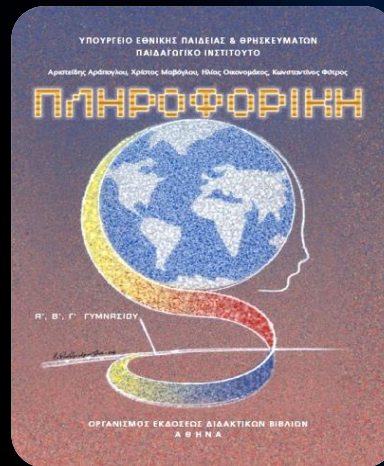




Η ΕΝΝΟΙΑ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

ΚΕΦ. 1



ΕΥΑΓΓΕΛΟΣ Χ. ΖΙΟΥΛΑΣ (ΚΑΘΗΓΗΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ)

ΠΡΟΒΛΗΜΑ

*Είναι ένα ζήτημα που τίθεται προς επίλυση, μια κατάσταση που πρέπει να αντιμετωπιστεί.
Η λύση του όμως δεν είναι γνωστή ούτε προφανής.*

- Τα προβλήματα του σχολείου είναι συνήθως **υπολογιστικά** που απαιτούν μια σειρά από :
 - **λογικές σκέψεις** (συγκρίσεις)
 - **μαθηματικές πράξεις** (προσθέσεις, αφαιρέσεις κλπ)

■ Προβλήματα αντιμετωπίζουμε καθημερινά είτε στο **προσωπικό** είτε στον **κοινωνικό** μας χώρο

- η εύρεση της συντομότερης διαδρομής για το σχολείο,
- η οργάνωση μιας σχολικής εκδρομής,
- η τακτοποίηση των βιβλίων μας σε μία βιβλιοθήκη
- η επίλυση μιας πρωτοβάθμιας ή δευτεροβάθμιας εξίσωσης
- η εύρεση της περιμέτρου και του εμβαδού ενός κύκλου

■ Υπάρχουν και πιο **πολύπλοκα προβλήματα** όπως :

- Η ρύπανση της ατμόσφαιρας
- η τρύπα του όζοντος
- το ενεργειακό πρόβλημα
- η εξερεύνηση του διαστήματος

ΣΤΟΙΧΕΙΑ ΠΡΟΒΛΗΜΑΤΟΣ

ΔΕΔΟΜΕΝΑ



ΕΠΙΛΥΣΗ



ΖΗΤΟΥΜΕΝΟ

Είναι όσα στοιχεία
είναι γνωστά και
αδιαμφισβήτητα μέσα
στο πρόβλημα, που
βοηθούν στην επίλυσή
του

Είναι η διαδικασία
μέσω της οποίας
επιτυγχάνουμε τον
στόχο του
προβλήματος, που
είναι η εύρεση του
ζητουμένου

Είναι το στοιχείο που
ψάχνουμε να βρούμε
μέσα στο πρόβλημα,
μετά από έρευνα και
αναζήτηση

ΣΤΑΔΙΑ ΑΝΤΙΜΕΤΩΠΙΣΗΣ ΠΡΟΒΛΗΜΑΤΟΣ

ΚΑΤΑΝΟΗΣΗ
ΠΡΟΒΛΗΜΑΤΟΣ



ΑΝΑΛΥΣΗ
ΠΡΟΒΛΗΜΑΤΟΣ



ΕΠΙΛΥΣΗ
ΠΡΟΒΛΗΜΑΤΟΣ

- Είναι η σωστή και πλήρης **αποσαφήνιση** των **δεδομένων** και των **ζητούμενων** μέσα σε ένα πρόβλημα, καθώς και του περιβάλλοντος στο οποίο εντάσσεται το πρόβλημα
- Το αρχικό πρόβλημα λόγω της δυσκολίας του να αντιμετωπιστεί άμεσα, το **διασπάμε σε απλούστερα** επιμέρους προβλήματα, που έχουν απλούστερη λύση
- Είναι η **υλοποίηση της λύσης** του προβλήματος με σαφείς και απλές στην διατύπωσή του εντολές

ΠΑΡΑΔΕΙΓΜΑ

ΑΝΑΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ



Ανάλυση του προβλήματος «Οργάνωση Εκπαιδευτικής Εκδρομής» σε απλούστερα προβλήματα

ΚΑΤΗΓΟΡΙΕΣ ΠΡΟΒΛΗΜΑΤΩΝ

Δυνατότητα Επίλυσης	Βαθμός Δόμησης	Είδος επίλυσης
<u>ΕΠΙΛΥΣΙΜΑ</u> Προβλήματα που η λύση τους είναι ήδη γνωστή και έχει διατυπωθεί. π.χ. ο υπολογισμός της περιμέτρου του κύκλου	<u>ΔΟΜΗΜΕΝΑ</u> Προβλήματα που η επίλυσή τους προέρχεται από μία αυτοματοποιημένη διαδικασία γνωστών βημάτων π.χ. η επίλυση μιας πρωτοβάθμιας ή δευτεροβάθμιας εξίσωσης	<u>ΑΠΟΦΑΣΗΣ</u> Προβλήματα των οποίων η απόφαση που πρόκειται να ληφθεί σαν λύση, απαντά σε κάποιο ερώτημα κλειστού τύπου π.χ είναι ο αριθμός 5 άρτιος; Ναι ή όχι;
<u>ΑΝΟΙΚΤΑ</u> Προβλήματα που η λύση τους δεν έχει ακόμα βρεθεί, αλλά δεν έχει αποδειχτεί και ότι δεν δέχονται λύση π.χ. ύπαρξη ζωής σε άλλους πλανήτες κλπ	<u>ΗΜΙΔΟΜΗΜΕΝΑ</u> Προβλήματα που η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επιλογής π.χ. η επιλογή του μεταφορικού μέσου για ένα ταξίδι	<u>ΥΠΟΛΟΓΙΣΤΙΚΑ</u> Προβλήματα στα οποία απαιτείται μια σειρά υπολογισμών προκειμένου να δοθεί σαν απάντηση στο πρόβλημα ένας αριθμός π.χ. ο υπολογισμός μιας μαθηματικής πράξης
<u>ΑΛΥΤΑ</u> Προβλήματα για τα οποία έχει αποδειχτεί ότι δεν έχουν λύση π.χ. τετραγωνισμός του κύκλου, η αποφυγή του γήρατος κλπ	<u>ΑΔΟΜΗΤΑ</u> Προβλήματα που η λύση τους βασίζεται στην ανθρώπινη διαίσθηση και δεν ακολουθούν αυτοματοποιημένη διαδικασία επίλυσης π.χ. ο τρόπος της διεξαγωγής ενός πάρτι	<u>ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ</u> Προβλήματα στα οποία ζητάμε να βρούμε το βέλτιστο αποτέλεσμα, δηλαδή αυτό που ικανοποιεί με το καλύτερο δυνατό τρόπο τα δεδομένα π.χ. εύρεση της ταχύτερης (ή αργότερης) διαδρομής για να φτάσουμε σε ένα σημείο

