ΑΡΧΕΣ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

(ΓΕΝΙΚΗΣ ΠΑΙΔΕΙΑΣ - Β' ΛΥΚΕΙΟΥ)

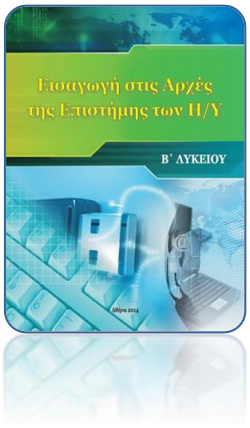
ΚΕΦ. 3 – ΑΛΓΟΡΙΘΜΟΙ (ΒΑΣΙΚΕΣ ΑΡΧΕΣ)

ΣΗΜΕΙΩΣΕΙΣ ΘΕΩΡΙΑΣ
2016 - 2017

Καθηγητής

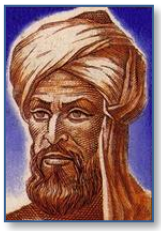
ΕΥΑΓΓΕΛΟΣ Χ. ΖΙΟΥΛΑΣ

e-mail: vczioulas@yahoo.com
site: http://www.zioulas.gr



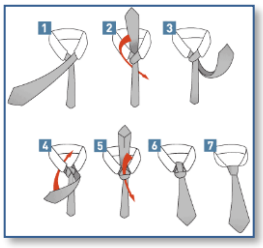
ΚΕΦ. 3 – ΑΛΓΟΡΙΘΜΟΙ (ΒΑΣΙΚΕΣ ΑΡΧΕΣ)

ΟΡΙΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ



Η λέξη **αλγόριθμος** (**algorithm**) προέρχεται από μια μελέτη του Πέρση μαθηματικού Muhammad ibn Musa al-Khwarizmi, που έζησε περί το 825 μ.Χ. και αφορούσε κυρίως σε μαθηματικούς υπολογισμούς. Στη σημερινή της μορφή έφτασε λόγω των υπολογιστών στις αρχές του 20ου αιώνα. Σήμερα, το πεδίο της μελέτης των αλγορίθμων, που καλείται **θεωρία αλγορίθμων**, αποτελεί πεδίο σοβαρής έρευνας στο χώρο της Πληροφορικής.

Αλγόριθμος είναι μια πεπερασμένη ακολουθία βημάτων ή ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο μήκος χρόνου, με σκοπό την επίλυση ενός προβλήματος.



Ο αλγόριθμος είναι έννοια που δεν συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής.

Για παράδειγμα, μια συνταγή μαγειρικής, ή το δέσιμο μιας γραβάτας αποτελούν προβλήματα τα οποία λύνονται μέσα από μία βηματική διαδικασία, δηλαδή με έναν αλγόριθμο.

Ο ΑΛΓΟΡΙΘΜΟΣ ΤΟΥ ΕΥΚΛΕΙΔΗ

Ένας από τους πρώτους ιστορικούς αλγόριθμους, είναι η ο **αλγόριθμος του Ευκλείδη** για την εύρεση του **μέγιστου κοινού διαιρέτη** μεταξύ δύο θετικών ακεραίων αριθμών **x, y**.

Ο αλγόριθμος αυτός βασίζεται σε μια επαναληπτική διαδικασία που συνεχίζεται όσο το ακέραιο υπόλοιπο της διαίρεσης x διά του y είναι διάφορο του μηδενός. Στο τέλος, η τιμή της μεταβλητής x, είναι ο μέγιστος κοινός διαιρέτης (ζητούμενο).

Παρακάτω παρατίθεται ο αλγόριθμος του Ευκλείδη σε μορφή **κωδικοποίησης**, στην οποία η αρίθμηση των εντολών γίνεται για λόγους κατανόησης.

```
1.  Αλγόριθμος Ευκλείδης
2.  Διάβασε x, y
3.  z ← y
4.  Όσο z ≠ 0 επανάλαβε
5.     z ← x mod y
6.     x ← y
7.     y ← z
8.  Τέλος_επανάληψης
9.  Εμφάνισε x
10. Τέλος Ευκλείδης
```

Στον Ευκλείδειο αλγόριθμο δεν έχει σημασία ποιον από τους δύο αριθμούς θεωρούμε ως x και ποιον ως y.

