

ΕΠΛ 002:
ΕΙΣΑΓΩΓΗ ΣΤΗΝ
ΕΠΙΣΤΗΜΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Προγραμματισμός

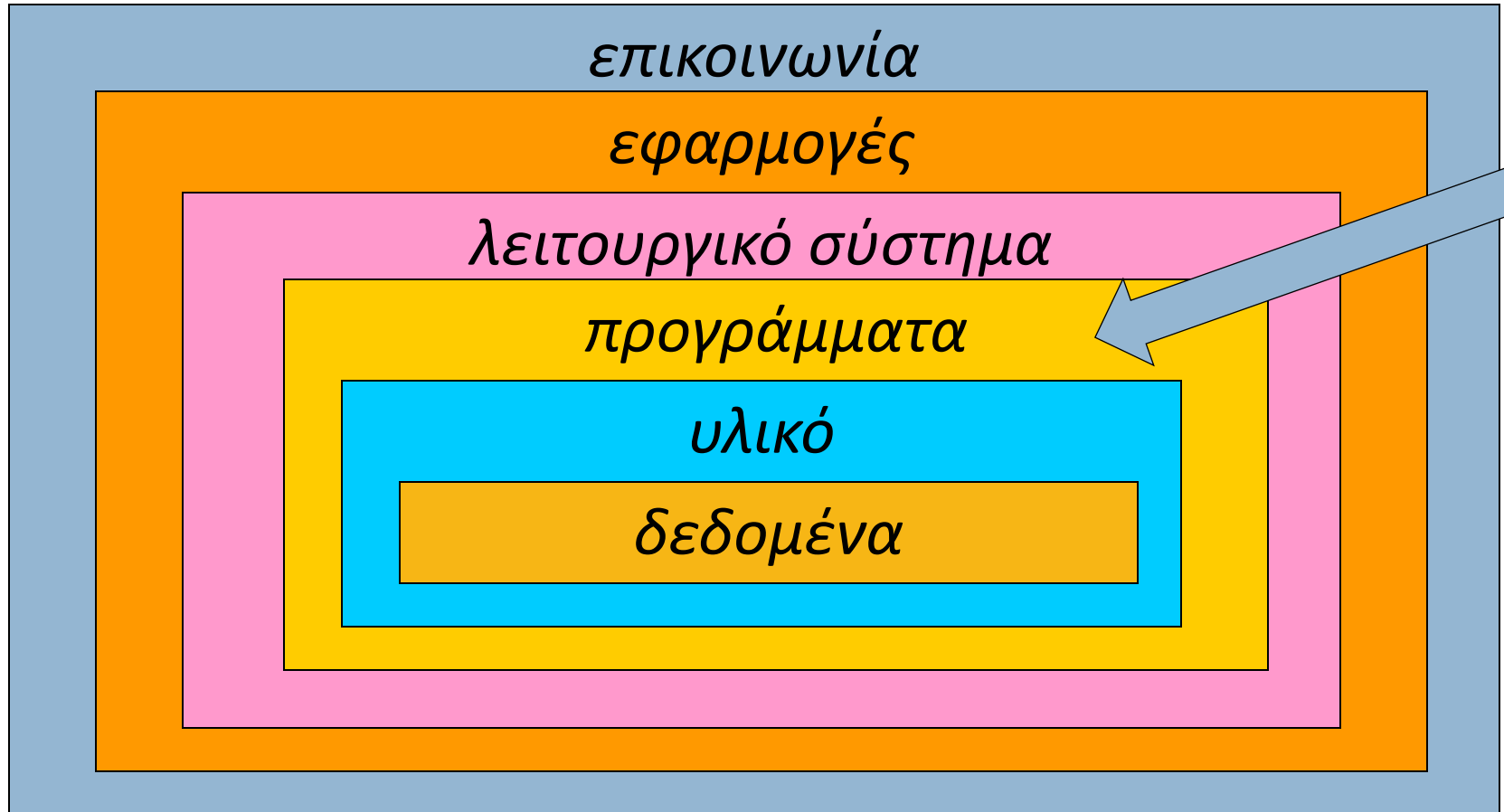
Στόχοι

1

- Να περιγράψουμε τις έννοιες του **υπολογιστικού προβλήματος** και του **αλγορίθμου**.
- Να περιγράψουμε την πορεία από ένα υπολογιστικό πρόβλημα έως ένα **εκτελέσιμο αρχείο** που το λύνει.
- Να εξηγήσουμε τι είναι **προγραμματισμός**, τι είναι **πρόγραμμα**, και τι **γλώσσα προγραμματισμού**.
- Να περιγράψουμε τα στάδια του **κύκλου ανάπτυξης προγράμματος**.
- Να περιγράψουμε τα βασικά συστατικά των **διαγραμμάτων ροής** και του **ψευδοκώδικα**.
- Να επιλύσουμε κάποια υπολογιστικά προβλήματα, μέχρι την φάση της **σχεδίασης**.

Υπολογιστικά συστήματα: Στρώματα

2



Υπολογιστικά προβλήματα

3

Υπολογιστικό πρόβλημα είναι κάθε πρόβλημα στο οποίο μας προμηθεύουν κάποια δεδομένα και μας ζητούν να υπολογίσουμε κάποια αποτελέσματα.

Για να περιγράψουμε ένα υπολογιστικό πρόβλημα, χρησιμοποιούμε κείμενο της μορφής:

Δίνεται

Να υπολογιστεί/βρεθεί/ελεγχθεί αν

Υπολογιστικά προβλήματα

4

Τα υπολογιστικά προβλήματα με τα οποία είμαστε πιο εξοικειωμένοι είναι τα **αριθμητικά προβλήματα**.

Παράδειγμα 1:

Δίνεται ακέραιος αριθμός x .

Να υπολογιστεί το τετράγωνό του, x^2 .

Παράδειγμα 2:

Δίνονται πραγματικοί αριθμοί x και y .

Να υπολογιστεί το άθροισμά τους, $x+y$.

Υπολογιστικά προβλήματα

5

Τα υπολογιστικά προβλήματα με τα οποία είμαστε πιο εξοικειωμένοι είναι τα **αριθμητικά προβλήματα**.

Παράδειγμα 3:

Δίνονται πραγματικοί αριθμοί x , y , και z .

Να υπολογιστεί ο ελάχιστός τους, $\min(x, y, z)$.

Παράδειγμα 4:

Δίνονται ποσό X και σύνολο κερμάτων K .

Να βρεθούν στο K κέρματα με άθροισμα X .

Υπολογιστικά προβλήματα

6

Όμως ένα υπολογιστικό πρόβλημα μπορεί να αφορά και σε άλλα, μη αριθμητικά αντικείμενα.

Παράδειγμα 5:

Δίνονται γράμματα Γ_1 και Γ_2 .

Να βρεθεί λέξη από Γ_1 που να τελειώνει σε Γ_2 .

Παράδειγμα 6:

Δίνονται χάρτης X και δύο σημεία του A, B .

Να βρεθεί διαδρομή στον X από το A στο B .

Αλγόριθμοι

7

Ως (συστηματική) **λύση** ενός υπολογιστικού πρόβληματος θεωρούμε κάθε ακολουθία σαφών βημάτων που μας οδηγεί από τα δεδομένα (όποιες τιμές κι αν έχουν) στα αποτελέσματα (που αντιστοιχούν σωστά στα δεδομένα).

Π.χ.: Λύση για το πρόβλημα του Παραδείγματος 2

Δίνονται πραγματικοί αριθμοί x και y .

Να υπολογιστεί το άθροισμά τους, $x+y$.

είναι κάθε ακολουθία σαφών βημάτων που όταν την εκτελέσουμε πάνω σε δύο *οποιοσδήποτε* πραγματικούς αριθμούς θα καταλήξουμε στο άθροισμά τους.

(Μια τέτοια ακολουθία βημάτων μάθατε στο δημοτικό.)

Αλγόριθμοι

8

Ως (συστηματική) **λύση** ενός υπολογιστικού πρόβληματος θεωρούμε κάθε ακολουθία σαφών βημάτων που μας οδηγεί από τα δεδομένα (όποιες τιμές κι αν έχουν) στα αποτελέσματα (που αντιστοιχούν σωστά στα δεδομένα).

Κάθε τέτοια ακολουθία βημάτων λέγεται **αλγόριθμος**.

Προγραμματισμός

9

**υπολογιστικό
πρόβλημα**

Δίνεται ακέραιος x .
Να υπολογιστεί το
τετράγωνό του, x^2 .

αλγόριθμος

1. Αρχή.
2. Διάβασε x .
3. Τύπωσε $x*x$.
4. Τέλος.

πρόγραμμα

```
#include <stdio.h>
void main(void) {
    int x;
    scanf("%d", &x);
    printf("%d", x*x);
}
```

**εκτελέσιμο
αρχείο**

```
0010101001010111101
1101010101101001001
0011010101010001010
100
```

Προγραμματισμός: η διαδικασία του να επινοήσουμε αλγόριθμο και να συντάξουμε πρόγραμμα.

Πρόγραμμα: αλγόριθμος σε τυπική μορφή (δηλ. μορφή έτοιμη να μεταφραστεί σε μορφή κατανοητή από τον Η/Υ).

Μετάφραση (compilation)

Εκτελέσιμο αρχείο: αλγόριθμος σε μορφή κατανοητή από τον Η/Υ.

Προγραμματισμός

10

Το σύνολο των γραμματικών και συντακτικών κανόνων που ακολουθούμε κατά την συγγραφή ενός προγράμματος λέγεται **γλώσσα προγραμματισμού** (*programming language*).

Η μετατροπή του προγράμματος σε εκτελέσιμο αρχείο λέγεται **μετάφραση** (*compilation*).

Το ειδικό λογισμικό που επιτελεί την μετάφραση λέγεται **μεταφραστής** (*compiler*).

Κύκλος ανάπτυξης προγράμματος

11

Γενικά, η πορεία μέχρι το εκτελέσιμο αρχείο είναι πιο σύνθετη από την απλή ακολουθία «προγραμματισμός + μετάφραση». Είναι μια διαδικασία που αποτελείται από πολλά στάδια και ενδέχεται να μην τερματίσει ποτέ.

Η διαδικασία αυτή ονομάζεται **κύκλος ανάπτυξης προγράμματος** (*program development cycle*).

Κύκλος ανάπτυξης προγράμματος

12

Τον κύκλο ανάπτυξης προγράμματος συνιστούν 6 στάδια:

1. **Περιγραφή** του προβλήματος και τυχόν περιορισμών.
2. **Ανάλυση** του προβλήματος και επιλογή μιας λύσης.
3. **Σχεδίαση** της επιλεγμένης λύσης.
4. **Κωδικοποίηση** σε γλώσσα προγραμματισμού.
5. **Έλεγχος** του προγράμματος και διόρθωση λαθών.
6. **Συντήρηση** του προγράμματος.

Στάδιο 1: Περιγραφή

13

Τις περισσότερες φορές, το υπολογιστικό πρόβλημα δεν μας δίνεται στην απλή μορφή «Δίνεται... . Να βρεθεί... .». Αντιθέτως, αυτό που έχουμε μπροστά μας είναι ένα πρόβλημα του πραγματικού κόσμου. Πρέπει οι ίδιοι να διακρίνουμε ποια συστατικά αυτού του προβλήματος είναι σημαντικά και να τα καταγράψουμε με σαφήνεια.

- ▣ Ποια είναι τα δεδομένα (*είσοδος, input*);
- ▣ Ποια αποτελέσματα πρέπει να προκύψουν (*έξοδος, output*);
- ▣ Εκτός από τον προφανή περιορισμό (να προκύπτουν πάντα τα σωστά αποτελέσματα που αντιστοιχούν στα δεδομένα), μήπως πρέπει να ικανοποιούνται και άλλοι, πρόσθετοι περιορισμοί;

Στάδιο 1: Περιγραφή (Παράδειγμα)

14

Παράδειγμα: Μόλις τελείωσε το εξάμηνο, και θέλουμε να μάθουμε ποιος ήταν ο μέγιστος βαθμός στο ΕΠΛ00x.
Ποιο ακριβώς είναι το υπολογιστικό πρόβλημα;

■ **Είσοδος:**

~~Ο κατάλογος όλων των φοιτητών & των βαθμών τους.~~
Ο κατάλογος όλων των βαθμών.

■ **Έξοδος:**

~~Το όνομα του φοιτητή με τον μεγαλύτερο βαθμό.~~
Ο μεγαλύτερος από όλους τους βαθμούς.

■ **Περιορισμοί:**

Το πρόγραμμα θα πρέπει να λειτουργεί γρήγορα.

Στάδιο 1: Περιγραφή (Παράδειγμα)

15

Παράδειγμα: Μόλις τελείωσε το εξάμηνο, και θέλουμε να μάθουμε ποιος ήταν ο μέγιστος βαθμός στο ΕΠΛ003. Ποιο ακριβώς είναι το υπολογιστικό πρόβλημα;

■ **Είσοδος:**

Μια ακολουθία μη αρνητικών αριθμών x_1, x_2, \dots, x_N από το πληκτρολόγιο (σήμα τέλους: αριθμός < 0).

■ **Έξοδος:**

Ο μεγαλύτερος από τους δεδομένους αριθμούς, δηλ. ο $\max(x_1, x_2, \dots, x_N)$, τυπωμένος στην οθόνη.

■ **Περιορισμοί:**

Το πρόγραμμα θα πρέπει να λειτουργεί γρήγορα.

Στάδιο 2: Ανάλυση

16

Τώρα που το υπολογιστικό πρόβλημα είναι απολύτως σαφές, προσπαθούμε να επινοήσουμε κάποια λύση.

Θυμηθείτε: λύση υπολογιστικού προβλήματος = αλγόριθμος που λύνει το υπολογιστικό πρόβλημα.

Προσέξτε: Ίσως υπάρχουν πολλές λύσεις, δηλαδή πολλοί αλγόριθμοι που λύνουν το ίδιο πρόβλημα!

Για κάθε λύση που σκεφτόμαστε, προσπαθούμε να βεβαιωθούμε ότι σε κάθε είσοδο θα παράγει τη σωστή έξοδο και ότι θα ικανοποιεί τους περιορισμούς.

Διερευνούμε το ενδεχόμενο να υπάρχουν περισσότερες από μία λύσεις. Αν όντως επινοήσουμε περισσότερες από μία, τότε πρέπει να επιλέξουμε τη βέλτιστη.

