

# Μάθημα 165

Επανάληψη

B.3

Υποπρογράμματα

## Άσκηση / ΑΣΕΠ

Σε ένα διαγωνισμό του ΑΣΕΠ 500 υποψήφιοι διαγωνίζονται σε τρία μαθήματα για την κάλυψη θέσεων του Δημοσίου.

Να γραφτεί πρόγραμμα σε γλώσσα προγραμματισμού ΓΛΩΣΣΑ το οποίο:

- Να περιλαμβάνει το τμήμα δηλώσεων.
- Να διαβάζει τα ονόματα των 500 υποψηφίων και τη βαθμολογία καθενός υποψηφίου στα τρία διαφορετικά μαθήματα, καταχωρώντας τα αντίστοιχα στους πίνακες πίνακα ΥΠ[500] και ΒΑΘ[500,3].  
Το πρόγραμμα να κάνει έλεγχο βαθμολογίας ώστε να είναι δεκτοί οι βαθμοί από το 1 έως το 20, διαφορετικά να ζητείται εκ νέου ο βαθμός.
- Να εκτυπώνει τα ονόματα και τον μέσο όρο βαθμολογίας κάθε υποψηφίου στα τρία μαθήματα που εξετάστηκε.  
Για τον υπολογισμό του μέσου όρου κάθε υποψηφίου θα καλείται η συνάρτηση ΜΕΣΟΣ\_ΟΡΟΣ.
- Να γραφτεί η συνάρτηση ΜΕΣΟΣ\_ΟΡΟΣ, η οποία υπολογίζει το μέσο όρο βαθμολογίας κάθε υποψηφίου στα τρία μαθήματα που διαγωνίστηκε.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_165\_ΑΣΕΠ  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΧΑΡΑΚΤΗΡΕΣ:** ΥΠ[500]

**ΑΚΕΡΑΙΕΣ:** i, j

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΒΑΘ[500, 3], ΜΟ

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 500

**ΔΙΑΒΑΣΕ** ΥΠ[i]

**ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** 3

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** ΒΑΘ[i, j]

**ΜΕΧΡΙΣ\_ΟΤΟΥ** ΒΑΘ[i, j]  $\geq$  1 **ΚΑΙ** ΒΑΘ[i, j]  $\leq$  20

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 500

**ΜΟ**  $\leftarrow$  ΜΕΣΟΣ\_ΟΡΟΣ(ΒΑΘ, i)

**ΓΡΑΨΕ** ΥΠ[i], ΜΟ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΣΥΝΑΡΤΗΣΗ** ΜΕΣΟΣ\_ΟΡΟΣ(ΒΑΘ, i): **ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** ΒΑΘ[500, 3], S

**ΑΚΕΡΑΙΕΣ:** i, j

**ΑΡΧΗ**

S  $\leftarrow$  0

**ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** 3

S  $\leftarrow$  S + ΒΑΘ[i, j]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΜΕΣΟΣ\_ΟΡΟΣ  $\leftarrow$  S / 3

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## Συνάρτηση σε διαδικασία

Να ξαναγράψετε το παρακάτω πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία με τη χρήση διαδικασίας αντί συνάρτησης.

**ΠΡΟΓΡΑΜΜΑ** Συνάρτηση\_σε\_διαδικασία  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, z$

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $x$

**ΟΣΟ**  $x <> 1$  **ΚΑΙ**  $x <> 89$  **ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ**  $x$

$y \leftarrow \text{ΥΠ}(x)$

$z \leftarrow x$

$x \leftarrow y$

$y \leftarrow z$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ**  $x$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΣΥΝΑΡΤΗΣΗ**  $\text{ΥΠ}(x)$  :**ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, S$

