Ασκήσεις Πίνακες - (ΝΕΕΣ ασκήσεις 2) NEO

Άσκηση 1. Να αναπτυχθεί αλγόριθμος που θα διαβάζει σε μονοδιάστατο πίνακα την ημερήσια μέτρηση του διοξειδίου του άνθρακα (αριθμός μεταξύ του 0 και του 10) για ένα σημείο της Αθήνας, για έναν μήνα, και θα εκτυπώνει:
α. τις ημέρες που η μέτρηση ήταν μεγαλύτερη από την προηγούμενη και την επόμενη ημέρα, καθώς και το πλήθος αυτών των ημερών,
β. τις ημέρες που παρατηρήθηκε ρυθμός αύξησης μεγαλύτερος από 15%, καθώς και τον μεγαλύτερο ρυθμό αύξησης και την ημέρα που επιτεύχθηκε.

Λύση

```
Αλγόριθμος Μελέτη_μόλυνσης
 Για i από 1 μέχρι 30
  Αρχή_επανάληψης
   Διάβασε ΜΟΛΥΝΣΗ[i]
  Μέχρις_ότου (ΜΟΛΥΝΣΗ[i] >= 0) και (ΜΟΛΥΝΣΗ[i] <= 10)
 Τέλος_επανάληψης
 ημέρες ← 0       ! ερώτημα α
 ! πρέπει να υπάρχει προηγούμενη αλλά και επόμενη ημέρα
 Για i από 2 μέχρι 29
  Αν (ΜΟΛΥΝΣΗ[i] > ΜΟΛΥΝΣΗ[i + 1]) και (ΜΟΛΥΝΣΗ[i] > ΜΟΛΥΝΣΗ[i - 1])
τότε
   Εκτύπωσε "Ημέρα", i
   ημέρες ← ημέρες + 1
  Τέλος_αν
 Τέλος_επανάληψης
 Εκτύπωσε "Πλήθος ημερών", ημέρες
 μέγιστος ← -1      ! πολύ μικρή τιμή, ερώτημα β
 Για i από 2 μέχρι 30     ! πρέπει να υπάρχει προηγούμενη ημέρα
  ρυθμός ← ((ΜΟΛΥΝΣΗ[i] - ΜΟΛΥΝΣΗ[i - 1]) / ΜΟΛΥΝΣΗ[i - 1]) * 100
  Αν ρυθμός > 15 τότε
   Εκτύπωσε "Ημέρα", i
  Τέλος_αν
  Αν ρυθμός > μέγιστος τότε
   μέγιστος ← ρυθμός
   ημέρα ← i
  Τέλος_αν
 Τέλος_επανάληψης
 Εκτύπωσε "Μέγιστο την ημέρα", ημέρα
Τέλος Μελέτη_μόλυνσης
```

Άσκηση 2. Να αναπτυχθεί αλγόριθμος ο οποίος, με δεδομένα τα στοιχεία ενός τετραγωνικού πίνακα διαστάσεων NxN, θα ελέγχει αν ο πίνακας είναι άνω τριγωνικός, κάτω τριγωνικός ή διαγώνιος.
Παρατήρηση: α. Ένας πίνακας χαρακτηρίζεται ως άνω τριγωνικός όταν όλα τα στοιχεία που βρίσκονται κάτω της κύριας διαγωνίου είναι μηδέν. Τα στοιχεία αυτά είναι τα περιεχόμενα των κελιών A[i, j], όπου i > j.
β. Ένας πίνακας χαρακτηρίζεται ως κάτω τριγωνικός όταν όλα τα στοιχεία που βρίσκονται άνω της κύριας διαγωνίου είναι μηδέν. Τα στοιχεία αυτά είναι τα περιεχόμενα των κελιών A[i, j], όπου i < j.
γ. Τα κελιά A[i, j], όπου i = j, ανήκουν στην κύρια διαγώνιο.
δ. Ένας πίνακας χαρακτηρίζεται ως διαγώνιος αν είναι ταυτόχρονα άνω και κάτω τριγωνικός.

Λύση

```
Αλγόριθμος Τριγωνικός_πίνακας
 Δεδομένα // N, A //
 ! υποθέτουμε ότι ο πίνακας είναι άνω τριγωνικός
 άνω ← Αληθής
 Για i από 1 μέχρι N
  Για j από 1 μέχρι N   ! θα μπορούσε να είναι:  Για j από 1 μέχρι i
- 1
   ! αν βρεθεί έστω και ένα στοιχείο <> 0, τον αποχαρακτηρίζουμε
   Αν (A[i, j] <> 0) και (i > j) τότε
   άνω ← Ψευδής
   Τέλος_αν
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 ! υποθέτουμε ότι ο πίνακας είναι κάτω τριγωνικός
 κάτω ← Αληθής
 Για i από 1 μέχρι N
  Για j από 1 μέχρι N   ! θα μπορούσε να είναι:  Για j από i + 1
μέχρι N
   ! αν βρεθεί έστω και ένα στοιχείο <> 0, τον αποχαρακτηρίζουμε
   Αν (A[i, j] <> 0) και (i < j) τότε
    κάτω ← Ψευδής
   Τέλος_αν
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 Αν (άνω = Αληθής) και (κάτω = Αληθής) τότε
  Εκτύπωσε "Ο πίνακας είναι διαγώνιος"
 Αλλιώς_αν (άνω = Αληθής) τότε
  Εκτύπωσε "Ο πίνακας είναι άνω τριγωνικός"
 Αλλιώς_αν (κάτω = Αληθής) τότε
  Εκτύπωσε "Ο πίνακας είναι κάτω τριγωνικός"
 Αλλιώς
  Εκτύπωσε "Ο πίνακας δεν έχει καμία από τις ιδιότητες της εκφώνησης"
 Τέλος_αν
Τέλος Τριγωνικός_πίνακας
```

Άσκηση 3. Ο Ελληνικός Οργανισμός Τουρισμού αποφάσισε να διεξαγάγει μια έρευνα για τα ελληνικά ξενοδοχεία. Στην έρευνα συμμετέχουν τα 7000 ξενοδοχεία της επικράτειας, και γι' αυτά καταγράφονται οι μηνιαίες εισπράξεις για το περασμένο έτος. Να αναπτυχθεί αλγόριθμος που:

α. Θα διαβάζει τα απαραίτητα στοιχεία.