Στάδιο 2: Ανάλυση (Παράδειγμα)

17

Στο παράδειγμά μας, μια λύση είναι η εξής:

*Διαβάζουμε τους x_1, x_2, \dots, x_N από το πληκτρολόγιο και τους αποθηκεύουμε. Έπειτα, για κάθε x_i ελέγχουμε μήπως είναι μέγιστος, δηλαδή:
Διατρέχουμε τους άλλους και ελέγχουμε αν είναι όλοι $\leq x_i$. Αν ναι, τότε τυπώνουμε τον x_i στην οθόνη, και τερματίζουμε.*

Είναι αυτή η μέθοδος πράγματι λύση; Παράγει το σωστό αποτέλεσμα για οποιεσδήποτε τιμές των x_1, x_2, \dots, x_N ;

Ναι.

Στάδιο 2: Ανάλυση (Παράδειγμα)

18

Μια άλλη λύση είναι η εξής:

Διαβάζουμε τους x_1, x_2, \dots, x_N από το πληκτρολόγιο και ανά πάσα στιγμή παρακολουθούμε ποιος ήταν μέγιστος από τους μέχρι τώρα διαβασμένους. Στο τέλος τυπώνουμε στην οθόνη όποιον θυμόμαστε ως μέγιστο.

Είναι αυτή η μέθοδος πράγματι λύση; Παράγει το σωστό αποτέλεσμα για οποιεσδήποτε τιμές των x_1, x_2, \dots, x_N ;

Ναι.

Στάδιο 2: Ανάλυση (Παράδειγμα)

19

Ποια από τις δύο λύσεις είναι καλύτερη;

Η δεύτερη λύση διατρέχει τους αριθμούς μόνο 1 φορά.

Αντιθέτως, η πρώτη λύση ενδέχεται να τους διατρέχει πολλές φορές. (Στη χειρότερη περίπτωση, αν ο μέγιστος έχει δοθεί τελευταίος, οι αριθμοί διατρέχονται N φορές!)

Άρα η δεύτερη λύση είναι ταχύτερη. Επομένως πρέπει να επιλέξουμε αυτήν.

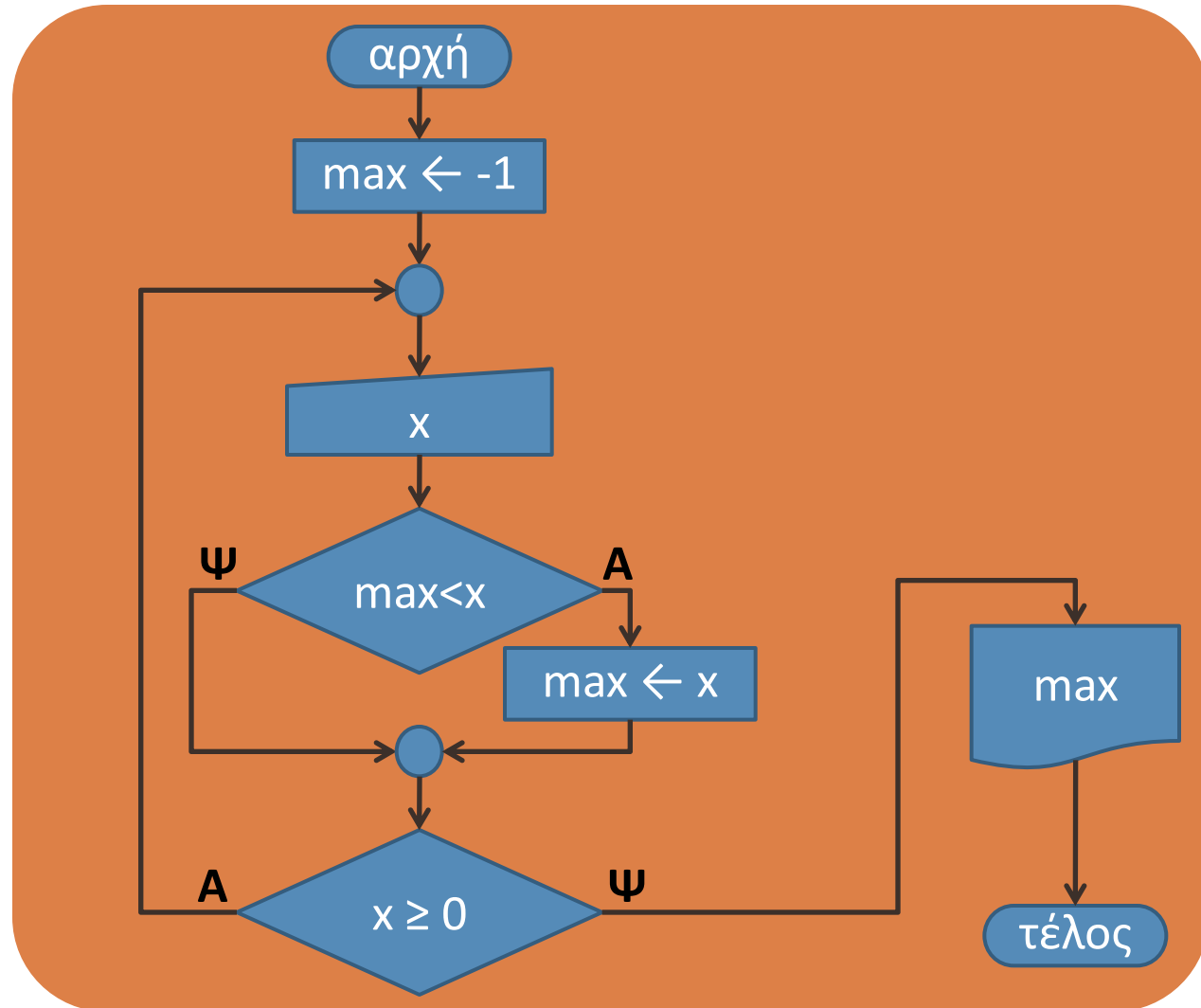
Στάδιο 3: Σχεδίαση

Τώρα που γνωρίζουμε ποια λύση θέλουμε να υλοποιήσουμε, αναπτύσσουμε τον αλγόριθμο λεπτομερώς:

1. Μετατρέπουμε την λύση που έχουμε περιγράψει σε **διαγραμμα ροής** (*flow chart*). Αυτό αναπαριστά σχηματικά τη ροή του ελέγχου στον αλγόριθμό μας.
2. Μετατρέπουμε το διάγραμμα ροής σε **ψευδοκώδικα** (*pseudocode*). Αυτός αναπαριστά τον αλγόριθμό μας ως ακολουθία βημάτων, καθένα από τα οποία περιγράφεται μέσω ενός μείγματος λέξεων της ελληνικής ή της αγγλικής γλώσσας και κάποιων εντολών που είναι κοινές σε πολλές γλώσσες προγραμματισμού.

Στάδιο 3: Σχεδίαση (Παράδειγμα)

21



*Διάγραμμα
ροής για τη
δεύτερη λύση.*

Στάδιο 3: Σχεδίαση (Παράδειγμα)

22

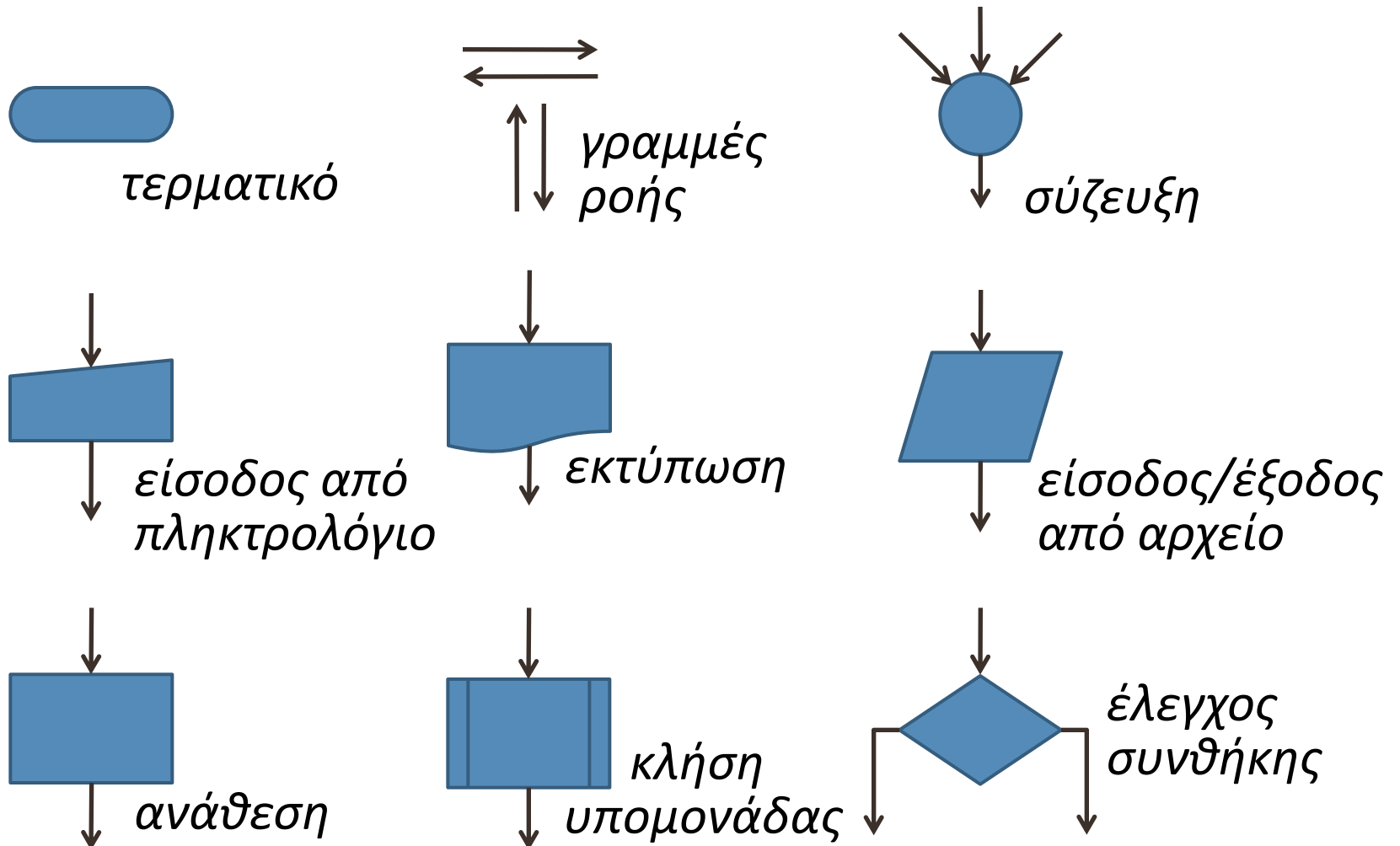
Ψευδοκώδικας
για τη δεύτερη
λύση.

1. Αρχή.
2. $max \leftarrow -1$.
3. Επανάληψη:
4. Διάβασε x .
5. Αν $max < x$ τότε $max \leftarrow x$.
6. Όσο $x \geq 0$.
7. Τύπωσε max .
8. Τέλος.

Στάδιο 3: Σχεδίαση

23

Βασικά σύμβολα σε διάγραμμα ροής:



Στάδιο 3: Σχεδίαση

24

Βασικές εντολές σε ψευδοκώδικα:

<i>Αρχή / Τέλος</i>	τερματικό
<i>Διάβασε ...</i>	είσοδος από πληκτρολόγιο
<i>Τύπωσε ...</i>	εκτύπωση
<i>Διάβασε/Γράψε ...</i>	είσοδος /έξοδος από αρχείο
<i>... ← ...</i>	ανάθεση
<i>Αν ... τότε ...</i>	έλεγχος συνθήκης
<i>Αν ... τότε ... αλλιώς ...</i>	έλεγχος συνθήκης
<i>Επανάληψη ... όσο ...</i>	βρόχος επανάληψης
<i>Όσο ... επανέλαβε ...</i>	βρόχος επανάληψης

Στάδιο 4: Κωδικοποίηση

25

Από τον ψευδοκώδικα, συντάσσουμε πρόγραμμα σε κάποια γλώσσα προγραμματισμού (π.χ. C, Java, κ.ά.).

Μέσω μεταφραστή μετατρέπουμε το πρόγραμμα σε γλώσσα η οποία είναι αναγνωρίσιμη από τη μηχανή.

Στο στάδιο αυτό γίνεται ο έλεγχος και η διόρθωση των **συντακτικών λαθών** (*syntax errors*), δηλαδή των λαθών που οφείλονται σε λανθασμένη χρήση των κανόνων της γλώσσας προγραμματισμού.

Στάδιο 4: Κωδικοποίηση (Παράδειγμα)

26

*Πρόγραμμα
στη γλώσσα C
για τη δεύτερη
λύση.*

```
#include <stdio.h>

void main(void) {
    int x, max = -1;

    do {
        scanf("%d", &x);
        if (max<x) max=x;
    } while (x>=0);

    printf("%d\n", max);
}
```

Στάδιο 4: Κωδικοποίηση (Παράδειγμα)

27

```
#include <stdio.h>

void main(void) {
    int x, max = -1;

    do {
        scanf("%d", &x);
        if (max<x) max=x;
    } while (x>=0);

    printf("%d\n", max);
}
```

*πρόγραμμα στη
γλώσσα C για τη
δεύτερη λύση*

μετάφραση

*Ο μεταφραστής δέχεται
στην είσοδο ένα αρχείο
κειμένου (π.χ.:
findmax.c) που περιέχει
το πρόγραμμα και
παράγει στην έξοδο ένα
εκτελέσιμο αρχείο
(findmax.exe).*

*(Η μία αναφορά με όλα
τα συντακτικά λάθη που
εντόπισε.)*

```
0010101001010110111101110110
01011101011001001011001101011
0010100101001001001010101010
010000001111110101111001001
110010011001001001000000111
1110101111011010010010011100
1001001001001001110010010101
010010001000000111111111101
1111110111101010010101010010
0100101110000100100101010101
0101000000111111010111110011
0000011111010101001011001001
```

*εκτελέσιμο
αρχείο για τη
δεύτερη λύση*

Στάδιο 5: Έλεγχος

28

Πραγματοποιούμε πολλές δοκιμές προκειμένου να βεβαιωθούμε ότι το πρόγραμμα δουλεύει σωστά.

Διορθώνουμε τυχόν **λογικά σφάλματα** που ανακύπτουν, δηλ. σφάλματα που αφορούν στη σχεδίαση της λύσης.

