣΙΚΗ ΘΕΩΡΙΑ

01

ΟΡΙΣΜΟΣ Δ.Δ

- ✓ Μια δομή δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών

02

ΚΟΜΒΟΣ Δ.Δ

- ✓ Κάθε Δ.Δ αποτελείται από ένα σύνολο στοιχείων που ονομάζονται κόμβοι

04

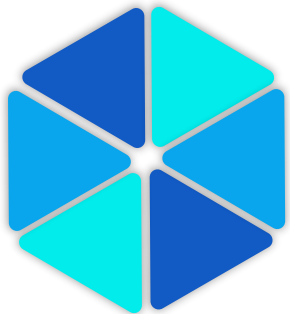
ΧΡΗΣΙΜΑ ΣΤΟΙΧΕΙΑ

- ✓ Στην πράξη ΔΕΝ χρησιμοποιούνται και οι 8 λειτουργίες πάνω σε κάποια δομή.
- ✓ Αλγόριθμοι + Δ.Δ => Προγράμματα
- ✓ Οι πίνακες (στα πλαίσια του μαθήματος) αποτελούν στατική Δ.Δ, δηλαδή τα στοιχεία τους αποθηκεύονται σε συνεχόμενες θέσεις μνήμης και το μέγεθος τους δεν αλλάζει κατά τη διάρκεια εκτέλεσης του προγράμματος
- ✓ Υπάρχουν και οι δυναμικές Δ.Δ (τα στοιχεία τους δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης – αν και δεν είναι υποχρεωτικό κάτι τέτοιο – και το μέγεθος τους είναι μεταβλητό)
- ✓ Συμμετρικός λέγεται ο τετραγωνικός πίνακας για τον οποίο ισχύει:
 $A[i] = A[N + 1 - i]$
όπου N αριθμός γραμμών/στηλών

03

ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΕΠΙ ΤΩΝ Δ.Δ

- ✓ ΠΡΟΣΠΕΛΑΣΗ
- ✓ ΕΙΣΑΓΩΓΗ
- ✓ ΔΙΑΓΡΑΦΗ
- ✓ ΑΝΑΖΗΤΗΣΗ
- ✓ ΤΑΞΙΝΟΜΗΣΗ
- ✓ ΑΝΤΙΓΡΑΦΗ
- ✓ ΣΥΓΧΩΝΕΥΣΗ
- ✓ ΔΙΑΧΩΡΙΣΜΟΣ



ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΕΠΙ ΜΙΑΣ Δ.Δ



Όλες οι Δ.Δ δεν μπορούν να υλοποιήσουν όλες τις βασικές λειτουργίες



Οι κόμβοι μιας δυναμικής Δ.Δ αποθηκεύονται υποχρεωτικά σε μη συνεχόμενες θέσεις μνήμης του υπολογιστή



Οι στατικές Δ.Δ υλοποιούν τις βασικές πράξεις

Η προσπέλαση, η διαγραφή και η αναζήτηση είναι όλες βασικές λειτουργίες επί των στατικών Δ.Δ

ΠΡΟΣΠΕΛΑΣΗ

Πρόσβαση σε έναν κόμβο(στοιχείο) για εξέταση ή επεξεργασία του περιεχομένου του

ΕΙΣΑΓΩΓΗ

Προσθήκη νέου κόμβου(στοιχείου) σε μια Δ.Δ

ΔΙΑΓΡΑΦΗ

Αφαίρεση ενός κόμβου(στοιχείου) από μια Δ.Δ

ΤΑΞΙΝΟΜΗΣΗ

Αύξουσα ή φθίνουσα τακτοποίηση των κόμβων(στοιχείων) μιας Δ.Δ

ΑΝΑΖΗΤΗΣΗ

Εντοπισμός κόμβου/κόμβων μέσα σε μια Δ.Δ που έχουν κάποια ιδιότητα

ΣΥΓΧΩΝΕΥΣΗ

Συνένωση 2 ή και περισσότερων Δ.Δ σε 1

ΔΙΑΧΩΡΙΣΜΟΣ

Διάσπαση μιας Δ.Δ σε τουλάχιστον 2 Δ.Δ

ΑΝΤΙΓΡΑΦΗ

Ένας ή περισσότεροι κόμβοι(στοιχεία) μιας Δ.Δ τοποθετούνται τοποθετούνται και σε μια άλλη Δ.Δ



