



Κεφάλαιο 2

Βασικές Έννοιες Αλγορίθμων

2.1 Τι είναι αλγόριθμος

- Η λέξη αλγόριθμος προέρχεται από μελέτη του Πέρση μαθηματικού Abu Ja'far Mohammed ibn al Khowarizmi
- Στα λατινικά ξεκινούσε με τη φράση Algoritmi dixit που σημαίνει «ο Αλγόριθμος λέει»
- Για χρόνια ο όρος αλγόριθμος σήμαινε κάτι σαν «συστηματική διαδικασία αριθμητικών χειρισμών»
- Τη σημερινή του έννοια πήρε στον 20^ο αιώνα



Τι είναι αλγόριθμος

Αλγόριθμος είναι

μια πεπερασμένη σειρά ενεργειών,

αυστηρά καθορισμένων και

εκτελέσιμων σε πεπερασμένο χρόνο, που

στοχεύουν στην επίλυση ενός προβλήματος

Για να είναι αλγόριθμος ...

- **Είσοδος**
 - Καμία, μία ή περισσότερες τιμές δεδομένων για είσοδο
- **Έξοδος**
 - Τουλάχιστον ένα αποτέλεσμα
- **Καθοριστικότητα**
 - Σαφώς καθορισμένες εντολές
- **Περατότητα**
 - Να τελειώνει μετά από πεπερασμένο αριθμό βημάτων
- **Αποτελεσματικότητα**
 - Απλές και εκτελέσιμες εντολές

Καθοριστικότητα

- Κάθε βήμα να είναι διατυπωμένο με σαφήνεια
- Δεν υπάρχουν αμφιβολίες για το τι πρέπει να κάνουμε.
 - Διαίρεση: Πρέπει ο αλγόριθμος της διαίρεσης να προβλέπει την περίπτωση διαίρεσης με το 0
 - Σε μια συνταγή, η έκφραση «Βάλε λίγο αλάτι» δεν είναι σαφώς καθορισμένη. Τι πάει να πει «λίγο»;

Αποτελεσματικότητα

- Κάθε εντολή είναι αρκετά απλή έτσι ώστε να μπορεί να εκτελεστεί από τον υπολογιστή.
- Για κάθε βήμα του αλγορίθμου θα πρέπει να υπάρχει αντίστοιχη εντολή στη γλώσσα προγραμματισμού έτσι ώστε να μπορεί ο αλγόριθμος να μετατραπεί σε εκτελέσιμο πρόγραμμα.
 - ▣ Η εντολή «Βρες το μεγαλύτερο από 100 αριθμούς» δεν υπάρχει. Πρέπει να αναλυθεί σε πολλές απλούστερες εντολές δηλ. Πάρε με τη σειρά κάθε αριθμό, έλεγξε αν αυτός είναι ο μεγαλύτερος και αφού τελειώσουν εμφάνισε τον.

Σχετικά με Καθοριστικότητα & Αποτελεσματικότητα

Η **παραβίαση** οποιουδήποτε από τα δύο κριτήρια οδηγεί σε **μη εκτελέσιμους** αλγορίθμους

- Παραβίαση της καθοριστικότητας σημαίνει έλλειψη σαφήνειας
- Παραβίαση της αποτελεσματικότητας σημαίνει πολύπλοκες εντολές που δεν υπάρχουν στη γλώσσα προγραμματισμού

2.2 ΣΠΟΥΔΑΙΟΤΗΤΑ ΑΛΓΟΡΙΘΜΩΝ

Η Πληροφορική μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές :

Του υλικού

Επηρεάζεται η εκτέλεση ενός προγράμματος από την απόδοση ή την αρχιτεκτονική του υλικού;

Των γλωσσών προγραμματισμού

Αν κάποια γλώσσα μπορεί να υποστηρίξει τον αλγόριθμο, ή χρειάζονται περισσότερες εντολές

Θεωρητική

Αν υπάρχει κάποιος αποδοτικός αλγόριθμος για τη λύση ενός προβλήματος

Αναλυτική

Μελετώνται οι πόροι που απαιτούνται από έναν αλγόριθμο (μνήμη, χρόνος, είσοδος/έξοδος, κλπ)

2.4 Περιγραφή και αναπαράσταση αλγορίθμων

Παραβίαση της αποτελεσματικότητας σημαίνει πολύπλοκες εντολές που δεν υπάρχουν στη γλώσσα προγραμματισμού

- Ελεύθερο κείμενο
 - Ανεπεξέργαστος και αδόμητος τρόπος, **ενδέχεται να παραβιάζει το κριτήριο της αποτελεσματικότητας.**
- Διαγραμματικές τεχνικές
 - Γραφικός τρόπος (π.χ. διάγραμμα ροής).
- Φυσική γλώσσα
 - Περιγραφή με φυσική γλώσσα κατά βήματα.
Ενδέχεται να παραβιάζει το κριτήριο της καθοριστικότητας
- Κωδικοποίηση
 - Πρόγραμμα γραμμένο σε ψευδογλώσσα ή σε κάποια γλώσσα προγραμματισμού. Το πρόγραμμα δίνει τα ίδια αποτελέσματα με τον αλγόριθμο