β. Θα υπολογίζει τις ετήσιες εισπράξεις κάθε ξενοδοχείου και θα τις εκχωρεί σε έναν νέο πίνακα.

γ. Θα εκτυπώνει το όνομα του ξενοδοχείου με τις περισσότερες εισπράξεις.

δ. Θα υπολογίζει και θα εκτυπώνει το όνομα κάθε ξενοδοχείου συνοδευόμενο από το πλήθος των μηνών που λειτούργησε (δηλαδή είχε έσοδα) την περασμένη χρονιά.

ε. Θα δημιουργεί και θα εκτυπώνει τον πίνακα ΤΡΙΜΗΝΑ[4], που περιέχει τις εισπράξεις που παρουσιάστηκαν ανά τρίμηνο. .

Λύση

```
Αλγόριθμος ΕΟΤ
 Για i από 1 μέχρι 7000      ! ερώτημα α
  Διάβασε ΟΝΟΜΑ[i]
  Για j από 1 μέχρι 12
   Διάβασε ΕΙΣΠΡΑΞΕΙΣ[i, j]
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 Για i από 1 μέχρι 7000      ! ερώτημα β
  άθροισμα ← 0
```

```
  Για j από 1 μέχρι 12
   άθροισμα ← άθροισμα + ΕΙΣΠΡΑΞΕΙΣ[i, j]
  Τέλος_επανάληψης
  ΕΤΗΣΙΕΣ_ΕΙΣΠΡΑΞΕΙΣ[i] ← άθροισμα
 Τέλος_επανάληψης
 ! ... ερώτημα γ: εύρεση μεγίστου και θέσης του στον πίνακα
ΕΤΗΣΙΕΣ_ΕΙΣΠΡΑΞΕΙΣ
 Εκτύπωσε "Το ξενοδοχείο με τις περισσότερες εισπράξεις", ΟΝΟΜΑ[θέση]
 Για i από 1 μέχρι 7000     ! ερώτημα δ
  μη_μηδεν ← 0
  Για j από 1 μέχρι 12
   Αν ΕΙΣΠΡΑΞΕΙΣ[i, j] <> 0 τότε
    μη_μηδεν ← μη_μηδεν + 1   ! δεν χρησιμοποιούμε πίνακα μετρητών
   Τέλος_αν
  Τέλος_επανάληψης
  Εκτύπωσε " Το ξενοδοχείο", ΟΝΟΜΑ[i], "λειτούργησε", μη_μηδεν,
"μήνες"
 Τέλος_επανάληψης
 Για j από 1 μέχρι 4      ! ερώτημα ε
  άθροισμα ← 0
  αρχή_τριμήνου ← 3 * (j - 1)  ! πού ξεκινάει το τρίμηνο j
  Για i από 1 μέχρι 7000
   Για k από 1 μέχρι 3    ! οι 3 μήνες του τριμήνου
     άθροισμα ← άθροισμα + ΕΙΣΠΡΑΞΕΙΣ[i, αρχή_τριμήνου + k]
   Τέλος_επανάληψης
  Τέλος_επανάληψης
  ΤΡΙΜΗΝΑ[j] ← άθροισμα
 Τέλος_επανάληψης
 Για j από 1 μέχρι 4
  Εκτύπωσε ΤΡΙΜΗΝΑ[j]
 Τέλος_επανάληψης
Τέλος ΕΟΤ
```

Άσκηση 4. Η εταιρεία Αρβίλογλου διαθέτει μια ομάδα πωλητών (Ν στο πλήθος) σε όλη την Ελλάδα. Καταγράφονται σε μονοδιάστατο πίνακα ΟΝΟΜΑ[Ν] τα ονόματα των πωλητών της εταιρείας και σε δισδιάστατο πίνακα ΠΩΛΗΣΕΙΣ[Ν, 12] οι μηνιαίες πωλήσεις που πέτυχαν τον περασμένο χρόνο. Να αναπτυχθεί αλγόριθμος που:

α. Θα διαβάζει τα απαραίτητα στοιχεία και θα τα αποθηκεύει στους παραπάνω πίνακες.

β. Θα δημιουργεί τον πίνακα ΣΥΝ_ΠΩΛΗΣΕΙΣ που θα περιέχει τις συνολικές ετήσιες πωλήσεις κάθε πωλητή.

γ. Θα δημιουργεί τον πίνακα ΚΑΛΟΙ που θα περιέχει τα ονόματα των πωλητών με συνολικές πωλήσεις περισσότερες από 1500 €.

δ. Θα δημιουργεί τον πίνακα ΜΠΟΝΟΥΣ που θα περιέχει το μπόνους κάθε πωλητή. Το μπόνους υπολογίζεται με βάση τις πωλήσεις κάθε πωλητή ως εξής:

Πωλήσεις S (σε €)	Μπόνους (%) επί των πωλήσεων
$0 \le S \le 200$	11
$200 < S \le 500$	15
$500 < S$	20

Ο πίνακας ΜΠΟΝΟΥΣ αποτελεί την έξοδο του αλγορίθμου.

Λύση

```
Αλγόριθμος Αρβίλογλου
```

```
 Διάβασε N
 Για i από 1 μέχρι N
  Διάβασε ΟΝΟΜΑ[i]
  Για j από 1 μέχρι 12
   Αρχή_επανάληψης
    Διάβασε ΠΩΛΗΣΕΙΣ[i, j]
   Μέχρις_ότου ΠΩΛΗΣΕΙΣ[i, j] >= 0   ! έλεγχος δεδομένων
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 Για i από 1 μέχρι N          ! ερώτημα β
  άθροισμα ← 0
  Για j από 1 μέχρι 12
   άθροισμα ← άθροισμα + ΠΩΛΗΣΕΙΣ[i, j]
  Τέλος_επανάληψης
  ΣΥΝ_ΠΩΛΗΣΕΙΣ[i] ← άθροισμα
 Τέλος_επανάληψης
 Π ← 0   ! δείκτης νέου πίνακα ΚΑΛΟΙ, που δεν είναι παράλληλος με
τους άλλους
 Για i από 1 μέχρι N
  Αν ΣΥΝ_ΠΩΛΗΣΕΙΣ[i] > 1500 τότε
   Π ← Π + 1
   ΚΑΛΟΙ[Π] ← ΟΝΟΜΑ[i]
  Τέλος_αν
 Τέλος_επανάληψης
 Για i από 1 μέχρι N          ! ερώτημα δ
  Αν ΣΥΝ_ΠΩΛΗΣΕΙΣ[i] <= 200 τότε
   ΜΠΟΝΟΥΣ[i] ← 11 / 100 * ΣΥΝ_ΠΩΛΗΣΕΙΣ[i]
  Αλλιώς_αν ΣΥΝ_ΠΩΛΗΣΕΙΣ[i] <= 500 τότε
   ΜΠΟΝΟΥΣ[i] ← 15 / 100 * ΣΥΝ_ΠΩΛΗΣΕΙΣ[i]
  Αλλιώς            ! > 500
   ΜΠΟΝΟΥΣ[i] ← 20 / 100 * ΣΥΝ_ΠΩΛΗΣΕΙΣ[i]
  Τέλος_αν
 Τέλος_επανάληψης
 Αποτελέσματα // Π, ΚΑΛΟΙ, ΜΠΟΝΟΥΣ //
Τέλος Αρβίλογλου
```