ΑΛΓΟΡΙΘΜΟΣ

Είναι ένα πεπερασμένο σύνολο ξεχωριστών εντολών – βημάτων, διατυπωμένων με ακρίβεια και σαφήνεια, που έχει ως σκοπό την επίλυση ενός προβλήματος

- Λέξη που προέρχεται από το όνομα του Πέρση μαθηματικού **Mohammed ibn-Musa al-Khuwarizmi** (750 – 850 μ.Χ) για την μαθηματική επεξεργασία αριθμών (**algorismus**).
- Τον 17^ο αιώνα η λέξη συνδυάστηκε με την ελληνική λέξη «**αριθμός**» και έτσι μετατράπηκε στην τελική της μορφή.

- Ο αλγόριθμος δεν συνδέεται μόνο με επιστημονικά προβλήματα, αλλά και με προσωπικά καθημερινά προβλήματα π.χ.

- η συνταγή μιας μαγειρικής
- η επίλυση μιας μαθηματικής πράξης
- Η εύρεση της ταχύτερης διαδρομής

**Παράδειγμα αλγορίθμου
Δημιουργία γεύματος**

1. Συγκέντρωσε τα υλικά
2. Προετοίμασε τα σκεύη
3. Παρασεκύασε το φαγητό
4. Ετοίμασε τη σαλάτα
5. Στρώσε το τραπέζι
6. Γευμάτισε το φαί
7. Καθάρισε το τραπέζι
8. Πλύνε τα πιάτα
9. Καθάρισε την κουζίνα

Όταν σχεδιάζουμε έναν αλγόριθμο, πρέπει να είμαστε ιδιαίτερα προσεκτικοί, ώστε να βάζουμε με **λογική σειρά** τις **οδηγίες (instructions)** που θα μας οδηγήσουν στη σωστή επίλυση του προβλήματός.

- Τα βήματα που αποτελούν τον αλγόριθμο, ονομάζονται **εντολές** ή **οδηγίες** και πρέπει πάντα να τοποθετούνται με λογική σειρά.
- Αυτός που εκτελεί τις εντολές ενός αλγορίθμου ονομάζεται **εκτελεστής** και μπορεί να είναι ο άνθρωπος ή ο υπολογιστής.
- Όταν ο αλγόριθμος εκφράζεται με εντολές που είναι κατανοητές στον υπολογιστή, τότε ονομάζεται **πρόγραμμα**.



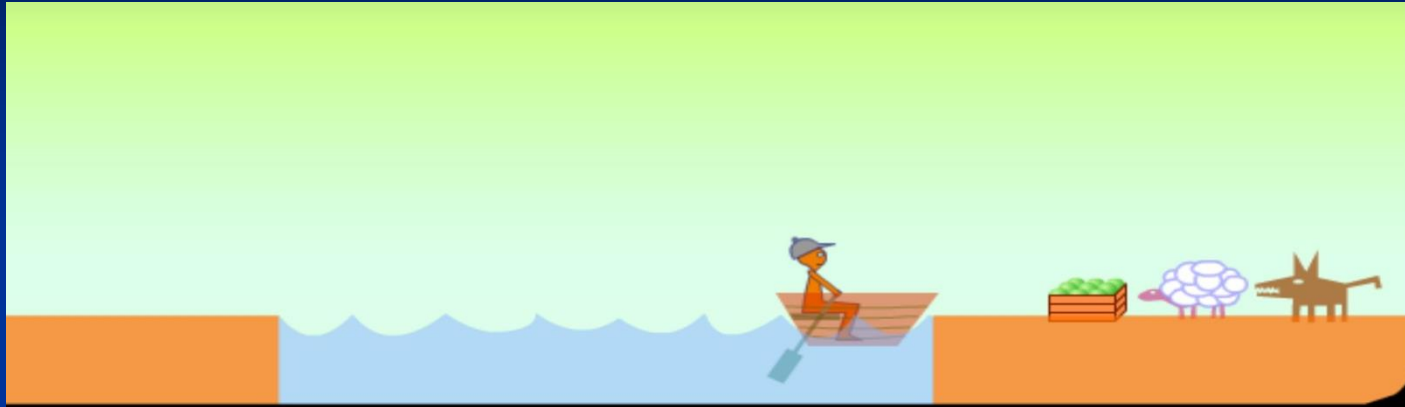
Play the game

ΠΑΡΑΔΕΙΓΜΑ

Έχει κάποιος ένα πρόβατο, ένα λύκο και ένα καφάσι με χόρτα στη μία όχθη ενός ποταμού και θέλει να τα περάσει στην απέναντι όχθη χρησιμοποιώντας μία βάρκα. Η βάρκα όμως είναι μικρή και μπορεί να μεταφέρει, εκτός από τον ίδιο, άλλο ένα από τα ζώα ή το καφάσι. Ωστόσο δεν πρέπει να μείνουν μαζί ο λύκος με το πρόβατο και το πρόβατο με τα χόρτα. Μπορείτε να δώσετε οδηγίες στο βαρκάρη για το πώς πρέπει να κάνει τη μεταφορά τους;