Παραβίαση της καθοριστικότητας σαφήνειας

ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ ΕΛΕΥΘΕΡΟ ΚΕΙΜΕΝΟ

Λύση Δευτεροβάθμιας εξίσωσης: $ax^2+bx +c=0$

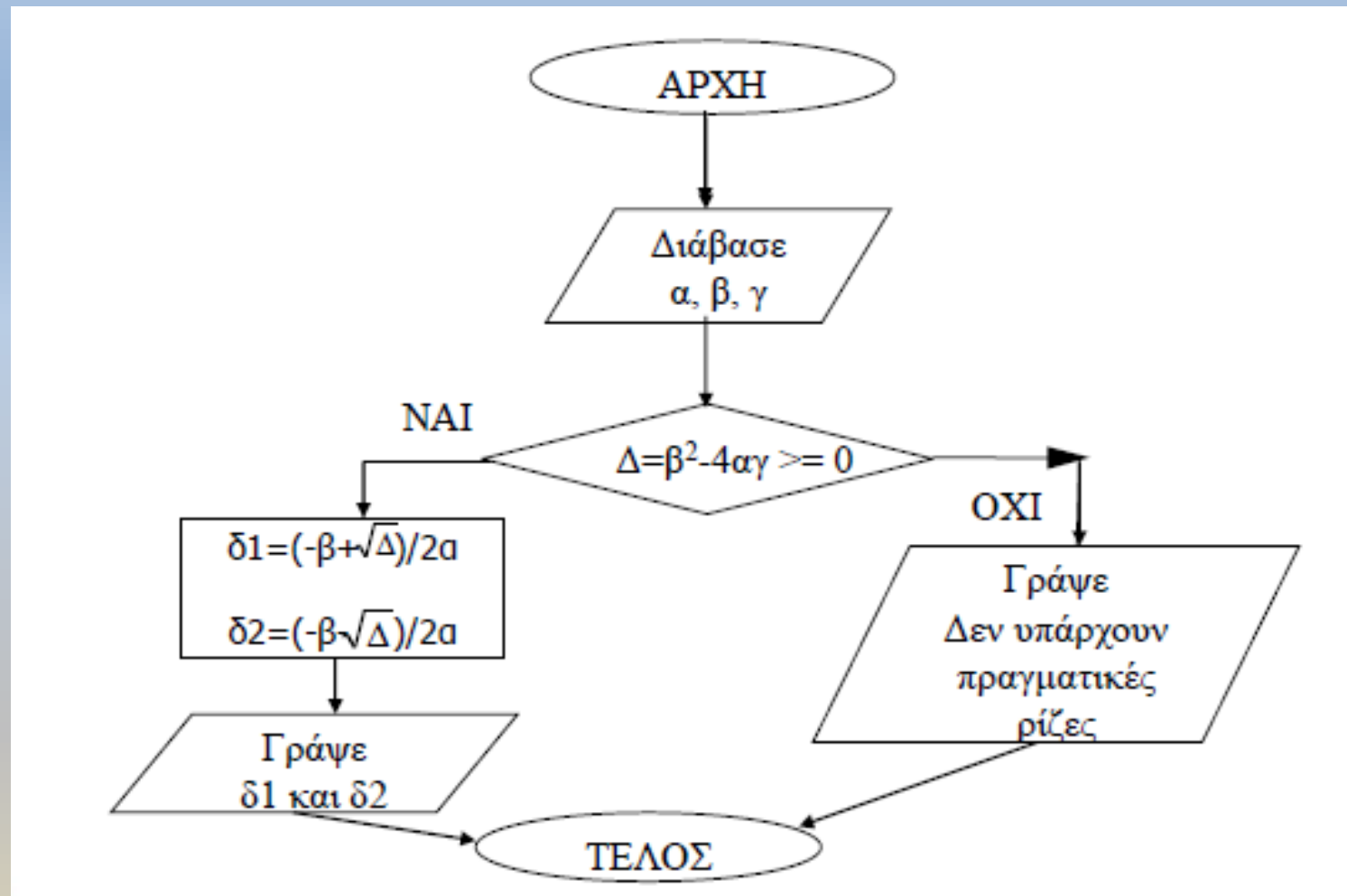
Ζητούμε να μας δοθούν οι τιμές των συντελεστών a , b , και c της εξίσωσης. Προχωρούμε σε υπολογισμό της διακρίνουσας $\Delta=b^2-4ac$. Σε περίπτωση που ισχύει ότι η διακρίνουσα Δ είναι μεγαλύτερη ή ίση του μηδενός οι δυο πραγματικές ρίζες της εξίσωσης δίνονται από τη σχέση , $\frac{-b \pm \sqrt{\Delta}}{2a}$ ενώ αν η διακρίνουσα Δ είναι μικρότερη του μηδενός, τότε ισχύει ότι η εξίσωση δεν έχει πραγματικές ρίζες.

ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ ΦΥΣΙΚΗ ΓΛΩΣΣΑ ΜΕ ΒΗΜΑΤΑ

Λύση Δευτεροβάθμιας εξίσωσης: $ax^2+bx+c=0$.

- Δίνονται οι συντελεστές a, b, c
- Υπολογίζεται η Διακρίνουσα $\Delta=b^2-4ac$
- Αν $\Delta>0$ υπολογίζονται και εμφανίζονται οι 2 πραγματικές ρίζες από τη σχέση $\frac{-b \pm \sqrt{\Delta}}{2a}$
- Αν $\Delta=0$ υπολογίζεται και εμφανίζεται η διπλή πραγματική ρίζα από τη σχέση $\frac{-b}{2a}$
- Αν $\Delta<0$ ενημερώνουμε ότι δεν υπάρχουν πραγματικές ρίζες.