Άσκηση 5. Η εταιρεία DeltaTime Systems εξοπλίζει έναν αθλητικό αγώνα με σύστημα χρονομέτρησης. Το σύστημα δημιουργεί δύο παράλληλους πίνακες: τον πίνακα ΟΝΟΜΑ, με το όνομα κάθε αθλητή, και τον πίνακα ΚΑΤΑΤΑΞΗ, του οποίου η πρώτη θέση περιέχει τον χρόνο που χρειάστηκε να τερματίσει ο πρώτος αθλητής (σε δευτερόλεπτα) και κάθε επόμενη θέση περιέχει τη διαφορά του συγκεκριμένου αθλητή από τον προηγούμενο. Να αναπτυχθεί αλγόριθμος που, με δεδομένους τους παραπάνω πίνακες για έναν αγώνα 5000 μέτρων όπου συμμετείχαν 52 αθλητές, θα εκτελεί τις εξής ενέργειες:

α. Θα διαβάζει το όνομα ενός αθλητή και θα εκτυπώνει τον χρόνο που χρειάστηκε για να τερματίσει.

β. Θα εκτυπώνει τον χρόνο που διήρκεσε η κούρσα (χρόνος τελευταίου αθλητή).

γ. Θα εκτυπώνει το πλήθος των αθλητών που είχαν τερματίσει στα μισά της κούρσας.

Λύση

```
Αλγόριθμος DeltaSystems
 Δεδομένα // ΟΝΟΜΑ, ΚΑΤΑΤΑΞΗ //
 Διάβασε όνομα_αναζ
 ! ερώτημα α : θα εντοπίσουμε θέση στον πίνακα
 ! σειριακή αναζήτηση στον πίνακα ΟΝΟΜΑ με κλειδί το όνομα_αναζ
 βρέθηκε ← Ψευδής
 θέση ← 0
 i ← 1
 Όσο (βρέθηκε = Ψευδής) και (i <= 52) επανάλαβε
```

```
  Αν ΟΝΟΜΑ[i] = όνομα_αναζ τότε
   βρέθηκε ← Αληθής
   θέση ← i
  Αλλιώς
   i ← i + 1
  Τέλος_αν
 Τέλος_επανάληψης
 Αν θέση <> 0 τότε
  χρόνος_αθλητή ← 0
  Για k από 1 μέχρι θέση
   χρόνος_αθλητή ← χρόνος_αθλητή + ΚΑΤΑΤΑΞΗ[k]
  Τέλος_επανάληψης
  Εκτύπωσε "Ο χρόνος του αθλητή είναι", χρόνος_αθλητή
 Αλλιώς     ! δεν βρέθηκε
  Εκτύπωσε "Δεν υπάρχει αθλητής με αυτό το όνομα"
 Τέλος_αν
 ! ερώτημα β: θα προσθέσουμε όλα τα στοιχεία του πίνακα ΚΑΤΑΤΑΞΗ
 χρόνος_τελευταίου ← 0
 Για k από 1 μέχρι 52
  χρόνος_τελευταίου ← χρόνος_τελευταίου + ΚΑΤΑΤΑΞΗ[k]
 Τέλος_επανάληψης
 Εκτύπωσε "Ο χρόνος του τελευταίου αθλητή είναι", χρόνος_τελευταίου
 ! ερώτημα γ
 μέσος_χρόνος ← χρόνος_τελευταίου / 2
 done ← Ψευδής
 αθλητές ← 0
 χρόνος ← 0
 Όσο done = Ψευδής επανάλαβε
  Αν χρόνος + ΚΑΤΑΤΑΞΗ[αθλητές + 1] <= μέσος_χρόνος τότε
   αθλητές ← αθλητές + 1
   χρόνος ← χρόνος + ΚΑΤΑΤΑΞΗ[αθλητές]
  Αλλιώς
   done ← Ψευδής
  Τέλος_αν
 Τέλος_επανάληψης
 Εκτύπωσε "Το πλήθος των αθλητών είναι", αθλητές
Τέλος DeltaSystems
```

Άσκηση 6. Γνωστή γκαλερί των Αθηνών χρησιμοποιεί τις ακόλουθες δομές δεδομένων για τη διαχείριση των έργων τέχνης:

- Πίνακας ΕΡΓΟ_ΤΕΧΝΗΣ[N, 2], του οποίου η πρώτη στήλη περιέχει το όνομα του καλλιτέχνη που δημιούργησε το αντίστοιχο έργο τέχνης και η δεύτερη στήλη την περιγραφή του.

- Παράλληλος πίνακας ΕΤΟΣ[N], που περιέχει το έτος δημιουργίας του έργου τέχνης.

- Παράλληλος πίνακας ΤΙΜΗ[N], που περιέχει την τιμή του συγκεκριμένου έργου.

Να αναπτυχθεί αλγόριθμος, ο οποίος με δεδομένα τα στοιχεία των πινάκων:

α. Θα διαβάζει το όνομα ενός ζωγράφου και θα εκτυπώνει όλα τα έργα του που υπάρχουν στη γκαλερί.

β. Θα διαβάζει το όνομα ενός ζωγράφου και την περιγραφή ενός έργου του και θα εκτυπώνει την τιμή του, εφόσον διατίθεται το έργο στη γκαλερί.

γ. Θα διαβάζει ένα ποσό και θα εκτυπώνει τα έργα της γκαλερί που μπορούν να αγοραστούν με το ποσό αυτό.

δ. Θα εκτυπώνει όλα τα έργα που υπάρχουν στη γκαλερί, από το παλαιότερο προς το νεότερο.

