

Συστήματα Αρίθμησης Αλγόριθμοι

I. Καβασίδης

Τι είναι ο υπολογιστής;

- Μηχανή
- Συνδυασμός Υλικού (Hardware, HW) και Λογισμικού (Software, SW)

Τι είναι ο υπολογιστής;

- Υλικό (Hardware)
 - Μηχανικά/Ηλεκτρονικά μέρη
 - Ότι μπορούμε να πιάσουμε με τα χέρια μας
- Λογισμικό (Software)
 - Προγράμματα
 - Σύνολο εντολών φτιαγμένο με τρόπο που να εξυπηρετεί κάποια ανάγκη
 - Διαχείριση πόρων μηχανής -> Λειτουργικό σύστημα (Operating System, OS)
 - Επικοινωνία μηχανής-υλικού -> Οδηγοί (drivers) / firmware (π.χ., BIOS)
 - Εργασία (Γενικά) -> Προγράμματα παραγωγικότητας (Office, OpenOffice)
 - Εργασία (Ειδικά) -> Σχεδιασμός (CAD), Προγραμματισμός (Compilers/IDEs), Ειδικά προγράμματα διαχείρισης

Υλικό

- Συσκευές Εισόδου
- Input

Input Devices



Υλικό

- Συσκευές Εξόδου
- Output

OUTPUT DEVICES



Υλικό

- Συσκευές Εισόδου/Εξόδου
- Input/Output



Υλικό

- Συσκευές Αποθήκευσης
- Storage



Hard Disk



RAM



ROM



CD/DVD



Floppy



Memory Card



Pen Drive



Tape

Υλικό

- Κεντρική Μονάδα Επεξεργασίας
- CPU



Υλικό

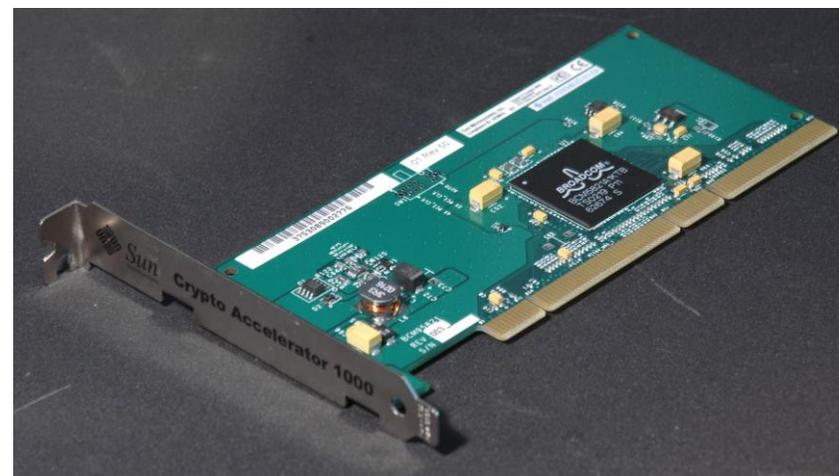
- Επιταχυντές
- Accelerators



Κάρτες γραφικών



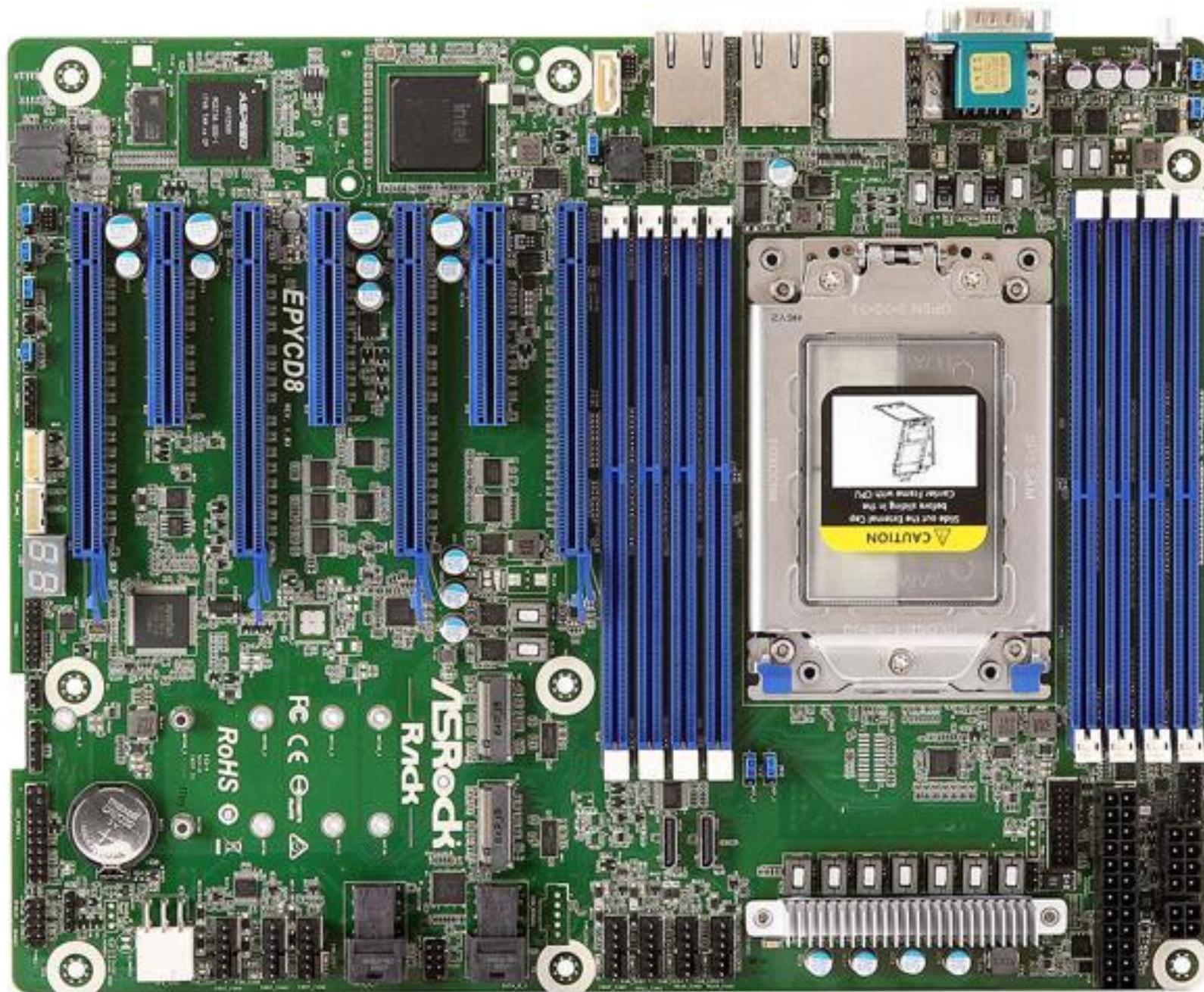
Κάρτες ήχου



Κάρτες κρυπτογράφησης

Υλικό

- Μητρική πλακέτα
- Motherboard



Υλικό

Τροφοδοτικό

Μητρική

Κάρτα γραφικών

Επεξεργαστής

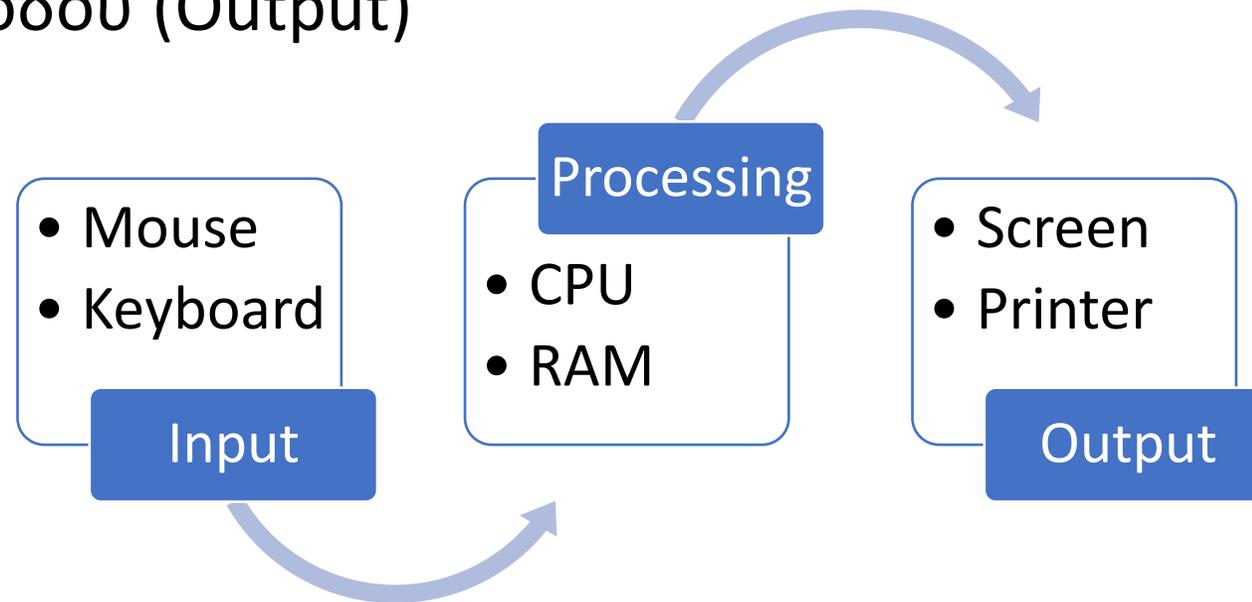
Μνήμη

Σκληρός
Δίσκος



Τι γίνεται μέσα στον Η/Υ;

- Ο χρήστης δίνει μία εντολή (μέσω Input)
- Ο επεξεργαστής παράγει ένα αποτέλεσμα σε συνεργασία με την κεντρική μνήμη (CPU-RAM)
- Το αποτέλεσμα μεταφέρεται και παρουσιάζεται στον χρήστη σε μία συσκευή εξόδου (Output)



Μονάδες μέτρησης

Ταχύτητα

- Hz (Χέρτζ)
- Byte/s (Μπάιτ ανά δευτερόλεπτο)
 - Mbyte/s, Mbit/s
- OPS/IPS (Πράξεις/Εντολές ανά δευτερόλεπτο)
 - FLOPS, MIPS

Χωρητικότητα

- Byte (Μπάιτ)
 - Gigabyte, Terabyte

Μετατροπές μονάδων μέτρησης πληροφορίας

1 Byte	8 bits
1 Kilobyte (KB)	1,024 Bytes
1 Megabyte (MB)	1,024 Kilobytes
1 Gigabyte (GB)	1,024 Megabytes
1 Terabyte (TB)	1,024 Gigabytes

Γιατί 1024?

