

# Μάθημα 166

Επανάληψη

B.3

Εκσφαλμάτωση

# Άσκηση 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** μάθημα\_166\_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**

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_166\_1\_α

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\alpha$  ,  $\beta$  ,  $\gamma$  ,  $\alpha_*$  ,  $\beta_*$  , Πράξη

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $\alpha$  ,  $\beta$

$\alpha_* \leftarrow \alpha$

$\beta_* \leftarrow \beta$

**ΚΑΛΕΣΕ** Πράξη1( $\alpha_*$ ,  $\beta_*$ , Πράξη)

$\gamma \leftarrow \alpha + \text{Πράξη}$

**ΓΡΑΨΕ**  $\gamma$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

!=====

**ΔΙΑΔΙΚΑΣΙΑ** Πράξη1( $\chi$ ,  $\psi$ , Πράξη)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\chi$  ,  $\psi$  , Πράξη

**ΑΡΧΗ**

**ΑΝ**  $\chi \geq \psi$  **ΤΟΤΕ**

    Πράξη  $\leftarrow \chi - \psi$

**ΑΛΛΙΩΣ**

    Πράξη  $\leftarrow \chi + \psi$

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_166\_1\_β

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:**  $\alpha$  ,  $\beta$  ,  $\gamma$  ,  $\chi$  ,  $\psi$  , Πράξη

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ**  $\alpha$  ,  $\beta$

$\chi \leftarrow \alpha$

$\psi \leftarrow \beta$

**ΑΝ**  $\chi \geq \psi$  **ΤΟΤΕ**

    Πράξη  $\leftarrow \chi - \psi$

**ΑΛΛΙΩΣ**

    Πράξη  $\leftarrow \chi + \psi$

**ΤΕΛΟΣ\_ΑΝ**

$\gamma \leftarrow \alpha + \text{Πράξη}$

**ΓΡΑΨΕ**  $\gamma$

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Άσκηση 2

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

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_166\_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**

Αρχικά δίνουμε το ίδιο όνομα στις αντίστοιχες παραμέτρους:

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_166\_2\_α  
**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(z, w)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** w , z

**ΑΡΧΗ**

z ← z + w

w ← w + 2

**ΓΡΑΨΕ** w

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΠΡΟΓΡΑΜΜΑ** μάθημα\_166\_2\_β  
**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** z , w

**ΑΡΧΗ**

z ← 1

w ← 3

**ΟΣΟ** z <= 35 **ΕΠΑΝΑΛΑΒΕ**

    z ← z + w

    w ← w + 2

**ΓΡΑΨΕ** w

**ΓΡΑΨΕ** z

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

## Άσκηση 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 (Κ, Λ) : ΑΚΕΡΑΙΑ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** Κ, Λ, Ι

**ΑΡΧΗ**

**Ι** ← Λ

**ΟΣΟ** Κ < Λ **ΕΠΑΝΑΛΑΒΕ**

**Ι** ← Ι - 1

**Κ** ← Κ + 3

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Σ1** ← Ι

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ**

**ΔΙΑΔΙΚΑΣΙΑ Δ1 (A1, A2)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** A1, A2, A1\_, A2\_, Σ1

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** A1, A2

**A1\_** ← A1

**A2\_** ← A2

**ΚΑΛΕΣΕ** Δ2(A1\_, A2\_, Σ1)

**ΓΡΑΨΕ** Σ1 + A1

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**ΔΙΑΔΙΚΑΣΙΑ Δ2(Κ, Λ, Σ1)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** Κ, Λ, Ι, Σ1

**ΑΡΧΗ**

**Ι** ← Λ

**ΟΣΟ** Κ < Λ **ΕΠΑΝΑΛΑΒΕ**

**Ι** ← Ι - 1

**Κ** ← Κ + 3

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**Σ1** ← Ι

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

## Άσκηση 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**

**ΔΙΑΔΙΚΑΣΙΑ ΨΗΦΙΑ1(X, ΨΗΦΙΑ)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** X, ΨΗΦΙΑ, count

**ΑΡΧΗ**

count  $\leftarrow$  0

**ΟΣΟ** X > 0 **ΕΠΑΝΑΛΑΒΕ**

count  $\leftarrow$  count + 1

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

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

ΨΗΦΙΑ  $\leftarrow$  count

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

## Άσκηση 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  $\leftarrow$  α + ΒΡΕΣ(α, β)

**ΑΛΛΙΩΣ**

τελ  $\leftarrow$  α

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** τελ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΣΥΝΑΡΤΗΣΗ ΒΡΕΣ(χ,ψ):ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** χ, ψ

**ΑΡΧΗ**

**ΑΝ** χ  $\geq$  ψ **ΤΟΤΕ**

ΒΡΕΣ  $\leftarrow$  (χ - ψ) \* 3

**ΑΛΛΙΩΣ**

ΒΡΕΣ  $\leftarrow$  (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.