Λύση

```
Αλγόριθμος Γκαλερί
 Δεδομένα // N, ΕΡΓΟ_ΤΕΧΝΗΣ, ΕΤΟΣ, ΤΙΜΗ //
```

```
 Διάβασε όνομα_καλλιτέχνη
 ! ερώτημα α: αναζητάμε όλα τα έργα του καλλιτέχνη, άρα
χρησιμοποιούμε τη Για
 Για i από 1 μέχρι N
  Αν ΕΡΓΟ_ΤΕΧΝΗΣ[i, 1] = όνομα_καλλιτέχνη τότε
   Εκτύπωσε ΕΡΓΟ_ΤΕΧΝΗΣ[i, 2], ΕΤΟΣ[i], ΤΙΜΗ[i]
  Τέλος_αν
 Τέλος_επανάληψης
 ! ερώτημα β: μοναδικό έργο, άρα σειριακή αναζήτηση
 Διάβασε όνομα_καλλιτέχνη, τίτλος_έργου
 βρέθηκε ← Ψευδής
 θέση ← 0
 i ← 1
 Όσο (βρέθηκε = Ψευδής) και (i <= N) επανάλαβε
  Αν (ΕΡΓΟ_ΤΕΧΝΗΣ[i, 1] = όνομα_καλλιτέχνη) και (ΕΡΓΟ_ΤΕΧΝΗΣ[i, 2] =
τίτλος_έργου) τότε
   βρέθηκε ← Αληθής
   θέση ← i
  Αλλιώς
   i ← i + 1
  Τέλος_αν
 Τέλος_επανάληψης
 Αν θέση <> 0 τότε
  Εκτύπωσε ΤΙΜΗ[θέση]
 Αλλιώς
  Εκτύπωσε "Δεν υπάρχει αυτό το έργο τέχνης στη γκαλερί"
 Τέλος_αν
 ! ερώτημα β
 Διάβασε ποσό
 Για i από 1 μέχρι N
  Αν ΤΙΜΗ[i] <= ποσό τότε
   Εκτύπωσε ΕΡΓΟ_ΤΕΧΝΗΣ[i, 1], ΕΡΓΟ_ΤΕΧΝΗΣ[i, 2], ΕΤΟΣ[i], ΤΙΜΗ[i]
  Τέλος_αν
 Τέλος_επανάληψης
 Για i από 2 μέχρι N
  Για j από N μέχρι i με_βήμα -1
   ! φθίνουσα διάταξη
   Αν ΕΤΟΣ[j - 1] < ΕΤΟΣ[j] τότε
    Αντιμετάθεσε ΕΤΟΣ[j - 1], ΕΤΟΣ[j]
    Αντιμετάθεσε ΕΡΓΟ_ΤΕΧΝΗΣ[j - 1, 1], ΕΡΓΟ_ΤΕΧΝΗΣ[j, 1]
    Αντιμετάθεσε ΕΡΓΟ_ΤΕΧΝΗΣ[j - 1, 2], ΕΡΓΟ_ΤΕΧΝΗΣ[j, 2]
    Αντιμετάθεσε ΤΙΜΗ[j - 1], ΤΙΜΗ[j]
   Τέλος_αν
  Τέλος_επανάληψης
 Τέλος_επανάληψης
Τέλος Γκαλερί
```

Άσκηση 7. Στη δεξίωση του πρέσβη μεγάλου ευρωπαϊκού κράτους στην Τενεούπολη έχει καταρτιστεί λίστα καλεσμένων. Στον πίνακα ΟΝΟΜΑ καταχωρείται το όνομα κάθε καλεσμένου και στον πίνακα ΤΡΑΠΕΖΙ καταχωρείται ο αριθμός του τραπεζιού όπου τοποθετείται. Σημειώνεται ότι τα τραπέζια διαθέτουν 10 θέσεις και ότι το συνολικό πλήθος των καλεσμένων είναι 1500. Να αναπτυχθεί αλγόριθμος που με δεδομένα τα στοιχεία των πινάκων:

α. Θα διαβάζει ένα όνομα ενός καλεσμένου και θα εκτυπώνει το τραπέζι στο οποίο έχει τοποθετηθεί.

β. Θα διαβάζει τον αριθμό ενός τραπεζιού και θα εκτυπώνει τη λίστα των ατόμων που κάθονται σε αυτό.

γ. Θα εκτυπώνει το όνομα κάθε καλεσμένου με αλφαβητική σειρά, καθώς και το τραπέζι του.

δ. Θα εκτυπώνει τα ονόματα των καλεσμένων της δεξίωσης ανά τραπέζι.

Λύση

```
Αλγόριθμος Δεξίωση
 Δεδομένα // ΟΝΟΜΑ, ΤΡΑΠΕΖΙ //
 ! ... το αναζητούμενο στοιχείο εντοπίζεται μία φορά στον πίνακα
 Διάβασε όνομα_αναζ
 ! ... σειριακή αναζήτηση της τιμή όνομα_αναζ στον πίνακα ΟΝΟΜΑ
 Αν θέση <> 0 τότε
  Εκτύπωσε "Ο αριθμός τραπεζιού είναι", ΤΡΑΠΕΖΙ[θέση]
 Αλλιώς
  Εκτύπωσε "Δεν υπάρχει καλεσμένος με αυτό το όνομα"
 Τέλος_αν
 Διάβασε αριθμός_τραπεζιού
 ! ... ερώτημα β: για ένα τραπέζι πολλά αποτελέσματα, άρα δομή Για
 Για i από 1 μέχρι 1500
  Αν ΤΡΑΠΕΖΙ[i] = αριθμός_τραπεζιού τότε
   Εκτύπωσε ΟΝΟΜΑ[i]
  Τέλος_αν
 Τέλος_επανάληψης
 ! ... ερώτημα γ: αύξουσα ταξινόμηση του πίνακα ΟΝΟΜΑ, με ταυτόχρονη
αντιμετάθεση
 ! του πίνακα ΤΡΑΠΕΖΙ
 Για i από 1 μέχρι 1500
  Εκτύπωσε ΟΝΟΜΑ[i], ΤΡΑΠΕΖΙ[i]
 Τέλος_επανάληψης
 ! ... ερώτημα δ: αύξουσα ταξινόμηση του πίνακα ΤΡΑΠΕΖΙ με ταυτόχρονη
αντιμετάθεση
 ! του πίνακα ΟΝΟΜΑ
 Για i από 1 μέχρι 1500
  Εκτύπωσε ΟΝΟΜΑ[i], ΤΡΑΠΕΖΙ[i]
 Τέλος_επανάληψης
Τέλος Δεξίωση
```

