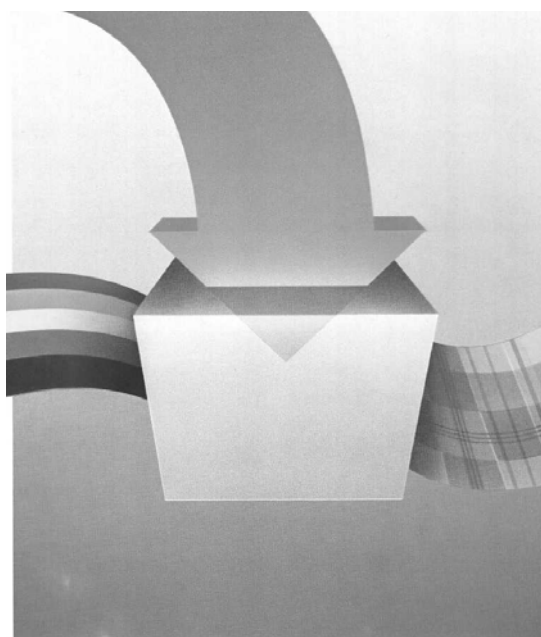


Τ.Ε.Ι. ΚΑΒΑΛΑΣ  
ΠΑΡΑΡΤΗΜΑ ΔΡΑΜΑΣ  
ΤΜΗΜΑ ΔΑΣΟΠΟΝΙΑΣ

# Εισαγωγή στο Λειτουργικό Σύστημα MS-DOS και στη Γλώσσα Προγραμματισμού BASIC



ΣΕΠΤΕΜΒΡΙΟΣ 1997

Δρ. Καραμπετάκης Νικόλαος  
Επιστημονικός Συνεργάτης

<b>ΕΙΣΑΓΩΓΗ</b>	<b>4</b>
<b>ΓΝΩΡΙΜΙΑ ΜΕ ΤΟΝ ΗΛΕΚΤΡΟΝΙΚΟ ΥΠΟΛΟΓΙΣΤΗ</b>	<b>4</b>
ΔΟΜΗ ΤΟΥ Η/Υ.	5
ΕΞΩΤΕΡΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ Η/Υ.	6
ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΕΝΤΡΙΚΗΣ ΜΟΝΑΔΑΣ ΤΟΥ Η/Υ.	10
ΛΟΓΙΣΜΙΚΟ Η SOFTWARE	13
<b>ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ MS-DOS</b>	<b>15</b>
ΑΡΧΕΙΑ – ΟΝΟΜΑΣΙΕΣ ΑΡΧΕΙΩΝ	15
ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΧΕΙΩΝ ΤΟΥ MS-DOS	17
H ΕΝΤΟΛΗ CLS	18
H ΕΝΤΟΛΗ DATE	19
H ΕΝΤΟΛΗ TIME	20
H ΕΝΤΟΛΗ COPY CON	20
H ΕΝΤΟΛΗ TYPE	21
H ΕΝΤΟΛΗ DIR	21
H ΕΝΤΟΛΗ RENAME	24
H ΕΝΤΟΛΗ COPY	24
H ΕΝΤΟΛΗ DEL	25
ΟΙ ΧΑΡΑΚΤΗΡΕΣ ΜΠΑΛΑΝΤΕΡ (WILDCARDS) * ΚΑΙ ?	26
ΑΣΚΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ	28
ΥΠΟΚΑΤΑΛΟΓΟΙ - ΜΟΝΟΠΑΤΙΑ	29
ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΤΑΛΟΓΩΝ	32
H ΕΝΤΟΛΗ MD	33
H ΕΝΤΟΛΗ CD	34
H ΕΝΤΟΛΗ TREE	35
H ΕΝΤΟΛΗ RD	36
H ΕΝΤΟΛΗ DELTREE	37
ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΧΕΙΩΝ ΚΑΙ ΚΑΤΑΛΟΓΩΝ.	39
ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΣΚΩΝ	42
H ΕΝΤΟΛΗ FORMAT	42
H ΕΝΤΟΛΗ DISKCOPY	42
ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΣΤΟ MS-DOS.	43
<b>Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ BASIC</b>	<b>46</b>
Η έννοια του προγραμματισμού	47
Είσοδος-έξοδος στο περιβάλλον της BASIC	48
Κανόνες γραφής προγράμματος στην BASIC	48
Εντολές της BASIC	49
Εντολές ελέγχου στην BASIC	49
H εντολή CLS	49
H εντολή DELETE	50
H εντολή LIST	50
H εντολή LOAD	51
H εντολή NEW	52
H εντολή RENUM	52
H εντολή RUN	53

Η εντολή SAVE	53
Το αλφάβητο της BASIC	54
Σταθερές-Μεταβλητές στην BASIC	54
Εκτέλεση αριθμητικών πράξεων στην BASIC	58
Εκτέλεση αλφαριθμητικών πράξεων στην BASIC	61
Εντολές εισόδου – αντικατάστασης - εξόδου στην BASIC	61
Η εντολή αντικατάστασης LET	62
Εντολές εισόδου	63
Α) Η εντολή INPUT.	63
Β) Οι εντολές READ-DATA-RESTORE.	65
Εντολές εξόδου	69
Α) Η εντολή PRINT.	69
Β) Η εντολή PRINT USING.	70
Άλλες εντολές	71
Η εντολή CLS	71
Η εντολή STOP	71
Η εντολή END	71
Σχόλια μέσω της REM	72
Παραδείγματα απλών προγραμμάτων	72
Επαναληπτικές Ασκήσεις στις Εντολές Εισόδου-Εξόδου	76
Εντολές ελέγχου και διακλαδώσεων.	79
Η εντολή GOTO.	80
Η εντολή IF-THEN-ELSE.	82
Επαναληπτικές Ασκήσεις στις Εντολές Ελέγχου και Διακλάδωσης	89
Ανακύκλωση και χρήση μετρητών σε προγράμματα.	90
Επαναληπτικές ασκήσεις στην ανακύκλωση και χρήση μετρητών.	103
Εντολές Επαναλήψεων	104
Η εντολή FOR-NEXT	106
Η εντολή WHILE-WEND	119
Επαναληπτικές Ασκήσεις στις εντολές FOR-NEXT και WHILE-WEND	122
ΠΙΝΑΚΕΣ	127
Ορισμός Πίνακα	128
Εισαγωγή τιμών σε έναν πίνακα	129
Εμφάνιση τιμών ενός πίνακα	131
Επεξεργασία των στοιχείων του πίνακα.	131
Α) Εύρεση πλήθους – γινομένου - αθροίσματος στοιχείων του πίνακα που ικανοποιούν κάποια συνθήκη (ή όχι).	132
Β) Εύρεση ελαχίστου-μεγίστου.	136
Γ) Ταξινόμηση στοιχείων πίνακα.	141
Επαναληπτικές Ασκήσεις στους Πίνακες	145
BIBΛΙΟΓΡΑΦΙΑ	149

## ΕΙΣΑΓΩΓΗ

Είναι γνωστή σε όλους η συνεισφορά των ηλεκτρονικών υπολογιστών σε όλους τους τομείς των επιστημών. Για τον σκοπό αυτό θεωρούμε ότι είναι απαραίτητη η εξοικείωση των φοιτητών του Τμήματος με τον ίδιο τον ηλεκτρονικό υπολογιστή καθώς και με τον τρόπο χειρισμού του. Σκοπός λοιπόν αυτού του συγγράμματος, είναι α) η συνοπτική περιγραφή του ηλεκτρονικού υπολογιστή, β) η περιγραφή ενός αποσπασματικού μέρους εντολών του λειτουργικού συστήματος MS-DOS που μας βοηθάει να επικοινωνούμε με τα διάφορα μέρη του Η/Υ, καθώς και γ) η ανάπτυξη ικανοτήτων προγραμματισμού στην γλώσσα BASIC. Μια πιο εμπειριστατωμένη μελέτη πάνω στην θεωρία των Η/Υ δίνεται από το βιβλίο του κ. Χουρμουζιάδη “Εισαγωγή στους Η/Υ”. Θα ήθελα να ευχαριστήσω τον φίλο και συνεργάτη κ. Ν. Σωτηριάδη για την πολύτιμη συνεισφορά του στην δημιουργία των σημειώσεων που ακολουθούν.

## ΓΝΩΡΙΜΙΑ ΜΕ ΤΟΝ ΗΛΕΚΤΡΟΝΙΚΟ ΥΠΟΛΟΓΙΣΤΗ

Ο ηλεκτρονικός υπολογιστής (Η/Υ) είναι μια ηλεκτρονική συσκευή η οποία επεξεργάζεται με μεγάλη ακρίβεια και ταχύτητα έναν τεράστιο όγκο πληροφοριών. Η βασική του διαφορά από άλλες ηλεκτρονικές συσκευές βρίσκεται στην δυνατότητα προγραμματισμού του. Είναι δυνατό ο ίδιος Η/Υ, διαμέσου διαφορετικών *προγραμμάτων* να είναι το ίδιο χρήσιμος σε έναν Δασοπόνο, σε έναν Μαθηματικό, σε έναν Μηχανικό κ.τ.λ.

Ο Η/Υ αποτελείται από το *Hardware* ή *υλικό* και το *Software* ή *λογισμικό*. Με τον όρο *Hardware* αναφερόμαστε στις διάφορες μονάδες εισόδου, επεξεργασίας και εξόδου του Η/Υ όπως για πρδ. το πληκτρολόγιο, η κεντρική μονάδα επεξεργασίας, η οθόνη κ.τ.λ. ενώ με τον όρο *Software* στα προγράμματα τα οποία κατευθύνουν τον Η/Υ. Τα προγράμματα αυτά αποτελούν ένα σύνολο από οδηγίες γραμμένες κατά τέτοιο τρόπο ώστε να είναι άμεσα κατανοητές από τον Η/Υ.

Πρωταρχικός σκοπός του φοιτητή είναι να εξοικειωθεί με το υλικό του Η/Υ και για τον σκοπό αυτό παραθέτουμε α) μια σύντομη περιγραφή της δομής του, β) κάποιες φωτογραφίες των διαφόρων μερών του υλικού του Η/Υ και γ) μια συνοπτική περιγραφή των μονάδων αυτών. Μια πιο λεπτομερής

περιγραφή όλων εκείνων των μονάδων που αποτελούν το υλικό του Η/Υ θα έχει την δυνατότητα ο φοιτητής να βρει στο αντίστοιχο βιβλίο της Θεωρίας. Δευτερεύον στόχος του φοιτητή είναι να εξοικειωθεί με το λογισμικό του Η/Υ που θα του δώσει άμεσα την δυνατότητα να επικοινωνήσει με τον Η/Υ και στην συνέχεια να τον προγραμματίσει.

## ΔΟΜΗ ΤΟΥ Η/Υ.

Στο παρακάτω σχήμα φαίνονται τα βασικά μέρη από τα οποία αποτελείται ο Η/Υ (και συνεπώς το υλικό του Η/Υ):



όπου :

**Μονάδα Εισόδου** : Είναι το σύνολο των μονάδων με τις οποίες επιτυγχάνεται η είσοδος των πληροφοριών στον Η/Υ π.χ. πληκτρολόγιο, ποντίκι, κ.τ.λ.

**Κύρια Μνήμη (RAM)** : Η μονάδα αυτή χρησιμοποιείται για την αποθήκευση του προγράμματος των δεδομένων, καθώς επίσης και των ενδιάμεσων και τελικών αποτελεσμάτων του προγράμματος πριν αυτά εμφανιστούν στην οθόνη ή τυπωθούν στον εκτυπωτή ή γραφτούν σε κάποια μονάδα δευτερεύουσας μνήμης όπως η μαγνητική δισκέτα ή ο μαγνητικός δίσκος.

**Κεντρική Μονάδα Επεξεργασίας (CPU)** : Η μονάδα στην οποία γίνονται οι κάθε είδους επεξεργασίες των πληροφοριών καθώς και ο συντονισμός και έλεγχος των λειτουργιών του Η/Υ. Αποτελείται από την αριθμητική και λογική μονάδα καθώς και την μονάδα ελέγχου.

**Μονάδα Εξόδου** : Είναι το σύνολο των μονάδων με τις οποίες επιτυγχάνεται η έξοδος των αποτελεσμάτων από τον Η/Υ στον εξωτερικό κόσμο π.χ. οθόνη, εκτυπωτής, δισκέτα κ.τ.λ.

Μερικές μονάδες του Η/Υ όπως οι δισκέτες και το modem μπορούν να θεωρηθούν ως μονάδες εισόδου-εξόδου γιατί και δίνουν πληροφορίες στον Η/Υ αλλά και παίρνουν από αυτόν.

## ΕΞΩΤΕΡΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ Η/Υ.

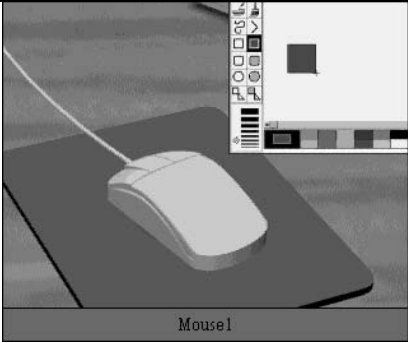
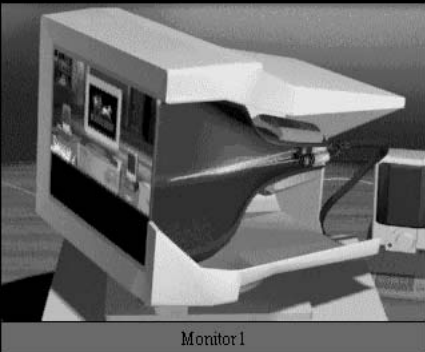
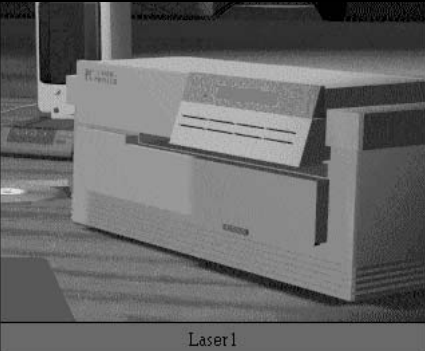



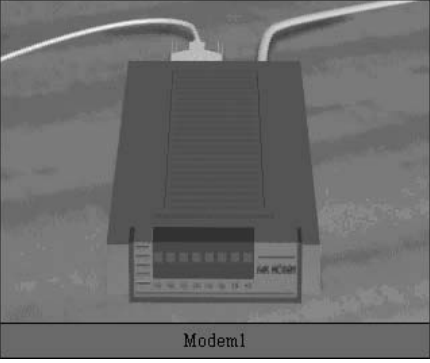
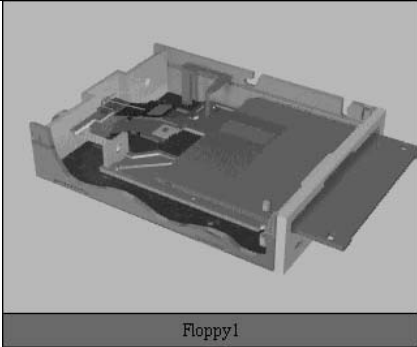
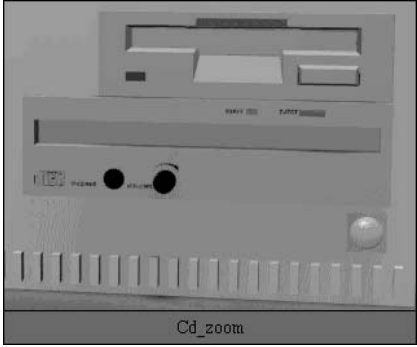
1. Πληκτρολόγιο (Keyboard)	6. Fax-Modem
2. Ποντίκι (Mouse)	7. Οδηγός Δισκέτας (Floppy Drive)
3. Οθόνη (Monitor)	8. Οδηγός CD-ROM (CD-ROM Drive)
4. Εκτυπωτής (Printer)	9. Κεντρική Μονάδα
5. Ηχεία (Speakers)	10. Μονάδα Ελέγχου (Control Panel)

**Σχήμα 1. Εξωτερική περιγραφή του εμπρόσθιου μέρους του Η/Υ**

### Μονάδες που αποτελούν το εμπροσθεν μέρος του Η/Υ

	<p><b>Πληκτρολόγιο (Keyboard)</b></p> <p>Βασικό μέσο επικοινωνίας με τον υπολογιστή, μια και επιτρέπει στους χρήστες να εισάγουν δεδομένα ή να δίνουν διαταγές, οι οποίες κατευθύνουν τον υπολογιστή στην εκτέλεση συγκεκριμένων λειτουργιών.</p>
--	---

 <p>Mouse1</p>	<p><b>Ποντίκι (Mouse)</b></p> <p>Χειροκίνητη συσκευή εισόδου που όταν μετακινείται πάνω σε μια επίπεδη επιφάνεια, προκαλεί ανάλογη μετατόπιση του δρομέα στην οθόνη, ενώ πάτημα ενός πλήκτρου του συνήθως οδηγεί σε εκτέλεση μιας διαταγής.</p>
 <p>Monitor1</p>	<p><b>Οθόνη (Monitor)</b></p> <p>Συσκευή εξόδου, όμοια με τηλεόραση, στην οποία παρουσιάζονται τα δεδομένα.</p>
 <p>Laser1</p>	<p><b>Εκτυπωτής (Printer)</b></p> <p>Συσκευή εξόδου που τυπώνει σε χαρτί τα αποτελέσματα του υπολογιστή, με τη μορφή αριθμών, γραμμάτων ή γραφικών εικόνων.</p>
 <p>Almanac Tour_io</p>	<p><b>Ηχεία (Speakers)</b></p> <p>Συσκευή εξόδου που μεταδίδει τον ήχο από συγκεκριμένες εφαρμογές πολυμέσων.</p>