1.1

-

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

ΘΕΩΡΙΑ ΠΙΝΑΚΩΝ

Κάθε στοιχείο του πίνακα(κελί) στην ουσία αποτελεί μια ξεχωριστή μεταβλητή και μπορεί να υποβληθεί σε επεξεργασία ως τέτοια

Οι πίνακες είναι στατικές δομές δεδομένων που σημαίνει πως οι διαστάσεις τους είναι σταθερές και δεν αλλάζουν. Το μέγεθος τους δηλώνεται στο τμήμα δηλώσεων του προγράμματος ενώ τα στοιχεία του αποθηκεύονται σε συνεχόμενες θέσεις μνήμης

ΤΙ ΕΙΝΑΙ ΕΝΑΣ ΠΙΝΑΚΑΣ

Ονομάζεται ένα σύνολο στοιχείων, απαραίτητα ιδίου τύπου, τα οποία αναφέρονται με ένα κοινό όνομα. Η αναφορά σε ατομικά στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από κάποιον ακέραιο δείκτη(αριθμητική έκφραση, μεταβλητή ή σταθερά). Ένας πίνακας μπορεί να είναι μονοδιάστατος, δυσδιάστατος κτλ. ενώ απαιτείται ο αντίστοιχος αριθμός δεικτών για την αναφορά μεμονωμένων στοιχείων του πίνακα

ΓΙΑΤΙ ΤΟΥΣ ΧΡΕΙΑΖΟΜΑΣΤΕ

Όπως κάθε Δ.Δ, δίνουν τη δυνατότητα αποθήκευσης μεγάλου όγκου δεδομένων αλλά και επεξεργασίας τους. Σε πολλές περιπτώσεις απαιτείται η πολλαπλή επεξεργασία και η διατήρηση στη μνήμη των στοιχείων και για μετέπειτα επεξεργασία τους. Εκεί οι μεταβλητές είναι δεσμευτικές και κρίνεται απαραίτητη η χρήση των πινάκων.

ΠΩΣ ΔΗΛΩΝΟΥΜΕ ΕΝΑΝ ΠΙΝΑΚΑ ΣΕ ΓΛΩΣΣΑ;

Όπως δηλώνουμε μεταβλητές στο τμήμα δηλώσεων σε ΓΛΩΣΣΑ δηλώνουμε και τους πίνακες.

Συγκεκριμένα δηλώνουμε το όνομα του πίνακα ακολουθούμενο από αγκύλες που προσδιορίζουν το μέγεθος του



```
1 ΜΕΤΑΒΛΗΤΕΣ
2   ΑΚΕΡΑΙΕΣ: ΗΛΙΚΙΕΣ[50]
3   ΠΡΑΓΜΑΤΙΚΕΣ: ΥΨΗ[100]
4   ΧΑΡΑΚΤΗΡΕΣ: ΟΝΟΜΑΤΑ[70]
```

ΣΩΣΤΟ - ΛΑΘΟΣ

ΕΠΙΠΕΔΟ 1

- Ένας πίνακας μπορεί να είναι μονοδιάστατος, δισδιάστατος ή και ν-διάστατος
- Ένας πίνακας έχει σταθερό αριθμό κόμβων
- Ένας πίνακας μπορεί να περιέχει αριθμητικές τιμές, χαρακτήρες ή και λογικά δεδομένα, αρκεί να είναι όλα του ιδίου τύπου
- Σε έναν πίνακα απαγορεύεται να υπάρχει η ίδια τιμή παραπάνω από 1 φορές
- Για την προσπέλαση ενός πίνακα μπορεί να χρησιμοποιηθεί οποιαδήποτε δομή επανάληψης
- Το μέγεθος ενός πίνακα καθορίζεται πάντα από το μεταγλωττιστή προτού εκτελεστεί το πρόγραμμα και όχι κατά τη διάρκεια του
- Κάθε πρόγραμμα που χρησιμοποιεί πίνακες μπορεί εναλλακτικά να γραφτεί και χωρίς τη χρήση τους