Άσκηση 8. Ένας φανατικός συλλέκτης δίσκων βινυλίου αποφάσισε να αναπτύξει αλγόριθμο, ώστε να μπορεί να επεξεργαστεί στατιστικά τη συλλογή του. Διαθέτει 2500 δίσκους και για καθέναν από αυτούς επιθυμεί να καταχωρεί τίτλο, καλλιτέχνη και έτος κυκλοφορίας. Να αναπτυχθεί αλγόριθμος που θα διαβάζει σε τρεις μονοδιάστατους πίνακες τα παραπάνω στοιχεία και στη συνέχεια:

α. Θα διαβάζει έναν αριθμό που αντιστοιχεί σε έτος (με τιμή 1900 – 2005) και θα εκτυπώνει το πλήθος των δίσκων που κυκλοφόρησαν εκείνη τη χρονιά.

β. Θα εκτυπώνει το πλήθος και τα ονόματα των διαφορετικών καλλιτεχνών για τους οποίους υπάρχουν δίσκοι στη συλλογή.

Λύση

```
Αλγόριθμος Συλλογή_δίσκων
 Για i από 1 μέχρι 2500
  Διάβασε ΤΙΤΛΟΣ[i], ΚΑΛΛΙΤΕΧΝΗΣ[i], ΕΤΟΣ_ΚΥΚΛΟΦ[i]
 Τέλος_επανάληψης
 Διάβασε έτος
 δίσκοι ← 0
 Για i από 1 μέχρι 2500
  Αν ΕΤΟΣ_ΚΥΚΛΟΦ[i] = έτος τότε
   δίσκοι ← δίσκοι + 1
  Τέλος_αν
 Τέλος_επανάληψης
 Εκτύπωσε "Οι δίσκοι που κυκλοφόρησαν αυτό το έτος είναι", δίσκοι
 μετρητής ← 0        ! μετρητής του πίνακα ΜΟΝΑΔΙΚΟΙ
 Για k από 1 μέχρι 2500
```

```
  υπάρχει ← Ψευδής      ! αναζήτηση στον πίνακα ΜΟΝΑΔΙΚΟΙ
  i ← 1
  Όσο (υπάρχει = Ψευδής) και (i <= μετρητής) επανάλαβε
   Αν ΚΑΛΛΙΤΕΧΝΗΣ[k] = ΜΟΝΑΔΙΚΟΙ[i] τότε
    υπάρχει ← Αληθής
   Αλλιώς
    i ← i + 1
   Τέλος_αν
  Τέλος_επανάληψης
  Αν υπάρχει = Ψευδής τότε   ! αν δεν υπάρχει, τον τοποθετούμε
   μετρητής ← μετρητής + 1
   ΜΟΝΑΔΙΚΟΙ[μετρητής] ← ΚΑΛΛΙΤΕΧΝΗΣ[k]
  Τέλος_αν
 Τέλος_επανάληψης
 Για i από 1 μέχρι μετρητής    ! εκτυπώνουμε τον πίνακα ΜΟΝΑΔΙΚΟΙ
  Εκτύπωσε ΜΟΝΑΔΙΚΟΙ[i]
 Τέλος_επανάληψης
 ! εναλλακτικά για το τελευταίο ερώτημα:
 ! αύξουσα ταξινόμηση πίνακα ΚΑΛΛΙΤΕΧΝΕΣ με αντιμετάθεση των άλλων
δυο
 ΜΟΝΑΔΙΚΟΙ[1] ← ΚΑΛΛΙΤΕΧΝΗΣ[1]  ! περνάμε τον πρώτο
 μετρητής ← 1          ! μετρητής του πίνακα ΜΟΝΑΔΙΚΟΙ
 Για i από 2 μέχρι 2500
  Αν ΚΑΛΛΙΤΕΧΝΗΣ[i] <> ΚΑΛΛΙΤΕΧΝΗΣ[i - 1] τότε
   μετρητής ← μετρητής + 1
   ΜΟΝΑΔΙΚΟΙ[μετρητής] ← ΚΑΛΛΙΤΕΧΝΗΣ[i]
  Τέλος_αν
 Τέλος_επανάληψης
Τέλος Συλλογή_δίσκων
```

Άσκηση 9. Με δεδομένη την τυχερή εξάδα του ΛΟΤΤΟ, να αναπτυχθεί αλγόριθμος που θα διαβάζει καθεμιά από τις 650000 στήλες - εξάδες που παίχτηκαν στα πρακτορεία και θα εκτυπώνει αν η στήλη νίκησε. Στην περίπτωση που η στήλη δεν νίκησε, θα εκτυπώνει πόσες επιλογές ήταν επιτυχημένες.

Λύση

```
Αλγόριθμος ΛΟΤΤΟ
 Δεδομένα // ΝΙΚΗΤΡΙΑ //    ! νικήτρια στήλη – 6 αριθμοί
 Για στήλες από 1 μέχρι 650000  ! θα εξετάσουμε όλες τις στήλες, μία
προς μία
  Για μετρητής από 1 μέχρι 6  ! διάβασε τη νέα στήλη
   Διάβασε ΣΤΗΛΗ[μετρητής]
  Τέλος_επανάληψης
  επιτυχίες ← 0
  ! θα ελέγξουμε καθένα από τα στοιχεία της στήλης, ψάχνοντας αν
βρίσκεται
  ! μέσα στη νικήτρια στήλη
  Για μετρητής από 1 μέχρι 6
   βρέθηκε ← Ψευδής
   θέση ← 0
   i ← 1
   Όσο (βρέθηκε = Ψευδής) και (i <= 6) επανάλαβε
    Αν ΝΙΚΗΤΡΙΑ[i] = ΣΤΗΛΗ[μετρητής] τότε
     βρέθηκε ← Αληθής
     θέση ← i
    Αλλιώς
     i ← i + 1
    Τέλος_αν
```

```
    Τέλος_επανάληψης
    Αν θέση <> 0 τότε
     επιτυχίες ← επιτυχίες + 1
    Τέλος_αν
   Τέλος_επανάληψης
   Αν επιτυχίες = 6 τότε
    Εκτύπωσε "Η στήλη αυτή κερδίζει"
   Αλλιώς
    Εκτύπωσε επιτυχίες
   Τέλος_αν
  Τέλος_επανάληψης
Τέλος ΛΟΤΤΟ
```