 <p>Modem1</p>	<p><b>FAX-MODEM</b></p> <p>Συσκευή που επιτρέπει τη μεταβίβαση δεδομένων μεταξύ υπολογιστών, συνήθως μέσω των τηλεφωνικών γραμμών, αλλά επίσης, μερικές φορές μέσω καλωδίου οπτικών ινών ή μέσω ραδιοσυχνοτήτων.</p>
 <p>Floppy1</p>	<p><b>Οδηγός Δισκέτας (Floppy Drive)</b></p> <p>Μονάδα ανάγνωσης και εγγραφής δεδομένων σε μαγνητικές δισκέτες.</p>
 <p>Cd_zoom</p>	<p><b>Οδηγός CD-ROM (CD-ROM Drive)</b></p> <p>Μονάδα ανάγνωσης και εγγραφής δεδομένων σε συμπαγείς δίσκους CD-ROM.</p>

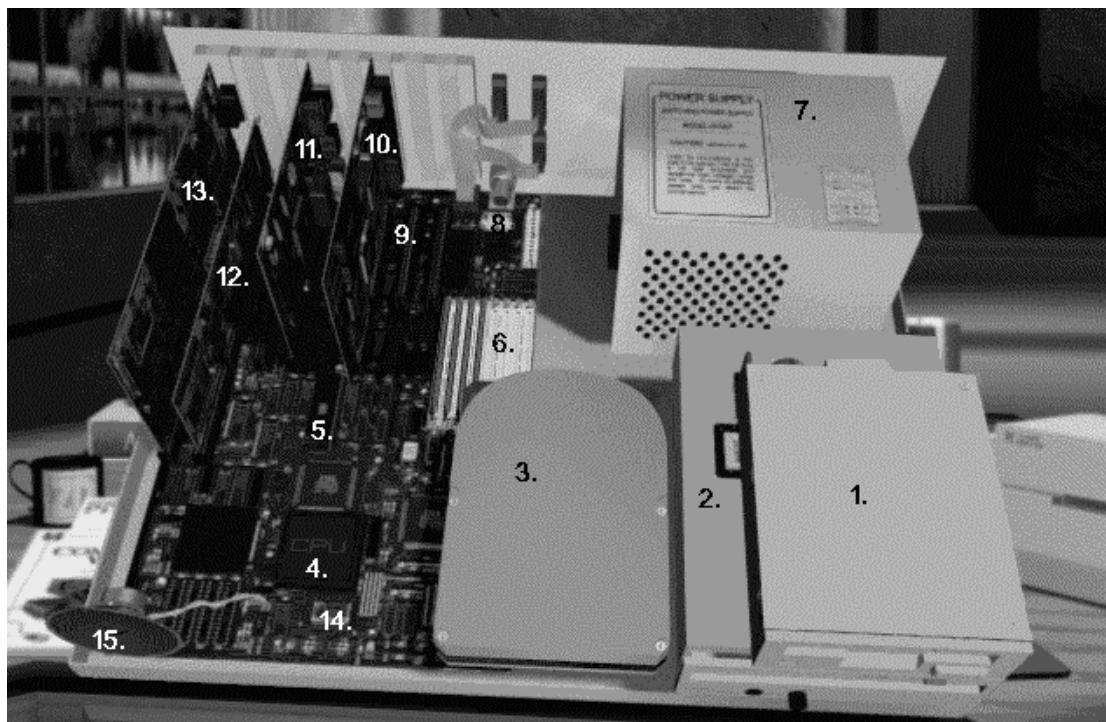




1. Παροχή Ρεύματος (Power Supply).	4. Κάρτα Ήχου (Sound Card).
2. Υποδοχές Εισόδου-Εξόδου (Input-Output Ports) (για ποντίκι, εκτυπωτή, οθόνη κ.τ.λ.)	5. Κάρτα Δικτύου (Network Card).
3. Υποδοχή Πληκτρολογίου (Keyboard Port).	6. Κάρτα Video (Video Card).

**Σχήμα 2. Εξωτερική περιγραφή του οπίσθιου μέρους του Η/Υ**

## ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΕΝΤΡΙΚΗΣ ΜΟΝΑΔΑΣ ΤΟΥ Η/Υ.

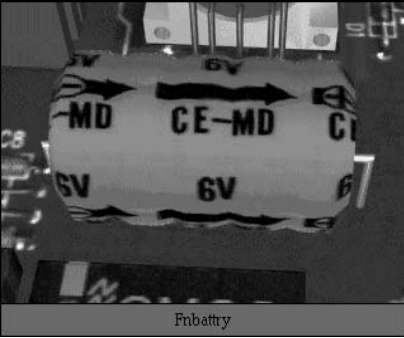
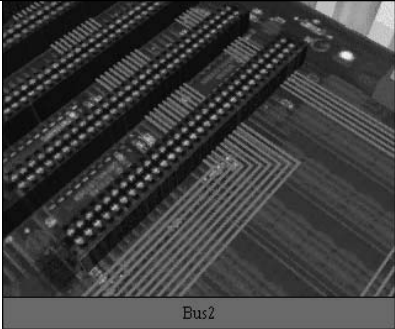
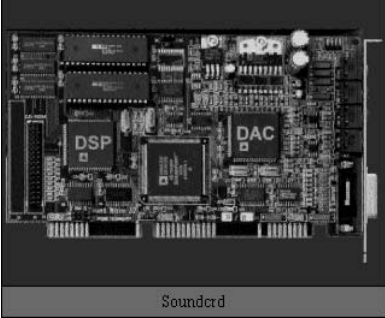
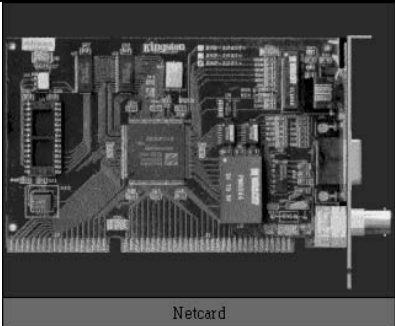


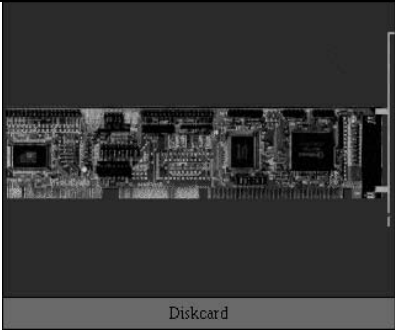
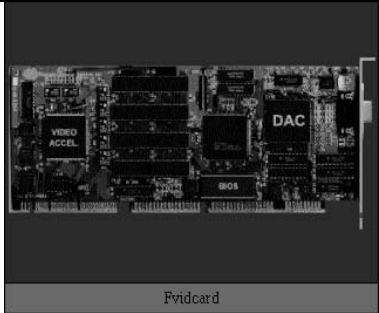
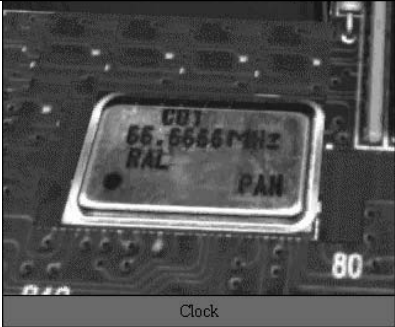
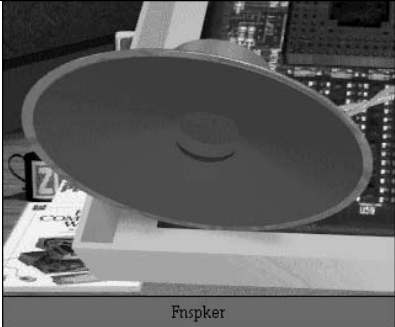
1. Οδηγός Δισκέτας (Floppy Drive).	9. Bus.
2. Οδηγός CD-ROM (CD-ROM Drive).	10. Κάρτα Ήχου (Sound Card).
3. Σκληρός Δίσκος (Hard Disk Drive).	11. Κάρτα Δικτύου (Network Card).
4. Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) ή CPU (Central Processing Unit).	12. Ελεγκτής Δίσκου (Disk Controller).
5. Μνήμη Μόνο Ανάγνωσης ή ROM (Read Only Memory).	13. Κάρτα Video (Video Card).
6. Μνήμη Τυχαίας Προσπέλασης ή RAM (Random Access Memory).	14. Ρολόι (Clock Chip).
7. Τροφοδοτικό (Power Supply).	15. Εσωτερικό Ηχείο (Internal Speaker).
8. Μπαταρία (Battery).	

### Μονάδες που αποτελούν την κεντρική μονάδα του Η/Υ

	<p><b>Σκληρός Δίσκος (Hard Drive)</b></p> <p>Είναι η κύρια περιοχή αποθήκευσης του Η/Υ και μπορεί να φιλοξενήσει μεγάλες ποσότητες προγραμμάτων και πληροφοριών σε μαγνητική μορφή και να διατηρεί αυτά τα δεδομένα ακόμα και όταν διακόπτεται η τροφοδοσία του Η/Υ.</p>
--	--

 <p>Cpu1</p>	<p><b>Κεντρική Μονάδα Επεξεργασίας (CPU)</b></p> <p>Μέρος του Η/Υ που ερμηνεύει και εκτελεί εντολές, και επιβλέπει την μεταφορά των δεδομένων από και προς εξωτερικές πηγές. Αποτελείται από έναν επεξεργαστή (αριθμητική και λογική μονάδα), μια μονάδα ελέγχου και μια μικρής χωρητικότητας μνήμη.</p>
 <p>Rombios</p>	<p><b>Μνήμη μόνο ανάγνωσης (ROM)</b></p> <p>Μόνιμη εσωτερική μνήμη που περιέχει δεδομένα ή προγράμματα (BIOS) τα οποία είναι δυνατό να διαβαστούν από τον χρήστη αλλά όχι να τροποποιηθούν.</p>
 <p>Stam1</p>	<p><b>Μνήμη Τυχαίας Προσπέλασης (RAM)</b></p> <p>Μνήμη στην οποία αποθηκεύονται προσωρινά τα δεδομένα και τα προγράμματα. Το περιεχόμενο της μπορεί να τροποποιηθεί από τον χρήστη καθώς και χάνεται με την διακοπή παροχής ρεύματος προς τον Η/Υ.</p>
 <p>Egwp1 sup</p>	<p><b>Τροφοδοτικό</b></p> <p>Μετατρέπει το εναλλασσόμενο ρεύμα 220V του ηλεκτρικού δικτύου στο συνεχές ρεύμα χαμηλής τάσης που απαιτούν τα ευαίσθητα ηλεκτρονικά εξαρτήματα και περιφερειακά.</p>

 <p>Fnbattery</p>	<p><b>Μπαταρία</b></p> <p>Φροντίζει για την διατήρηση πληροφοριών που βρίσκονται στην μνήμη CMOS (όπως ημερομηνία, ώρα, χωρητικότητα του σκληρού δίσκου κ.τ.λ.) κατά την περίοδο που ο Η/Υ είναι κλειστός.</p>
 <p>Bus2</p>	<p><b>Bus ή Δίαυλος</b></p> <p>Ομάδα καλωδίων που μεταφέρουν σήματα σε διάφορα σημεία του Η/Υ.</p>
 <p>Soundcard</p>	<p><b>Κάρτα Ήχου</b></p> <p>Απαραίτητες για την ηχογράφηση και αναπαραγωγή ποιοτικών ήχων. Μετατρέπουν τον ήχο που είναι αναλογικό σήμα σε ψηφιακό που είναι αντιληπτό από τον Η/Υ και αντίστροφα.</p>
 <p>Netcard</p>	<p><b>Κάρτα Δικτύου</b></p> <p>Επιτρέπει την διασύνδεση του Η/Υ με άλλους Η/Υ (δίκτυο Η/Υ).</p>

 <p>Diskcard</p>	<p><b>Ελεγκτής Δίσκου (Disk Controller)</b></p> <p>Αναλαμβάνει την διαχείριση των λειτουργιών των δίσκων.</p>
 <p>Fvidcard</p>	<p><b>Κάρτα Video</b></p> <p>Μεταρέπει τα δεδομένα από τα προγράμματα σε σήματα που αναγνωρίζει η οθόνη.</p>
 <p>Clock</p>	<p><b>Ρολοί</b></p> <p>Παρέχει μια συχνότητα (σε κύκλους ανά δευτερόλεπτο) που συντονίζει τον επεξεργαστή, τον δίαυλο και τα υπόλοιπα ηλεκτρονικά εξαρτήματα.</p>
 <p>Fnspeaker</p>	<p><b>Εσωτερικό Ηχείο</b></p> <p>Παράγει τα γνωστά bíp σε περίπτωση λαθών από τον χρήστη. Δεν μπορεί να χρησιμοποιηθεί σε εφαρμογές πολυμέσων.</p>

## ΛΟΓΙΣΜΙΚΟ Η SOFTWARE

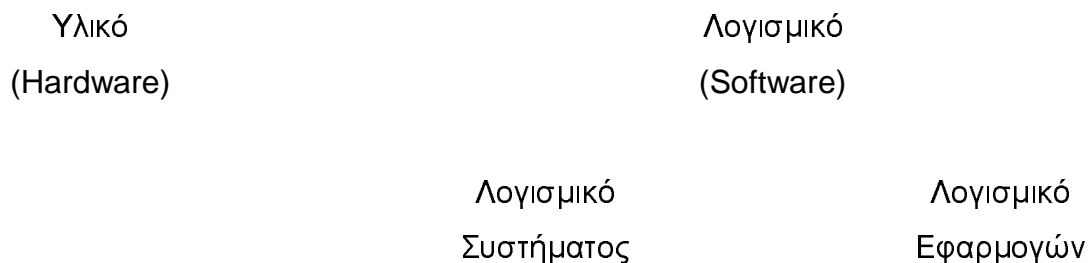
Ο Η/Υ αποτελεί μια ηλεκτρονική συσκευή η οποία λειτουργεί μόνο κάτω από ένα σύνολο προγραμμάτων τα οποία όλα μαζί αποτελούν το λογισμικό ή

software του Η/Υ. Το λογισμικό του Η/Υ χωρίζεται στις δύο παρακάτω κατηγορίες :

**Λογισμικό Συστήματος** : Αποτελείται από όλα εκείνα τα προγράμματα που εξασφαλίζουν τη σύνδεση όλων των μονάδων μεταξύ τους και τον έλεγχο των λειτουργιών τους. Το λογισμικό συστήματος περιλαμβάνει κυρίως το λειτουργικό σύστημα (με το οποίο θα ασχοληθούμε στο επόμενο κεφάλαιο) και τα μεταφραστικά προγράμματα.

**Λογισμικό Εφαρμογών** : Ανήκουν όλα τα προγράμματα ή σύνολα προγραμμάτων που γράφονται για να διεκπεραιώνουμε με την βοήθεια του Η/Υ διάφορες εργασίες. Με αυτά τα προγράμματα όπως επεξεργασία κειμένου, λογιστικά φύλλα κ.λ.π. θα ασχοληθούν οι φοιτητές στο εργαστήριο του «Προγραμματισμός Η/Υ II».

Η/Υ



Σχήμα 3. Δομή του Η/Υ

## ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ MS-DOS

Το *λειτουργικό σύστημα* (Λ.Σ.) είναι ο συνδυετικός κρίκος μεταξύ του υλικού από τη μια μεριά και των προγραμμάτων εφαρμογής από την άλλη. Πρωταρχικός ρόλος του Λ.Σ. είναι η εύκολη και αποδοτική χρησιμοποίηση του υλικού από κάποιο χρήστη. Σε γενικές γραμμές το Λ.Σ. καθορίζει στον υπολογιστή πώς να καλεί κάποιο πρόγραμμα από μια μονάδα αποθήκευσης, πώς να αποθηκεύει δεδομένα σ'αυτές, πώς να χειρίζεται την οθόνη και τον εκτυπωτή, και σε γενικές γραμμές αναλαμβάνει να συντονίζει τα διάφορα μέρη του υπολογιστή που απαιτούνται για την εκτέλεση μιας ολοκληρωμένης εργασίας. Το Λ.Σ. και η αρχιτεκτονική των Η/Υ έχουν επηρεάσει το ένα το άλλο σε μεγάλο βαθμό. Η ραγδαία εξέλιξη λοιπόν στον χώρο της αρχιτεκτονικής των Η/Υ οδήγησε και στην αντίστοιχη εξέλιξη του χώρου των λειτουργικών συστημάτων (αν και πολλές φορές συνέβη και το αντίθετο). Σήμερα υπάρχουν δύο κατηγορίες λειτουργικών συστημάτων :

Αυτά που μπορούν να εξυπηρετούν έναν χρήστη κάθε φορά και χρησιμοποιούνται στους μικροϋπολογιστές. Τέτοια λειτουργικά συστήματα είναι το MS-DOS, τα WINDOWS 3.1, τα WINDOWS 95 κ.τ.λ.