Παρακάτω, δίνεται ένα παράδειγμα εκτέλεσης του αλγορίθμου με χρήση **πίνακα τιμών** αν δεδομένα εισόδου δοθούν οι αριθμοί **35** και **25.** Για τις ανάγκες εκτέλεσης του αλγορίθμου, θεωρούμε ότι x = 35 και y = 25.

Αριθμός Εντολής	x	y	z	z ≠ 0	Έξοδος
2	35	25			
3			25		
4				Αληθής	
5			10		
6	25				
7		10			
4				Αληθής	
5			15		
6	10				
7		15			
4				Αληθής	
5			10		
6	15				
7		10			
4				Αληθής	
5			5		
6	10				
7		5			
4				Αληθής	
5			0		
6	5				
7		0			
4				Ψευδής	
9					5

Η αρίθμηση των εντολών ενός αλγορίθμου γίνεται για τις ανάγκες της **εκτέλεσης** του αλγορίθμου, και όχι τις ανάγκες της **κωδικοποίησης**.

Για κάθε εντολή που εκτελείται, καταγράφουμε τον αριθμό της γραμμής και το αποτέλεσμα της εκτέλεσης στο αντίστοιχο κελί.

Οι εντολές της γραμμής 1 και 10 είναι δηλωτικές εντολές και δεν αποτυπώνονται.

Κάθε συνδυασμός δοκιμαστικών δεδομένων που χρησιμοποιεί ένας αλγόριθμος κατά την εκτέλεση του, ονομάζεται **στιγμιότυπο** του προβλήματος. Στο προηγούμενο παράδειγμα, το στιγμιότυπο του προβλήματος είναι οι τιμές 35 και 25.

Ο αλγόριθμος του Ευκλείδη μπορεί να επιλύσει κάθε πρόβλημα που απαιτεί την εύρεση του μέγιστου κοινού διαιρέτη δύο αριθμών, ανεξάρτητα από το ποιοι είναι οι αριθμοί. Άρα, ο αλγόριθμος αυτός λειτουργεί σωστά για κάθε στιγμιότυπο του προβλήματος.

Ο ευκλείδειος αλγόριθμος βρίσκει σήμερα πολλές εφαρμογές π.χ. στην κρυπτογραφία, στο ηλεκτρονικό εμπόριο, στην επίλυση διοφαντικών εξισώσεων κ.α.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΑΛΓΟΡΙΘΜΟΥ (ΚΡΙΤΗΡΙΑ ΠΛΗΡΟΤΗΤΑΣ)

ΕΙΣΟΔΟΣ (Input)

Κάθε αλγόριθμος δέχεται ένα σύνολο μεταβλητών εισόδου (που μπορεί να είναι και το κενό σύνολο), οι οποίες αποτελούν τα δεδομένα του.

```
Αλγόριθμος Είσοδος
Διάβασε Α, Β
Χ ← Α + Β + Γ
Εμφάνισε Χ
Τέλος Είσοδος
```

Πρόβλημα εισόδου

Στην εντολή εκχώρησης **Χ ← Α+Β+Γ** η μεταβλητή Γ είναι αόριστη, δηλαδή χρησιμοποιείται χωρίς πρώτα να έχει διαβαστεί.

ΕΞΟΔΟΣ (Output)

Κάθε αλγόριθμος χρειάζεται να δημιουργεί τουλάχιστον ένα αποτέλεσμα, που είναι η έξοδος του.

```
Αλγόριθμος Έξοδος
Διάβασε Α, Β
S ← Α + Β
P ← Α * Β
Τέλος Έξοδος
```

Πρόβλημα εξόδου

Ο αλγόριθμος αν και υπολογίζει δύο μεταβλητές, ωστόσο δεν κάνει χρήση της εντολής **Εμφάνισε** ώστε να εμφανίσει κάποιο αποτέλεσμα στην οθόνη.

ΚΑΘΟΡΙΣΤΙΚΟΤΗΤΑ (Definiteness)

Κάθε εντολή ενός αλγορίθμου πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της, δηλαδή να καλύπτει κάθε δυνατό περιορισμό.

```
Αλγόριθμος Καθοριστικότητα
Διάβασε Α, Β
Χ ← Α / Β
Εμφάνισε Χ
Τέλος Καθοριστικότητα
```

Πρόβλημα καθοριστικότητας

Η εντολή εκχώρησης **Χ ← Α/Β** δεν είναι αυστηρά καθορισμένη, δηλαδή ο παρανομαστής Β ενδέχεται να είναι 0.

