

## Ενότητα 1

Ενότητες βιβλίου: 2.1, 2.3, 6.1 (εκτός ύλης αλλά χρειάζεται για την συνέχεια)

Ώρες διδασκαλίας: 1

### Τι είναι αλγόριθμος

Σύμφωνα με το σχολικό βιβλίο:

**Ορισμός:** Μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

### Κριτήρια αλγορίθμων

Σύμφωνα με το σχολικό βιβλίο:

Κάθε αλγόριθμος απαραίτητα ικανοποιεί τα επόμενα κριτήρια.

- **Είσοδος** (input). Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.
- **Έξοδος** (output). Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.
- **Καθοριστικότητα** (definiteness). Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Λόγου χάριν, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση όπου ο διαιρέτης λαμβάνει μηδενική τιμή.
- **Περατότητα** (finiteness). Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία (computational procedure).
- **Αποτελεσματικότητα** (effectiveness). Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

### Σχόλιο

Σύμφωνα με τον Donald Ervin Knuth κάθε αλγόριθμος πρέπει να ικανοποιεί τα ακόλουθα κριτήρια:

- **Είσοδος.** Κάθε αλγόριθμος πρέπει να δέχεται καμία, μία ή και περισσότερες τιμές δεδομένων.
- **Έξοδος.** Κάθε αλγόριθμος πρέπει να παράγει μία τουλάχιστον τιμή δεδομένων.
- **Καθοριστικότητα.** Το κριτήριο αυτό συχνά παραβιάζεται στις συνταγές μαγειρικής. Ένα τέτοιο παράδειγμα είναι «... και προσθέστε λίγο αλάτι ...». Το «λίγο» είναι σχετικό! Καθοριστικότητα είναι το να είναι τα πάντα στον αλγόριθμο καθορισμένα και να ξέρει κάθε

στιγμή αυτός που εκτελεί τον αλγόριθμο τι πρέπει να κάνει. Δηλαδή να μην έχει απορίες της μορφής «τι πρέπει να κάνω τώρα;» Υπάρχουν 2 περιπτώσεις παραβίασης της καθοριστικότητας:

- ο Ο αλγόριθμος να μην περιέχει όλες τις δυνατές περιπτώσεις που μπορεί να προκύψουν. Παράδειγμα είναι η διαίρεση (όπως λέει και το βιβλίο). Δηλαδή να μην εξετάζεται η περίπτωση ο διαιρέτης να είναι μηδέν. Τέτοιου είδους παραβίαση της καθοριστικότητας οδηγεί σε σφάλμα κατά την εκτέλεση του αλγορίθμου (αν προκύψει η συγκεκριμένη περίπτωση).
- ο Τα βήματα που περιέχονται να μην είναι σαφώς καθορισμένα. Παράδειγμα είναι σε μια μαγειρική συνταγή να περιέχεται η εντολή «προσθέστε λίγο αλάτι». Τι πάει να πει λίγο; Το λίγο είναι σχετικό. Ένας υπολογιστής δεν καταλαβαίνει τι σημαίνει λίγο. Δηλαδή έχουμε ασάφεια. Τέτοιου είδους παραβίαση οδηγεί σε μη εκτελέσιμο αλγόριθμο. Στη γενική περίπτωση αυτός που εκτελεί τον αλγόριθμο δε θα ξέρει τι πρέπει να κάνει εκείνη τη στιγμή. Συγκεκριμένα στον υπολογιστή, όταν έρθει η ώρα ο αλγόριθμος να γίνει πρόγραμμα σε κάποια γλώσσα δε θα υπάρχει κάποια αντίστοιχη εντολή γλώσσας που να μπορεί να γίνει κατανοητή από τον υπολογιστή. Αυτή η περίπτωση λύνεται αν χρησιμοποιούνται οι εντολές που περιέχει το βιβλίο. Οι εντολές αυτές μόνο τυχαίες δεν είναι. Είναι αυτές που υποστηρίζονται από τις πραγματικές γλώσσες προγραμματισμού και έτσι η μετάβαση από τον αλγόριθμο στην πραγματική γλώσσα είναι ρουτίνα χωρίς παραπάνω σκέψη.
- **Περατότητα.** Ο Αλγόριθμος πρέπει κάποια στιγμή να τερματίζεται.
- **Αποτελεσματικότητα.** Κάθε εντολή πρέπει να είναι τόσο απλή που να μπορεί να εκτελεστεί και από ένα άνθρωπο που διαθέτει χαρτί και μολύβι. Π.χ. η πρόσθεση άρρητων αριθμών παραβιάζει το κριτήριο της αποτελεσματικότητας. Η εντολή "Φτιάξε τη σαλάτα" παραβιάζει την αποτελεσματικότητα καθώς δεν είναι απλή ούτε εκτελέσιμη: περιλαμβάνει πληθώρα υπο-εντολών που πρέπει να περιγραφούν επακριβώς ώστε να εκτελεστούν. Αποτελεσματικότητα είναι το να είναι η κάθε εντολή αρκετά απλή έτσι ώστε να μπορεί να εκτελεστεί από τον υπολογιστή. Δηλαδή θα πρέπει να υπάρχει εντολή αντίστοιχη στη γλώσσα προγραμματισμού έτσι ώστε να μπορεί ο αλγόριθμος να μετατραπεί σε εκτελέσιμο πρόγραμμα. Παράδειγμα παραβίασης αυτού του κριτηρίου είναι η εντολή «Βρες το μεγαλύτερο από αυτούς τους 100 αριθμούς». Η γλώσσες προγραμματισμού (του επιπέδου που μιλάμε) δεν έχουν τέτοια εντολή. Χρειάζεται σύνολο εντολών για να βρούμε το μεγαλύτερο από 100 αριθμούς.

Τέτοιου είδους παραβιάσεις οδηγούν σε μη εκτελέσιμο αλγόριθμο.

## Περιγραφή και αναπαράσταση αλγορίθμων

Σύμφωνα με το σχολικό βιβλίο:

Οι τέσσερις τρόποι αναπαράστασης ενός αλγορίθμου είναι:

- με **ελεύθερο κείμενο** (free text), που αποτελεί τον πιο ανεπεξέργαστο και αδóμητο τρόπο παρουσίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη παρουσίαση παραβιάζοντας το τελευταίο χαρακτηριστικό των αλγορίθμων, δηλαδή την αποτελεσματικότητα.

- με **διαγραμματικές τεχνικές** (diagramming techniques), που συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο παλιά και η πιο γνωστή ίσως, είναι το διάγραμμα ροής (flow chart). Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτελεί την καλύτερη λύση, γι' αυτό και εμφανίζονται όλο και σπανιότερα στη βιβλιογραφία και στην πράξη.
- με **φυσική γλώσσα** (natural language) **κατά βήματα**. Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί το τρίτο βασικό χαρακτηριστικό ενός αλγορίθμου, όπως προσδιορίστηκε προηγουμένως, δηλαδή το κριτήριο του καθορισμού.
- με **κωδικοποίηση** (coding), δηλαδή με ένα πρόγραμμα γραμμένο είτε σε μία ψευδογλώσσα είτε σε κάποια γλώσσα προγραμματισμού που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.

### Σχόλιο

Οι τρόποι αναπαράστασης αλγορίθμου εμφανίστηκαν χρονολογικά με την ακόλουθη χρονολογική σειρά:

- **ελεύθερο κείμενο**. Σκέτο κείμενο χωρίς κάποια προκαθορισμένη δομή. Ο παλαιότερος και «χειρότερος» τρόπος αναπαράστασης.
- **φυσική γλώσσα κατά βήματα**. Ο αλγόριθμος αποτελείται από βήματα όπου κάθε τέτοιο βήμα περιγράφεται με ελεύθερο κείμενο.
- **διαγραμματικές τεχνικές**. Προσπάθησαν να εκμεταλλευτούν την δύναμη της εικόνας. Απέτυχαν διότι σε μεγάλους αλγόριθμους η εικόνα είναι χασοτική.
- **Κωδικοποίηση**. Ο κυρίαρχος σήμερα τρόπος αναπαράστασης αλγορίθμων.