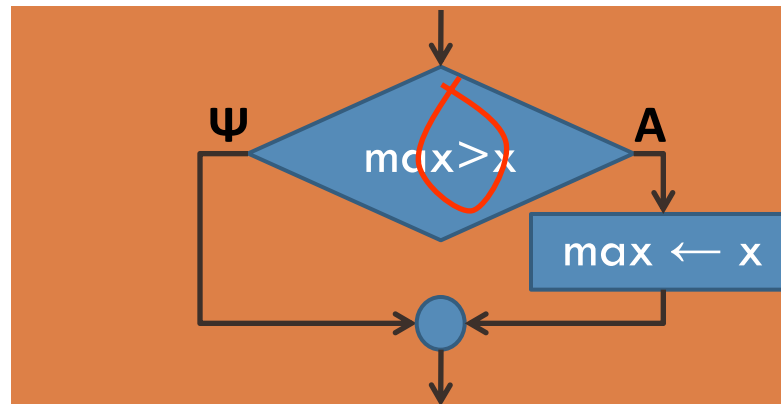
Στάδιο 5: Έλεγχος (Παράδειγμα)

29

Στο παράδειγμά μας, ίσως κατά την κωδικοποίηση να πληκτρολογήσαμε “max>x” αντί για “max<x” :

```
scanf ("%d", &x) ;  
if (max>x) max=x;
```

Ή ίσως να κάναμε το ίδιο λάθος κατά την σχεδίαση:



Ο μεταφραστής δεν μπορεί να ανιχνεύσει τέτοια λογικά σφάλματα. Εντοπίζονται μόνο στο στάδιο του ελέγχου.

Στάδιο 6: Συντήρηση

Συντάσσουμε **τεκμηρίωση** (*documentation*), δηλαδή ένα εγχειρίδιο που να εξηγεί την χρήση του προγράμματος.

Εγκαθιστούμε το πρόγραμμα και αρχίζουμε να το χρησιμοποιούμε.

Παρακολουθούμε την χρήση του προγράμματος και ενημερωνόμαστε για τυχόν νέες απαιτήσεις ή σφάλματα που ανακύπτουν. Όποτε είναι απαραίτητο, τροποποιούμε το πρόγραμμα καταλλήλως.

Προβλήματα

31

Άκολουθεί σειρά προβλημάτων, που πρέπει να επιλύσουμε έως και τη φάση της σχεδίασης. Δηλαδή πρέπει:

- 1. Περιγραφή:** Να παραθέσουμε μια σαφή περιγραφή του προβλήματος (αν δεν μας έχει ήδη δοθεί).
- 2. Ανάλυση:** Να επινοήσουμε κάποιον αλγόριθμο (ή αλγορίθμους, και να επιλέξουμε τον καλύτερο).
- 3. Σχεδίαση:** Για τον αλγόριθμό μας, να σχεδιάσουμε ένα διάγραμμα ροής και έπειτα έναν ψευδοκώδικα.

Πρόβλημα 1

32

Να γραφεί πρόγραμμα που να υπολογίζει τον **μέσο όρο** δύο αριθμών.

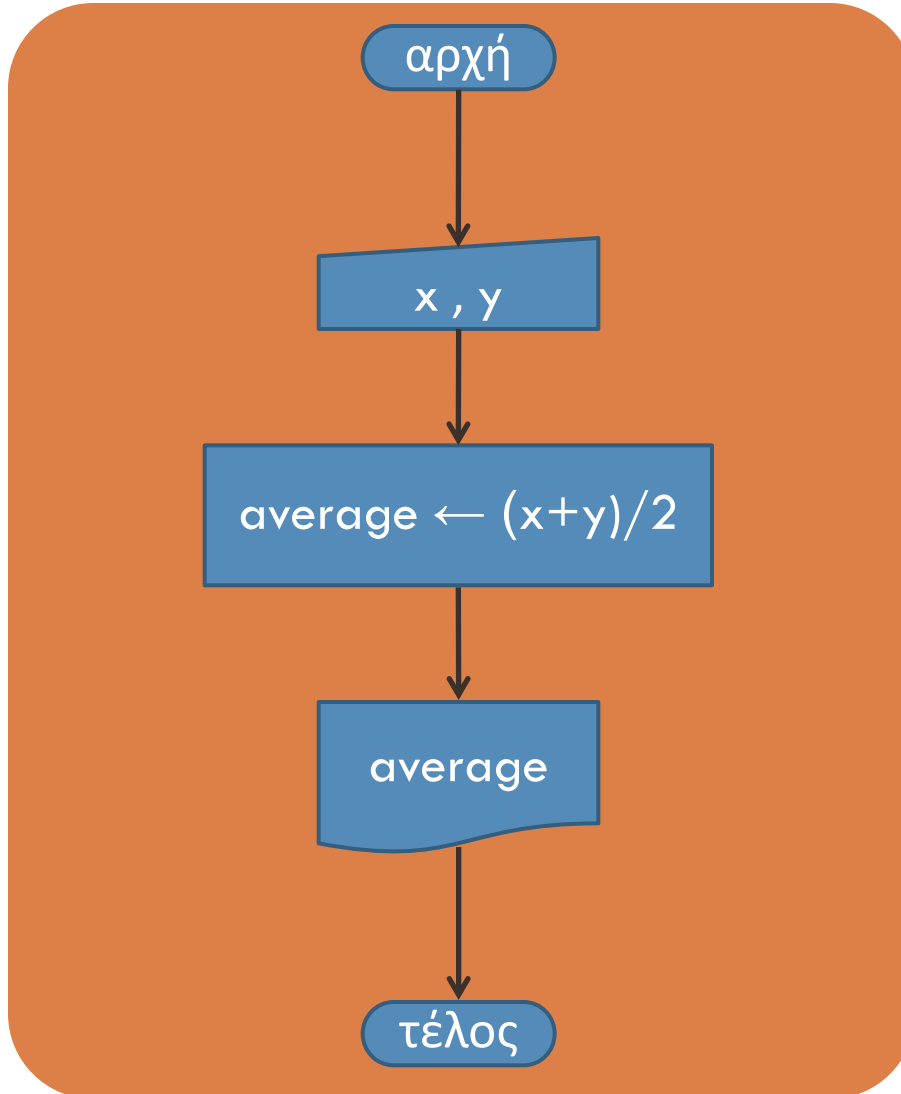
Μια πιο σαφής περιγραφή:

Δίνονται δύο πραγματικοί αριθμοί x και y , από το πληκτρολόγιο.

Να υπολογιστεί ο μέσος όρος τους, $(x+y)/2$, και να τυπωθεί στην οθόνη.

Πρόβλημα 1: Διάγραμμα ροής

33



x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
average	ο ζητούμενος μέσος όρος

Πρόβλημα 1: Ψευδοκώδικας

34

1. Αρχή.
2. Διάβασε x , y .
3. $average \leftarrow (x+y)/2$.
4. Τύπωσε $average$.
5. Τέλος.

x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
$average$	ο ζητούμενος μέσος όρος

Πρόβλημα 2

35

Να γραφεί πρόγραμμα που να υπολογίζει τον **ελάχιστο** δύο αριθμών.

Μια πιο σαφής περιγραφή:

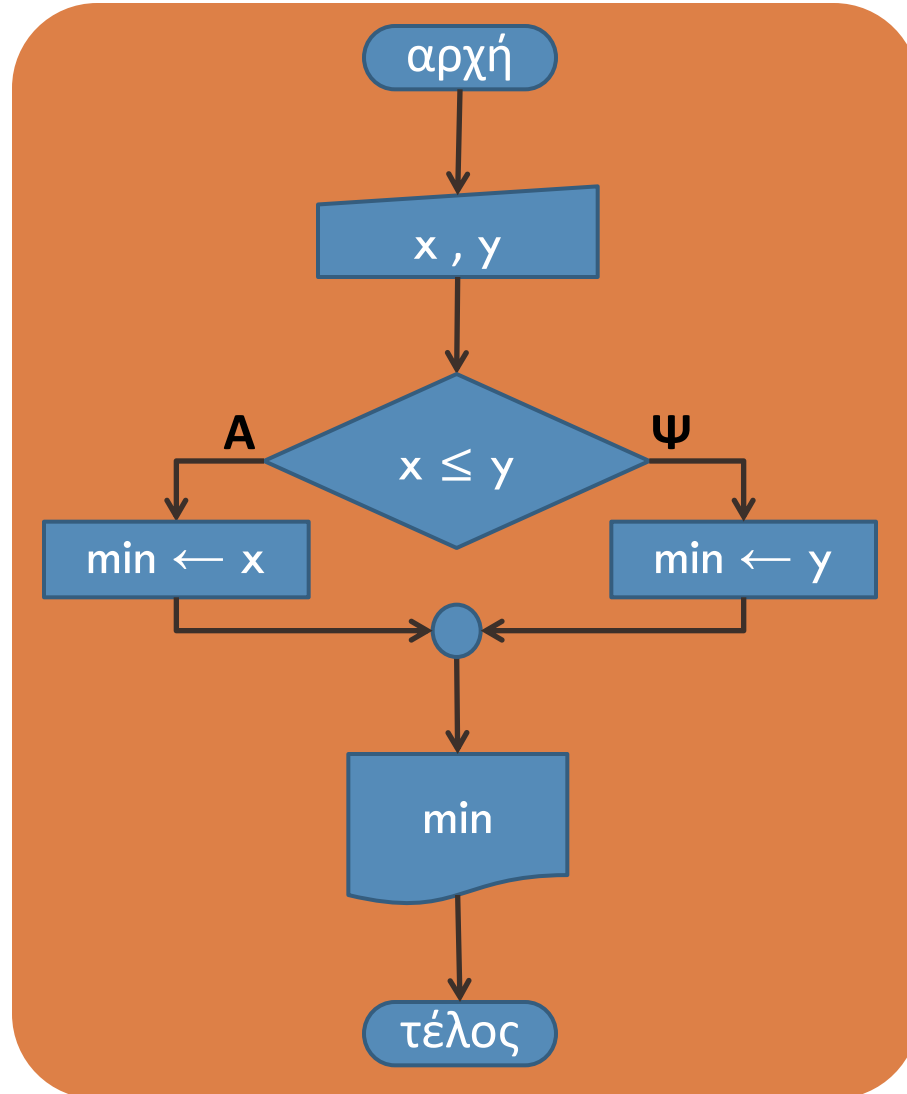
Δίνονται δύο πραγματικοί αριθμοί x και y , από το πληκτρολόγιο.

Να υπολογιστεί ο ελάχιστος των αριθμών, $\min(x,y)$, και να τυπωθεί στην οθόνη.

*(Διευκρίνιση: Σε μια ανάθεση, η παράσταση δεξιά του ← μπορεί να περιέχει μόνο τις πράξεις +, -, *, και / .)*

Πρόβλημα 2: Διάγραμμα ροής

36



x ο 1^{ος} αριθμός εισόδου

y ο 2^{ος} αριθμός εισόδου

min ο ζητούμενος ελάχιστος

Πρόβλημα 2: Ψευδοκώδικας

37

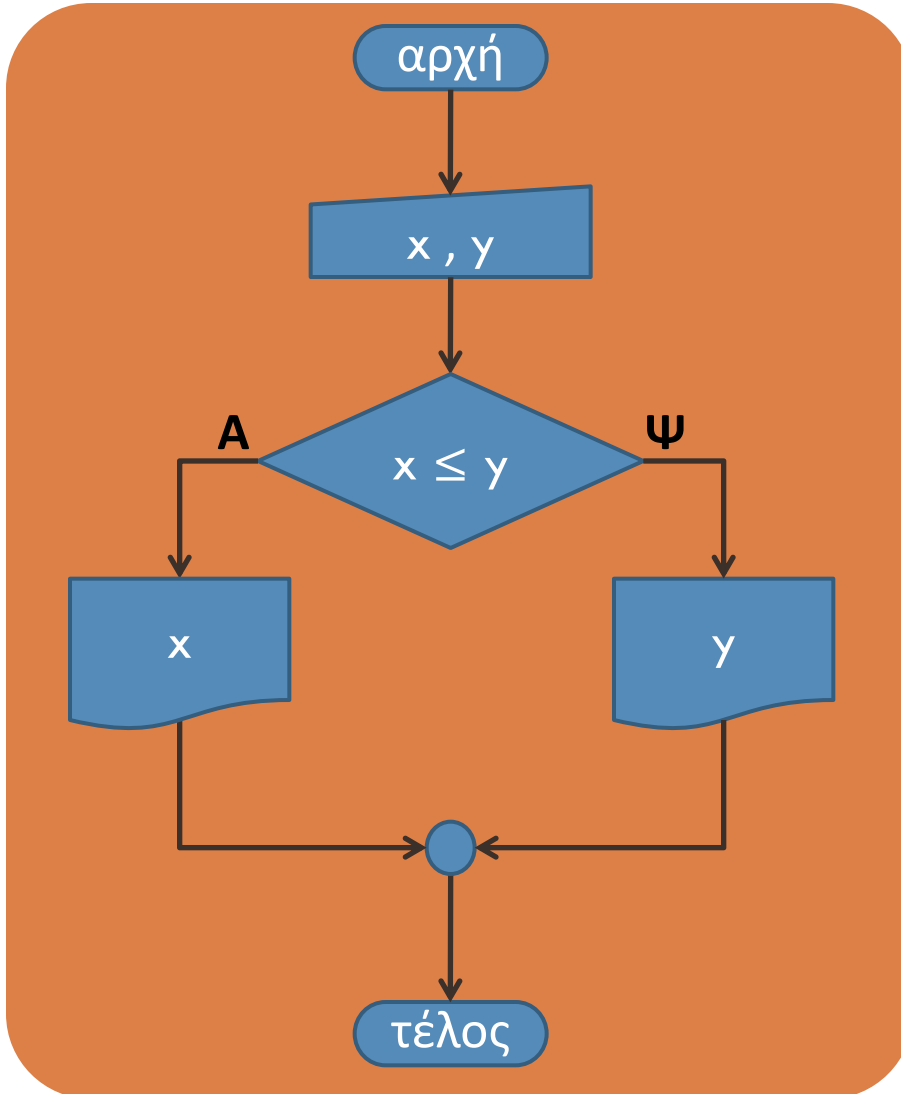
1. Αρχή.
2. Διάβασε x, y .
3. Αν $x \leq y$
4. τότε $min \leftarrow x$
5. αλλιώς $min \leftarrow y$.
6. Τύπωσε min .
7. Τέλος.

x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
min	ο ζητούμενος ελάχιστος

Πρόβλημα 2: Εναλλακτικό διάγραμμα ροής

38

x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου



Πρόβλημα 2: Εναλλακτικός ψευδοκώδικας

39

1. Αρχή.
2. Διάβασε x, y .
3. Αν $x \leq y$
4. τότε Τύπωσε x
5. αλλιώς Τύπωσε y .
6. Τέλος.

x ο 1^{ος} αριθμός εισόδου

y ο 2^{ος} αριθμός εισόδου

Ασκήσεις 1 & 2

40

Δώστε διάγραμμα ροής και ψευδοκώδικα για αλγορίθμους που λύνουν τα παρακάτω προβλήματα:

Δίνονται τρεις πραγματικοί αριθμοί x, y, z , από το πληκτρολόγιο.