**ΑΡΧΗ**

$S \leftarrow 0$

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

$y \leftarrow x \bmod 10$

$S \leftarrow S + y^2$

$x \leftarrow x \operatorname{div} 10$

**ΜΕΧΡΙΣ\_ΟΤΟΥ**  $x = 0$

$\text{ΥΠ} \leftarrow S$

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**ΠΡΟΓΡΑΜΜΑ** Συνάρτηση\_σε\_διαδικασία  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, z, x_, \text{ΥΠ}$

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $x$

**ΟΣΟ**  $x <> 1$  **ΚΑΙ**  $x <> 89$  **ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ**  $x$

$x_ \leftarrow x$

**ΚΑΛΕΣΕ**  $\text{ΥΠ1}(x_, \text{ΥΠ})$

$y \leftarrow \text{ΥΠ}$

$z \leftarrow x$

$x \leftarrow y$

$y \leftarrow z$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ**  $x$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΔΙΑΔΙΚΑΣΙΑ**  $\text{ΥΠ1}(x, \text{ΥΠ})$

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, S, \text{ΥΠ}$

**ΑΡΧΗ**

$S \leftarrow 0$

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

$y \leftarrow x \bmod 10$

$S \leftarrow S + y^2$

$x \leftarrow x \operatorname{div} 10$

**ΜΕΧΡΙΣ\_ΟΤΟΥ**  $x = 0$

$\text{ΥΠ} \leftarrow S$

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

## Συνάρτηση σε πρόγραμμα

Να ξαναγράψετε το παρακάτω πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία χωρίς να κάνει χρήση υποπρογράμματος.

**ΠΡΟΓΡΑΜΜΑ** Συνάρτηση\_σε\_πρόγραμμα  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, z$

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $x$

**ΟΣΟ**  $x <> 1$  **ΚΑΙ**  $x <> 89$  **ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ**  $x$

$y \leftarrow \text{ΥΠ}(x)$

$z \leftarrow x$

$x \leftarrow y$

$y \leftarrow z$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ**  $x$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΣΥΝΑΡΤΗΣΗ**  $\text{ΥΠ}(x)$ :**ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, S$

**ΑΡΧΗ**

$S \leftarrow 0$

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

$y \leftarrow x \bmod 10$

$S \leftarrow S + y^2$

$x \leftarrow x \text{ div } 10$

**ΜΕΧΡΙΣ\_ΟΤΟΥ**  $x = 0$

$\text{ΥΠ} \leftarrow S$

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**ΠΡΟΓΡΑΜΜΑ** Συνάρτηση\_σε\_πρόγραμμα  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $x, y, z, x_, y_, S, \text{ΥΠ}$

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $x$

**ΟΣΟ**  $x <> 1$  **ΚΑΙ**  $x <> 89$  **ΕΠΑΝΑΛΑΒΕ**

**ΓΡΑΨΕ**  $x$

$x_ \leftarrow x$

!---

$S \leftarrow 0$

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

$y_ \leftarrow x_ \bmod 10$

$S \leftarrow S + y_^2$

$x_ \leftarrow x_ \text{ div } 10$

**ΜΕΧΡΙΣ\_ΟΤΟΥ**  $x_ = 0$

$\text{ΥΠ} \leftarrow S$

!---

$y \leftarrow \text{ΥΠ}$

$z \leftarrow x$

$x \leftarrow y$

$y \leftarrow z$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ**  $x$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Διαδικασία σε πρόγραμμα

Να ξαναγράψετε το παρακάτω πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία χωρίς να κάνει χρήση υποπρογράμματος.