Συστήματα αρίθμησης

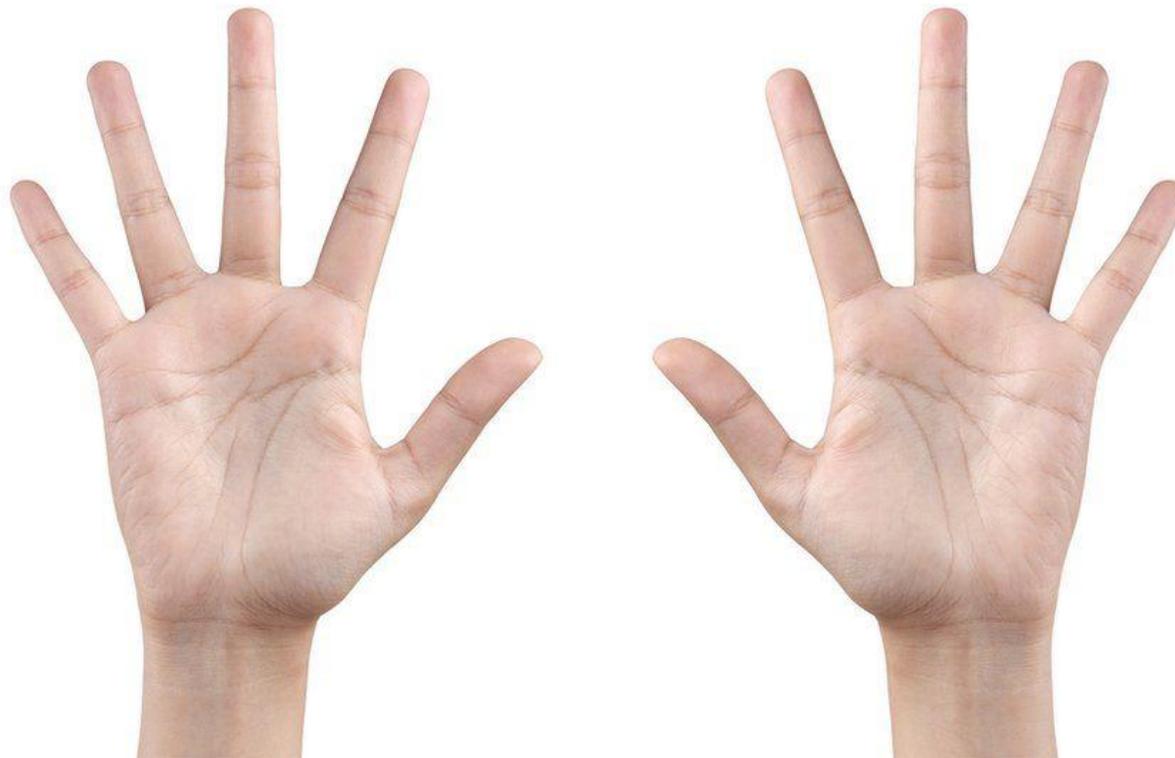
- Δεκαδικό σύστημα
 - 0,1,2,3,4,5,6,7,8,9
- Π.χ. ο αριθμός 4567_{10}

$$4 * 10^3 + 5 * 10^2 + 6 * 10^1 + 7 * 10^0$$

- Ψηφίο
- Βάση
- Θέση ή δύναμη

Συστήματα αρίθμησης

- Γιατί το δεκαδικό σύστημα;



Συστήματα αρίθμησης

- Δυαδικό σύστημα
 - 0,1 (bit – Binary digit, δυαδικό ψηφίο)
- Π.χ. ο αριθμός 1011_2

$$1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 11_{10}$$

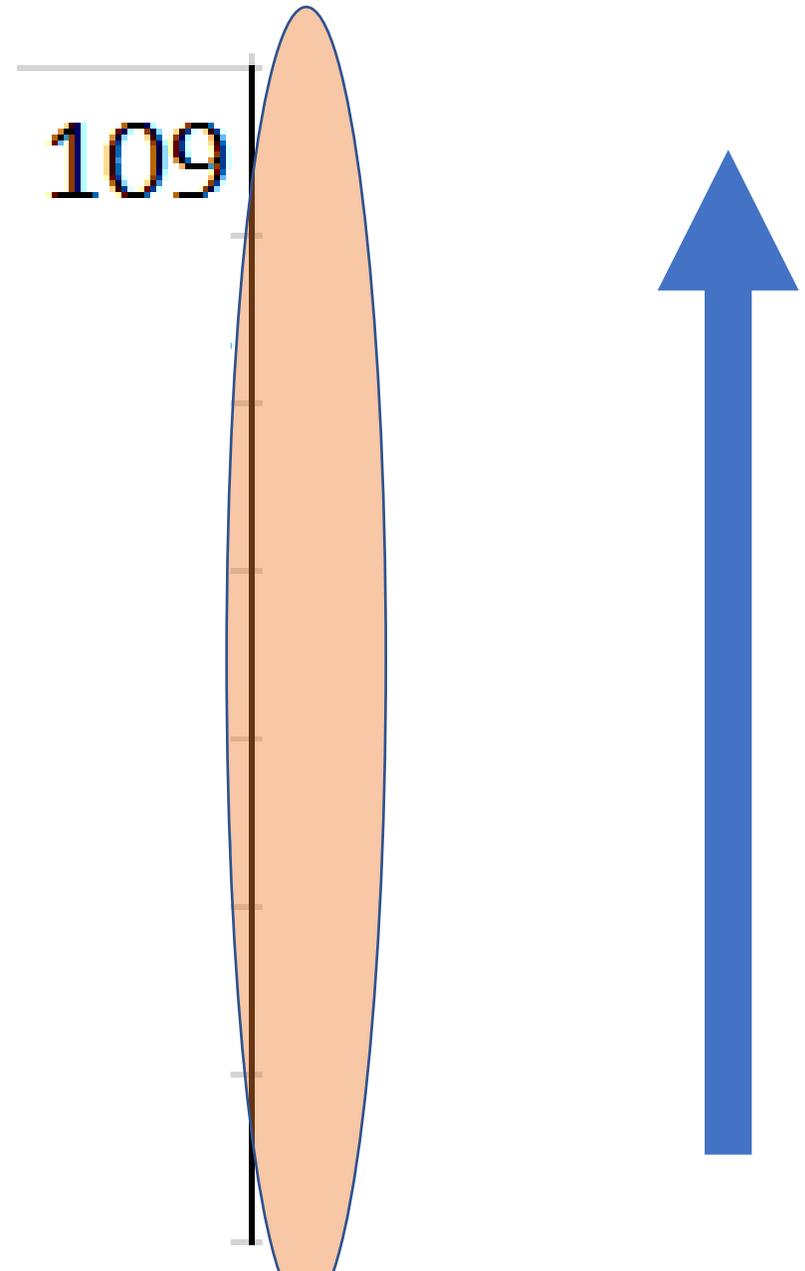
- Ψηφίο
- Βάση
- Θέση ή δύναμη

- $1024_{10} = 2^{10}$

Συστήματα αρίθμησης

- Μετατροπή δεκαδικό \rightarrow δυαδικό
- Π.χ. 109_{10}
- Ακέραια διαίρεση δια του 2
- Διαβάζουμε από κάτω προς τα πάνω

$$109_{10} = 1101101_2$$



Συστήματα αρίθμησης

- Πρόσθεση δυαδικών αριθμών
- $0 + 0 = 0$
- $1 + 0 = 0 + 1 = 1$
- $1 + 1 = 10$ (μηδέν και ένα το κρατούμενο)
- Π.χ.,

$$\begin{array}{r} \begin{array}{cccc} \downarrow & \downarrow & \downarrow & \downarrow \\ 1010_2 \\ +0111_2 \\ \hline 10001 \end{array} \end{array}$$

