

ΠΙΝΑΚΕΣ

ΠΡΟΓΡΑΜΜΑ Γράψιμο_Πίνακα_1δ

! Με το πρώτο «όσο» ανατίθενται τιμές στον πίνακα.

! Με το δεύτερο «όσο» ο πίνακας εμφανίζεται στην οθόνη.

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: table[10]

ΑΚΕΡΑΙΕΣ: i

ΑΡΧΗ

```
i <- 1
ΟΣΟ i <= 10 ΕΠΑΝΑΛΑΒΕ
  table[i] <- i
  i <- i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
i <- 1
ΟΣΟ i <= 10 ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ table[i]
  i <- i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Γράψιμο_Πίνακα_1δ

ΠΡΟΓΡΑΜΜΑ Γράψιμο_Πίνακα_3δ

! Αναθέτει τιμές σε τρισδιάστατο πίνακα

! και συγχρόνως τις εμφανίζει

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: table[2, 3, 4]

ΑΚΕΡΑΙΕΣ: i, k, m, n

ΑΡΧΗ

```
i <- 1
n <- 1
ΟΣΟ i <= 2 ΕΠΑΝΑΛΑΒΕ
  k <- 1
  ΟΣΟ k <= 3 ΕΠΑΝΑΛΑΒΕ
    m <- 1
    ΟΣΟ m <= 4 ΕΠΑΝΑΛΑΒΕ
      table[i, k, m] <- n
      n <- n + 1
      ΓΡΑΨΕ table[i, k, m]
      m <- m + 1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    k <- k + 1
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  i <- i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Γράψιμο_Πίνακα_3δ

ΠΡΟΓΡΑΜΜΑ Ελάχιστο_στοιχείο_Πίνακα

ΣΤΑΘΕΡΕΣ

n = 10

ΜΕΤΑΒΛΗΤΕΣ

```

ΠΡΑΓΜΑΤΙΚΕΣ: table[n], min
ΑΚΕΡΑΙΕΣ: i
ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n
  ΔΙΑΒΑΣΕ table[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

min <- table[1]
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ n
  ΑΝ table[i] < min ΤΟΤΕ
    min <- table[i]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το ελάχιστο στοιχείο του πίνακα είναι το ', min

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Ελάχιστο_στοιχείο_πίνακα

```

ΠΡΟΓΡΑΜΜΑ Αθροισμα_στοιχείων_Πίνακα_2Δ
! σελίδα 59 του βιβλίου

ΣΤΑΘΕΡΕΣ

```

m = 4
n = 5

```

ΜΕΤΑΒΛΗΤΕΣ

```

ΑΚΕΡΑΙΕΣ: i, j
ΠΡΑΓΜΑΤΙΚΕΣ: table[m, n], sum_row[m], sum_col[n], sum

```

ΑΡΧΗ

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ m
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ n
    ΔΙΑΒΑΣΕ table[i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

sum <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ m
  sum_row[i] <- 0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ n
  sum_col[j] <- 0
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ m
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ n
    sum <- sum + table[i, j]
    sum_row[i] <- sum_row[i] + table[i, j]
    sum_col[j] <- sum_col[j] + table[i, j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ m
  ΓΡΑΨΕ 'Το άθροισμα της ', i, 'ης γραμμής των στοιχείων του πίνακα είναι:
', sum_row[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ n
  ΓΡΑΨΕ 'Το άθροισμα της ', j, 'ης στήλης των στοιχείων του πίνακα είναι:
', sum_col[j]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```
    ΓΡΑΨΕ 'Το άθροισμα όλων των στοιχείων του πίνακα είναι: ', sum
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αθροισμα_στοιχείων_πίνακα_2Δ
```

```
ΠΡΟΓΡΑΜΜΑ Διαγραφή_στοιχείου_πίνακα
```

```
ΣΤΑΘΕΡΕΣ
```

```
  n = 10
```

```
ΜΕΤΑΒΛΗΤΕΣ
```

```
  ΠΡΑΓΜΑΤΙΚΕΣ: table[n]
```

```
  ΑΚΕΡΑΙΕΣ: i, k, position
```

```
ΑΡΧΗ
```

```
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n
```

```
    ΔΙΑΒΑΣΕ table[i]
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
  ΓΡΑΨΕ 'Ποιο στοιχείο να διαγραφεί'
```

```
  ΔΙΑΒΑΣΕ position
```

```
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n - 1
```

```
    ΑΝ (i >= position) ΤΟΤΕ
```

```
      table[i] <- table[i + 1]
```

```
    ΤΕΛΟΣ_ΑΝ
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
  table[i] <- 0
```

```
! Αντιγραφή των στοιχείων στον νέο πίνακα  
! εκτός από το n-οστό
```

```
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n - 1
```

```
    ΓΡΑΨΕ 'Το στοιχείο του πίνακα ', i, ' είναι το ', table[i]
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Διαγραφή_στοιχείου_πίνακα
```