## Άσκηση 5α

α. Να ξαναγράψετε το πρόγραμμα ώστε να επιτελεί την ίδια λειτουργία χρησιμοποιώντας διαδικασία αντί της συνάρτησης, την οποία διαδικασία και να κατασκευάσετε.

**ΠΡΟΓΡΑΜΜΑ Β1**

**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**

**ΠΡΟΓΡΑΜΜΑ Β1**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** α, β, τελ, α\_, β\_, ΒΡΕΣ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** α, β

α\_ ← α

β\_ ← β

**ΚΑΛΕΣΕ** ΔΙΑΔ(α\_, β\_, ΒΡΕΣ)

**ΑΝ** ΒΡΕΣ = 10 **ΤΟΤΕ**

τελ ← α + ΒΡΕΣ

**ΑΛΛΙΩΣ**

τελ ← α

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** τελ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ(χ, ψ, ΒΡΕΣ)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** χ, ψ, ΒΡΕΣ

**ΑΡΧΗ**

**ΑΝ** χ >= ψ **ΤΟΤΕ**

ΒΡΕΣ ← (χ - ψ) \* 3

**ΑΛΛΙΩΣ**

ΒΡΕΣ ← (χ + ψ) \* 2

**ΤΕΛΟΣ\_ΑΝ**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

## Άσκηση 5β

β. Να ξαναγράψετε το πρόγραμμα που δόθηκε αρχικά ώστε να επιτελεί την ίδια λειτουργία χωρίς την χρήση υποπρογράμματος.

**ΠΡΟΓΡΑΜΜΑ Β1**

**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**

**ΠΡΟΓΡΑΜΜΑ Β1**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΠΡΑΓΜΑΤΙΚΕΣ:** α, β, τελ, χ, ψ, ΒΡΕΣ

**ΑΡΧΗ**

**ΔΙΑΒΑΣΕ** α, β

χ ← α

ψ ← β

**ΑΝ** χ >= ψ **ΤΟΤΕ**

ΒΡΕΣ ← (χ - ψ) \* 3

**ΑΛΛΙΩΣ**

ΒΡΕΣ ← (χ + ψ) \* 2

**ΤΕΛΟΣ\_ΑΝ**

**ΑΝ** ΒΡΕΣ = 10 **ΤΟΤΕ**

τελ ← α + ΒΡΕΣ

**ΑΛΛΙΩΣ**

τελ ← α

**ΤΕΛΟΣ\_ΑΝ**

**ΓΡΑΨΕ** τελ

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**



## Εκσφαλμάτωση / Άσκηση 1 / Εκφώνηση

Η παρακάτω κωδικοποίηση διαβάζει τα ονόματα και τους μέσους όρους 30 μαθητών και εμφανίζει τους 2 απουσιολόγους (θεωρώντας ότι υπάρχουν ακριβώς 2 απουσιολόγοι χωρίς ισοβαθμίες). Περιέχει όμως λάθη, τα οποία καλείστε να υποδείξετε και να χαρακτηρίσετε το είδος τους.

