Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον

Γ' Λυκείου

ΚΕΦΑΛΑΙΟ 2

Βασικές έννοιες αλγορίθμων - Δομή επανάληψης

Λύσεις ασκήσεων

Χρήστος Μουρατίδης - Έκδοση 2022
mouratx@yahoo.com
http://users.sch.gr/mouratx

Περιεχόμενα

Λύσεις ασκήσεων στην δομή επανάληψης

1. Γράψτε έναν αλγόριθμο που διαβάζει μία ακολουθία ακέραιων αριθμών που τελειώνει με 0 και υπολογίζει το άθροισμά τους.

Απάντηση

Επειδή δεν γνωρίζουμε πόσοι αριθμοί θα διαβαστούν παρά μόνο ότι ο τελευταίος είναι ο 0 (*τιμή-φρουρός*) θα χρησιμοποιήσουμε τη δομή **Όσο..επανάλαβε**.

```
Αλγόριθμος Άθροισμα_ακεραίων

   !Χρειαζόμαστε μία ειδική μεταβλητή-συσσωρευτή
   !(αθροιστή) που θα κρατάει το τρέχον άθροισμα.
   !Πρέπει να πάρει μία αρχική τιμή.
   Sum ← 0

   !Επαναληπτική είσοδος δεδομένων (οι ακέραιοι αριθμοί).
   !Κάθε φορά που διαβάζεται ένας προστίθεται στη Sum.
   !Πρώτα, διαβάζεται ένας αριθμός πριν ξεκινήσει η
   !επαναληπτική δομή.
   Διάβασε x

   Όσο x <> 0 επανάλαβε

      Sum ← Sum + x

      Διάβασε x

   Τέλος_επανάληψης

   !Έξοδος αποτελέσματος.
   Εκτύπωσε Sum

Τέλος Άθροισμα_ακεραίων
```

---

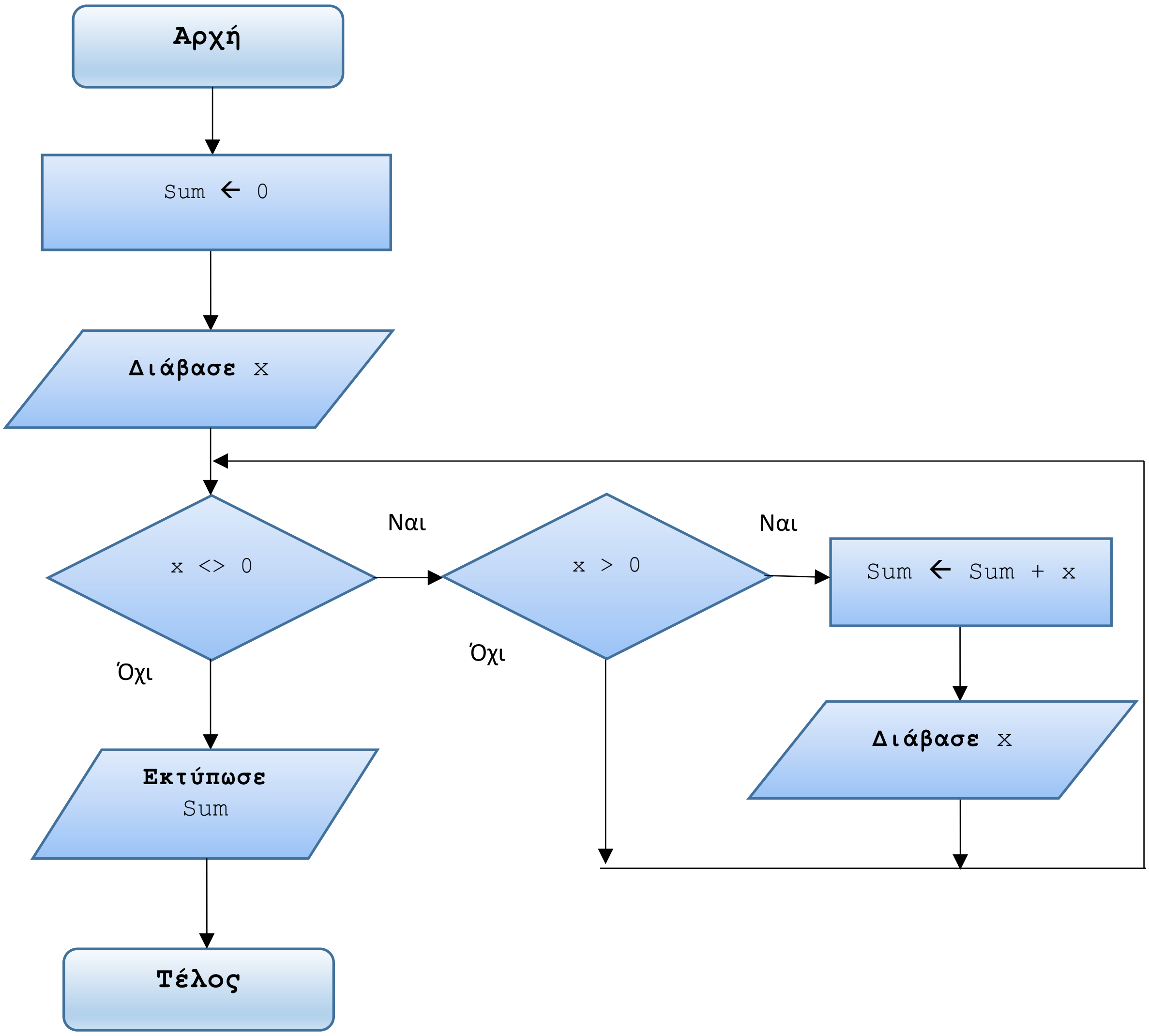
2. Γράψτε έναν αλγόριθμο που διαβάζει μία ακολουθία ακέραιων αριθμών που τελειώνει με 0 και υπολογίζει το άθροισμα μόνο των θετικών. Κάντε και το λογικό διάγραμμα.

Απάντηση

Το πρόβλημα είναι παρόμοιο με το προηγούμενο. Η διαφορά είναι ότι μόλις διαβασθεί ο αριθμός θα πρέπει να **ελεγχθεί** (με δομή `Αν..τότε`) αν είναι θετικός. Αν ναι, τον προσθέτει στον συσσωρευτή.

```
Αλγόριθμος Άθροισμα_Θετικών_ακεραίων

    !Χρειαζόμαστε μία ειδική μεταβλητή-συσσωρευτή
    !(αθροιστή) που θα κρατάει το τρέχον άθροισμα.
    !Πρέπει να πάρει μία αρχική τιμή.
    Sum ← 0

    !Επαναληπτική είσοδος δεδομένων (οι ακέραιοι αριθμοί).
    !Κάθε φορά που διαβάζεται ένας, ελέγχεται αν είναι
    !θετικός και αν ναι τότε προστίθεται στη Sum.
    !Πρώτα, διαβάζεται ένας αριθμός πριν ξεκινήσει η
    !επαναληπτική δομή.
    Διάβασε x

    Όσο x <> 0 επανάλαβε

        Αν x > 0 τότε

            Sum ← Sum + x

        Τέλος_αν

        Διάβασε x

    Τέλος_επανάληψης

    !Έξοδος αποτελέσματος.
    Εκτύπωσε Sum

Τέλος Άθροισμα_Θετικών_ακεραίων
```

Και το αντίστοιχο λογικό διάγραμμα:



3. Γράψτε έναν αλγόριθμο που διαβάζει μία ακολουθία ακέραιων αριθμών που τελειώνει με 0 και υπολογίζει:
α) Το πλήθος των αριθμών που διαβάστηκαν (εκτός από το 0).
β) Το πλήθος και το άθροισμα ξεχωριστά των θετικών και αρνητικών

Κάντε και το λογικό διάγραμμα.

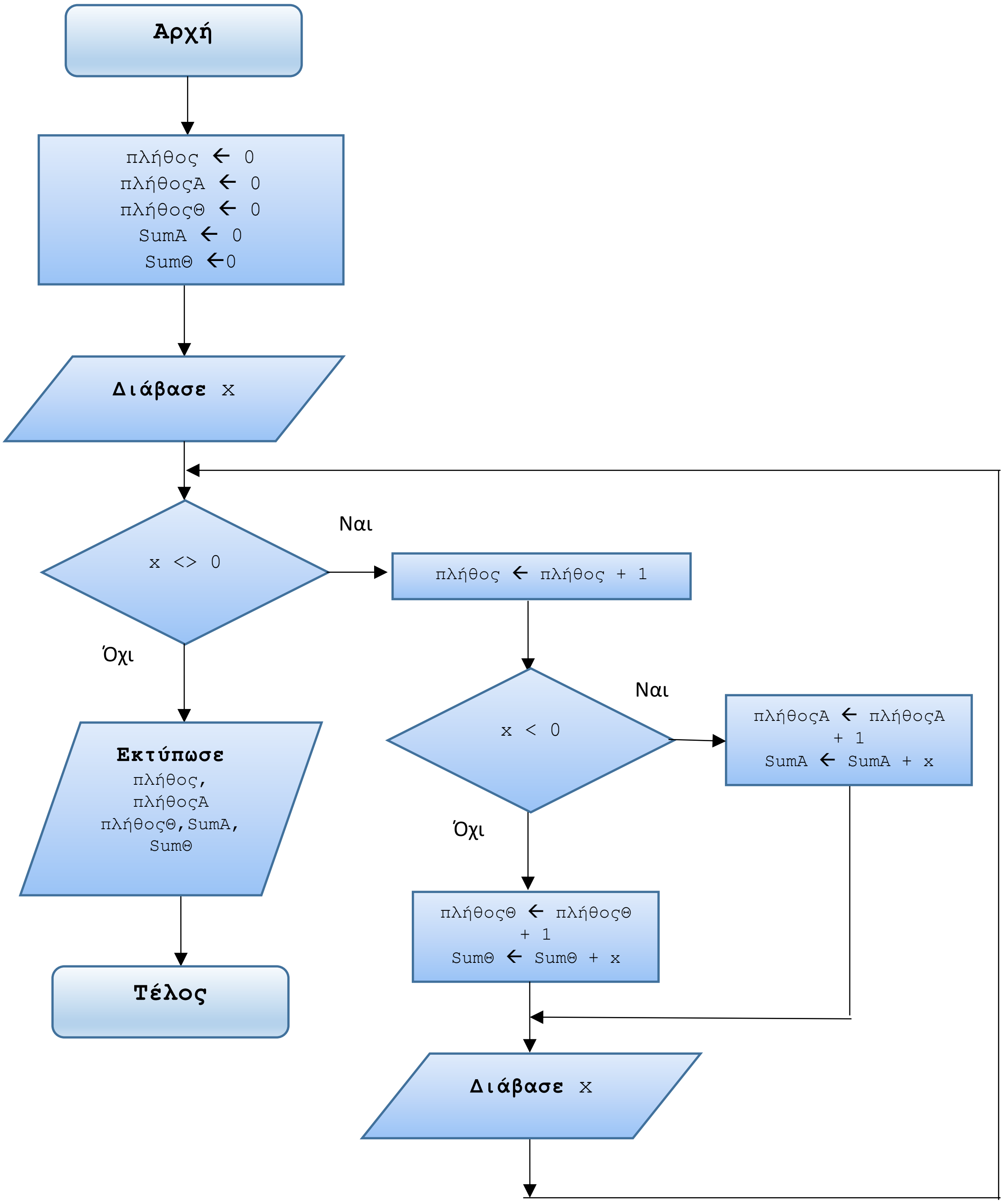
Απάντηση

Η λογική του αλγορίθμου είναι παρόμοια με των προηγούμενων ασκήσεων. Εδώ, θα χρειαστούμε εκτός από έναν συσσωρευτή (αθροιστή), συνήθως με το όνομα `Sum`, κι έναν μετρητή (counter) για την καταμέτρηση του πλήθους. Θα χρειαστούμε:

- Έναν γενικό μετρητή για την καταμέτρηση όλων των αριθμών που διαβάστηκαν.
- Δύο μετρητές και δύο αθροιστές ξεχωριστά για τους θετικούς και αρνητικούς.

```
Αλγόριθμος Καταμετρήσεις_και_αθροίσματα

   !Αρχικές τιμές.
   πλήθος ← 0    !γενικός μετρητής
   πλήθοςΑ ← 0  !μετρητής αρνητικών
   πλήθοςΘ ← 0  !μετρητής θετικών
   SumA ← 0  !αθροιστής αρνητικών
   SumΘ ← 0  !αθροιστής θετικών

   !Επαναληπτική είσοδος δεδομένων (οι ακέραιοι αριθμοί).
   !Πρώτα, διαβάζεται ένας αριθμός πριν ξεκινήσει η
   !επαναληπτική δομή.
   Διάβασε x

   Όσο x <> 0 επανάλαβε

      πλήθος ← πλήθος + 1

      Αν x < 0 τότε

         πλήθοςΑ ← πλήθοςΑ + 1
         SumA ← SumA + x

      αλλιώς

         πλήθοςΘ ← πλήθοςΘ + 1
         SumΘ ← SumΘ + x

      Τέλος_αν

      Διάβασε x

   Τέλος_επανάληψης

   !Έξοδος αποτελέσματος.
   Εκτύπωσε πλήθος, πλήθοςΑ, πλήθοςΘ, SumA, SumΘ

Τέλος Καταμετρήσεις_και_αθροίσματα
```

Και το αντίστοιχο λογικό διάγραμμα:



4. Σε συνέχεια του προηγούμενου, Με ΔΔ τους αριθμούς 4 7 −2 8 0 φτιάξτε τον πίνακα τιμών.

Απάντηση

Έχουμε:

Βήμα	x	x <> 0	πλήθος	x < 0	πλήθοςΑ	πλήθοςΘ	SumA	SumΘ
1ο			0					
2ο					0			
3ο						0		
4ο							0	
5ο								0
6ο	4							
7ο		ΑΛΗΘΗΣ						
8ο			(0+1) 1					
9ο				ΨΕΥΔΗΣ				
10ο						(0+1) 1		
11ο								(0+4) 4
12ο	7							
13ο		ΑΛΗΘΗΣ						
14ο			(1+1) 2					
15ο				ΨΕΥΔΗΣ				
16ο						(1+1) 2		
17ο								(4+7) 11
18ο	-2							
19ο		ΑΛΗΘΗΣ						
20ο			(2+1) 3					

21ο				ΑΛΗΘΗΣ				
22ο					(0+1) **1**			
23ο							(0+(-2)) **-2**	
24ο	8							
25ο		ΑΛΗΘΗΣ						
26ο			(3+1) **4**					
27ο				ΨΕΥΔΗΣ				
28ο						(2+1) **3**		
29ο								(11+8) **19**
30ο	0							
31ο		ΨΕΥΔΗΣ						

Με έντονο μπλε χρώμα φαίνονται οι τελικές τιμές των μεταβλητών.

---

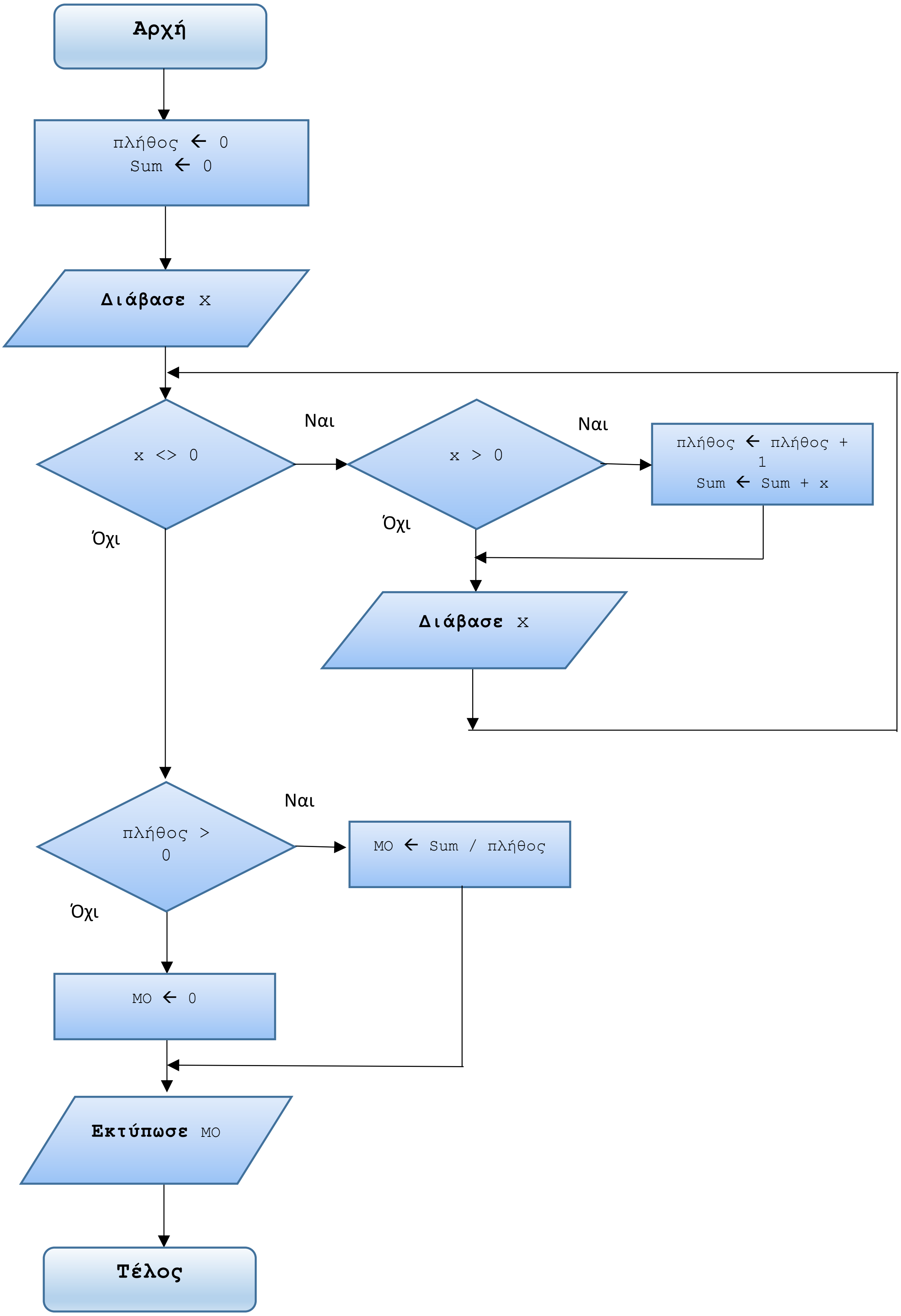
5. Να δοθεί αλγόριθμος που διαβάζει μία ομάδα ακέραιων αριθμών που τελειώνει με 0 και υπολογίζει τον μέσο όρο των θετικών.
Κάντε και το λογικό διάγραμμα.