ΑΝΑΠΑΡΑΣΤΑΣΗ ΜΕ ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ



ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

```
#include<stdio.h>
#include<math.h>
void main()
{ float a,b,c,D,d1,d2;
  scanf("%f \n %f \n %f", &a, &b,&c);
  D=b*b-4*a*c;
  d1=(-b+sqrt(D))/(2*a);
  d2=(-b-sqrt(D))/(2*a);
  if (D >= 0)
    printf("The equation has real roots:\n %f \n %f",d1, d2);
  else
    printf("The equation have not real roots");
}
```

ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ ΕΛΕΥΘΕΡΟ ΚΕΙΜΕΝΟ

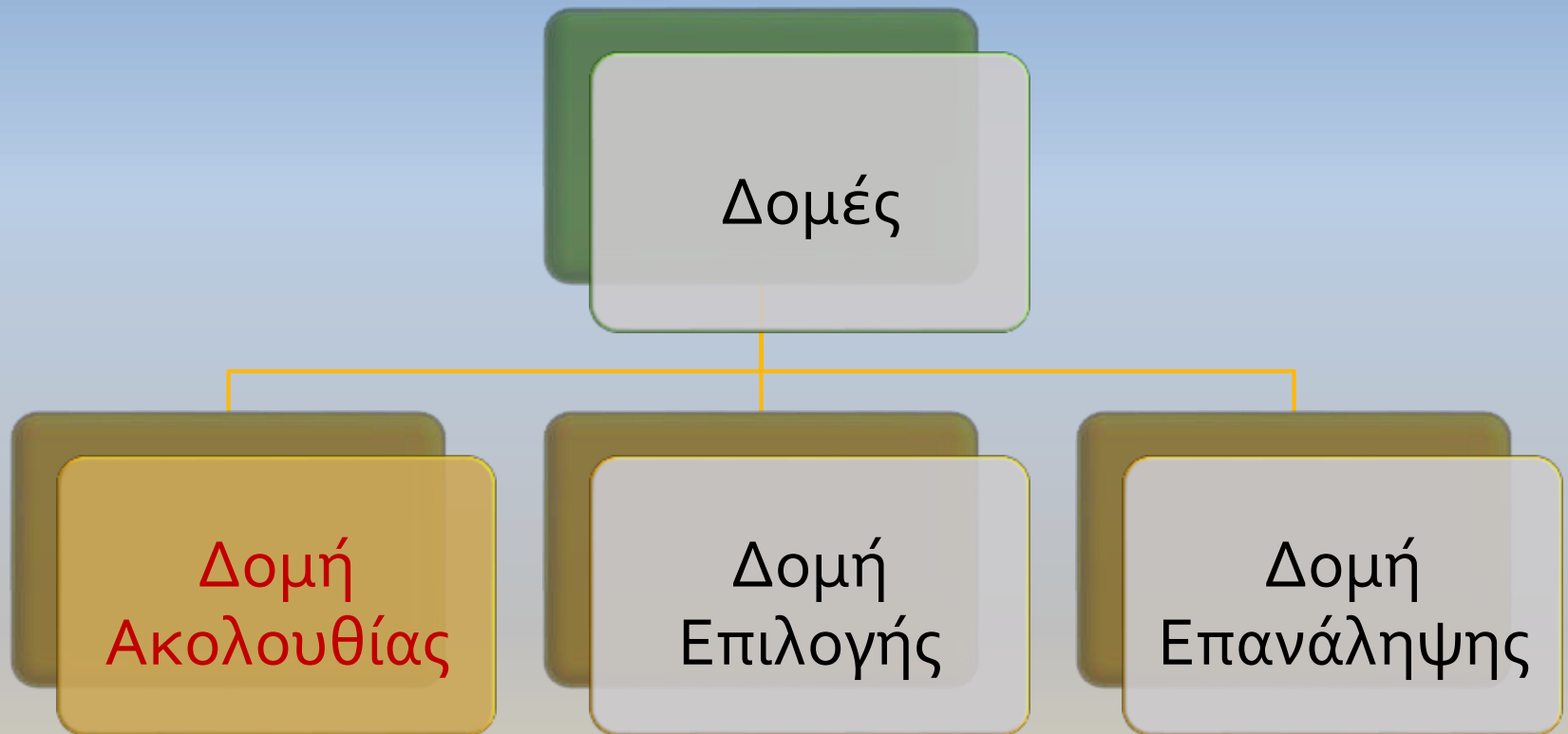
Λύση Δευτεροβάθμιας εξίσωσης: $ax^2+bx +c=0$

Ζητούμε να μας δοθούν οι τιμές των συντελεστών a , b , και c της εξίσωσης. Προχωρούμε σε υπολογισμό της διακρίνουσας $\Delta=b^2-4ac$. Σε περίπτωση που ισχύει ότι η διακρίνουσα Δ είναι μεγαλύτερη ή ίση του μηδενός οι δυο πραγματικές ρίζες της εξίσωσης δίνονται από τη σχέση , $\frac{-b \pm \sqrt{\Delta}}{2a}$ ενώ αν η διακρίνουσα Δ είναι μικρότερη του μηδενός, τότε ισχύει ότι η εξίσωση δεν έχει πραγματικές ρίζες.