ΕΠΙΠΕΔΟ 2

- Οι λειτουργίες της εισαγωγής και της διαγραφής μπορούν να εφαρμοσθούν σε έναν πίνακα
- Ένας πίνακας που χρησιμοποιεί μόνο 1 δείκτη για την αναφορά των στοιχείων του είναι σίγουρα μονοδιάστατος
- Για να προσπελάσουμε μαζικά τα στοιχεία ενός πίνακα, χρειαζόμαστε απαραίτητα επαναληπτική δομή
- Οι πίνακες πρέπει να χρησιμοποιούνται σε κάθε περίπτωση που θέλουμε να εισάγουμε πολλά στοιχεία σε έναν αλγόριθμο
- Μέσα στις αγκύλες που αναφέρονται στη θέση ενός πίνακα μπορεί να υπάρχει οποιαδήποτε ακέραια αριθμητική έκφραση εκτός από το 0
- Ένα πρόγραμμα σε ΓΛΩΣΣΑ δεσμεύει τόσες συνεχόμενες θέσεις μνήμης για κάποιον πίνακα, όσες και οι θέσεις που εμφανίζονται στη δήλωση του στο τμήμα δηλώσεων του προγράμματος

ΕΠΙΠΕΔΟ 3

- Σε ένα μεγάλο και σύνθετο πρόγραμμα η περιττή χρήση πινάκων οδηγεί σε επιβάρυνση, ακόμα και σε αδυναμία εκτέλεσης του προγράμματος
- Η χρήση πινάκων σε ένα πρόγραμμα αυξάνει την απαιτούμενη μνήμη
- Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος
- Κάθε πίνακας που στα κελιά του περιέχει αριθμητικά στοιχεία, έχει ελάχιστο και μέγιστο στοιχείο
- Ένα κελί του πίνακα δεν μπορεί να χρησιμοποιηθεί, αν πρώτα δεν έχει αρχικοποιηθεί
- Οι αριθμητικοί τελεστές εφαρμόζονται σε μονοδιάστατους πίνακες
- Η δυναμική παραχώρηση μνήμης χρησιμοποιείται στις δομές των πινάκων

ΠΑΡΑΔΕΙΓΜΑΤΑ

```
ΠΡΟΓΡΑΜΜΑ ΑΘΡΟΙΣΜΑ
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: ΤΙΜΕΣ[5], i, Σ
ΑΡΧΗ
  ! ΑΡΧΙΚΟΠΟΙΗΣΗ ΤΙΜΩΝ ΠΙΝΑΚΑ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    ΓΡΑΨΕ 'Δωσε τιμή: ', i
    ΔΙΑΒΑΣΕ ΤΙΜΕΣ[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

  Σ <- 0
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 5
    Σ <- Σ + ΤΙΜΕΣ[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

  ΓΡΑΨΕ 'Το άθροισμα των στοιχείων είναι', Σ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

```
Αλγόριθμος test
Για i από 1 μέχρι 10
  Στοιχεία[i] <- i + 10
Τέλος_επανάληψης
Για i από 1 μέχρι 10
  Αν i <= 5 τότε
    Εμφάνισε Στοιχεία[i]
  Τέλος_αν
Τέλος_επανάληψης
Τέλος test
```

Τι επιτυγχάνουν τα 2 παραπάνω κομμάτια κώδικα;

