

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΔΙΣΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ



ΘΕΩΡΙΑ

01

ΓΙΑΤΙ ΧΡΕΙΑΖΟΝΤΑΙ

- ✓ Πολλές φορές τα δεδομένα χρειάζεται να ομαδοποιηθούν με τέτοιο τρόπο ώστε οι μονοδιάστατοι πίνακες να μην είναι αποτελεσματικοί. Σκεφτείτε την περίπτωση να θέλουμε να αποθηκεύσουμε τους πόντους που πέτυχε κάθε ένας παίκτης του ΠΑΟ(σύνολο 15) στις 32 αγωνιστικές της ευρωλίγκας. Θα χρειαζόμασταν 32 μονοδιάστατους πίνακες από 15 θέσεις ο κάθε ένας

02

ΠΟΣΕΣ ΔΙΑΣΤΑΣΕΙΣ

- ✓ Σε μια γλώσσα προγραμματισμού(και στη ΓΛΩΣΣΑ) μπορούμε να έχουμε πίνακες N διαστάσεων, όμως στην συντριπτική πλειοψηφία των περιπτώσεων έχουμε να κάνουμε με πίνακες 1 ή 2 διαστάσεων. Αυτό όμως δεν σημαίνει πως υπάρχει περιορισμός

04

ΧΡΗΣΙΜΑ ΣΤΟΙΧΕΙΑ

- ✓ Τα στοιχεία ενός δισδιάστατου πίνακα πρέπει να είναι του ίδιου τύπου
- ✓ Οι δείκτες που προσδιορίζουν μοναδικά τη θέση ενός στοιχείου στον 2D πίνακα μπορούν να αποτελούν οποιαδήποτε ακέραια έκφραση
- ✓ Όπως και με τους 1D πίνακες, έτσι και στους 2D η δήλωση των διαστάσεων τους γίνεται στο τμήμα δηλώσεων και δεν τροποποιείται(στατική δομή δεδομένων)
- ✓ Τετραγωνικός λέγεται ο πίνακας ο οποίος αποτελείται από τον ίδιο αριθμό γραμμών και στηλών. Στους πίνακες αυτούς ορίζονται 2 διαγώνιοι (κύρια και δευτερεύουσα)
 - ✓ Αν τα στοιχεία κάτω από την κύρια διαγώνιο είναι όλα 0 => Άνω τριγωνικός
 - ✓ Αν τα στοιχεία πάνω από την κύρια διαγώνιο είναι όλα 0 => Κάτω τριγωνικός
 - ✓ Αν ο τετραγωνικός πίνακας είναι και άνω και κάτω τριγωνικός => Διαγώνιος
- ✓ Ένας 2D πίνακας έχει τουλάχιστον 4 στοιχεία