ΠΕΡΑΤΟΤΗΤΑ (Finiteness)

Κάθε αλγόριθμος πρέπει να ολοκληρώνεται μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του.

```
Αλγόριθμος Περατότητα
Α ← 1
Όσο Α <= 10 επανάλαβε
    Εμφάνισε Α
    Α ← Α - 1
Τέλος_επανάληψης
Τέλος Περατότητα
```

Πρόβλημα περατότητας

Ο αλγόριθμος κάνει άπειρες επαναλήψεις (ατέρμων βρόχος) διότι η συνθήκη της εντολής επανάληψης (Α<=10) θα είναι πάντα Αληθής.

ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΤΗΤΑ (Effectiveness)

Κάθε εντολή ενός αλγορίθμου χρειάζεται να είναι διατυπωμένη απλά και κατανοητά, ώστε να μπορεί να εκτελεστεί επακριβώς και σε πεπερασμένο μήκος χρόνου.

```
Αλγόριθμος Αποτελεσματικότητα
Διάβασε Α,Β
Διάβασε Χ,Υ
Π1 ← Α div Β
Π2 ← Χ / Υ
Εμφάνισε Π1, Π2
Τέλος Αποτελεσματικότητα
```

Πρόβλημα αποτελεσματικότητας

Αν οι μεταβλητές Α, Β είναι ακέραιες και οι μεταβλητές Χ, Υ είναι πραγματικές, τότε η εντολή εκχώρησης **Π2 ← Χ/Υ** δεν είναι αποτελεσματική γιατί η διαίρεση δύο πραγματικών αριθμών με άπειρα δεκαδικά ψηφία γίνεται κατά προσέγγιση.

ΒΑΣΙΚΟΙ ΤΥΠΟΙ ΑΛΓΟΡΙΘΜΩΝ

Σύμφωνα με τις δυνατότητες που έχει η Κεντρική Μονάδα Επεξεργασίας (CPU) ενός υπολογιστή, οι αλγόριθμοι διακρίνονται σε δύο βασικές κατηγορίες:

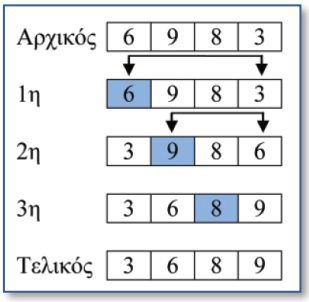
Σειριακοί Αλγόριθμοι	Παράλληλοι Αλγόριθμοι
Χρησιμοποιούν **μία CPU** και οι εντολές τους εκτελούνται **σειριακά** η μία μετά την άλλη.	χρησιμοποιούν **πολλαπλές CPU** έτσι ώστε μια σειρά από εντολές να εκτελούνται **παράλληλα** (ταυτόχρονα).

ΠΑΡΑΔΕΙΓΜΑ: Αν θέλουμε να ταξινομήσουμε τα στοιχεία ενός πίνακα σε αύξουσα σειρά, μπορούμε να ακολουθήσουμε είτε σειριακή είτε παράλληλη επεξεργασία.

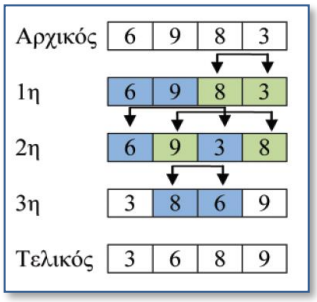
Έστω ότι έχουμε τον πίνακα

6	9	8	3

Σειριακή Επεξεργασία (1 CPU)
Μία σύγκριση στοιχείων κάθε φορά.



Παράλληλη Επεξεργασία (2 CPU)
Δύο ταυτόχρονες συγκρίσεις στοιχείων κάθε φορά.



Είναι προφανές ότι πολλά προβλήματα δεν μπορούν να λυθούν με παράλληλο τρόπο, αφού κάποιο τμήμα εντολών τους προϋποθέτει την ολοκλήρωση της εκτέλεσης κάποιου άλλου τμήματος εντολών.