1. **ΠΡΟΓΡΑΜΜΑ** Σχολείο
2. **ΜΕΤΑΒΛΗΤΕΣ**
3.     **ΑΚΕΡΑΙΕΣ:** i, j
4.     **ΧΑΡΑΚΤΗΡΕΣ:** ΟΝ[30]
5.     **ΠΡΑΓΜΑΤΙΚΕΣ:** ΜΟ[30]
6. **ΑΡΧΗ**
7.     **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30
8.         **ΔΙΑΒΑΣΕ** ΟΝ[i], ΜΟ[i]
9.     **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
10.    **ΚΑΛΕΣΕ** Ταξινόμηση(ΜΟ)
11.    **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** 2
12.         **ΓΡΑΨΕ** ΟΝ[j], ΜΟ[j]
13.    **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
14. **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**
15. !======
16. **ΔΙΑΔΙΚΑΣΙΑ** Ταξινόμηση(ΜΟ)
17. **ΜΕΤΑΒΛΗΤΕΣ**
18.     **ΠΡΑΓΜΑΤΙΚΕΣ:** ΜΟ[30], temp
19.     **ΑΚΕΡΑΙΕΣ:** i, j
20. **ΑΡΧΗ**
21.     **ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 30
22.         **ΓΙΑ** j **ΑΠΟ** 30 **ΜΕΧΡΙ** i **ΜΕ ΒΗΜΑ** -1
23.             **ΑΝ** ΜΟ[j - 1] > ΜΟ[j] **ΤΟΤΕ**
24.                 temp ← ΜΟ[j - 1]
25.                 ΜΟ[j - 1] ← ΜΟ[j]
26.                 ΜΟ[j] ← temp
27.             **ΤΕΛΟΣ\_ΑΝ**
28.         **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
29.     **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
30. **ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

Θα πρέπει κατά την ταξινόμηση των μέσων όρων να ακολουθεί και ο παράλληλος πίνακας των ονομάτων. Οπότε στη γραμμή 10 πρέπει να είναι παράμετρος και ο πίνακας ΟΝ, στη γραμμή 16, να υπάρχει και 2<sup>η</sup> παράμετρος που θα δηλωθεί ως πίνακας χαρακτήρων 30 θέσεων και επίσης να δηλωθεί μία μεταβλητή χαρακτήρα π.χ. temp2, για την ταξινόμηση του 2<sup>ου</sup> πίνακα. Στη γραμμή 23 πρέπει να αλλάξει φορά η ανισότητα (φθίνουσα ταξινόμηση) και τέλος, μετά τη γραμμή 26 να έχουμε αντιμετάθεση των στοιχείων του 2<sup>ου</sup> πίνακα.

# Εκσφαλμάτωση / Άσκηση 1 / Απάντηση

Η παρακάτω κωδικοποίηση διαβάζει τα ονόματα και τους μέσους όρους 30 μαθητών και εμφανίζει τους 2 απουσιολόγους (θεωρώντας ότι υπάρχουν ακριβώς 2 απουσιολόγοι χωρίς ισοβαθμίες). Περιέχει όμως λάθη, τα οποία καλείστε να υποδείξετε και να χαρακτηρίσετε το είδος τους.

1. **ΠΡΟΓΡΑΜΜΑ** Σχολείο
2. **ΜΕΤΑΒΛΗΤΕΣ**
3.     **ΑΚΕΡΑΙΕΣ:** i, j
4.     **ΧΑΡΑΚΤΗΡΕΣ:** ON[30]
5.     **ΠΡΑΓΜΑΤΙΚΕΣ:** MO[30]
6. **ΑΡΧΗ**
7.     **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30
8.         **ΔΙΑΒΑΣΕ** ON[i], MO[i]
9.     **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
10.     **ΚΑΛΕΣΕ** Ταξινόμηση(MO, ON)
11.     **ΓΙΑ** j **ΑΠΟ** 1 **ΜΕΧΡΙ** 2
12.         **ΓΡΑΨΕ** ON[j], MO[j]
13.     **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
14. **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**
15. !======
16. **ΔΙΑΔΙΚΑΣΙΑ** Ταξινόμηση(MO, ON)
17. **ΜΕΤΑΒΛΗΤΕΣ**
18.     **ΠΡΑΓΜΑΤΙΚΕΣ:** MO[30], temp
19.     **ΑΚΕΡΑΙΕΣ:** i, j
20.     **ΧΑΡΑΚΤΗΡΕΣ:** ON[30], temp2
21. **ΑΡΧΗ**
22.     **ΓΙΑ** i **ΑΠΟ** 2 **ΜΕΧΡΙ** 30
23.         **ΓΙΑ** j **ΑΠΟ** 30 **ΜΕΧΡΙ** i **ΜΕ ΒΗΜΑ** -1
24.             **ΑΝ** MO[j - 1] < MO[j] **ΤΟΤΕ**
25.                 temp ← MO[j - 1]
26.                 MO[j - 1] ← MO[j]
27.                 MO[j] ← temp
28.                 temp2 ← ON[j - 1]
29.                 ON[j - 1] ← ON[j]
30.                 ON[j] ← temp2
31.             **ΤΕΛΟΣ\_ΑΝ**
32.     **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
33.     **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**
34. **ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

Συνολικά πρόκειται για λογικό λάθος, καθώς θα εμφανιστούν οι δύο καλύτεροι βαθμοί και τα ονόματα δύο μαθητών, αλλά κατά πάσα πιθανότητα οι δύο μαθητές δεν θα έχουν αυτούς τους βαθμούς.