Απάντηση

```
Αλγόριθμος Υπολογισμός_μέσου_όρου_θετικών

    !Αρχικές τιμές.
    πλήθος ← 0    !μετρητής θετικών
    Sum ← 0       !αθροιστής θετικών

    !Επαναληπτική είσοδος δεδομένων (οι ακέραιοι αριθμοί).
    !Πρώτα, διαβάζεται ένας αριθμός πριν ξεκινήσει η
    !επαναληπτική δομή.
    Διάβασε x
```

```
   Όσο x <> 0 επανάλαβε

      Αν x > 0 τότε

         πλήθος ← πλήθος + 1
         Sum ← Sum + x

      Τέλος_αν

      Διάβασε x

   Τέλος_επανάληψης

   !Κρίσιμο σημείο. Θα υπολογιστεί ο ΜΟ μόνο αν το
   !πλήθος > 0. Με άλλα λόγια, μόνο αν διαβάστηκε
   !τουλάχιστον ένας θετικός. Κι αυτό διότι, αν δεν
   !διαβάστηκε κανένας θετικός,  η μεταβλητή πλήθος θα
   !έχει 0 και δεν μπορεί να γίνει διαίρεση.

   Αν πλήθος > 0 τότε

         ΜΟ ← Sum / πλήθος

   αλλιώς

         ΜΟ ← 0

   Τέλος_αν

   !Έξοδος αποτελέσματος.
   Εκτύπωσε ΜΟ

Τέλος Υπολογισμός_μέσου_όρου_θετικών
```

Και το αντίστοιχο λογικό διάγραμμα:


Αρχή
πλήθος ← 0
Sum ← 0
Διάβασε x
x <> 0
Ναι
x > 0
Ναι
πλήθος ← πλήθος + 1
Sum ← Sum + x
Όχι
Όχι
Διάβασε x
πλήθος > 0
Ναι
ΜΟ ← Sum / πλήθος
Όχι
ΜΟ ← 0
Εκτύπωσε ΜΟ
Τέλος

6. Γράψτε έναν αλγόριθμο που διαβάζει μία ακολουθία ακέραιων αριθμών που τελειώνει με 9999 και εκτυπώνει εκείνους που διαιρούνται ακριβώς με το 5.

Απάντηση

```
Αλγόριθμος Εκτύπωση_αριθμών
    Διάβασε x
    Όσο x <> 9999 επανάλαβε
        Αν x MOD 5 = 0 τότε
            Εκτύπωσε x
        Τέλος_αν
        Διάβασε x
    Τέλος_επανάληψης
Τέλος Εκτύπωση_αριθμών
```

---

7. Να δοθεί αλγόριθμος που υπολογίζει το άθροισμα των ζυγών αριθμών από το 2 έως το 100.
Κάντε και το λογικό διάγραμμα.

Απάντηση

Επειδή ο αριθμός των επαναλήψεων είναι καθορισμένος μπορεί να επιλυθεί με την **Όσο..επανάλαβε** αλλά καλύτερα με την **Για..από..μέχρι**. Θα δούμε και με τους 2 τρόπους:

1ος τρόπος:

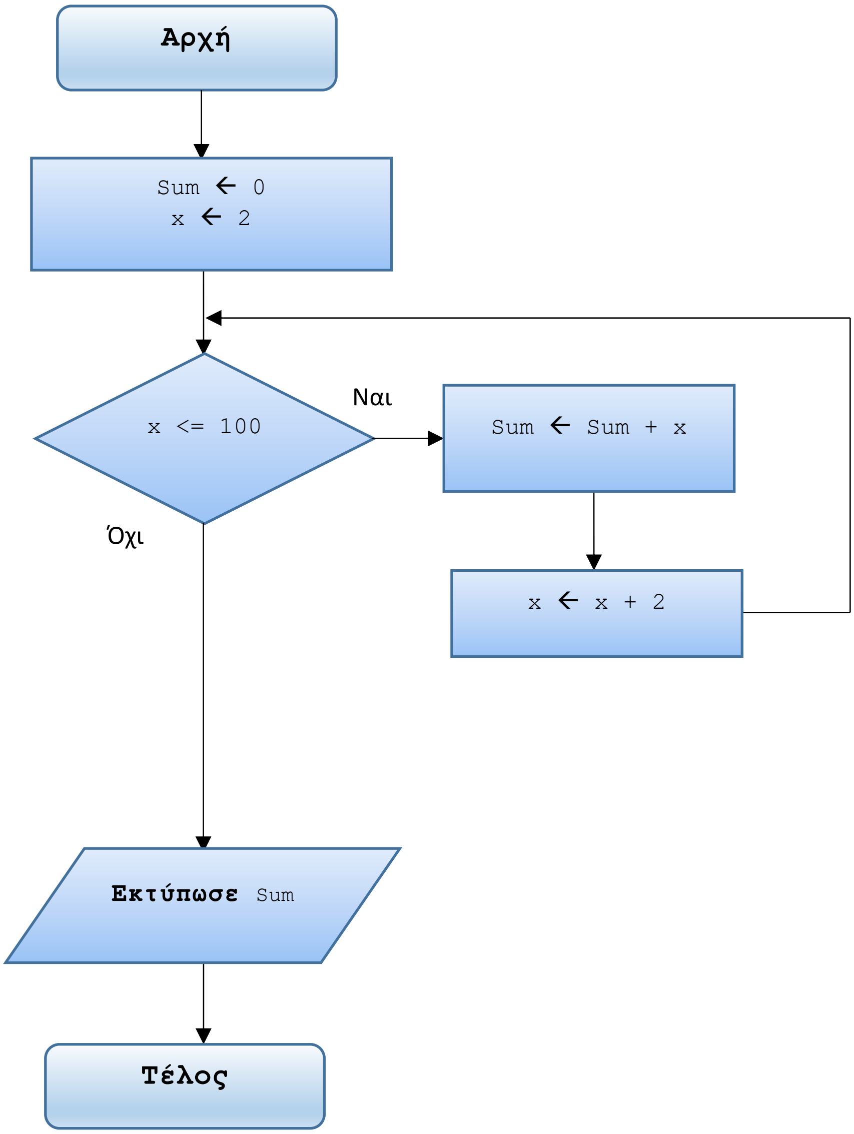
```
Αλγόριθμος Άθροισμα_ζυγών
    !Αρχικές τιμές.
    Sum ← 0        !αθροιστής ζυγών.
```

```
    !Ξεκίνα από το 2.
    x ← 2

    Όσο x <= 100 επανάλαβε

        Sum ← Sum + x

        x ← x + 2  !Το βήμα αύξησης της x είναι κατά 2.

    Τέλος_επανάληψης

    !Έξοδος αποτελέσματος.
    Εκτύπωσε Sum

Τέλος Άθροισμα_ζυγών
```

2ος τρόπος:

```
Αλγόριθμος Άθροισμα_ζυγών

    !Αρχικές τιμές.
    Sum ← 0       !αθροιστής ζυγών.

    Για x από 2 μέχρι 100 με_βήμα 2

        Sum ← Sum + x

    Τέλος_επανάληψης

    !Έξοδος αποτελέσματος.
    Εκτύπωσε Sum

Τέλος Άθροισμα_ζυγών
```

Και το αντίστοιχο λογικό διάγραμμα:



8. Να δοθεί αλγόριθμος που διαβάζει μία ομάδα ακέραιων αριθμών που τελειώνει με 0 και υπολογίζει
α) Τον μέγιστο όλων των αριθμών που διαβάστηκαν (εκτός του 0).
β) Τον μέγιστο μόνο των αρνητικών αριθμών που διαβάστηκαν.

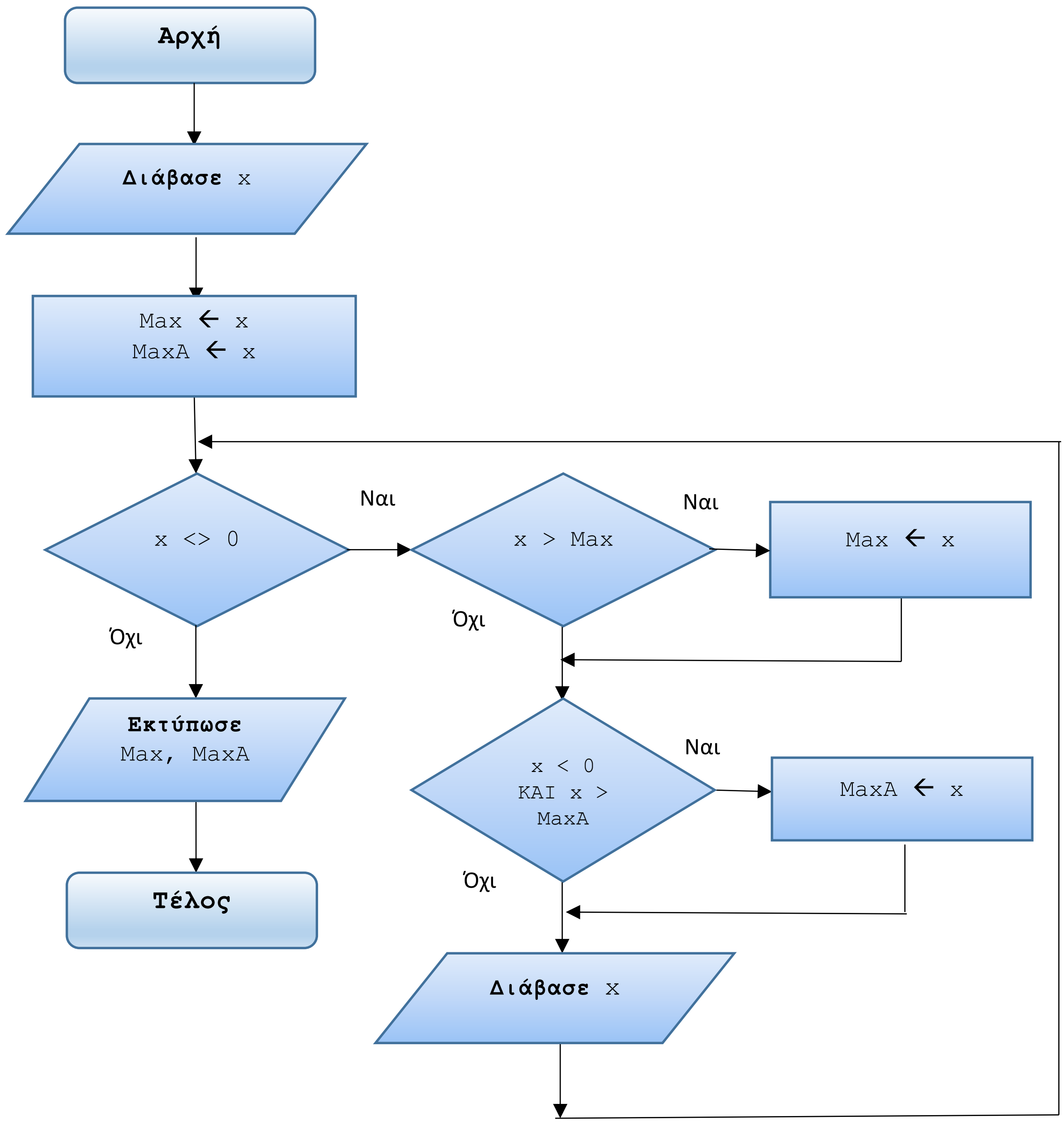
Κάντε και το λογικό διάγραμμα.

Απάντηση

Για το αποτέλεσμα θα χρειαστούμε δύο μεταβλητές, μία για τον μέγιστο όλων των αριθμών που διαβάστηκαν και μία για τον μέγιστο μόνο των αρνητικών.

```
Αλγόριθμος Υπολογισμός_μεγίστων

   !Ως αρχική τιμή των μεταβλητών αποτελέσματος θα είναι ο
   !πρώτος αριθμός που διαβάστηκε.
   Διάβασε x
   Max ← x   !μέγιστος όλων
   MaxA ← x  !μέγιστος αρνητικών

   Όσο x <> 0 επανάλαβε

     !Έλεγξε αν ο τρέχον αριθμός x που διαβάστηκε είναι
     !μεγαλύτερος του τρέχοντος γενικού μέγιστου Max.
      Αν x > Max  τότε

         Max ← x

      Τέλος_αν

     !Τώρα, έλεγξε αν ο τρέχον αριθμός x που διαβάστηκε
     !είναι αρνητικός και μεγαλύτερος του τρέχοντος
     !μέγιστου των αρνητικών MaxA.
      Αν x < 0  ΚΑΙ x > MaxA τότε

         MaxA ← x

      Τέλος_αν

      !Διάβασε τον επόμενο αριθμό x της ομάδας.
      Διάβασε x

   Τέλος_επανάληψης

   !Έξοδος αποτελέσματος.
   Εκτύπωσε Max, MaxA

Τέλος Υπολογισμός_μεγίστων
```

Και το αντίστοιχο λογικό διάγραμμα:



9. Γράψτε έναν αλγόριθμο που διαβάζει Ν ακέραιους αριθμούς και υπολογίζει το μέσο όρο τους.
Κάντε και το λογικό διάγραμμα.

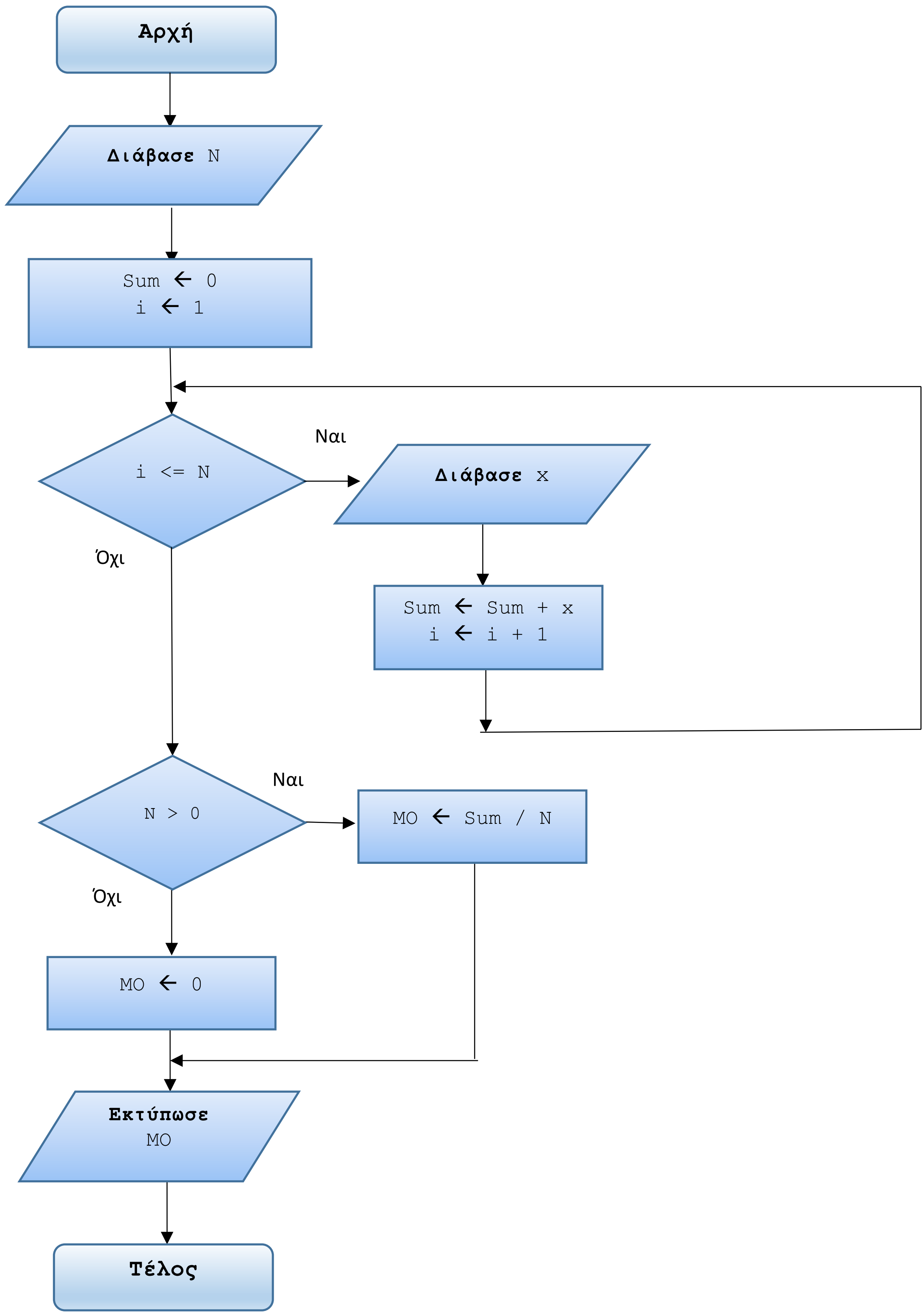
Απάντηση

Επειδή η επαναληπτική διαδικασία διαβάσματος των αριθμών είναι καθορισμένη (Ν επαναλήψεις) η καταλληλότερη δομή επανάληψης είναι η **Για..από..μέχρι**

Αλγόριθμος Υπολογισμός_μέσου_όρου_Ν_αριθμών

```
Δεδομένα // N //

   !Αρχικές τιμές.
   Sum ← 0  !αθροιστής.

   !Χρήση της μεταβλητής i ως μετρητής επαναλήψεων.
   Για i από 1 μέχρι N

      Διάβασε x
      Sum ← Sum + x

   Τέλος_επανάληψης

   !Εύρεση του μέσου όρου. Προσοχή χρειάζεται η N να έχει
   !τιμή > 0 (με άλλα λόγια, να διαβαστεί τουλάχιστον ένας
   !αριθμός).
   !Αν η εκφώνηση έλεγε ρητά N > 0 τότε δεν χρειαζόταν ο
   !έλεγχος αυτός.
   Αν N > 0  τότε

         MO ← Sum / N

   αλλιώς

         MO ← 0

   Τέλος_αν


Αποτελέσματα // MO //

Τέλος Υπολογισμός_μέσου_όρου_N_αριθμών
```

Και το αντίστοιχο λογικό διάγραμμα:


Αρχή
Διάβασε N
Sum ← 0
i ← 1
i <= N
Ναι
Διάβασε x
Sum ← Sum + x
i ← i + 1
Όχι
N > 0
Ναι
MO ← Sum / N
Όχι
MO ← 0
Εκτύπωσε MO
Τέλος

10. Γράψτε ένα τμήμα αλγορίθμου χρησιμοποιώντας τη δομή **Αρχή..Μέχρις_ότου** όπου διαβάζει έναν αριθμό και ελέγχει αν αυτός είναι μεταξύ 1 και 20.