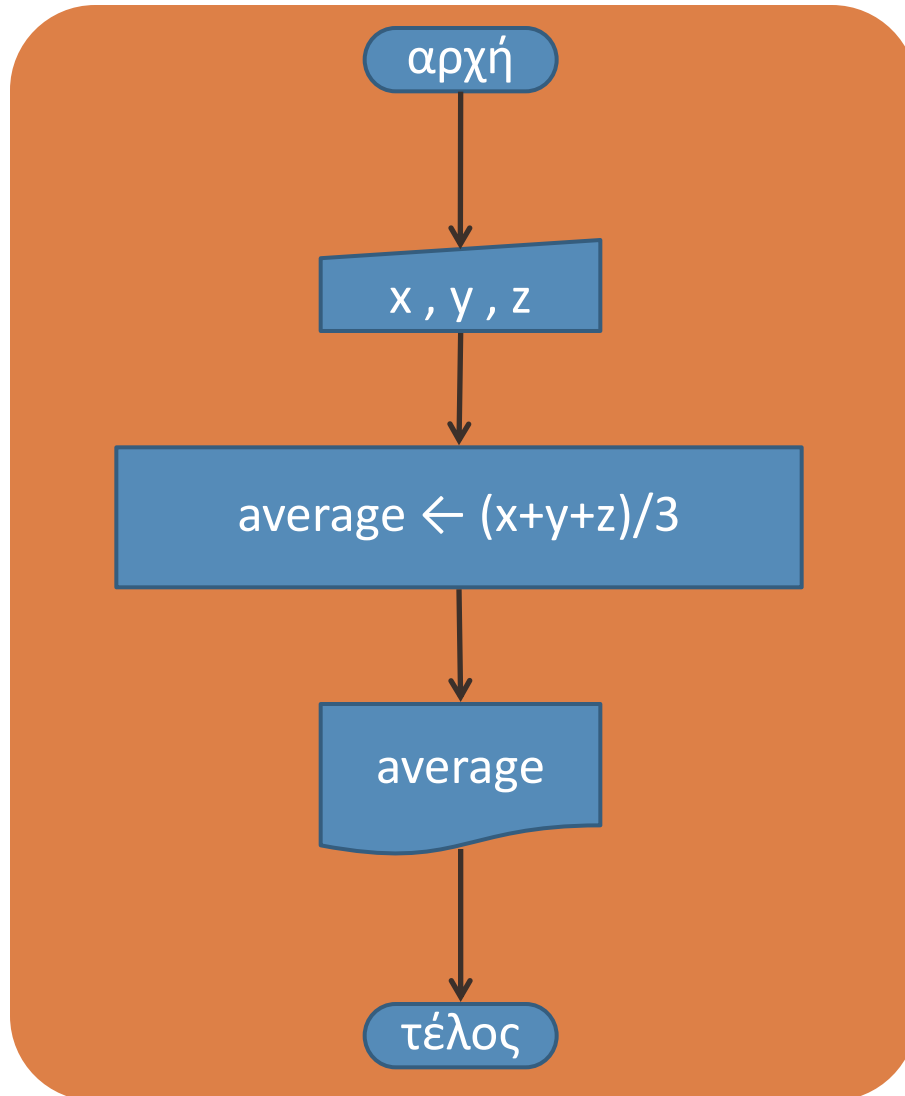
Να υπολογιστεί ο μέσος όρος τους, $(x+y+z)/3$, και να τυπωθεί στην οθόνη.

Δίνονται τρεις πραγματικοί αριθμοί x, y, z , από το πληκτρολόγιο.

Να υπολογιστεί ο ελάχιστος των αριθμών, $\min(x,y,z)$, και να τυπωθεί στην οθόνη.

Άσκηση 1: Διάγραμμα ροής

41



x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
z	ο 3 ^{ος} αριθμός εισόδου
average	ο ζητούμενος μέσος όρος

Άσκηση 1: Ψευδοκώδικας

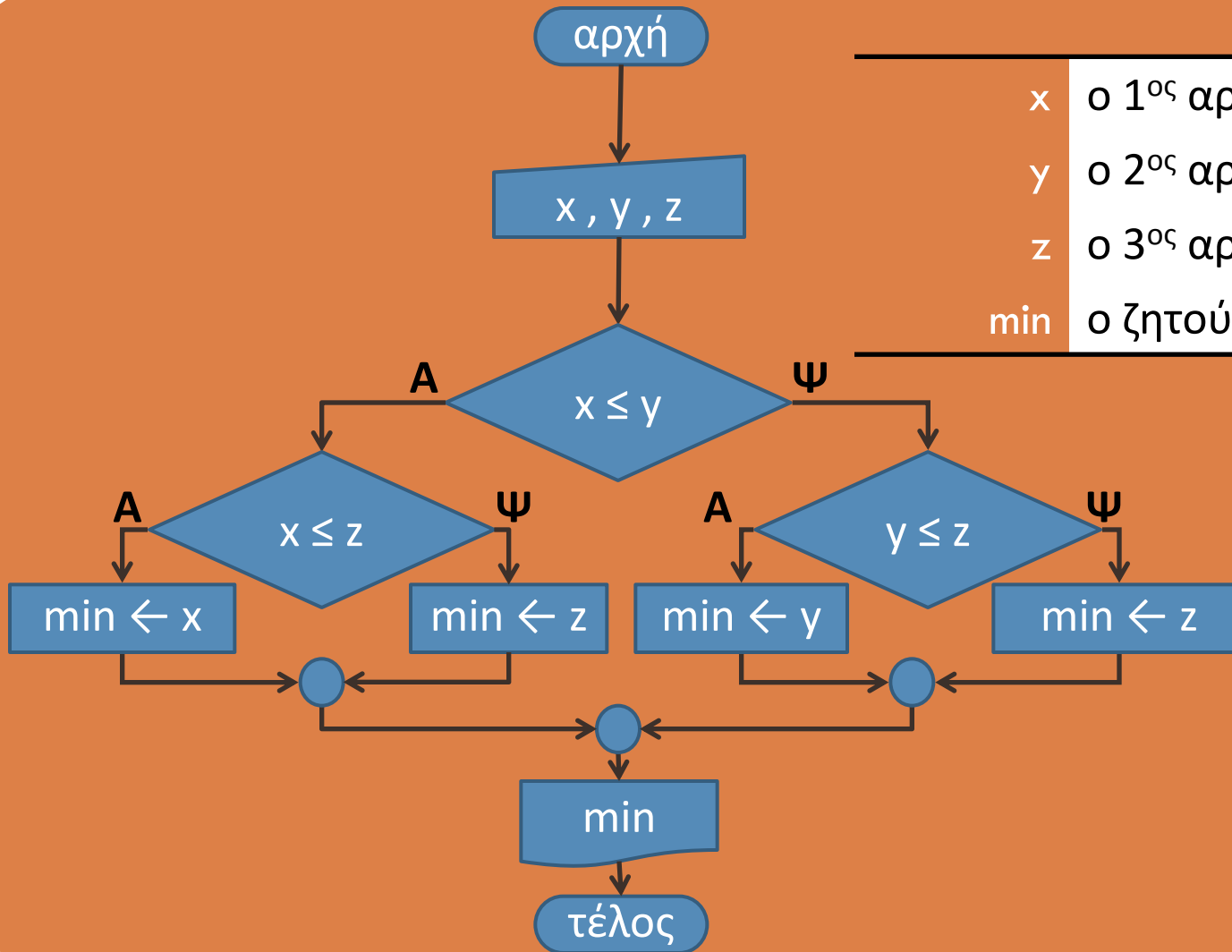
42

1. Αρχή.
2. Διάβασε x , y , z .
3. $average \leftarrow (x+y+z)/3$.
4. Τύπωσε $average$.
5. Τέλος.

x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
z	ο 3 ^{ος} αριθμός εισόδου
$average$	ο ζητούμενος μέσος όρος

Άσκηση 2: Διάγραμμα ροής

43



x ο 1^{ος} αριθμός εισόδου

y ο 2^{ος} αριθμός εισόδου

z ο 3^{ος} αριθμός εισόδου

min ο ζητούμενος ελάχιστος

Άσκηση 2: Ψευδοκώδικας

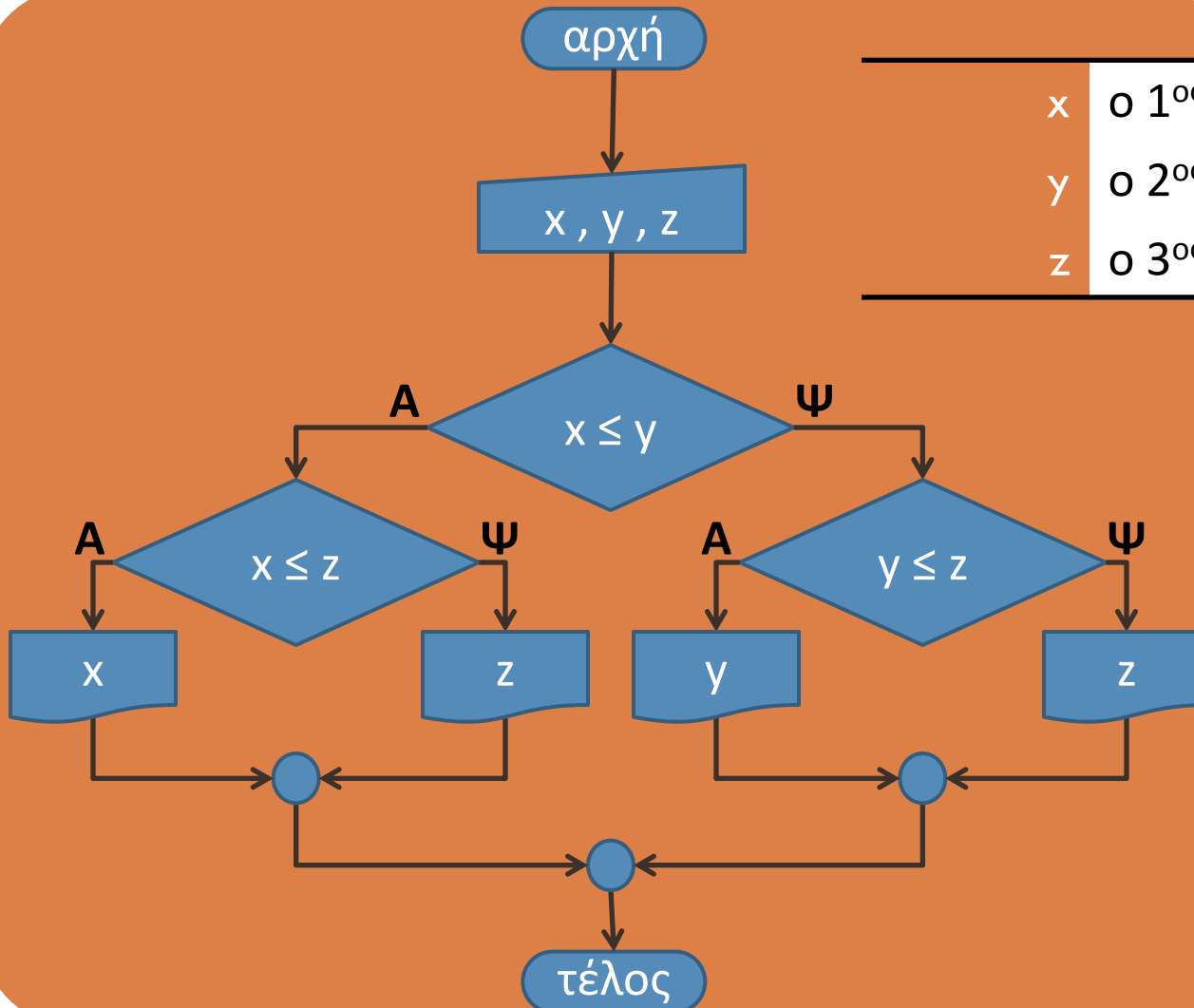
44

1. Αρχή.
2. Διάβασε x, y, z .
3. Αν $x \leq y$
4. τότε Αν $x \leq z$
5. τότε $min \leftarrow x$
6. αλλιώς $min \leftarrow z$
7. αλλιώς Αν $y \leq z$
8. τότε $min \leftarrow y$
9. αλλιώς $min \leftarrow z$.
10. Τύπωσε min .
11. Τέλος.

x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
z	ο 3 ^{ος} αριθμός εισόδου
min	ο ζητούμενος ελάχιστος

Άσκηση 2: Εναλλακτικό διάγραμμα ροής

45



x	ο 1 ^{ος} αριθμός εισόδου
γ	ο 2 ^{ος} αριθμός εισόδου
z	ο 3 ^{ος} αριθμός εισόδου

Άσκηση 2:

Εναλλακτικός ψευδοκώδικας

46

1. Αρχή.
2. Διάβασε x, y, z .
3. Αν $x \leq y$
4. τότε Αν $x \leq z$
5. τότε Τύπωσε x
6. αλλιώς Τύπωσε z
7. αλλιώς Αν $y \leq z$
8. τότε Τύπωσε y
9. αλλιώς Τύπωσε z .
10. Τέλος.

x	ο 1 ^{ος} αριθμός εισόδου
y	ο 2 ^{ος} αριθμός εισόδου
z	ο 3 ^{ος} αριθμός εισόδου

Πρόβλημα 3

47

Να γραφεί πρόγραμμα που να δέχεται μια απροσδιόριστα μεγάλη ακολουθία αριθμών (η οποία τερματίζει με τον αριθμό 0) και να υπολογίζει τα **τετράγωνα** αυτών.

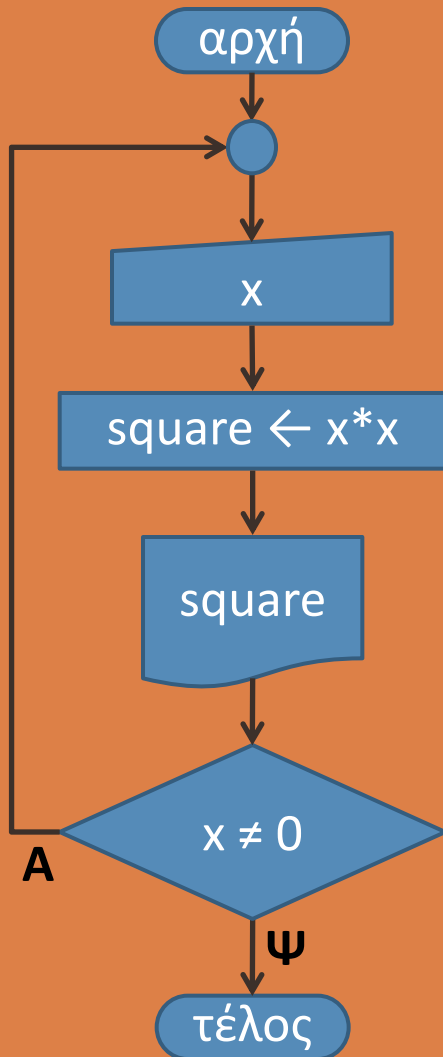
Μια πιο σαφής περιγραφή:

Δίνεται μια ακολουθία αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 0$).