```
Αλγόριθμος test
Δεδομένα // Στοιχεία //
Για i από 1 μέχρι 5000
  Αν ..... τότε
    Σ <- Στοιχεία[i]
  Τέλος_αν
Τέλος_επανάληψης
Τέλος test
```

Τι συνθήκη θα πρέπει να εισάγετε στην απλή δομή επιλογής αν θέλετε να υπολογίζεται το άθροισμα:

- Των στοιχείων των περιττών θέσεων του πίνακα;
- Των περιττών στοιχείων του πίνακα;
- Των διψήφιων στοιχείων του πίνακα;
- Των στοιχείων των διψήφιων θέσεων του πίνακα;
- Των στοιχείων με τιμή μικρότερη του 100;

ΠΑΡΑΔΕΙΓΜΑΤΑ

ΠΡΟΓΡΑΜΜΑ ΑΘΡΟΙΣΜΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[5]

ΑΡΧΗ

A[A_M(14/4)] <- 2

A[A[3]] <- A[3] * 14 MOD 5

A[4] <- A[2] * A[3] + 2

A[A[3] MOD 2 + 1] <- A[6 DIV 3] MOD 2

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Ποια θα είναι τα περιεχόμενα του πίνακα A μετά την εκτέλεση των διπλανών εντολών;

ΠΡΟΓΡΑΜΜΑ ΑΘΡΟΙΣΜΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A[5], i, j, πρόχειρο

ΑΡΧΗ

A[1] <- 3

A[2] <- 5

A[3] <- 8

A[4] <- 13

i <- 1

j <- 4

ΟΣΟ i <= 3 ΕΠΑΝΑΛΑΒΕ

πρόχειρο <- A[j]

A[j] <- A[i]

A[i] <- πρόχειρο

ΓΡΑΨΕ A[1], A[2], A[3]

i <- i + 1

j <- j - 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Εξηγήστε το διπλανό πρόγραμμα

- Τι θα εμφανίσει;

- Πως λέγεται η διαδικασία που υλοποιείται μέσα στην δομή επανάληψης ΟΣΟ και τι επιτυγχάνει;

1.2

-

ΜΟΝΟΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ & ΕΡΓΑΛΕΙΑ ΠΡΟΣΠΕΛΑΣΗΣ

ΠΡΟΣΠΕΛΑΣΗ

Πρόσβαση σε έναν κόμβο(στοιχείο) για εξέταση ή επεξεργασία του περιεχομένου του

1.2.1

—

**ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ
ΜΕΣΟΣ ΟΡΟΣ
ΕΥΡΕΣΗ ΠΟΣΟΣΤΟΥ
ΑΠΟΚΛΙΣΗ
ΕΥΡΕΣΗ ΣΥΧΝΟΤΗΤΑΣ**

ΠΡΟΣΠΕΛΑΣΗ

Δεν θα πρέπει να παραβιάζονται τα όρια του πίνακα ούτε να αναφερόμαστε στην θέση 0 του πίνακα

Σε μονοδιάστατο πίνακα χρησιμοποιούμε συνήθως(χωρίς να είναι δεσμευτικό) την δομή επανάληψης Για, καθώς γνωρίζουμε το πλήθος των επαναλήψεων

Είναι καλή πρακτική να χρησιμοποιείται το i της επαναληπτικής δομής Για και για τον ρόλο του δείκτη του πίνακα για πρόσβαση σε κάποιο συγκεκριμένο στοιχείο

Οι πίνακες σε ένα πρόγραμμα πρέπει να δηλώνονται, με το όνομα και το σταθερό πλήθος των θέσεων, στο τμήμα δηλώσεων

ΠΡΟΓΡΑΜΜΑ Άσκηση1
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: μικ50, i, μεγ_μισο_μο, θετικοί, αρνητικοί
 ΠΡΑΓΜΑΤΙΚΕΣ: άθροισμα, AP[500], μο, ποσ_θετικών, ποσ_αρνητικών

Τμήμα δηλώσεων προγράμματος

Να αναπτυχθεί πρόγραμμα το οποίο θα διαβάσει 500 αριθμούς και θα εμφανίζει

1. Το πλήθος των στοιχείων που είναι μικρότερα του 50
2. Το πλήθος των στοιχείων που είναι μεγαλύτερα από το μισό του μέσου όρου
3. Το ποσοστό των θετικών και των αρνητικών στοιχείων του πίνακα

Απαιτείται η χρήση πινάκων για την επίλυση της άσκησης και αν ναι για ποιο λόγο;

ΑΡΧΗ

```
μικ50 <- 0
άθροισμα <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 500
    ΓΡΑΨΕ 'ΔΩΣΕ ΑΡΙΘΜΟ ', i
    ΔΙΑΒΑΣΕ AP[i]
    ΑΝ AP[i] < 50 ΤΟΤΕ
        μικ50 <- μικ50 + 1
    ΤΕΛΟΣ_ΑΝ
    άθροισμα <- άθροισμα + AP[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το πλήθος των αριθμών που είναι μικρότεροι του 50 είναι', μικ50
```

1

```
μο <- άθροισμα / 500
μεγ_μισο_μο <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 500
    ΑΝ AP[i] > μο / 2 ΤΟΤΕ
        μεγ_μισο_μο <- μεγ_μισο_μο + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το πλήθος των αριθμών που είναι μεγαλύτεροι του μισού του μέσου όρου είναι', μεγ_μισο_μο
```

2

```
θετικοί <- 0
αρνητικοί <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 500
    ΑΝ AP[i] > 0 ΤΟΤΕ
        θετικοί <- θετικοί + 1
    ΑΛΛΙΩΣ_ΑΝ AP[i] < 0 ΤΟΤΕ
        αρνητικοί <- αρνητικοί + 1
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'Οι μηδενικοί δεν μας ενδιαφέρουν στον υπολογισμό'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ποσ_θετικών <- (θετικοί / 500) * 100
ποσ_αρνητικών <- (αρνητικοί / 500) * 100
ΓΡΑΨΕ 'Το ποσοστό των θετικών και των αρνητικών αριθμών είναι αντίστοιχα: ', ποσ_θετικών, ποσ_αρνητικών
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