Απάντηση

Πρόκειται για τυπικό δείγμα αλγορίθμου που ελέγχει την εγκυρότητα εισαγόμενων δεδομένων, κυρίως αριθμών.

```
Αρχή

   Διάβασε x

   !Εδώ κάνουμε έναν έλεγχο κι εμφανίζουμε ένα μήνυμα
   !σφάλματος αν ο αριθμός δεν είναι έγκυρος.
   Αν 'ΟΧΙ(x >= 1 ΚΑΙ x <= 20) τότε

      Εκτύπωσε 'Ο αριθμός δεν είναι έγκυρος'

   Τέλος_αν

Μέχρις_ότου  x >= 1 ΚΑΙ x <= 20
```

11. Γράψτε έναν αλγόριθμο διαβάζει έναν ακέραιο μεταξύ 1 και 80 και τυπώνει ισάριθμους χαρακτήρες '*' (άστρο).

Απάντηση

Επειδή η επαναληπτική διαδικασία διαβάσματος των αριθμών είναι καθορισμένη, η καταλληλότερη δομή επανάληψης είναι η **Για..από..μέχρι**

```
Αλγόριθμος Εκτύπωση_αστεριών

   !Είσοδος δεδομένων. Πόσα αστέρια θα τυπώσει. Πρέπει,
   !όμως, να ελέγξουμε ότι ο αριθμός που διαβάστηκε είναι
   !μεταξύ 1 και 80 (Έλεγχος εγκυρότητας).
   Αρχή

      Διάβασε σύνολο_αστεριών

      Αν 'ΟΧΙ(σύνολο_αστεριών >= 1 ΚΑΙ
                              σύνολο_αστεριών <= 80) τότε

         Εκτύπωσε 'Ο αριθμός δεν είναι έγκυρος'

       Τέλος_αν
```

```
  Μέχρις_ότου σύνολο_αστεριών >= 1 ΚΑΙ
                                  σύνολο_αστεριών <= 80

  !Χρήση της μεταβλητής i ως μετρητής επαναλήψεων.
   Για i από 1 μέχρι σύνολο_αστεριών

      Εκτύπωσε '*'

   Τέλος_επανάληψης

Τέλος Εκτύπωση_αστεριών
```

---

12. Να δοθεί αλγόριθμος που διαβάζει ένα σύνολο χαρακτήρων που τελειώνει με # και υπολογίζει πόσοι χαρακτήρες διαβάστηκαν.
Κάντε και το λογικό διάγραμμα.

Απάντηση

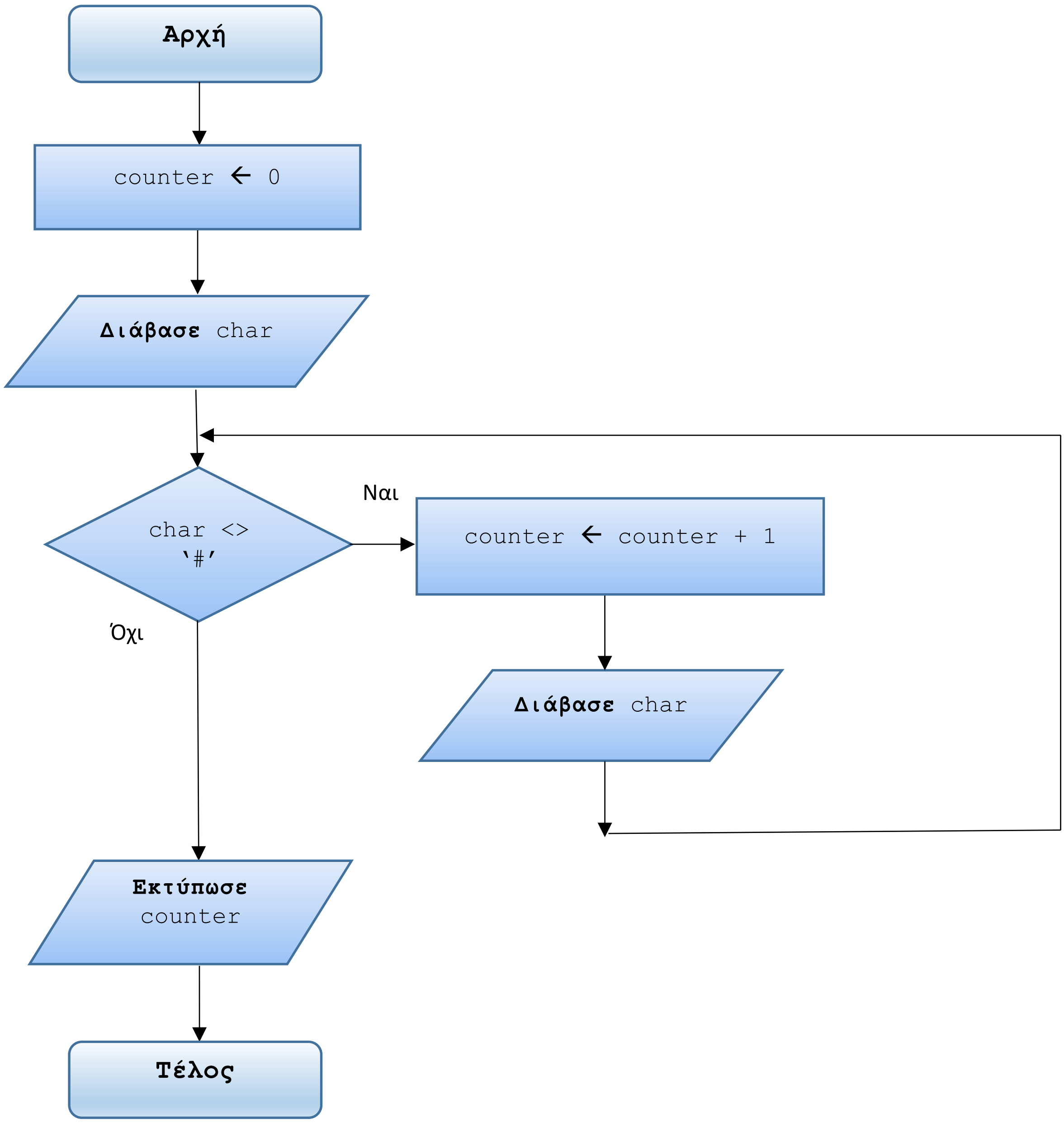
Επειδή η επαναληπτική διαδικασία διαβάσματος των χαρακτήρων δεν είναι καθορισμένη (δεν γνωρίζουμε πόσοι χαρακτήρες θα διαβαστούν παρά μόνο ότι ο τελευταίος είναι ο #) , η καταλληλότερη δομή επανάληψης είναι η **Όσο..επανάλαβε.**

Επίσης, θα χρειαστούμε μία **μεταβλητή-μετρητή** (συνήθως της δίνουμε το όνομα **counter**) που θα κρατάει το τρέχον σύνολο χαρακτήρων.

```
Αλγόριθμος Υπολογισμός_πλήθους_χαρακτήρων

   !Αρχική τιμή μετρητή.
   counter ← 0

   !Η πρώτη τιμή, ως γνωστόν, διαβάζεται πριν ξεκινήσει η
   !επανάληψη.
   Διάβασε char

   Όσο char <> '#' επανάλαβε

     counter ← counter + 1

     !Διάβασε την επόμενη τιμή.
     Διάβασε char

   Τέλος_επανάληψης
```

```
    !Έξοδος αποτελέσματος.
    Εκτύπωσε counter

Τέλος Υπολογισμός_πλήθους_χαρακτήρων
```

Και το αντίστοιχο λογικό διάγραμμα:



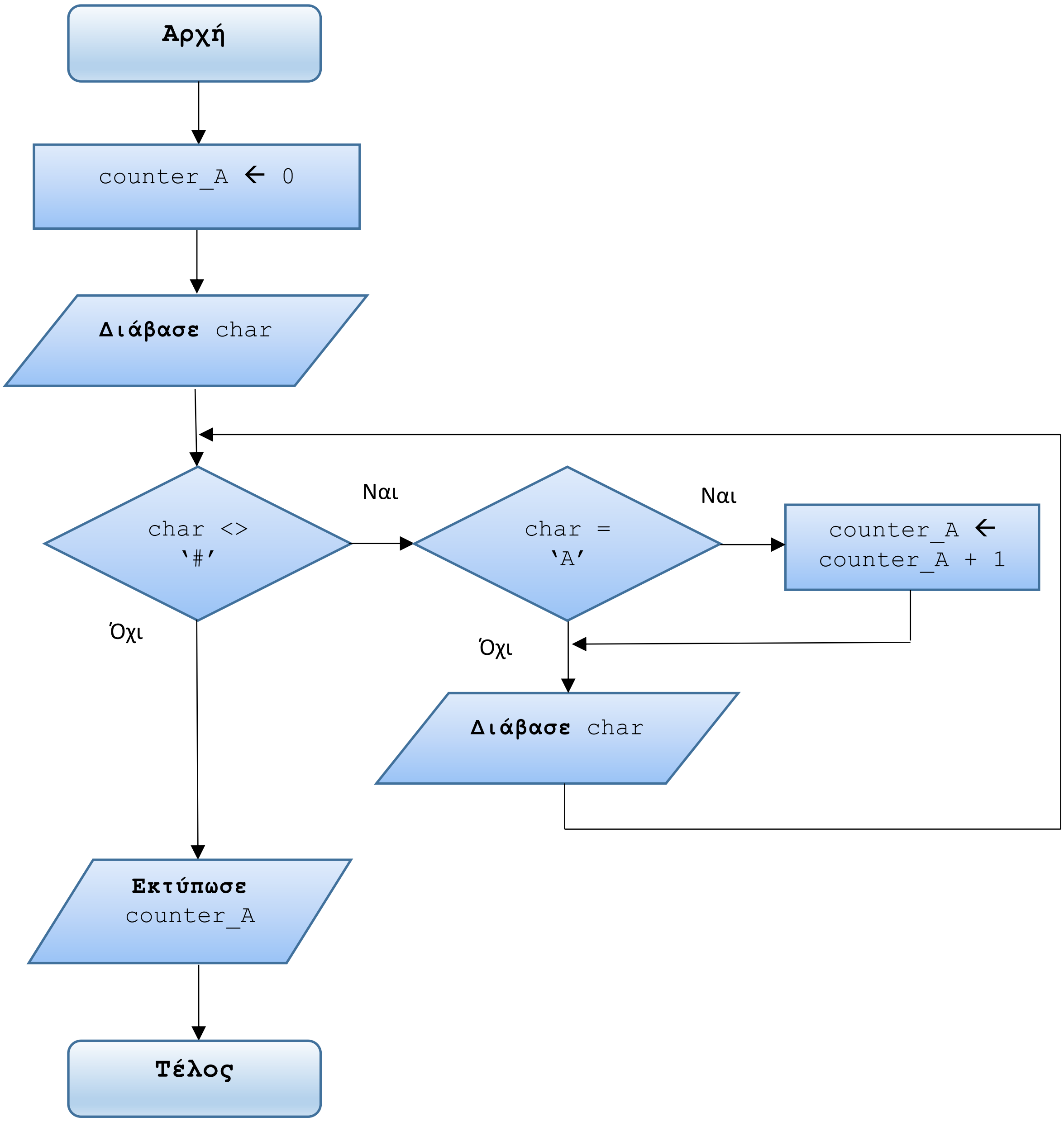
13. Να δοθεί αλγόριθμος που διαβάζει ένα σύνολο χαρακτήρων που τελειώνει με # και τυπώνει πόσα ''Α'' διαβάστηκαν.
Κάντε και το λογικό διάγραμμα.

Απάντηση

Η άσκηση αυτή μοιάζει με την προηγούμενη με τη διαφορά ότι όταν διαβάζεται ένας χαρακτήρας ελέγχεται αν είναι ο 'Α'. Αν ναι, τότε προσθέτουμε 1 σε έναν μετρητή.

```
Αλγόριθμος Υπολογισμός_πλήθους_χαρακτήρων_Α

   !Αρχική τιμή μετρητή.
   counter_A ← 0

   !Η πρώτη τιμή, ως γνωστόν, διαβάζεται πριν ξεκινήσει η
   !επανάληψη.
   Διάβασε char

   Όσο char <> '#' επανάλαβε

      Αν char = 'A' τότε

         counter_A ← counter_A + 1

      Τέλος_αν

     !Διάβασε την επόμενη τιμή.
     Διάβασε char

   Τέλος_επανάληψης

   !Έξοδος αποτελέσματος.
   Εκτύπωσε counter_A

Τέλος Υπολογισμός_πλήθους_χαρακτήρων_Α
```

Και το αντίστοιχο λογικό διάγραμμα:



14. Γράψτε έναν αλγόριθμο που μετράει και εκτυπώνει όλους τους θετικούς τετραψήφιους ακέραιους αριθμούς που μπορούν να αναγνωστούν και ανάποδα.

Παραδείγματα τέτοιων αριθμών είναι ο 1331, 6666, 7447 κ.α. Βοήθεια: Βρείτε πρώτα τα ψηφία των χιλιάδων, εκατοντάδων, δεκάδων και μονάδων.

Απάντηση

Εδώ θα χρησιμοποιήσουμε εκτενώς τους ακέραιους τελεστές **DIV** και **MOD** προκειμένου να βρούμε ένα-ένα τα ψηφία των χιλιάδων, εκατοντάδων, δεκάδων και μονάδων. Εφόσον γίνει αυτό, τότε απλά θα συγκρίνουμε το ψηφίο των χιλιάδων αν ισούται με αυτό των μονάδων και εκείνο των εκατοντάδων με αυτό των δεκάδων.

```
Αλγόριθμος Έλεγχος_τετραψηφίων_αν_διαβάζεται_ανάποδα

   !Αρχική τιμή μετρητή.
   counter_ανάποδοι ← 0

   !Σάρωσε όλους τους θετικούς τετραψήφιους κι έλεγξε
   !καθέναν αν είναι ανάποδος.
   Για x από 1000 μέχρι 9999

      !Πάρε το ψηφίο των χιλιάδων και με το υπόλοιπο
      !συνέχισε παρακάτω.
      ψηφίο_χιλιάδων ← x DIV 1000
      υπόλοιπο ← x MOD 1000

      !Πάρε το ψηφίο των εκατοντάδων και με το νέο
      !υπόλοιπο συνέχισε παρακάτω.
      ψηφίο_εκατοντάδων ← υπόλοιπο DIV 100
      υπόλοιπο ← υπόλοιπο MOD 100

      !Πάρε το ψηφίο των δεκάδων και το νέο τελικό
      !υπόλοιπο είναι το ψηφίο των μονάδων.
      ψηφίο_δεκάδων ← υπόλοιπο DIV 10
      ψηφίο_μονάδων ← υπόλοιπο MOD 10

      !Κάνε τον έλεγχο.
      Αν ψηφίο_χιλιάδων = ψηφίο_μονάδων ΚΑΙ
         ψηφίο_εκατοντάδων = ψηφίο_δεκάδων τότε

         counter_ανάποδοι ← counter_ανάποδοι + 1
         Εκτύπωσε 'Ο ', x, ' διαβάζεται επίσης ανάποδα'

      Τέλος_αν

  Τέλος_επανάληψης

  Εκτύπωσε 'Το πλήθος που βρέθηκε είναι ',
                                          counter_ανάποδοι

Τέλος Έλεγχος_τετραψήφιου_αν_διαβάζεται_ανάποδα
```

Ας πάρουμε ένα παράδειγμα: Έστω `x ← 1331`. Τότε, σύμφωνα με τον αλγόριθμο θα γίνουν οι εξής υπολογισμοί:

```
ψηφίο_χιλιάδων ← 1331 DIV 1000 θα δώσει 1
υπόλοιπο ← 1331 MOD 1000 θα δώσει 331

ψηφίο_εκατοντάδων ← 331 DIV 100 θα δώσει 3
υπόλοιπο ← 331 MOD 100 θα δώσει 31

ψηφίο_δεκάδων ← 31 DIV 10 θα δώσει 3
ψηφίο_μονάδων ← 31 MOD 10 θα δώσει 1
```

Ο αναγνώστης μπορεί να δοκιμάσει με τους αριθμούς 6666, 7447, 9229, 9999

---

15. Να δοθεί αλγόριθμος που υπολογίζει το άθροισμα 1-1/2+1/3-1/4+1/5+…. για ν όρους. Το ν θα δίνεται ως δεδομένο.