Να υπολογιστούν και να τυπωθούν στην οθόνη τα τετράγωνα των αριθμών, $x_1^2, x_2^2, \dots, x_N^2$.

Πρόβλημα 3: Διάγραμμα ροής

48



x	ο τρέχων αριθμός
square	το τρέχον τετράγωνο

Πρόβλημα 3: Ψευδοκώδικας

49

1. Αρχή.
2. Επανάληψη:
3. Διάβασε x .
4. $square \leftarrow x * x$.
5. Τύπωσε $square$.
6. όσο $x \neq 0$.
7. Τέλος.

x	ο τρέχων αριθμός
$square$	το τρέχον τετράγωνο

Πρόβλημα 4

50

Να γραφεί πρόγραμμα που να δέχεται μια απροσδιόριστα μεγάλη ακολουθία αριθμών (η οποία τερματίζει με τον αριθμό 0) και να υπολογίζει το **άθροισμα** αυτών.

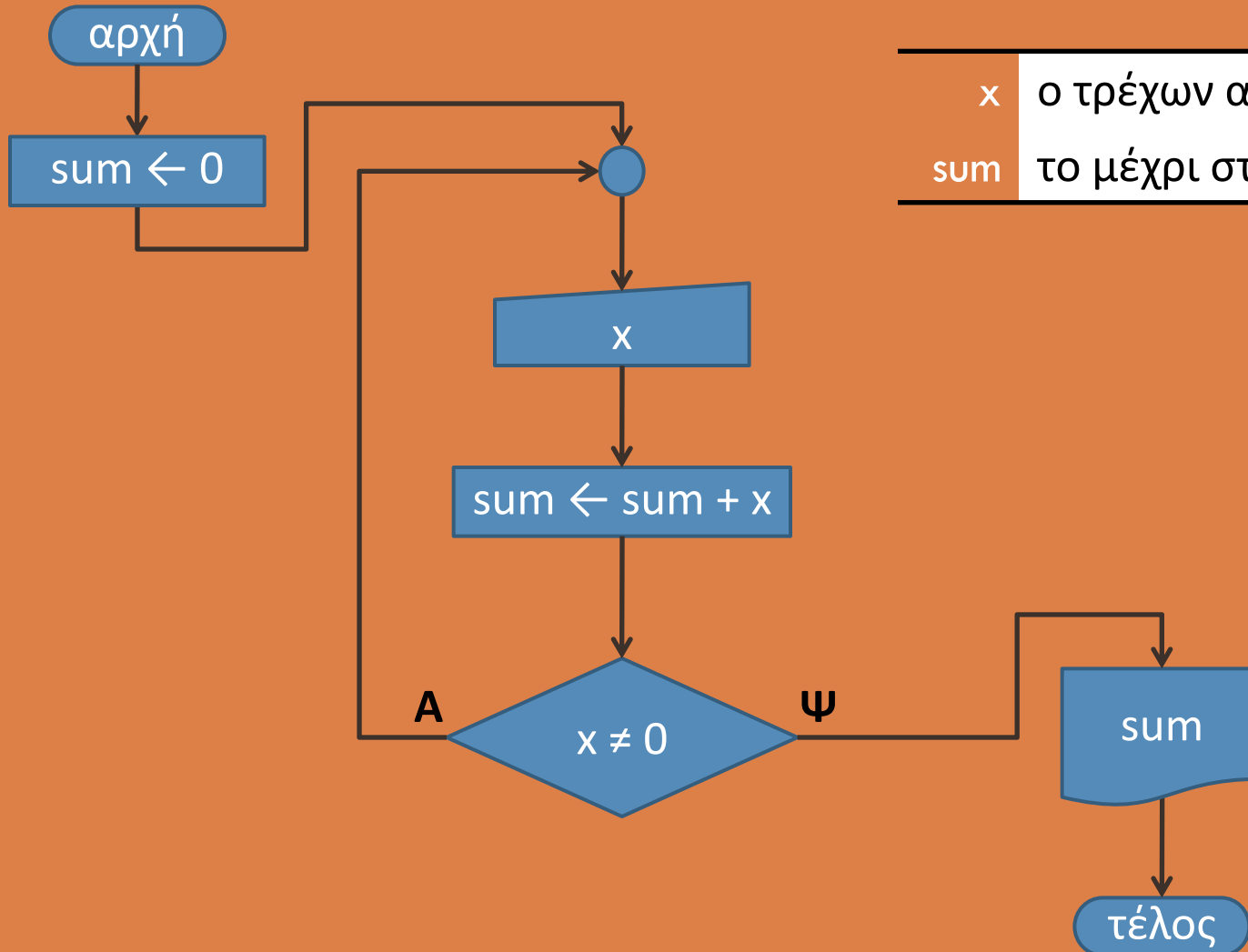
Μια πιο σαφής περιγραφή:

Δίνεται μια ακολουθία αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 0$).

Να υπολογιστεί και να τυπωθεί στην οθόνη το άθροισμά τους, $x_1 + x_2 + \dots + x_N$.

Πρόβλημα 4: Διάγραμμα ροής

51



x ο τρέχων αριθμός

sum το μέχρι στιγμής άθροισμα

Πρόβλημα 4: Ψευδοκώδικας

52

1. Αρχή.
2. $sum \leftarrow 0$.
3. Επανάληψη:
4. Διάβασε x .
5. $sum \leftarrow sum + x$.
6. όσο $x \neq 0$.
7. Τύπωσε sum .
8. Τέλος.

x ο τρέχων αριθμός

sum το μέχρι στιγμής άθροισμα

Ασκήσεις 3 & 4

53

Δώστε διάγραμμα ροής και ψευδοκώδικα για αλγορίθμους που λύνουν τα παρακάτω προβλήματα:

Δίνεται μια ακολουθία αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 1$).

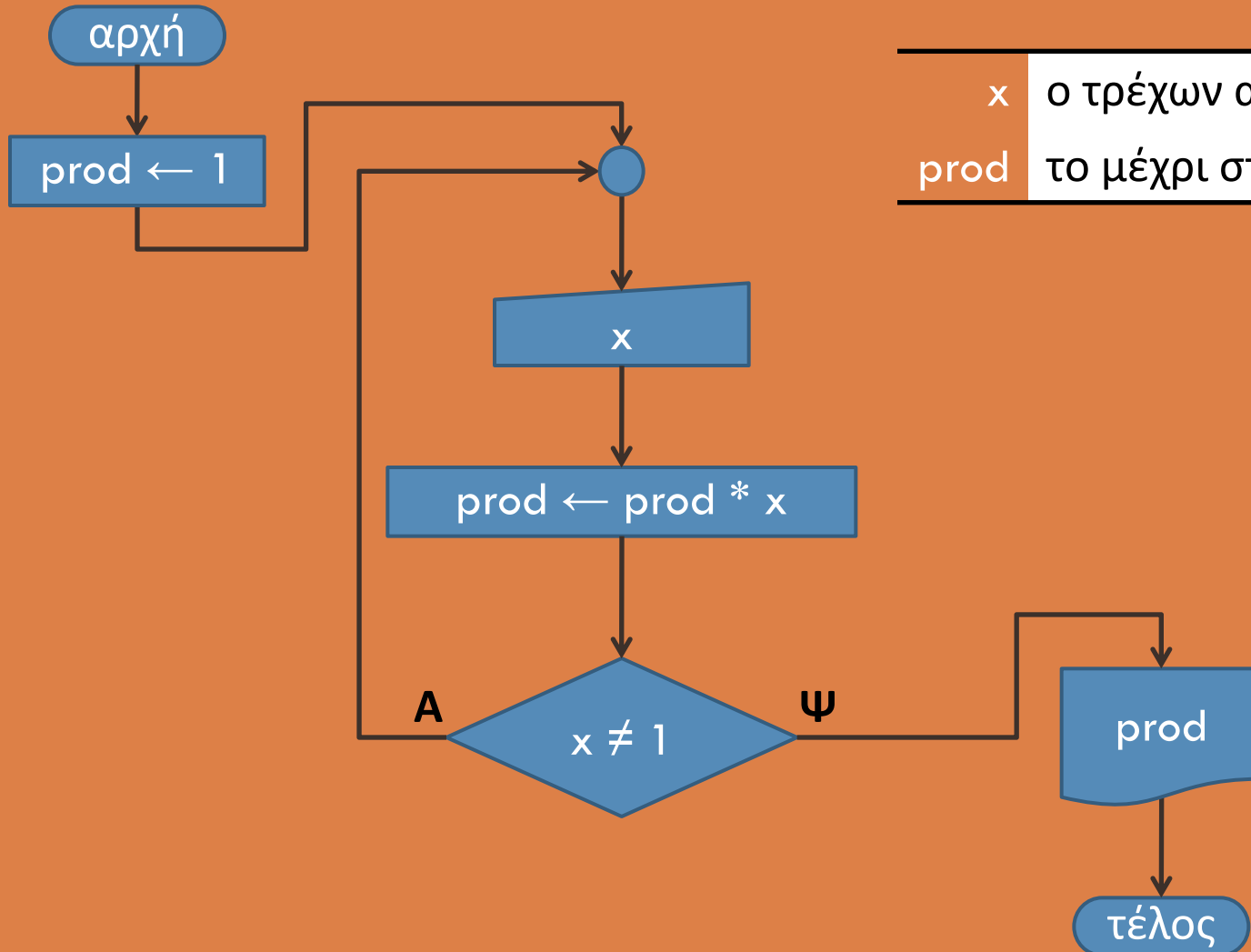
Να υπολογιστεί και να τυπωθεί στην οθόνη το γινόμενο τους, $x_1 * x_2 * \dots * x_N$.

Δίνεται μια ακολουθία αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 0$).

Να υπολογιστεί το πλήθος τους, N , και να τυπωθεί στην οθόνη.

Άσκηση 3: Διάγραμμα ροής

54



x ο τρέχων αριθμός

prod το μέχρι στιγμής γινόμενο

Άσκηση 3: Ψευδοκώδικας

55

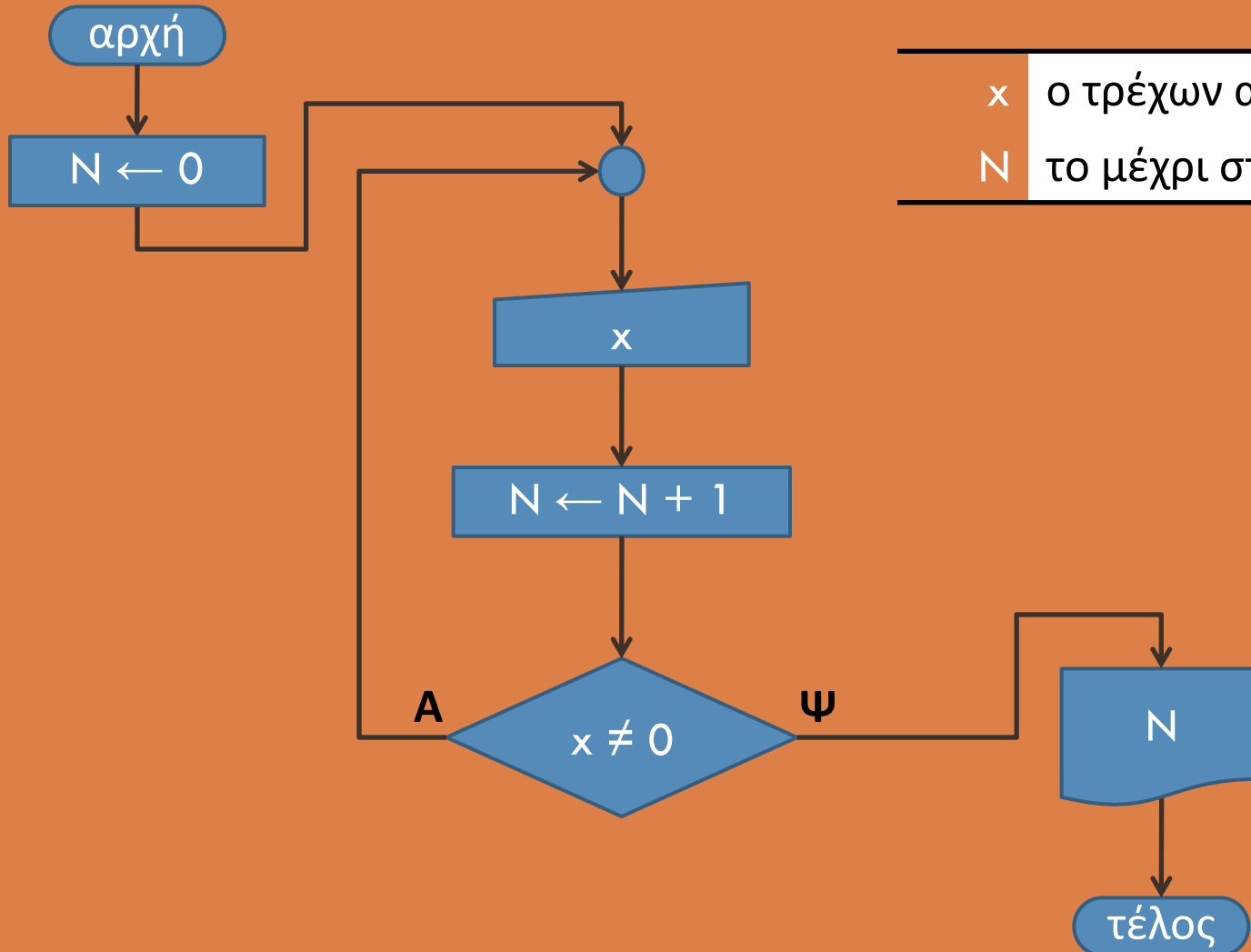
1. Αρχή.
2. $prod \leftarrow 1$.
3. Επανάληψη:
4. Διάβασε x .
5. $prod \leftarrow prod * x$.
6. όσο $x \neq 1$.
7. Τύπωσε $prod$.
8. Τέλος.

x ο τρέχων αριθμός

$prod$ το μέχρι στιγμής γινόμενο

Άσκηση 4: Διάγραμμα ροής

56



x ο τρέχων αριθμός

N το μέχρι στιγμής πλήθος

Άσκηση 4: Ψευδοκώδικας

57

1. Αρχή.
2. $N \leftarrow 0$.
3. Επανάληψη:
4. Διάβασε x .
5. $N \leftarrow N + 1$.
6. όσο $x \neq 0$.
7. Τύπωσε N .
8. Τέλος.

x ο τρέχων αριθμός

N το μέχρι στιγμής πλήθος

Πρόβλημα 5

58

Να γραφεί πρόγραμμα που να δέχεται μια απροσδιόριστα μεγάλη ακολουθία αριθμών (η οποία τερματίζει με τον αριθμό 0) και να υπολογίζει τον **μέσο όρο** τους.