Αυτά που έχουν την δυνατότητα να εξυπηρετούν συγχρόνως περισσότερους από ένα χρήστες (σε δίκτυα υπολογιστών). Τέτοια λειτουργικά συστήματα είναι το UNIX, τα WINDOWS 95, τα WINDOWS NT κ.τ.λ.

Το MS-DOS είναι το λειτουργικό σύστημα που αναπτύχθηκε από την Microsoft για τον IBM PC και αποτελεί μια μεταφορά του λειτουργικού συστήματος QDOS που αναπτύχθηκε από τον Tim Paterson. Στόχος μας όπως θα δούμε και παρακάτω δεν είναι να περιγράψουμε πλήρως το λειτουργικό σύστημα MS-DOS αλλά να αναπτύξουμε βασικές δεξιότητες των φοιτητών στην διαχείριση αρχείων και καταλόγων μέσω εντολών του MS-DOS. Προτού μιλήσουμε εκτενέστερα για διαχείριση αρχείων, θεωρούμε σκόπιμο να ορίσουμε το τι είναι αρχείο.

## ΑΡΧΕΙΑ – ΟΝΟΜΑΣΙΕΣ ΑΡΧΕΙΩΝ

Με τον όρο *αρχείο* εννοούμε ένα σύνολο από ομοειδείς πληροφορίες που έχουν αποθηκευτεί σε δίσκο ή δισκέτα με συγκεκριμένο όνομα. Υπάρχουν πολλά είδη αρχείων όπως αρχεία εικόνας, αρχεία κειμένου, αρχεία δεδομένων, αρχεία προγραμμάτων κ.τ.λ.

Ένα αρχείο έχει συγκεκριμένο όνομα με το οποίο μπορούμε να το αναζητήσουμε στον δίσκο ανάμεσα σε άλλα αρχεία. Το όνομα αυτό πρέπει πάντα να έχει την εξής μορφή :

*Κυρίως Όνομα[.Επέκταση]*

Το κυρίως όνομα πρέπει να υπάρχει υποχρεωτικά και να έχει το πολύ 8 χαρακτήρες, ενώ η επέκταση είναι προαιρετική, διαχωρίζεται από το κυρίως όνομα με τελεία και έχει το πολύ 3 χαρακτήρες. Ως χαρακτήρες ορίζω τα γράμματα του αγγλικού αλφαβήτου, τους αριθμούς καθώς και τα ειδικά σύμβολα ! @ # \$ % ^ & ( ) \_ - ~ { } ' | < > \ τα οποία όμως δεν θα χρησιμοποιούμε από εδώ και στο εξής για μνημονικούς λόγους. Δεν εννοώ ως χαρακτήρες στην συγκεκριμένη περίπτωση τα γράμματα του Ελληνικού αλφαβήτου, το κενό και τα σύμβολα (αν και μερικά σύμβολα είναι επιτρεπτά). Οι παραπάνω κανόνες ισχύουν μόνο στην περίπτωση του λειτουργικού συστήματος MS-DOS και όχι άλλων λειτουργικών συστημάτων όπως τα Windows 95.

**Άσκηση** Ποια από τα παρακάτω ονόματα αρχείων είναι σωστά<sup>1</sup> ;

Όνομα Αρχείου	Σωστό-Λάθος	Εξήγηση
ΝΙΚΟΣ	ΛΑΘΟΣ	Έχει Ελληνικά γράμματα
ANASTASIA	ΛΑΘΟΣ	Το κυρίως όνομα έχει παραπάνω από 8 χαρακτήρες.
ΝΙΚΟΣ	ΣΩΣΤΟ	Το κυρίως όνομα που είναι υποχρεωτικό έχει ως 8 επιτρεπτούς χαρακτήρες.
ANNA.DOC	ΣΩΣΤΟ	Το κυρίως όνομα και η επέκταση έχουν ως 8 και 3 αντίστοιχα επιτρεπτούς χαρακτήρες.
TEXT*N	ΛΑΘΟΣ	Περιέχει σύμβολα
.DOC	ΛΑΘΟΣ	Λείπει το κυρίως όνομα που

<sup>1</sup> Αντί του κενού από εδώ και στο εξής θα χρησιμοποιούμε τον χαρακτήρα \_ για εμφανείς λόγους.



		είναι υποχρεωτικό.
N_1.AN	ΛΑΘΟΣ	Το κυρίως όνομα έχει κενό.
N.1	ΣΩΣΤΟ	Το κυρίως όνομα και η επέκταση έχουν ως 8 και 3 αντίστοιχα επιτρεπτούς χαρακτήρες.

## ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΧΕΙΩΝ ΤΟΥ MS-DOS

Ένα πλήθος εντολών συνοδεύει το MS-DOS οι οποίες έχουν ως σκοπό να βοηθήσουν τον χρήστη να διαχειριστεί τα αρχεία των δίσκων και τις διάφορες μονάδες του Η/Υ. Οι εντολές αυτές πρέπει να δίνονται με το ακριβές όνομα τους και την ακριβή σύνταξη τους διαφορετικά ο Η/Υ απαντά με μηνύματα λάθους. Οι εντολές μπορούν να γραφούν είτε με μικρά είτε με κεφαλαία γράμματα χωρίς αυτό να επηρεάζει την εκτέλεση τους.

Αρχικά όταν ανοίξουμε τον Η/Υ από το κουμπί POWER που θα βρούμε πάνω στην κεντρική μονάδα, πιθανόν<sup>2</sup> να έρθουμε αντιμέτωποι με το παρακάτω σήμα στην οθόνη :

C:\>

Το παραπάνω σήμα, που είναι γνωστό και ως σήμα ετοιμότητας (prompt) αποτελείται από δύο μέρη.

Το πρώτο μέρος που αποτελείται από ένα αγγλικό γράμμα και άνω και κάτω τελεία υποδηλώνει τον οδηγό δίσκου στον οποίο βρισκόμαστε εκείνη την στιγμή (δηλ. στην συγκεκριμένη περίπτωση τον σκληρό δίσκο του Η/Υ που έχει το όνομα C).

ΟΔΗΓΟΣ	ΟΝΟΜΑΣΙΑ
Όνομα Δισκέτας	A:, B: A: για τον πρώτο οδηγό δισκέτας και B: για τον δεύτερο οδηγό δισκέτας (αν υπάρχει)
Όνομα Δίσκου	C: (εάν υπάρχει και δεύτερος οδηγός δίσκου ή ο δίσκος μου είναι διαμερισμένος στα δύο τότε ο δεύτερος δίσκος έχει το όνομα D:)

<sup>2</sup> Το παραπάνω σήμα δεν εμφανίζεται αυτόματα στους υπολογιστές που έχουν ως βασικό λειτουργικό σύστημα τα Windows 95, όπου θα πρέπει να πάμε πρώτα στην επιλογή Προγράμματα και στην συνέχεια να επιλέξουμε το MS-DOS.

Όνομα CD-ROM	D:  (εκτός αν υπάρχουν δύο σκληροί δίσκοι ή διαμερισμένος σκληρός δίσκος, οπότε και έχει το όνομα E:)
--------------	---

Το δεύτερο μέρος (\) ονομάζεται *ριζικός κατάλογος* ή *κατάλογος ρίζα*, και δεν είναι τίποτα άλλο από ένα αρχείο το οποίο δημιουργείται κατά την μορφοποίηση του δίσκου (FORMAT) και περιέχει πληροφορίες για τα αρχεία του δίσκου όπως οι παρακάτω :

Όνομα Αρχείου	Επέκταση Ονόματος	Ώρα Τελευταίας Μεταβολής	Ημέρα Τελευταίας Μεταβολής	Μέγεθος Αρχείου	.....
---------------	-------------------	--------------------------	----------------------------	-----------------	-------

Το σήμα ετοιμότητας υποδηλώνει ότι πλέον μπορείτε να πληκτρολογήσετε μετά από το σήμα αυτό μια εντολή του MS-DOS. Κάθε εντολή που γράφουμε μετά από το σήμα ετοιμότητας, για να ενεργοποιηθεί πρέπει να πατήσουμε το ENTER.

Για να μεταφερθούμε από τον ένα οδηγό δίσκου στον άλλο, πρέπει να γράψουμε δίπλα στο σήμα ετοιμότητας το όνομα του οδηγού δίσκου που θέλουμε να πάμε, στην συνέχεια (:), και να πατήσουμε ENTER π.χ. αν το σήμα ετοιμότητας είναι C:\> και θέλουμε να πάμε στον οδηγό δισκέτας A, γράφουμε A: και πατάμε ENTER.

Παρακάτω θα αναφέρουμε μερικές από τις πιο απλές εντολές του MS-DOS. Στις παρακάτω εντολές θα αναφέρουμε πάντα τον σκοπό, την σύνταξη, ένα παράδειγμα και κάποια σχόλια. Όταν κατά την σύνταξη χρησιμοποιούμε αγκύλες, αυτό σημαίνει ότι οι συγκεκριμένες παράμετροι της εντολής είναι προαιρετικοί.

## Η ΕΝΤΟΛΗ CLS

**Σκοπός :** Η εντολή CLS καθαρίζει την οθόνη αφήνοντας μόνο το σήμα ετοιμότητας με τον τρέχων κατάλογο.

**Σύνταξη :** CLS

## Η ΕΝΤΟΛΗ DATE

**Σκοπός :** Η εντολή DATE παρουσιάζει την ημερομηνία και ζητά μια νέα.

**Σύνταξη :** DATE [dd-mm-yy]

### Παράδειγμα.

```
C:\>DATE (πατάτε ENTER)
```

```
Current date is Tue 16/09/97
```

```
Enter new date (dd-mm-yy) : 17/09/97 (πληκτρολογείται καινούργια μέρα και πατάτε ENTER)
```

### Παράδειγμα.

```
C:\>DATE (πατάτε ENTER)
```

```
Current date is Tue 16/09/97
```

```
Enter new date (dd-mm-yy) : 17/40/97
```

```
Invalid date
```

```
Enter new date (dd-mm-yy) : 17/09/97
```

Βλέπουμε δηλαδή ότι αν δώσετε λάθος ημερομηνία ο Η/Υ σας απαντάει με Invalid date και περιμένει να του δώσετε καινούργια ημερομηνία. Θα πρέπει λοιπόν να δίνουμε πάντα σωστά τις παραμέτρους της κάθε εντολής.

### Παράδειγμα.

```
C:\>DATTE
```

```
Bad command or file name
```

**Σχόλια :** Το MS-DOS σας προτείνει να γράψετε την ημερομηνία ακολουθώντας τον εξής κανόνα : πρώτα γράφω την ημέρα (dd), μετά τον μήνα (mm) και τέλος τον χρόνο (yy). Εάν το MS-DOS σας προτείνει τον κανόνα (mm-dd-yy) πρώτα θα γράψετε τον μήνα, μετά την ημέρα και τέλος τον χρόνο. Στην περίπτωση που η μέρα είναι η σωστή, δεν χρειάζεται να γράψω μια καινούργια μέρα αλλά να πατήσω ENTER. Βλέπουμε επίσης ότι εάν γράψουμε λάθος την εντολή ο Η/Υ μας απαντάει με το μήνυμα (λάθος

εντολή ή όνομα αρχείου). Παρατηρούμε λοιπόν από τα παραπάνω ότι υπάρχει ένας συνεχής διάλογος μεταξύ χρήστη και λειτουργικού συστήματος.

## Η ΕΝΤΟΛΗ TIME

**Σκοπός :** Η εντολή TIME παρουσιάζει την ώρα και ζητά μια νέα.

**Σύνταξη :** TIME [hours:minutes[:seconds [.hundredths]]]

### Παράδειγμα.

C:\>TIME (πατάτε ENTER)

Current time is 11:37:54,70a

Enter new time : 12:37:54 (πληκτρολογείται καινούργια ώρα και πατάτε ENTER)

**Σχόλια :** Στην περίπτωση που η ώρα είναι η σωστή, δεν χρειάζεται να γράψω μια καινούργια ώρα αλλά να πατήσω ENTER.

## Η ΕΝΤΟΛΗ COPY CON

**Σκοπός :** Η δημιουργία ενός αρχείου του οποίου το περιεχόμενο θα το γράψουμε εμείς.

**Σύνταξη :** COPY CON [οδηγός δίσκου:][μονοπάτι<sup>3</sup>]<ονομασία αρχείου> (ENTER)

.....	(γράφω το περιεχόμενο του
.....	αρχείου και πατώ ENTER
.....	στο τέλος της κάθε γραμμής)
Ctrl+Z (ή πλήκτρο F6)	(δηλώνω το τέλος του αρχείου και πατώ ENTER))

### Παράδειγμα.

C:\>COPY CON LETTER.DOC

Αξιότιμε κύριε Σωτηριάδη,

Θα ήθελα .....

---

<sup>3</sup> Θα εξηγήσουμε στην συνέχεια την έννοια «μονοπάτι».

.....

Ctrl+Z και στην συνέχεια ENTER

1 file(s) copied

**Σχόλια :** Ουσιαστικά η παραπάνω εντολή αντιγράφει το περιεχόμενο που βρίσκεται μεταξύ της εντολής και του ^Z στο αρχείο με το όνομα LETTER.DOC. Από την στιγμή που δημιουργήθηκε το αρχείο, ενημερώθηκε και ο κατάλογος ρίζα (\) με τα στοιχεία του αρχείου (ώρα δημιουργίας, μέρα δημιουργίας κ.τ.λ.).

## Η ΕΝΤΟΛΗ TYPE

**Σκοπός :** Η εντολή TYPE εμφανίζει το περιεχόμενο ενός αρχείου ASCII στην οθόνη.

**Σύνταξη :** TYPE [οδηγός δίσκου:][[μονοπάτι] <όνομα αρχείου>

### Παράδειγμα.

C:\>TYPE LETTER.DOC (πατάτε ENTER)

Αξιότιμε κύριε Σωτηριάδη,

Θα ήθελα .....

.....

**Σχόλια :** Εάν το περιεχόμενο ενός αρχείου είναι μεγάλο , μπορείτε να χρησιμοποιήσετε την παράμετρο more, έτσι ώστε να εμφανίζεται το περιεχόμενο του κειμένου σε μια οθόνη κάθε φορά.

### Παράδειγμα.

C:\>TYPE LETTER.DOC | MORE (πατάτε ENTER)

## Η ΕΝΤΟΛΗ DIR

**Σκοπός :** Η εντολή DIR εμφανίζει τα αρχεία ενός καταλόγου.

**Σύνταξη :** DIR [οδηγός δίσκου:][[μονοπάτι][[/p][[/w]

**Παράδειγμα.**

```
C:\>DIR (ή DIR C:\, ή DIR \)
```

```
Volume in drive C has no label
```

```
Volume Serial Number is 1131-18DC
```

```
Directory of C:\
```

COMMAND	DOS	54.619	30/09/93	6:20a	COMMAND.DOS
DOS	<DIR>		01/06/96	5:26p	DOS
AUTOEXEC	DOS	168	26/06/96	9:30p	AUTOEXEC.DOS
CONFIG	DOS	545	26/06/96	9:27p	CONFIG.DOS
.....	.....	.....	.....	.....	.....

```
48 file(s) 1.745.008 bytes
```

```
24 dir(s) 38.797.312 bytes free
```

**Σημείωση :** Η εντολή DIR μας εμφάνισε το περιεχόμενο του τρέχοντος καταλόγου. Το όνομα του τρέχοντος καταλόγου βρίσκεται αριστερά του σήματος ετοιμότητας (>). Στην συγκεκριμένη περίπτωση πρόκειται για τον κατάλογο ρίζα (\) του σκληρού δίσκου (C). Η εντολή DIR \ εμφανίζει πάντα τα περιεχόμενα του καταλόγου ρίζα του δίσκου ή της δισκέτας στην οποία βρισκόμαστε εκείνη την στιγμή.

**Σχόλια :** Η εντολή DIR χωρίς οδηγό δίσκου και μονοπάτι εμφανίζει το περιεχόμενο του τρέχοντος καταλόγου δηλ. του καταλόγου στον οποίο βρισκόμαστε. Μέχρι τώρα γνωρίζουμε μόνον τον κατάλογο ρίζα (\) ο οποίος περιέχει πληροφορίες σχετικές με τα αρχεία αλλά και τους καταλόγους που δημιουργούνται στον δίσκο. Θα δούμε στην συνέχεια ότι μπορούμε να έχουμε και άλλους καταλόγους που θα μας βοηθήσουν στην ανακατανομή των αρχείων μας.

Η εντολή DIR δέχεται τις παρακάτω παραμέτρους :

Παράμετρος	Σκοπός
/P	Εμφανίζει τα περιεχόμενα του καταλόγου ανά σελίδα. Όταν παρουσιάζεται μια σελίδα η οθόνη ακινητοποιείται και πρέπει να πατήσουμε ένα οποιοδήποτε πλήκτρο για να συνεχιστεί η παρουσίαση της επόμενης οθόνης.
/W	Εμφανίζει τα περιεχόμενα του καταλόγου χωρίς επιπλέον πληροφορίες όπως ημερομηνία και ώρα δημιουργίας αρχείου κ.τ.λ. Συνεπώς εμφανίζει περισσότερα ονόματα αρχείων ανά σελίδα.