**ΠΡΟΓΡΑΜΜΑ** Διαδικασία\_σε\_πρόγραμμα  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** A[5] , i

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

**ΔΙΑΒΑΣΕ** A[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΚΑΛΕΣΕ** ΤΑΞ(A)

**ΓΡΑΨΕ** A[5]

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΔΙΑΔΙΚΑΣΙΑ** ΤΑΞ(B)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** B[5] , temp , i , j

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 5

**ΓΙΑ** j **ΑΠΟ** 5 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** B[j - 1] < B[j] **ΤΟΤΕ**

temp ← B[j - 1]

B[j - 1] ← B[j]

B[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΠΡΟΓΡΑΜΜΑ** Διαδικασία\_σε\_πρόγραμμα  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** A[5] , i , j , temp

**ΑΡΧΗ**

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

**ΔΙΑΒΑΣΕ** A[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 5

**ΓΙΑ** j **ΑΠΟ** 5 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** A[j - 1] < A[j] **ΤΟΤΕ**

temp ← A[j - 1]

A[j - 1] ← A[j]

A[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** A[5]

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

# Άσκηση 1

Να ξαναγράψετε το παρακάτω πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία με τη χρήση συνάρτησης αντί διαδικασίας.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_165\_1

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΠΡΑΓΜΑΤΙΚΕΣ:** X, max

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** X

max ← X

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 10

**ΔΙΑΒΑΣΕ** X

**ΚΑΛΕΣΕ** MAXIMUM(X, max)

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** X, max

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΔΙΑΔΙΚΑΣΙΑ** MAXIMUM(X, max)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΠΡΑΓΜΑΤΙΚΕΣ:** X, max

**ΑΡΧΗ**

**ΑΝ** X > max **ΤΟΤΕ**

max ← X

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_165\_1\_α

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΠΡΑΓΜΑΤΙΚΕΣ:** X, max

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** X

max ← X

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 10

**ΔΙΑΒΑΣΕ** X

max ← MAXIMUM(X, max)

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** X, max

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΣΥΝΑΡΤΗΣΗ** MAXIMUM(X, max):**ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i

**ΠΡΑΓΜΑΤΙΚΕΣ:** X, max

**ΑΡΧΗ**

**ΑΝ** X > max **ΤΟΤΕ**

max ← X

**ΤΕΛΟΣ\_ΑΝ**

MAXIMUM ← max

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## Άσκηση 2 / Δυαδική αναζήτηση σε ταξινομημένο πίνακα για ύπαρξη τιμής

Να γραφεί πρόγραμμα που:

α. Καταχωρεί 1000 ονόματα στον πίνακα ON.

β. Ταξινομεί αλφαβητικά τον πίνακα ON.

γ. Διαβάζει ένα όνομα και ελέγχει εάν το όνομα αυτό υπάρχει στον πίνακα, κάνοντας χρήση της δυαδικής αναζήτησης.

δ. Εμφανίζει μήνυμα, σχετικά με το εάν το όνομα βρέθηκε ή όχι.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_165\_4\_ΔΥΑΔΙΚΗ  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, j, L, R, M, pos

**ΧΑΡΑΚΤΗΡΕΣ:** ON[1000], temp, key

**ΑΡΧΗ**

!α

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 1000

**ΔΙΑΒΑΣΕ** ON[i]

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

!β

**ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 1000

**ΓΙΑ** j **ΑΠΟ** 1000 **ΜΕΧΡΙ** i **ΜΕ\_ΒΗΜΑ** -1

**ΑΝ** ON[j - 1] > ON[j] **ΤΟΤΕ**

temp ← ON[j - 1]

ON[j - 1] ← ON[j]

ON[j] ← temp

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

!γ

**ΔΙΑΒΑΣΕ** key

pos ← 0 ! done ← ΨΕΥΔΗΣ

L ← 1

R ← 1000

**ΟΣΟ** pos = 0 **ΚΑΙ** L ≤ R **ΕΠΑΝΑΛΑΒΕ** ! done = ΨΕΥΔΗΣ **ΚΑΙ** L ≤ R

M ← (L + R) div 2

**ΑΝ** ON[M] = key **ΤΟΤΕ**

pos ← M ! done ← ΑΛΗΘΗΣ

**ΑΛΛΙΩΣ\_ΑΝ** ON[M] < key **ΤΟΤΕ**

L ← M + 1

**ΑΛΛΙΩΣ**

R ← M - 1

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

!δ

**ΑΝ** pos <> 0 **ΤΟΤΕ** ! done = ΑΛΗΘΗΣ

**ΓΡΑΨΕ** 'Βρέθηκε'

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δεν βρέθηκε'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Άσκηση 3 / Μαντεύω τον αριθμό