## Παραδείγματα

### Ελεύθερο κείμενο

Αρχικά, θα εμφανίσουμε ένα μήνυμα που θα λέει «Παρακαλώ πληκτρολογήστε τον πρώτο αριθμό». Την τιμή που θα πληκτρολογήσει ο χρήστης θα πρέπει να την αποθηκεύσουμε σε μία μεταβλητή  $x$ . Στην συνέχεια θα εμφανίσουμε ένα μήνυμα που θα λέει «Παρακαλώ πληκτρολογήστε τον δεύτερο αριθμό. Την τιμή που θα πληκτρολογήσει ο χρήστης θα πρέπει να την αποθηκεύσουμε σε μία μεταβλητή  $y$ . Έπειτα θα υπολογίσουμε την παράσταση  $(x+y)/2$  και το αποτέλεσμα θα το εμφανίσουμε στην οθόνη του υπολογιστή συνοδευόμενο από το μήνυμα «Ο μέσος όρος είναι:».

### φυσική γλώσσα κατά βήματα

Βήμα 1: Να εμφανιστεί μήνυμα που θα λέει «Πληκτρολογήστε τον πρώτο αριθμό»

Βήμα 2: Την τιμή που θα πληκτρολογήσει ο χρήστης θα πρέπει να την αποθηκεύσουμε σε μία μεταβλητή  $x$ .

Βήμα 3: Να εμφανιστεί μήνυμα που θα λέει «Πληκτρολογήστε τον δεύτερο αριθμό»

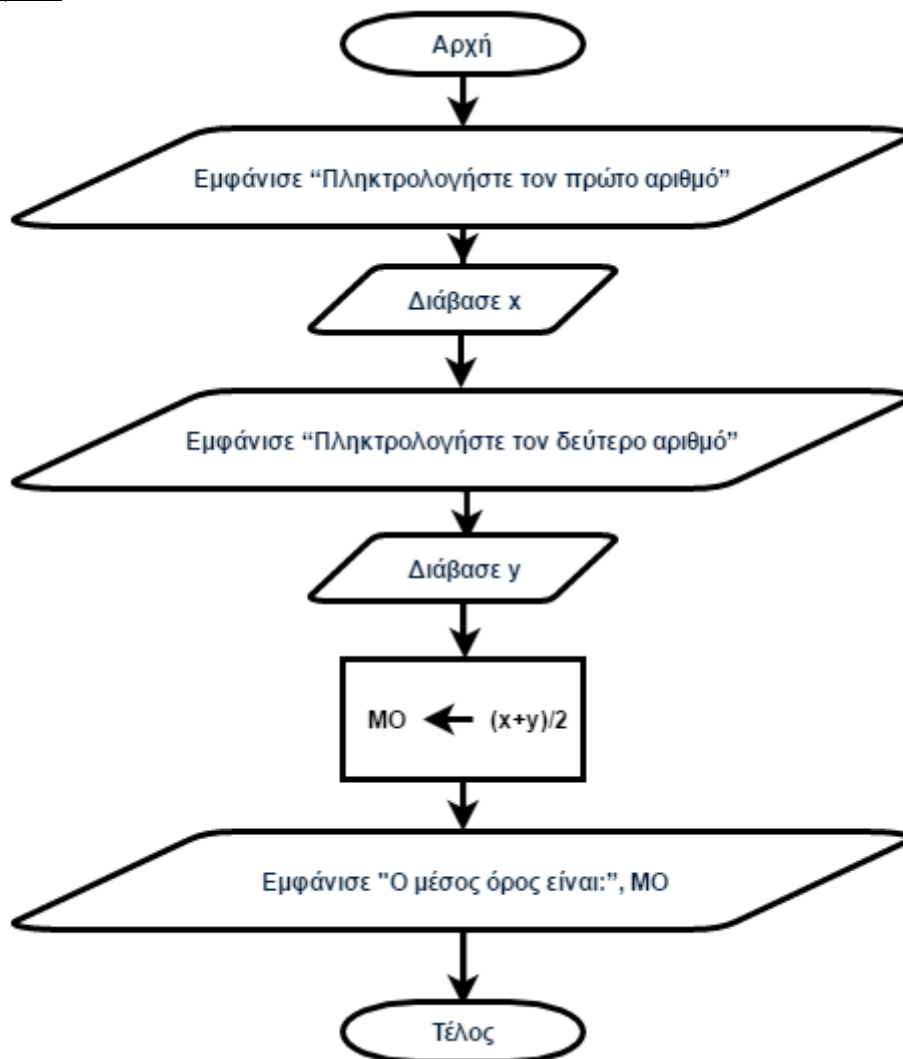
Βήμα 4: Την τιμή που θα πληκτρολογήσει ο χρήστης θα πρέπει να την αποθηκεύσουμε σε μία μεταβλητή  $y$ .