Συστήματα αρίθμησης

- Πολλαπλασιασμός δυαδικών αριθμών
- $0 * 0 = 0$
- $1 * 0 = 0 * 1 = 0$
- $1 * 1 = 1$
- Π.χ.,

$$\begin{array}{r} 1110_2 \\ \times 101_2 \\ \hline 1110 \\ 0000 \\ +1110 \\ \hline 1000110 \end{array}$$

Συστήματα αρίθμησης

- Δεκαεξαδικό σύστημα
 - 0,1,2,3,4,5,6,7,8,9,A=10,B=11,C=12,D=13,E=14,F=15
- Π.χ. ο αριθμός $1A2F_{16}$

$$1 * 16^3 + A * 16^2 + 2 * 16^1 + F * 16^0 = 6703_{10}$$

- Ψηφίο
- Βάση
- Θέση ή δύναμη

Συστήματα αρίθμησης

- Μετατροπή δεκαδικό -> δεκαεξαδικό
- Π.χ. 125_{10}
- Ακέραια διαίρεση δια του 16
- Διαβάζουμε από κάτω προς τα πάνω

$$\begin{array}{r|l} 125 & D \\ \hline 7 & 7 \end{array}$$

$$125_{10} = 7D_{16}$$

Συστήματα αρίθμησης

- Μετατροπή δυαδικό \leftrightarrow δεκαεξαδικό
- 2 \rightarrow 16, π.χ., 1011100110_2
- Χωρίζουμε τον αριθμό σε τετράδες (από τα δεξιά προς τα αριστερά)
 - 10, 1110, 0110
- Αν η πρώτη τετράδα δεν συμπληρώνεται συμπληρώνουμε με μηδενικά (από αριστερά) μέχρι να συμπληρωθεί
 - 0010, 1110, 0110
- Κοιτάμε στον πίνακα και αντικαθιστούμε τις αντίστοιχες τιμές
 - 2, E, 6 = $2E6_{16}$

HEX	DEC	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Συστήματα αρίθμησης

- Μετατροπή δεκαεξαδικό \leftrightarrow δυαδικό
- Π.χ., $1A2F_{16}$
- Για κάθε ψηφίο στον αριθμό αντικαθιστούμε με τον αντίστοιχο δυαδικό
 - $1A2F_{16} = 0001\ 1010\ 0010\ 1111_2$

HEX	DEC	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Ακρίβεια (Precision)

- Πόσα bit χρησιμοποιούμε για να εκφράσουμε έναν δυαδικό αριθμό
- Π.χ., ο αριθμός 43
- Με ακρίβεια 6 bit: 101011
- Με ακρίβεια 8 bit: 00101011
- Με ακρίβεια 32 bit: 0000000000000000000000000000000000101011
- Ο μέγιστος αριθμός που μπορούμε να εκφράσουμε με N bits:

$$2^N - 1$$

- Ο απαραίτητος αριθμός bit (N) για να μπορέσουμε να εκφράσουμε έναν αριθμό X:

$$N = \text{ceil}(\log_2 X)$$

Αφαίρεση

- Είναι μια πρόσθεση
- $X - Y = X + (-Y)$
- Αναπαράσταση δεδομένων: Πώς?

Αριθμοί με πρόσημο (Signed numbers)

- Ιδέα: το πρώτο bit για το πρόσημο τα υπόλοιπα για την απόλυτη τιμή
 - 0 για +, 1 για -
- Π.χ., ο αριθμός 13 (με 8 bits)
 - 00001101 = 13
 - 10001101 = -13
- Μειονεκτήματα
 - Πρόσθετη λογική για την πραγματοποίηση της αφαίρεσης
 - Δύο τιμές για το μηδέν (00000000, 10000000)
 - Αριθμοί που μπορούμε να αναπαραστήσουμε με N bits:
 - $-2^{N-1}-1 < x < 2^{N-1}-1$  2^N-1

Αριθμοί με πρόσημο (Signed numbers)

- **Συμπλήρωμα ως προς 2** (2's complement)
- Αντιστρέφουμε όλα τα ψηφία ενός αριθμού
- Προσθέτουμε 1
- Π.χ., ο αριθμός 5 με 4bit
- 0101 -> 1010 (αντιστροφή)
- 1010 -> 1011 (πρόσθεση 1)
- Στην ουσία το συμπλήρωμα ως προς 2 ενός αριθμού X (σε N bit) είναι το αποτέλεσμα της αφαίρεσης:

$$2^N - X$$

Αριθμοί με πρόσημο (Signed numbers)

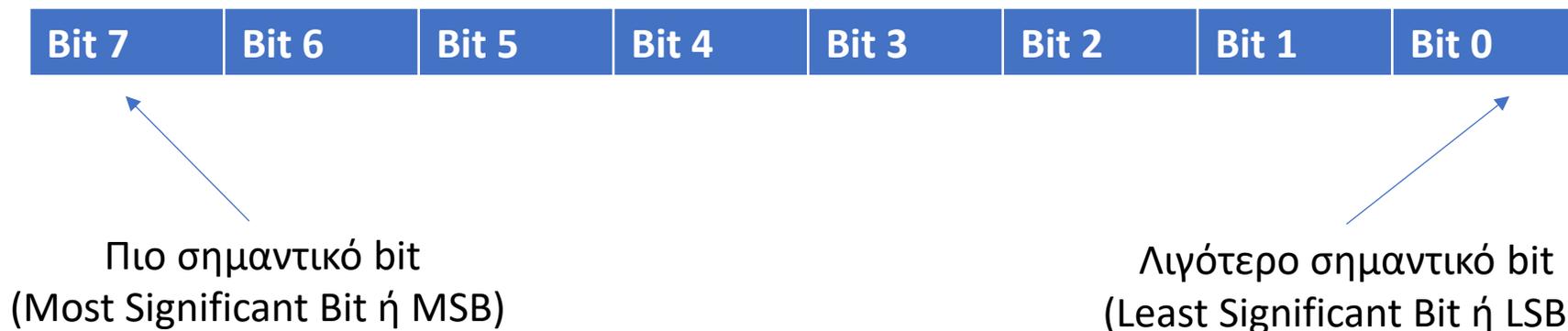
- **Συμπλήρωμα ως προς 2** (2's complement)
- Άλλος τρόπος υπολογισμού (Π.χ., αριθμός 110100)
- Ξεκινώντας από δεξιά προς τα αριστερά:
 - 1) Αντιγράφω όλα τα ψηφία μέχρι να βρω το πρώτο 1 (00)
 - 2) Αντιγράφω και το 1 (100)
 - 3) Αντιστρέφω όλα τα υπόλοιπα (001100)

Αντιστοιχία για 3 bit

BIN	Signed DEC	Unsigned DEC
000	0	0
001	1	1
010	2	2
011	3	3
100	-4	4
101	-3	5
110	-2	6
111	-1	7

Τύποι δεδομένων

- Bit
 - Η μικρότερη ποσότητα πληροφορίας (0,1)
- Byte
 - 8 bits σε σειρά
 - Ελάχιστη ποσότητα που ένας υπολογιστής μπορεί να χρησιμοποιήσει στις πράξεις



Τύποι δεδομένων

- Floats (Κινητής υποδιαστολής)

- Με πρόσημο

- Float or Single



- Double



Τύποι δεδομένων

- Chars (Χαρακτήρες)
 - 1 Byte
 - 'A', '1', '!'
- Strings (Αλφαριθμητικά)
 - 1 Byte/char
 - "Hello world!!! 123"

Τύποι δεδομένων

Όνομα	C	VBA
Boolean	- (int)	Boolean
Byte	char	Byte
Integer	int	Integer
Long	long	Long
Floating point	float	Single
Floating point (double precision)	double	Double
Alphanumeric (Strings)	char*	String

Άλγεβρα Boole (Μπουλ)

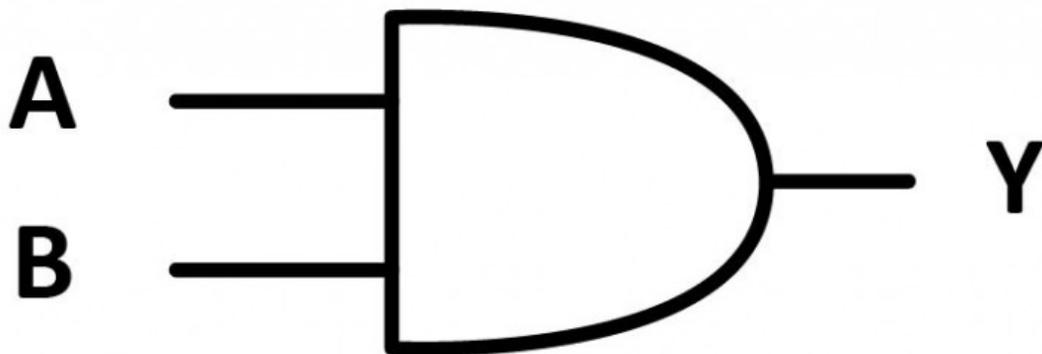
- Αριθμητική βασισμένη στο δυαδικό σύστημα
- Υποστηρίζει 3 βασικές πράξεις:
 - ΚΑΙ (σύζευξη)
 - Ή (διάζευξη)
 - ΟΧΙ (Άρνηση)
- Πίνακες αλήθειας

Άλγεβρα Boole (Μπουλ)