Απάντηση

Παρατηρούμε μία εναλλαγή του προσήμου σε κάθε όρο που προστίθεται καθώς και μία μεταβλητότητα στον παρονομαστή. Συγκεκριμένα:

α) Οι όροι με ζυγό παρονομαστή έχουν αρνητικό πρόσημο και οι όροι με περιττό παρονομαστή έχουν θετικό πρόσημο.
β) ο παρονομαστής ενός όρου είναι αυξημένος κατά 1 από τον παρονομαστή του προηγούμενου όρου.

```
Αλγόριθμος Υπολογισμός_αθροίσματος_σειράς
Δεδομένα // ν //

   !Αρχική τιμή αθροιστή.
   Sum ← 0

   !Σάρωσε όλους τους θετικού τετραψήφιους κι έλεγξε
   !καθέναν αν είναι ανάποδος.
   Για i από 1 μέχρι ν

      Αν i MOD 2 = 0 τότε

         Sum ← Sum - 1 / i

      αλλιώς
```

```
              Sum ← Sum + 1 / i
           Τέλος_αν
       Τέλος_επανάληψης

     Αποτελέσματα // Sum //
     Τέλος Υπολογισμός_αθροίσματος_σειράς
```

---

16. Να γράψετε αλγόριθμο που υπολογίζει την τιμή της συνάρτησης

$$y(x) = x^2 - 3x + 2$$

για τις τιμές του x από –1 έως 3 με βήμα 0.1

α) Διαμορφώστε τον αλγόριθμο με τη δομή **Όσο…επανάλαβε**
β) Διαμορφώστε τον αλγόριθμο με τη δομή **Για…από..μέχρι**
γ) Ποιά θεωρείτε πιο κατάλληλη;
δ) Κάντε και το λογικό διάγραμμα.
.

Απάντηση

α)

```
Αλγόριθμος Υπολογισμός_τιμών_συνάρτησης

   !Αρχική τιμή του x.
   x ← -1

   Όσο x <= 3 επανάλαβε

      y ← x^2 - 3*x + 2

      Εκτύπωσε x, y

      !Αύξησε την τιμή της x κατά 0.1
      x ←  x + 0.1

  Τέλος_επανάληψης

Τέλος Υπολογισμός_τιμών_συνάρτησης
```

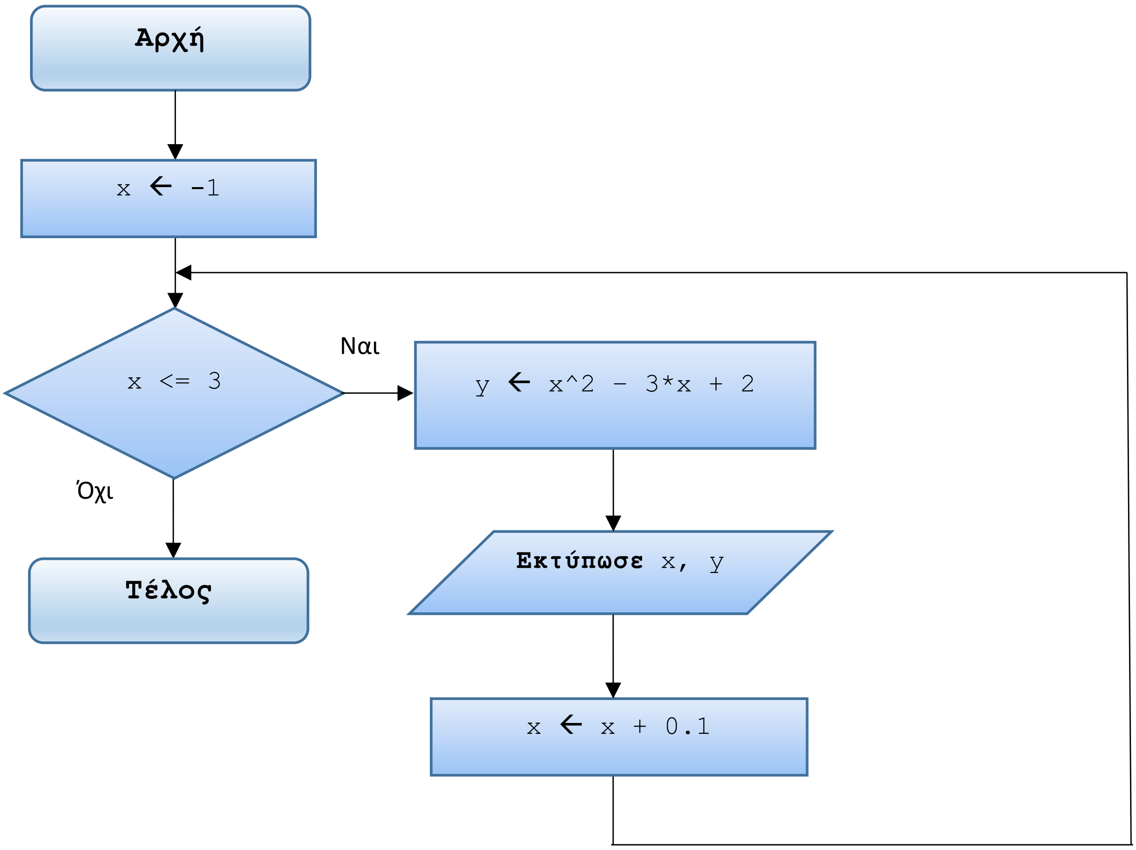
β)

```
Αλγόριθμος Υπολογισμός_τιμών_συνάρτησης
```

```
    Για x από -1 μέχρι 3 με_βήμα 0.1
        y ← x^2 - 3*x + 2
        Εκτύπωσε x, y
    Τέλος_επανάληψης
Τέλος Υπολογισμός_τιμών_συνάρτησης
```

γ) Η δομή **Για..από..μέχρι** είναι πιο κατάλληλη διότι γνωρίζουμε την αρχική και τελική τιμή και ο αριθμός των επαναλήψεων είναι καθορισμένος.

δ) Το λογικό διάγραμμα είναι το εξής:



17. Σε μία ακολουθία αριθμών κάθε όρος είναι το άθροισμα των 2 προηγούμενων (*ακολουθία Fibonacci*). Π.χ. 1 2 3 5 8 13 21 κλπ. Να δοθεί αλγόριθμος που λαμβάνει ως είσοδο τους δύο πρώτους όρους και υπολογίζει τους 20 επόμενους.

Απάντηση

```
Αλγόριθμος Fibonacci

   !Είσοδος των 2 πρώτων όρων.
   Διάβασε α, β

   !Επεξεργασία.
   Για i από 1 μέχρι 20

      !Το άθροισμα των 2 προηγούμενων όρων.
      Sum ← α + β
      Εκτύπωσε Sum

      !Πριν προχωρήσει στην επόμενη επανάληψη, για τον
      !υπολογισμό του επόμενου όρου αναδιάταξε τους 2
      !προηγούμενους όρους.
      α ← β
      β ← Sum

  Τέλος_επανάληψης

Τέλος Fibonacci
```

---

18. Να δοθεί αλγόριθμος που παίρνει ως είσοδο έναν ακέραιο N και υπολογίζει το παραγοντικό του. Ως γνωστόν είναι N! = 1x2x3x…..xN Π.χ. για 4!=1x2x3x4. Υπόψη ότι 0!=1.

Απάντηση

```
Αλγόριθμος Παραγοντικό_ακεραίου

   !Είσοδος του ακεραίου.
   Διάβασε N

   !Επεξεργασία.
   Αν N = 0 τότε
```

```
            παραγοντικό ← 0
        αλλιώς
            !Αρχική τιμή μεταβλητής που κρατάει το παραγοντικό.
            παραγοντικό ← 1    !όχι 0. Θα γίνουν πολλαπ/σμοί.
            Για i από 2 μέχρι N
                παραγοντικό ← παραγοντικό * i
            Τέλος_επανάληψης
        Τέλος_αν
        !Εκτύπωση αποτελέσματος.
        Εκτύπωσε N, παραγοντικό
    Τέλος Παραγοντικό_ακεραίου
```

---

19. Γράψτε έναν αλγόριθμο που τυπώνει την προπαίδεια (χρησιμοποιείστε δύο εμφωλευμένα **Για.. από..μέχρι**).

Απάντηση

Για την επίλυσή του θα χρειαστούμε 2 εμφωλευμένες δομές επανάληψης **Για..από..μέχρι**

```
Αλγόριθμος Προπαίδεια
   !Εξωτερική επανάληψη.
   Για i από 1 μέχρι 10
       !Εσωτερική επανάληψη.
       Για j από 1 μέχρι 10
          Εκτύπωσε i * j
       Τέλος_επανάληψης
   Τέλος_επανάληψης
Τέλος Προπαίδεια
```

20. Δίνεται ο παρακάτω αλγόριθμος:

```
Αλγόριθμος Μετρήσεις
  Διάβασε x
  Όσο  x >= 1 ΚΑΙ x <= 20 επανάλαβε
    Αν x >= 1 ΚΑΙ x <= 10  τότε
        μετρητής1 ← μετρητής1 + 1
    αλλιώς
        μετρητής2 ← μετρητής2 + 1
    Τέλος_αν
    Διάβασε x
  Τέλος_επανάληψης
  Εκτύπωσε μετρητής1, μετρητής2
Τέλος Μετρήσεις
```

α) Περιγράψτε τί κάνει ο παραπάνω αλγόριθμος.
β) Τί αποθηκεύει ο `μετρητής1` και τί ο `μετρητής2`.
γ) Τί σημαντικό έχουμε ξεχάσει όσον αφορά τις μεταβλητές των μετρητών;
δ) Κάντε τον πίνακα τιμών με τα εξής ΔΔ (δοκιμαστικά δεδομένα):
- **δ1)** `4, 6, 15, 10, 8, 0`
- **δ2)** `17, 18, 20, 7, 1, 0`
- **δ3)** `16, 11, 21`
- **δ4)** `0`

ε) Κάντε το αντίστοιχο λογικό διάγραμμα.

Απάντηση

α) Ο αλγόριθμος διαβάζει μία ακολουθία αριθμών από το 1 μέχρι και το 20 και υπολογίζει:
- Πόσοι είναι στο διάστημα [1,10]. Το πλήθος τους αποθηκεύεται στη μεταβλητή `μετρητής1`.
- Πόσοι είναι στο διάστημα (10,20]. Το πλήθος τους αποθηκεύεται στη μεταβλητή `μετρητής2`.

Μόλις διαβαστεί κάποιος αριθμός εκτός του διαστήματος [1,20] η επανάληψη τελειώνει και εκτυπώνεται η τιμή των μετρητών.

β) Το πλήθος των αριθμών στο διάστημα [1,10] αποθηκεύεται στη μεταβλητή `μετρητής1`.
Το πλήθος των αριθμών στο διάστημα (10,20] αποθηκεύεται στη μεταβλητή `μετρητής2`.

γ) Πριν ξεκινήσει η επανάληψη **πρέπει οι μετρητές να αρχικοποιηθούν**, **δηλαδή να πάρουν αρχικές τιμές**. Συνεπώς, έχουν ξεχαστεί οι εξής εντολές:

```
μετρητής1 ← 0
μετρητής2 ← 0
```

δ)

δ1) ΔΔ: 4, 6, 15, 10, 8, 0

ΑΑ	x	x >= 1 ΚΑΙ x <= 20	x >= 1 ΚΑΙ x <= 10	μετρητής1	μετρητής2
1				0	
2					0
3	4				
4		ΑΛΗΘΗΣ			
5			ΑΛΗΘΗΣ		
6				0+1 = 1	
7	6				
8		ΑΛΗΘΗΣ			
9			ΑΛΗΘΗΣ		
10				1+1 = 2	
11	15				
12		ΑΛΗΘΗΣ			
13			ΨΕΥΔΗΣ		
14					0+1 = **1**
15	10				
16		ΑΛΗΘΗΣ			
17			ΑΛΗΘΗΣ		
18				2+1 = 3	
19	8				
20		ΑΛΗΘΗΣ			
21			ΑΛΗΘΗΣ		
22				3+1 = **4**	

23	0				
24		ΨΕΥΔΗΣ			

Οι τελικές τιμές είναι `μετρητής1=4` και `μετρητής2=1`

δ2) **ΔΔ:** **`17, 18, 20, 7, 1, 0`**

ΑΑ	x	x >= 1 ΚΑΙ x <= 20	x >= 1 ΚΑΙ x <= 10	μετρητής1	μετρητής2
1				0	
2					0
3	17				
4		ΑΛΗΘΗΣ			
5			ΨΕΥΔΗΣ		
6					0+1 = 1
7	18				
8		ΑΛΗΘΗΣ			
9			ΨΕΥΔΗΣ		
10					1+1 = 2
11	20				
12		ΑΛΗΘΗΣ			
13			ΨΕΥΔΗΣ		
14					2+1 = **3**
15	7				
16		ΑΛΗΘΗΣ			
17			ΑΛΗΘΗΣ		
18				0+1 = 1	
19	1				
20		ΑΛΗΘΗΣ			
21			ΑΛΗΘΗΣ		
22				1+1 = **2**	

23	0				
24		ΨΕΥΔΗΣ			

Οι τελικές τιμές είναι `μετρητής1=2` και `μετρητής2=3`

δ3) ΔΔ: `16, 11, 21`

ΑΑ	x	x >= 1 ΚΑΙ x <= 20	x >= 1 ΚΑΙ x <= 10	μετρητής1	μετρητής2
1				**0**	
2					0
3	16				
4		ΑΛΗΘΗΣ			
5			ΨΕΥΔΗΣ		
6					0+1 = 1
7	11				
8		ΑΛΗΘΗΣ			
9			ΨΕΥΔΗΣ		
10					1+1 = **2**
11	21				
12		ΨΕΥΔΗΣ			

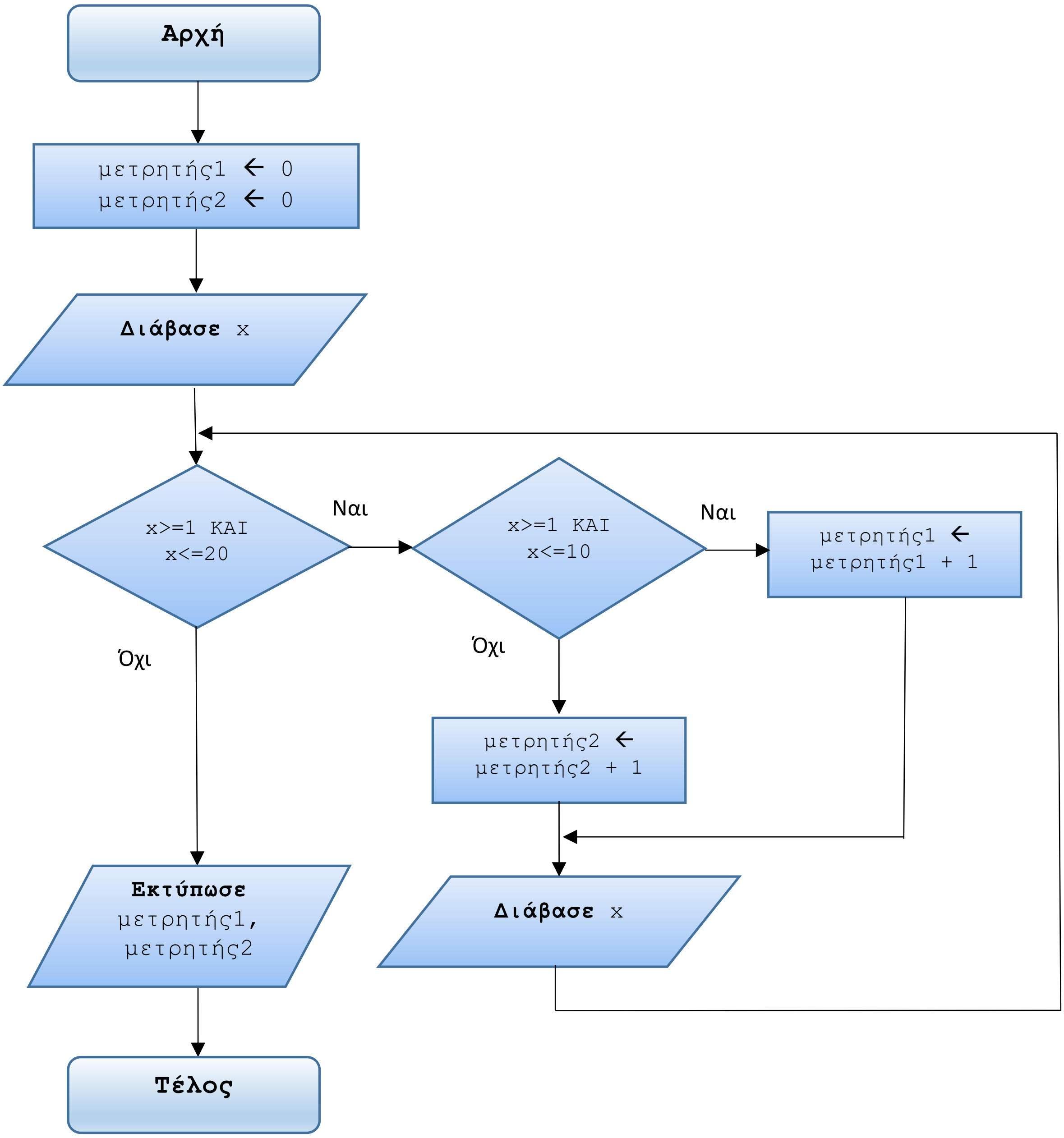
Οι τελικές τιμές είναι `μετρητής1=0` και `μετρητής2=2`

δ4) ΔΔ: 0

ΑΑ	x	x >= 1 ΚΑΙ x <= 20	x >= 1 ΚΑΙ x <= 10	μετρητής1	μετρητής2
1				0	
2					0
3	0				
4		ΨΕΥΔΗΣ			

Οι τελικές τιμές είναι `μετρητής1=0` και `μετρητής2=0`

ε) Το αντίστοιχο λογικό διάγραμμα είναι το εξής:



---

21. Γράψτε έναν αλγόριθμο που διαβάζει 50 ακέραιους αριθμούς και υπολογίζει:
- **α)** Πόσοι είναι διψήφιοι και το ποσοστό τους.
- **β)** Πόσοι είναι τριψήφιοι και το ποσοστό τους.
- **γ)** Το σύνολο των διψήφιων και τριψήφιων και το συνολικό ποσοστό τους.
- **δ)** Κάντε και το λογικό διάγραμμα.