Άσκηση 10. Στον τελικό του τριπλούν προκρίνονται οι 8 πρώτοι αθλητές των ημιτελικών. Η εταιρεία μηχανογράφησης των αγώνων χρησιμοποιεί τον πίνακα ΟΝΟΜΑ_ΗΜ[30] με τα ονόματα των αθλητών που συμμετέχουν στον ημιτελικό αγώνα, καθώς και τον πίνακα ΕΠΙΔΟΣΕΙΣ_ΗΜ[30, 3], που περιέχει τις επιδόσεις των αθλητών με στόχο την πρόκριση στον τελικό. Να αναπτυχθεί αλγόριθμος ο οποίος:

α. Θα διαβάζει τα στοιχεία των παραπάνω πινάκων και θα εκτυπώνει τα ονόματα των αθλητών που προκρίνονται στον τελικό.

β. Θα δημιουργεί τους νέους πίνακες ΟΝΟΜΑ_ΤΕΛ[8] με τα ονόματα των αθλητών που συμμετέχουν στον τελικό και ΕΠΙΔΟΣΕΙΣ_ΤΕΛ[8, 6], του οποίου οι 3 πρώτες στήλες περιέχουν τις επιδόσεις στους προκριματικούς για τους αντίστοιχους αθλητές του ημιτελικού. Στη συνέχεια, θα διαβάζει και τις επόμενες 3 επιδόσεις (στήλες 4, 5, 6) για κάθε αθλητή και θα εκτυπώνει τους αθλητές που παίρνουν μετάλλια. (Δίνεται ότι υπάρχουν ακριβώς 8 αθλητές που συμμετέχουν στον τελικό.)

Παρατήρηση: Δίνεται ότι τελικά οι 2 πρώτοι αθλητές έχουν διαφορετικές επιδόσεις.

Λύση

```
Αλγόριθμος Τρίαθλο
 Για i από 1 μέχρι 30
  Διάβασε ΟΝΟΜΑ_ΗΜ[i]
 Τέλος_επανάληψης
 Για i από 1 μέχρι 30
  Για j από 1 μέχρι 3
   ! έλεγχος δεδομένων
   Αρχή_επανάληψης
    Διάβασε ΕΠΙΔΟΣΕΙΣ_ΗΜ[i, j]
   Μέχρις_ότου ΕΠΙΔΟΣΕΙΣ_ΗΜ[i, j] >= 0
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 ! δημιουργία πίνακα ΜΕΓ_ΕΠΙΔ
 Για i από 1 μέχρι 30
  μέγιστο ← ΕΠΙΔΟΣΕΙΣ_ΗΜ[i, 1]
  Για j από 2 μέχρι 3
   Αν ΕΠΙΔΟΣΕΙΣ_ΗΜ[i, j] > μέγιστος τότε
    μέγιστο ← ΕΠΙΔΟΣΕΙΣ_ΗΜ[i, j]
   Τέλος_αν
  Τέλος_επανάληψης
  ΜΕΓ_ΕΠΙΔ_ΗΜ[i] ← μέγιστο
 Τέλος_επανάληψης
 ! ταξινόμηση του πίνακα ΜΕΓ_ΕΠΙΔ_ΗΜ, για να βρούμε τους 8 πρώτους
 Για i από 2 μέχρι 30
  Για j από 30 μέχρι i με_βήμα -1
   ! φθίνουσα διάταξη
   Αν ΜΕΓ_ΕΠΙΔ_ΗΜ[j - 1] < ΜΕΓ_ΕΠΙΔ_ΗΜ[j] τότε
```

```
    Αντιμετάθεσε ΜΕΓ_ΕΠΙΔ_ΗΜ[j – 1], ΜΕΓ_ΕΠΙΔ_ΗΜ[j]
    Αντιμετάθεσε ΟΝΟΜΑ_ΗΜ[j – 1], ΟΝΟΜΑ_ΗΜ[j]
    Για k από 1 μέχρι 3
     Αντιμετάθεσε ΕΠΙΔΟΣΕΙΣ_ΗΜ[j – 1, 3], ΕΠΙΔΟΣΕΙΣ_ΗΜ[j, 3]
    Τέλος_επανάληψης
   Τέλος_αν
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 Για i από 1 μέχρι 8
  Εκτύπωσε ΟΝΟΜΑ_ΗΜ[i]
 Τέλος_επανάληψης
 ! ερώτημα β: δημιουργία νέου πίνακα
 ! ΟΝΟΜΑ_ΤΕΛ (θεωρούμε ότι υπάρχουν  ! 8 ξεκάθαροι νικητές)
 Για i από 1 μέχρι 8
  ΟΝΟΜΑ_ΤΕΛ[i] ← ΟΝΟΜΑ_ΗΜ[i]
 Τέλος_επανάληψης
 ! δημιουργία νέου πίνακα ΕΠΙΔΟΣΕΙΣ_ΤΕΛ
 Για i από 1 μέχρι 8
  ! περνάμε τις επιδόσεις των 8 πρώτων
  Για j από 1 μέχρι 3
   ΕΠΙΔΟΣΕΙΣ_ΤΕΛ[i, j] ← ΕΠΙΔΟΣΕΙΣ_ΗΜ[i, j]
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 ! διαβάζουμε τις επόμενες επιδόσεις για κάθε αθλητή
 Για i από 1 μέχρι 8
  Για j από 4 μέχρι 6     ! οι επόμενες επιδόσεις των πρώτων αθλητών
   Διάβασε ΕΠΙΔΟΣΕΙΣ_ΤΕΛ[i, j]
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 ! δημιουργία πίνακα  ! ΜΕΓ_ΕΠΙΔ_ΤΕΛ  ! για να βρούμε την  ! τελική
κατάταξη
 Για i από 1 μέχρι 30
  μέγιστο ← ΕΠΙΔΟΣΕΙΣ_ΤΕΛ[i, 1]
  Για j από 2 μέχρι 6
   Αν ΕΠΙΔΟΣΕΙΣ_ΤΕΛ[i, j] > μέγιστος τότε
    μέγιστο ← ΕΠΙΔΟΣΕΙΣ_ΤΕΛ[i, j]
   Τέλος_αν
  Τέλος_επανάληψης
  ΜΕΓ_ΕΠΙΔ_ΤΕΛ[i] ← μέγιστο
 Τέλος_επανάληψης
 ! φθίνουσα ταξινόμηση του πίνακα ΜΕΓ_ΕΠΙΔ_ΤΕΛ για να βρούμε τους 3
πρώτους
 ! με αντιμετάθεση του πίνακα ΟΝΟΜΑ_ΤΕΛ
 ! παραλλάσσουμε τον αλγόριθμο αναζήτησης
 συνέχεια ← Αληθής
 i ← 1
 Όσο (συνέχεια = Αληθής) και (i <= 8) επανάλαβε
  Αν ΜΕΓ_ΕΠΙΔΟΣΗ[i] >= ΜΕΓ_ΕΠΙΔΟΣΗ[3] τότε
   Εκτύπωσε i, ΟΝΟΜΑ_ΤΕΛ[i], ΜΕΓ_ΕΠΙΔ_ΤΕΛ[i]
   i ← i + 1
  Αλλιώς
   συνέχεια ← Ψευδής
  Τέλος_αν
 Τέλος_επανάληψης
Τέλος Τρίαθλο
```