1	0
(Ν)ΑΙ	(Ο)ΧΙ
(Α)ΛΗΘΕΙΑ	(Ψ)ΕΜΑΤΑ
(Σ)ΩΣΤΟ	(Λ)ΑΘΟΣ
(Υ)ΕΣ	(Ν)Ο
(Τ)ΡΥΕ	(F)ΑΛΣΕ

Άλγεβρα Boole (Μπουλ)

- ΚΑΙ (AND)

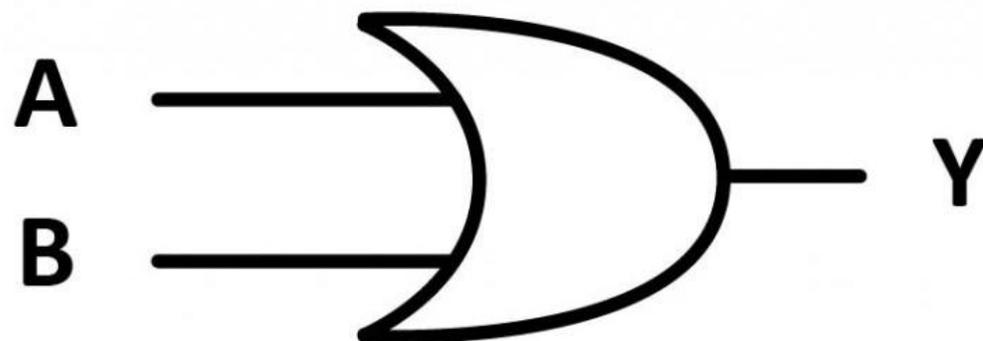


Πίνακας αλήθειας

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Άλγεβρα Boole (Μπουλ)

- Ή (OR)

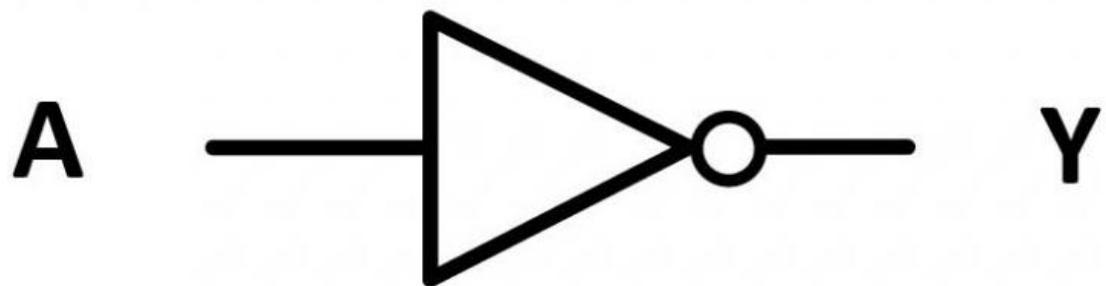


Πίνακας αλήθειας

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Άλγεβρα Boole (Μπουλ)

- Όχι (NOT)



Πίνακας αλήθειας

A	Y
0	1
1	0

Αλγόριθμοι: Τι είναι;

- Ορισμένη ακολουθία εντολών για την επίλυση ενός προβλήματος
 - Πεπερασμένος χώρος και χρόνος
- Απαραίτητες ιδιότητες
 - Τιμές εισόδου (Input): Πρέπει να προέρχονται από ένα ορισμένο σύνολο τιμών
 - Τιμές εξόδου (Output): Πρέπει να προέρχονται από ένα ορισμένο σύνολο τιμών και πρέπει να αποτελούν την λύση του προβλήματος
 - Καθοριστικότητα: Όλα τα βήματα του αλγορίθμου πρέπει να είναι ορισμένα με ακρίβεια
 - Περατότητα: Ο αλγόριθμος πρέπει να τελειώνει μετά την εκτέλεση ενός πεπερασμένου αριθμού βημάτων
 - Αποτελεσματικότητα: Κάθε βήμα του αλγορίθμου πρέπει να είναι απλό και να τελειώνει σε πεπερασμένο χρόνο. Απλό σημαίνει ότι το βήμα δεν μπορεί να «σπάσει» σε μικρότερα βήματα.

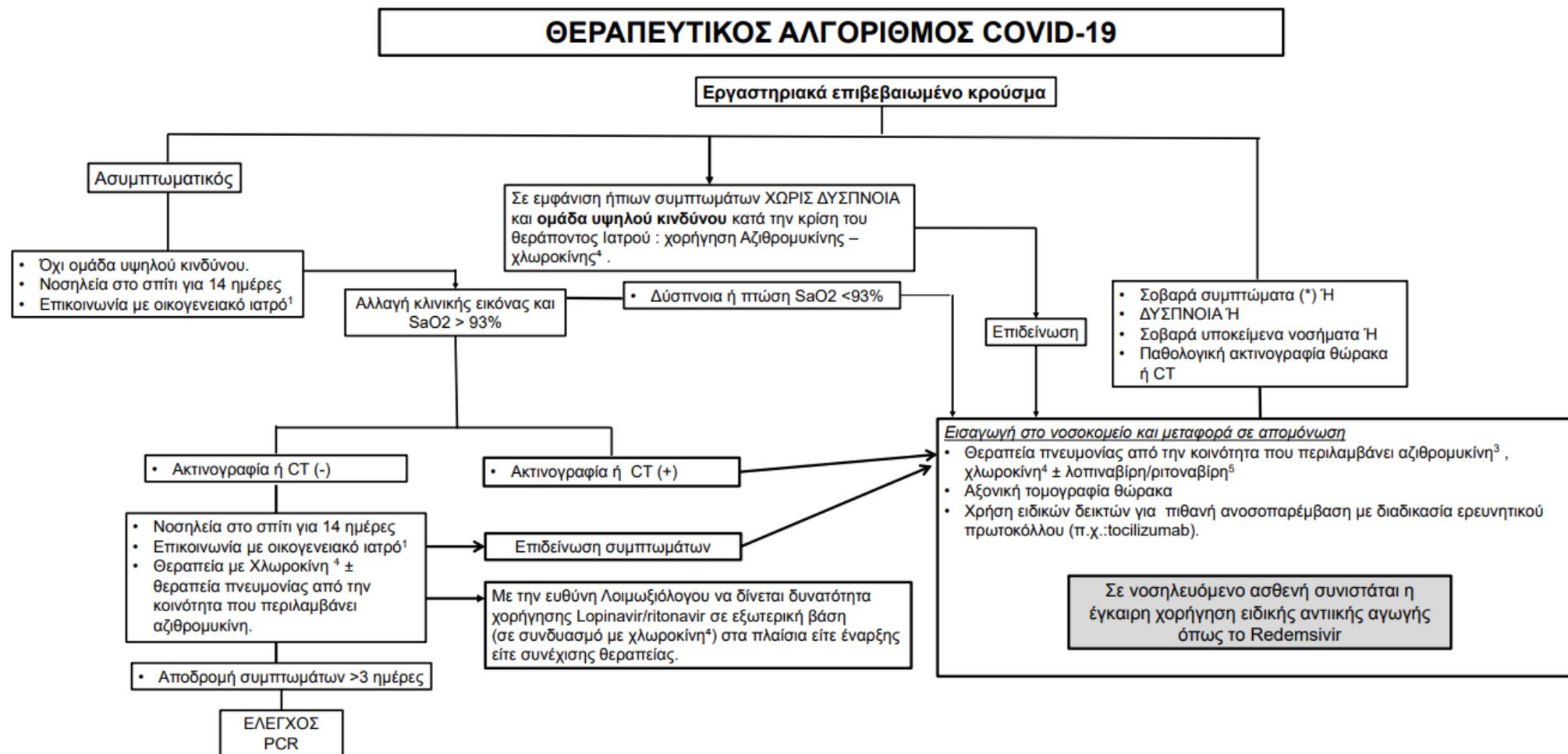
Διεργασίες – βήματα – εντολές - σύμβολα

- Αρχή (Έναρξη)
 - Το πρώτο βήμα που εκτελεί ένας αλγόριθμος
- Τέλος (Λήξη)
 - Το τελευταίο βήμα που εκτελεί ένας αλγόριθμος
- Βέλη (Ροή)
 - Ο αλγόριθμος μετακινείται από το ένα βήμα στο επόμενο
 - ΠΡΟΣΟΧΗ: Μόνο βέλη μίας κατεύθυνσης
- Input (Είσοδος)
 - Εισάγουμε δεδομένα στον αλγόριθμο
- Output (Έξοδος)
 - Εξάγουμε δεδομένα από τον αλγόριθμο
- Επεξεργασία
- Συνθήκη (Έλεγχος, Απόφαση)
 - Ο αλγόριθμος ελέγχει αν μία συνθήκη είναι αληθής ή ψευδής και συνεχίζει την ροή ανάλογα με το αποτέλεσμα