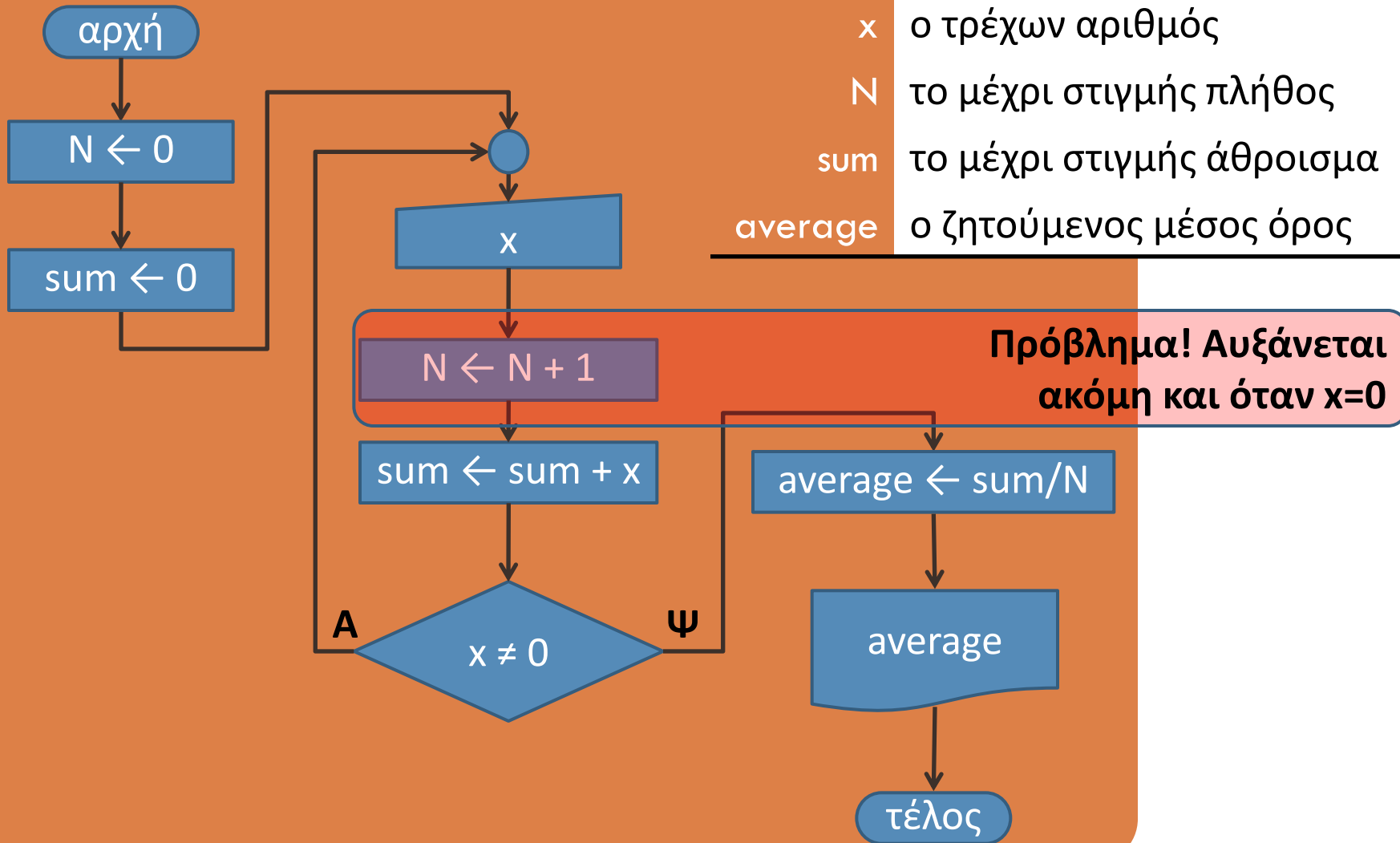
Μια πιο σαφής περιγραφή:

Δίνεται μια ακολουθία αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 0$).

Να υπολογιστεί και να τυπωθεί στην οθόνη ο μέσος όρος τους, $(x_1 + x_2 + \dots + x_{N-1})/N-1$.

Πρόβλημα 5: Διάγραμμα ροής

59



Πρόβλημα 5: Ψευδοκώδικας

60

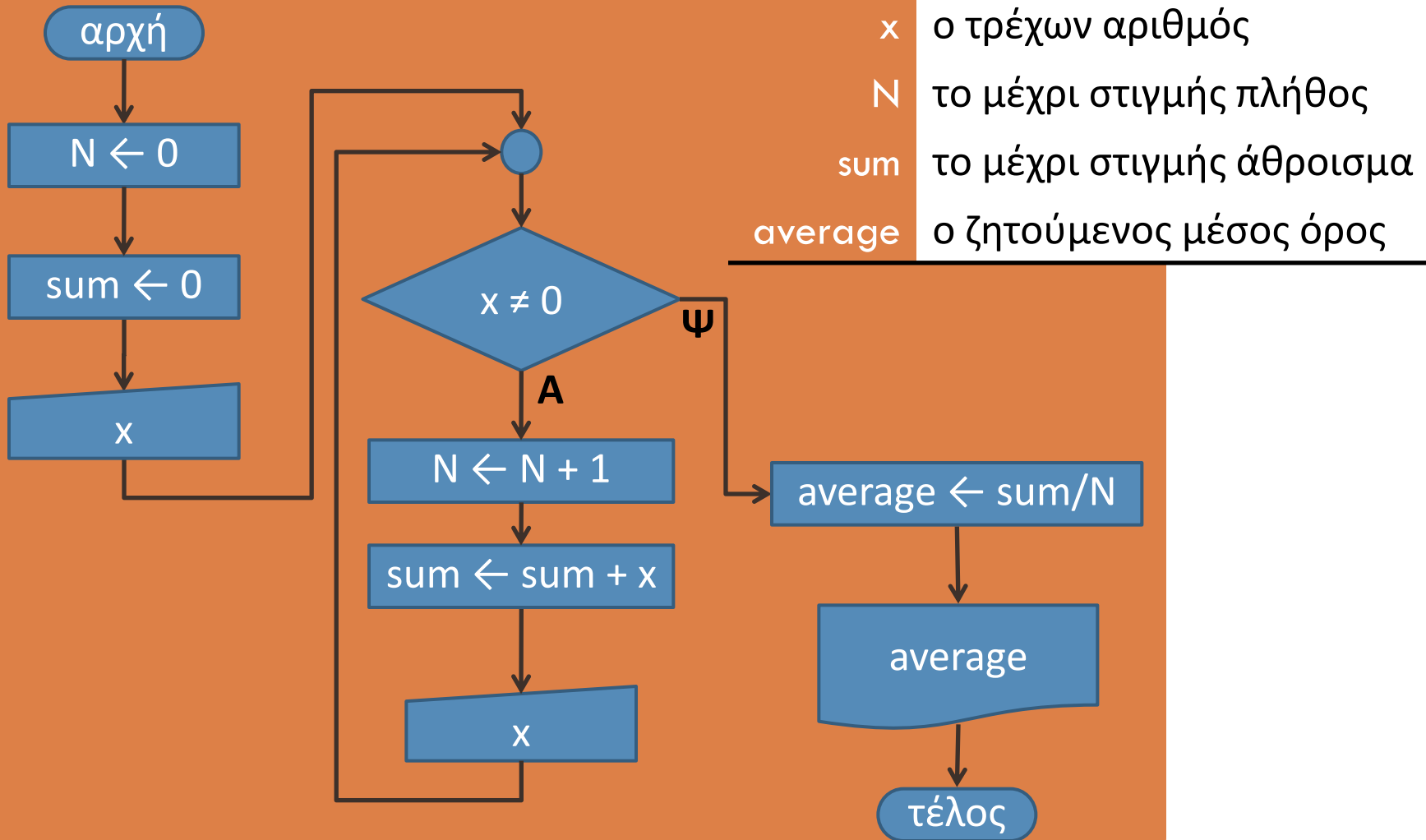
1. Αρχή.
2. $N \leftarrow 0$.
3. $sum \leftarrow 0$.
4. Επανάληψη:
5. Διάβασε x .
6. $N \leftarrow N + 1$.
7. $sum \leftarrow sum + x$.
8. όσο $x \neq 0$.
9. $average \leftarrow sum / N$.
10. Τύπωσε $average$.
11. Τέλος.

x	ο τρέχων αριθμός
N	το μέχρι στιγμής πλήθος
sum	το μέχρι στιγμής άθροισμα
$average$	ο ζητούμενος μέσος όρος

Πρόβλημα! Αυξάνεται ακόμη και όταν $x=0$

Πρόβλημα 5: Διάγραμμα ροής

61



Πρόβλημα 5: Ψευδοκώδικας

62

1. Αρχή
2. $N \leftarrow 0$
3. $sum \leftarrow 0$
4. Διάβασε x
5. Όσο $x \neq 0$ επανέλαβε:
 6. $N \leftarrow N + 1$
 7. $sum \leftarrow sum + x$
 8. Διάβασε x
9. όσο $x \neq 0$
10. $average \leftarrow sum / N$
11. Τύπωσε $average$
12. Τέλος

x	ο τρέχων αριθμός
N	το μέχρι στιγμής πλήθος
sum	το μέχρι στιγμής άθροισμα
$average$	ο ζητούμενος μέσος όρος

Πρόβλημα 6

63

Να γραφεί πρόγραμμα που να δέχεται μια απροσδιόριστα μεγάλη ακολουθία φυσικών αριθμών (που τερματίζει με τον αριθμό 0) και να υπολογίζει τον **μέγιστο** αυτών.

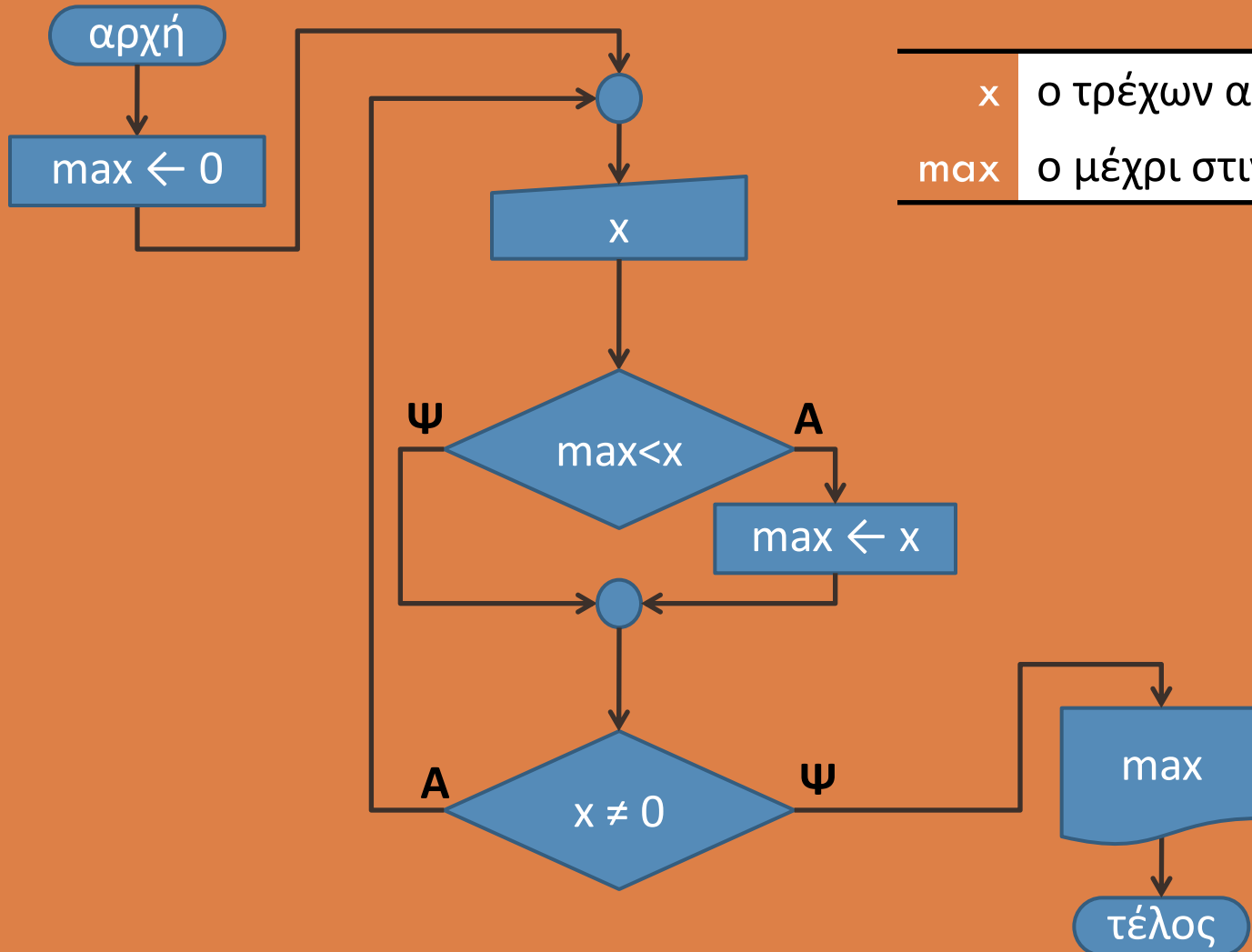
Μια πιο σαφής περιγραφή:

Δίνεται μια ακολουθία φυσικών αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 0$).

Να υπολογιστεί ο μέγιστος αυτών, $\max(x_1, x_2, \dots, x_N)$, και να τυπωθεί στην οθόνη.

Πρόβλημα 6: Διάγραμμα ροής

64



x ο τρέχων αριθμός

max ο μέχρι στιγμής μέγιστος

Πρόβλημα 6: Ψευδοκώδικας

65

1. Αρχή.
2. $max \leftarrow 0$.
3. Επανάληψη:
4. Διάβασε x .
5. Αν $max < x$
6. τότε $max \leftarrow x$.
7. όσο $x \neq 0$.
8. Τύπωσε max .
9. Τέλος.

x ο τρέχων αριθμός

max ο μέχρι στιγμής μέγιστος

Ασκήσεις 5 & 6

66

Δώστε διάγραμμα ροής και ψευδοκώδικα για αλγορίθμους που λύνουν τα παρακάτω προβλήματα:

Δίνεται μια ακολουθία αριθμών x_1, \dots, x_N στο $[0..100]$ από το πληκτρολόγιο (σήμα τέλους: $x_N = 101$).