Όταν ο κατάλογος στον οποίο εργάζεστε περιέχει περισσότερα αρχεία από αυτά που μπορείτε να δείτε στην οθόνη, τότε χρησιμοποιήστε μια από τις παραπάνω παραμέτρους.

Παραδείγματα της εντολής DIR με άλλους καταλόγους εκτός από τον κατάλογο ρίζα θα δούμε στην συνέχεια.

Συχνά οι φοιτητές μπερδεύουν τις εντολές TYPE και DIR. Ενώ η πρώτη μας δείχνει το περιεχόμενο ενός συγκεκριμένου αρχείου, η δεύτερη μας παρουσιάζει τα ονόματα των αρχείων (και όχι τα περιεχόμενα), του καταλόγου που ζητάμε.

### Παράδειγμα.

```
C:\>DIR /W (Πατώ ENTER)
Volume in drive C has no label
Volume Serial Number is 1131-18DC
Directory of C:\
COMMAND.DOS [DOS]      AUTOEXEC.DOS  CONFIG.DOS   [WINGK]
[DOSAPPL]  [WINAPPL]  FILE0004.CHK [MSPUB]     [MSWORKS]
[QL2FAX]   CONFIG.SYS [WINFAX]    [TELIX]     [TFW]      _
NIKOS.TXT .....
                48 file(s)   1.745.008 bytes
                24 dir(s)   38.797.312 bytes free
```

### Παράδειγμα.

```
C:\>DIR A:\ /W (Πατώ ENTER)
Εμφανίζει τα περιεχόμενα του καταλόγου ρίζα της δισκέτας A, χωρίς επιπλέον πληροφορίες για το κάθε αρχείο και κατάλογο.
```

## Η ΕΝΤΟΛΗ RENAME

**Σκοπός :** Η εντολή RENAME αλλάζει το όνομα ενός αρχείου.

**Σύνταξη :**

RENAME [οδηγός δίσκου:] [μονοπάτι] <όνομα αρχείου1> <όνομα αρχείου 2>  
όπου :

<όνομα αρχείου 1> είναι το παλιό όνομα του αρχείου

<όνομα αρχείου 2> είναι το καινούργιο όνομα του αρχείου

**Παράδειγμα.**

C:\>RENAME LETTER.DOC TEXT.DOC (πατάτε ENTER)

Άλλαξε το όνομα του αρχείου από LETTER.DOC που βρίσκεται στον κατάλογο ρίζα του σκληρού δίσκου C, σε TEXT.DOC. Όταν λοιπόν θα εκτελέσω την εντολή DIR στην συνέχεια δεν θα παρουσιαστεί το όνομα LETTER.DOC αλλά το όνομα TEXT.DOC.

**Σχόλια :** Μπορώ αντί της λέξης RENAME να χρησιμοποιήσω την λέξη REN.

*Όταν μπροστά από το όνομα ενός αρχείου λείπει ο κατάλογος στον οποίο βρίσκεται, τότε το MS-DOS ψάχνει να το βρει στον τρέχων κατάλογο (δες παραπάνω πρδ.).*

*Σε ασκήσεις που δεν σας δίνεται ο τρέχων κατάλογος θα δίνεται την πλήρη ονομασία του αρχείου ξεκινώντας από τον δίσκο (αν δίνεται στην εκφώνηση), το μονοπάτι και στην συνέχεια το όνομα του αρχείου.*

## Η ΕΝΤΟΛΗ COPY

**Σκοπός :** Η εντολή COPY αντιγράφει ένα ή περισσότερα αρχεία από μια περιοχή του δίσκου σε μια άλλη περιοχή του ίδιου ή διαφορετικού δίσκου



**Σύνταξη :**

COPY [οδηγός δίσκου:][μονοπάτι]<όνομα αρχ.1> [οδηγός δίσκου:][μονοπάτι][όνομα αρχ.2]  
όπου :

<όνομα αρχείου 1> είναι το όνομα του αρχείου που θέλω να αντιγράψω.

<όνομα αρχείου 2> είναι το όνομα του αντίγραφου αρχείου.

**Παράδειγμα.**

C:\>COPY TEXT.DOC TEXT1.DOC (πατάτε ENTER)

Δημιουργεί ένα αντίγραφο του αρχείου TEXT.DOC που βρίσκεται στον κατάλογο ρίζα του σκληρού δίσκου C, στον ίδιο κατάλογο και στον ίδιο δίσκο αλλά με όνομα TEXT1.DOC. Εάν μετά την εντολή αυτή εκτελέσω την εντολή DIR θα διαπιστώσω ότι στον κατάλογο ρίζα του σκληρού δίσκου C, υπάρχουν τα αρχεία TEXT.DOC και TEXT1.DOC με ακριβώς τα ίδια στοιχεία.

**Σχόλια :** Πολλές φορές οι φοιτητές μπερδεύουν την εντολή RENAME και την εντολή COPY. Ενώ η πρώτη απλώς αλλάζει το όνομα του αρχείου, η δεύτερη δημιουργεί ένα επιπλέον αρχείο, αντίγραφο του πρώτου. Επίσης η σύνταξη τους είναι διαφορετική.

**Παράδειγμα.**

C:\>COPY TEXT.DOC A:\ (πατάτε ENTER)

Δημιουργεί ένα αντίγραφο του αρχείου TEXT.DOC που βρίσκεται στον κατάλογο ρίζα του σκληρού δίσκου C, στον κατάλογο ρίζα της δισκέτας A:. Επειδή δεν δίνουμε όνομα στο αντίγραφο αρχείο αυτό θα έχει ακριβώς το ίδιο όνομα με το αρχικό αρχείο δηλ. TEXT.DOC.

**Η ΕΝΤΟΛΗ DEL**

**Σκοπός :** Η εντολή DEL διαγράφει τα προσδιοριζόμενα αρχεία.

**Σύνταξη :** DEL [οδηγός δίσκου:][μονοπάτι]<όνομα αρχείου>

**Παράδειγμα.**

C:\>DEL TEXT1.DOC (πατάτε ENTER)

Διαγράφει το αρχείο TEXT1.DOC που βρίσκεται στον κατάλογο ρίζα του σκληρού δίσκου C.

### **Παράδειγμα.**

```
C:\>DEL A:\TEXT.DOC (πατάτε ENTER)
```

Διαγράφει το αρχείο TEXT.DOC που βρίσκεται στον κατάλογο ρίζα της δισκέτας A.

## **ΟΙ ΧΑΡΑΚΤΗΡΕΣ ΜΠΑΛΑΝΤΕΡ (WILDCARDS) \* ΚΑΙ ?**

Για την ομαδική επεξεργασία αρχείων χρησιμοποιούμε τους χαρακτήρες \* και ?. Με τον όρο ομαδική επεξεργασία εννοώ όταν θέλω για παράδειγμα να σβήσω μια ομάδα αρχείων (και όχι ένα αρχείο) που έχουν μια συγκεκριμένη ιδιότητα, να αντιγράψω μια ομάδα αρχείων με μια συγκεκριμένη ιδιότητα κ.τ.λ.

Το (?) όταν χρησιμοποιείται στην ονομασία ενός αρχείου σημαίνει ότι την θέση του μπορεί να την πάρει οποιοσδήποτε ένας χαρακτήρας. Το (\*) όταν χρησιμοποιείται στην ονομασία ενός αρχείου σημαίνει ότι την θέση του μπορεί να πάρει ένας ή και περισσότεροι χαρακτήρες.

**Παράδειγμα.** Έστω ότι ο κατάλογος ρίζα του σκληρού δίσκου C, περιέχει τα εξής αρχεία :

```
Text.doc, letter.doc, text2.doc, text123.doc, a.txt, help.txt, f2.exe
```

```
C:\>del *.doc
```

Η εντολή αυτή θα διαγράψει τα αρχεία που βρίσκονται στον κατάλογο ρίζα του σκληρού δίσκου C, που το κυρίως όνομα τους αποτελείται από οποιαδήποτε σειρά χαρακτήρων και η επέκτασή τους είναι doc, δηλ. τα πρώτα 4 αρχεία.

```
C:\>copy text?.doc a:\
```

Η εντολή αυτή αντιγράφει όλα τα αρχεία που βρίσκονται στον κατάλογο ρίζα του σκληρού δίσκου C και το κυρίως όνομα τους αποτελείται από 5 χαρακτήρες των οποίων οι 4 πρώτοι είναι text και έχουν επέκταση doc, στον κατάλογο ρίζα της δισκέτας A με ακριβώς το ίδιο όνομα. Δηλαδή στην συγκεκριμένη περίπτωση αντιγράφει το text2.doc.

```
C:\>rename *.doc *.txt
```

Η εντολή αυτή αλλάζει το όνομα των αρχείων που βρίσκονται στον κατάλογο ρίζα του σκληρού δίσκου C και το κυρίως όνομα τους είναι οποιοδήποτε, ενώ η επέκταση τους είναι doc. Τα ονόματα των καινούργιων αρχείων θα είναι η ίδια, ενώ η επέκταση θα είναι txt αντί doc. Δηλαδή τα ονόματα των αρχείων text.doc, nikos.doc, anna.doc θα μετονομαστούν σε text.txt, nikos.txt και anna.txt αντίστοιχα.

```
C:\>dir *.exe
```

Η εντολή αυτή εμφανίζει τα ονόματα όλων των αρχείων που βρίσκονται στον κατάλογο ρίζα του σκληρού δίσκου C και το κυρίως όνομα τους είναι οποιοδήποτε, ενώ η επέκταση τους είναι exe. Δηλαδή στην περίπτωση μας μόνο το αρχείο f2.exe. Η χρήση των μπαλαντέρ είναι πολύ συχνή στην εντολή DIR λόγω του ότι συνήθως υπάρχουν πολλά αρχεία στον σκληρό δίσκο και δεν θέλουμε να εμφανισθούν όλα παρά μόνο εκείνα που έχουν κάποια ιδιότητα, όπως για παράδειγμα τα εκτελέσιμα αρχεία (με κατάληξη com, exe, bat).

*Οι εντολές type, copy con, date, time δεν δέχονται χαρακτήρες μπαλαντέρ.*

Εντολή	Περιγραφή
Cls	Καθαρίζει την οθόνη
Date	Δείχνει την ημερομηνία
Time	Δείχνει την ώρα
Copy con	Δημιουργεί αρχείο
Type	Εμφανίζει το περιεχόμενο αρχείου
Dir	Εμφανίζει τα περιεχόμενα καταλόγου
Rename	Αλλάζει το όνομα αρχείου
Copy	Αντιγράφει αρχείο
Del	Διαγράφει αρχείο
*	Αντικαθιστά έναν ή περισσότερους χαρακτήρες
?	Αντικαθιστά μόνον ένα χαρακτήρα

**Μνημονικός πίνακας εκμάθησης εντολών διαχείρισης αρχείων**

## ΑΣΚΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ

**Άσκηση 1.** Να γράψετε την εντολή

- α) που αντιγράφει όλα τα αρχεία που βρίσκονται στον κατάλογο ρίζα της δισκέτας A και έχουν κατάληξη com στον κατάλογο ρίζα του σκληρού δίσκου C, με ακριβώς το ίδιο όνομα και την ίδια κατάληξη,
- β) δημιουργίας ενός αρχείου στον κατάλογο ρίζα της δισκέτας A με το όνομα nikos.txt,
- γ) διαγραφής όλων των αρχείων του καταλόγου ρίζα της δισκέτας A που το κυρίως όνομα τους είναι letter ενώ η επέκταση τους αποτελείται από 2 οποιαδήποτε γράμματα,
- δ) εμφάνισης του αρχείου nikos.doc που βρίσκεται στον κατάλογο ρίζα του σκληρού δίσκου C.
- ε) αλλαγής ονόματος του αρχείου kostas.txt που βρίσκεται στον κατάλογο ρίζα του σκληρού δίσκου C σε maria.txt

**Άσκηση 2.** Ποια από τα παρακάτω ονόματα αρχείων είναι σωστά ή λάθος και γιατί :

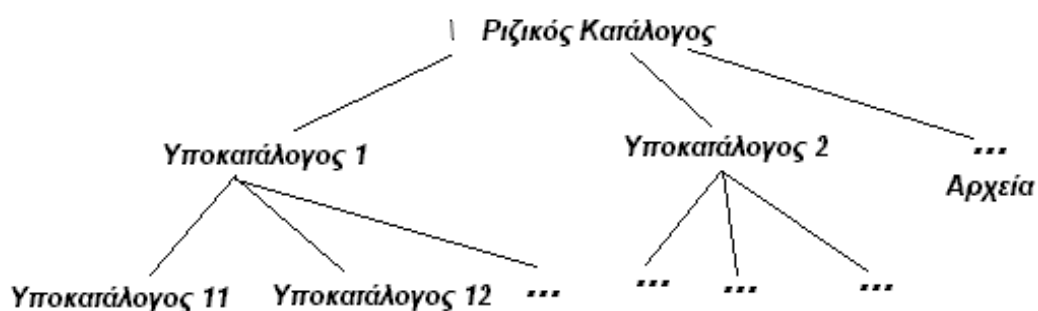
- α) KATERINA, β) 123.45, γ) TEXT.SONY, δ) N-1.DOC, ε) .TXT, ζ) TAXIS, η) F2.SOS, θ) N 3.4

## ΥΠΟΚΑΤΑΛΟΓΟΙ - ΜΟΝΟΠΑΤΙΑ

Η δημιουργία πολλών αρχείων στον ριζικό κατάλογο θα δημιουργούσε προφανώς προβλήματα στην εύρεση και εμφάνιση των αρχείων που ψάχνουμε. Κάντε για παράδειγμα έναν παραλληλισμό του αρχείου με ένα βιβλίο σε μια βιβλιοθήκη. Επίσης φανταστείτε ότι τον ρόλο του ριζικού καταλόγου παίζει ένα βιβλίο ευρετήριο όπου γράφουμε τους τίτλους των βιβλίων που έχουμε στην βιβλιοθήκη καθώς και την θέση τους στην βιβλιοθήκη. Αν η βιβλιοθήκη αυτή περιείχε 100000 βιβλία, καταλαβαίνετε ότι θα μας ήταν αδύνατο να βρούμε το συγκεκριμένο βιβλίο που ψάχνουμε μέσα από αυτό το βιβλίο ευρετήριο. Θα ήταν λοιπόν σκόπιμο στην συγκεκριμένη περίπτωση να έχουμε μια σειρά από άλλα ευρετήρια όπως για παράδειγμα «Ευρετήριο βιβλίων Ιστορίας», «Ευρετήριο βιβλίων Μαθηματικών» κ.τ.λ. στα οποία θα μας καλεί να ανατρέξουμε το βασικό ευρετήριο ανάλογα με τις προτιμήσεις μας. Τα συγκεκριμένα ευρετήρια με την σειρά τους θα μας καλούν να ανατρέξουμε σε άλλα ευρετήρια που καλύπτουν κάποιες ειδικότητες του επιλεγμένου ευρετηρίου κ.ο.κ. Η δεντρική αυτή δομή φαίνεται παρακάτω :

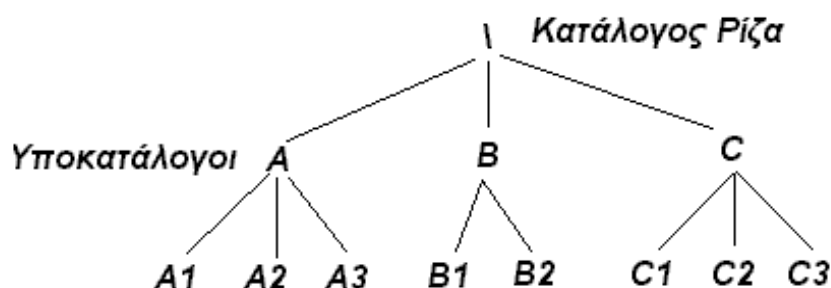


Την δεντρική αυτή δομή χρησιμοποιούμε και στο λειτουργικό σύστημα MS-DOS. Πιο συγκεκριμένα υπάρχει ο *ριζικός κατάλογος* (\) που είναι και ο πρωτεύον κατάλογος μέσα στον δίσκο. Στον κατάλογο αυτό περιέχονται ονόματα αρχείων αλλά και άλλων καταλόγων. Με τη σειρά τους αυτοί οι κατάλογοι περιέχουν μια ομάδα αρχείων με κοινά χαρακτηριστικά, άλλους καταλόγους ή και τίποτα κ.ο.κ. Όλοι οι υπόλοιποι κατάλογοι εκτός του ριζικού καταλόγου ονομάζονται *υποκατάλογοι* (subdirectories).



**Παρατήρηση :** Οι ονομασίες των καταλόγων ακολουθούν την ίδια ονοματολογία με αυτή των αρχείων που αναφέραμε σε προηγούμενο κεφάλαιο.