Δεδομένα:	1 πρόβατο, 1 λύκος, 1 καφάσι με χόρτα, μία θέση επιπλέον στη βάρκα, 2 όχθες ποταμού.
Πλαίσιο προβλήματος:	Ο λύκος δεν πρέπει να μείνει μαζί με το πρόβατο. Το πρόβατο δεν πρέπει να μείνει μαζί με τα χόρτα.
Ζητούμενο:	Να περάσει ο λύκος, το πρόβατο και το καφάσι με τα χόρτα στην απέναντι όχθη.

ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ



1. Βάλε το πρόβατο στη βάρκα.
2. Πήγαινε στην απέναντι όχθη.
3. Άφησε το πρόβατο στην όχθη.
4. Γύρνα πίσω στην αρχική όχθη.
5. Φόρτωσε το καφάσι με τα χόρτα.
6. Πήγαινε στην απέναντι όχθη.
7. Άφησε το καφάσι στην όχθη.
8. Βάλε το πρόβατο στη βάρκα.

9. Πήγαινε στην αρχική όχθη.
10. Άφησε το πρόβατο στην όχθη.
11. Βάλε το λύκο στη βάρκα.
12. Πήγαινε στην απέναντι όχθη.
13. Άφησε το λύκο στην όχθη.
14. Γύρνα πίσω στην αρχική όχθη.
15. Βάλε το πρόβατο στη βάρκα.
16. Πήγαινε στην απέναντι όχθη.
17. Άφησε το πρόβατο στην όχθη.

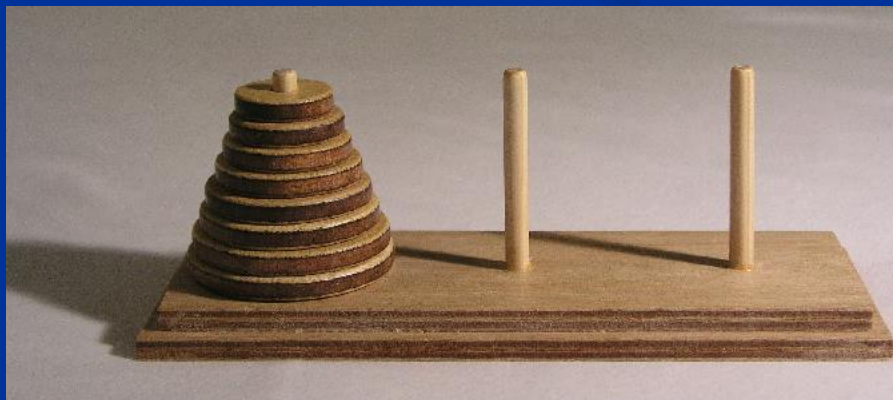


Play the game

ΟΙ ΠΥΡΓΟΙ ΤΟΥ ΑΝΟΙ

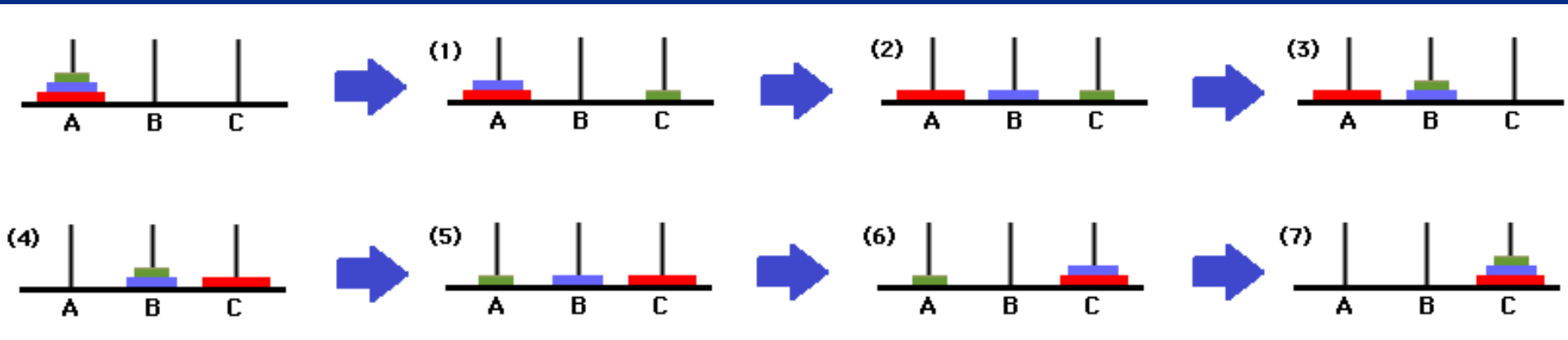
Υπάρχουν 3 στύλοι, και μια σειρά από δίσκους διαφορετικών μεγεθών οι οποίοι μπορούν να μετακινηθούν σε οποιοδήποτε στύλο. Στην αρχή οι δίσκοι βρίσκονται τοποθετημένοι στον πρώτο στύλο κατά σειρά μεγέθους από τον μεγαλύτερο προς τον μικρότερο. Το πρόβλημα απαιτεί να μετακινήσουμε τους δίσκους στον τρίτο στύλο αλλά με την ίδια σειρά μεγέθους και σεβόμενοι μια σειρά από κανόνες:

- A) ένας δίσκος μετακινείται κάθε φορά
- B) Μετακινείται μόνο ο δίσκος που βρίσκεται στην κορυφή της στοίβας
- Γ) Κανένας δίσκος δεν μπορεί να τοποθετηθεί πάνω σε μικρότερο δίσκο



ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Παράδειγμα λύσης με 3 δίσκους



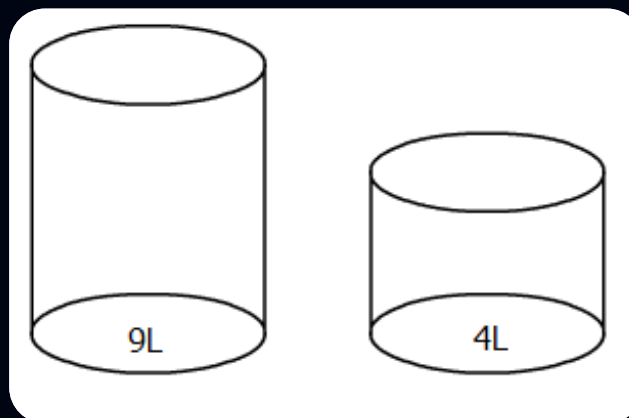
1. Ο μικρός δίσκος στον 3^ο στύλο
2. Ο μεσαίος δίσκος στον 2^ο στύλο
3. Ο μικρός δίσκος στον 2^ο στύλο
4. Ο μεγάλος δίσκος στον 3^ο στύλο
5. Ο μικρός δίσκος στον 1^ο στύλο
6. Ο μεσαίος δίσκος στον 3^ο στύλο
7. Ο μικρός δίσκος στον 3^ο στύλο

ΙΔΙΟΤΗΤΕΣ ΑΛΓΟΡΙΘΜΟΥ

ΣΑΦΗΣ	Κανένα βήμα δεν αφήνει αμφιβολίες
ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΣ	Δίνει αποτελέσματα σε πεπερασμένο χρόνο
ΠΡΑΓΜΑΤΟΠΟΙΗΣΙΜΟΣ	Είναι εκτελέσιμος από τον εκτελεστή
ΤΑΧΥΣ	Χρησιμοποιεί λίγες εντολές
ΟΙΚΟΝΟΜΙΚΟΣ	Απαιτεί όσο το δυνατό λιγότερα μέσα (πόρους)
ΓΕΝΙΚΟΣ	Μπορεί να επιλύει ένα σύνολο συγγενών προβλημάτων

ΠΑΡΑΔΕΙΓΜΑ

- ΚΕΦ.1 – ΔΡ.3 (σελ.201)
- Να περιγραφεί ο αλγόριθμος επίλυσης του παρακάτω προβλήματος **Μετρητή Νερού**.
- Έχετε στη διάθεση σας δύο μπουκάλια. Το πρώτο χωράει **9 λίτρα** και το δεύτερο **4**. Χρησιμοποιώντας αυτά τα δύο μπουκάλια πώς μπορούμε να μετρήσουμε ακριβώς **6 λίτρα**;



ΛΥΣΗ

Εντολή	Μεγάλο	Μικρό
Γέμισε το μικρό	0	4
Μετέφερε στο μεγάλο	4	0
Γέμισε το μικρό	4	4
Μετέφερε στο μεγάλο	8	0
Γέμισε το μικρό	8	4
Μετέφερε στο μεγάλο	9	3
Άδειασε το μεγάλο	0	3
Μετέφερε στο μεγάλο	3	0
Γέμισε το μικρό	3	4

Εντολή	Μεγάλο	Μικρό
Μετέφερε στο μεγάλο	7	0
Γέμισε το μικρό	7	4
Μετέφερε στο μεγάλο	9	2
Άδειασε το μεγάλο	0	2
Μετέφερε στο μεγάλο	2	0
Γέμισε το μικρό	2	4
Μετέφερε στο μεγάλο	6	0

ΠΑΡΑΔΕΙΓΜΑ

- ΚΕΦ.1 – ΔΡ.3 (σελ.201)
- Προσπαθήστε να επιλύσετε το παρακάτω πρόβλημα: Έστω ότι δεν διαθέτουμε ρολόι, αλλά δύο κλεψύδρες των 7 λεπτών η μία και των 4 λεπτών η άλλη.
- Πώς μπορούμε να ξέρουμε πότε περάσανε 9 ακριβώς λεπτά;



ΛΥΣΗ

Εντολή	Κλεψύδρα 7'	Κλεψύδρα 4'	Σύνολο
γεμίζω την μικρή και παράλληλα την μεγάλη	4 λεπτά	4 λεπτά	4 λεπτά
αναποδογυρίζω την μικρή		0 λεπτά	
γεμίζω την μεγάλη και παράλληλα τη μικρή	7 λεπτά	3 λεπτά	7 λεπτά
αναποδογυρίζω τη μεγάλη	0 λεπτά		
γεμίζω τη μικρή και παράλληλα την μεγάλη	1 λεπτό	4 λεπτά	8 λεπτά
αναποδογυρίζω τη μεγάλη			
αδειάζω τη μεγάλη	1 λεπτό		9 λεπτά

ΠΡΟΓΡΑΜΜΑ

*Ονομάζεται ένας αλγόριθμος που είναι διατυπωμένος κατά τέτοιο τρόπο, ώστε οι εντολές του να είναι **κατανοητές από τον υπολογιστή.***

- Άρα, ενώ κάθε πρόγραμμα είναι ένας αλγόριθμος, κάθε αλγόριθμος δεν σημαίνει ότι αποτελεί και πρόγραμμα
- Για να γίνει ένας αλγόριθμος κατανοητός από τον υπολογιστή, οφείλει να σέβεται μια σειρά από **ορθογραφικούς και συντακτικούς κανόνες.**

```

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.