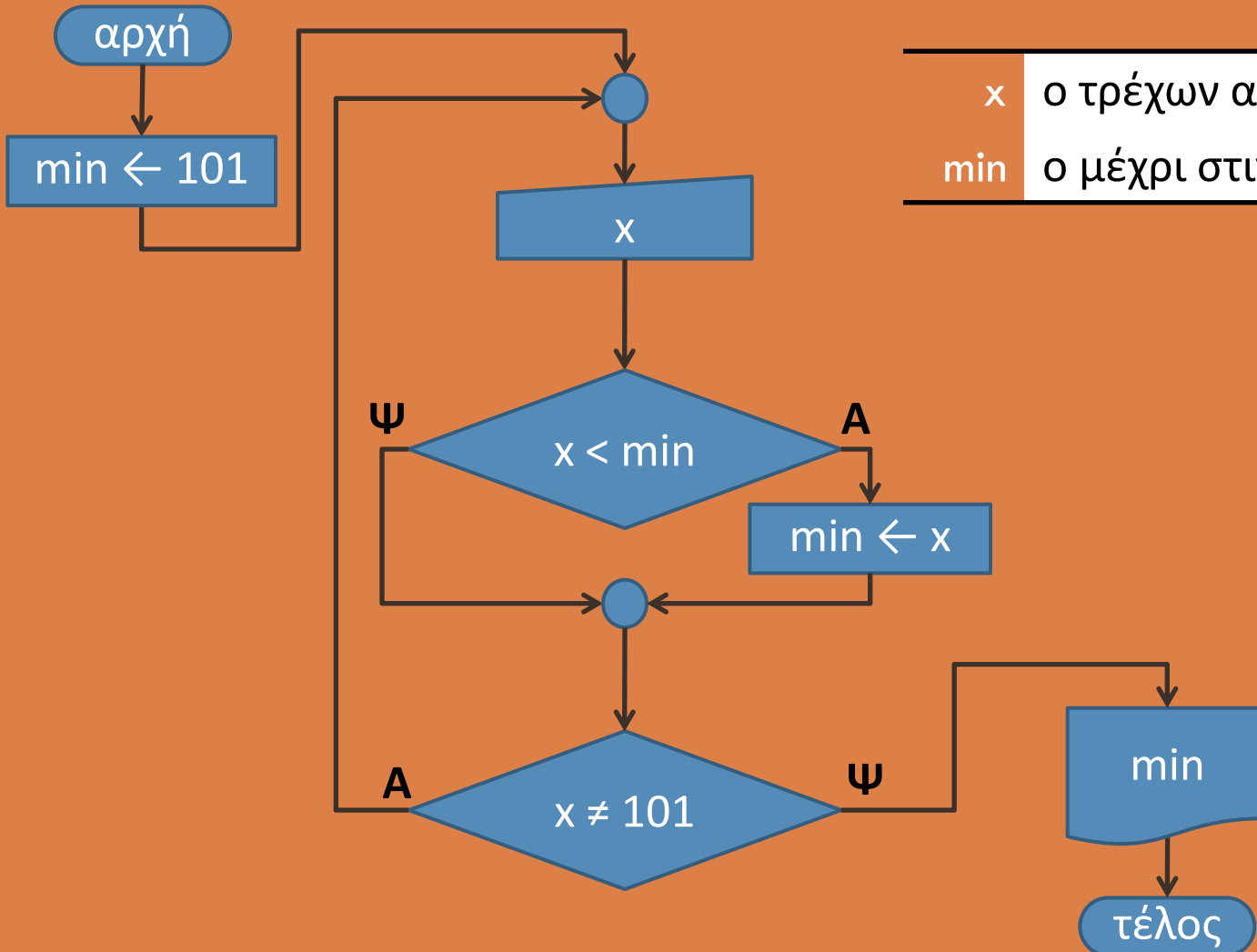
Να υπολογιστεί ο ελάχιστος αυτών, $\min(x_1, \dots, x_N)$, και να τυπωθεί στην οθόνη.

Δίνεται μια ακολουθία φυσικών αριθμών x_1, x_2, \dots, x_N , από το πληκτρολόγιο (σήμα τέλους: $x_N = 0$).

Να υπολογιστεί το πλήθος, ο μέσος όρος, και ο μέγιστος αυτών, και να τυπωθούν στην οθόνη.

Άσκηση 5: Διάγραμμα ροής

67



x ο τρέχων αριθμός

min ο μέχρι στιγμής ελάχιστος

Άσκηση 5: Ψευδοκώδικας

68

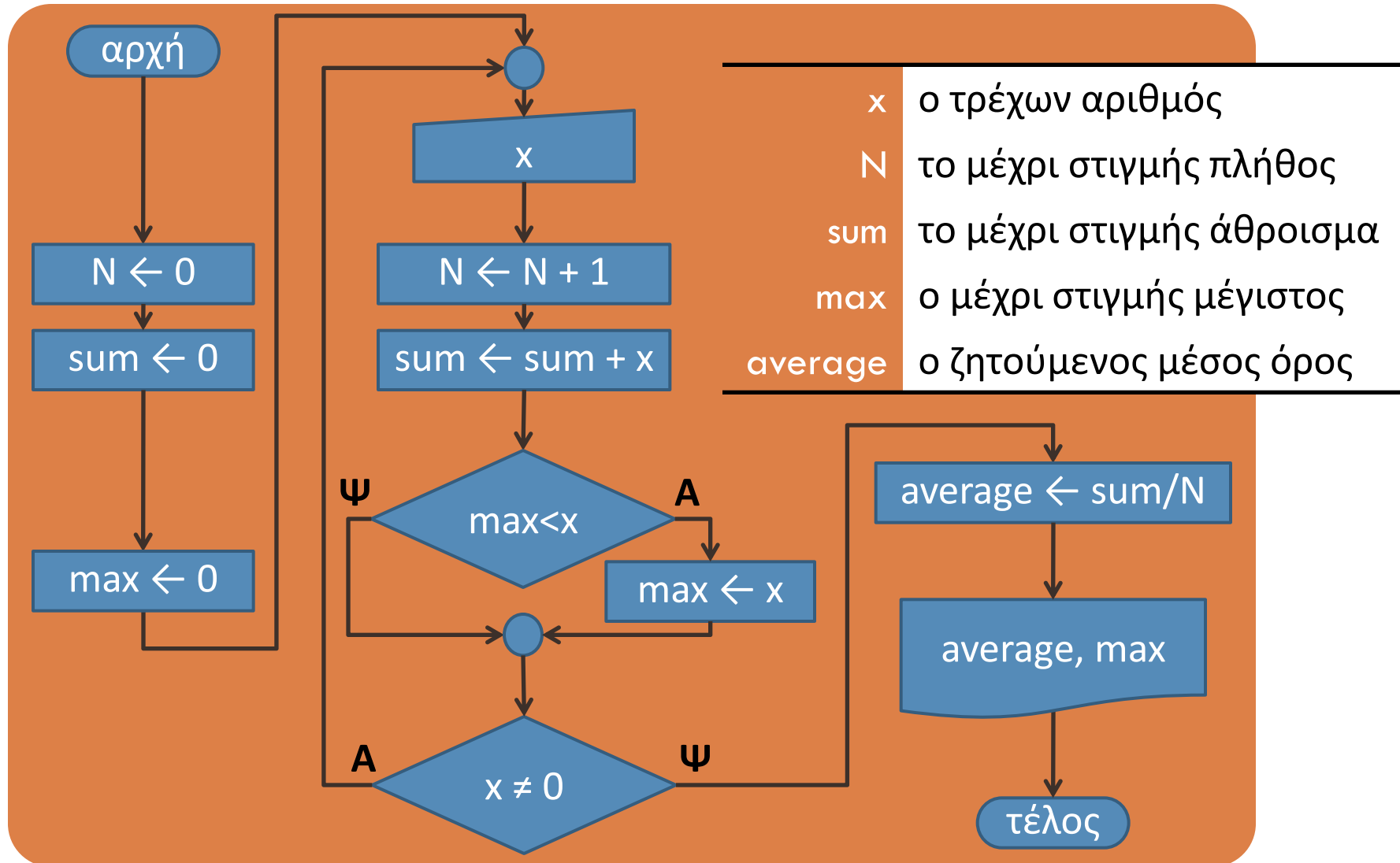
1. Αρχή.
2. $min \leftarrow 101$.
3. Επανάληψη:
4. Διάβασε x .
5. Αν $x < min$
6. τότε $min \leftarrow x$.
7. όσο $x \neq 101$.
8. Τύπωσε min .
9. Τέλος.

x ο τρέχων αριθμός

min ο μέχρι στιγμής ελάχιστος

Άσκηση 6: Διάγραμμα ροής

69



Άσκηση 6: Ψευδοκώδικας

70

1. Αρχή.
2. $N \leftarrow 0$.
3. $sum \leftarrow 0$.
4. $max \leftarrow 0$.
5. Επανάληψη:
 6. Διάβασε x .
 7. $N \leftarrow N + 1$.
 8. $sum \leftarrow sum + x$.
 9. Αν $max < x$
 10. τότε $max \leftarrow x$.
 11. όσο $x \neq 0$.
 12. $average \leftarrow sum / N$.
 13. Τύπωσε $average, max$.
 14. Τέλος.

x	ο τρέχων αριθμός
N	το μέχρι στιγμής πλήθος
sum	το μέχρι στιγμής άθροισμα
max	ο μέχρι στιγμής μέγιστος
$average$	ο ζητούμενος μέσος όρος

Πρόβλημα 7

71

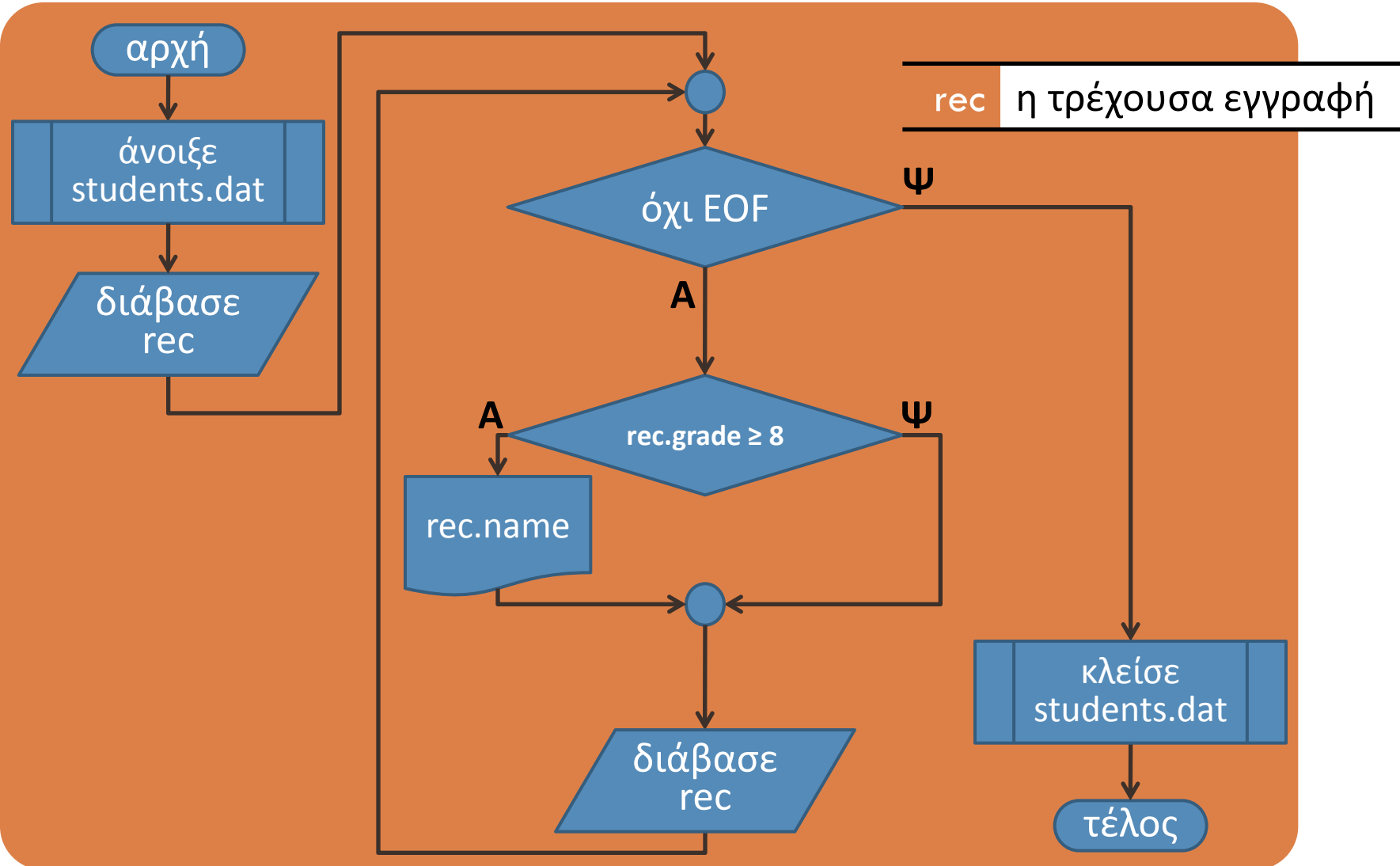
Τα στοιχεία των φοιτητών του ΕΠΛ00x είναι αποθηκευμένα σε αρχείο με όνομα `students.dat`, ως εξής:

id	name	year	grade
123456	Τάκης Χατζητάκης	1	6.5
121212	Στέλλα Μαυροπούλου	2	9
552244	Πέτρος Πετρόπουλος	1	7
...

Να γραφεί πρόγραμμα που διαβάζει το αρχείο και τυπώνει στην οθόνη τα ονόματα των φοιτητών που αρίστευσαν (δηλ. έλαβαν βαθμό ≥ 8).

Πρόβλημα 7: Διάγραμμα ροής

72



Πρόβλημα 7: Ψευδοκώδικας

73

`rec` η τρέχουσα εγγραφή

1. Αρχή.
2. Άνοιξε `students.dat` .
3. Διάβασε `rec` .
4. Όσο όχι EOF επανέλαβε:
5. Αν `rec.grade` ≥ 8
6. τότε Τύπωσε `rec.name` .
7. Διάβασε `rec` .
8. Κλείσε `students.dat` .
9. Τέλος.