Σημείωση: Προσέξτε ότι διψήφιοι και τριψήφιοι μπορεί να είναι θετικοί ή αρνητικοί.

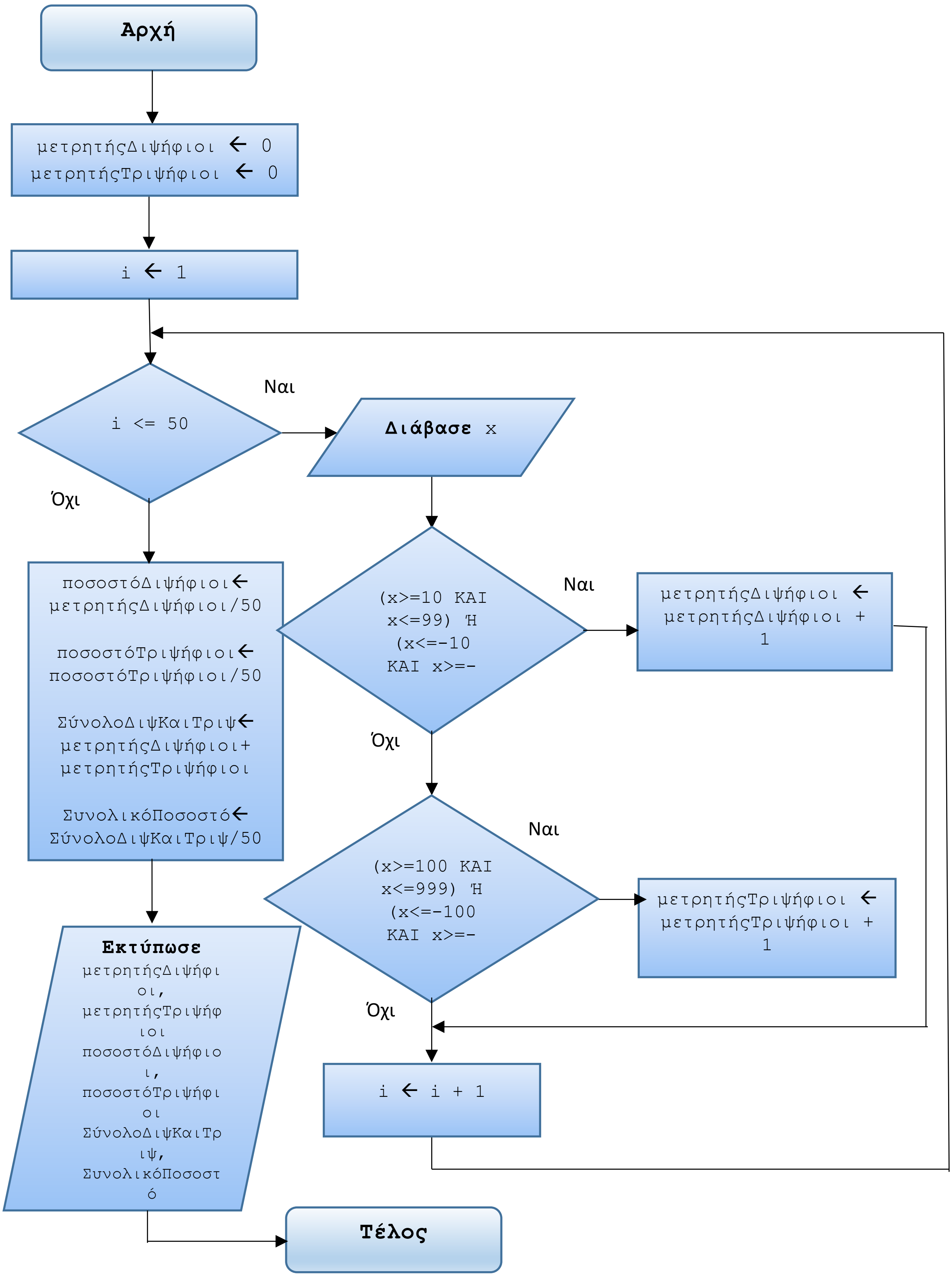
Απάντηση

Τα ερωτήματα **«πόσοι»** υποδηλώνουν ότι πρέπει να χρησιμοποιήσουμε **μετρητές**. Επίσης, η κατηγοριοποίηση σε διψήφιους και τριψήφιους μεταφράζεται ως: *ακέραιοι από το 10 έως το 99 ή -10 έως και -99 (διψήφιοι) και ακέραιοι από το 100 έως και 999 ή -100 έως και -999 (τριψήφιοι).*

```
Αλγόριθμος Μετρήσεις_ακεραίων

   !Αρχικές τιμές μετρητών.
   μετρητήςΔιψήφιοι ← 0
   μετρητήςΤριψήφιοι ← 0

   !Προκαθορισμένες (50) επαναλήψεις. Σε κάθε επανάληψη,
   !διαβάζεται ο x αριθμός και ανάλογα αυξάνουμε τον
   !αντίστοιχο μετρητή.

   !Επεξεργασία.

   Για i από 1 μέχρι 50

       Διάβασε x

       Αν (x >= 10 ΚΑΙ x =< 99) 'Η
                              (x <=- 10 ΚΑΙ x >= -99) τότε

          μετρητήςΔιψήφιοι ← μετρητήςΔιψήφιοι + 1

       αλλιώς_αν (x >= 100 ΚΑΙ x =< 999) 'Η
                            (x <=- 100 ΚΑΙ x >= -999) τότε

          μετρητήςΤριψήφιοι ← μετρητήςΤριψήφιοι + 1

       Τέλος_αν

   Τέλος_επανάληψης

   !Αφού ολοκληρωθεί η επαναληπτική είσοδος και
   !επεξεργασία των 50 αριθμών και έχουν «γεμίσει» οι
   !μετρητές, να υπολογιστεί το ποσοστό τους και το σύνολο
   ποσοστόΔιψήφιοι ← μετρητήςΔιψήφιοι / 50
   ποσοστόΤριψήφιοι ← ποσοστόΤριψήφιοι / 50

   ΣύνολοΔιψΚαιΤριψ ← μετρητήςΔιψήφιοι + μετρητήςΤριψήφιοι
   ΣυνολικόΠοσοστό ← ΣύνολοΔιψΚαιΤριψ / 50

  Τέλος_αν

  !Εκτύπωση αποτελεσμάτων.
  Εκτύπωσε μετρητήςΔιψήφιοι, μετρητήςΤριψήφιοι
  Εκτύπωσε ποσοστόΔιψήφιοι, ποσοστόΤριψήφιοι
  Εκτύπωσε ΣύνολοΔιψΚαιΤριψ, ΣυνολικόΠοσοστό
```

Τέλος Μετρήσεις_ακεραίων

Το αντίστοιχο λογικό διάγραμμα είναι το εξής:



22. Γράψτε έναν αλγόριθμο που διαβάζει το πολύ 100 ακέραιους αριθμούς και μέχρι 30 αρνητικούς. Στη συνέχεια, υπολογίζει πόσους αρνητικούς αριθμούς διάβασε και το ποσοστό τους.
Κάντε και το λογικό διάγραμμα.

Απάντηση

Κι εδώ πρόκειται για μία (μη προκαθορισμένη) επαναληπτική είσοδο δεδομένων που όμως έχει μία ιδιαιτερότητα. Η επανάληψη σταματάει σε δύο περιπτώσεις:
- Είτε όταν διαβαστούν 100 αριθμοί
- Είτε όταν διαβαστούν 30 αρνητικοί.

Για παράδειγμα, μπορεί στους 45 αριθμούς που διαβάστηκαν οι 30 να είναι αρνητικοί ή να διαβάστηκαν 100 αριθμοί και μόνο οι 5 να είναι αρνητικοί.

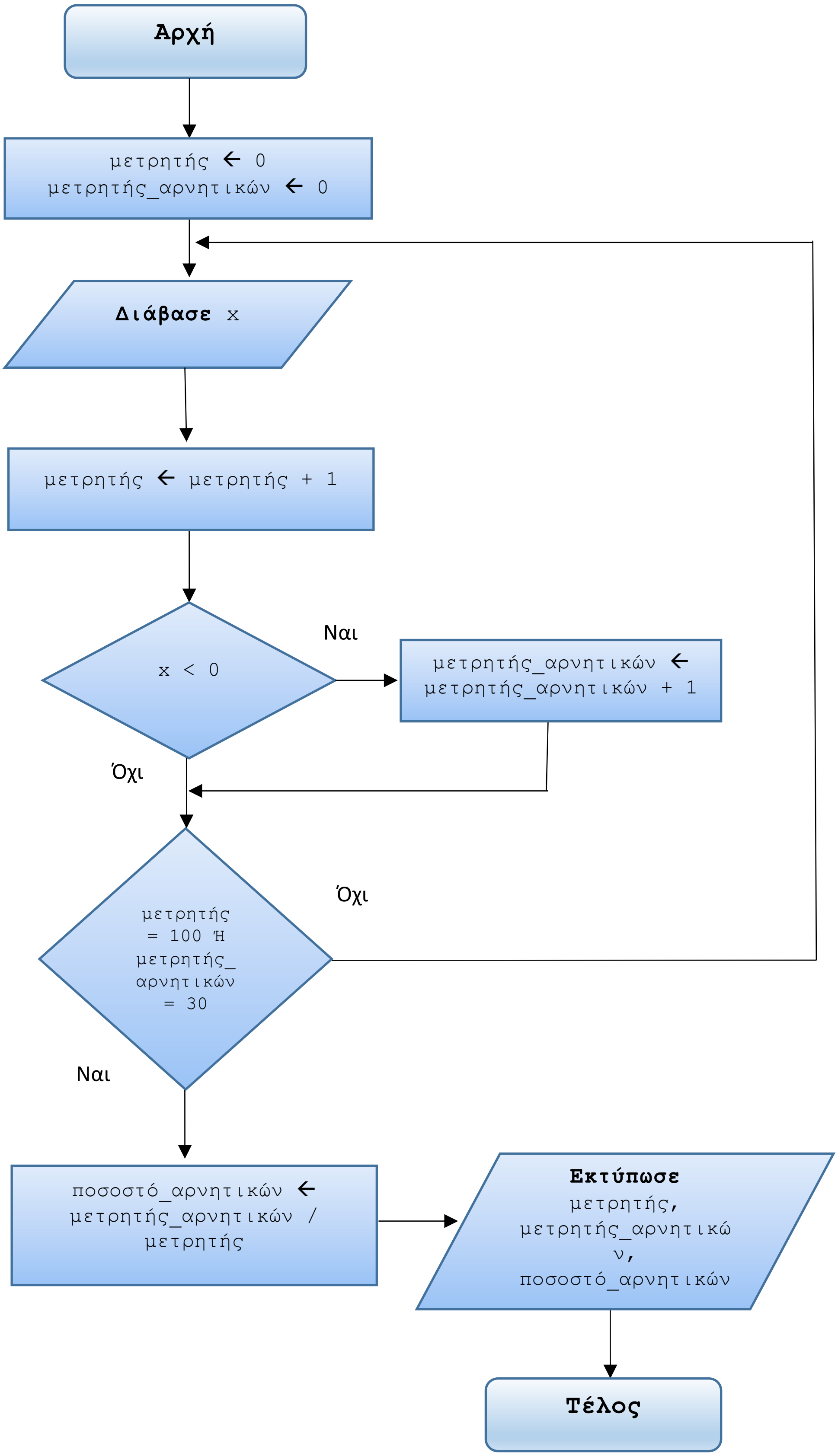
Προφανώς θα χρειαστούμε έναν γενικό μετρητή που αποθηκεύει το τρέχον πλήθος των αριθμών που διαβάστηκαν και έναν μετρητή αρνητικών. Στην περίπτωση του γενικού μετρητή είναι ταυτόχρονα και μετρητής επαναλήψεων.

Επίσης, επειδή η επανάληψη θα εκτελεστεί τουλάχιστον μία φορά μπορούμε να χρησιμοποιήσουμε τη δομή **Αρχή_επανάληψης..Μέχρις_ότου**. (*Ο αναγνώστης μπορεί να μετατρέψει τον παρακάτω αλγόριθμο σε ισοδύναμο με τη δομή* ***Όσο..επανάλαβε***)

```
Αλγόριθμος Υπολογισμός_πλήθους_αρνητικών

   !Αρχικές τιμές μετρητών.
   μετρητής ← 0
   μετρητής_αρνητικών ← 0

   !Επαναληπτική είσοδος στοιχείων.
   Αρχή_επανάληψης

       Διάβασε x
       μετρητής ← μετρητής + 1

       Αν x < 0 τότε
         μετρητής_αρνητικών ← μετρητής_αρνητικών + 1

      Τέλος_αν

   Μέχρις_ότου μετρητής = 100 Ή μετρητής_αρνητικών = 30

   !Υπολόγισε το ποσοστό των αρνητικών.
   ποσοστό_αρνητικών ← μετρητής_αρνητικών / μετρητής
```

```
    !Έξοδος αποτελεσμάτων.
    Εκτύπωσε μετρητής, μετρητής_αρνητικών,
             ποσοστό_αρνητικών

Τέλος Υπολογισμός_πλήθους_αρνητικών
```

Το αντίστοιχο λογικό διάγραμμα είναι το εξής:


Αρχή
μετρητής ← 0
μετρητής_αρνητικών ← 0
Διάβασε x
μετρητής ← μετρητής + 1
x < 0
Ναι
μετρητής_αρνητικών ← μετρητής_αρνητικών + 1
Όχι
μετρητής = 100 Ή μετρητής_αρνητικών = 30
Όχι
Ναι
ποσοστό_αρνητικών ← μετρητής_αρνητικών / μετρητής
Εκτύπωσε μετρητής, μετρητής_αρνητικών, ποσοστό_αρνητικών
Τέλος

23. Γράψτε έναν αλγόριθμο που διαβάζει 200 επώνυμα (με κεφαλαία γράμματα) και υπολογίζει πόσες φορές διαβάστηκε το επώνυμο ‘ΜΟΥΡΑΤΙΔΗΣ’’.

Απάντηση

Κι εδώ η φράση «πόσες φορές» υποδηλώνει τη χρήση ενός μετρητή. Η επαναληπτική είσοδος των επωνύμων είναι προκαθορισμένη (200 φορές), άρα η πιο κατάλληλη δομή επανάληψης είναι η **Για..από..μέχρι**

```
Αλγόριθμος Μέτρηση_επωνύμου

   !Αρχική τιμή μετρητή.
   μετρητής ← 0

   !Προκαθορισμένες (200) επαναλήψεις. Σε κάθε επανάληψη,
   !διαβάζεται ένα επώνυμο και ελέγχουμε αν είναι το
   !‘ΜΟΥΡΑΤΙΔΗΣ’. Αν ναι, τότε αυξάνουμε τον αντίστοιχο
   !μετρητή.

   !Επεξεργασία.

   Για i από 1 μέχρι 200

       Διάβασε επώνυμο

       Αν επώνυμο = ‘ΜΟΥΡΑΤΙΔΗΣ’ τότε

          μετρητής ← μετρητής + 1

       Τέλος_αν

   Τέλος_επανάληψης

  !Εκτύπωση αποτελέσματος.
  Εκτύπωσε μετρητής

Τέλος Μέτρηση_επωνύμου
```

---

24. Ένας μετεωρολόγος συνέλεξε τις θερμοκρασίες όλων των ημερών του μηνός Ιουνίου. Φτιάξτε έναν αλγόριθμο ο οποίος διαβάζει τις θερμοκρασίες του μήνα και:

α) Υπολογίζει την μέση θερμοκρασία του μήνα.
β) Υπολογίζει την μέγιστη θερμοκρασία του μήνα

Απάντηση

Ο υπολογισμός των (α) και (β) μπορεί να γίνει ταυτόχρονα στην επαναληπτική είσοδο των θερμοκρασιών.
Για τον υπολογισμό της μέσης θερμοκρασίας θα χρειαστούμε έναν αθροιστή (Sum).
Το πλήθος των δεδομένων είναι καθορισμένο (30 θερμοκρασίες του Ιουνίου).

```
Αλγόριθμος Στατιστικά_θερμοκρασιών_Ιουνίου

   !Αρχική τιμή μετρητή επαναλήψεων και αθροιστή.
   i ← 1
   Sum ← 0

   !Διάβασε την πρώτη θερμοκρασία και θέσε την ως αρχική
   !τιμή της μεταβλητής Max.
   Διάβασε θ
   Max ← θ

   !Επεξεργασία.
   Όσο i <= 30 επανάλαβε

       Sum ← Sum + θ

       Αν θ > Max τότε

          Max ← θ

       Τέλος_αν

       i ← i + 1

   Τέλος_επανάληψης

   !Υπολόγισε τη μέση θερμοκρασία.
   MO ← Sum / 30

  !Εκτύπωση αποτελεσμάτων.
  Εκτύπωσε MO, Max

Τέλος Στατιστικά_θερμοκρασιών_Ιουνίου
```

25. Σε έναν διαγωνισμό συμμετείχαν 500 υποψήφιοι. Οι βαθμολογίες των υποψηφίων συλλέχτηκαν από μία επιτροπή η οποία θέλει να τις επεξεργαστεί μέσω Η/Υ. Σας αναθέτουν να φτιάξετε έναν αλγόριθμο ο οποίος διαβάζει τους βαθμούς των υποψηφίων και :

α) Υπολογίζει το μέσο όρο (ΜΟ).

β) Πόσοι υποψήφιοι ξεπέρασαν τα 5/6 του μέσου όρου.

Σημείωση: Το (β) υποερώτημα προϋποθέτει ότι έχει υπολογιστεί πρώτα ο ΜΟ. Εφόσον δεν χρησιμοποιούμε πίνακες για την αποθήκευση των βαθμών θα πρέπει σε κάθε υποερώτημα να τους διαβάζουμε.