3

Η διοίκηση μιας εταιρείας στα πλαίσια μιας έρευνας σχετικά με τους υπαλλήλους της χρειάζεται ένα πρόγραμμα που θα διαβάσει σε 2 πίνακες το φύλο και τον μισθό των 750 υπαλλήλων της. Για το φύλο κάθε υπαλλήλου ο χρήστης πρέπει να δίνει το “Α” ή το “Γ” ενώ για το μισθό να δίνει ακέραιο μεγαλύτερο του 0(Να γίνεται έλεγχος για αυτά)

Το πρόγραμμα θα πρέπει αφού διαβάσει τα στοιχεία αυτά να υπολογίζει:

- Το πλήθος των εργαζομένων με μισθό άνω των 1000 ευρώ. Αν δεν υπάρχει υπάλληλος με τέτοιο μισθό θα εμφανίζει κατάλληλο μήνυμα
- Το ποσοστό των γυναικών με μισθό άνω των 1000 ευρώ, εφόσον υπάρχει
- Το ποσοστό των ανδρών που έχουν μισθό πάνω από το μισό του μέσου όρου των μισθών
- **Πόσοι εργαζόμενοι αμείβονται με μισθό πιο κοντά στον μέσο όρο**
- Το πλήθος των εργαζομένων που έχουν μισθό μεγαλύτερο από 2000 ευρώ. (Αν δεν υπάρχει κανείς, τότε να εμφανίζεται κατάλληλο μήνυμα)

```
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 750
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΡΑΨΕ 'Δώσε φύλο υπαλλήλου...'
        ΔΙΑΒΑΣΕ Sex[i]
        ΓΡΑΨΕ 'Δώσε μισθό υπαλλήλου...'
        ΔΙΑΒΑΣΕ Salary[i]
        ΜΕΧΡΙΣ_ΟΤΟΥ (Sex[i]='Α' Η Sex[i]='Γ') ΚΑΙ Salary[i] > 0
        ΓΡΑΨΕ 'Ο υπάλληλος', i, 'καταχωρήθηκε!'
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

άνω_των_1000 <- 0
γυναίκες_άνω_των_1000 <- 0
σύνολο_μισθών <- 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 750
    σύνολο_μισθών <- σύνολο_μισθών + Salary[i]
    ΑΝ Salary[i] > 1000 ΤΟΤΕ
        άνω_των_1000 <- άνω_των_1000 + 1
        ΑΝ Sex[i] = 'Γ' ΤΟΤΕ
            γυναίκες_άνω_των_1000 <- γυναίκες_άνω_των_1000 + 1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ άνω_των_1000 > 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Οι εργαζόμενοι με μισθό άνω των 1000 είναι ', άνω_των_1000
    ΑΝ γυναίκες_άνω_των_1000 > 0 ΤΟΤΕ
        ποσοστό_γυναικών <- (γυναίκες_άνω_των_1000 / 750) * 100
        ΓΡΑΨΕ 'Το ποσοστό των γυναικών με μισθό άνω των 1000 είναι ', ποσοστό_γυναικών
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

μο_μισθών <- σύνολο_μισθών / 750
άντρες_άνω_μο <- 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 750
    ΑΝ Sex[i] = 'Α' ΚΑΙ Salary[i] > μο_μισθών ΤΟΤΕ
        άντρες_άνω_μο <- άντρες_άνω_μο + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ποσοστό_ανδρών_άνω_μο <- (άντρες_άνω_μο / 750) * 100
ΓΡΑΨΕ 'Το ποσοστό των ανδρών με μισθό άνω του μέσου όρου είναι ', ποσοστό_ανδρών_άνω_μο
```

```
! ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΚΛΙΣΗΣ
ελάχιστη_απόκλιση <- Α_Τ(Salary[1] - μο_μισθών)
ΓΡΑΨΕ ελάχιστη_απόκλιση
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 750
    ΑΝ Α_Τ(Salary[i] - μο_μισθών) < ελάχιστη_απόκλιση ΤΟΤΕ
        ελάχιστη_απόκλιση <- Α_Τ(Salary[i] - μο_μισθών)
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

εργ_κοντά_στον_μο <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 750
    ΑΝ Α_Τ(Salary[i] - μο_μισθών) = ελάχιστη_απόκλιση ΤΟΤΕ
        εργ_κοντά_στον_μο <- εργ_κοντά_στον_μο + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Οι εργαζόμενοι που έχουν μισθό πιο κοντά στον ΜΟ ', εργ_κοντά_στον_μο

υπάρχει_άνω_2000 <- ΨΕΥΔΗΣ
πλήθος_άνω_2000 <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 750
    ΑΝ Salary[i] > 2000 ΤΟΤΕ
        υπάρχει_άνω_2000 <- ΑΛΗΘΗΣ
        ΓΡΑΨΕ 'Ο εργαζόμενος', i, ' έχει μισθό άνω των 2000'
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ υπάρχει_άνω_2000 = ΑΛΗΘΗΣ ΤΟΤΕ
    ΓΡΑΨΕ 'Δεν υπάρχει εργαζόμενος με εισόδημα άνω των 2000 ευρώ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΑΣΚΗΣΗ

Αναπτύξτε πρόγραμμα το οποίο θα διαβάσει 200 ακέραιους βαθμούς στην κλίμακα 1-20(να πραγματοποιήσετε έλεγχο) στο μάθημα των μαθηματικών και θα εκτυπώνει την συχνότητα εμφάνισης κάθε βαθμού.

Επιπρόσθετα το πρόγραμμα θα εμφανίζει ποιος βαθμός εμφανίστηκε τις περισσότερες φορές