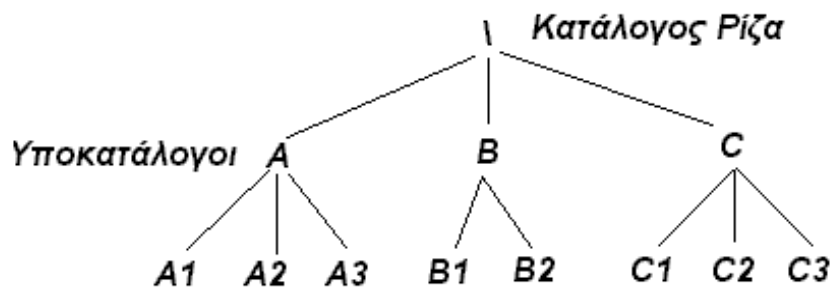
**Παράδειγμα.** Θεωρείστε ότι έχουμε ένα Γυμνάσιο στο οποίο η πρώτη τάξη έχει 3 τμήματα, η δεύτερα τάξη έχει 2 τμήματα και η τρίτη τάξη έχει 3 τμήματα. Θέλουμε λοιπόν να δημιουργήσουμε μια δενδρική δομή στον δίσκο για να αποθηκεύουν οι μαθητές τις εργασίες τους και να μην συνυπάρχουν όλες στον ριζικό κατάλογο. Η δενδρική δομή που θα διαλέξουμε είναι η εξής :



Ως *μονοπάτι* ορίζουμε μια ακολουθία καταλόγων, χωριζόμενων με το σύμβολο \, στην οποία κάθε κατάλογος αποτελεί υποκατάλογο αυτού που βρίσκεται στα αριστερά του. Για παράδειγμα η ακολουθία \WINAPPL\EGRAFA είναι ένα μονοπάτι στο οποίο ο κατάλογος EGRAFA είναι υποκατάλογος του WINAPPL ο οποίος με τη σειρά του είναι υποκατάλογος του καταλόγου ρίζα. Ένα μονοπάτι εκφράζει σε γενικές γραμμές την πορεία αναζήτησης που θα ακολουθήσει το MS-DOS, στην

προσπάθεια του να εντοπίσει ένα αρχείο ή ομάδα αρχείων. Εάν η πορεία αυτή ξεκινάει από τον κατάλογο ρίζα τότε ονομάζω το μονοπάτι αυτό και *απόλυτη διαδρομή* ενώ στην περίπτωση που η πορεία αυτή ξεκινάει από τον κατάλογο στον οποίο βρισκόμαστε εκείνη την στιγμή τότε το μονοπάτι αυτό ονομάζεται και *σχετική διαδρομή*. Συγκεκριμένα η σχετική διαδρομή γράφεται αν από την απόλυτη διαδρομή αφαιρέσουμε το όνομα της διαδρομής στην οποία βρισκόμαστε εκείνη την στιγμή και την οποία καταλαβαίνουμε πάντα από το σήμα ετοιμότητας, καθώς και το διαχωριστικό \.

**Παράδειγμα.** Θεωρείστε την δενδρική δομή του προηγούμενου παραδείγματος :



- α) Να αναφέρετε την απόλυτη διαδρομή όλων των υποκαταλόγων.
- β) Εάν το σήμα ετοιμότητας είναι το C:\A> να αναφέρετε την σχετική διαδρομή του υποκαταλόγου A1.

### Απάντηση

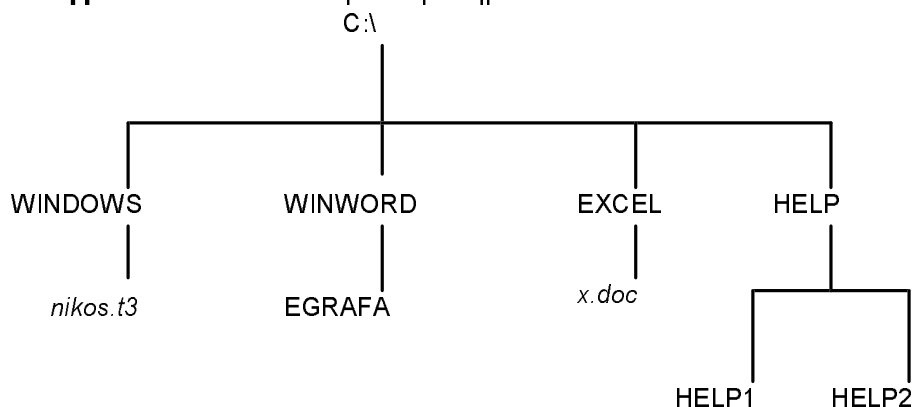
α)

Κατάλογος	Ονομασία απόλυτης διαδρομής
Κατάλογος ρίζα	\
A	\A
B	\B
C	\C
A1	\A\A1
A2	\A\A2
A3	\A\A3
B1	\B\B1
B2	\B\B2
C1	\C\C1
C2	\C\C2
C3	\C\C3

Πρέπει να τονίσουμε εδώ ότι ενώ το πρώτο \ στην απόλυτη διαδρομή δηλώνει τον κατάλογο ρίζα, τα επόμενα \ είναι απλώς διαχωριστικά για να ξεχωρίζουμε τα ονόματα των καταλόγων.

β) Η σχετική διαδρομή προέρχεται από την απόλυτη διαδρομή \A\A1 αν αφαιρέσουμε την θέση στην οποία βρισκόμαστε εκείνη την στιγμή και το διαχωριστικό \ (δηλ. \A\ ) και άρα είναι A1.

**Παράδειγμα.** Δίνεται το δέντρο ευρετηρίου :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων.

α) Να δώσετε τις ονομασίες των αρχείων *nikos.t3* και *x.doc*.

β) Να δώσετε τις ονομασίες των υποκαταλόγων HELP1, EXCEL και EGRAFA.

### **Απάντηση**

α) C:\WINDOWS\nikos.t3 και C:\EXCEL\x.doc

β) C:\HELP\HELP1, C:\EXCEL και C:\WINWORD\EGRAFA

## **ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΤΑΛΟΓΩΝ**

Κάποια εύλογα ερωτήματα που προκύπτουν από την προηγούμενη ενότητα είναι τα εξής : α) με ποιον τρόπο δημιουργούμε έναν υποκατάλογο, β) με ποιον τρόπο πηγαίνουμε σε έναν υποκατάλογο, γ) με ποιόν τρόπο διαγράφουμε έναν υποκατάλογο και τέλος με ποιον τρόπο βλέπουμε την δενδρική δομή του δίσκου μας. Στα ερωτήματα αυτά θα απαντήσουμε δίνοντας τις παρακάτω εντολές διαχείρισης καταλόγων του MS-DOS.

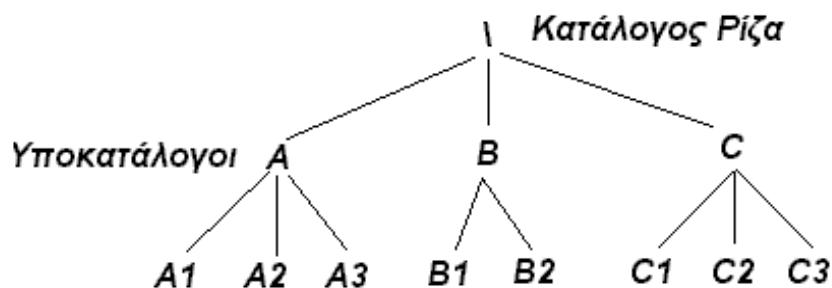


## Η ΕΝΤΟΛΗ MD

**Σύνταξη :** md [οδηγός δίσκου :]μονοπάτι

**Σκοπός :** Δημιουργεί έναν καινούργιο υποκατάλογο. Απαραίτητη προϋπόθεση δημιουργίας ενός υποκαταλόγου B μέσα σε έναν υποκατάλογο A, είναι η ύπαρξη του υποκαταλόγου A.

**Παράδειγμα.** Θεωρείστε ότι ο σκληρός δίσκος του Η/Υ δεν περιέχει υποκαταλόγους. Δημιουργήστε την παρακάτω δενδρική δομή στον σκληρό δίσκο C :



**Απάντηση.** Οι εντολές που θα χρησιμοποιήσουμε είναι οι εξής :

```

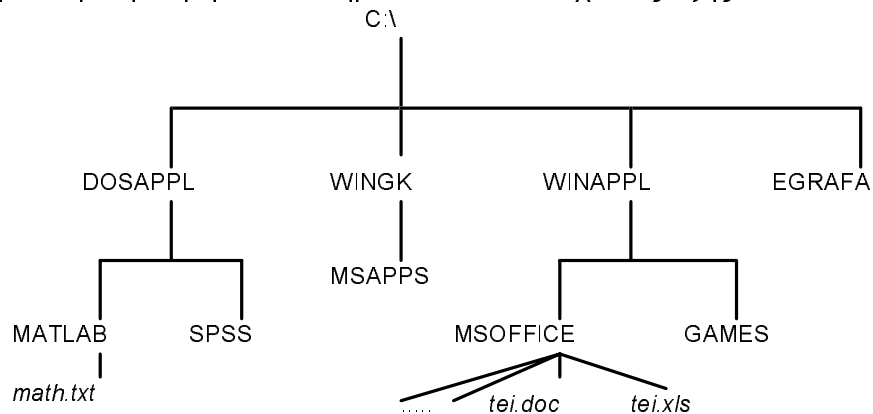
Md \a
Md \b
Md \c
Md \a\1
Md \a\2
Md \a\3
Md \b\1
Md \b\2
Md \c\1
Md \c\2
Md \c\3
  
```

Πρώτα δηλαδή δημιουργούμε τους υποκαταλόγους A, B, C όπου θα στεγαστούν οι υπόλοιποι κατάλογοι και τέλος δημιουργώ τους καταλόγους A1, A2, A3, B1, B2, C1, C2, C3 με οποιαδήποτε σειρά θέλω εγώ.

**Παράδειγμα.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\WINAPPL>\_

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Θεωρείστε ότι οι υποκατάλογοι MSOFFICE και GAMES δεν υπάρχουν και θέλω να τους δημιουργήσω. Ποιες εντολές θα χρησιμοποιήσω ;

**Απάντηση.**

Εάν χρησιμοποιήσω την *απόλυτη διαδρομή* οι εντολές που θα δώσω θα είναι οι εξής :

`Md \winappl\msoffice`

`Md \winappl\games`

Ενώ αν χρησιμοποιήσω την *σχετική διαδρομή* (λόγω του ότι είμαι στον κατάλογο \winappl) οι εντολές που θα δώσω είναι οι εξής :

`Md office`

`Md games`

## **H ΕΝΤΟΛΗ CD**

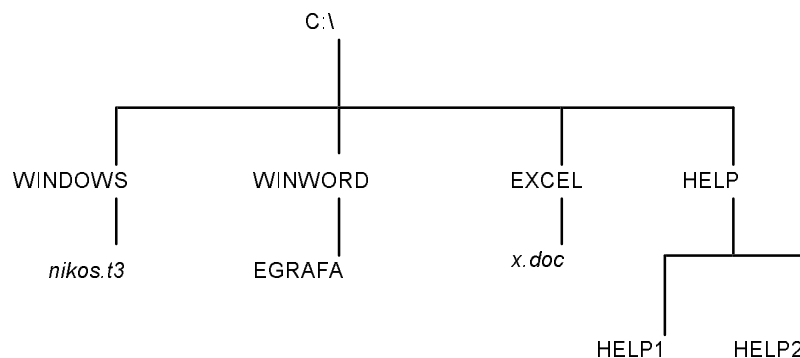
**Σύνταξη :** `cd [οδηγός δίσκου :]μονοπάτι`

**Σκοπός :** Μεταφορά από τον τρέχων υποκατάλογο (ευρετήριο) σε άλλον υποκατάλογο. Χρησιμοποιούνται επίσης οι εντολές `cd..` για να οδηγηθούμε στον προηγούμενο κατάλογο καθώς και η `cd \` για να οδηγηθούμε στον κατάλογο ρίζα.

**Παράδειγμα.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\ HELP\HELP1>

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε :

- α) για να μεταφερθείτε στον υποκατάλογο HELP,
- β) για να μεταφερθείτε στον υποκατάλογο EGRAFA,
- γ) για να μεταφερθείτε στον κατάλογο ρίζα

### Απάντηση

- α) `cd \HELP` ή `cd..`
- β) `cd \WINWORD\EGRAFA`
- γ) `cd \`

## H ΕΝΤΟΛΗ TREE

**Σύνταξη :** `tree [οδηγός δίσκου :] [/f]`

**Σκοπός :** Παρουσιάζει όλη την δενδρική δομή του προτεινόμενου οδηγού δίσκου. Εάν παραλείψουμε το όνομα του οδηγού δίσκου, παρουσιάζει την δενδρική δομή του τρέχοντος οδηγού δίσκου. Η προαιρετική παράμετρος `/f`

χρησιμοποιείται όταν θέλω να δω επιπλέον των υποκαταλόγων και τα αρχεία που βρίσκονται μέσα σε αυτούς.

## Η ΕΝΤΟΛΗ RD

**Σύνταξη :** rd [οδηγός δίσκου :]μονοπάτι

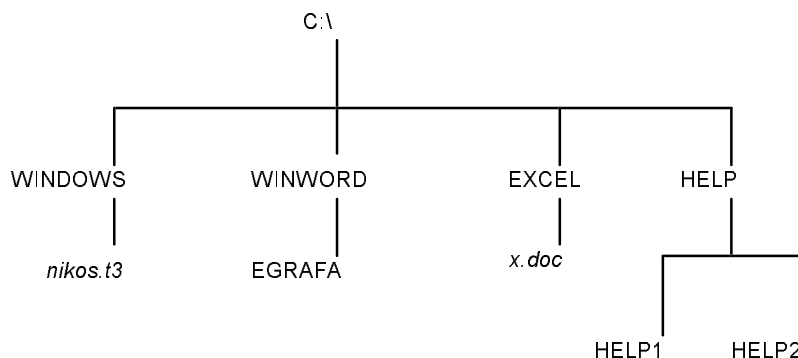
**Σκοπός :** Διαγράφει τον προτεινόμενο υποκατάλογο. Απαραίτητη προϋπόθεση διαγραφής του συγκεκριμένου κατάλογου είναι : α) να μην υπάρχουν άλλα αρχεία ή κατάλογοι στον προτεινόμενο προς διαγραφή κατάλογο και β) να μην εκτελούμε την εντολή αυτή από τον κατάλογο που θέλουμε να διαγράψουμε, διαφορετικά εμφανίζεται το παρακάτω μήνυμα λάθους :

*Invalid path, not directory, or directory not empty*

**Παράδειγμα.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\ HELP\HELP1>

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε :

- α) για να διαγράψετε τον υποκατάλογο HELP,
- β) για να διαγράψετε τον υποκατάλογο EGRAFA,
- γ) για να διαγράψετε τον υποκατάλογο WINDOWS

## Απάντηση

α) Πρώτα πρέπει να βγούμε από τον κατάλογο που θέλουμε να διαγράψουμε και συνεπώς εκτελούμε την εντολή :

```
cd \
```

που μας μεταφέρει στον κατάλογο ρίζα. Στην συνέχεια διαγράφουμε πρώτα τους καταλόγους HELP1 και HELP2 που βρίσκονται μέσα στον κατάλογο HELP με τις εντολές :

```
rd \HELP\HELP1
```

```
rd \HELP\HELP2
```

και στην συνέχεια διαγράφουμε τον κατάλογο HELP με την εντολή

```
rd \HELP
```

β) Εφόσον ήδη βρισκόμαστε εκτός του καταλόγου EGRAFA, θα εκτελέσουμε την εντολή :

```
rd \WINWORD\EGRAFA
```

γ) Εφόσον βρίσκεται ήδη ένα αρχείο με το όνομα nikos.t3 μέσα στον κατάλογο WINDOWS, θα πρέπει πρώτα να διαγράψουμε το αρχείο αυτό με την εντολή :

```
del \WINDOWS\nikos.t3
```

ή `del \WINDOWS\*.*` (σε περίπτωση περισσοτέρων αρχείων)

και στην συνέχεια θα διαγράψουμε τον κατάλογο WINDOWS με την εντολή :

```
rd \WINDOWS
```

## ***H ENTOLH DELTREE***

**Σύνταξη :** `deltree [οδηγός δίσκου :]μονοπάτι`

**Σκοπός :** Διαγράφει τον προτεινόμενο υποκατάλογο καθώς και όλους τους υποκαταλόγους ή αρχεία που εμπεριέχονται μέσα σ' αυτόν. Για τον λόγο αυτό πλεονεκτεί έναντι της εντολής `rd`, αλλά ταυτόχρονα χρειάζεται και περισσότερη προσοχή. Μετά την εκτέλεση της παραπάνω εντολής πάντα εμφανίζεται το μήνυμα :

*Delete directory 'όνομα υποκαταλόγου' and all its subdirectories? [yn]*

Στην περίπτωση που θέλουμε να σβήσουμε τον κατάλογο αυτό και τους υποκαταλόγους του απαντάμε με `y` διαφορετικά με `n` και στην συνέχεια