Διάγραμμα ροής

- Συμβολική γλώσσα
- Χρησιμοποιούνται γεωμετρικά σχήματα
- Τα γεωμετρικά σχήματα είναι σύμβολα
- Κάθε σύμβολο δηλώνει μια ενέργεια ή μια λειτουργία
- Τα σύμβολα έχουν περιεχόμενο όταν αυτό απαιτείται για να αποκτήσουν νόημα

Βασικές συνιστώσες/ εντολές αλγορίθμου





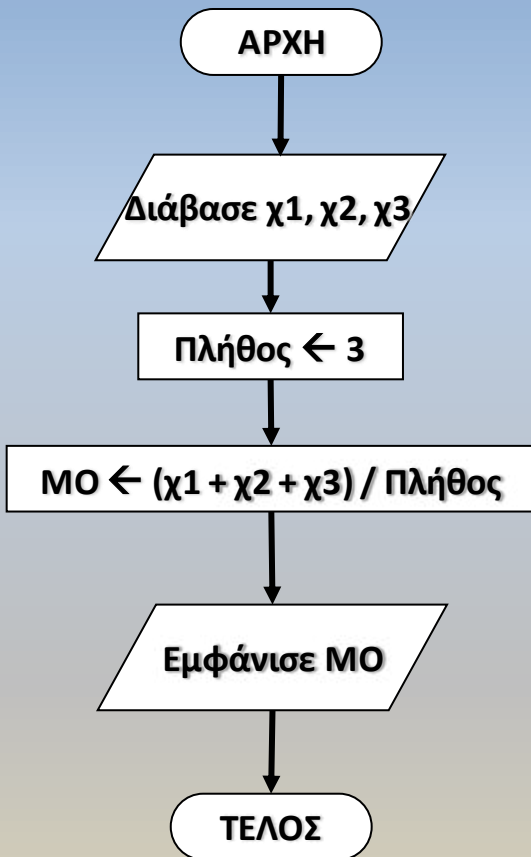
Δομή ακολουθίας

- Ακολουθιακή δομή εντολών (σειριακή εκτέλεση εντολών)
- Δεδομένη η σειρά εκτέλεσης
- Συνήθως απλά προβλήματα

4. ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ

Παράδειγμα 1

Με διάγραμμα
ροής



Με ψευδοκώδικα

Αλγόριθμος μέσος_όρος

Διάβασε χ1, χ2, χ3

Πλήθος ← 3

$ΜΟ \leftarrow (\chi_1 + \chi_2 + \chi_3) / \text{Πλήθος}$

Εμφάνισε ΜΟ

Τέλος μέσος_όρος

Με πρόγραμμα

ΠΡΟΓΡΑΜΜΑ μέσος_όρος
ΜΕΤΑΒΛΗΤΕΣ
 ΑΚΕΡΑΙΕΣ: χ1, χ2, χ3, Πλήθος
 ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΑΡΧΗ

ΔΙΑΒΑΣΕ χ1, χ2, χ3

 Πλήθος ← 3

$ΜΟ \leftarrow (\chi_1 + \chi_2 + \chi_3) / \text{Πλήθος}$

ΓΡΑΨΕ ΜΟ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Σταθερές

Προκαθορισμένες τιμές που μένουν σταθερές σε όλη τη διάρκεια εκτέλεσης του αλγορίθμου

Είδη Σταθερών

- Αριθμητικές 123 , 3,14, -1,2
- Αλφαριθμητικές «Όνομα», «Επώνυμο»
«2004»
- Λογικές Αληθής, Ψευδής

Μεταβλητές

- Γλωσσικό αντικείμενο που χρησιμοποιείται να παραστήσει ένα στοιχείο δεδομένου
- Μπορεί να μεταβάλλεται κατά τη διάρκεια εκτέλεσης του αλγορίθμου
- Είδη Μεταβλητών
 - Ακέραιες
 - Πραγματικές
 - Αλφαριθμητικές
 - Λογικές

Τελεστές

Σύμβολα που χρησιμοποιούνται για να δηλώνουν πράξεις

Αριθμητικοί

- $+$, $-$, $*$, $/$
- $\wedge$
- **Div**
- **Mod**

Συγκριτικοί

- $<=$, $>=$, $=$, $<>$, $>$, $<$

Λογικοί

- **και ή όχι**

Εκφράσεις

- Διαμορφώνονται από
 - Σταθερές, μεταβλητές
 - Τελεστές
 - Συναρτήσεις
 - Παρενθέσεις () *ΌΧΙ αγκύλες*
- Δίνω τιμές σε μεταβλητές
- Εκτελούνται πράξεις
 - $\Pi\chi$
 - $A \leftarrow B$ (οι A και B ίδιου τύπου δεδομένων)
 - $A \leftarrow 3$
 - $A \leftarrow 3+5$
 - $A \leftarrow B \text{ ή } \Gamma$
 - $A \leftarrow B > 3$

Ιεραρχία πράξεων

- $\wedge$
- $*, /, \text{div}, \text{mod}$
- $+, -$
- Οι πράξεις ίδιας προτεραιότητας εκτελούνται από αριστερά προς δεξιά
- Για να εξασφαλίσω την ορθότητα της σειράς εκτέλεσης πράξεων χρησιμοποιώ παρενθέσεις

Η εντολή Αν - Απλή Μορφή

1. Αν <συνθήκη>

τότε

2.