```
ΠΡΟΓΡΑΜΜΑ Εισαγωγή_στοιχείου_σε_πίνακα
```

```
ΣΤΑΘΕΡΕΣ
```

```
  n = 10
```

```
ΜΕΤΑΒΛΗΤΕΣ
```

```
  ΠΡΑΓΜΑΤΙΚΕΣ: table[n], new_table[n + 1], new_element
```

```
  ΑΚΕΡΑΙΕΣ: i, position
```

```
ΑΡΧΗ
```

```
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n
```

```
    ΔΙΑΒΑΣΕ table[i]
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
  ΓΡΑΨΕ 'Δώσε το νέο στοιχείο'
```

```
  ΔΙΑΒΑΣΕ new_element
```

```
  ΓΡΑΨΕ 'σε ποια θέση θέλεις να εισαχθεί'
```

```
  ΔΙΑΒΑΣΕ position
```

```
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ position - 1
```

```
    new_table[i] <- table[i]
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
! Αντιγραφή των στοιχείων στον νέο πίνακα
```

```

new_table[position] <- new_element

ΓΙΑ i ΑΠΟ position ΜΕΧΡΙ n           ! Αντιγραφή των στοιχείων στον νέο πίνακα
  new_table[i + 1] <- table[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n + 1
  ΓΡΑΨΕ 'Το στοιχείο του Νέου Πίνακα ', i, ' είναι το ', new_table[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Εισαγωγή_Στοιχείου_σε_Πίνακα

```

ΠΡΟΓΡΑΜΜΑ Αντιστροφή_Ονομάτων
*! Διαβάζει τρία ονόματα και στην συνέχεια
! τα ξαναεμφανίζει με την αντίστροφη σειρά.*

ΜΕΤΑΒΛΗΤΕΣ
ΧΑΡΑΚΤΗΡΕΣ: Ονόματα[3]
ΑΚΕΡΑΙΕΣ: i

ΑΡΧΗ

```

i <- 1
ΟΣΟ i <= 3 ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ 'Δώσε ένα όνομα: '
  ΔΙΑΒΑΣΕ Ονόματα[i]
  i <- i + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Αντίστροφη σειρά:'
i <- 3
ΟΣΟ i >= 1 ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ Ονόματα[i]
  i <- i - 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αντιστροφή_Ονομάτων

ΠΡΟΓΡΑΜΜΑ Αριθμός_Νομισμάτων
*! σελίδα 85 του βιβλίου
! Ο χρήστης καλείται να εισάγει ένα ακέραιο ποσό σε ευρώ
! και ο υπολογιστής του απαντά πόσα συνολικά χαρτονομίσματα ή νομίσματα
! χρειάζονται για να εκφραστεί αυτό το ποσό.*

ΣΤΑΘΕΡΕΣ
n = 9
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: kind_of_money[n], money, count, i

ΑΡΧΗ

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n
  ΔΙΑΒΑΣΕ kind_of_money[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Δώσε ένα ακέραιο ποσό σε ευρώ'
ΔΙΑΒΑΣΕ money

count <- 0

```

```

i <- n
ΟΣΟ money > 0 ΚΑΙ i > 0 ΕΠΑΝΑΛΑΒΕ
  ΑΝ kind_of_money[i] <= money ΤΟΤΕ
    count <- count + 1
    money <- money - kind_of_money[i]
  ΑΛΛΙΩΣ
    i <- i - 1
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'πλήθος νομισμάτων : ', count

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αριθμός_Νομισμάτων

```

ΠΡΟΓΡΑΜΜΑ Αριθμός_και_Είδος_Νομισμάτων
! σελίδα 85 του βιβλίου
! ο χρήστης καλείται να εισάγει ένα ακέραιο ποσό σε ευρώ
! και ο υπολογιστής του απαντά πόσα χαρτονομίσματα ή νομίσματα
! των 500, 200, 100, 50, 20, 10, 5, 2, 1 ευρώ χρειάζονται
! για να εκφραστεί αυτό το ποσό.

```

ΣΤΑΘΕΡΕΣ
n = 9
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: kind_of_money[n], quantity_of_kind[n], money, count, i

```

ΑΡΧΗ

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ n
  ΔΙΑΒΑΣΕ kind_of_money[i]
  quantity_of_kind[i] <- 0 ! Αρχικοποίησε τις ποσότητες
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΡΑΨΕ 'Δώσε ένα ακέραιο ποσό σε ευρώ: '
ΔΙΑΒΑΣΕ money
ΓΡΑΨΕ 'Για το ποσό των ', money, ' ευρώ χρειάζονται: '
ΓΡΑΨΕ
count <- 0
i <- n

```

```

ΟΣΟ money > 0 ΕΠΑΝΑΛΑΒΕ
  ΟΣΟ money < kind_of_money[i] ΕΠΑΝΑΛΑΒΕ
    i <- i - 1
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  quantity_of_kind[i] <- money div kind_of_money[i]
  money <- money mod kind_of_money[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΙΑ i ΑΠΟ n ΜΕΧΡΙ 1 ΜΕ_ΒΗΜΑ -1
  ΑΝ quantity_of_kind[i] <> 0 ΤΟΤΕ
    count <- count + quantity_of_kind[i]
    ΓΡΑΨΕ quantity_of_kind[i], ' νομίσματα των ', kind_of_money[i], ' ευρώ'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΓΡΑΨΕ 'πλήθος νομισμάτων : ', count