03

ΠΡΟΣΠΕΛΑΣΗ 2D ΠΙΝΑΚΑ

- ✓ Για την προσπέλαση ενός 2D πίνακα χρησιμοποιούμε συνήθως 2 δομές επανάληψης, με σύμβαση το i να αντιπροσωπεύει τις γραμμές και το j τις στήλες του πίνακα
- ✓ Αυτό δεν σημαίνει ότι δεν μπορούμε να χρησιμοποιήσουμε την δομή επανάληψης ΟΣΟ για την προσπέλαση ενός 2D πίνακα, όπως θα δούμε στα παραδείγματα

```
! ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΤΑ ΓΡΑΜΜΕΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ ΓΡΑΜΜΕΣ
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ ΣΤΗΛΕΣ
    ΕΜΦΑΝΙΣΕ Α[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
! ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΤΑ ΣΤΗΛΕΣ
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ ΣΤΗΛΕΣ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ ΓΡΑΜΜΕΣ
    ΕΜΦΑΝΙΣΕ Α[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

1.2

-

ΔΙΣΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ (ΑΣΚΗΣΕΙΣ)

ΠΑΡΑΔΕΙΓΜΑ 1

ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ5

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j, ΠΙΝ[10,10]

ΛΟΓΙΚΕΣ: άνω, κάτω

ΑΡΧΗ

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    ΓΡΑΨΕ 'Δώσε Ακέραιο...'
    ΔΙΑΒΑΣΕ ΠΙΝ[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
άνω <- ΑΛΗΘΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    ΑΝ (i > j) ΚΑΙ (ΠΙΝ[i,j] <> 0) ΤΟΤΕ
      άνω <- ΨΕΥΔΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
κάτω <- ΑΛΗΘΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    ΑΝ (i < j) ΚΑΙ (ΠΙΝ[i,j] <> 0) ΤΟΤΕ
      κάτω <- ΨΕΥΔΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ (άνω=ΑΛΗΘΗΣ ΚΑΙ κάτω=ΨΕΥΔΗΣ) ΤΟΤΕ
  ΓΡΑΨΕ 'Ο πίνακας είναι διαγώνιος'
ΑΛΛΙΩΣ_ΑΝ άνω=ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'Ο πίνακας είναι άνω τριγωνικός'
ΑΛΛΙΩΣ_ΑΝ κάτω=ΑΛΗΘΗΣ ΤΟΤΕ
  ΓΡΑΨΕ 'Ο πίνακας είναι κάτω τριγωνικός'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Ο πίνακας δεν είναι ούτε άνω, ούτε κάτω, ούτε διαγώνιος'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Να αναπτυχθεί πρόγραμμα, το οποίο με δεδομένα τα στοιχεία ενός τετραγωνικού πίνακα 10*10, θα ελέγχει αν ο πίνακας είναι άνω τριγωνικός, κάτω τριγωνικός ή διαγώνιος

ΠΑΡΑΔΕΙΓΜΑ 2

ΑΡΧΗ

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
    ΓΡΑΨΕ 'Δώσε Αριθμητική τιμή...'
    ΔΙΑΒΑΣΕ ΠΙΝ[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

sum10throw <- 0
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
  sum10throw <- sum10throw + ΠΙΝ[10,j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το άθροισμα της 10ης γραμμής είναι: ', sum10throw / 20
```

```
sum15thcolumn <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  sum15thcolumn <- sum15thcolumn + ΠΙΝ[i,15]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Ο μέσος όρος των στοιχείων της 20ης στήλης: ', sum15thcolumn / 20
```

```
max_main_diagonal <- ΠΙΝ[1,1]
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  ΑΝ ΠΙΝ[i,j] > max_main_diagonal ΤΟΤΕ
    max_main_diagonal <- ΠΙΝ[i,j]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το μέγιστο στοιχείο της κυρίας διαγωνίου είναι: ', max_main_diagonal
```

```
min_secondary_diagonal <- ΠΙΝ[1,20]
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  ΑΝ ΠΙΝ[i, 20 + 1 - i] < min_secondary_diagonal ΤΟΤΕ
    min_secondary_diagonal <- ΠΙΝ[i, 20 + 1 - i]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Το ελάχιστο στοιχείο της δευτερεύουσας διαγωνίου είναι: ', min_secondary_diagonal
```

```
sum <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 20
    sum <- sum + ΠΙΝ[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
average <- sum / 400

ΓΡΑΨΕ 'Ο μέσος όρος του πίνακα είναι: ', average
```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Δημιουργήστε ένα πρόγραμμα το οποίο θα δέχεται τιμές από το χρήστη για έναν πίνακα 20*20. Στην συνέχεια θα εμφανίζει:

1. Το άθροισμα των στοιχείων της 10^{ης} γραμμής
2. Τον μέσο όρο της 15^{ης} στήλης του πίνακα
3. Την μέγιστη τιμή της κύριας διαγωνίου του πίνακα
4. Την ελάχιστη τιμή της δευτερεύουσας διαγωνίου του πίνακα
5. Τον μέσο όρο όλων των στοιχείων του πίνακα

1.2.1

-

- **ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΤΑ ΓΡΑΜΜΕΣ / ΣΤΗΛΕΣ**
- **ΜΕΓΙΣΤΟ / ΕΛΑΧΙΣΤΟ**
- **ΘΕΣΗ ΕΜΦΑΝΙΣΗΣ**

ΠΑΡΑΔΕΙΓΜΑ 3

Να αναπτυχθεί πρόγραμμα, το οποίο θα διαβάζει έναν πίνακα πραγματικών θετικών αριθμών 300 γραμμών και 200 στηλών. Στη συνέχεια θα υπολογίζει τον μέσο όρο κάθε γραμμής και θα αποθηκεύει τα αποτελέσματα σε έναν νέο πίνακα με το όνομα ΜΟ