Διαγράμματα ροής

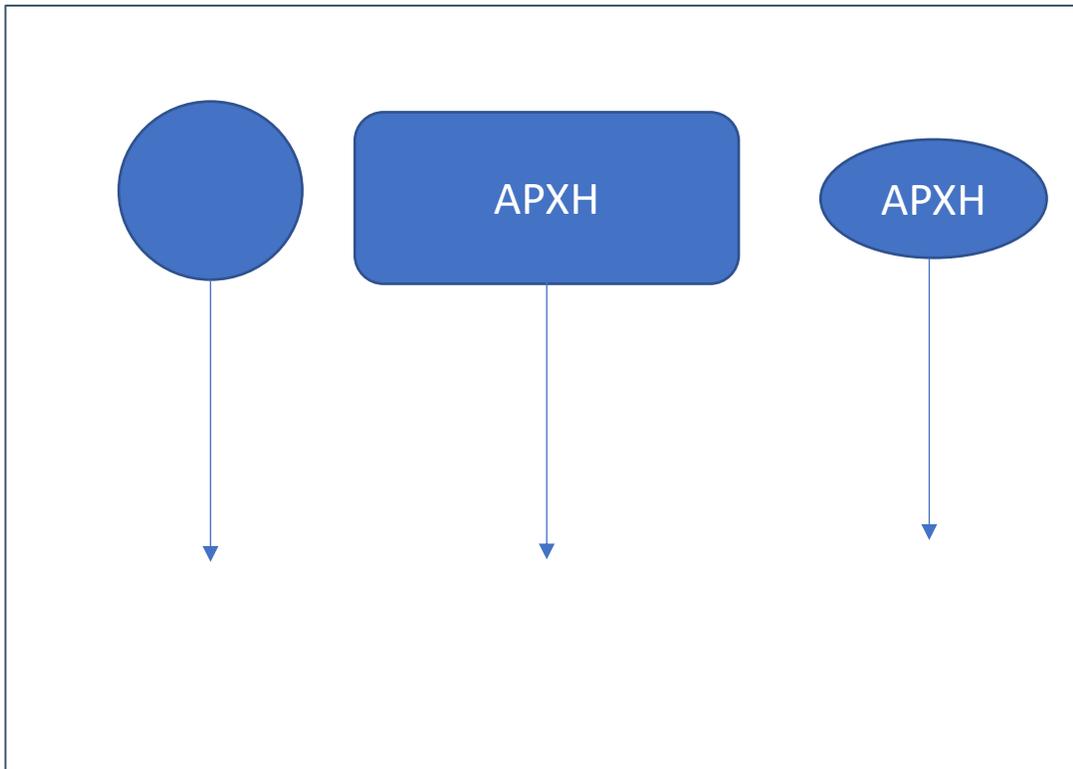
- Γραφική αναπαράσταση ενός αλγορίθμου
- Σύμβολα που αντιστοιχούν σε κάθε βήμα
- Βέλη που αντιπροσωπεύουν την ροή του αλγορίθμου

Διαγράμματα ροής – Παράδειγμα COVID

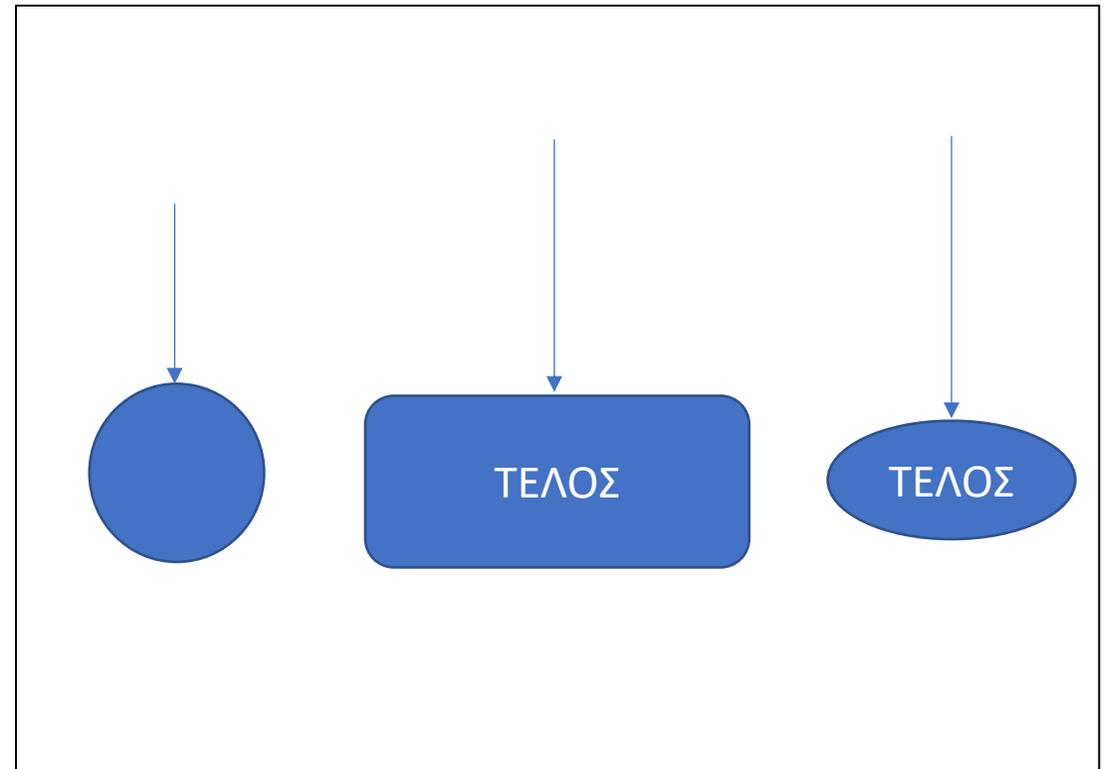


Αρχή και Τέλος

Αρχή



Τέλος



Είσοδος και Έξοδος

Είσοδος



Έξοδος

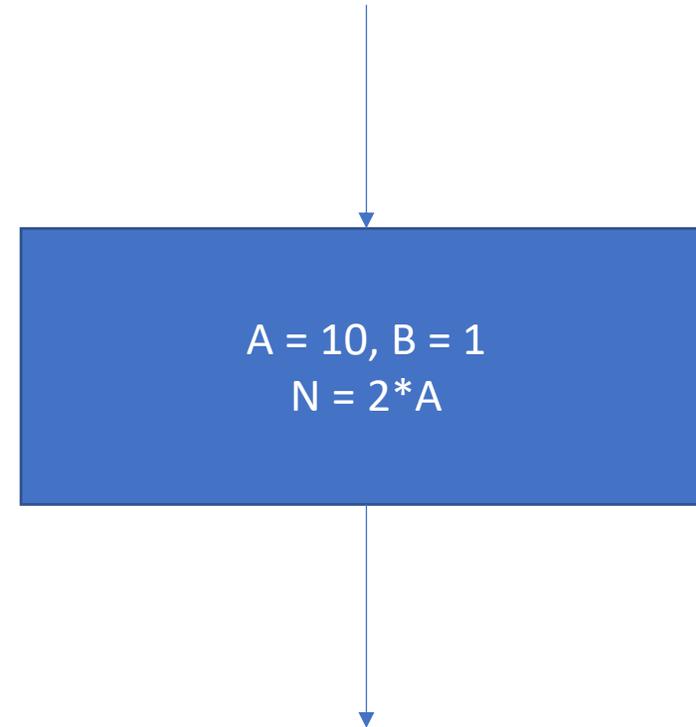


Επεξεργασία

- Μεταβλητές
 - Χώρος στην μνήμη που αποθηκεύουμε προσωρινά τα δεδομένα
 - Όνομα
 - Τιμή
 - Αν δεν ορίζεται θεωρούμε ότι είναι μηδέν, την θέτουμε με το ίσον (=)
 - Τύπος
 - Ο τύπος του δεδομένου που θα αποθηκεύει η μεταβλητή
 - Π.χ., Ακέραιος αριθμός, Αλφαριθμητικό κλπ.
 - Θα μιλήσουμε για τον τύπο όταν μπορούμε στον προγραμματισμό
 - Π.χ.
 - **Εγγραφή:** $N = 1$ ή $N <- 1$: Τοποθετεί την τιμή 1 στην μεταβλητή με το όνομα N
 - **Ανάγνωση:** (π.χ. σε ένα βήμα output) Εκτύπωσε το N στην οθόνη
 - **Ανάγνωση και εγγραφή με διαφορετικές μεταβλητές:** $N = M + 3$ ή $N <- M + 3$: Τοποθετεί στην μεταβλητή N την τιμή της M αφού προσθέσει 3
 - **Ανάγνωση και εγγραφή στην ίδια μεταβλητή:** $N = N + 3$ ή $N <- N + 3$: Τοποθετεί στην μεταβλητή N την προηγούμενη τιμή της N αφού προσθέσει 3
 - **ΠΡΟΣΟΧΗ: ΠΡΙΝ ΤΗΝ ΑΝΑΓΝΩΣΗ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ ΘΑ ΠΡΕΠΕΙ ΝΑ ΈΧΕΙ ΟΡΙΣΤΕΙ ΤΟ ΟΝΟΜΑ ΚΑΙ Η ΤΙΜΗ ΤΗΣ**
 - Δεν ισχύει το ίδιο για την εγγραφή