```

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Αριθμός_και_Είδος_Νομισμάτων

ΠΡΟΓΡΑΜΜΑ Στατιστική_Ανάλυση
! σελίδα 158 του βιβλίου

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: *i, j, N, X[100], Αθροισμα, Αθροισμα_2, Βοηθητική*
ΠΡΑΓΜΑΤΙΚΕΣ: *ΜΤ, Τυπ_Απόκλιση, Διάμεσος*

ΑΡΧΗ

! Εισαγωγή δεδομένων
ΓΡΑΨΕ 'Δώσε το πλήθος των αριθμών (μέγιστο 100)'
ΔΙΑΒΑΣΕ N
ΓΙΑ *i* **ΑΠΟ** 1 **ΜΕΧΡΙ** N
 ΓΡΑΨΕ 'Δώσε τον ', *i*, '-το αριθμό'
 ΔΙΑΒΑΣΕ X[*i*]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Υπολογισμός αθροισμάτων
Αθροισμα <- 0
Αθροισμα_2 <- 0
ΓΙΑ *i* **ΑΠΟ** 1 **ΜΕΧΡΙ** N
 ΓΡΑΨΕ 'Δώσε τον ', *i*, '-στο αριθμό'
 ΔΙΑΒΑΣΕ X[*i*]
 Αθροισμα <- Αθροισμα + X[*i*]
 Αθροισμα_2 <- Αθροισμα_2 + X[*i*]²
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Υπολογισμός μέσου όρου
ΜΤ <- Αθροισμα/N

! Υπολογισμός τυπικής απόκλισης
Τυπ_Απόκλιση <- T_P(Αθροισμα_2/N - ΜΤ²)

! Ταξινόμηση των στοιχείων του πίνακα

ΓΙΑ *i* **ΑΠΟ** 2 **ΜΕΧΡΙ** N
 ΓΙΑ *j* **ΑΠΟ** N **ΜΕΧΡΙ** *i* **ΜΕ_ΒΗΜΑ** -1
 ΑΝ X[*j* - 1] > X[*j*] **ΤΟΤΕ**
! Αντιμετάθεση των στοιχείων j και j-1
 Βοηθητική <- X[*j* - 1]
 X[*j* - 1] <- X[*j*]
 X[*j*] <- Βοηθητική

ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! j
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! i

! Υπολογισμός διαμέσου

ΑΝ N mod 2 = 0 **ΤΟΤΕ**
 Διάμεσος <- (X[A_M(N/2)] + X[A_M(N/2) + 1])/2

ΑΛΛΙΩΣ

 Διάμεσος <- X[A_M((N + 1)/2)]

ΤΕΛΟΣ_ΑΝ

! Εκτύπωση αποτελεσμάτων

ΓΡΑΨΕ 'ΑΠΟΤΕΛΕΣΜΑΤΑ'
ΓΡΑΨΕ '=====
ΓΡΑΨΕ 'πλήθος τιμών = ', N
ΓΡΑΨΕ 'Μέση τιμή = ', ΜΤ
ΓΡΑΨΕ 'Τυπική απόκλιση = ', Τυπ_Απόκλιση
ΓΡΑΨΕ 'Διάμεσος = ', Διάμεσος

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Στατιστική_Ανάλυση

ΠΡΟΓΡΑΜΜΑ Δύναμη_στη_N

! σελίδα 84 του βιβλίου

! Δημιουργούμε έναν πίνακα με δυνάμεις

! του συγκεκριμένου αριθμού εις την 1, 2, 4, 8, 16,....

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: x, n, power, used, k, pow[100]

ΑΚΕΡΑΙΕΣ: i

ΑΡΧΗ

ΓΡΑΨΕ 'Δώσε έναν αριθμό'

ΔΙΑΒΑΣΕ x

ΓΡΑΨΕ 'Δώσε τη δύναμη'

ΔΙΑΒΑΣΕ n

! αποθήκευση στοιχείων πίνακα

pow[1] <- x

i <- 1

k <- 1

ΟΣΟ k < n **ΕΠΑΝΑΛΑΒΕ**

 i <- i + 1

 k <- 2* k

 pow[i] <- pow[i - 1]* pow[i - 1]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! ανεύρεση της δύναμης

used <- 0

power <- 1

ΟΣΟ used < n **ΕΠΑΝΑΛΑΒΕ**

ΑΝ used + k <= n **ΤΟΤΕ**

 power <- power* pow[i]

 used <- used + k

ΤΕΛΟΣ_ΑΝ

 k <- k/ 2

 i <- i - 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Η δύναμη του ', x, ' στην ', n, ' είναι : ', power

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Δύναμη_στη_N