```

Παράδειγμα
προγράμματος
(Turbo Pascal)



Έλεγχος αν ένας αριθμός
είναι πρώτος ή όχι

ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

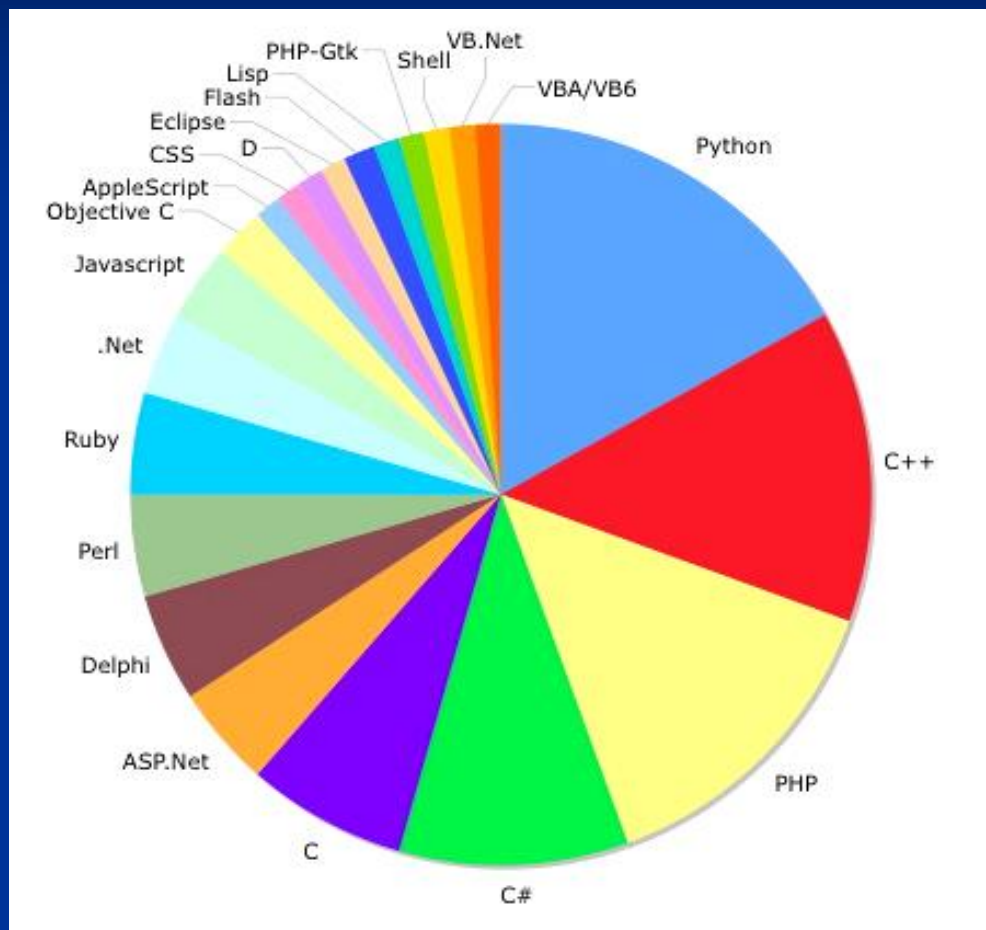
- Είναι μια **τεχνητή γλώσσα επικοινωνίας** που χρησιμοποιεί ο άνθρωπος για να επικοινωνήσει με τον υπολογιστή και να εκφράσει κατανοητά τους αλγορίθμους του.
- Όπως και κάθε φυσική γλώσσα, μια γλώσσα προγραμματισμού διαθέτει τους δικούς της **συντακτικούς** και **εννοιολογικούς κανόνες** για την σύνταξη των εντολών της.

- Κάθε γλώσσα έχει το δικό της:
 - **συντακτικό**, **λεξιλόγιο** και **αλφάβητο**
- Σήμερα υπάρχουν εκατοντάδες γλώσσες προγραμματισμού, χωρίς αυτό να σημαίνει ότι όλες είναι αναγκαίες στον προγραμματιστή π.χ.

Pascal, Basic, Fortran, Cobol, Logo, C, C++, Java, Python,
Perl, Lisp, Visual Basic, C#, SQL, Ada, Prolog

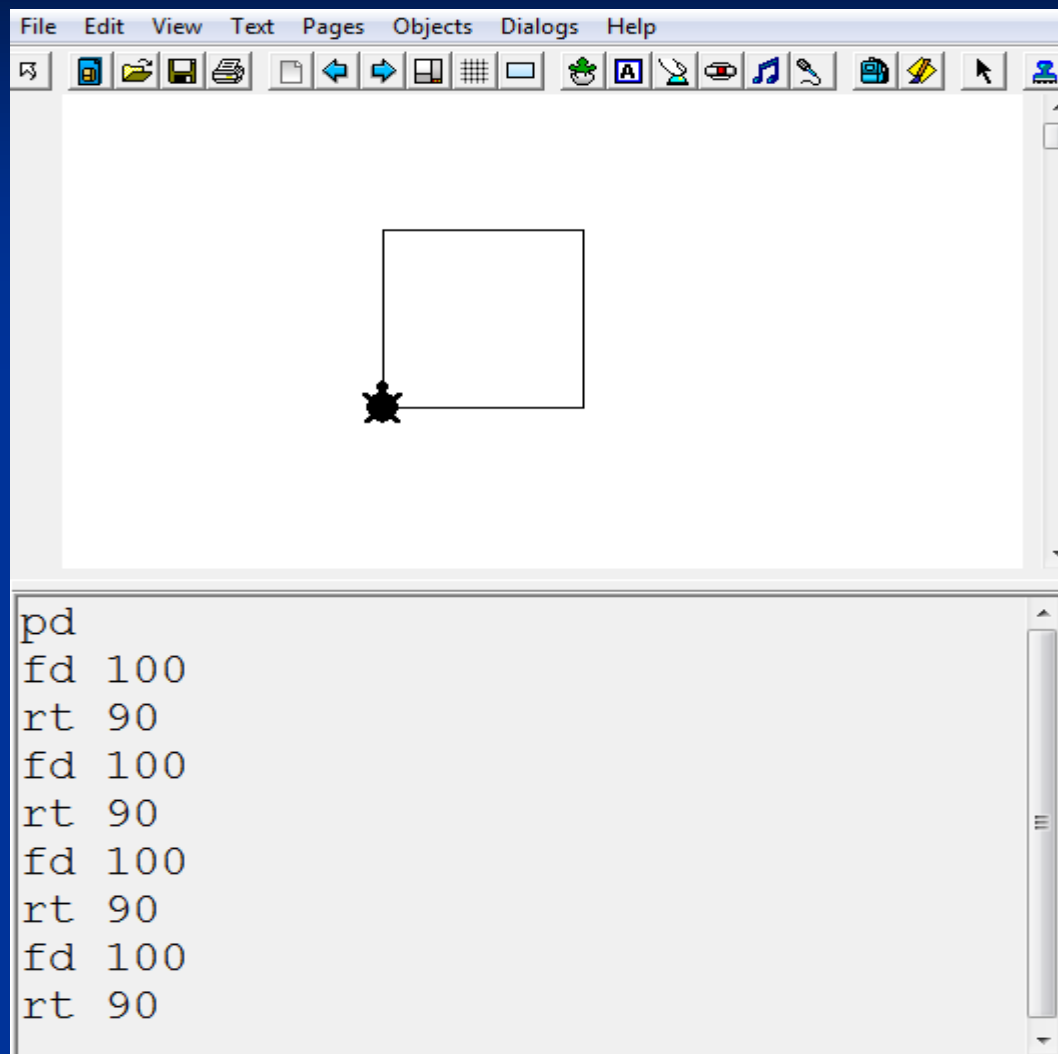
- Κάθε γλώσσα προγραμματισμού είναι φτιαγμένη για να υλοποιεί συγκεκριμένου τύπου εφαρμογές