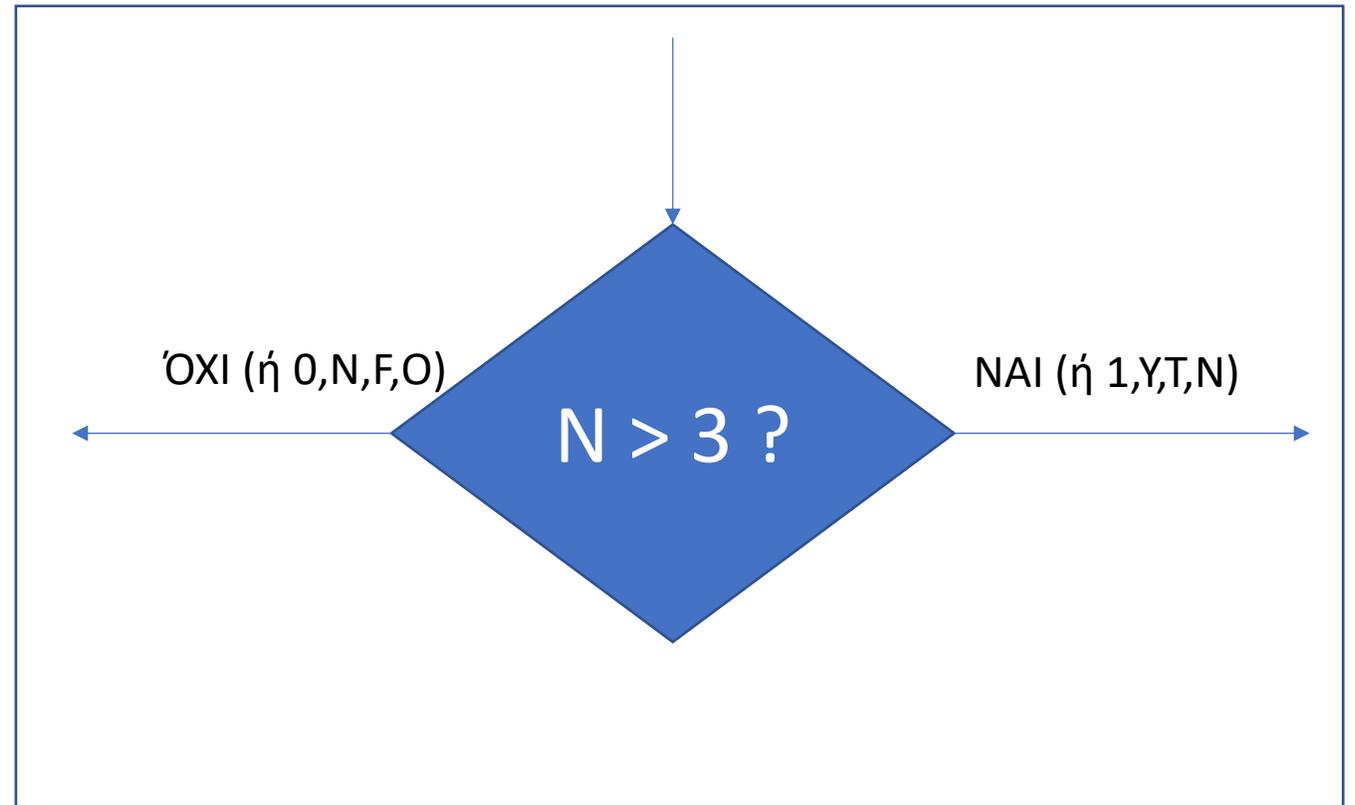
Επεξεργασία

- Κατά την επεξεργασία ο αλγόριθμος εκτελεί την εντολή που βρίσκεται μέσα
- Αν υπάρχουν περισσότερες εντολές τις εκτελεί με σειρά από αριστερά προς τα δεξιά και από πάνω προς τα κάτω



Συνθήκη

- 1 Βέλος μπαίνει
- 2 Βέλη βγαίνουν
- Αν η συνθήκη ελέγχου μέσα στον ρόμβο είναι αληθής η ροή ακολουθεί το ΝΑΙ αλλιώς ακολουθεί το ΌΧΙ



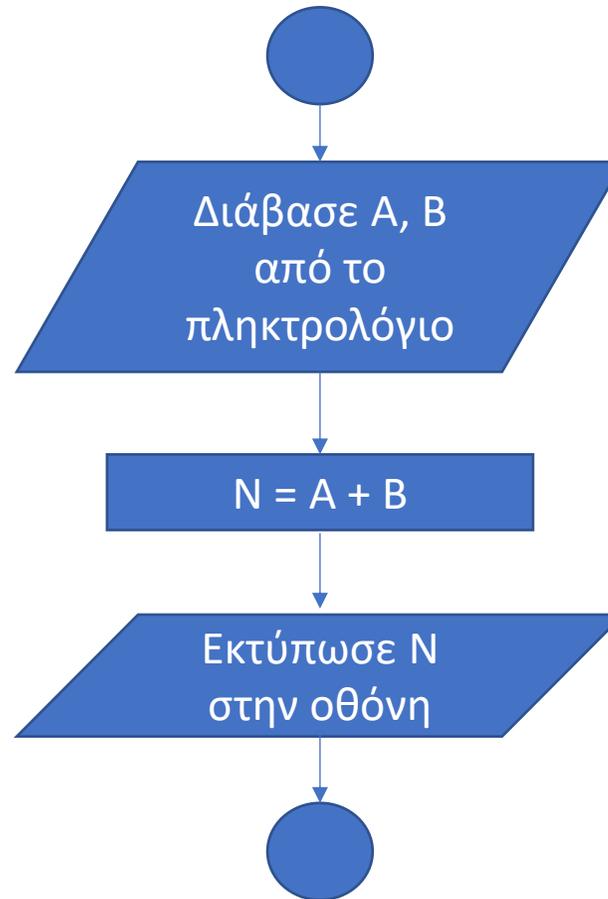
Συνθήκες και πράξεις

- Αριθμητικές Πράξεις: +, -, *, /, =
- Συνθήκες

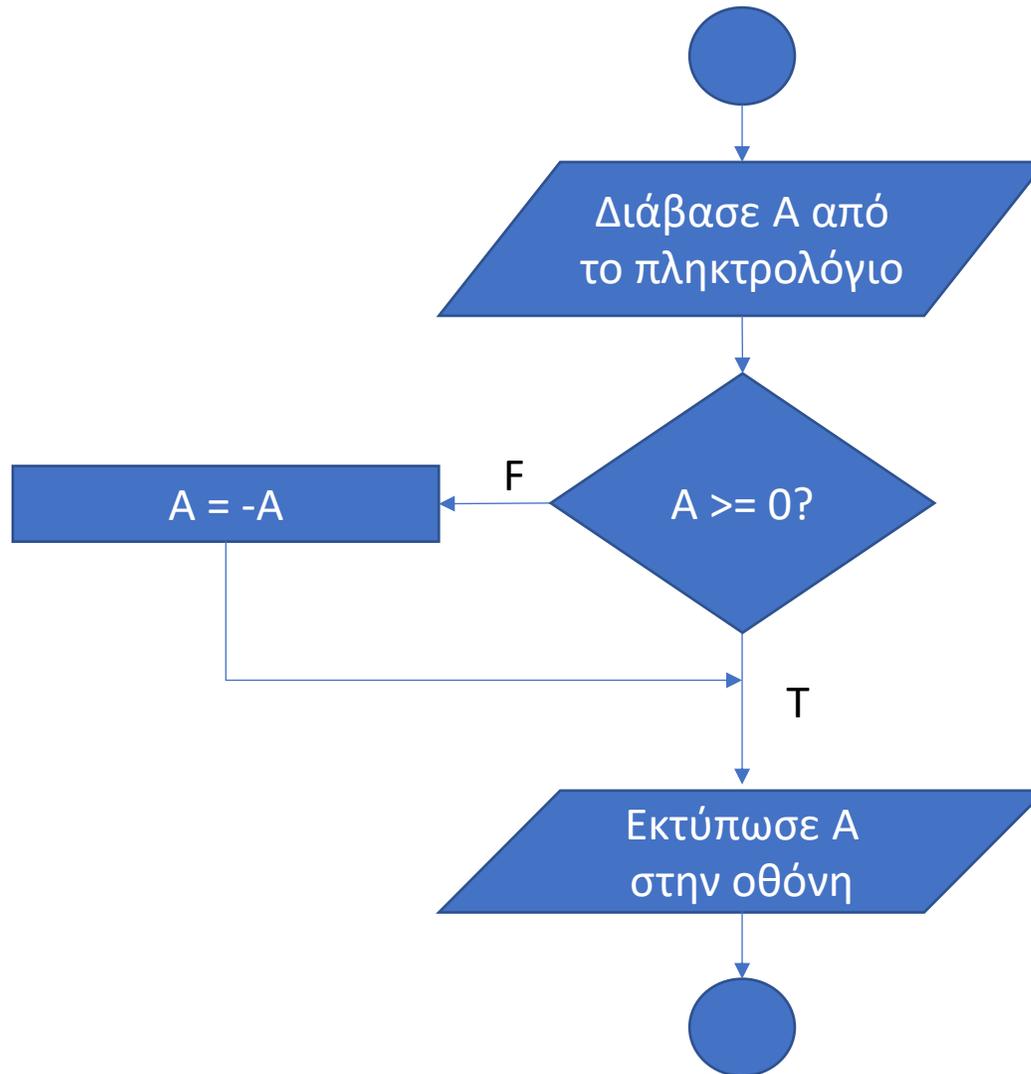
Συνθήκη	Σύμβολο
Ισότητα	A == B
Διαφορά	A != B (<>)
Ανισότητα	A > B, A >= B, A < B, A <= B
AND	A && B
OR	A B

- A και B στον παραπάνω πίνακα μπορεί να είναι και σύνθετες παραστάσεις
 - Π.χ. (A < 2) && (A != 0) (Το A πρέπει να είναι μικρότερο του 2 ΚΑΙ διάφορο του 0)