Απάντηση

```
Αλγόριθμος Διαγωνισμός

   !Αρχική τιμή αθροιστή.
   Sum ← 0

   !Επεξεργασία. 1η επαναληπτική είσοδος βαθμών.
   Για i από 1 μέχρι 500

       Διάβασε βαθμός

       Sum ← Sum + βαθμός

       Τέλος_αν

   Τέλος_επανάληψης

   !Υπολόγισε μέσο όρο
   ΜΟ ← Sum / 500

   !2η επαναληπτική είσοδος βαθμών. Με βάση τον ΜΟ θα
   !ελέγξουμε αν ο κάθε βαθμός που διαβάζεται είναι πάνω
   !από τα 5/6 του μέσου όρου. Αν ναι, αύξησε τον
   !αντίστοιχο μετρητή.

   !Αρχική τιμή μετρητή.
   μετρητής ← 0

   Για i από 1 μέχρι 500

       Διάβασε βαθμός

       Αν βαθμός > 5/6 * ΜΟ τότε

           μετρητής ← μετρητής + 1

       Τέλος_αν
```

```
    Τέλος_επανάληψης

    !Εκτύπωση αποτελεσμάτων.
    Εκτύπωσε ΜΟ, μετρητής

Τέλος Διαγωνισμός
```

---

26. Τί κάνει το παρακάτω κομμάτι αλγορίθμου: (X, S1, S2 είναι ακέραιες μεταβλητές)

```
S1 ← 0
S2 ← 0
X ← 0
Όσο  X <= 10 επανάλαβε
    Αν (X MOD 3) = 0 τότε
        S1 ← S1 + X
    αλλιώς
        S2 ← S2 + X
    Τέλος_αν
    X ← X + 1
Τέλος_επανάληψης
```

α) Περιγράψτε τί κάνει το παραπάνω κομμάτι.
β) Πώς διαμορφώνονται οι τιμές των X, S1, S2 σε κάθε επανάληψη; (πίνακας τιμών)
γ) Τί θα συνέβαινε αν παραλείπαμε την εντολή X ← X + 1 ;
δ) Τί θα συνέβαινε αν βάζαμε την εντολή X ← X + 1 στο τμήμα **αλλιώς** ;
ε) Μετατρέψτε την παραπάνω επαναληπτική δομή στην αντίστοιχη 2η μορφή (**Αρχή_επανάληψης..Μέχρις_ότου**) και 3η μορφή (**Για..από..μέχρι**)

Απάντηση

α) Υπάρχει μία επανάληψη 11 φορών όπου η τιμή της X ξεκινάει από το 0 μέχρι και το 10 και σε κάθε επανάληψη ελέγχει την τιμή της X: Αν διαιρείται ακριβώς με το 3 τότε η τιμή της X προστίθεται στον αθροιστή S1 διαφορετικά προστίθεται στον αθροιστή S2.

β) Κατασκευάζουμε τον παρακάτω **πίνακα τιμών**:

AA	X	X <= 10	(X MOD 3) = 0	S1	S2
1				0	
2					0
3	0				
4		ΑΛΗΘΗΣ			
5			ΨΕΥΔΗΣ		
6					(0+0)=0
7	1				
8		ΑΛΗΘΗΣ			
9			ΨΕΥΔΗΣ		
10					(0+1)=1
11	2				
12		ΑΛΗΘΗΣ			
13			ΨΕΥΔΗΣ		
14					(1+2)=3
15	3				
16		ΑΛΗΘΗΣ			
17			ΑΛΗΘΗΣ		
18				(0+1)=1	
19	4				
20		ΑΛΗΘΗΣ			
21			ΨΕΥΔΗΣ		
22					(3+4)=7

23	5				
24		ΑΛΗΘΗΣ			
25			ΨΕΥΔΗΣ		
26					(7+5)=12
27	6				
28		ΑΛΗΘΗΣ			
29			ΑΛΗΘΗΣ		
30				(1+6)=7	
31	7				
32		ΑΛΗΘΗΣ			
33			ΨΕΥΔΗΣ		
34					(12+7)=19
35	8				
36		ΑΛΗΘΗΣ			
37			ΨΕΥΔΗΣ		
38					(19+8)=27
39	9				
40		ΑΛΗΘΗΣ			
41			ΑΛΗΘΗΣ		
42				(7+9)=**16**	
43	10				
44		ΑΛΗΘΗΣ			
45			ΨΕΥΔΗΣ		
46					(27+10)=**37**

47	11				
48		ΨΕΥΔΗΣ			

γ) Εφόσον η τιμή της X δεν θα άλλαζε (θα ήταν πάντα 0) η συνθήκη της επανάληψης X <= 10 θα ήταν πάντα ΑΛΗΘΗΣ και συνεπώς θα έμπαινε σε έναν ατέρμονο βρόχο.

δ) Εδώ θα συνέβαινε το εξής περίεργο (*και φυσικά λανθασμένο από άποψη αλγοριθμικής λογικής*): Η επανάληψη θα ξεκινούσε κανονικά με τιμή της X το 0 και όσο η συνθήκη (X MOD 3) = 0 αποτιμάτο ΨΕΥΔΗΣ η τιμή της X θα άλλαζε κατά 1. Όταν όμως, θα έφτανε στην τιμή 3 τότε η συνθήκη (X MOD 3) = 0 θα αποτιμάτο ΑΛΗΘΗΣ, δεν θα άλλαζε η τιμή της X και θα παρέμενε μόνιμα στην τιμή 3. Έτσι, πάλι θα καταλήγαμε σε ατέρμονο βρόχο. Με άλλα λόγια, μετά από 4 επαναλήψεις η τιμή της X θα σταμάταγε να αλλάζει!

ε)

i) Μετατροπή στη δομή **Αρχή_επανάληψης..Μέχρις_ότου** :

```
S1 ← 0
S2 ← 0
X ← 0

Αρχή_επανάληψης

   Αν (X MOD 3) = 0 τότε

      S1 ← S1 + X

   αλλιώς

      S2 ← S2 + X

   Τέλος_αν

   X ← X + 1

Μέχρις_ότου X > 10
```

ii) Μετατροπή στη δομή **Για..από..μέχρι** :

```
S1 ← 0
S2 ← 0

Για X από 0 μέχρι 10

   Αν (X MOD 3) = 0 τότε

      S1 ← S1 + X
```

```
        αλλιώς
            S2 ← S2 + X
        Τέλος_αν

    Τέλος_επανάληψης
```

---

27. Δίνεται το παρακάτω κομμάτι αλγορίθμου:

```
X ← 0
Όσο X <= 10 επανάλαβε
    Y ← 2 * X + 5
    Αν (Y MOD 3) = 0 τότε
        Εμφάνισε Y
    Τέλος_αν
    X ← X + 2
Τέλος_επανάληψης
```

α) Περιγράψτε τι κάνει ο παραπάνω αλγόριθμος.
β) Δώστε αναλυτικά τον πίνακα των τιμών.
γ) Αν παραλείψουμε την εντολή `X ← X + 2` τί θα συμβεί;
δ) Μετατρέψτε το παραπάνω απόσπασμα ώστε να χρησιμοποιεί την επαναληπτική δομή **ι)** `Για..από..μέχρι`
ιι) `Αρχή_επανάληψης..Μέχρις_ότου`
ε) Αναφέρατε συνοπτικά ποιές οι διαφορές μεταξύ των 3 επαναληπτικών δομών των αλγορίθμων και σε ποιές περιπτώσεις είναι κατάλληλη η καθεμιά.
στ) Μετατρέψτε το παραπάνω κομμάτι του αλγορίθμου από ψευδογλώσσα σε λογικό διάγραμμα.

Απάντηση

α) Ξεκινάει μία επανάληψη με αρχική τιμή της X το 0 και τελική το 10 όπου κάθε φορά υπολογίζει την παράσταση 2x+5 αποθηκεύει την αποτίμησή της στη μεταβλητή Y και κατόπιν ελέγχει αν η τιμή της Y διαιρείται ακριβώς με το 3. Αν ναι, τότε εμφανίζει την τιμή της Y. Η τιμή της X αυξάνεται κατά 2 σε κάθε επανάληψη.

β) Κατασκευάζουμε τον παρακάτω **πίνακα τιμών**:

ΑΑ	X	X <= 10	Y	(Y MOD 3)=0	Οθόνη
1	0				
2		ΑΛΗΘΗΣ			
3			2*0+5=5		
4				ΨΕΥΔΗΣ	
5	2				
6		ΑΛΗΘΗΣ			
7			2*2+5=9		
8				ΑΛΗΘΗΣ	
9	4				9
10		ΑΛΗΘΗΣ			
11			2*4+5=13		
12				ΨΕΥΔΗΣ	
13	6				
14		ΑΛΗΘΗΣ			
15			2*6+5=17		
16				ΨΕΥΔΗΣ	
17	8				
18		ΑΛΗΘΗΣ			
19			2*8+5=21		
20				ΑΛΗΘΗΣ	
21					21
22	10				

23		ΑΛΗΘΗΣ			
24			2*10+5=25		
25				ΨΕΥΔΗΣ	
26	12				
27		ΨΕΥΔΗΣ			

γ) Εφόσον η τιμή της X δεν θα άλλαζε (θα ήταν πάντα 0) η συνθήκη της επανάληψης X <= 10 θα ήταν πάντα ΑΛΗΘΗΣ και συνεπώς θα έμπαινε σε έναν ατέρμονο βρόχο.

δ)

i) Μετατροπή στη δομή **Για..από..μέχρι** :

```
Για X από 0 μέχρι 10 με_βήμα 2
   Y ← 2 * X + 5
   Αν (Y MOD 3) = 0 τότε
      Εμφάνισε Y
   Τέλος_αν
Τέλος_επανάληψης
```

ii) Μετατροπή στη δομή **Αρχή_επανάληψης..Μέχρις_ότου** :

```
X ← 0
Αρχή_επανάληψης
   Y ← 2 * X + 5
   Αν (Y MOD 3) = 0 τότε
      Εμφάνισε Y
   Τέλος_αν
   X ← X + 2
Μέχρις_ότου X > 10
```

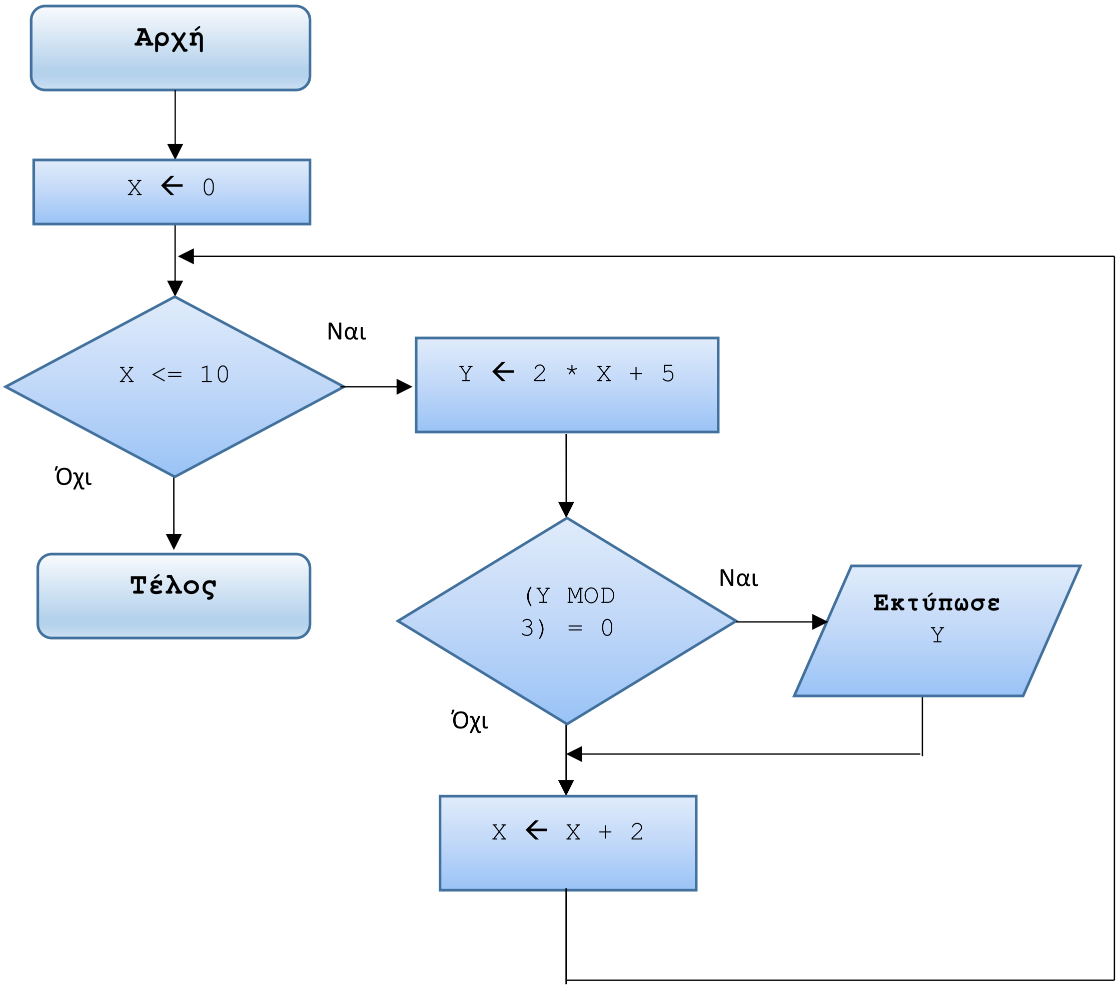
δ)

- Η δομή `Όσο..επανάλαβε` είναι μία γενική δομή επανάληψης και μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση προβλήματος που απαιτεί επαναληπτική διαδικασία (είτε γνωρίζουμε τον αριθμό των επαναλήψεων είτε όχι). Χαρακτηριστική περίπτωση η επαναληπτική είσοδος αγνώστου πλήθους δεδομένων. Η επανάληψη μπορεί να μην εκτελεστεί ούτε μία φορά αν η συνθήκη ελέγχου είναι εξ' αρχής ΨΕΥΔΗΣ.

- Η δομή `Αρχή_επανάληψης..Μέχρις_ότου` μπορεί να χρησιμοποιεί όπως και η `Όσο..επανάλαβε` με τη διαφορά ότι εκτελείται τουλάχιστον μία φορά. Συνήθως χρησιμοποιείται για να ελέγξουμε την εγκυρότητα κάποιου δεδομένου εισόδου π.χ. αν ο βαθμός που εισάγεται είναι μεταξύ 1 και 20.

- Η δομή `Για..από..μέχρι` είναι αρκετά συνήθης και χρησιμοποιείται ευρέως όταν γνωρίζουμε τον αριθμό των επαναλήψεων π.χ. η επαναληπτική είσοδος γνωστού πλήθους δεδομένων (να διαβαστούν 50 αριθμοί κλπ.). Επίσης, χρησιμοποιείται πολύ σε σχέση με δομές δεδομένων (π.χ. πίνακες, στοίβες, ουρές, λίστες κλπ).

ε) Το αντίστοιχο λογικό διάγραμμα είναι το εξής:



28. Ένα μικρό αμφιθέατρο διαθέτει 30 καθίσματα στην πρώτη σειρά και σε κάθε επόμενη, από τις συνολικά 15, προστίθενται 5 καθίσματα. Η κατασκευή των καθισμάτων έχει κάποιο κόστος. Μέχρι και την δέκατη σειρά, το κάθε κάθισμα κοστίζει 7€ και για τις άλλες σειρές 9€. Να γραφτεί αλγόριθμος που υπολογίζει και εμφανίζει:

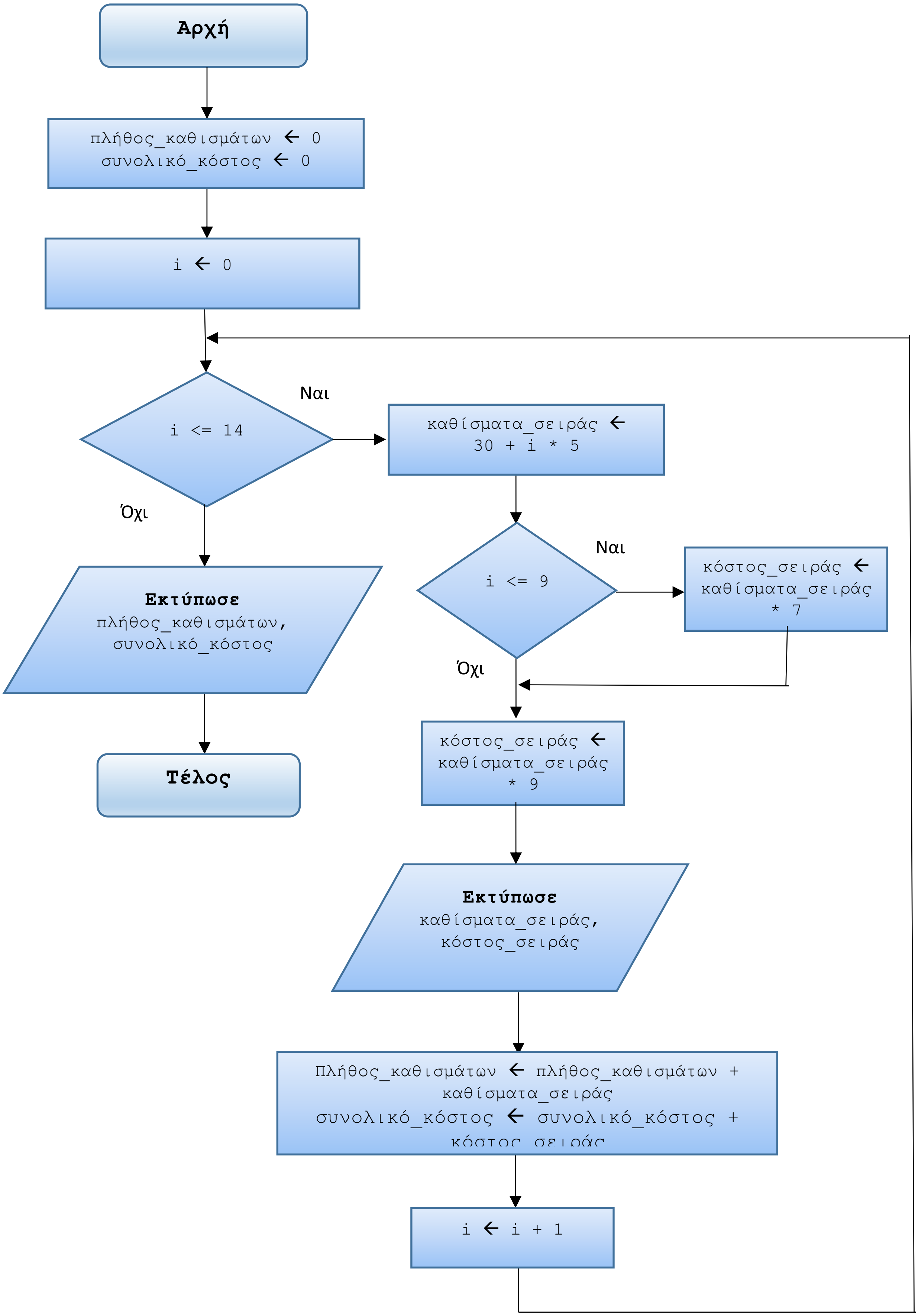
α) Τα καθίσματα της κάθε σειράς και το κόστος τους.
β) Το συνολικό πλήθος και κόστος των καθισμάτων.
γ) Κάντε και το λογικό διάγραμμα.