<ομάδα _εντολών_1 >

3. Τέλος_αν

Η εντολή Αν Απλή Μορφή

1. Διάβασε α, β
2. Αν $\alpha > \beta$ τότε
3. Γράψε 'το α είναι μεγαλύτερο'
4. Τέλος_αν

Η εντολή αν Απλή α

1. Διάβασε α,β
2. Αν $\alpha > \beta$ τότε
3. Γράψε 'το α είναι μεγαλύτερο'
4. Γράψε 'το β είναι μικρότερο'
5. Τέλος_αν

Η εντολή αν Απλή Μορφή

1. Διάβασε α,β
2. $\gamma \leftarrow \alpha > \beta$
3. Αν γ τότε
4. Γράψε 'το α είναι μεγαλύτερο'
5. Γράψε 'το β είναι μικρότερο'
6. Τέλος_αν

Η εντολή αν Σύνθετη Μορφή

1. **Αν** <συνθήκη> **τότε**
2. <ομάδα _εντολών_1 >
3. **αλλιώς**
4. <ομάδα _εντολών_2>
5. **Τέλος_αν**

Η εντολή αν Σύνθετη Μορφή

1. Διάβασε α,β
2. Αν $\alpha > \beta$ τότε
3. Γράψε 'το α είναι μεγαλύτερο'
4. Γράψε 'το β είναι μικρότερο'
5. αλλιώς
6. Γράψε 'το β είναι μεγαλύτερο'
7. Γράψε 'το α είναι μικρότερο'
8. Τέλος_αν

Η εντολή Αν Εμφωλευμένη Μορφή

Αν <συνθήκη1>

τότε

<ομάδα_εντολών_1>

αλλιώς

Τέλος_αν



Η εντολή Αν Εμφωλευμένη

Αν <συνθήκη1>

τότε

<ομάδα_εντολών_1>

αλλιώς

<ομάδα_εντολών_2>

Τέλος_αν



Βρείτε το μέγιστο μεταξύ
τριών αριθμών

```
1 Αλγόριθμος μεσος_ορος_επιλογη
2 διαβασε α,β,γ
3 μεγιστο ← α
4 Αν α>β τότε
5     αν γ>α τότε
6         μεγιστο ← γ
7     αλλιώς
8         μεγιστο ← α
9     τελος_αν
10    αλλιώς
11        αν β>γ τότε
12            μεγιστο ← β
13        αλλιώς
            αν γ>=β τότε
                μεγιστο ← γ
            Τέλος_αν
        Τέλος_αν
    Τέλος_αν

Εκτυπωσε 'Μεγιστο:', μεγιστο
Τέλος μεσος_ορος_επιλογη
```

Η εντολή αν Πολλαπλή Μορφή



```
Αν <συνθήκη1> τότε
    <ομάδα_εντολών_1>
αλλιώς_αν <συνθήκη2> τότε
    <ομάδα_εντολών_2>
....
αλλιώς_αν <συνθήκην> τότε
    <ομάδα_εντολών_ν>
αλλιώς
    <ομάδα_εντολών_ν+1>
Τέλος_αν
```

Η εντολή αν

1. Αλγόριθμος απλη_επιλογη
2. διαβασε α, β
3. Αν $\alpha > \beta$ τότε
4. μεγιστο $\leftarrow \alpha$
5. αλλιώς
6. μεγιστο $\leftarrow \beta$
7. Τέλος_αν
8. Εκτυπωσε μεγιστο
9. Τέλος απλη_επιλογη

Η εντολή Επίλεξε (Πολλαπλή Μορφή Επιλογής)

Άλλη μορφή πολλαπλής επιλογής

Επίλεξε μεταβλητή
Περίπτωση τιμή1 ή σύγκριση

.....

Περίπτωση τιμή2 ή σύγκριση

.....

Περίπτωση τιμή3 ή σύγκριση

.....

Περίπτωση αλλιώς

.....

Τέλος_επιλογών

Λειτουργεί ακριβώς όπως και
η πολλαπλή ΑΝ

Αλγόριθμος αποτελέσματα

Διάβασε ηλικία

Επίλεξε ηλικία

Περίπτωση < 1

Εμφάνισε «βρέφος»

Περίπτωση < 5

Εμφάνισε «νήπιο»

Περίπτωση < 12

Εμφάνισε «παιδί»

Περίπτωση < 19

Εμφάνισε «ανήλικος»

Περίπτωση < 65

Εμφάνισε «ενήλικας»

Περίπτωση αλλιώς

Εμφάνισε «ηλικιωμένος»

Τέλος_επιλογών

Τέλος αποτελέσματα



Διαβάζω 100 αριθμούς και
τους εμφανίζω. Πόσες
μεταβλητές χρειάζομαι;

1. Διαβασε α
2. Εμφανισε α
3. Διαβασε β
4. Εμφανισε β
5. Διαβασε γ
6. Εμφανισε γ
7.

199. Διαβασε γγγ
200. Εμφανισε γγγ

ΔΟΜΕΣ ΕΠΑΝΑΛΗΨΗΣ (ή Βρόχοι) (Τί είναι ;)

Οι δομές επανάληψης χρησιμοποιούνται στις περιπτώσεις όπου μια ακολουθία εντολών πρέπει να εκτελεστεί πολλές φορές

Προσοχή! Εννοούμε την ίδια ακολουθία εντολών, δηλαδή τις ίδιες ενέργειες συνεχώς. Π.χ. :

- Εμφάνιση ενός μηνύματος πολλές φορές

- Εισαγωγή πολλών στοιχείων

- Εισαγωγή στοιχείου μέχρι να δοθεί κάποιο συγκεκριμένο (έλεγχος εισόδου)

- Επαναλαμβανόμενες πράξεις σε μια αριθμητική ακολουθία

Οι δομές επανάληψης πρέπει να ολοκληρώνονται!