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

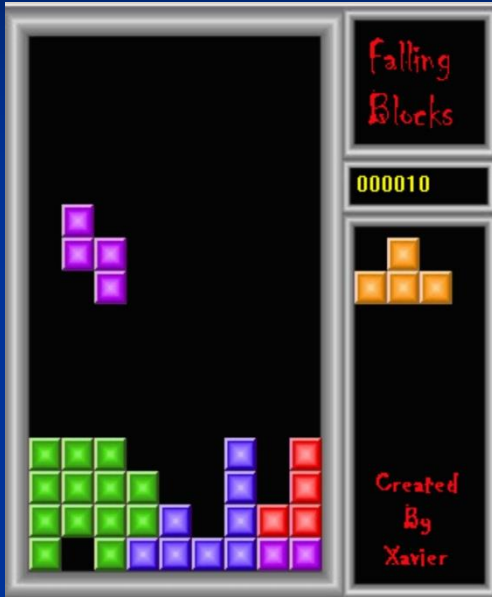


Παράδειγμα προγράμματος σε περιβάλλον Γλωσσμάθεια

Παράδειγμα προγράμματος σε περιβάλλον MicroWorlds Pro



ΠΡΟΓΡΑΜΜΑΤΙΖΟΝΤΑΣ ΠΑΙΧΝΙΔΙΑ



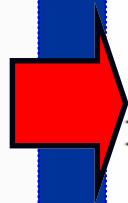
```
void DisplayBlock(SBlock Block)
{
    if (Block.nY < 1) return;
    RECT rcBlock = g_rcBlock;
    rcBlock.left = Block.nColor * BLOCK_DIAMETER;
    rcBlock.right = Block.nColor * BLOCK_DIAMETER + BLOCK_DIAMETER;

    g_pDisplay->Blit( (DWORD)Block.nX * BLOCK_DIAMETER - 2 ,
                      (DWORD)Block.nY * BLOCK_DIAMETER ,
                      g_pSecondarySurface, &rcBlock );
}
```

Το γνωστό παιχνίδι **TETRIS** είναι ένα πρόγραμμα το οποίο περιλαμβάνει μια σειρά εντολών . Ένα μικρό υποσύνολο των εντολών βλέπουμε στα δεξιά της εικόνας σε γλώσσα C

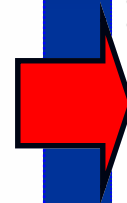
ΕΞΕΛΙΞΗ ΓΛΩΣΣΩΝ

```
10101000 00001010
10001100 00000001
00111100
01010001 00000001
01000011 00000001
11000000 11111010
10001100 00000010
11111111
```



LOOP

```
INDEX=$01
SUM=$02
LDA #10
STA INDEX
CLA
ADD INDEX
DEC INDEX
BNE LOOP
STA SUM
BRK
```



```
sum = 0
FOR index=1 TO 10
    sum=sum+index
NEXT index
END
```

Γλώσσα μηχανής
(Machine Language)

Συμβολική Γλώσσα
(Assembly)

Γλώσσα Υψηλού
επιπέδου



ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ

- Είναι η **μητρική γλώσσα** του υπολογιστή (μόνο αυτήν καταλαβαίνει).
- Οι εντολές της γράφονται σε **ακολουθίες από bits** (δυαδικά ψηφία 0 και 1).
- Αποτελεί δύσκολο και ακατανόητο τρόπο προγραμματισμού.
- Δεν είναι ενιαία σε όλους τους υπολογιστές (καθένας έχει τη δική του).