Απάντηση

Η 1η σειρά έχει 30 καθίσματα, η $2^{η}$ σειρά έχει 35 καθίσματα, η $3^{η}$ σειρά έχει 40 καθίσματα κ.ο.κ. Άρα, γενικά, η `i` σειρά έχει `30+(i-1)*5` καθίσματα αν το `1<=i<=15` ή `30+(i*5)` καθίσματα αν το `0<=i<=14`. Παρακάτω, επιλέγουμε τον δεύτερο τρόπο σκέψης.

```
Αλγόριθμος Αμφιθέατρο

   !Αρχικές τιμές μετρητών και αθροιστών.
   πλήθος_καθισμάτων ← 0
   συνολικό_κόστος ← 0

   !Επεξεργασία. 15 σειρές καθισμάτων.
   !Ο μετρητής i υποδηλώνει τη σειρά. Για ευκολία
   !υπολογισμών ξεκινάει από το 0 (πρώτη σειρά) μέχρι και
   !το 14 (15η σειρά).
   Για i από 0 μέχρι 14

       καθίσματα_σειράς ← 30 + (i * 5)

      !Υπολόγισε το κόστος των καθισμάτων ανάλογα με τη
      !σειρά που βρίσκονται.
       Αν i <= 9 τότε

         κόστος_σειράς ← καθίσματα_σειράς * 7

       αλλιώς

         κόστος_σειράς ← καθίσματα_σειράς * 9

       Τέλος_αν

       !Έξοδος αποτελεσμάτων σειράς.
       Εκτύπωσε καθίσματα_σειράς, κόστος_σειράς

       !Πρόσθεσε τα καθίσματα στο πλήθος.
       πλήθος_καθισμάτων  ← πλήθος_καθισμάτων +
                                          καθίσματα_σειράς

       !Πρόσθεσε το κόστος της σειράς στο συνολικό κόστος.
       συνολικό_κόστος ← συνολικό_κόστος + κόστος_σειράς

   Τέλος_επανάληψης

   !Έξοδος συνολικών αποτελεσμάτων.
   Εκτύπωσε πλήθος_καθισμάτων, συνολικό_κόστος

Τέλος Αμφιθέατρο
```

Το αντίστοιχο λογικό διάγραμμα είναι το εξής:


Αρχή
πλήθος_καθισμάτων ← 0
συνολικό_κόστος ← 0
i ← 0
i <= 14
Ναι
Όχι
καθίσματα_σειράς ← 30 + i * 5
i <= 9
Ναι
Όχι
κόστος_σειράς ← καθίσματα_σειράς * 7
κόστος_σειράς ← καθίσματα_σειράς * 9
Εκτύπωσε καθίσματα_σειράς, κόστος_σειράς
Πλήθος_καθισμάτων ← πλήθος_καθισμάτων + καθίσματα_σειράς
συνολικό_κόστος ← συνολικό_κόστος + κόστος_σειράς
i ← i + 1
Εκτύπωσε πλήθος_καθισμάτων, συνολικό_κόστος
Τέλος

29. Να γράψετε έναν αλγόριθμο που διαβάζει τα ονοματεπώνυμα και τους βαθμούς των 25 μαθητών του τμήματος Γ1 στο μάθημα «Ανάπτυξη εφαρμογών» και:

α) Υπολογίζει τον μέσο όρο τους.

β) Υπολογίζει το πλήθος και το ποσοστό των μαθητών στις εξής κλίμακες: 0-10, 10.1-15 και 15.1-20.

Κάντε και το λογικό διάγραμμα.

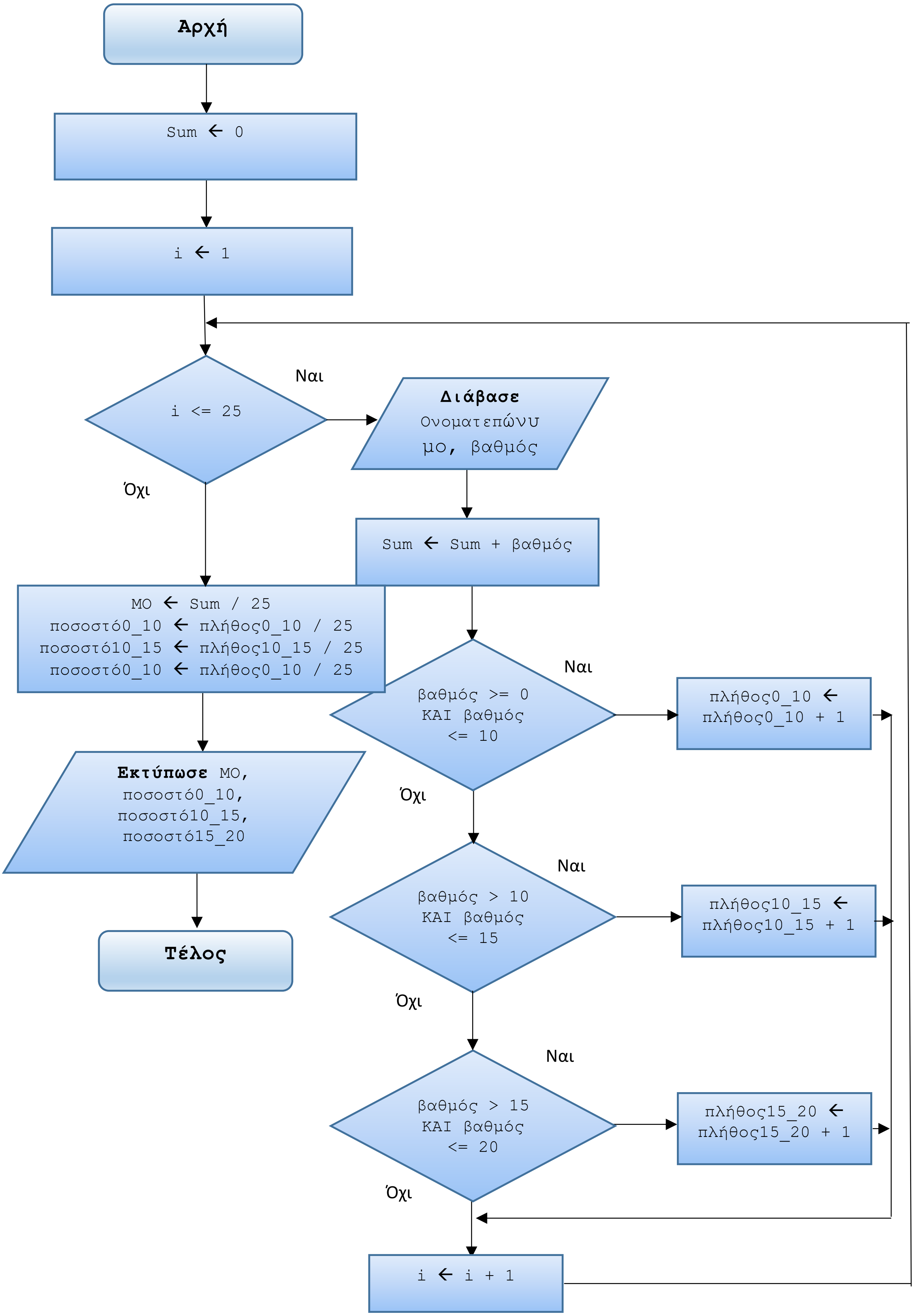
Απάντηση

Η επεξεργασία θα γίνει επί ενός συγκεκριμένου συνόλου δεδομένων (στοιχεία 15 μαθητών). Συνεπώς, η καταλληλότερη δομή επανάληψης που θα χρησιμοποιήσουμε είναι η **Για..από..μέχρι**

```
Αλγόριθμος Στατιστικά_μαθητών

   !Αρχικές τιμές αθροιστών και μετρητών.
   Sum ← 0
   πλήθος0_10 ← 0
   πλήθος10_15 ← 0
   πλήθος15_20 ← 0

   !Επεξεργασία.
   Για i από 1 μέχρι 25

       Διάβασε Ονοματεπώνυμο, βαθμός

       Sum ← Sum + βαθμός

       !Έλεγξε σε ποια κλίμακα είναι ο βαθμός και ανάλογα
       !πρόσθεσε τον αντίστοιχο μετρητή της κλίμακας.
       Επίλεξε βαθμός

         Περίπτωση βαθμός >= 0 ΚΑΙ βαθμός <= 10
                πλήθος0_10 ← πλήθος0_10 + 1

         Περίπτωση βαθμός > 10 ΚΑΙ βαθμός <= 15
                πλήθος10_15 ← πλήθος10_15 + 1

         Περίπτωση βαθμός > 15 ΚΑΙ βαθμός <= 20
                πλήθος15_20 ← πλήθος15_20 + 1

       Τέλος_επιλογών

    Τέλος_επανάληψης

    !Υπολόγισε μέσο όρο.
```

```
      ΜΟ ← Sum / 25

      !Υπολόγισε ποσοστά.
      ποσοστό0_10 ← πλήθος0_10 / 25
      ποσοστό10_15 ← πλήθος10_15 / 25
      ποσοστό15_20 ← πλήθος15_20 / 25

      !Έξοδος αποτελεσμάτων.
     Εκτύπωσε ΜΟ, ποσοστό0_10, ποσοστό10_15, ποσοστό15_20

   Τέλος Στατιστικά_μαθητών
```

Το αντίστοιχο λογικό διάγραμμα είναι το εξής:


Αρχή
Sum ← 0
i ← 1
i <= 25
Ναι
Όχι
Διάβασε Ονοματεπώνυ μο, βαθμός
Sum ← Sum + βαθμός
MO ← Sum / 25
ποσοστό0_10 ← πλήθος0_10 / 25
ποσοστό10_15 ← πλήθος10_15 / 25
ποσοστό0_10 ← πλήθος0_10 / 25
βαθμός >= 0 ΚΑΙ βαθμός <= 10
Ναι
πλήθος0_10 ← πλήθος0_10 + 1
Όχι
Εκτύπωσε MO, ποσοστό0_10, ποσοστό10_15, ποσοστό15_20
Τέλος
βαθμός > 10 ΚΑΙ βαθμός <= 15
Ναι
πλήθος10_15 ← πλήθος10_15 + 1
Όχι
βαθμός > 15 ΚΑΙ βαθμός <= 20
Ναι
πλήθος15_20 ← πλήθος15_20 + 1
Όχι
i ← i + 1

30. Να γράψετε έναν αλγόριθμο που διαβάζει τα ονοματεπώνυμα και τους βαθμούς των 75 μαθητών της Γ' τάξης και:

α) Εκτυπώνει το ονοματεπώνυμό του μαζί με τον χαρακτηρισμό του βαθμού σύμφωνα με τον εξής πίνακα:

Βαθμός	Χαρακτηρισμός
Μέχρι 9.9	Κακώς
10-12.4	Μέτρια
12.5-15.4	Καλά
15.5-18.4	Πολύ καλά
18.5-20	Άριστα

β) Αν ο μαθητής αρίστευσε να εμφανίζεται επιπλέον κι ένας αστερίσκος ("*")
γ) Να υπολογίζει πόσοι μαθητές αρίστευσαν.
δ) Ο αλγόριθμός κατά το διάβασμα ενός βαθμού να ελέγχει αν είναι έγκυρος (δηλαδή μεταξύ 1 και 20).

Απάντηση

```
Αλγόριθμος Χαρακτηρισμός_βαθμού

!Αρχική τιμή μετρητή.
Αρίστευσαν ← 0

!Επεξεργασία.
Για i από 1 μέχρι 75

    Διάβασε Ονοματεπώνυμο

    !Έλεγξε τον βαθμό αν είναι έγκυρος.
    !Για τέτοιες περιπτώσεις κατάλληλη είναι η δομή
    !Αρχή_επανάληψης..Μέχρις_ότου. Δηλαδή, θα ζητείται
    !συνεχώς ο βαθμός μέχρι να δοθεί έγκυρος.
    Αρχή_επανάληψης

        Διάβασε βαθμός

        Αν ΌΧΙ(βαθμός >= 1 ΚΑΙ βαθμός <= 20) τότε

            Εκτύπωσε 'Ο βαθμός δεν είναι έγκυρος. Παρακαλώ,
                                                ξαναδώστε'
        Τέλος_αν

    Μέχρις_ότου βαθμός >= 1 ΚΑΙ βαθμός <= 20
```

```
        !Αφού ο βαθμός είναι έγκυρος προχώρα κι έλεγξε σε ποια
        !κλίμακα είναι ο βαθμός και ανάλογα
        !εκτύπωσε το ονοματεπώνυμο και τον χαρακτηρισμό.
        Επίλεξε βαθμός

          Περίπτωση βαθμός <= 9.9
                 Εκτύπωσε Ονοματεπώνυμο, 'Κακώς'

          Περίπτωση βαθμός <= 12.4
                 Εκτύπωσε Ονοματεπώνυμο, 'Μέτρια'

          Περίπτωση βαθμός <= 15.4
                 Εκτύπωσε Ονοματεπώνυμο, 'Καλά'

          Περίπτωση βαθμός <= 18.4
                 Εκτύπωσε Ονοματεπώνυμο, 'Πολύ καλά'

          Περίπτωση αλλιώς !Είναι σίγουρα μεταξύ 18.5 και 20
                 Αρίστευσαν ← Αρίστευσαν + 1
                 Εκτύπωσε '*', Ονοματεπώνυμο, 'Άριστα'

        Τέλος_επιλογών

     Τέλος_επανάληψης

     !Έξοδος αποτελεσμάτων. Εδώ πόσοι αρίστευσαν.
    Εκτύπωσε Αρίστευσαν

Τέλος Χαρακτηρισμός_βαθμού
```

---

31. Δίνεται το παρακάτω κομμάτι αλγορίθμου:

```
Sum ← 0

Για X από 1 μέχρι 10

    Sum ← Sum + X
    X ← X + 1

Τέλος_επανάληψης
```

α) Ποιές είναι διαδοχικά οι τιμές των X και SUM σε κάθε επανάληψη;
β) Υπάρχει λόγος ύπαρξης της εντολής X ← X + 1 στο εσωτερικό της επανάληψης; Αν επιθυμούμε αύξηση της X κατά 2 πώς πρέπει να διαμορφώσουμε την εντολή επανάληψης;

Απάντηση

α) Ουσιαστικά ζητείται **πίνακας τιμών**:

AA	X	X <= 10	Sum
1			0
2	1		
3		ΑΛΗΘΗΣ	
4			0+1=1
5	1+1=2		
6	2+1=3		
7		ΑΛΗΘΗΣ	
8			1+3=5
9	3+1=4		
10	4+1=5		
11		ΑΛΗΘΗΣ	
12			5+5=10
13	5+1=6		
14	6+1=7		
15		ΑΛΗΘΗΣ	
16			10+7=17
17	7+1=8		
18	8+1=9		
19		ΑΛΗΘΗΣ	
20			17+9=**26**
21	9+1=10		
22	10+1=11		

23		ΨΕΥΔΗΣ	

β) Αν θέλουμε σε κάθε επανάληψη η `X` να αυξάνεται κατά `1` προφανώς η ύπαρξη της εντολής `X ← X + 1` είναι λάθος διότι προκαλεί αύξηση κατά `2`. Γενικά, πρέπει να αποφεύγουμε να αλλάζουμε την τιμή του μετρητή (εδώ της `X`) στο εσωτερικό της επανάληψης διότι αυτό γίνεται αυτόματα από την εντολή επανάληψης **`Για..από..μέχρι.`**

Έτσι, αν θέλαμε αύξηση του μετρητή κατά `2` θα γράφαμε την εντολή επανάληψης **`Για..από..μέχρι`** ως εξής:

```
Για X από 1 μέχρι 10 με_βήμα 2
    Sum ← Sum + X
Τέλος_επανάληψης
```

Διαδοχικά οι τιμές της `X` θα είναι `1, 3, 5, 7, 9.`

---

32. Δίνεται το παρακάτω κομμάτι αλγορίθμου:

```
Sum ← 0
Για i από 1 μέχρι 5
    Για j από 1 μέχρι 3
        Sum ← Sum + i * j
    Τέλος_επανάληψης
Τέλος_επανάληψης
```

α) Πόσες φορές θα εκτελεστεί ο εξωτερικός βρόχος (με το `i`) και ο εσωτερικός βρόχος (με το `j`);
β) Κάντε τον πίνακα τιμών για τις μεταβλητές `Sum, i` και `j`.
γ) Κάντε το λογικό διάγραμμα.