Αλγοριθμικό κριτήριο περατότητας



Διαβάζω 100 αριθμούς και
τους εμφανίζω. Πόσες
μεταβλητές χρειάζομαι;

1. Διαβασε α

2. Εμφανισε α

3. Διαβασε α

4. Εμφανισε α

5. Διαβασε α

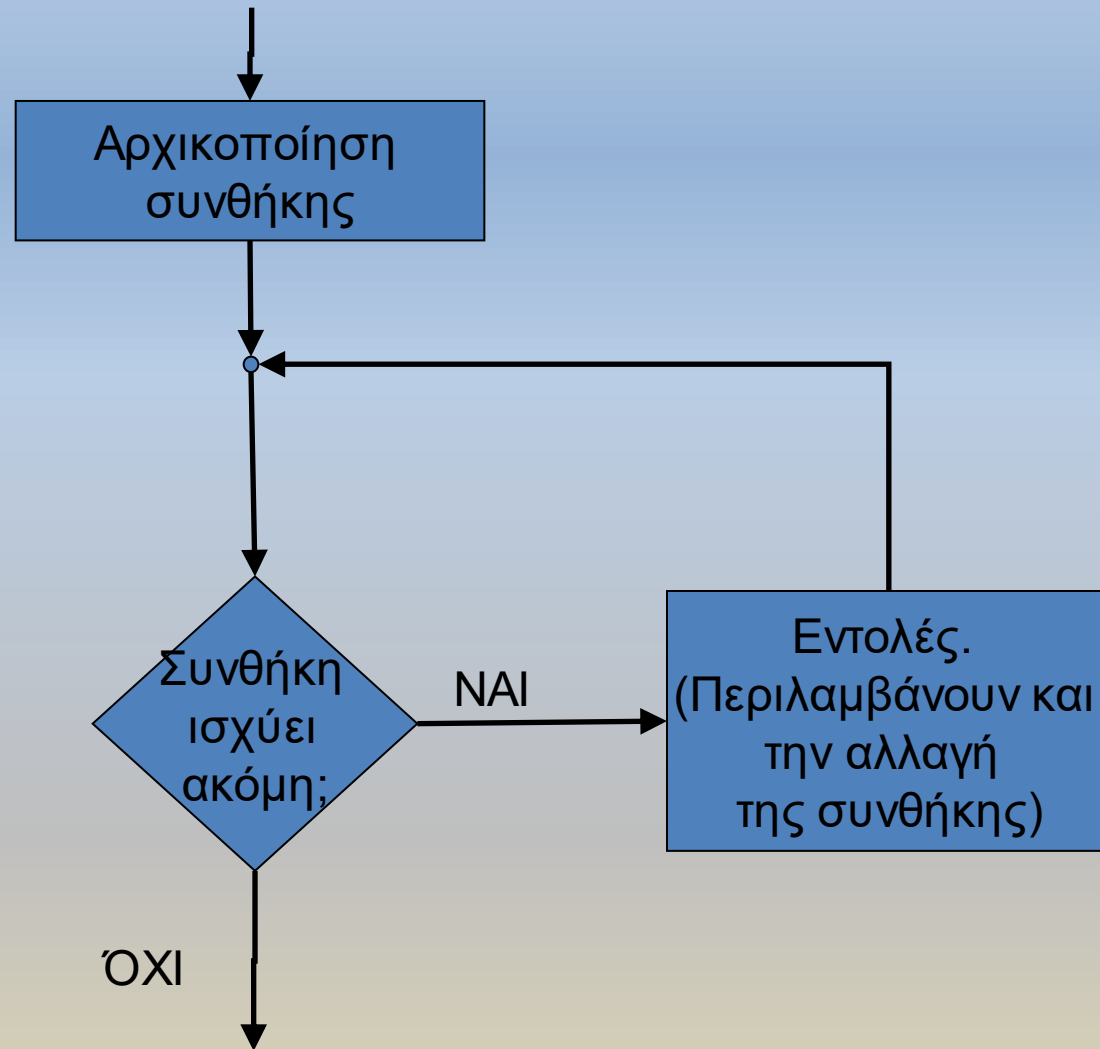
6. Εμφανισε α

7.

199. Διαβασε α

200. Εμφανισε α

4. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ «ΟΣΟ» (διαγραμματικά)



Αλγόριθμος επαναληψη_διαβασμα_100

$i \leftarrow 1$

Όσο $i \leq 100$ επαναλαβε

Διάβασε α

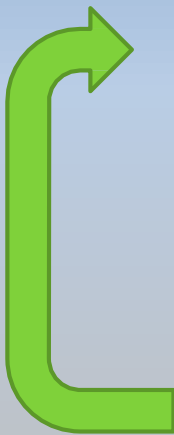
Εμφάνισε α

$i \leftarrow i + 1$

Τέλος_επανάληψης

Τέλος επαναληψη_διαβασμα_100

100
φορές



4. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ «ΜΕΧΡΙΣ ΟΤΟΥ»

Χρησιμοποιείται επίσης όταν **ΔΕΝ** είναι γνωστός ο αριθμός επαναλήψεων

Αρχή_επανάληψης

εντολές

.....