ΑΡΧΗ

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 300
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 200
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΓΡΑΨΕ 'Δώσε θετικό αριθμό...'
      ΔΙΑΒΑΣΕ AP
      ΜΕΧΡΙΣ_ΟΤΟΥ AP>0
      ΑΡΙΘΜΟΙ[i,j] <- AP
      ΓΡΑΨΕ 'Ο αριθμός ', AP, ' καταχωρήθηκε στο κελί ', i,j
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Εισαγωγή 2D πίνακα

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 300
  sum_row <- 0
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 200
    sum_row <- sum_row + ΑΡΙΘΜΟΙ[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  average_row <- sum_row / 200
  ΜΟ[i] <- average_row
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Επεξεργασία κατά γραμμές και υπολογισμός μέσου όρου για κάθε γραμμή του 2D πίνακα

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 300
  ΓΡΑΨΕ 'Ο μέσος όρος της γραμμής ', i, ' είναι: ', ΜΟ[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Έξοδος πίνακα και εμφάνιση αποτελεσμάτων

ΠΑΡΑΔΕΙΓΜΑ 4

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j, points[15,30], πόντοι, sum_points, sum_player_points[15], max_points, max_index
ΧΑΡΑΚΤΗΡΕΣ: name[15]

ΑΡΧΗ

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
  ΓΡΑΨΕ 'Δώσε το όνομα του παίκτη του ΠΑΟ'
  ΔΙΑΒΑΣΕ name[i]
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
      ΓΡΑΨΕ 'Δώσε αριθμό πόντων για την ', j, ' αγωνιστική'
      ΔΙΑΒΑΣΕ πόντοι
      ΜΕΧΡΙΣ_ΟΤΟΥ πόντοι >= 0
        points[i,j] <- πόντοι
      ΓΡΑΨΕ 'Τα δεδομένα για την αγωνιστική καταχωρήθηκαν'
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

!Υπολογισμός αθροίσματος πόντων για κάθε παίκτη και για τις 30 αγωνιστικές

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 15
  sum_points <- 0
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 30
    sum_points <- sum_points + points[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  sum_player_points[i] <- sum_points
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

!Εύρεση index του max στον πίνακα των πόντων

```
max_points <- sum_player_points[1]
max_index <- 1
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 15
  ΑΝ sum_player_points[i] > max_points ΤΟΤΕ
    max_points <- sum_player_points[i]
    max_index <- i
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΓΡΑΨΕ 'Οι περισσότεροι πόντοι για τις 30 αγωνιστικές είναι ', sum_player_points[max_index]
ΓΡΑΨΕ 'Ο παίκτης που πέτυχε αυτούς τους πόντους είναι ο ', name[max_index]
```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Δημιουργήστε ένα πρόγραμμα το οποίο θα καταγράφει τους βαθμούς που πέτυχαν οι 15 παίκτες του ΠΑΟ στα 30 παιχνίδια της Euroleague 24'(Ακέραιοι αριθμοί). Επίσης θα διαβάζει και τα ονόματα των παικτών. Δημιουργήστε κατάλληλους πίνακες για την αποθήκευση αυτών των στοιχείων

Στην συνέχεια το πρόγραμμα θα πρέπει να εμφανίζει τον παίκτη που έχει πετύχει τους περισσότερους πόντους συνολικά στα 30 παιχνίδια και τους συνολικούς πόντους που πέτυχε

ΠΡΟΓΡΑΜΜΑ ΑΣΚΗΣΗ8

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j

ΠΡΑΓΜΑΤΙΚΕΣ: ΑΡΙΘΜΟΙ[80,160], max_current_row, min_current_column, MAX_ROW[80], MIN_COLUMN[160]

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 80

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 160

ΓΡΑΨΕ 'Δώσε αριθμό'

ΔΙΑΒΑΣΕ ΑΡΙΘΜΟΙ[i, j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 80

max_current_row <- ΑΡΙΘΜΟΙ[i, 1]

ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 160

ΑΝ ΑΡΙΘΜΟΙ[i, j] > max_current_row ΤΟΤΕ

max_current_row <- ΑΡΙΘΜΟΙ[i, j]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

MAX_ROW[i] <- max_current_row

ΓΡΑΨΕ 'Το μέγιστο για την γραμμή ', i, ' βρέθηκε και αποθηκεύτηκε'

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 160

min_current_column <- ΑΡΙΘΜΟΙ[1, j]

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 80

ΑΝ ΑΡΙΘΜΟΙ[i, j] < min_current_column ΤΟΤΕ

min_current_column <- ΑΡΙΘΜΟΙ[i, j]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