## Άσκηση 2

Δίνεται το παρακάτω τμήμα προγράμματος στο οποίο χρησιμοποιείται ο πίνακας  $A[7]$ , που περιέχει θερμοκρασίες που μετρήθηκαν από έναν αισθητήρα, ο οποίος όταν διακόπτει τη λειτουργία του καταχωρεί την τιμή  $-273$ . Στόχος είναι να εμφανίσει τον μέσο όρο των θετικών θερμοκρασιών που μετρήθηκαν.

```
1. S ← 0
2. πλ ← 0
3. i ← 1
4. ΟΣΟ A[i] <> -273 Η i <= 7 ΕΠΑΝΑΛΑΒΕ
5.   ΑΝ A[i] >= 0 ΤΟΤΕ
6.     S ← S + A[i]
7.     πλ ← πλ + 1
8.   ΤΕΛΟΣ_ΑΝ
9.   i ← i + 1
10. ΤΕΛΟΣ_ΟΣΟ
11. M_O ← S / πλ
12. ΓΡΑΨΕ M_O
```

```
1. S ← 0
2. πλ ← 0
3. i ← 1
4. ΟΣΟ A[i] <> -273 ΚΑΙ i < 7 ΕΠΑΝΑΛΑΒΕ
5.   ΑΝ A[i] > 0 ΤΟΤΕ
6.     S ← S + A[i]
7.     πλ ← πλ + 1
8.   ΤΕΛΟΣ_ΑΝ
9.   i ← i + 1
10. ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΝ πλ <> 0 ΤΟΤΕ
11.   M_O ← S / πλ
12.   ΓΡΑΨΕ M_O
    ΤΕΛΟΣ_ΑΝ
```

4. ΛΟΓΙΚΟ

(το i θα πάρει την τιμή 8)  
ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ

5. ΛΟΓΙΚΟ

10. ΣΥΝΤΑΚΤΙΚΟ

11. πιθανή διαίρεση με 0  
ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ

Το παραπάνω τμήμα προγράμματος περιέχει λάθη και των τριών κατηγοριών. Να αναφέρετε το λάθος, τη γραμμή που υπάρχει και σε ποια κατηγορία ανήκει;

## Άσκηση 3

Το ελάχιστο κοινό πολλαπλάσιο (ΕΚΠ) δυο θετικών ακεραίων  $\alpha$  και  $\beta$  μπορεί να βρεθεί υπολογίζοντας τα διαδοχικά πολλαπλάσια του  $\alpha$  μέχρι να βρεθεί ένα που διαιρείται ακριβώς με το  $\beta$ .

Η παρακάτω συνάρτηση επιστρέφει το ΕΚΠ των  $\alpha$  και  $\beta$  ωστόσο περιέχει λάθη, συντακτικά και λογικά.

Να μεταφέρετε διορθωμένη τη συνάρτηση στο τετράδιό σας ώστε να επιτελεί σωστά την παραπάνω λειτουργία.

**ΣΥΝΑΡΤΗΣΗ ΕΚΠ( $\alpha$ ,  $\beta$ ): ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\alpha$ ,  $\beta$

**ΑΡΧΗ**

$\pi \leftarrow \alpha$

**ΟΣΟ**  $\pi \bmod \beta \neq 0$  **ΕΠΑΝΑΛΑΒΕ**

$\pi \leftarrow \pi + \alpha$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

$\text{ΕΚΠ} \leftarrow \pi$

**ΤΕΛΟΣ**

**ΣΥΝΑΡΤΗΣΗ ΕΚΠ( $\alpha$ ,  $\beta$ ): ΑΚΕΡΑΙΑ ?**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:**  $\alpha$ ,  $\beta$ ,  $\pi$  **ΣΥΝΤΑΚΤΙΚΟ**