Μέχρις_ότου μια_συνθήκη_ισχύσει

} περιοχή εντολών

Όπου:

μια_συνθήκη_ισχύσει : είναι μια έκφραση, της οποίας το αποτέλεσμα είναι η λογική τιμή **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**.
π.χ. **ΜΕΣΟΣ_ΟΡΟΣ** ≥ 10 ή **ΤΑΞΗ** = "Γ" κ.λπ.

Οι εντολές θα εκτελεστούν τουλάχιστον μία φορά, αφού η συνθήκη αποτιμάται στο τέλος.

Καθώς εκτελούνται οι εντολές, πρέπει κάποια στιγμή να αλλάξουν τη συνθήκη, ώστε να αποφασιστεί η έξοδος ή όχι, από τη δομή.

Αν λοιπόν σε κάθε επανάληψη, η συνθήκη παραμένει πάντα **ΨΕΥΔΗΣ**, τότε θα εκτελείται επ' άπειρον (ατέρμων βρόχος) και θα παρατηρήσουμε ότι η εφαρμογή μας έχει κολλήσει.

Διαφέρει από τη δομή «ΟΣΟ», στο ότι (α) η συνθήκη εξετάζεται στο τέλος και (β) η συνθήκη είναι η αντίθετη (έχουμε επανάληψη όσο είναι **ΨΕΥΔΗΣ**)

$x \leftarrow 50$

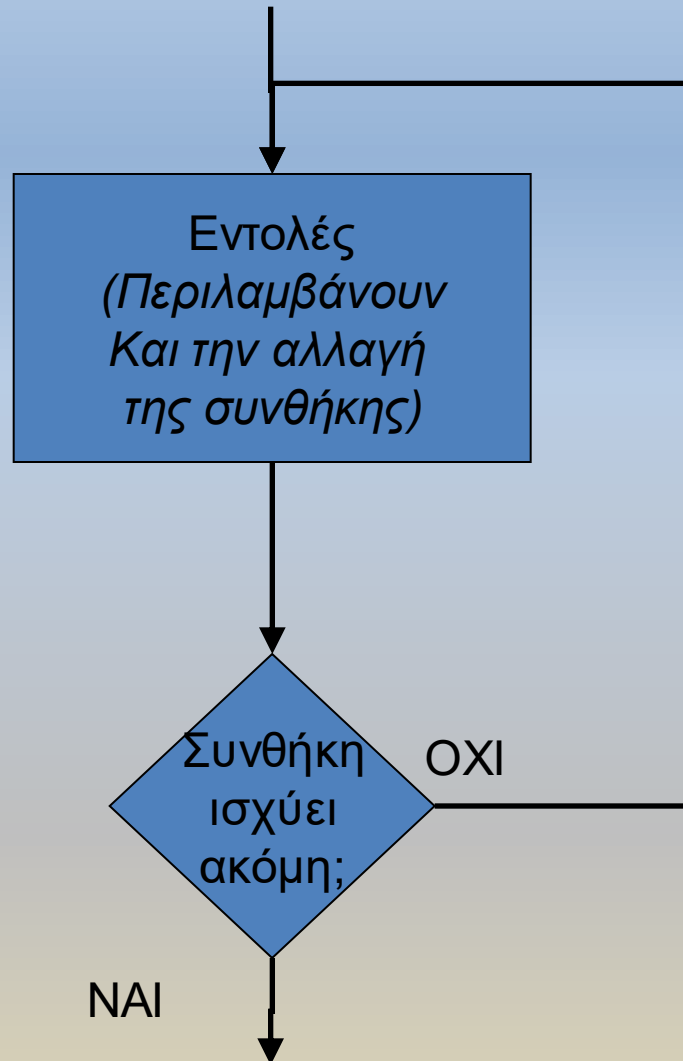
Αρχή_επανάληψης

$y \leftarrow y + 2 * x + 5$

$x \leftarrow x - 1$

Μέχρις_ότου $x = 0$

4. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ «ΜΕΧΡΙΣ ΟΤΟΥ» (διαγραμματικά)



.....
Τέλος_επανάληψης

} περιοχή εντολών

Όπου :