Να υλοποιηθεί πρόγραμμα κατά το οποίο κάποιος παίκτης σκέπτεται έναν ακέραιο αριθμό από το 1 έως το 1000 και ο υπολογιστής προσπαθεί να τον μαντέψει σε 10 το πολύ προσπάθειες, προτείνοντας κάθε φορά έναν αριθμό και ρωτώντας εάν τον βρήκε ή εάν είναι μεγαλύτερος ή μικρότερος του ζητούμενου.

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_165\_5\_MANTEYΩ\_TON\_ΑΡΙΘΜΟ

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** C, L, R, M

**ΧΑΡΑΚΤΗΡΕΣ:** ΑΠ, ΕΠ

**ΛΟΓΙΚΕΣ:** done

**ΑΡΧΗ**

**ΓΡΑΨΕ** 'Σκεφτείτε έναν ακέραιο αριθμό από το 1 μέχρι το 1000'

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** 'Είμαστε έτοιμοι; (N/O) '

**ΔΙΑΒΑΣΕ** ΑΠ

**ΜΕΧΡΙΣ\_ΟΤΟΥ** ΑΠ = 'N'

C ← 0

L ← 1

R ← 1000

done ← **ΨΕΥΔΗΣ**

**ΟΣΟ** done = **ΨΕΥΔΗΣ** **ΚΑΙ** L ≤ R **ΕΠΑΝΑΛΑΒΕ**

C ← C + 1

M ← (L + R) div 2

**ΓΡΑΨΕ** '1. Είναι ο ', M, ';' '

**ΓΡΑΨΕ** '2. Είναι μεγαλύτερος του ', M, ';' '

**ΓΡΑΨΕ** '3. Είναι μικρότερος του ', M, ';' '

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** ΕΠ

**ΜΕΧΡΙΣ\_ΟΤΟΥ** ΕΠ = '1' **Η** ΕΠ = '2' **Η** ΕΠ = '3'

**ΕΠΙΛΕΞΕ** ΕΠ

**ΠΕΡΙΠΤΩΣΗ** '1'

done ← **ΑΛΗΘΗΣ**

**ΠΕΡΙΠΤΩΣΗ** '2'

L ← M + 1

**ΠΕΡΙΠΤΩΣΗ** **ΑΛΛΙΩΣ**

R ← M - 1

**ΤΕΛΟΣ\_ΕΠΙΛΟΓΩΝ**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΑΝ** done = **ΑΛΗΘΗΣ** **ΚΑΙ** C < 11 **ΤΟΤΕ**

**ΓΡΑΨΕ** 'Χρειάστηκαν ', C, ' προσπάθειες'

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** 'Δώσατε λάθος απαντήσεις'

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**



# ΔΟΥΛΕΙΑ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Να ξαναγράψετε το παρακάτω πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία:
  - α. με τη χρήση διαδικασίας αντί συνάρτησης
  - β. χωρίς να κάνει χρήση υποπρογράμματος.

**ΠΡΟΓΡΑΜΜΑ** Άσκηση\_1

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\alpha$  ,  $\beta$  ,  $\gamma$

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $\alpha$ ,  $\beta$

$\gamma \leftarrow \alpha + \text{Πράξη}(\alpha, \beta)$

**ΓΡΑΨΕ**  $\gamma$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΣΥΝΑΡΤΗΣΗ** Πράξη( $\chi$ ,  $\psi$ ): **ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\chi$  ,  $\psi$

**ΑΡΧΗ**

**ΑΝ**  $\chi \geq \psi$  **ΤΟΤΕ**

    Πράξη  $\leftarrow \chi - \psi$

**ΑΛΛΙΩΣ**

    Πράξη  $\leftarrow \chi + \psi$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## ΔΟΥΛΕΙΑ ΓΙΑ ΤΟ ΣΠΙΤΙ

2. Να ξαναγράψετε το παρακάτω πρόγραμμα, ώστε να επιτελεί την ίδια λειτουργία χωρίς να κάνει χρήση υποπρογράμματος.