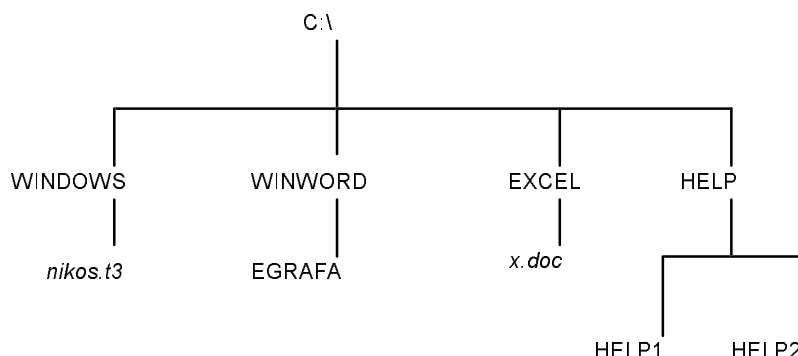
πατάμε ENTER. Το MS-DOS στην συνέχεια καθώς διαγράφει τον κατάλογο αυτό και τους υποκαταλόγους του μας απαντάει με το μήνυμα :

*Deleting 'όνομα υποκαταλόγου'...*

**Παράδειγμα.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\ >

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε :

- α) για να διαγράψετε τον υποκατάλογο HELP,
- β) για να διαγράψετε τον υποκατάλογο WINWORD,
- γ) για να διαγράψετε τον υποκατάλογο WINDOWS

**Απάντηση**

- A) deltree \HELP
- B) deltree \WINWORD
- Γ) deltree \WINDOWS

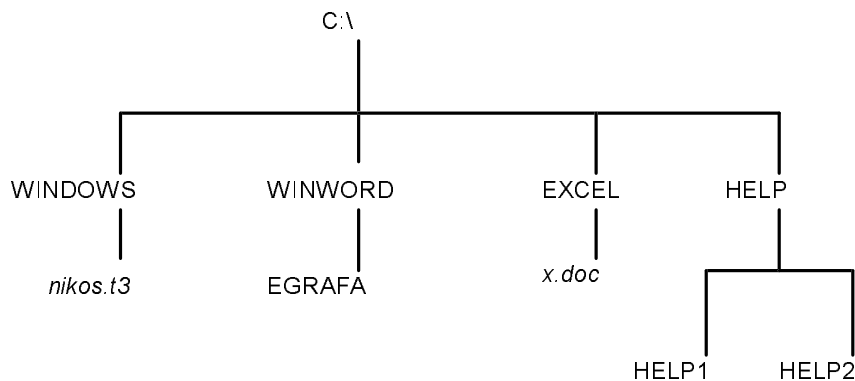
Εντολή	Περιγραφή
md	Δημιουργεί υποκατάλογο
cd	Αλλάζει τον τρέχοντα υποκατάλογο
rd	Διαγράφει υποκατάλογο
deltree	Διαγράφει έναν υποκατάλογο καθώς και ότι εμπεριέχει
tree	Εμφανίζει την δενδρική δομή του δίσκου

**Μνημονικός πίνακας εκμάθησης εντολών διαχείρισης καταλόγων**

**Προσοχή :** Δεν χρησιμοποιούμε τους χαρακτήρες μπαλαντέρ (\*, ?) στην διαχείριση καταλόγων.

## ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΧΕΙΩΝ ΚΑΙ ΚΑΤΑΛΟΓΩΝ.

**Άσκηση 1.** Δίνεται το δέντρο ευρετηρίου :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε :

- α) για να αντιγραφούν όλα τα αρχεία του ευρετηρίου EGRAFA στο HELP1,
- β) για να μετονομαστούν όλα τα αρχεία του ευρετηρίου EGRAFA με προέκταση .doc σε .txt,
- γ) για να διαγράψετε τα αρχεία του ευρετηρίου HELP2, που το πρώτο τους γράμμα είναι *k*, το δεύτερο γράμμα οποιοδήποτε, τα επόμενα τρία *kos*, και η επέκταση είναι *.dat*,
- δ) για να δημιουργήσετε ένα αρχείο με το όνομα *kostas.dat* στο ευρετήριο HELP1,
- ε) για να εμφανίσω το περιεχόμενο του αρχείου *nikos.t3*,
- στ) για να δημιουργήσω ένα ευρετήριο με το όνομα HELP μέσα στο ευρετήριο WINDOWS,
- ζ) για να διαγράψω το ευρετήριο HELP.

### Απάντηση

- α) `copy \WINWORD\EGRAFA\*. * \HELP\HELP1`

Επειδή θέλουμε τα αρχεία αυτά να έχουν το ίδιο κυρίως όνομα και επέκταση, μπορούμε να παραλείψουμε το όνομα του αρχείου στην πηγή προορισμού (δηλ. έχω \HELP\HELP1 αντί \HELP\HELP1\\*.\*)).

β)

1<sup>ος</sup> τρόπος. Πηγαίνω πρώτα στον συγκεκριμένο κατάλογο και στην συνέχεια αλλάζω όνομα στα αρχεία :

### Cd \WINWORD\EGRAFA

Rename \*.doc \* .txt

2<sup>ος</sup> τρόπος. Εκτελώ κατευθείαν την εντολή rename :

```
rename \WINWORD\EGRAFA\*.doc *.txt
```

Προσοχή : Δεν χρησιμοποιώ μονοπάτι στο καινούργιο όνομα του αρχείου.

γ) del \HELP\HELP2\k?kos.dat

δ) copy con \HELP\HELP1\kostas.dat

ε) type \WINDOWS\nikos.t3

στ) md \WINDOWS\HELP

ζ) deltree \HELP

ή

rd \HELP\HELP1

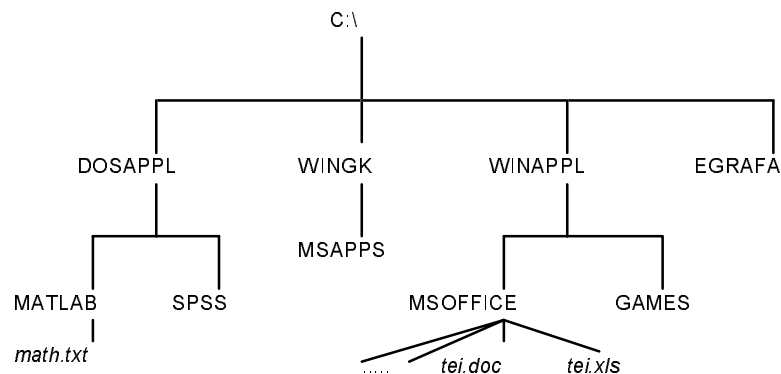
rd \HELP\HELP2

rd \HELP

**Άσκηση 2.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\>\_

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :





όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε :

- α) για να εμφανισθεί το περιεχόμενο του αρχείου *math.txt*,
- β) για να αλλάξετε το όνομα του αρχείου *math.txt* σε *math.doc*,
- γ) για να αντιγράψετε όλα τα αρχεία του καταλόγου MSOFFICE που έχουν κατάληξη DOC στον κατάλογο EGRAFA,
- δ) για να διαγραφούν όλα τα αρχεία του καταλόγου WINGK που το κυρίως όνομα τους έχει δύο γράμματα εκ των οποίων το πρώτο γράμμα είναι το X ενώ έχουν επέκταση DVI,
- ε) για να δημιουργηθεί ένας κατάλογος με το όνομα TEI στον υποκατάλογο WINAPPL,
- στ) για να διαγραφεί ο κατάλογος MSOFFICE,
- ζ) για να πάτε στον υποκατάλογο SPSS.

### Απάντηση

α) type \DOSAPPL\MATLAB\math.txt

β)

*1<sup>ος</sup> τρόπος.* Πηγαίνω πρώτα στον συγκεκριμένο κατάλογο και στην συνέχεια αλλάζω όνομα στα αρχεία :

**Cd \DOSAPPL\MATLAB**

Rename math.txt math.doc

*2<sup>ος</sup> τρόπος.* Εκτελώ κατευθείαν την εντολή rename :

rename \DOSAPPL\MATLAB\math.txt math.doc

Προσοχή : Δεν χρησιμοποιώ μονοπάτι στο καινούργιο όνομα του αρχείου.

γ) copy \WINAPPL\OFFICE\\*.doc \EGRAFA

δ) del \WINGK\X?.DVI

ε) md \WINAPPL\TEI

στ) deltree \WINAPPL\MSOFFICE

ή

Del \WINAPPL\MSOFFICE\\*.\*

Rd \WINAPPL\MSOFFICE

ζ) cd \DOSAPPL\SPSS

## ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΣΚΩΝ

Παρακάτω θα αναφέρουμε δύο από τις πιο βασικές εντολές διαχείρισης δίσκων.

### Η ΕΝΤΟΛΗ *FORMAT*

**Σύνταξη :** format οδηγό δίσκου : [/s]

**Σκοπός :** Μορφοποιεί την δισκέτα που βρίσκεται στον συγκεκριμένο οδηγό δίσκου ούτως ώστε να δέχεται αρχεία του MS-DOS. Κατά την διαδικασία της μορφοποίησης δημιουργείται ο κατάλογος ρίζα καθώς και ο File Allocation Table (FAT) (δες θεωρία Προγραμματισμού Η/Υ Ι) όπου θα καταχωρούνται οι διευθύνσεις στην μνήμη των δημιουργηθέντων αρχείων. Πρέπει να χρησιμοποιούμε την εντολή αυτή σε όλες τις καινούργιες δισκέτες εκτός της περιπτώσεως που αναγράφεται πάνω στην δισκέτα ότι είναι ήδη μορφοποιημένη από τον κατασκευαστή (formatted). Η παράμετρος /s χρησιμοποιείται όταν θέλουμε να εγκατασταθούν στην δισκέτα κατά την μορφοποίηση της, τα κρυφά αρχεία MSDOS.SYS και IO.SYS που είναι απαραίτητα για την εκκίνηση του Η/Υ από την δισκέτα.

#### Παράδειγμα.

Format a:

Μορφοποιείται η δισκέτα που βρίσκεται στον οδηγό A.

### Η ΕΝΤΟΛΗ *DISKCOPY*

**Σύνταξη :** diskcopy [οδηγό πηγαίου δίσκου 1] [οδηγό δίσκου προορισμού 2]

**Σκοπός :** Αντιγράφει το περιεχόμενο της δισκέτας που βρίσκεται στον πηγαίο οδηγό, στην μορφοποιημένη (formatted) ή μη μορφοποιημένη δισκέτα του οδηγού δίσκου προορισμού. Στην περίπτωση που η δισκέτα προορισμού δεν είναι μορφοποιημένη, το MS-DOS την μορφοποιεί. Η παραπάνω εντολή κατά την διάρκεια εκτέλεσης της θα σας προτρέψει να τοποθετείτε την πηγαία δισκέτα (source diskette) δηλ. την δισκέτα που θέλετε να αντιγράψετε, καθώς

και την δισκέτα προορισμού (target diskette) στην οποία θέλετε να αντιγραφούν οι πληροφορίες. Μετά το τέλος της αντιγραφής θα σας παρουσιαστεί το μήνυμα :

*Copy another diskette (Y/N)?*

Θα απαντήσετε με Y εάν θέλετε να αντιγράψετε και δεύτερη δισκέτα ή διαφορετικά με N.

### **Παραδείγματα.**

Diskcopy a: b:

Αντιγράφει το περιεχόμενο της δισκέτας που βρίσκεται στον οδηγό a, στην δισκέτα που βρίσκεται στον οδηγό b. Πριν την διαδικασία της αντιγραφής σας προτρέπει να τοποθετήσετε την δισκέτα που θέλετε να αντιγράψετε (source diskette) στον οδηγό a και την δισκέτα στην οποία θέλετε να αντιγραφούν οι πληροφορίες (target diskette) να την τοποθετήσετε στον οδηγό b.

Diskcopy a: a: (ή Diskcopy a:)

Αντιγράφει το περιεχόμενο της πηγαίας δισκέτας (source diskette) που βρίσκεται στον οδηγό a, στην δισκέτα προορισμού (target diskette) που βρίσκεται πάλι στον οδηγό a. Για να γίνει εφικτό αυτό, το MS-DOS πάντα σας ενημερώνει για το ποια δισκέτα θα τοποθετείται στον οδηγό a.

**Προσοχή :** Στην περίπτωση που η δισκέτα προορισμού (target diskette) περιέχει αρχεία, αυτά θα καταστραφούν κατά την διαδικασία εκτέλεσης της εντολής diskcopy.

### **ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΣΤΟ MS-DOS.**

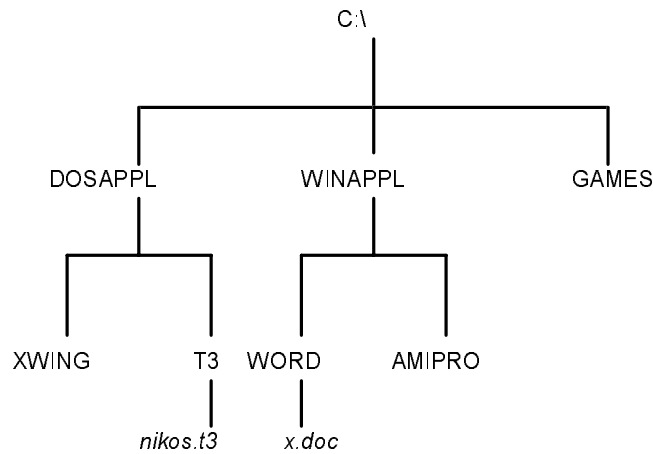
**Άσκηση 1.** Ποια από τα παρακάτω ονόματα αρχείων και καταλόγων είναι σωστά ή λάθος και γιατί :

Anakoinosi, petros, .kos, kos., dir, νικος.doc, n\_12.doc, n\1.d, n12.3456, 1ab.doc, ba1.doc, cia.

**Άσκηση 2.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\ >\_

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :



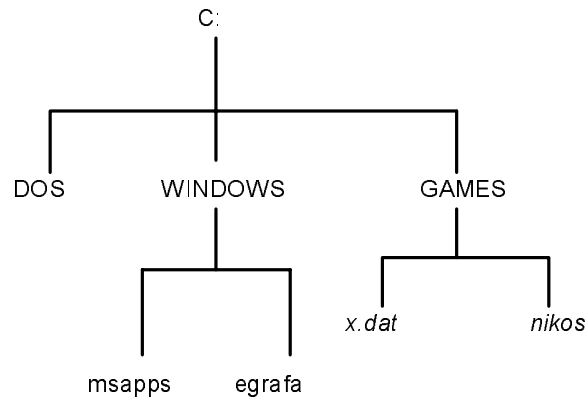
όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε

- α) για να δημιουργήσετε τον υποκατάλογο XWING στον C:\GAMES ;
- β) για να αντιγράψετε όλα τα αρχεία του υποκαταλόγου XWING που βρίσκεται στον C:\DOSAPPL, στον αντίστοιχο υποκατάλογο XWING του C:\GAMES που δημιουργήσατε στο ερώτημα (α) ;
- γ) για να εκτυπώσετε το περιεχόμενο του αρχείου x.doc που βρίσκεται στον υποκατάλογο WORD του C:\WINAPPL ;
- δ) για να αλλάξετε το όνομα του αρχείου nikos.t3 σε kostas.t3 ;
- ε) για να πάτε στον υποκατάλογο T3 του C:\DOSAPPL ;

**Άσκηση 3.** Θεωρείστε ότι στην οθόνη του Η/Υ υπάρχει το σήμα :

C:\DOS>\_

ενώ η δενδρική δομή του σκληρού δίσκου C έχει ως εξής :



όπου με πλάγια γράμματα παριστάνονται τα ονόματα αρχείων ενώ με κανονικά γράμματα τα ονόματα υποκαταλόγων. Ποιες εντολές του λειτουργικού συστήματος MS-DOS θα χρησιμοποιήσετε :

- α) για να διαγράψετε τον υποκατάλογο C:\GAMES ;
- β) για να μεταβείτε στον υποκατάλογο egrafa του C:\WINDOWS ;
- γ) για να διαγράψετε το αρχείο nikos,
- δ) για να δημιουργήσετε ένα αρχείο με το όνομα tei97 στον υποκατάλογο msapps,
- ε) για να εμφανίσετε το περιεχόμενο του αρχείου x.dat
- στ) για να αλλάξετε το όνομα του αρχείου nikos σε tei96,
- ζ) για να δημιουργήσετε έναν κατάλογο με το όνομα tei μέσα στον υποκατάλογο windows.

## Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ BASIC

Η ανάγκη επικοινωνίας του ανθρώπου με τον Η/Υ σε μια κοινή γλώσσα ανέπτυξε τις διάφορες γλώσσες προγραμματισμού που υπάρχουν σήμερα. *Γλώσσα προγραμματισμού* είναι ένα σύνολο λέξεων, γραμμάτων, αριθμών και συντομογραφικών μνημονικών σημείων που διέπονται από ειδικό συντακτικό και χρησιμοποιούνται στην ανάπτυξη αλγορίθμων στον υπολογιστή. Η ποικιλία των γλωσσών προγραμματισμού που υπάρχει σήμερα οφείλετε στο ότι κάποιες γλώσσες υπερτερούν έναντι άλλων σε συγκεκριμένες εφαρμογές, είτε προσφέρουν καλύτερη υποστήριξη, είτε είναι εύκολες στην εκμάθηση τους κ.λ.π. Γενικά τις γλώσσες προγραμματισμού μπορούμε να τις χωρίσουμε σε δύο κατηγορίες :

Οι *γλώσσες χαμηλού επιπέδου* που βρίσκονται κοντά στις στοιχειώδεις εντολές μηχανής ενός συγκεκριμένου Η/Υ και γι' αυτό διαφέρουν από Η/Υ σε Η/Υ. Η γλώσσα που είναι πιο κοντά στο επίπεδο της μηχανής είναι η γλώσσα μηχανής. Η ανάγκη επίτευξης μια γλώσσας χαμηλού επιπέδου που να μοιάζει περισσότερο στην γλώσσα του ανθρώπου, χωρίς να ξεφεύγει από το επίπεδο της μηχανής δημιούργησε την γλώσσα *Assembly*. Υπάρχει μάλιστα ειδικό πρόγραμμα που ονομάζεται *Assembler* που αναλαμβάνει να μεταφράσει τα προγράμματα που είναι γραμμένα στην γλώσσα αυτή σε γλώσσα μηχανής.