```
ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 200
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΓΡΑΨΕ 'Δώσε Βαθμό...'
      ΔΙΑΒΑΣΕ Grade[i]
      ΜΕΧΡΙΣ_ΟΤΟΥ Grade[i] <= 20 ΚΑΙ Grade[i] >= 0
      ΓΡΑΨΕ 'Ο βαθμός', i, 'καταχωρήθηκε!'
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
      ΕΜΦ[i] <- 0
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 200
      counter <- Grade[i]
      ΕΜΦ[counter] <- ΕΜΦ[counter] + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
      ΓΡΑΨΕ 'Ο Βαθμός ', i, ' εμφανίστηκε ', ΕΜΦ[i], ' φορές'
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    index_max_appearances <- 1
    ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 20
      ΑΝ ΕΜΦ[i] > ΕΜΦ[index_max_appearances] ΤΟΤΕ
        index_max_appearances <- i
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

    ΓΡΑΨΕ 'Ο βαθμός με τις περισσότερες εμφανίσεις είναι ο ', index_max_appearances
  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```


1.2.2

–

ΕΝΤΟΠΙΣΜΟΣ ΜΕΓΙΣΤΟΥ / ΕΛΑΧΙΣΤΟΥ

Αναπτύξτε πρόγραμμα που θα διαβάσει 200 αριθμούς σε έναν πίνακα και θα εμφανίζει τον μέγιστο καθώς και την θέση που βρίσκεται το μέγιστο μέσα στον πίνακα

1. Δεν μας αναφέρει αν ο πίνακας είναι ταξινομημένος (όταν έχουμε μη ταξινομημένο πίνακα, τότε για την εύρεση του μέγιστου ή του ελάχιστου απαιτείται η προσπέλαση όλων των στοιχείων)
2. Αν το μέγιστο συναντάται παραπάνω από μια φορά(πχ στην 3^η και την 18^η θέση), ποια θέση πρέπει να εμφανίσουμε; Γενικά, αν μας ζητάει ενικό όπως εδώ(τον μέγιστο) τότε αρκεί ο εντοπισμός ενός ελαχίστου/μεγίστου. Αν μας ζητάει πληθυντικό (βρείτε τα μέγιστα) ζητείται ο εντοπισμός όλων των θέσεων

Στην επόμενη διαφάνεια θα αναλύσουμε και τις 3 περιπτώσεις, δηλαδή:

- Να εντοπίσουμε το μέγιστο που εμφανίζεται για 1η φορά στον πίνακα
- Να εντοπίσουμε το μέγιστο που εμφανίζεται για τελευταία φορά στον πίνακα
- Να εντοπίσουμε όλα τα μέγιστα που υπάρχουν στον πίνακα

Τα παραπάνω έχουν νόημα αν το μέγιστο πράγματι υπάρχει παραπάνω από 1 φορά

ΠΡΟΓΡΑΜΜΑ Άσκηση1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ι, θmax

ΠΡΑΓΜΑΤΙΚΕΣ: AP[200], max

ΑΡΧΗ

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 200

ΓΡΑΨΕ 'ΔΩΣΕ ΑΡΙΘΜΟ ', ι

ΔΙΑΒΑΣΕ AP[ι]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

max <- AP[1]