X : κάτω όριο

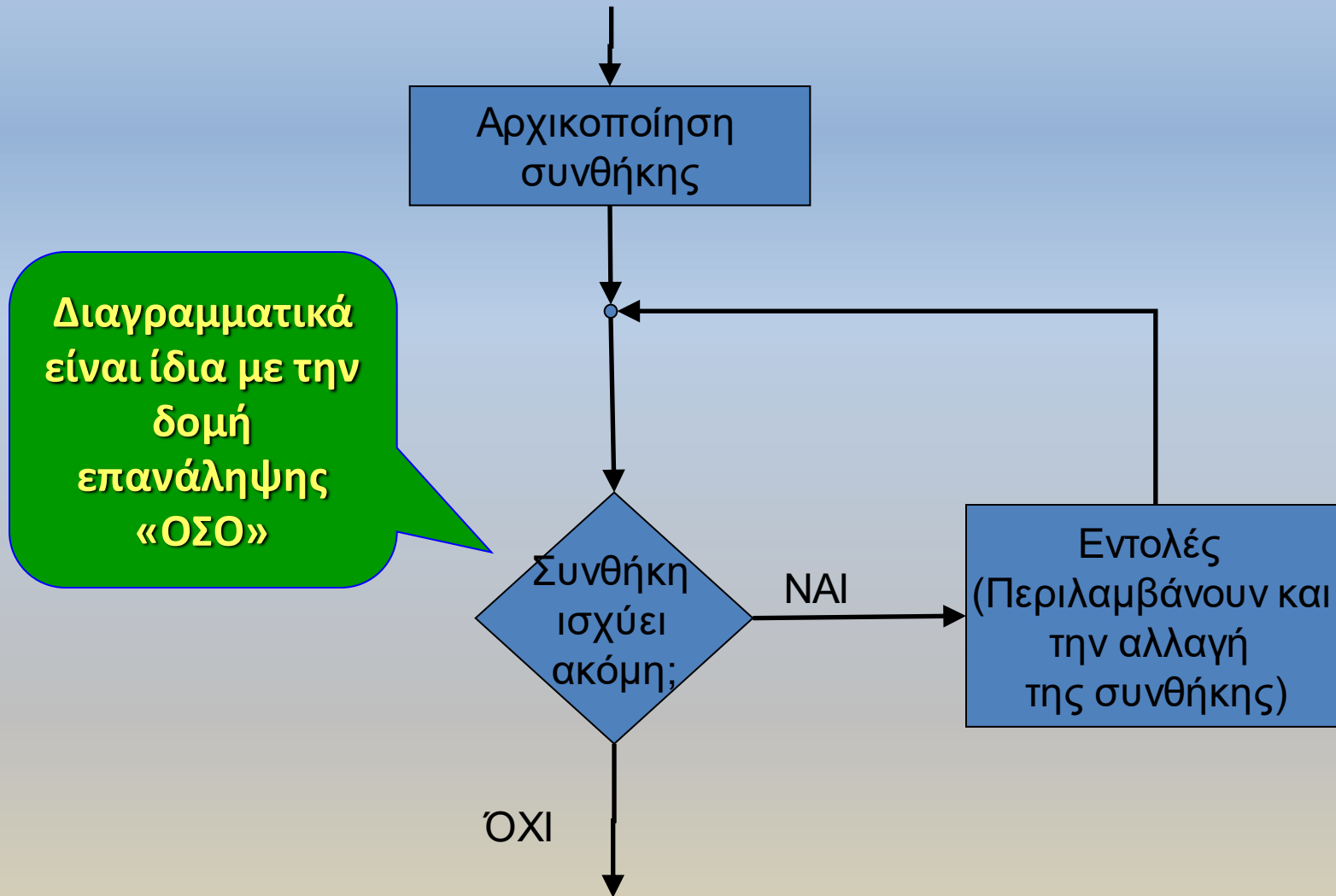
Για x από 1 μέχρι 10

$$y \leftarrow y + 2 * x + 5$$
$$z \leftarrow x - 1$$

Εμφάνισε γ, z

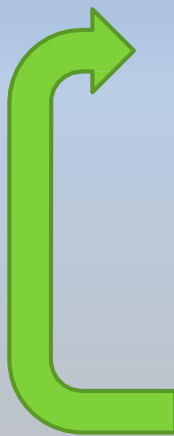
τέλος_επανάληψης

4. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ «ΓΙΑ» (διαγραμματικά)



Αλγόριθμος επαναληψη_διαβασμα_100

100
φορές



Για i από 1 μέχρι 100 με βήμα 1

Διάβασε α

Εμφάνισε α

Τέλος_επανάληψης

Τέλος επαναληψη_διαβασμα_100

ΕΜΦΩΛΕΥΜΕΝΗ «ΓΙΑ»

Αλγόριθμος Λύσεις

Για **x** από 1 μέχρι 100

Για **y** από 1 μέχρι 100

Για **z** από 1 μέχρι 100

Αν $3*x+2*y-7*z = 5$ τότε

Τύπωσε **x, y, z**

Τέλος_Αν

Τέλος_επανάληψης

Τέλος_επανάληψης

Τέλος_επανάληψης

Τέλος Λύσεις

Ο εσωτερικός βρόχος βρίσκεται πάντα ολόκληρος μέσα στον εξωτερικό, δηλ. ο βρόχος που ξεκινάει τελευταίος ολοκληρώνεται πρώτος
Η είσοδος σε κάθε βρόχο γίνεται υποχρεωτικά από την αρχή του
ΔΕΝ πρέπει να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων (**x, y, z**)
Η «Αν» θα εκτελεστεί $100 \times 100 \times 100$ φορές!

ΑΛΓΟΡΙΘΜΟΣ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΥ ΑΛΛΑ ΡΩΣΙΚΑ ΣΕ ΦΥΣΙΚΗ ΓΛΩΣΣΑ

Παίρνουμε δύο θετικούς ακέραιους αριθμούς
A, B

Διπλασιάζουμε τον A και υποδιπλασιάζουμε
τον B (αγνοώντας τα δεκαδικά ψηφία)

Συνεχίζουμε όσο ο B είναι > 0

Προσθέτουμε τους αριθμούς της 1^{ης} στήλης
για τους οποίους ο αριθμός στη 2^η στήλη είναι
περιττός

Το άθροισμα που προκύπτει είναι το
γινόμενο των δύο αρχικών αριθμών

A	B	
45	19	45
90	9	90
180	4	
360	2	
720	1	720
Άθροισμα		855