Μάθημα 166

Επανάληψη

Β.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: α , β , γ
ΑΡΧΗ
   ΔΙΑΒΑΣΕ α, β
   γ ← α + Πράξη(α, β)
   ΓΡΑΨΕ γ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
!==============================
ΣΥΝΑΡΤΗΣΗ Πράξη(χ, ψ): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
   ΠΡΑΓΜΑΤΙΚΕΣ: χ , ψ
ΑΡΧΗ
   ΑΝ χ >= ψ ΤΟΤΕ
      Πράξη ← χ – ψ
   ΑΛΛΙΩΣ
      Πράξη ← χ + ψ
   ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

```
ΠΡΟΓΡΑΜΜΑ μάθημα_166_1_α
ΜΕΤΑΒΛΗΤΕΣ
   ΠΡΑΓΜΑΤΙΚΕΣ: α , β , γ , α_ , β_ , Πράξη
ΑΡΧΗ
   ΔΙΑΒΑΣΕ α, β
   α_ ← α
   β_ ← β
   ΚΑΛΕΣΕ Πράξη1(α_, β_, Πράξη)
   γ ← α + Πράξη
   ΓΡΑΨΕ γ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
!==========================
ΔΙΑΔΙΚΑΣΙΑ Πράξη1(χ, ψ, Πράξη)
ΜΕΤΑΒΛΗΤΕΣ
   ΠΡΑΓΜΑΤΙΚΕΣ: χ , ψ , Πράξη
ΑΡΧΗ
   ΑΝ χ >= ψ ΤΟΤΕ
      Πράξη ← χ – ψ
   ΑΛΛΙΩΣ
      Πράξη ← χ + ψ
   ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

```
ΠΡΟΓΡΑΜΜΑ μάθημα_166_1_β
ΜΕΤΑΒΛΗΤΕΣ
   ΠΡΑΓΜΑΤΙΚΕΣ: α , β , γ , χ , ψ , Πράξη
ΑΡΧΗ
   ΔΙΑΒΑΣΕ α, β
   χ ← α
   ψ ← β
   ΑΝ χ >= ψ ΤΟΤΕ
      Πράξη ← χ – ψ
   ΑΛΛΙΩΣ
      Πράξη ← χ + ψ
   ΤΕΛΟΣ_ΑΝ
   γ ← α + Πράξη
   ΓΡΑΨΕ γ
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 (Α1, Α2)
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: Α1, Α2
ΑΡΧΗ
   ΔΙΑΒΑΣΕ Α1, Α2
   ΓΡΑΨΕ Σ1(Α1,Α2) + Α1
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 (Α1, Α2)
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: Α1, Α2, Α1_, Α2_, Σ1
ΑΡΧΗ
   ΔΙΑΒΑΣΕ Α1, Α2
   Α1_ ← Α1
   Α2_ ← Α2
   ΚΑΛΕΣΕ Δ2(Α1_, Α2_, Σ1)
   ΓΡΑΨΕ Σ1 + Α1
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(Χ) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: Χ, count
ΑΡΧΗ
   count ← 0
   ΟΣΟ Χ > 0 ΕΠΑΝΑΛΑΒΕ
      count ← count + 1
      Χ ← Χ div 10
   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
      ΨΗΦΙΑ ← count
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

```
ΔΙΑΔΙΚΑΣΙΑ ΨΗΦΙΑ1(Χ, ΨΗΦΙΑ)
   ΜΕΤΑΒΛΗΤΕΣ
      ΑΚΕΡΑΙΕΣ: Χ, Χ_, ΨΗΦΙΑ, count
   ΑΡΧΗ
      Χ_ ← Χ
      count ← 0
      ΟΣΟ Χ > 0 ΕΠΑΝΑΛΑΒΕ
        count ← count + 1
        Χ ← Χ div 10
      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
       Χ ← Χ_
      ΨΗΦΙΑ ← count
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Άσκηση 5

Δίνεται παρακάτω ένα πρόγραμμα με ένα υποπρόγραμμα:

```
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
```

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: 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)
11.     ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 2
12.         ΓΡΑΨΕ ON[j], MO[j]
13.     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
14. ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
15. !======================
16. ΔΙΑΔΙΚΑΣΙΑ Ταξινόμηση(MO)
17. ΜΕΤΑΒΛΗΤΕΣ
18.     ΠΡΑΓΜΑΤΙΚΕΣ: MO[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 MO[j - 1] > MO[j] ΤΟΤΕ
24.                 temp ← MO[j - 1]
25.                 MO[j - 1] ← MO[j]
26.                 MO[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 ON, στη γραμμή 16, να υπάρχει και $2^{η}$ παράμετρος που θα δηλωθεί ως πίνακας χαρακτήρων 30 θέσεων και επίσης να δηλωθεί μία μεταβλητή χαρακτήρας π.χ. temp2, για την ταξινόμηση του $2^{ου}$ πίνακα. Στη γραμμή 23 πρέπει να αλλάξει φορά η ανισότητα (φθίνουσα ταξινόμηση) και τέλος, μετά τη γραμμή 26 να έχουμε αντιμετάθεση των στοιχείων του $2^{ου}$ 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
        ΧΑΡΑΚΤΗΡΕΣ: ON[30], temp2
20. ΑΡΧΗ
21.     ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 30
22.         ΓΙΑ j ΑΠΟ 30 ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1
23.             ΑΝ MO[j - 1] < MO[j] ΤΟΤΕ
24.                 temp ← MO[j - 1]
25.                 MO[j - 1] ← MO[j]
26.                 MO[j] ← temp
                    temp2 ← ON[j - 1]
                    ON[j - 1] ← ON[j]
                    ON[j] ← temp2
27.             ΤΕΛΟΣ_ΑΝ
28.         ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
29.     ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
30. ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Συνολικά πρόκειται για λογικό λάθος, καθώς θα εμφανιστούν οι δύο χαμηλότεροι βαθμοί και τα ονόματα δύο "τυχαίων" μαθητών.

Άσκηση 2

Δίνεται το παρακάτω τμήμα προγράμματος στο οποίο χρησιμοποιείται ο πίνακας Α[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. Μ_Ο ← S / πλ
12. ΓΡΑΨΕ Μ_Ο
```

```
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.     Μ_Ο ← S / πλ
12.     ΓΡΑΨΕ Μ_Ο
    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 α και β μπορεί να βρεθεί υπολογίζοντας τα διαδοχικά πολλαπλάσια του α μέχρι να βρεθεί ένα που διαιρείται ακριβώς με το β.
Η παρακάτω συνάρτηση επιστρέφει το ΕΚΠ των α και β ωστόσο περιέχει λάθη, συντακτικά και λογικά.
Να μεταφέρετε διορθωμένη τη συνάρτηση στο τετράδιό σας ώστε να επιτελεί σωστά την παραπάνω λειτουργία.

```
ΣΥΝΑΡΤΗΣΗ ΕΚΠ(α, β): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: α, β
ΑΡΧΗ
   π ← α
   ΟΣΟ π mod 2 <> 0 ΕΠΑΝΑΛΑΒΕ
      π ←  π + α
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
   ΕΚΠ ← α
ΤΕΛΟΣ
```

```
ΣΥΝΑΡΤΗΣΗ ΕΚΠ(α, β): ΑΚΕΡΑΙΑ     ΣΥΝΤΑΚΤΙΚΟ
ΜΕΤΑΒΛΗΤΕΣ
   ΑΚΕΡΑΙΕΣ: α, β, π     ΣΥΝΤΑΚΤΙΚΟ
ΑΡΧΗ
   π ← α
   ΟΣΟ π mod β <> 0 ΕΠΑΝΑΛΑΒΕ     ΛΟΓΙΚΟ
      π ←  π + α
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
   ΕΚΠ ← π     ΛΟΓΙΚΟ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ     ΣΥΝΤΑΚΤΙΚΟ
```

Άσκηση 4

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

```
1.  Κ ← 0
2.  Αρχή_επανάληψης
3.      Κ ← Κ + 1
4.  Μέχρις_ότου Κ > 100 ή Α[Κ] = ζητούμενο
5.  Αν Κ = 100 τότε
6.      Εμφάνισε 'Δεν βρέθηκε'
7.  αλλιώς
8.      Γράψε 'Βρέθηκε'
9.  Τέλος_αν
```

```
1.  Κ ← 0
2.  Αρχή_επανάληψης
3.      Κ ← Κ + 1
4.  Μέχρις_ότου Κ = 100 ή Α[Κ] = ζητούμενο
5.  Αν  Α[Κ] <> 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 Κ μπορεί να πάρει την τιμή 101 και θα βρεθούμε εκτός ορίων του πίνακα.

Γραμμή 5:

Λογικό, καθώς στη περίπτωση που το ζητούμενο είναι το Α[100], δηλ. Κ = 100 και Α[Κ] = 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 Α[Ν], όπου το Ν = 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'.