Άσκηση 11. Η εταιρεία Αρβίλογλου διαθέτει 200 πωλητές σ' ολόκληρη την Ελλάδα. Να αναπτύξετε αλγόριθμο που θα διαβάζει τα ονόματα των πωλητών στον μονοδιάστατο πίνακα ΟΝΟΜΑ και τις μηνιαίες πωλήσεις καθενός απ' αυτούς, για τον χρόνο που πέρασε, στον πίνακα ΠΩΛΗΣΕΙΣ[200, 12]. Στη συνέχεια ο αλγόριθμος θα πρέπει:

α. Να υπολογίζει τους μέσους όρους εισπράξεων ανά πωλητή και να τους αποθηκεύει στον πίνακα ΜΟ_ΠΩΛΗΤΗΣ.

β. Να υπολογίζει τους μέσους όρους εισπράξεων ανά μήνα και να τους αποθηκεύει στον πίνακα ΜΟ_ΜΗΝΑΣ.

γ. Να εμφανίζει για κάθε πωλητή τους μήνες στους οποίους έχει πωλήσεις περισσότερες από τον μέσο όρο του.

δ. Να εμφανίζει για κάθε μήνα το πλήθος των πωλητών που έχουν πωλήσεις μεγαλύτερες από τον μέσο όρο του μήνα.

Λύση

```
Αλγόριθμος Criflame
 Για i από 1 μέχρι 200
  Διάβασε ΟΝΟΜΑ[i]
 Τέλος_επανάληψης
 Για i από 1 μέχρι 200
  Για j από 1 μέχρι 12
   Διάβασε ΠΩΛΗΣΕΙΣ [i, j]
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 Για i από 1 μέχρι 200      ! ερώτημα α
  άθροισμα ← 0
  Για j από 1 μέχρι 12
   άθροισμα ← άθροισμα + ΠΩΛΗΣΕΙΣ[i, j]
  Τέλος_επανάληψης
  ΜΟ_ΠΩΛΗΤΗΣ[i] ← άθροισμα / 12
 Τέλος_επανάληψης
 Για j από 1 μέχρι 12       ! ερώτημα β
  άθροισμα ← 0
  Για i από 1 μέχρι 200
   άθροισμα ← άθροισμα + ΠΩΛΗΣΕΙΣ[i, j]
  Τέλος_επανάληψης
  ΜΟ_ΜΗΝΑΣ[j] ← άθροισμα / 200
 Τέλος_επανάληψης
 Για i από 1 μέχρι 200      ! ερώτημα γ
  Για j από 1 μέχρι 12
   Αν ΠΩΛΗΣΕΙΣ[i, j] > ΜΟ_ΠΩΛΗΤΗΣ[i] τότε
    Εμφάνισε j
   Τέλος_αν
  Τέλος_επανάληψης
 Τέλος_επανάληψης
 Για j από 1 μέχρι 12       ! ερώτημα δ
  πλήθος ← 0
  Για i από 1 μέχρι 200
   Αν ΠΩΛΗΣΕΙΣ[i, j] > ΜΟ_ΜΗΝΑΣ[j] τότε
    πλήθος ← πλήθος + 1
   Τέλος_αν
  Τέλος_επανάληψης
  Εμφάνισε j, πλήθος
 Τέλος_επανάληψης
Τέλος Criflame
```

Άσκηση 12. Για την πρόκριση στον τελικό των 200 μέτρων ανδρών πραγματοποιούνται 2 προκριματικοί αγώνες με 10 αθλητές στον καθένα, ενώ στον τελικό προκρίνονται 4 αθλητές από κάθε προκριματικό. Να αναπτύξετε αλγόριθμο που θα δέχεται τους πίνακες ΟΝΟΜΑ_Α[10], ΕΠΙΔΟΣΗ_Α[10] και ΟΝΟΜΑ_Β[10], ΕΠΙΔΟΣΗ_Β[10], και στη συνέχεια:

α. Θα δημιουργεί τους νέους πίνακες ΟΝΟΜΑ_ ΤΕΛΙΚΟΣ και ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ με τα ονόματα και τις αντίστοιχες επιδόσεις όσων συμμετέχουν στον τελικό.

β. Θα εμφανίζει τα στοιχεία ως εξής: Αθλητής_1: 9.81 Αθλητής_2: +0.02 Αθλητής_3: +2.23 (ο χρόνος του πρώτου σε δευτερόλεπτα και για κάθε επόμενο αθλητή η διαφορά του με τον πρώτο).