Παράδειγμα: Πρόσθεση δύο αριθμών



Παράδειγμα: Απόλυτη τιμή

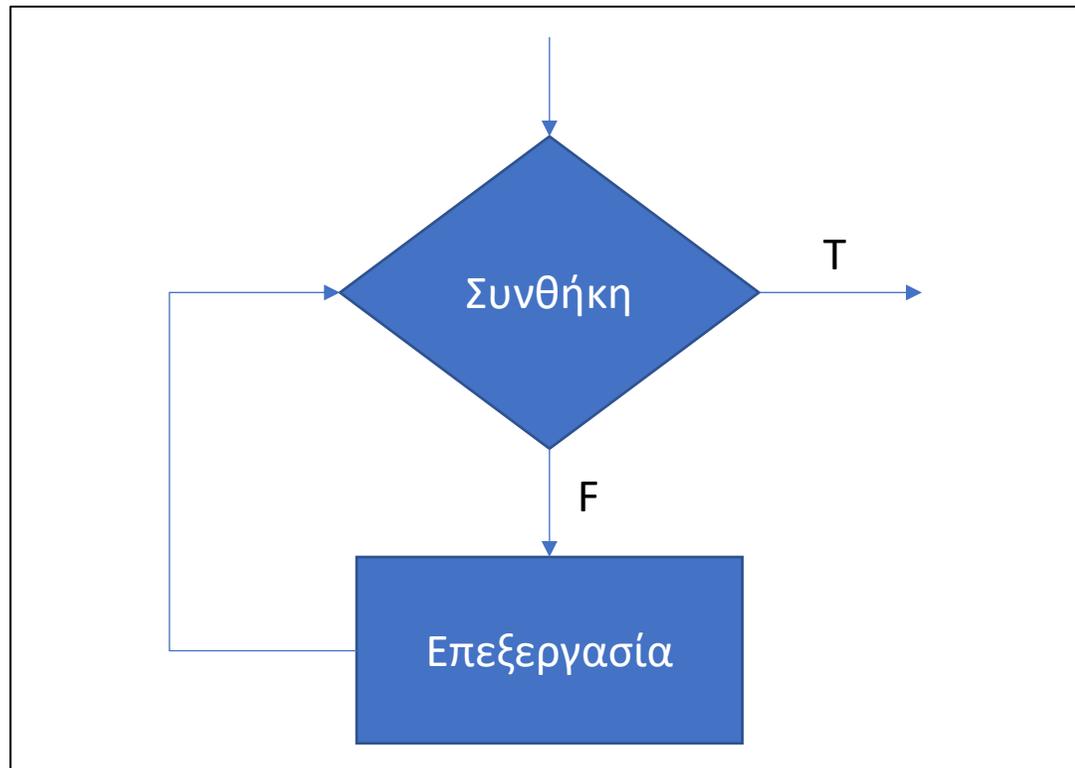


Βρόχοι (Loops)

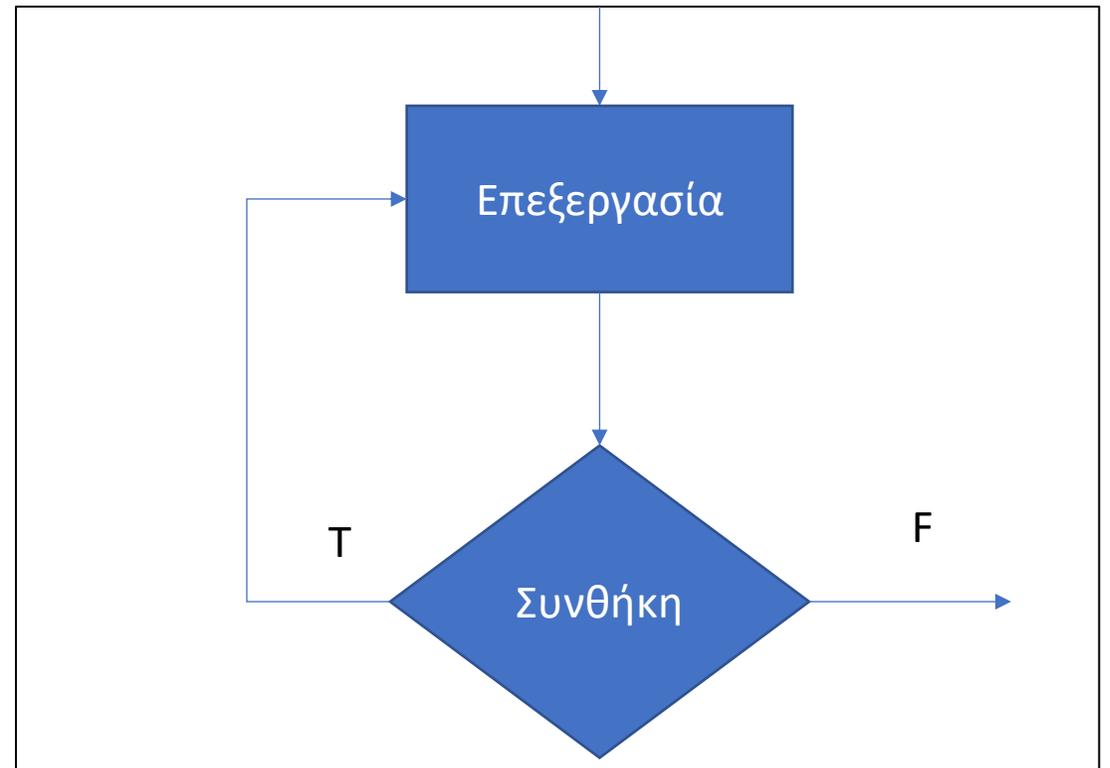
- Επαναλαμβανόμενες εργασίες
 - Η ροή του προγράμματος επιστρέφει σε προηγούμενο σημείο στο διάγραμμα ροής
- Ο αριθμός των επαναλήψεων εξαρτάται από την συνθήκη
- 2 είδη:
 - Βρόχος συνθήκης εισόδου
 - Βρόχος συνθήκης εξόδου

Βρόχοι

Εισόδου (for, while) : Πρώτα ελέγχουμε αν η συνθήκη είναι αληθής και μετά μπαίνουμε στον βρόχο

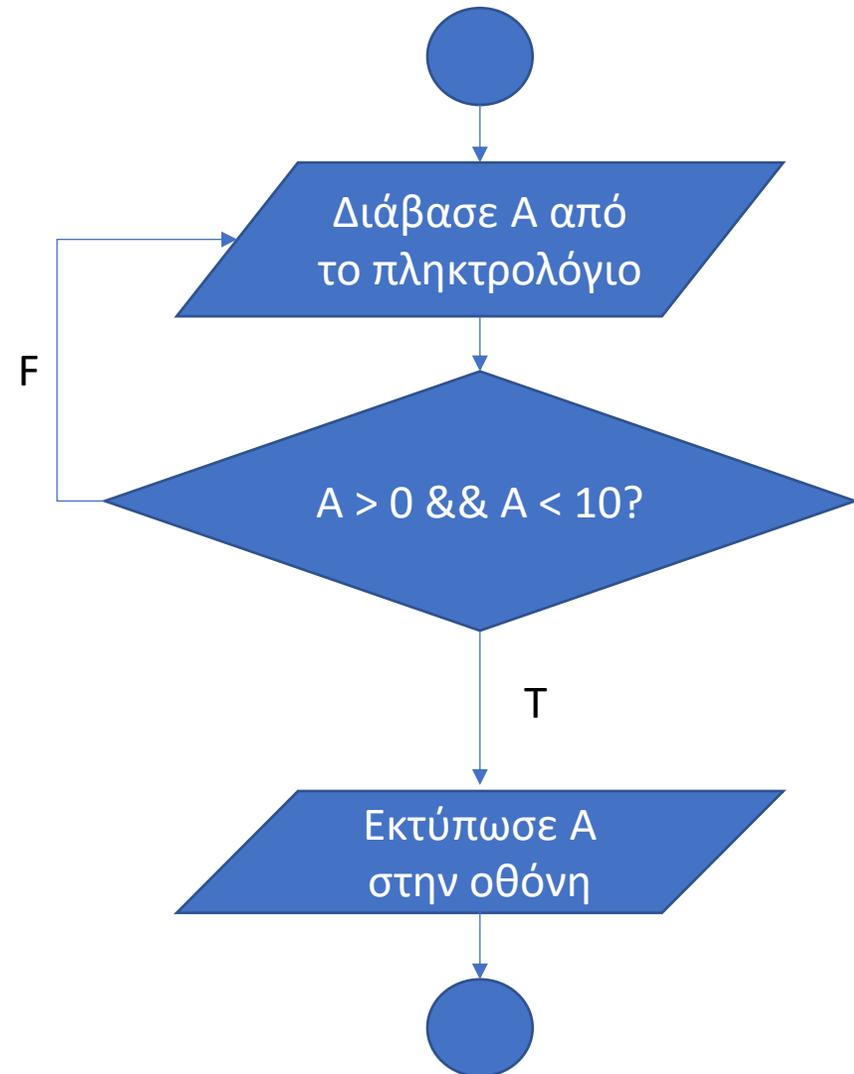


Εξόδου (do-while): Πρώτα μπαίνουμε στον βρόχο και μετά ελέγχουμε αν η συνθήκη είναι αληθής