Πρόβλημα 8

74

Τα στοιχεία των φοιτητών του ΕΠΛ00x είναι αποθηκευμένα σε αρχείο με όνομα `students.dat`, ως εξής:

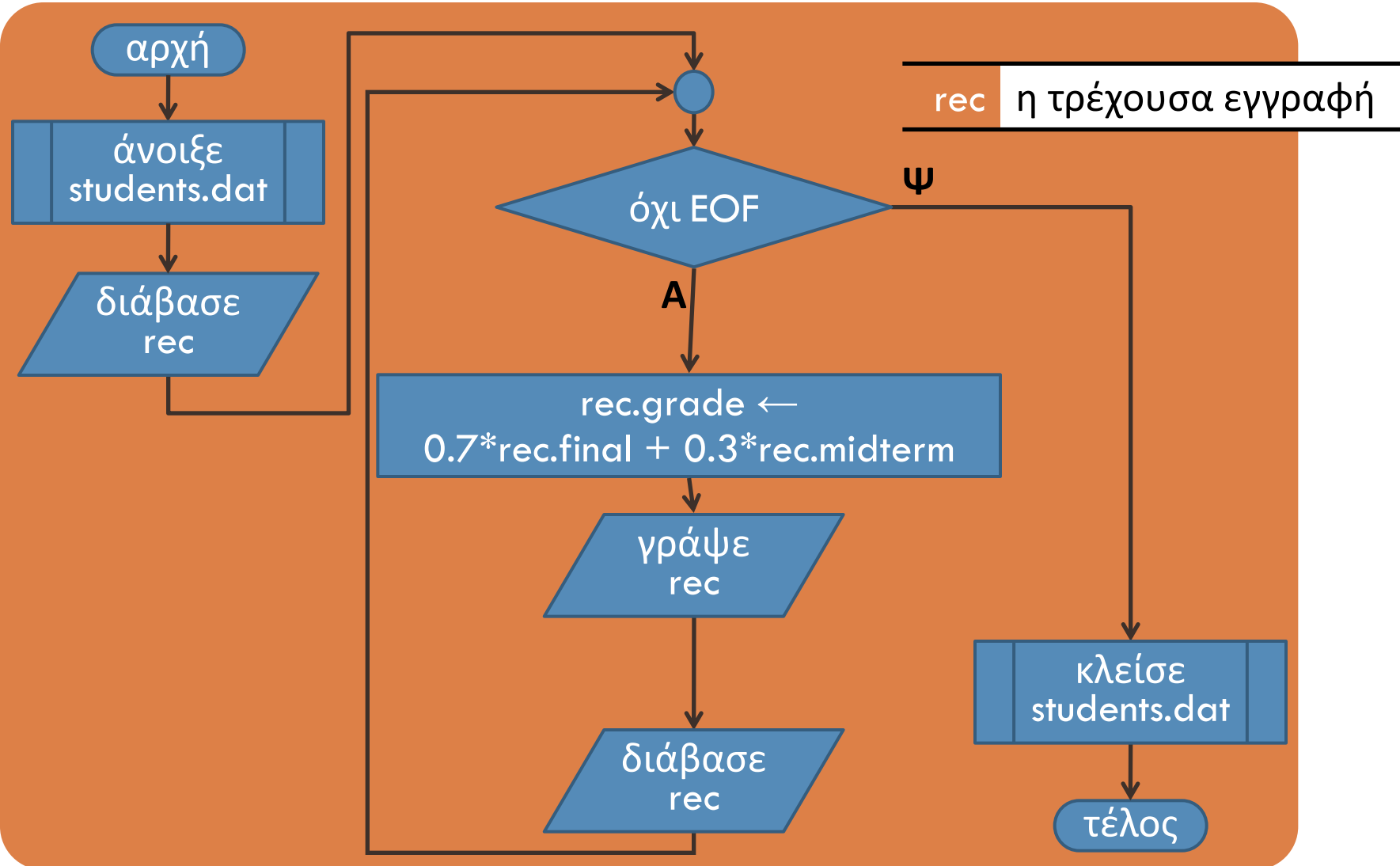
id	name	grade	final	midterm
123456	Τάκης Χατζητάκης		6.5	5
121212	Στέλλα Μαυροπούλου		9	7
552244	Πέτρος Πετρόπουλος		7	8
...

Να γραφεί πρόγραμμα που διαβάζει το αρχείο και ενημερώνει το πεδίο `grade` κάθε φοιτητή με τον βαθμό που προκύπτει από την ενδιάμεση και την τελική εξέταση, σύμφωνα με τον τύπο:

$$\text{grade} = 0.7 * \text{final} + 0.3 * \text{midterm}$$

Πρόβλημα 8: Διάγραμμα ροής

75



Πρόβλημα 8: Ψευδοκώδικας

76

`rec` η τρέχουσα εγγραφή

1. Αρχή.
2. Άνοιξε `students.dat` .
3. Διάβασε `rec` .
4. Όσο όχι EOF επανέλαβε:
5. $rec.grade \leftarrow 0.7 * rec.final + 0.3 * rec.midterm$.
6. Γράψε `rec` .
7. Διάβασε `rec` .
8. Κλείσε `students.dat` .
9. Τέλος.

Άσκηση 7

77

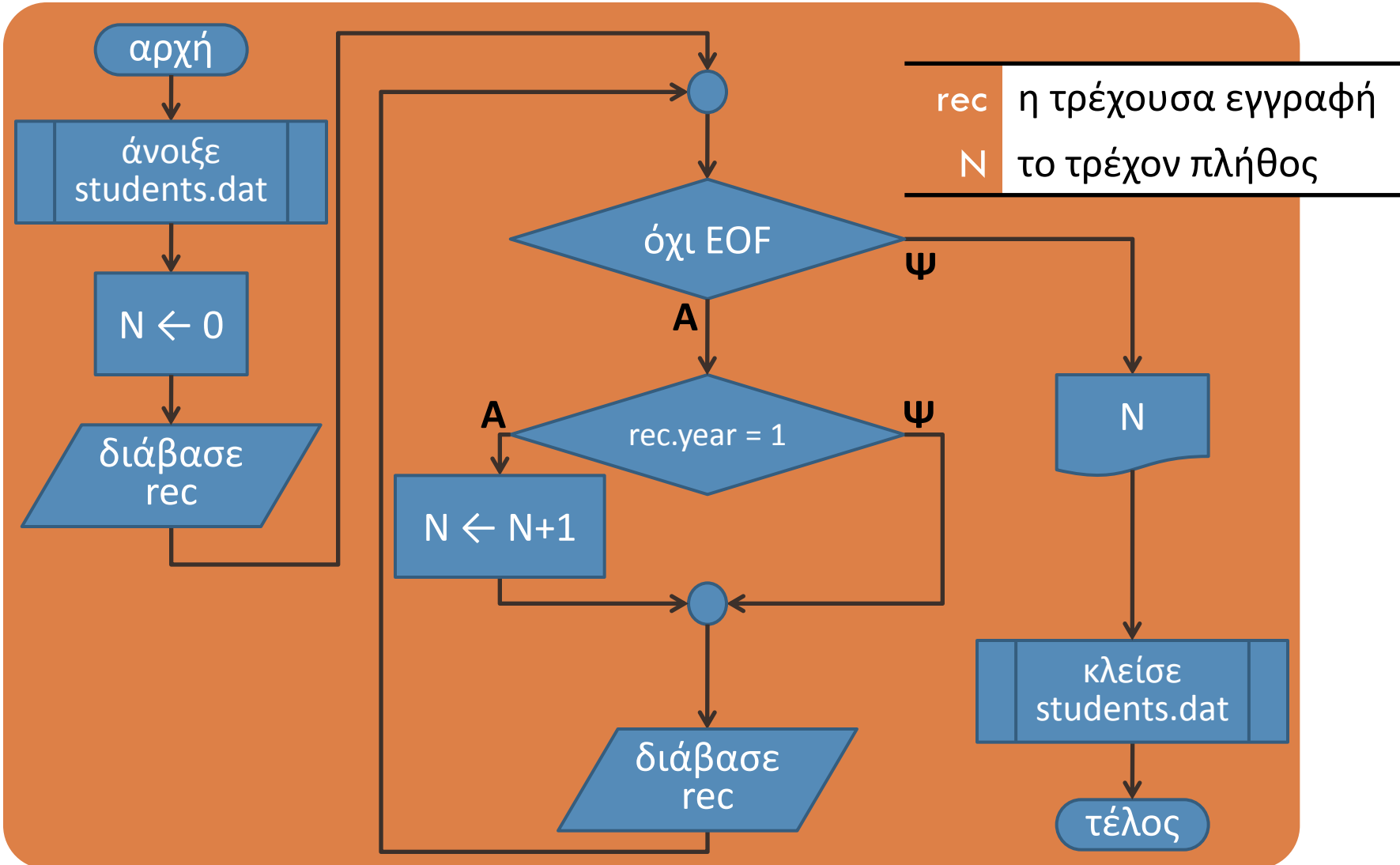
Τα στοιχεία των φοιτητών του ΕΠΛ00x είναι αποθηκευμένα σε αρχείο με όνομα `students.dat`, ως εξής:

id	name	year	grade
123456	Τάκης Χατζητάκης	1	6.5
121212	Στέλλα Μαυροπούλου	2	9
552244	Πέτρος Πετρόπουλος	1	7
...

Να γραφεί πρόγραμμα που διαβάζει το αρχείο, υπολογίζει το **πλήθος** των φοιτητών που βρίσκονται στο πρώτο έτος των σπουδών τους (δηλ. `year = 1`), και τυπώνει στην οθόνη αυτό το πλήθος.

Άσκηση 7: Διάγραμμα ροής

78



Άσκηση 7: Ψευδοκώδικας

79

1. Αρχή.
2. Άνοιξε *students.dat* .
3. $N \leftarrow 0$.
4. Διάβασε *rec* .
5. Όσο όχι EOF επανέλαβε:
6. Αν *rec.year* = 1
7. τότε $N \leftarrow N + 1$.
8. Διάβασε *rec* .
9. Τύπωσε *N* .
10. Κλείσε *students.dat* .
11. Τέλος.

<i>rec</i>	η τρέχουσα εγγραφή
<i>N</i>	το τρέχον πλήθος

Άσκηση 8

80

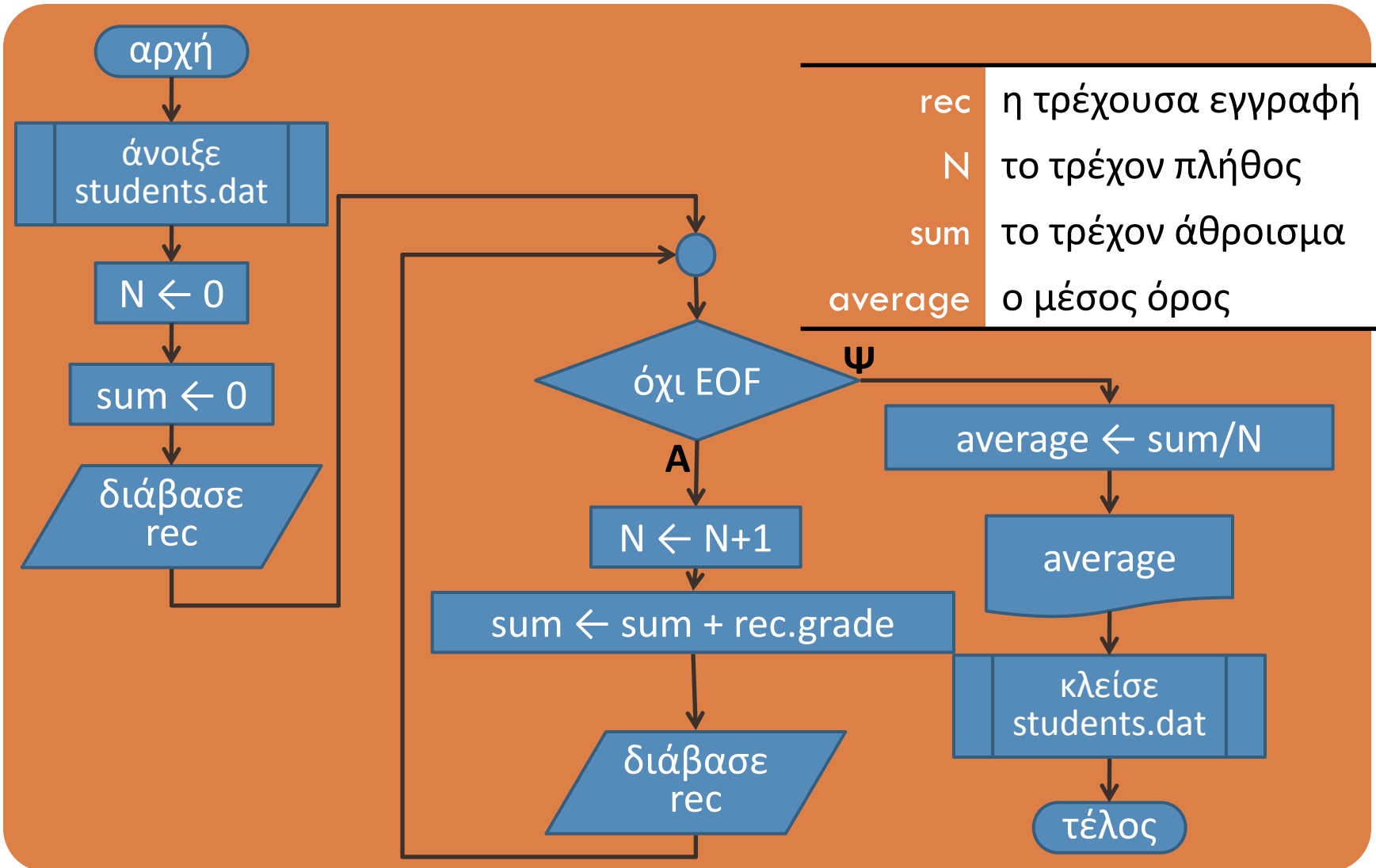
Τα στοιχεία των φοιτητών του ΕΠΛ00χ είναι αποθηκευμένα σε αρχείο με όνομα `students.dat`, ως εξής:

id	name	year	grade
123456	Τάκης Χατζητάκης	1	6.5
121212	Στέλλα Μαυροπούλου	2	9
552244	Πέτρος Πετρόπουλος	1	7
...

Να γραφεί πρόγραμμα που διαβάζει το αρχείο, υπολογίζει τον **μέσο όρο** των βαθμών όλων των φοιτητών, και τυπώνει στην οθόνη αυτόν τον μέσο όρο.

Άσκηση 8: Διάγραμμα ροής

81



Άσκηση 8: Ψευδοκώδικας

82

1. Αρχή.
2. Άνοιξε *students.dat* .
3. $N \leftarrow 0$.
4. $sum \leftarrow 0$.
5. Διάβασε *rec* .
6. Όσο όχι EOF επανέλαβε:
7. $N \leftarrow N + 1$.
8. $sum \leftarrow sum + rec.grade$.
9. Διάβασε *rec* .
10. $average \leftarrow sum / N$.
11. Τύπωσε *average* .
12. Κλείσε *students.dat* .
13. Τέλος.

<i>rec</i>	η τρέχουσα εγγραφή
<i>N</i>	το τρέχον πλήθος
<i>sum</i>	το τρέχον άθροισμα
<i>average</i>	ο μέσος όρος