**ΠΡΟΓΡΑΜΜΑ** Άσκηση\_2

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** z , w

**ΑΡΧΗ**

z ← 1

w ← 3

**ΟΣΟ** z <= 35 **ΕΠΑΝΑΛΑΒΕ**

**ΚΑΛΕΣΕ** Διαδ(z, w)

**ΓΡΑΨΕ** z

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΔΙΑΔΙΚΑΣΙΑ** Διαδ(w, z)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** z , w

**ΑΡΧΗ**

w ← w + z

z ← z + 2

**ΓΡΑΨΕ** z

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

## ΔΟΥΛΕΙΑ ΓΙΑ ΤΟ ΣΠΙΤΙ

3. Να ξαναγράψετε τη διαδικασία Δ1 έχοντας μετατρέψει την συνάρτηση Σ1 σε διαδικασία, την οποία επίσης να υλοποιήσετε.

**ΔΙΑΔΙΚΑΣΙΑ Δ1 (A1, A2)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ: A1, A2**

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ A1, A2**

**ΓΡΑΨΕ Σ1(A1,A2) + A1**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΣΥΝΑΡΤΗΣΗ Σ1 (K, Λ) : ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ: K, Λ, I**

**ΑΡΧΗ**

**I ← Λ**

**ΌΣΟ K < Λ ΕΠΑΝΑΛΑΒΕ**

**I ← I - 1**

**K ← K + 3**

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Σ1 ← I**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## ΔΟΥΛΕΙΑ ΓΙΑ ΤΟ ΣΠΙΤΙ

4. Να μετατρέψετε την παραπάνω Συνάρτηση ΨΗΦΙΑ σε ισοδύναμη Διαδικασία.

**ΣΥΝΑΡΤΗΣΗ ΨΗΦΙΑ(X) : ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** X, count

**ΑΡΧΗ**

count  $\leftarrow$  0

**ΟΣΟ** X > 0 **ΕΠΑΝΑΛΑΒΕ**

count  $\leftarrow$  count + 1

X  $\leftarrow$  X **div** 10

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΨΗΦΙΑ  $\leftarrow$  count

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

## ΔΟΥΛΕΙΑ ΓΙΑ ΤΟ ΣΠΙΤΙ

5. Δίνεται παρακάτω ένα πρόγραμμα με ένα υποπρόγραμμα:

**ΠΡΟΓΡΑΜΜΑ B1**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** α, β, τελ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** α, β

**ΑΝ** ΒΡΕΣ(α,β) = 10 **ΤΟΤΕ**

τελ ← α + ΒΡΕΣ(α, β)

**ΑΛΛΙΩΣ**

τελ ← α

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** τελ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΣΥΝΑΡΤΗΣΗ ΒΡΕΣ(χ,ψ):ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** χ, ψ

**ΑΡΧΗ**

**ΑΝ** χ >= ψ **ΤΟΤΕ**

ΒΡΕΣ ← (χ - ψ) \* 3

**ΑΛΛΙΩΣ**

ΒΡΕΣ ← (χ + ψ) \* 2

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

- α. Να ξαναγράψετε το πρόγραμμα ώστε να επιτελεί την ίδια λειτουργία χρησιμοποιώντας διαδικασία αντί της συνάρτησης, την οποία διαδικασία και να κατασκευάσετε.
- β. Να ξαναγράψετε το πρόγραμμα που δόθηκε αρχικά ώστε να επιτελεί την ίδια λειτουργία χωρίς την χρήση υποπρογράμματος.