Λύση

```
Αλγόριθμος Τελικός
 Δεδομένα // ΟΝΟΜΑ_Α, ΕΠΙΔΟΣΗ_Α, ΟΝΟΜΑ_Β, ΕΠΙΔΟΣΗ_Β //
 ! φθίνουσα ταξιν. πίνακα ΕΠΙΔΟΣΗ_Α με αντιμετάθεση πίνακα ΟΝΟΜΑ_Α
 ! φθίνουσα ταξιν. πίνακα ΕΠΙΔΟΣΗ_Β με αντιμετάθεση πίνακα ΟΝΟΜΑ_Β
 Για j από 1 μέχρι 4       ! ερώτημα α
  ΟΝΟΜΑ_ΤΕΛΙΚΟΣ[i] ← ΟΝΟΜΑ_Α[i]
  ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ[i] ← ΕΠΙΔΟΣΗ_Α[i]
 Τέλος_επανάληψης
 Για j από 5 μέχρι 8
  ΟΝΟΜΑ_ΤΕΛΙΚΟΣ[i] ← ΟΝΟΜΑ_Β[i - 4]
  ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ[i] ← ΕΠΙΔΟΣΗ_Β[i - 4]
 Τέλος_επανάληψης
 ! φθίνουσα ταξινόμηση του πίνακα ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ
 ! με αντιμετάθεση του πίνακα ΟΝΟΜΑ_ΤΕΛΙΚΟΣ
 Εκτύπωσε ΟΝΟΜΑ_ΤΕΛΙΚΟΣ[1], ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ[1]     ! ερώτημα β
 Για i από 2 μέχρι 8
  διαφορά ← ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ[i] - ΕΠΙΔΟΣΗ_ΤΕΛΙΚΟΣ[1]
  Εκτύπωσε ΟΝΟΜΑ_ΤΕΛΙΚΟΣ[i], διαφορά
 Τέλος_επανάληψης
Τέλος Τελικός
```

Άσκηση 13. Ο κύριος Αρβίλογλου αγόρασε ένα ταξί και εργάζεται με αυτό και καταγράφει στο σημειωματάριό του, τις ημερήσιες εισπράξεις από τη δουλειά του. Αν κάποια ημέρα δεν εργάστηκε καταγράφεται η τιμή μηδέν. Σε πίνακα ΕΙΣ[365], εισάγονται τα στοιχεία που αφορούν τις ημερήσιες εισπράξεις του έτους 2007. Να αναπτύξετε αλγόριθμο, ο οποίος:

α. θα πραγματοποιεί είσοδο των απαραίτητων στοιχείων.

β. θα εμφανίζει το ποσοστό των ημερών του έτους, που ο κύριος Αρβίλογλου δεν εργάστηκε.

γ. θα εμφανίζει τα έσοδα του Ιανουαρίου και του Μαρτίου.

δ. θα διαβάζει το όνομα ενός μήνα και θα εμφανίζει τα έσοδα αυτού του μήνα. Αν δοθεί λάθος όνομα μήνα, θα εμφανίζεται κατάλληλο μήνυμα.

ε. θα εντοπίζει το μικρότερο ποσό που συγκέντρωσε κάποια ημέρα (θεωρούμε ότι συνέβη μόνο μια φορά) και θα εμφανίζει σε ποιο μήνα έγινε αυτό.

στ. θα εμφανίζει ανά μήνα τις εισπράξεις που πραγματοποιήθηκαν.

Παρατήρηση: Να θεωρήσετε δεδομένους πίνακες ΜΗΝΑΣ[12], που περιέχει τα ονόματα των μηνών και ΗΜ[12], που περιέχει το πλήθος των ημερών ανά μήνα (ΗΜ[1] = 31, ΗΜ[2] = 28, ΗΜ[3] = 31 κ.ο.κ.).

Λύση

```
Αλγόριθμος Ταξί
 Δεδομένα // ΜΗΝΑΣ, ΗΜ //
 Για i από 1 μέχρι 365
   Διάβασε ΕΙΣ[i]
 Τέλος_επανάληψης
 π ← 0    !  (β)
 Για i από 1 μέχρι 365
   Αν ΕΙΣ[i] = 0 τότε
      π ← π + 1
   Τέλος_αν
 Τέλος_επανάληψης
 ποσ ← 100 * π / 365
 Εμφάνισε ποσ
```

```
! (γ): Ιανουάριος, πρώτες 31 μέρες
Σ1 ← 0
Για i από 1 μέχρι ΗΜ[1]
  Σ1 ← Σ1 + ΕΙΣ[i]
Τέλος_επανάληψης
! Μάρτιος, αρχή: ΗΜ[1]+ΗΜ[2]+1
! Μάρτιος, τελος: ΗΜ[1]+ΗΜ[2]+ΗΜ[3]
πριν ← ΗΜ[1] + ΗΜ[2]
Σ3 ← 0
Για i από πριν+1 μέχρι πριν+ΗΜ[3]
  Σ3 ← Σ3 + ΕΙΣ[i]
Τέλος_επανάληψης
Εμφάνισε Σ1, Σ3
Διάβασε ονkey  ! (δ)
done ← ψευδής
pos ← 0
μ ← 1
Όσο (done = ψευδής) και (μ <= 12) επανάλαβε
 Αν ΜΗΝΑΣ[i] = ονkey  τότε
  pos ← μ
  done ← αληθής
 Αλλιώς
  μ ← μ + 1
 Τέλος_αν
Τέλος_επανάληψης
Αν done = αληθής τότε
   πριν ← 0
   Για μ από 1 μέχρι pos-1
     πριν ← πριν + ΗΜ[μ]
   Τέλος_επανάληψης
   Σ ← 0
   Για i από πριν+1 μέχρι πριν+ΗΜ[pos]
     Σ ← Σ + ΕΙΣ[i]
   Τέλος_επανάληψης
   Εμφάνισε Σ
Αλλιώς
 Εμφάνισε "Λάθος όνομα μήνα"
Τέλος_αν
ελάχιστο ← ΕΙΣ[1]  ! (ε)
θελαχ ← 1
Για i από 2 μέχρι 365
 Αν ΕΙΣ[i] < ελάχιστο τότε
  ελάχιστο ← ΕΙΣ[i]
  θμεγ ← i
 Τέλος_αν
Τέλος_επανάληψης
Σ ← 0
δ ← ψευδής
μ ← 1
Όσο (δ = ψευδής) και (μ <= 12) επανάλαβε
  Σ ← Σ + ΗΜ[μ]
  Αν θμεγ <= Σ τότε
    Εμφάνισε ΜΗΝΑΣ[μ]
    δ ← αληθής
  Αλλιώς
    μ ← μ + 1
  Τέλος_αν
Τέλος_επανάληψης
Για μ από 1 μέχρι 12
  πριν ← 0
  Για i από 1 μέχρι μ-1
```

```
      πριν ← πριν + ΗΜ[i]
    Τέλος_επανάληψης
    Σ ← 0
    Για i από πριν+1 μέχρι πριν+ΗΜ[μ]
      Σ ← Σ + ΕΙΣ[i]
    Τέλος_επανάληψης
    Εμφάνισε Σ
  Τέλος_επανάληψης
Τέλος Ταξί
```