Απάντηση

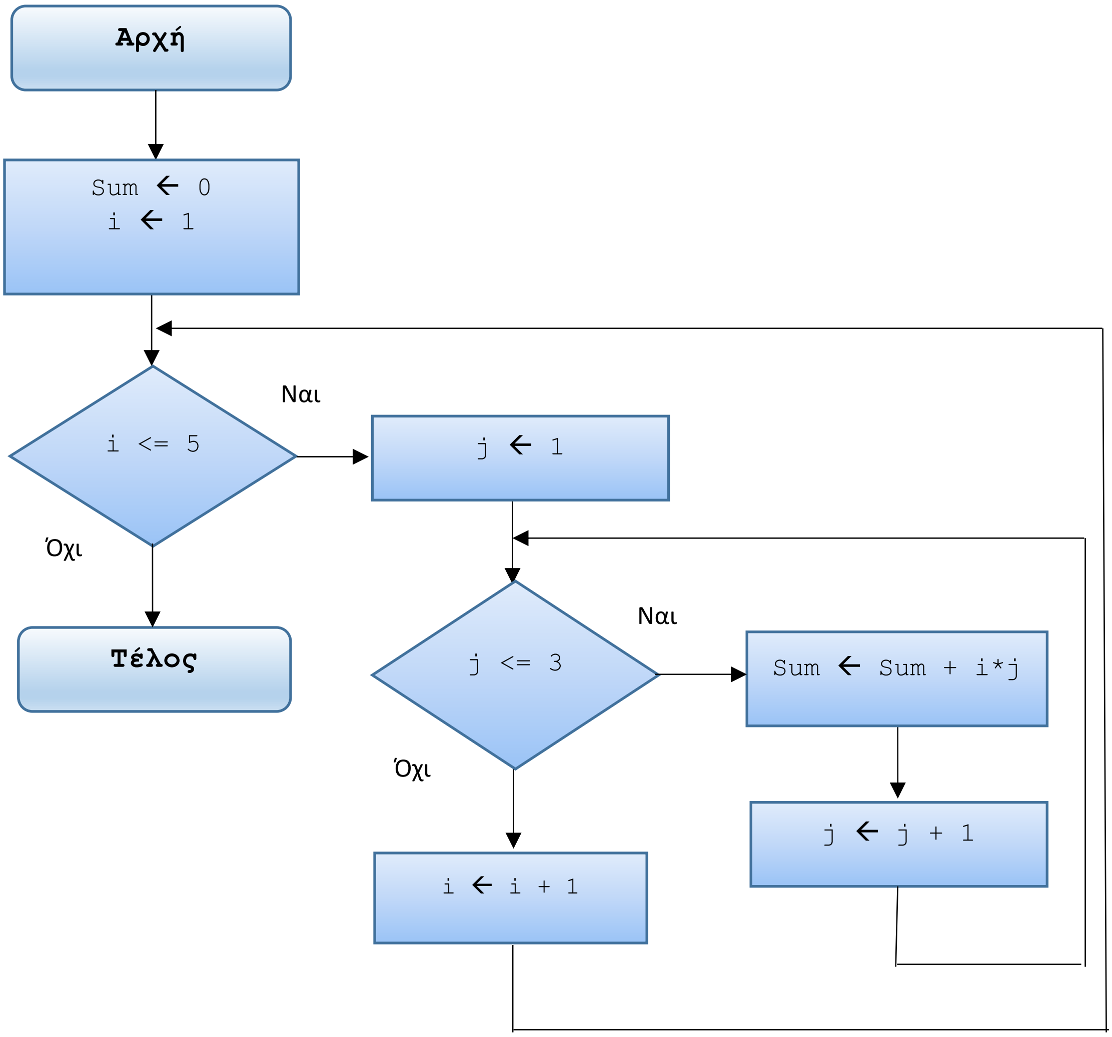
α) Ο εξωτερικός βρόχος (με το `i`) θα εκτελεστεί 5 φορές και ο εσωτερικός βρόχος (με το `j`) 5x3 = 15 φορές.

β) Ο **πίνακας των τιμών** είναι:

AA	i	j	Sum
1			0
2	1		
3		1	
4			0+1*1=1
5		2	
6			1+1*2=3
7		3	
8			3+1*3=6
9	2		
10		1	
11			6+2*1=8
12		2	
13			8+2*2=12
14		3	
15			12+2*3=18
16	3		
17		1	
18			18+3*1=21
19		2	
20			21+3*2=27
21		3	
22			27+3*3=36

23	4		
24		1	
25			36+4*1=40
26		2	
27			40+4*2=48
28		3	
29			48+4*3=60
30	5		
31		1	
32			60+5*1=65
33		2	
34			65+5*2=75
35		3	
36			75+5*3=**90**

γ) Το αντίστοιχο λογικό διάγραμμα είναι το εξής:



33. Δίνεται το παρακάτω κομμάτι αλγορίθμου:

```
Για i από 1 μέχρι 5

    Sum ← 0
    Για j από 1 μέχρι 3

        Διάβασε Ποσό
        Sum ← Sum + Ποσό

    Τέλος_επανάληψης

    MO ← Sum / 3
    Εμφάνισε MO

Τέλος_επανάληψης
```

α) Περιγράψτε τί κάνει ο παραπάνω αλγόριθμος;
β) Πόσες φορές θα εκτελεστεί η εντολή **`Διάβασε`**;
γ) Πόσες φορές θα εκτελεστεί η εντολή **`Εμφάνισε;`**.
δ) Κάντε το λογικό διάγραμμα.
ε) Μετατρέψτε τις εντολές **`Για..από..μέχρι`** σε αντίστοιχες **`Όσο...επανάλαβε`**

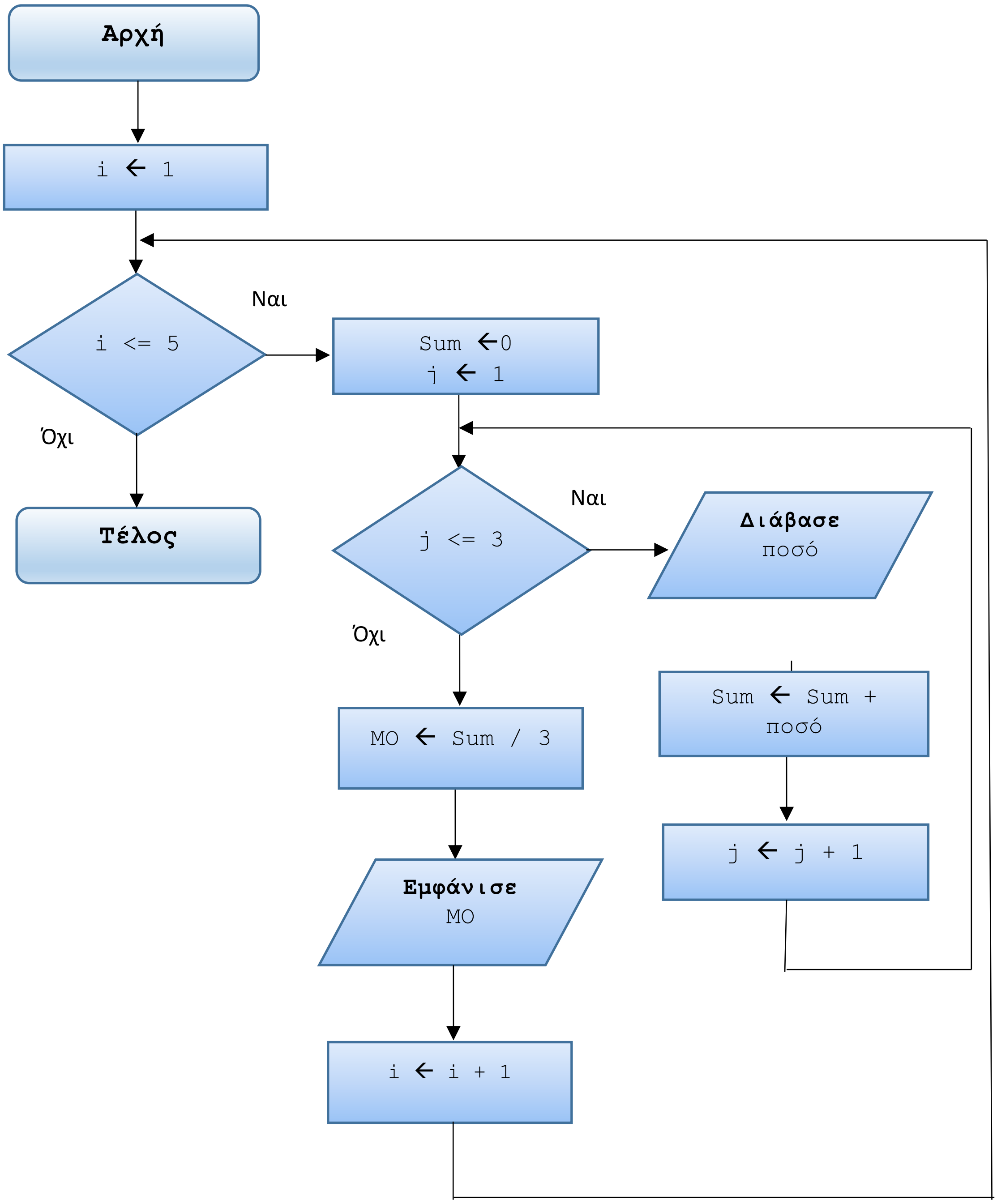
Απάντηση

α) Υπολογίζει 5 διαφορετικούς μέσους όρους (`MO`) ποσών και τους τυπώνει αντίστοιχα στην οθόνη. Ο κάθε `MO` προκύπτει από το διάβασμα 3 διαφορετικών ποσών.
Προσέχουμε ότι η αρχικοποίηση της `Sum` (`Sum`←`0`) γίνεται πριν ξεκινήσει η εσωτερική επανάληψη διότι υπολογίζουμε διαφορετικό άθροισμα σε κάθε επανάληψη.

β) 15 φορές (5 εξωτερικές επαναλήψεις με το `i` και για κάθε εξωτερική επανάληψη ξεκινάει μία εμφωλευμένη (εσωτερική) του `j` που εκτελείται 3 φορές). Η εντολή **`Διάβασε`** βρίσκεται μέσα στην εσωτερική επανάληψη του `j` και εκτελείται πάντα χωρίς περιορισμούς (π.χ. αν υπήρχε μία εντολή **`Αν..τότε`**).

γ) 5 φορές, όσες φορές δηλαδή εκτελείται η εξωτερική επανάληψη του `i` διότι η εντολή **`Εμφάνισε`** βρίσκεται μέσα στο μπλοκ εντολών αυτής.

δ) Το αντίστοιχο λογικό διάγραμμα είναι το εξής:



ε) Μετατροπή των εντολών **Για..από..μέχρι** σε **Όσο..επανάλαβε**:

```
i ← 1

Όσο i <= 5 επανάλαβε

    Sum ← 0
    j ← 1
```

```
        Όσο j <= 3 επανάλαβε

            Διάβασε Ποσό
            Sum ← Sum + Ποσό
            j ← j + 1

        Τέλος_επανάληψης

        ΜΟ ← Sum / 3
        Εμφάνισε ΜΟ

        i ← i + 1

    Τέλος_επανάληψης
```

---

34. Δίνεται το παρακάτω κομμάτι αλγορίθμου:

```
Sum ← 0
Για i από 1 μέχρι 5

    Sum_μερικό ← 0
    Διάβασε Χ
    Όσο (Χ MOD 3 = 0) ΚΑΙ (Χ > 0) επανάλαβε

       Sum_μερικό ← Sum_μερικό + Χ
       Διάβασε Χ

    Τέλος_επανάληψης

    Εμφάνισε Sum_μερικό
    Sum ← Sum + Sum_μερικό

Τέλος_επανάληψης

Εμφάνισε Sum
```

α) Περιγράψτε τί κάνει ο παραπάνω αλγόριθμος;
β) Ο εσωτερικός βρόχος πότε τερματίζεται;
γ) Κάντε τον πίνακα τιμών με τα εξής δοκιμαστικά δεδομένα (ΔΔ) του Χ:
    3, 3, 4, 7, 9, -3, 12, 0, 15, -5
δ) Κάντε το λογικό διάγραμμα.

Απάντηση

α) Ο αλγόριθμος διαβάζει συνεχόμενα ακεραίους (υποθέτουμε ότι δίνουμε ακεραίους) και τους προσθέτει στον μερικό αθροιστή `Sum_μερικό`, εφόσον είναι θετικοί που διαιρούνται ακριβώς με το 3. Το μερικό άθροισμα εμφανίζεται στην οθόνη και κατόπιν προστίθεται σε έναν γενικό αθροιστή `Sum`.

Η διαδικασία του διαβάσματος ακεραίων αριθμών και ο υπολογισμός του μερικού αθροίσματος γίνεται 5 φορές (άρα θα προκύψουν 5 μερικά αθροίσματα από κάθε ομάδα ακεραίων που διαβάζεται σε κάθε επανάληψη της i).
Στο τέλος, το γενικό άθροισμα εμφανίζεται στην οθόνη.

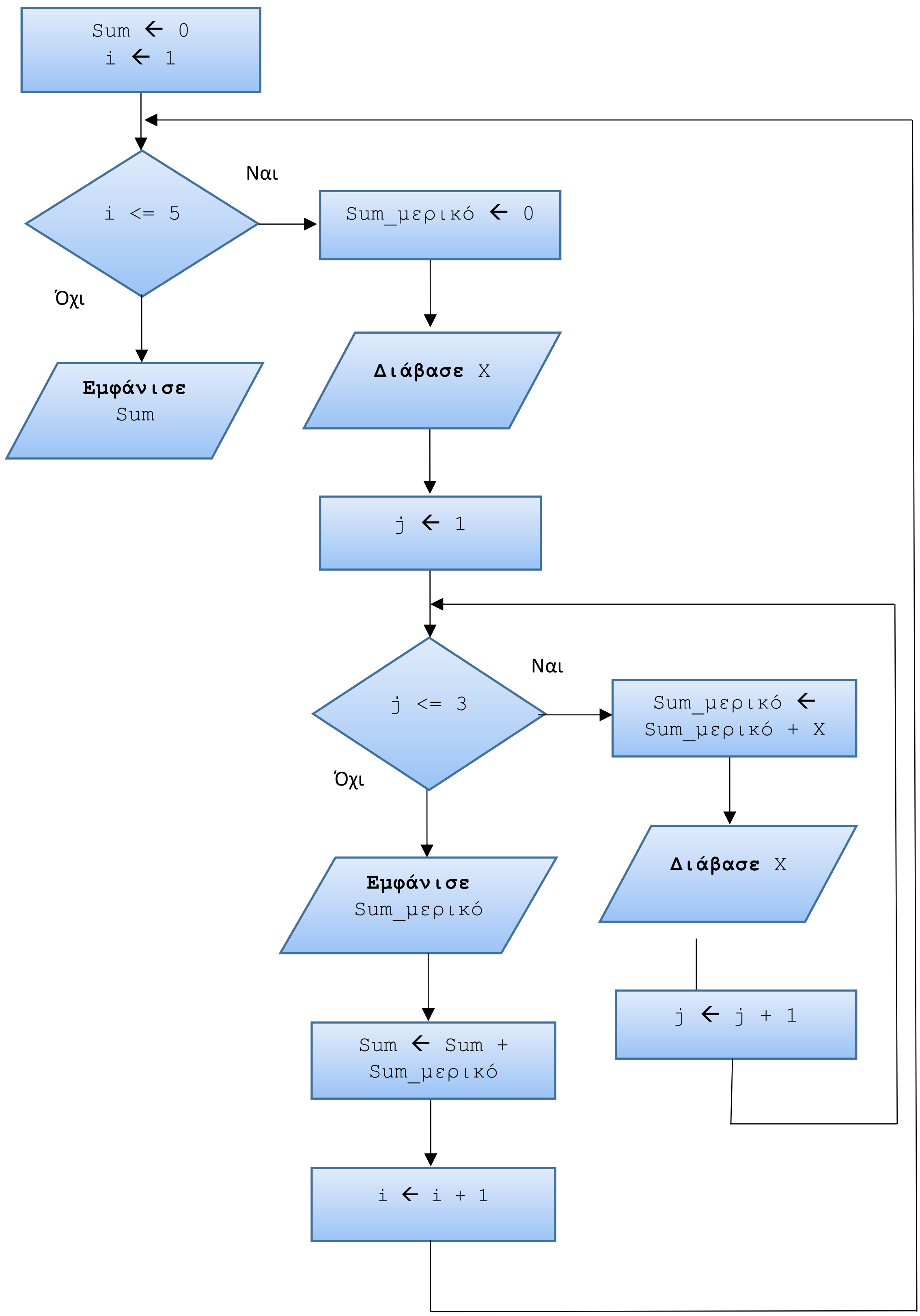
β) Ο εσωτερικός βρόχος (**`Όσο επανάλαβε`**) τερματίζεται όταν δοθεί θετικός αριθμός που διαιρείται ακριβώς με το 3.

γ) Ο πίνακας των τιμών είναι ο εξής:

AA	i	X	(X MOD 3 = 0) ΚΑΙ (X > 0	Sum_μερικό	Sum
1					0
2	1				
3				0	
4		3			
5			ΑΛΗΘΗΣ		
6				0+3=3	
7		3			
8			ΑΛΗΘΗΣ		
9				3+3=6	
10		1			
11				6+2*1=8	
12		2			
13				8+2*2=12	
14		3			
15				12+2*3=18	
16	3				
17		1			
18				18+3*1=21	
19		2			
20				21+3*2=27	
21		3			
22				27+3*3=36	

23	4				
24		1			
25				36+4*1=40	
26		2			
27				40+4*2=48	
28		3			
29				48+4*3=60	
30	5				
31		1			
32				60+5*1=65	
33		2			
34				65+5*2=75	
35		3			
36				75+5*3=**90**	

γ) Το λογικό διάγραμμα είναι το εξής:


Sum ← 0
i ← 1
i <= 5
Ναι
Όχι
Sum_μερικό ← 0
Διάβασε X
j ← 1
Εμφάνισε Sum
j <= 3
Ναι
Όχι
Sum_μερικό ← Sum_μερικό + X
Διάβασε X
j ← j + 1
Εμφάνισε Sum_μερικό
Sum ← Sum + Sum_μερικό
i ← i + 1

ΤΕΛΟΣ ΛΥΣΕΩΝ ΑΣΚΗΣΕΩΝ
στη δομή Επανάληψης
ΚΕΦΑΛΑΙΟΥ 2