MIN_COLUMN[j] <- min_current_column

ΓΡΑΨΕ 'Το ελάχιστο για την στήλη ', j, ' βρέθηκε και αποθηκεύτηκε'

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 80

ΓΡΑΨΕ 'Το μέγιστο για την γραμμή ', i, ' είναι ', MAX_ROW[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 160

ΓΡΑΨΕ 'Το ελάχιστο για την στήλη ', j, ' είναι ', MIN_COLUMN[j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Να αναπτύξετε πρόγραμμα το οποίο θα διαβάζει πραγματικούς αριθμούς σε ένα πίνακα 80 * 160.

Στην συνέχεια, θα πρέπει να υπολογίζει το ελάχιστο στοιχείο κάθε στήλης και το μέγιστο στοιχείο κάθε γραμμής και θα τα τοποθετεί σε αντίστοιχους πίνακες

Τέλος θα εμφανίζει τα αποτελέσματα των πινάκων που προέκυψαν

ΠΑΡΑΔΕΙΓΜΑΤΑ

5

```

ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 200
      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΓΡΑΨΕ 'Δώσε βαθμολογία...'
        ΔΙΑΒΑΣΕ ΒΑΘΜ
        ΜΕΧΡΙΣ_ΟΤΟΥ ΒΑΘΜ >=0 ΚΑΙ ΒΑΘΜ <= 20
        GRADES[i,j] <- ΒΑΘΜ
        ΓΡΑΨΕ 'Η βαθμολογία καταχωρήθηκε επιτυχώς'
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

counter_20 <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 200
    ΑΝ GRADES[i,j] = 20 ΤΟΤΕ
      counter_20 <- counter_20 + 1
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ counter_20 > 0 ΤΟΤΕ
  ΓΡΑΨΕ 'Η βαθμολογία 20 εμφανίστηκε ', counter_20, ' φορές στον πίνακα'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Δεν υπάρχει βαθμολογία 20 στον πίνακα'
ΤΕΛΟΣ_ΑΝ

```

```

min_grade <- GRADES[1,1]
index_row_min_first <- 1
index_row_column_first <- 1
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 200
    ΑΝ GRADES[i,j] < min_grade ΤΟΤΕ
      min_grade <- GRADES[i,j]
      index_row_min_first <- i
      index_row_column_first <- j
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Η μικρότερη βαθμολογία στον πίνακα είναι ', GRADES[index_row_min_first, index_row_column_first]
ΓΡΑΨΕ 'Η πρώτη εμφάνιση της βαθμολογίας αυτής είναι στη θέση ', index_row_min_first, index_row_column_first

```

Για πίνακα ακεραίων βαθμολογιών 100*200 να αναπτύξετε πρόγραμμα το οποίο

1. Πόσες φορές θα εμφανίζεται ο αριθμός 20 στον δισδιάστατο πίνακα;
2. Σε ποια θέση εμφανίζεται η ελάχιστη βαθμολογία του πίνακα για 1^η φορά;
3. Σε ποια θέση εμφανίζεται η ελάχιστη βαθμολογία του πίνακα για τελευταία φορά;
4. Τον μέσο όρο των μεγίστων κάθε γραμμής του πίνακα

ΠΑΡΑΔΕΙΓΜΑΤΑ

6

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 200
    ΑΝ GRADES[i,j] = min_grade ΤΟΤΕ
      index_row_min_last <- i
      index_column_min_last <- j
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ index_row_min_last = index_row_min_first ΚΑΙ index_column_min_last = index_row_column_first ΤΟΤΕ
  ΓΡΑΨΕ 'Το ελάχιστο αυτό εμφανίστηκε μόνο μια φορά'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Η τελευταία εμφάνιση του ελαχίστου βρίσκεται στην θέση ', index_row_min_last, index_column_min_last
ΤΕΛΟΣ_ΑΝ

```

```

sum_max_all_rows <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
  max_current_row <- GRADES[i,1]
  ΓΙΑ j ΑΠΟ 2 ΜΕΧΡΙ 200
    ΑΝ GRADES[i,j] > max_current_row ΤΟΤΕ
      max_current_row <- GRADES[i,j]
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  sum_max_all_rows <- sum_max_all_rows + max_current_row
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Ο μέσος όρος των μεγίστων όλων των γραμμών είναι ', sum_max_all_rows / 100

```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