Τα προβλήματα επίσης μπορούν να διακριθούν σε απλά και σε σύνθετα. Τα **απλά** λύνονται με σειριακή εκτέλεση ενός πλήθους εντολών, ενώ τα **σύνθετα** απαιτούν την εκτέλεση μόνο συγκεκριμένων βημάτων η ακόμα και την επαναληπτική εκτέλεση των βημάτων τους.

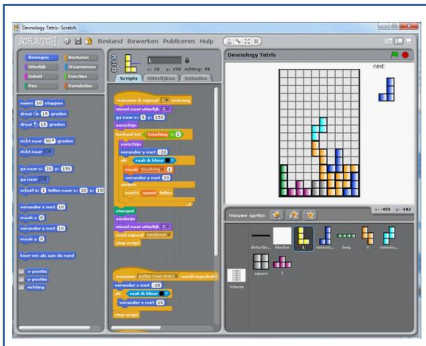
ΤΡΟΠΟΙ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΑΛΓΟΡΙΘΜΩΝ

Η αναπαράσταση των αλγορίθμων μπορεί να πραγματοποιηθεί με:

ΤΡΟΠΟΙ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΑΛΓΟΡΙΘΜΩΝ

- ΦΥΣΙΚΗ ΓΛΩΣΣΑ
- ΨΕΥΔΟΚΩΔΙΚΑΣ / ΨΕΥΔΟΓΛΩΣΣΑ
- ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ
- ΔΙΑΓΡΑΜΜΑΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

ΦΥΣΙΚΗ ΓΛΩΣΣΑ	Η αναπαράσταση γίνεται με την **ομιλούμενη γλώσσα**, μέσω της οποίας περιγράφονται τα βήματα επίλυσης του προβλήματος, με τον κίνδυνο ωστόσο να υπάρχουν ασάφειες στις οδηγίες.
ΨΕΥΔΟΓΛΩΣΣΑ ή ΨΕΥΔΟΚΩΔΙΚΑΣ	Είναι μια **υποθετική γλώσσα** για την αναπαράσταση αλγορίθμων με στοιχεία από κάποιες γλώσσες προγραμματισμού, παραλείποντας κάποιες λεπτομέρειες.
ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	Είναι μια τεχνητή γλώσσα, που έχει αναπτυχθεί για να δημιουργεί ή να εκφράζει προγράμματα για τον υπολογιστή. Σήμερα υπάρχουν **οπτικές** και **κειμενικές γλώσσες**. Οι οπτικές κάνουν γραφικό χειρισμό προγραμματιστικών στοιχείων, ενώ οι κειμενικές χρησιμοποιούν κείμενο με λέξεις, αριθμούς και σημεία στίξης.
ΔΙΑΓΡΑΜΜΑΤΙΚΕΣ ΤΕΧΝΙΚΕΣ	Είναι γραφικός τρόπος παρουσίασης του αλγόριθμου, με πιο διαδεδομένη τεχνική τα **Διαγράμματα Ροής** τα οποία χρησιμοποιούν γεωμετρικά σχήματα και σύμβολα για να παραστήσουν συγκεκριμένες ενέργειες και εντολές.



ΟΠΤΙΚΕΣ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

```
PROGRAM primecheck;
  { This is Turbo Pascal. }

VAR n,i,max: INTEGER;
    continue: BOOLEAN;
    status: STRING[13];

BEGIN
 REPEAT
  write('Enter a whole number (type 0 to quit): ');
  readln(n);
  status := ' is prime';
  IF n>2 THEN continue := TRUE
         ELSE continue := FALSE;
  i := 1;
  max := TRUNC(SQRT(n));
     { max is the largest divisor that must be
       checked to see if n is prime}
  WHILE continue DO
    BEGIN
      i := i + 1;
      IF (n MOD i)=0 THEN
        BEGIN
          status := ' is not prime';
          continue := FALSE
        END;
      continue := (i < max);
    END; {end of WHILE loop}
  write(n);
  writeln(status);
 UNTIL n=0
END.
```

ΚΕΙΜΕΝΙΚΕΣ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Τα κυριότερα **<u>γεωμετρικά σχήματα</u>** στα Διαγράμματα Ροής είναι:

Η **έλλειψη** δηλώνει την αρχή και το τέλος του αλγορίθμου.