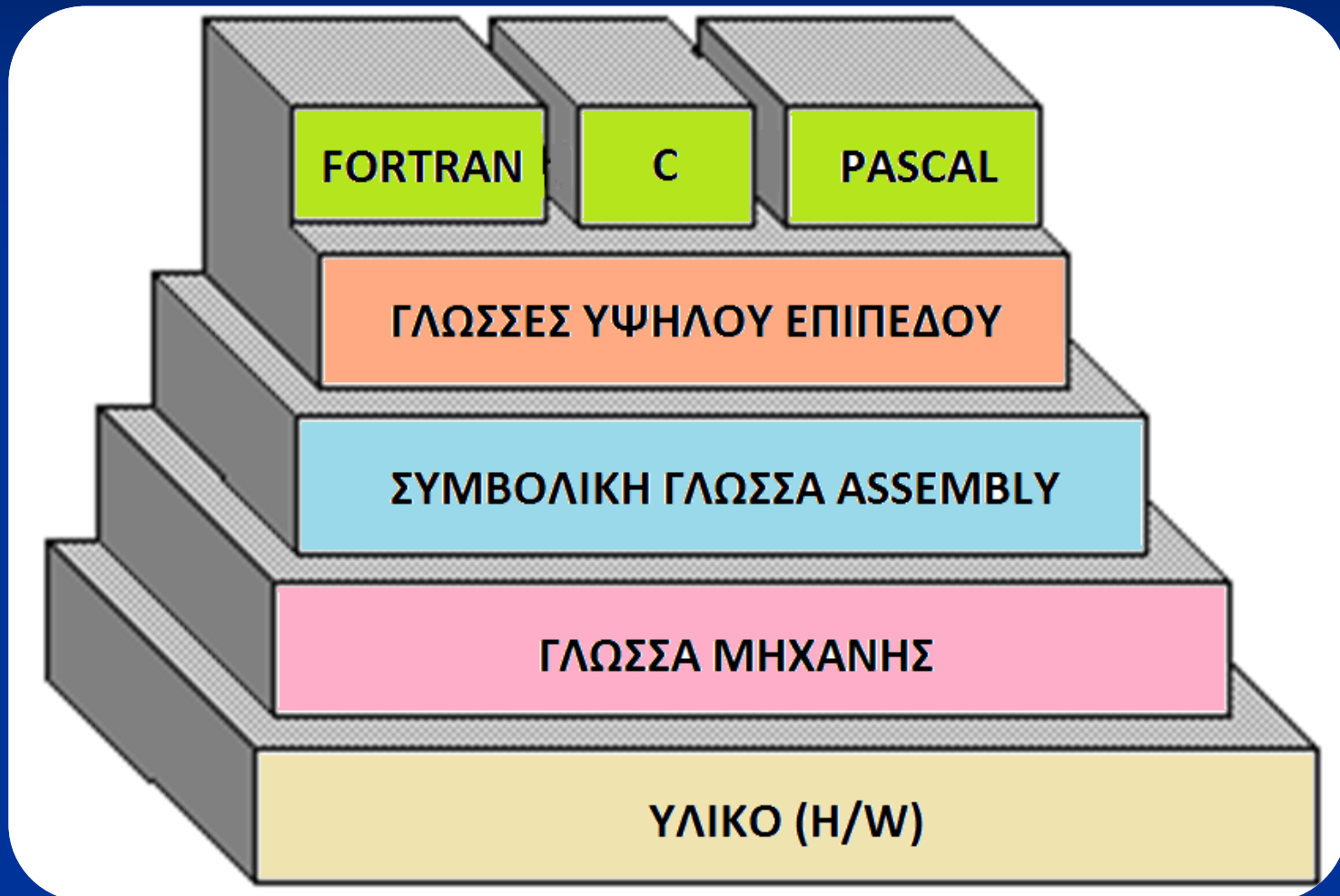
ΣΥΜΒΟΛΙΚΗ ΓΛΩΣΣΑ

- Είναι **πιο ανθρώπινη** γλώσσα που χρησιμοποιεί λέξεις αντί αριθμούς.
- Είναι **γλώσσα χαμηλού επιπέδου** διότι δεν είναι ανεξάρτητη από τον τύπο του υπολογιστή
- Το πρόγραμμα για να γίνει κατανοητό στον υπολογιστή, πρέπει να μεταφραστεί σε γλώσσα μηχανής
- Το πρόγραμμα που κάνει την μετάφραση λέγεται **συμβολομεταφραστής (assembler)**

ΓΛΩΣΣΕΣ ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ

- Μοιάζουν με τις ανθρώπινες (μικρό λεξιλόγιο, αγγλικές λέξεις, εύκολη σύνταξη, μαθηματικά σύμβολα).
- Κάθε εντολή αντιστοιχεί και μεταφράζεται σε πολλές εντολές γλώσσας μηχανής.
- Για την μετάφρασή τους σε γλώσσα μηχανής, διαθέτουν μεταφραστικά προγράμματα όπως:
 - **μεταγλωττιστές (compilers)**
 - **διερμηνευτές (interpreters)**
- Η γλώσσες αυτές έχουν **μεταφερσιμότητα**, δηλαδή είναι ανεξάρτητες από την αρχιτεκτονική του υπολογιστή.

ΚΑΤΗΓΟΡΙΕΣ ΓΛΩΣΣΩΝ



ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΓΛΩΣΣΩΝ

ΑΛΦΑΒΗΤΟ	Είναι το σύνολο των χαρακτήρων – συμβόλων που χρησιμοποιεί η γλώσσα για να εκφράσει τις εντολές της (γράμματα, ψηφία, σημεία στίξης, μαθηματικά σύμβολα)
ΛΕΞΙΛΟΓΙΟ	Είναι το σύνολο των λέξεων που αναγνωρίζει η γλώσσα ως αποδεκτές εντολές της και έχουν κάποια καθορισμένη σημασία.
ΣΥΝΤΑΚΤΙΚΟ	Είναι το σύνολο των κανόνων της γλώσσας που καθορίζει την ορθογραφία και τον τρόπο σύνδεσης των λέξεων, ώστε ο υπολογιστής να τις αντιλαμβάνεται ως εντολές. Αποτελεί την γραμματική της γλώσσας.