θmax <- 1

ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 200

ΑΝ AP[ι] > max ΤΟΤΕ

max <- AP[ι]

θmax <- ι

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το μέγιστο είναι ', max

ΓΡΑΨΕ 'Η θέση που βρίσκεται το μέγιστο είναι η ', θmax

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Με αυτόν τον τρόπο
εντοπίζουμε το μέγιστο και
την ΠΡΩΤΗ θέση που
αυτό συναντάται στον
πίνακα

ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 200

ΑΝ AP[ι] >= max ΤΟΤΕ

max <- AP[ι]

θmax <- ι

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Αν θέλαμε να εντοπίσουμε
το ΤΕΛΕΥΤΑΙΟ μέγιστο
αρκεί να προσθέσουμε την
ισότητα στην λογική
συνθήκη

ΠΡΟΓΡΑΜΜΑ Άσκηση1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ι, θmax

ΠΡΑΓΜΑΤΙΚΕΣ: AP[10], max

ΑΡΧΗ

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 10

ΓΡΑΨΕ 'ΔΩΣΕ ΑΡΙΘΜΟ ', ι

ΔΙΑΒΑΣΕ AP[ι]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

max <- AP[1]

ΓΙΑ ι ΑΠΟ 2 ΜΕΧΡΙ 10

ΑΝ AP[ι] >= max ΤΟΤΕ

max <- AP[ι]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το μέγιστο είναι ', max

Εφόσον θέλουμε να εντοπίσουμε όλες τις θέσεις που
βρίσκεται το μέγιστο, αρχικά χρησιμοποιούμε τον
πρώτο τρόπο για να εντοπίσουμε την αριθμητική τιμή
του μέγιστου (Η θέση σε αυτή τη φάση δεν μας
ενδιαφέρει)

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ 10

ΑΝ AP[ι] = max ΤΟΤΕ

ΓΡΑΨΕ 'Η τρέχουσα θέση που βρέθηκε το μέγιστο είναι η ', ι

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Στην συνέχεια χρησιμοποιούμε μια νέα επανάληψη,
όπου ελέγχουμε σε κάθε θέση του πίνακα αν υπάρχει
αυτή η μέγιστη τιμή και επιτόπου προβάλλουμε στην
οθόνη τη θέση του

Το μέγιστο είναι 100.00

Η τρέχουσα θέση που βρέθηκε το μέγιστο είναι η 2

Η τρέχουσα θέση που βρέθηκε το μέγιστο είναι η 5

Η τρέχουσα θέση που βρέθηκε το μέγιστο είναι η 7

Η τρέχουσα θέση που βρέθηκε το μέγιστο είναι η 10

1.2.3

—

ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΑΠΟ ΧΡΗΣΤΗ

Για μια παράσταση σε κάποιο θέατρο χωρητικότητας 100 θέσεων, γνωρίζουμε την ηλικία του κάθε θεατή που έχει κλείσει εισιτήριο. Να γράψετε πρόγραμμα το οποίο θα διαβάσει το πλήθος των θεατών (προφανώς δεν μπορεί να είναι αριθμός μικρότερος του 0 ή μεγαλύτερος του 100) και για κάθε θεατή την ηλικία του(πρέπει να είναι θετικός αριθμός). Το πρόγραμμα θα εμφανίζει το μικρότερο και το μεγαλύτερο σε ηλικία θεατή που έχει κλείσει εισιτήριο.

Επίσης, θα εμφανίζει την μεγαλύτερη ηλικία ανάμεσα στους θεατές κάτω των 18 ετών(εφόσον φυσικά υπάρχουν τέτοιοι)

ΠΡΟΓΡΑΜΜΑ Άσκηση1

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ι, πλήθος, ΗΛΙΚΙΑ[100], min, max, max_κάτω_18

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε τον αριθμό των θεατών'

ΔΙΑΒΑΣΕ πλήθος

ΜΕΧΡΙΣ_ΟΤΟΥ πλήθος>0 ΚΑΙ πλήθος<=100

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ πλήθος

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε την ηλικία για τον θεατή ', ι

ΔΙΑΒΑΣΕ ΗΛΙΚΙΑ[ι]

ΜΕΧΡΙΣ_ΟΤΟΥ ΗΛΙΚΙΑ[ι]>0

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

min <- ΗΛΙΚΙΑ[1]

max <- ΗΛΙΚΙΑ[1]