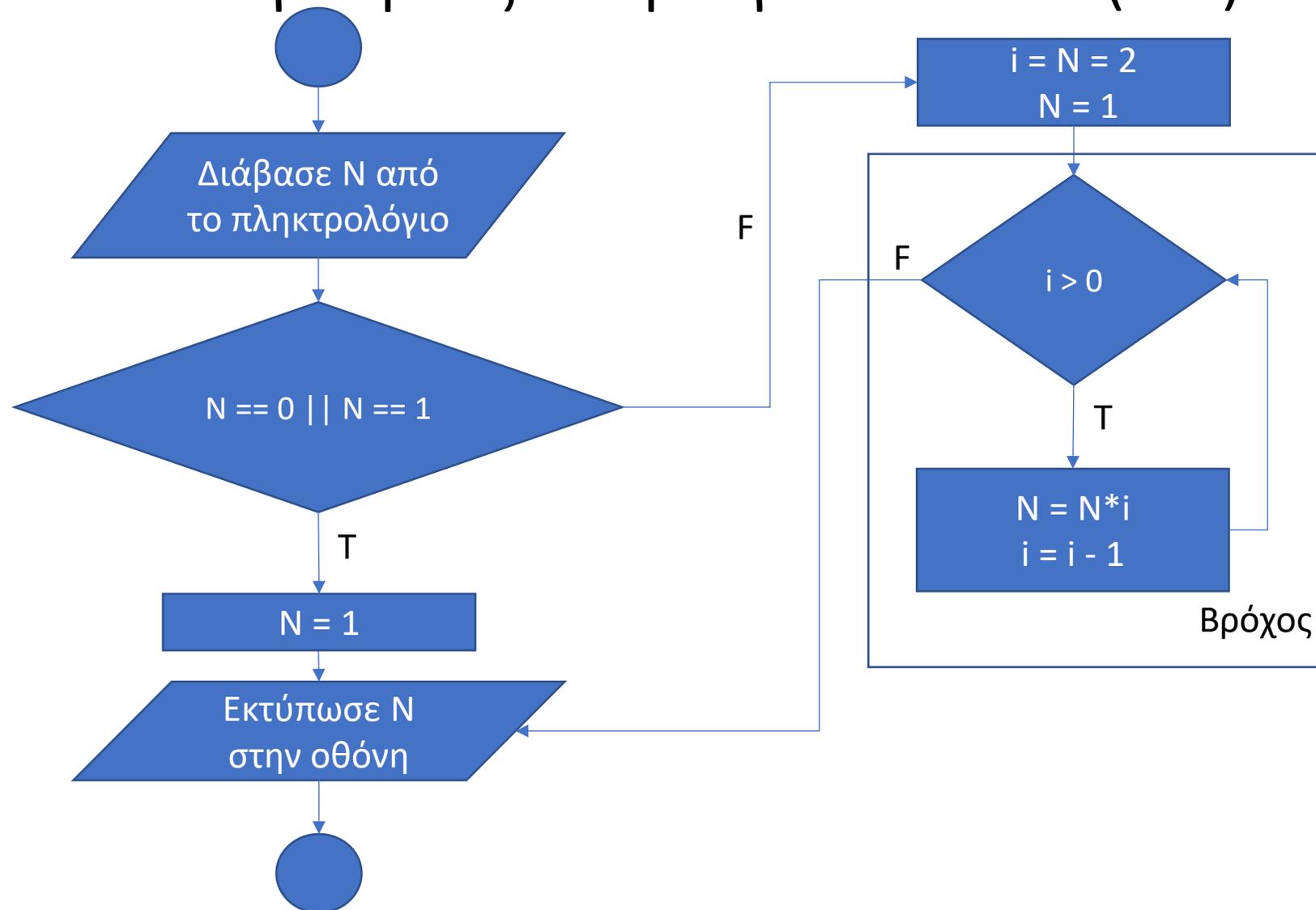
Παράδειγμα: Έλεγχος Input από το πληκτρολόγιο

- Βρόχος συνθήκης εξόδου
- Πρέπει να σιγουρευτούμε αν ο χρήστης πάτησε στο πληκτρολόγιο έναν αριθμό από το 1 μέχρι το 9
- Αν όχι, τότε ζητά από τον χρήστη να εισάγει έναν αριθμό εκ νέου
- Αν ναι τότε εκτύπωσε τον αριθμό και τέλος

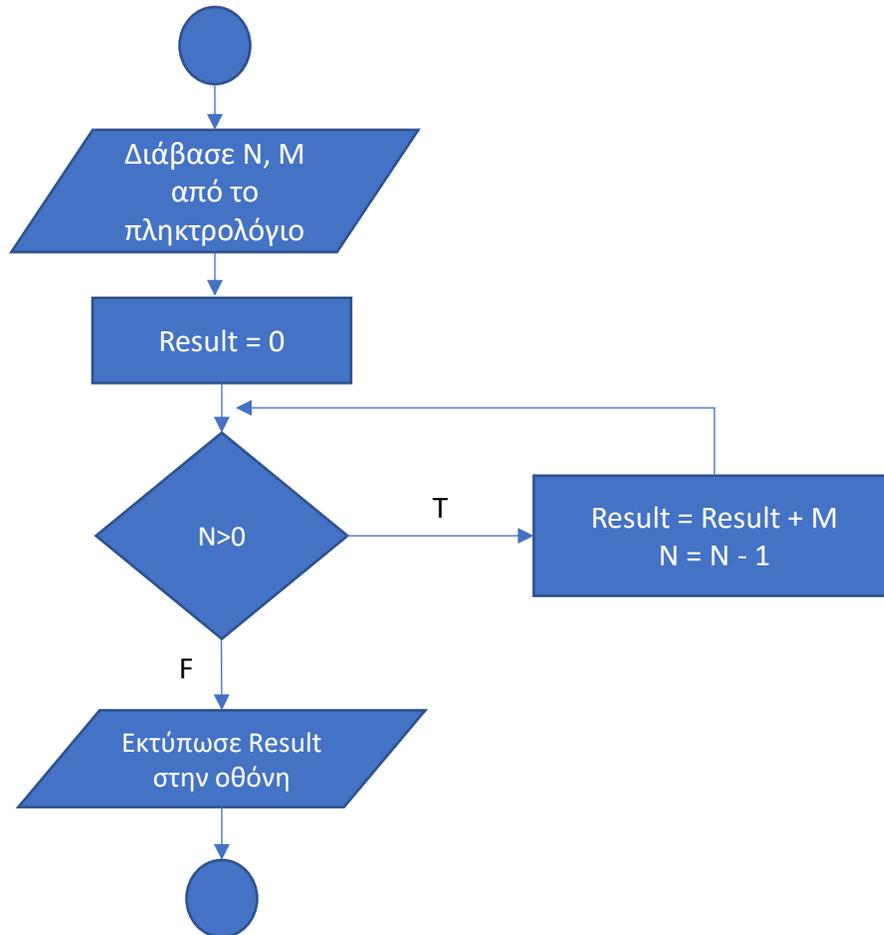


Παράδειγμα: Υπολογισμός παραγοντικού (N!)

- Βρόχος συνθήκης εισόδου
- Εισάγουμε έναν αριθμό N από το πληκτρολόγιο
- Αν $N == 0$ ή $N == 1$ τότε το αποτέλεσμα είναι 1
- Αλλιώς το αποτέλεσμα είναι:
 - $N! = 1 * 2 * 3 * \dots * (N-1) * N$



Πολ/σμός 2 αριθμών



Ασκήσεις

- Σχεδιάστε έναν αλγόριθμο που να ζητάει να εισαχθούν 2 αριθμοί από το πλ/γιο και να δείχνει στην οθόνη ποιος από τους 2 είναι μεγαλύτερος.
 - Μεταβάλλετε τον αλγόριθμο ώστε να κάνει το ίδιο για 3 αριθμούς
- Σχεδιάστε έναν αλγόριθμο που να προσομοιώνει τη λειτουργία ενός υπολογιστή τσέπης (calculator). Στην αρχή ζητάει να εισαχθεί 1 σύμβολο από το πλ/γιο (+, -, *, /), έπειτα ζητάει να εισαχθούν 2 αριθμοί από το πληκτρολόγιο και εκτελεί την πράξη που ζητήθηκε.
 - Μεταβάλλετε τον αλγόριθμο ώστε να ελέγχει στην περίπτωση της διαίρεσης ότι ο διαιρέτης είναι διάφορος του 0.

Ασκήσεις

- Σχεδιάστε έναν αλγόριθμο που να ζητάει να εισαχθεί 1 θετικός αριθμός από το πλ/γιο και να δείχνει στην οθόνη όλους τους ακέραιους αριθμούς από το 0 μέχρι και τον αριθμό που εισήχθη
 - Μεταβάλλετε τον αλγόριθμο ώστε να συμπεριλάβετε και τους αρνητικούς αριθμούς
- Σχεδιάστε έναν αλγόριθμο που να ζητάει να εισαχθεί 1 θετικός αριθμός N από το πλ/γιο. Στη συνέχεια ο αλγόριθμος ζητάει από τον χρήστη να εισάγει N αριθμούς και στο τέλος δείχνει στην οθόνη το άθροισμα όλων των αριθμών που εισήχθησαν.
 - Μεταβάλλετε τον αλγόριθμο με τέτοιο τρόπο ώστε να σταματάει μόλις ο χρήστης εισάγει το 0

Ερωτήσεις;