**ΑΡΧΗ**

$\pi \leftarrow \alpha$

**ΟΣΟ**  $\pi \bmod \beta \neq 0$  **ΕΠΑΝΑΛΑΒΕ** **ΛΟΓΙΚΟ**

$\pi \leftarrow \pi + \alpha$

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

$\text{ΕΚΠ} \leftarrow \pi$  **ΛΟΓΙΚΟ**

**ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** **ΣΥΝΤΑΚΤΙΚΟ**

## Άσκηση 4

Δίνεται το εξής τμήμα εντολών που αναζητά την τιμή της μεταβλητής ζητούμενο σε έναν πίνακα  $A[100]$  :

1.  $K \leftarrow 0$
2. Αρχή\_επανάληψης
3.  $K \leftarrow K + 1$
4. Μέχρις\_ότου  $K > 100$  ή  $A[K] = \text{ζητούμενο}$
5. Αν  $K = 100$  τότε
6. Εμφάνισε 'Δεν βρέθηκε'
7. αλλιώς
8. Γράψε 'Βρέθηκε'
9. Τέλος\_αν

1.  $K \leftarrow 0$
2. Αρχή\_επανάληψης
3.  $K \leftarrow K + 1$
4. Μέχρις\_ότου  $K = 100$  ή  $A[K] = \text{ζητούμενο}$
5. Αν  $A[K] \neq \text{ζητούμενο}$  τότε
6. Εμφάνισε 'Δεν βρέθηκε'
7. αλλιώς
8. Γράψε 'Βρέθηκε'
9. Τέλος\_αν

Σημειώστε τους αριθμούς γραμμών που υπάρχει λάθος και το είδος του κάθε λάθους.

Γραμμή 4:

Χρόνου εκτέλεσης, καθώς το  $K$  μπορεί να πάρει την τιμή 101 και θα βρεθούμε εκτός ορίων του πίνακα.

Γραμμή 5:

Λογικό, καθώς στη περίπτωση που το ζητούμενο είναι το  $A[100]$ , δηλ.  $K = 100$  και  $A[K] = \text{ζητούμενο}$ , θα εμφανίσει πως δεν βρέθηκε.

## ΔΟΥΛΕΙΑ ΓΙΑ ΤΟ ΣΠΙΤΙ

Να γραφεί ΠΡΟΓΡΑΜΜΑ που να επεξεργάζεται μία αρχικά άδεια στοίβα ονομάτων, υλοποιούμενη με τον πίνακα  $A[N]$ , όπου το  $N = 10$  θα δηλωθεί ως σταθερά, με τον εξής τρόπο:

Εμφανίζει στο χρήστη μενού με τις επιλογές:

1. ΩΘΗΣΗ
2. ΑΠΩΘΗΣΗ
3. ΓΕΜΙΣΜΑ
4. ΑΔΕΙΑΣΜΑ

και περιμένει από το χρήστη μία έγκυρη επιλογή.

Στην περίπτωση 1, θα ζητά από το χρήστη ένα όνομα το οποίο θα ωθεί στη στοίβα, κάνοντας χρήση της διαδικασίας PUSH. Στην περίπτωση που η στοίβα είναι γεμάτη, απλά δεν θα γίνεται ώθηση.

Στην περίπτωση 2, θα γίνεται απώθηση ενός στοιχείου από τη στοίβα, κάνοντας χρήση της διαδικασίας POP και το πρόγραμμα θα εμφανίζει το όνομα που απωθήθηκε. Στην περίπτωση που η στοίβα είναι άδεια, η διαδικασία θα επιστρέφει αντί για όνομα τη λέξη 'Empty'.

Στην περίπτωση 3, θα ζητά επαναληπτικά ονόματα τα οποία θα ωθεί στη στοίβα, μέχρι αυτή να γεμίσει, κάνοντας χρήση της διαδικασίας PUSH για κάθε ώθηση.

Στην περίπτωση 4, το πρόγραμμα θα εμφανίζει όλα τα ονόματα που υπάρχουν στη στοίβα, κάνοντας επαναληπτική χρήση της διαδικασίας POP.

Μετά από κάθε περίπτωση ακολουθεί το ερώτημα: 'Συνεχίζουμε; (ΝΑΙ / ΟΧΙ)' ελέγχοντας πως ο χρήστης δίνει μία από τις τιμές 'ΝΑΙ' ή 'ΟΧΙ' και το πρόγραμμα ολοκληρώνεται όταν ο χρήστης απαντήσει 'ΟΧΙ'.