Οι *γλώσσες υψηλού επιπέδου*, που προσεγγίζουν την ανθρώπινη γλώσσα περισσότερο από ότι οι γλώσσες χαμηλού επιπέδου και είναι ειδικά σχεδιασμένες ώστε προγράμματα τα οποία είναι γραμμένες σε αυτές να εκτελούνται από οποιονδήποτε υπολογιστή. Τέτοιες γλώσσες είναι οι BASIC, FORTRAN, PASCAL, PROLOG, C, κ.τ.λ. Το πρόγραμμα που γράφουμε σε μια γλώσσα υψηλού επιπέδου πάντοτε μεταφράζεται μέσω ειδικών προγραμμάτων στην γλώσσα που καταλαβαίνει ο Η/Υ δηλαδή την γλώσσα μηχανής. Τα προγράμματα που κάνουν την μετάφραση αυτή είναι ο *Interpreter* ή ο *Compiler*.

Interpreter (Διερμηνέας)	Μεταφράζει μια-μια τις εντολές συγχρόνως με την εκτέλεση τους.
Compiler (Μεταφραστής)	Μεταφράζει μια φορά ολόκληρο το πρόγραμμα. Στην συνέχεια το μεταφρασμένο πρόγραμμα εκτελείται.

Η BASIC, που θα ασχοληθούμε σ' αυτό το κεφάλαιο, είναι μια απλή γλώσσα προγραμματισμού που υπάρχει σε κάθε μικροϋπολογιστή και είναι κατάλληλη για την εισαγωγή αρχαρίων στον προγραμματισμό των Η/Υ. Έχουν ως τώρα εμφανισθεί πολλές διαφορετικές εκδόσεις της BASIC όπως η GW-BASIC, η QBASIC κ.τ.λ. με πιο πρόσφατη την Visual Basic 4.0 η οποία χρησιμοποιείται στο περιβάλλον των Windows. Εμείς θα ασχοληθούμε με εντολές της BASIC που είναι σχεδόν κοινές σε όλες τις παραπάνω εκδόσεις. Πρώτα όμως θεωρούμε σκόπιμο να αναφέρουμε τι είναι προγραμματισμός και ποια τα στάδια του προγραμματισμού.

## Η έννοια του προγραμματισμού

*Προγραμματισμός* είναι μια διαδικασία που αποτελείται από καθορισμένα στάδια, σε καθένα από τα οποία γίνονται διάφορες ενέργειες με σκοπό τον μετασχηματισμό του προβλήματος και της μεθόδου λύσης του σε μια μορφή που να είναι κατανοητή και αποδεκτή από τον Η/Υ.

Τα *στάδια του προγραμματισμού* είναι τα ακόλουθα :

1. Η αναγνώριση, ο ορισμός και ο καθορισμός των προδιαγραφών του προβλήματος.
2. Εντοπισμός των χρήσιμων εννοιών και απλοποίηση ορισμένων στοιχείων.
3. Σκιαγράφηση της λύσης και διάκριση ανεξαρτήτων διαδικασιών.
4. Επιλογή και περιγραφή ενός αλγόριθμου που να είναι πραγματοποιήσιμος, σαφής, γενικός αποτελεσματικός, ταχύς και οικονομικός.
5. Επιλογή της γλώσσας προγραμματισμού βάσει της φύσης του προβλήματος, των γνώσεων μας σε γλώσσες προγραμματισμού και τις δυνατότητες ή περιορισμούς που μας θέτει ο Η/Υ. Κωδικοποίηση του αλγορίθμου σε πρόγραμμα.
6. Έλεγχος προγράμματος για ανίχνευση λαθών.
7. Χρήση του προγράμματος με ποικίλα δεδομένα για να διαπιστωθούν τυχόν λάθη.

## Είσοδος-έξοδος στο περιβάλλον της BASIC

Στο περιβάλλον της BASIC μπορούμε να πάμε γράφοντας από το σήμα ετοιμότητας την εντολή GWBASIC και πατώντας ENTER.

C:\>GWBASIC (ENTER)

Η έξοδος από το περιβάλλον της BASIC επιτυγχάνεται γράφοντας την εντολή SYSTEM και πατώντας το πλήκτρο ENTER.

**ΠΡΟΣΟΧΗ** : Στο περιβάλλον της BASIC δεν ισχύουν οι εντολές που αναφέραμε για το λειτουργικό σύστημα MS-DOS. Πρόκειται για δύο εντελώς διαφορετικά περιβάλλοντα.

## Κανόνες γραφής προγράμματος στην BASIC

*XX ENTOLH1 (ΠΑΡΑΜΕΤΡΟΙ) (ENTER)*

*XY ENTOLH2 (ΠΑΡΑΜΕΤΡΟΙ) (ENTER)*

.....

1. Κάθε γραμμή του προγράμματος αρχίζει με έναν ακέραιο και θετικό αριθμό από το 0 μέχρι το 65529. Λέγεται αριθμός γραμμής και δείχνει την σειρά με την οποία θα εκτελεστούν οι εντολές του προγράμματος. Συνήθως δίνουμε τους αριθμούς 10,20,... (με βήμα 10) έτσι ώστε να έχουμε αργότερα την δυνατότητα να παρεμβάλουμε κάποιες εντολές στο πρόγραμμα.
2. Μετά από τον αριθμό γραμμής ακολουθεί η εντολή της BASIC μαζί με τις παραμέτρους της.
3. Μπορούμε να γράψουμε περισσότερες από μια εντολές στην ίδια γραμμή αρκεί να διαχωρίζονται με άνω και κάτω τελεία.
4. Στο τέλος κάθε γραμμής πρέπει να πατήσουμε το πλήκτρο ENTER για να καταχωρηθεί η εντολή αυτή στην μνήμη του Η/Υ και να μπει στην σειρά προς εκτέλεση.

### Παράδειγμα.

10 LET A=1 : LET B=2 (ENTER)

20 LET C=A+B (ENTER)



30 PRINT A,B,C (ENTER)

40 END (ENTER)

## Εντολές της BASIC

Από την στιγμή που βρεθούμε στο περιβάλλον της BASIC υπάρχουν δύο τρόποι να δώσουμε εντολές :

Ο πρώτος τρόπος που θεωρείται ο έμμεσος τρόπος είναι αυτός που αναφέραμε και παραπάνω στην σύνταξη του προγράμματος δηλ. γράφοντας πρώτα τον αριθμό γραμμής, μετά την εντολή και τέλος ENTER. Κάθε εντολή που γράφουμε με τον τρόπο αυτό αποθηκεύεται στην μνήμη του Η/Υ. Στην συνέχεια μπορούμε να προσθέσουμε ή να διαγράψουμε και άλλες εντολές. Τέλος όταν ζητήσουμε την εκτέλεση του προγράμματος αυτό θα εκτελεστεί κατά αύξουσα σειρά των γραμμών εντολών, ανεξάρτητα με ποια σειρά τις δώσαμε. Το πρόγραμμα αυτό επειδή είναι εγκαταστημένο πια στην μνήμη του Η/Υ μπορεί να εκτελεστεί όσες φορές θέλουμε.

Ο δεύτερος τρόπος που θεωρείται ο πλέον άμεσος είναι να δίνουμε τις εντολές χωρίς αρίθμηση και να πατάμε ENTER στο τέλος της κάθε εντολής. Τότε η εντολή αυτή εκτελείται αυτόματα και δεν παραμένει πλέον στην μνήμη του Η/Υ, με αποτέλεσμα αν χρειαστούμε να την ξαναεκτελέσουμε να πρέπει να την πληκτρολογήσουμε ξανά.

Στην συνέχεια θα αναφέρουμε εντολές ελέγχου της BASIC που χρησιμοποιούνται κατά τον δεύτερο τρόπο που αναφέραμε και οι οποίες μας βοηθούν στο να εμφανίσουμε ένα πρόγραμμα μας, να το αποθηκεύσουμε, να το εκτελέσουμε κ.τ.λ.

## Εντολές ελέγχου στην BASIC

### Η εντολή CLS

**Σκοπός :** Καθαρίζει την οθόνη.

**Σύνταξη :** CLS

**Παράδειγμα :** 10 CLS (με αριθμό γραμμής ή χωρίς)

## Η εντολή DELETE

**Σκοπός :** Διαγράφει γραμμές του προγράμματος..

**Σύνταξη :**

DELETE αριθμό γραμμής

(διαγράφει την εντολή με τον συγκεκριμένο αριθμό γραμμής)

DELETE αριθμό γραμμής –

(διαγράφει τις εντολές από τον συγκεκριμένο αριθμό γραμμής και μετά)

DELETE - αριθμό γραμμής

(διαγράφει τις εντολές έως και τον συγκεκριμένο αριθμό γραμμής)

DELETE αριθμό γραμμής 1 - αριθμό γραμμής 2

(διαγράφει τις εντολές από τον αριθμό γραμμής 1 έως και τις εντολές του αριθμού γραμμής 2)

**Παράδειγμα.**

DELETE 40

Διαγράφει τον αριθμό γραμμής 40

DELETE 40 –

Διαγράφει όλες τις γραμμές από την 40 και μετά.

DELETE – 40

Διαγράφει όλες τις γραμμές έως και την 40.

DELETE 30-60

Διαγράφει όλες τις γραμμές από την 30 έως και την 60.

**Σημείωση :** Η εντολή DELETE δεν χρησιμοποιείται ως εντολή μετά από αριθμό γραμμής.

## Η εντολή LIST

**Σκοπός :** Παρουσιάζει όλο ή μέρος του προγράμματος που υπάρχει στην μνήμη του Η/Υ.

**Σύνταξη :** LIST [αριθμός γραμμής] [-[αριθμός γραμμής]]

**Παράδειγμα :**

LIST

Εμφανίζει όλο το πρόγραμμα.

LIST 30-

Εμφανίζει το πρόγραμμα από την εντολή με αριθμό γραμμής 30 και μετά.

LIST -50

Εμφανίζει το πρόγραμμα έως και την εντολή με αριθμό γραμμής 50.

LIST 30-60

Εμφανίζει το μέρος του προγράμματος μεταξύ των εντολών 30 και 60.

**Σημείωση :** Η εντολή LLIST κάνει ακριβώς την ίδια διαδικασία με την μόνη διαφορά ότι τα αποτελέσματα πηγαίνουν στον εκτυπωτή. Η εντολή LIST δεν χρησιμοποιείται ως εντολή μετά από αριθμό γραμμής.

**Η εντολή LOAD**

**Σκοπός :** Καλεί το πρόγραμμα μας που το έχουμε γράψει με την μορφή αρχείου από τον οδηγό δίσκου στην μνήμη RAM του Η/Υ.

**Σύνταξη :** LOAD όνομα αρχείου[,R]

Η προαιρετική παράμετρος R χρησιμοποιείται όταν θέλουμε εκτός από το κάλεσμα του προγράμματος και την εκτέλεση του.

**Παράδειγμα :**

LOAD "A:\TEST.BAS",R

Καλεί το πρόγραμμα TEST.BAS που βρίσκεται στον κατάλογο ρίζα της δισκέτας A.

**Σημείωση :** Η εντολή LOAD δεν χρησιμοποιείται ως εντολή μετά από αριθμό γραμμής.

## Η εντολή NEW

**Σκοπός :** Διαγράφει το πρόγραμμα από την μνήμη RAM του Η/Υ.

**Σύνταξη :** NEW

**Παράδειγμα :** NEW

**Σημείωση :** Εάν μετά από την εκτέλεση ενός προγράμματος το οποίο έχουμε φορτώσει στην μνήμη RAM θέλουμε να γράψουμε ένα καινούργιο πρόγραμμα απαιτητήτως πρέπει να εκτελέσουμε την εντολή NEW. Διαφορετικά οι γραμμές του καινούργιου προγράμματος θα ενσωματωθούν στις γραμμές του παλιού προγράμματος. Η εντολή NEW δεν χρησιμοποιείται ως εντολή μετά από αριθμό γραμμής.

## Η εντολή RENUM

**Σκοπός :** Αλλάζει την αρίθμηση των γραμμών του προγράμματος.

**Σύνταξη :** RENUM [[καινούργια γραμμή][,[παλιά γραμμή][,βήμα]]

**Παράδειγμα :**

RENUM

Η πρώτη γραμμή θα έχει νούμερο 10 ενώ το βήμα θεωρείται εφόσον δεν ορίστηκε το 10 δηλ. οι γραμμές θα έχουν ως αριθμούς τους 10,20,30,...

RENUM 200,,50

Η πρώτη γραμμή θα έχει νούμερο το 200 ενώ το βήμα θα είναι 50 δηλ. τα νούμερα των γραμμών θα είναι 200,250,300,.....

RENUM 1300,900,50

Αλλάζει την αρίθμηση των γραμμών από την 900 και μετά. Έτσι η γραμμή που είχε το νούμερο 900 τώρα θα έχει 1300, ενώ οι υπόλοιπες που ακολουθούν θα είναι οι 1350, 1400, .... Εφόσον το βήμα είναι 50.

**Σημείωση :** Η εντολή αυτή χρησιμοποιείται όταν με την συνεχή παρεμβολή γραμμών η αρίθμηση γίνει τέτοια ώστε να μην υπάρχουν αρκετά κενά για

άλλες παρεμβολές. Η εντολή RENUM δεν χρησιμοποιείται ως εντολή μετά από αριθμό γραμμής.

## Η εντολή RUN

**Σκοπός :** Εκτελεί ένα πρόγραμμα που βρίσκεται στην μνήμη RAM του Η/Υ ή φορτώνει από τον δίσκο ένα πρόγραμμα στην μνήμη RAM και στην συνέχεια το εκτελεί.

### Σύνταξη :

```
RUN [αριθμός γραμμής]  
RUN "όνομα αρχείου",[R]
```

### Παράδειγμα :

```
RUN
```

Εκτελεί το πρόγραμμα που είναι αυτή τη στιγμή στη μνήμη RAM του Η/Υ.

```
RUN "A:\TEST.BAS",R
```

Φορτώνει στην μνήμη RAM το πρόγραμμα TEST.BAS που βρίσκεται στον κατάλογο ρίζα της δισκέτας A και το εκτελεί.

```
RUN 100
```

Εκτελεί το πρόγραμμα που είναι αυτή τη στιγμή στη μνήμη RAM του Η/Υ από την εντολή 100 και μετά.

## Η εντολή SAVE

**Σκοπός :** Αποθηκεύει ένα πρόγραμμα στον δίσκο του Η/Υ.

**Σύνταξη :** SAVE "όνομα αρχείου" [{A|P}]

### Παράδειγμα :

```
SAVE "TEST.BAS"
```

**Σημείωση :** Εάν το όνομα του αρχείου δεν έχει επέκταση τότε αυτόματα τοποθετείται ως επέκταση τα τρία γράμματα BAS. Εάν δεν χρησιμοποιήσω καμιά από τις παραμέτρους τότε το αρχείο αποθηκεύεται σε συμπιεσμένη

δυναδική μορφή. Εάν χρησιμοποιήσω την παράμετρο P, τότε το αρχείο αποθηκεύεται σε κωδικοποιημένη δυναδική μορφή. Το πλεονέκτημα αυτής της αποθήκευσης είναι ότι κανένας άλλος χρήστης αλλά ούτε και εμείς έχουμε την δυνατότητα να δούμε το πρόγραμμα εκτελώντας την εντολή LIST. Αυτό είναι όμως ταυτόχρονα και μειονέκτημα διότι δεν έχουμε την δυνατότητα να δούμε το πρόγραμμα για να κάνουμε τις κατάλληλες διορθώσεις. Συνιστάται η αποθήκευση αυτή όταν έχουμε τελειώσει το πρόγραμμα μας και δεν θέλουμε κάποιος άλλος να αντιγράψουν την δουλειά μας. Η παράμετρος A, χρησιμοποιείται όταν θέλουμε να αποθηκεύσουμε το κείμενο σε μορφή ASCII. Στην περίπτωση αυτή μπορούμε να δούμε το κείμενο μας όχι μόνο από το περιβάλλον της BASIC αλλά και από έναν επεξεργαστή κειμένου είτε μέσω της εντολής TYPE. Η εντολή αυτή δεν μπορεί να χρησιμοποιηθεί μετά από γραμμή εντολής.

## Το αλφάβητο της BASIC

Στο αλφάβητο της BASIC περιέχονται όλοι οι αγγλικοί χαρακτήρες, όλοι οι αριθμοί καθώς και ειδικοί χαρακτήρες όπως =, +, -, \*, ... ..