ΓΙΑ ι ΑΠΟ 1 μέχρι πλήθος

ΑΝ ΗΛΙΚΙΑ[ι] > max ΤΟΤΕ
max <- ΗΛΙΚΙΑ[ι]

ΤΕΛΟΣ_ΑΝ

ΑΝ ΗΛΙΚΙΑ[ι] < min ΤΟΤΕ
min <- ΗΛΙΚΙΑ[ι]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Η μεγαλύτερη και η μικρότερη ηλικία είναι αντίστοιχα: ', max, min

max_κάτω_18 <- -1

ΓΙΑ ι ΑΠΟ 1 ΜΕΧΡΙ πλήθος

ΑΝ ΗΛΙΚΙΑ[ι] > max_κάτω_18 ΚΑΙ ΗΛΙΚΙΑ[ι] < 18 ΤΟΤΕ
max_κάτω_18 <- ΗΛΙΚΙΑ[ι]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ max_κάτω_18 = -1 ΤΟΤΕ

ΓΡΑΨΕ 'Δεν βρέθηκε θεατής κάτω των 18 ετών'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Η μεγαλύτερη ηλικία θεατή κάτω των 18 ετών είναι ', max_κάτω_18

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

1.2.5

—

ΑΣΚΗΣΕΙΣ ΓΙΑ ΣΠΙΤΙ

1) Δημιουργήστε πρόγραμμα, το οποίο...



Θα διαβάξει 2000 ακέραιους αριθμούς

Θα υπολογίζει και θα εκτυπώνει το πλήθος των θετικών, των αρνητικών και των μηδενικών τιμών

Θα υπολογίζει το ποσοστό των θετικών, των αρνητικών και των μηδενικών τιμών επί του συνόλου

Τέλος, θα εμφανίζει ποια κατηγορία ήταν η πολυπληθέστερη

ΑΣΚΗΣΗ 2

Δημιουργήστε πρόγραμμα που θα διαβάξει το όνομα, το ύψος και το βάρος 100 αθλητών πυγμαχίας και θα τοποθετεί αυτά τα στοιχεία σε αντίστοιχους πίνακες. Στην συνέχεια, θα πρέπει:

- Να εκτυπώνει το μέσο όρο για το ύψος των αθλητών
- Να εκτυπώνει το όνομα του πιο ψηλού αθλητή
- Να εκτυπώνει το όνομα του πιο βαρύ αθλητή
- Ο πιο ψηλός και ο πιο βαρύς αθλητής είναι το ίδιο άτομο; Να γίνεται έλεγχος προβάλλοντας στην οθόνη κατάλληλο μήνυμα

	ΥΨΟΣ	ΟΝΟΜΑ	ΒΑΡΟΣ
1			
2			
3			
4			

Προφανώς δεν μπορούμε να έχουμε τα δεδομένα σε έναν κοινό πίνακα καθώς δεν είναι του ιδίου τύπου. Μπορούμε όμως να δημιουργήσουμε 3 παράλληλους πίνακες(υποχρεωτικό το ίδιο πλήθος γραμμών) που συσχετίζονται μεταξύ τους

Παραδείγματος χάριν, η αριθμητική τιμή του `ΥΨΟΣ[1]` θα αντιστοιχεί στο `ΟΝΟΜΑ[1]` και `ΒΑΡΟΣ[1]` που σημαίνει ότι τα στοιχεία αυτά θα ανήκουν στον 1^ο αθλητή

ΑΣΚΗΣΗ 3

Για την στατιστική εικόνα του μαθήματος των μαθηματικών ανάμεσα σε 1000 μαθητές, λαμβάνονται με ηλεκτρονικό τρόπο τα ονόματα, οι προφορικοί βαθμοί των 2 τετραμήνων και οι γραπτοί βαθμοί των μαθητών στις εξετάσεις

Ο τελικός βαθμός είναι ο μέσος όρος του μέσου προφορικού με τον γραπτό βαθμό

Να αναπτύξετε πρόγραμμα που θα υπολογίζει και θα εμφανίζει

1

Τους τελικούς βαθμούς όλων των μαθητών στο μάθημα των μαθηματικών

2

Τα ονόματα των μαθητών με τελικό βαθμό μικρότερο από 9.5

3

Τα ονόματα των μαθητών με τελικό βαθμό μεγαλύτερο από 18

4

Τον μέγιστο τελικό βαθμό

5

Το πλήθος των μαθητών που έχουν τελικό βαθμό ίσο με τον μέγιστο