Το **παραλληλόγραμμο** δηλώνει την είσοδο ή την έξοδο στοιχείων.

Το **ορθογώνιο** δηλώνει την εκτέλεση μίας ή περισσοτέρων πράξεων.

Ο **ρόμβος** δηλώνει μια συνθήκη (ερώτηση) που επιδέχεται απάντηση Αληθής ή Ψευδής.

<u>ΠΑΡΑΔΕΙΓΜΑ</u>: Έστω αλγόριθμος που διαβάζει τις τιμές δύο μεταβλητών και αντιμεταθέτει το περιεχόμενό τους. Στη συνέχεια εμφανίζει τις νέες τιμές.
Παρακάτω, ο αλγόριθμος παριστάνεται με τρεις διαφορετικούς τρόπους:

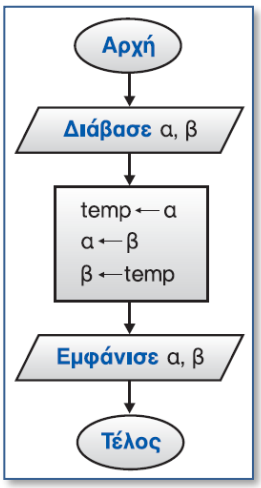
φυσική γλώσσα

Πρώτα εισάγουμε τις τιμές των δύο μεταβλητών, έστω α και β. Μετά εκχωρούμε την τιμή της μεταβλητής α σε μία νέα μεταβλητή, έστω temp (προσωρινή). Στη συνέχεια, εκχωρούμε την τιμή της μεταβλητής β στην μεταβλητή α. Τέλος εκχωρούμε την τιμή της μεταβλητής temp στην μεταβλητή β και εμφανίζουμε τις νέες τιμές.

ψευδογλώσσα

```
Αλγόριθμος Αντιμετάθεση
Διάβασε α, β
temp ← α
α ← β
β ← temp
Εμφάνισε α, β
Τέλος Αντιμετάθεση
```

διάγραμμα ροής



<u>ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΕΔΟΜΕΝΩΝ</u>

Κάθε αλγόριθμος λαμβάνει κάποια δεδομένα από την είσοδο, τα επεξεργάζεται με μια σειρά βημάτων και δίνει τα αποτελέσματα στην έξοδο. Στην ουσία ο αλγόριθμος μετατρέπει τα **δεδομένα** σε **πληροφορία**.

Πολλές φορές, η πληροφορία που παράγεται από την επεξεργασία των δεδομένων, χρειάζεται λόγω ανάδρασης, να λειτουργήσει ως νέο δεδομένο για μια νέα επεξεργασία που παράγει νέα πληροφορία κ.ο.κ.

Η θεωρία αλγορίθμων, μελετά τα δεδομένα κάτω από δύο **σκοπιές**:

- του **υλικού**
- των **γλωσσών προγραμματισμού**.

Το υλικό του υπολογιστή καθορίζει το είδος των αλγορίθμων που χρησιμοποιούνται για την επεξεργασία των δεδομένων γιατί επηρεάζει την μορφή των δεδομένων όταν αποθηκεύονται στην κύρια και βοηθητική μνήμη.

Όσον αφορά τις γλώσσες προγραμματισμού, κάθε γλώσσα μπορεί να υποστηρίζει τη χρήση διαφόρων τύπων δεδομένων, ή να δίνει στο χρήστη την ευκαιρία να δημιουργήσει τους δικούς του τύπους δεδομένων.

Οι πιο συνήθεις **τύποι δεδομένων** στις γλώσσες προγραμματισμού είναι:

Τύπος Δεδομένων	Αναπαράσταση	Παραδείγματα Τιμών
ΑΚΕΡΑΙΟΣ	ακέραια δεδομένα	8, 124, -25, 0, 2014
ΠΡΑΓΜΑΤΙΚΟΣ	πραγματικά δεδομένα	3.14, -28.1523, 0.002
ΑΛΦΑΡΙΘΜΗΤΙΚΟΣ	αλφαριθμητικά δεδομένα	"Hello", "1/10/14", "20", "AB125"
ΛΟΓΙΚΟΣ	λογικά δεδομένα	Αληθής, Ψευδής