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

- Είναι το λογισμικό περιβάλλον το οποίο μας επιτρέπει να αναπτύσσουμε τα προγράμματά μας.
- Το περιβάλλον αυτό διαθέτει δύο βασικά εργαλεία:
 - ΚΕΙΜΕΝΟΓΡΑΦΟΣ (editor)
 - Χρησιμεύει στη σύνταξη και διόρθωση του κώδικα του προγράμματος
 - ΜΕΤΑΦΡΑΣΤΗΣ (translator)
 - Μετατρέπει τις εντολές του προγράμματος από γλώσσα υψηλού επιπέδου σε γλώσσα μηχανής, κάνοντας ταυτόχρονα και έλεγχο σφαλμάτων

ΜΕΤΑΦΡΑΣΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ

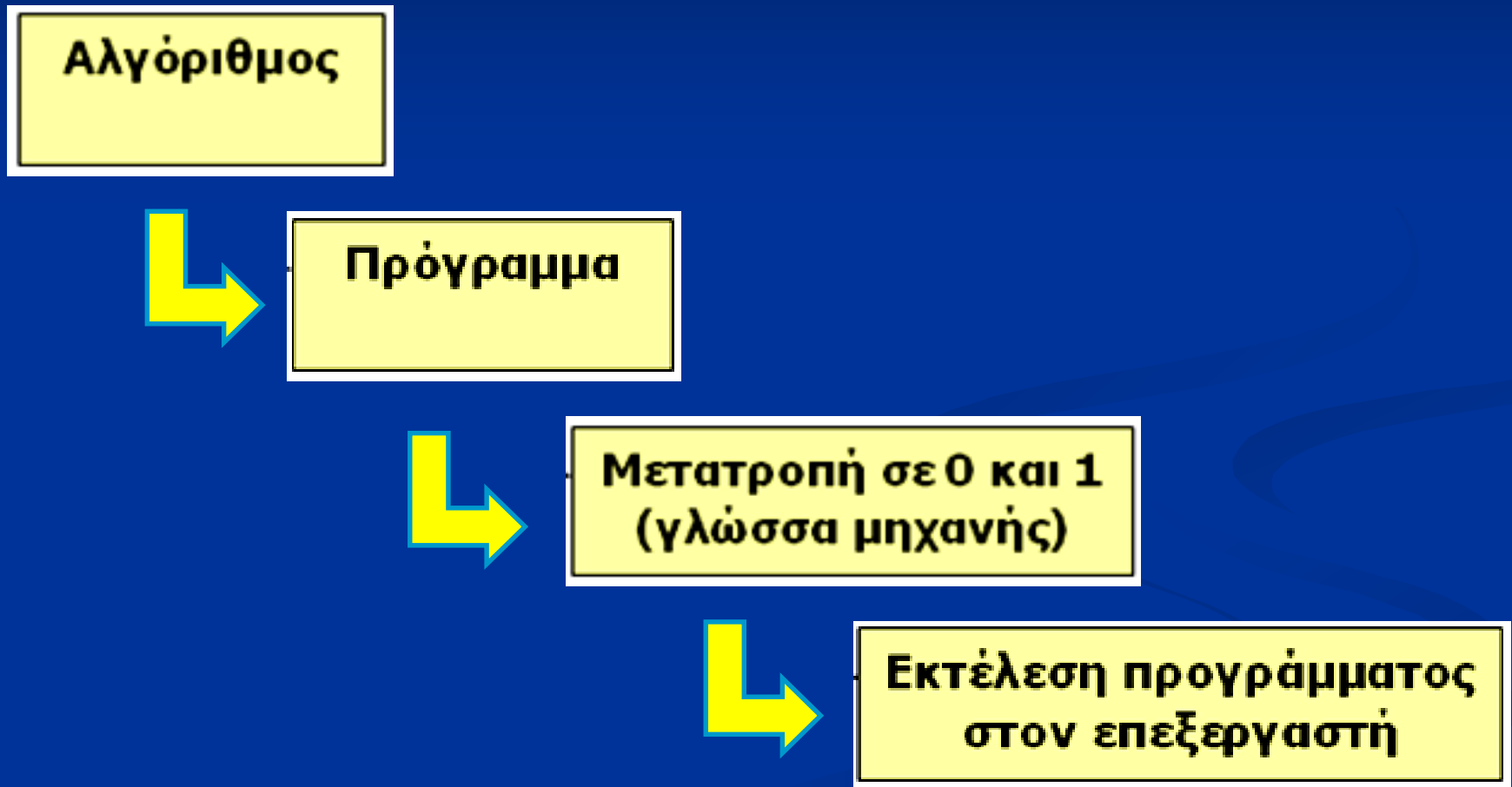
ΜΕΤΑΓΛΩΤΤΙΣΤΕΣ (COMPILERS)

Ελέγχουν **ολόκληρο** το **πρόγραμμα** για λάθη κι αν δεν εντοπίσουν κανένα, τότε το μετατρέπουν σε γλώσσα μηχανής ώστε να εκτελεστεί .

ΔΙΕΡΜΗΝΕΥΤΕΣ (INTERPRETERS)

Ελέγχουν **μία – μία** τις εντολές του προγράμματος κι αν είναι συντακτικά ορθή τότε την εκτελούν .

ΣΤΑΔΙΑ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΚΑΤΗΓΟΡΙΕΣ ΛΑΘΩΝ

- Υπάρχουν δύο βασικές κατηγορίες λαθών που συνήθως κάνει ένας προγραμματιστής:
- **ΣΥΝΤΑΚΤΙΚΑ (syntax errors)**
 - Είναι λάθη που **οφείλονται σε παραβίαση των κανόνων** της γλώσσας και μπορούν να εντοπιστούν από τον μεταγλωττιστή κατά την μετάφραση.
- **ΛΟΓΙΚΑ (runtime errors)**
 - Είναι λάθη που **οφείλονται στον αλγόριθμο** που κατασκευάσαμε για την λύση του προβλήματος και δεν μπορούν να εντοπιστούν (από τον μεταγλωττιστή) παρά μόνο σε φάση εκτέλεσης από τον ίδιο τον προγραμματιστή