Βήμα 5: Να εμφανιστεί μήνυμα που θα λέει «Ο μέσος όρος είναι:»

Βήμα 6: Να υπολογιστεί η παράσταση  $(x+y)/2$

Βήμα 7: Να εμφανιστεί στην οθόνη το αποτέλεσμα του προηγούμενου βήματος.

Διάγραμμα ροής



Κωδικοποίηση (σε ψευδογλώσσα)

**Αλγόριθμος** Μέσος\_όρος

**Εμφάνισε** "πληκτρολογήστε τον πρώτο αριθμό"

**Διάβασε** x

**Εμφάνισε** "πληκτρολογήστε τον δεύτερο αριθμό"

**Διάβασε** y

**MO** ← (x + y) / 2

**Εμφάνισε** "Ο μέσος όρος είναι:", MO

**Τέλος** Μέσος\_όρος

## Ποια η σχέση Αλγόριθμου και προγράμματος;

Ο Αλγόριθμος είναι η στρατηγική που θα ακολουθηθεί για την επίλυση ενός προβλήματος. Εστιάζει στην ανθρώπινη κατανόηση. Αντίθετα, το πρόγραμμα είναι η υλοποίηση του αλγόριθμου σε κάποια γλώσσα προγραμματισμού (π.χ. Python, C, Java, C++).

Οι γλώσσες προγραμματισμού είναι ομάδες εντολών που καταλαβαίνει ο υπολογιστής και παρουσιάζουν αρκετές διαφορές μεταξύ τους. Γι' αυτό ένα πρόβλημα μπορεί να έχει ένα αλγόριθμο αλλά πολλά προγράμματα (ένα για κάθε γλώσσα προγραμματισμού).

### Εύρεση μέσου όρου δύο αριθμών σε C

```
#include <stdio.h>
int main(void){
float num1,num2,avg;

printf("Πληκτρολογήστε τον πρώτο αριθμό ");
scanf("%f",&num1);
printf("Πληκτρολογήστε τον δεύτερο αριθμό ");
scanf("%f",&num2);
avg=(num1+num2)/2;
printf("Ο μέσος όρος είναι: %4.2f", avg);
return 0;
}
```

### Εύρεση μέσου όρου δύο αριθμών σε Python

```
num1 = input('Πληκτρολογήστε τον πρώτο αριθμό: ')
num2 = input('Enter second number: ')

avg= (float(num1) + float(num2))/2

print('Ο μέσος όρος είναι:', avg)
```

Για κάθε πρόβλημα υπάρχουν πολλοί αλγόριθμοι και κάθε αλγόριθμος μπορεί να υλοποιηθεί με πολλά προγράμματα.