## Σταθερές-Μεταβλητές στην BASIC

Σταθερές ονομάζουμε τις τιμές εκείνες στην BASIC που κατά την διάρκεια του προγράμματος δεν αλλάζουν τιμή. Τις σταθερές αυτές τις χωρίζουμε σε δύο κατηγορίες : τις *αριθμητικές* (μικροί ακέραιοι αριθμοί, μεγάλοι ακέραιοι αριθμοί, πραγματικοί αριθμοί απλής ακρίβειας, πραγματικοί αριθμοί διπλής ακρίβειας) και *αλφαριθμητικές* σταθερές (πρόκειται για μια σειρά από χαρακτήρες (γράμματα, αριθμούς ή σύμβολα) που βρίσκονται ανάμεσα σε εισαγωγικά.

ΤΥΠΟΣ	ΔΕΣΜΕΥΜΕΝΗ ΜΝΗΜΗ	ΠΕΡΙΟΧΗ ΤΙΜΩΝ	ΧΑΡΑΚΤΗΡΙΣΤΙΚΟ ΣΥΜΒΟΛΟ	ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΠΑΡΑΔΕΙΓΜΑ
ΜΙΚΡΟΙ ΑΚΕΡΑΙΟΙ	2 BYTES	-32768 - +32767	%	Μικρός χώρος αποθήκευσης. Μικρός χρόνος επεξεργασίας. Σημαντικά ψηφία (-5).	-12 0 425%
ΜΕΓΑΛΟΙ ΑΚΕΡΑΙΟΙ	4 BYTES	-2147483648 - +2147483647	&	Σχετικά μικρός χώρος αποθήκευσης. Σημαντικά ψηφία (-10).	345678 4232456&

ΠΡΑΓΜΑΤΙΚΟΙ ΑΠΛΗΣ ΑΚΡΙΒΕΙΑΣ	4 BYTES	+3.337x10 <sup>-38</sup> - +3.37x10 <sup>+38</sup> & -3.337x10 <sup>-38</sup> - -3.37x10 <sup>+38</sup>	! ή κανένα E για εκθετική μορφή	Σχετικά μικρός χώρος αποθήκευσης. Σημαντικά ψηφία (-7).	3456.3 42357! 0.12E-4
ΠΡΑΓΜΑΤΙΚΟΙ ΔΙΠΛΗΣ ΑΚΡΙΒΕΙΑΣ	8 BYTES	+1.67x10 <sup>-308</sup> - +1.67x10 <sup>+308</sup> & -1.67x10 <sup>-308</sup> - -1.67x10 <sup>+308</sup>	# ή κανένα D για εκθετική μορφή.	Μεγάλη ακρίβεια (-16 σημαντικά ψηφία)	123456789 24.34# 0.12D-4

### Σταθερές

Αλφαριθμητικές  
'ΝΙΚΟΣ'

Αριθμητικές  
3, 4.5, 1E-3

Ακέραιες  
1, 7, 124

Πραγματικές  
3.14, 4.56,  
1E-6

Πραγματικές  
Διπλής  
Ακρίβειας  
123D-30

**ΠΡΟΣΟΧΗ :** Στην γραφή των δεκαδικών αριθμών χρησιμοποιώ τελεία (.) και όχι (,) δηλ. ο αριθμός 12,36 γράφεται ως 12.36. Δεν χρησιμοποιώ τελεία για διαχωρισμό χιλιάδων, εκατομμυρίων κ.τ.λ. Τις δυνάμεις του 10 τις γράφω με την βοήθεια του E για τους αριθμούς απλής ακρίβειας ή του D για τους αριθμούς διπλής ακρίβειας δηλ. τον αριθμό 3200 θα τον έγγραφα 0.32E+4 αν ήθελα να τον παραστήσω ως αριθμό απλής ακρίβειας ή 0.32E+4 αν ήθελα να τον παραστήσω ως αριθμό διπλής ακρίβειας.

*Μεταβλητή* στην BASIC είναι μια θέση η οποία δημιουργείται στη μνήμη του Η/Υ για να δεχτεί έναν συγκεκριμένο τύπο δεδομένων, και η οποία έχει ένα χαρακτηριστικό όνομα που πληροί τους παρακάτω κανόνες :

ΕΠΙΤΡΕΠΤΑ ΟΝΟΜΑΤΑ	ΜΗ ΕΠΙΤΡΕΠΤΑ ΟΝΟΜΑΤΑ
<ol style="list-style-type: none"> <li>Ξεκινούν πάντα με αγγλικό γράμμα και στην συνέχεια ακολουθεί συνδυασμός γραμμάτων και αριθμών.</li> <li>Επιτρέπονται αγγλικά γράμματα και αριθμοί.</li> </ol>	<ol style="list-style-type: none"> <li>Δεν επιτρέπονται τα σύμβολα εκτός από τα \$, %, !, # που τοποθετούνται στο τέλος του ονόματος για να δηλώσουν τον τύπο της μεταβλητής.</li> <li>Δεν επιτρέπονται τα κενά.</li> </ol>

3. Επιτρέπεται η χρήση της τελείας.	3. Δεν επιτρέπονται τα Ελληνικά γράμματα.
4. Κεφαλαία ή πεζά θεωρούνται το ίδιο.	4. Δεν επιτρέπονται για ονόματα να έχουμε ονόματα εντολών ή συναρτήσεων της BASIC.
	5. Δύο πρώτοι χαρακτήρες διάφοροι του FN.

### Παράδειγμα.

Όνομα μεταβλητής	Σωστό/Λάθος
<b>FNΚ</b>	Λάθος γιατί ξεκινάει από FN.
<b>1ΑΒ</b>	Λάθος γιατί ξεκινάει από αριθμό
<b>Α\1</b>	Λάθος γιατί χρησιμοποιεί μη επιτρεπτά σύμβολα.
<b>Α\$</b>	Σωστό γιατί χρησιμοποιεί επιτρεπτό σύμβολο στο τέλος του ονόματος.
<b>CLS</b>	Λάθος γιατί είναι όνομα εντολής της BASIC.
<b>ΝΙΚΟΣ</b>	Λάθος γιατί περιέχει Ελληνικά γράμματα.
<b>N12</b>	Σωστό γιατί ξεκινάει από γράμμα και ακολουθεί συνδυασμός γραμμάτων ή/και αριθμών.
<b>A 12</b>	Λάθος γιατί περιέχει κενά.

Τα ονόματα που διαλέγουμε για τις μεταβλητές μας συνήθως έχουν σχέση με αυτό που παριστάνουν π.χ. TIME για τον χρόνο, S για το διάστημα κ.λ.π.

Οι μεταβλητές ανάλογα με τις τιμές που δέχονται (αριθμητικές ή αλφαριθμητικές) μπορούν να διαχωριστούν σε απλές μεταβλητές και σε μεταβλητές με δείκτη (που θα δούμε σε άλλο κεφάλαιο). Ο διαχωρισμός των απλών μεταβλητών φαίνεται στο παρακάτω σχήμα :

#### Μεταβλητές

Αλφαριθμητικές

A\$='ΝΙΚΟΣ'

Αριθμητικές

A=3

Ακέραιες

A%=1

Πραγματικές

B!=3.14

Πραγματικές

Διπλής

Ακρίβειας

C#=123D-30



Κάθε μεταβλητή μπορεί να πάρει μια και μόνο μία τιμή. Εάν η μεταβλητή πάρει κάποια τιμή στη συνέχεια μπορούμε να χρησιμοποιήσουμε το όνομα της σε πράξεις. Η τιμή της μεταβλητής αυτής αλλάζει στην περίπτωση που βρεθεί αριστερά του =. Το (=) στην BASIC σημαίνει αντικατάσταση και όχι ισότητα. Όταν ο H/Y διαβάσει μια εντολή που περιέχει το όνομα μιας καινούργιας μεταβλητής τότε αυτόματα δημιουργεί μια θέση στην μνήμη με το όνομα της μεταβλητής αυτής και αν αυτή η μεταβλητή βρίσκεται αριστερά του (=) τότε τοποθετεί την τιμή που βρίσκεται δεξιά του (=) στην μεταβλητή αυτή. Αντίθετα αν η μεταβλητή είναι δεξιά του (=) τότε : α) στην περίπτωση που η μεταβλητή είναι αριθμητική θεωρείται αυτόματα ότι έχει την τιμή 0 ενώ β) αν η μεταβλητή αυτή είναι αλφαριθμητική, τότε θεωρείται ότι περιέχει τον κενό χαρακτήρα.

### Παράδειγμα.

ΕΝΤΟΛΕΣ	ΜΝΗΜΗ		
	A	B	C\$
A=1	1		
C\$='ΝΙΚΟΣ'			ΝΙΚΟΣ
A=3	3		ΝΙΚΟΣ
B=A+1	3	4	ΝΙΚΟΣ
A=A+2	5	4	ΝΙΚΟΣ
B=B+A	5	9	ΝΙΚΟΣ

Το είδος των μεταβλητών μπορούμε να το ορίσουμε και με τις δηλωτικές εντολές :

Δηλωτική Εντολή	Είδος Μεταβλητής
DEFINT	Ακέραιες μεταβλητές.
DEFSNG	Πραγματικές μεταβλητές απλής ακρίβειας.
DEFDBL	Πραγματικές μεταβλητές διπλής ακρίβειας.
DEFSTR	Αλφαριθμητικές μεταβλητές.

Παράδειγμα.

DEFINT A

Όλες οι μεταβλητές που το όνομα τους ξεκινάει από A θα είναι ακέραιες.

DEFDBL B,C

Όλες οι μεταβλητές που το όνομα τους ξεκινάει από B ή C θα είναι πραγματικές διπλής ακριβείας.

DEFSTR A-C

Όλες οι μεταβλητές που το όνομα τους ξεκινάει από A-C θα είναι αλφαριθμητικές.

### Εκτέλεση αριθμητικών πράξεων στην BASIC

Ονομάζουμε τελεστή ένα απλό σύμβολο ή μια λέξη που δείχνει ποια πράξη πρέπει να γίνει με μια ή δύο τιμές. Στην BASIC έχουμε 4 διαφορετικούς τελεστές : αριθμητικούς, σχέσεων, λογικούς και αλφαριθμητικούς. Στο κεφάλαιο αυτό θα αναφέρουμε μόνο την πρώτη κατηγορία τελεστών.

Οι αριθμητικοί τελεστές είναι τα γνωστά σύμβολα των αριθμητικών πράξεων :

+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμό
/	Διαίρεση
\	Ακέραια διαίρεση
^	Ύψωση σε δύναμη
MOD	Υπόλοιπο διαίρεσης μεταξύ δύο ακεραίων

Η σειρά εκτέλεσης των πράξεων μια σύνθετη παράσταση ακολουθεί την παρακάτω σειρά προτεραιότητας :

- A) Δυνάμεις.
- B) Αλλαγές πρόσημου.
- Γ) Πολλαπλασιασμοί και διαιρέσεις.
- Δ) Ακέραιες διαιρέσεις.
- Ε) Προσθέσεις και αφαιρέσεις.

Στην περίπτωση που έχουμε παραπάνω από μία πράξη της ίδιας προτεραιότητας τότε προηγούνται οι πράξεις από τα αριστερά, εκτός της περιπτώσεως των δυνάμεων π.χ.  $3*4*5=12*5=60$ ,  $2^3^2=2^9=512$ .

Στην περίπτωση που έχουμε παραπάνω από μια παρενθέσεις τότε προηγούνται οι πράξεις στις πιο εσωτερικές παρενθέσεις.

Οι παραπάνω κανόνες μας βοηθούν στο να μεταφέρουμε γνωστούς μαθηματικούς τύπους υπό την μορφή εντολών στην BASIC αλλά και αντίστροφα, δηλαδή τύπους που έχουν γραφεί στην BASIC να δούμε σε ποιους μαθηματικούς τύπους αντιστοιχούν.

### Παράδειγμα.

Μαθηματικός Τύπος	Τύπος στην BASIC
$x = A \times T \times \frac{(1+T)^N}{(1+T)^N - 1}$	$X=A*T*((1+T)^N/((1+T)^N-1))$
$y = \frac{5}{a-b}$	$Y=5/(A-B)$
$S = \frac{1}{2} \times G \times T^2$	$S=1/2*G*T^2$
$E = \sqrt{S \times (S-A) \times (S-B) \times (S-C)}$	$E=(S*(S-A)*(S-B)*(S-C))^(1/2)$
$D = E \times (1+P)^N$	$D=E*(1+P)^N$
$K = K_0 \times (1+R)^N \times \left(1 + \frac{R \times H}{360}\right)$	$K=K0*(1+R)^N*(1+R*H/360)$
$S = \sqrt{\frac{2(P_1 - P_2)}{P \left(\frac{S_1^2}{S_2^2} - 1\right)}}$	$S=((2*(P1-P2))/(P*(S1^2/S2^2-1)))^(1/2)$
$V = \frac{P \times D^2 \times H}{4}$	$V=P*D^2*H/4$
Το υπόλοιπο της διαίρεσης του 32 με το 5 (που είναι $2 : 5*6+2$ )	$32 \text{ MOD } 5$
Το πηλίκο της ακέραιας διαίρεσης του 15 με το 7 (που είναι $2 : 2*7+1$ )	$15 \setminus 7$

Όταν δεν είμαστε σίγουροι για την προτεραιότητα των πράξεων χρησιμοποιούμε παρενθέσεις.

### Λάθη από μη επιτρεπτές πράξεις

1. Προσέχω η τιμή του διαιρέτη να είναι διάφορη του μηδενός αλλιώς θα έχω το μήνυμα Division by Zero.
2. Προσέχω να έχω το σωστό όρισμα πρδ. το υπόριζο να είναι θετικό.
3. Προσέχω τα όρια των αριθμών που ανέφερα προηγουμένως. Όταν οι αριθμοί που χρησιμοποιώ είναι μεγαλύτεροι από τα όρια του Η/Υ έχω το μήνυμα Overflow ή αντίθετα είναι μικρότεροι από τα όρια το μήνυμα Underflow.
4. Επειδή η παράσταση των αριθμών στον Η/Υ γίνεται στο δυαδικό σύστημα για τον λόγο αυτό αριθμοί όπως ο 0.1 που δεν έχουν ακριβή παράσταση στο δυαδικό σύστημα μπορούν να προκαλέσουν λάθη.
5. Όταν υψώνω έναν αριθμό σε μια δύναμη, θα πρέπει να μην είναι ο αριθμός αρνητικός και η δύναμη πραγματικός αριθμός.
6. Συνήθως στις προσθέσεις πολύ μικρών και πολύ μεγάλων αριθμών μπορούν να προκύψουν λάθη λόγω ακρίβειας του Η/Υ.

Στις αριθμητικές πράξεις μπορώ να χρησιμοποιώ και συναρτήσεις. Συνάρτηση είναι ένα πρόγραμμα που κάνει μια συγκεκριμένη δουλειά, και έχει ένα συγκεκριμένο όνομα με το οποίο καλείται. Στον παρακάτω πίνακα βλέπουμε μερικές από τις συναρτήσεις της BASIC.

Όνομα συνάρτησης	Επεξήγηση-Παράδειγμα
ABS(X)	Απόλυτη τιμή του X. ABS(-15) : Απόλυτη τιμή του -15.
SQR(X)	Τετραγωνική ρίζα του X. SQR(4) : Η τετραγωνική ρίζα του 4.
LOG(X)	Νεπέρειος ή φυσικός λογάριθμος του X. LOG(5) : Φυσικός λογάριθμος του 5.
EXP(X)	Ύψωση του e στον αριθμό X. EXP(4) : Μας δίνει τον αριθμό e <sup>4</sup> .
SIN(X)	Ημίτονο του X όπου το X δίνεται σε ακτίνια. Εάν έχω M μοίρες τότε αυτές είναι 3.14*M/180 ακτίνια. SIN(3.14) : Μας δίνει το ημίτονο του π που

	είναι 0.
COS(X)	Συνημίτονο του X όπου το X δίνεται σε ακτίνια. COS(3.14) : Μας δίνει το συνημίτονο του π που είναι -1.
TAN(X)	Εφαπτομένη του X όπου το X δίνεται σε ακτίνια. TAN(3.14) : Μας δίνει την εφαπτομένη του π που είναι 0.

## Εκτέλεση αλφαριθμητικών πράξεων στην BASIC

Επιτρέπεται η πρόσθεση αλφαριθμητικών σταθερών ή/και μεταβλητών :

**‘ΚΩΣΤΑΣ’+‘ΜΑΡΙΑ’ το οποίο μας δίνει ‘ΚΩΣΤΑΣΜΑΡΙΑ’**

Δεν επιτρέπεται :

1. Ανάμειξη αριθμών και χαρακτήρων π.χ. 5 + ‘ΝΙΚΟΣ’.
2. Να τοποθετήσω αριθμό σε αλφαριθμητική μεταβλητή και αντίστροφα π.χ. A=‘ΝΙΚΟΣ’ ή A\$=3.
3. Πράξεις άλλες εκτός της πρόσθεσης.

Υπάρχει και ένα πλήθος συναρτήσεων αλφαριθμητικών μεταβλητών όπως οι παρακάτω :

Όνομα Μεταβλητής	Επεξήγηση-Παράδειγμα
LEN(A\$)	Πλήθος χαρακτήρων. Αν A\$=‘ΝΙΚΟΣ’ τότε LEN(A\$)=5
LEFT(A\$,m)	Μας δίνει τους αριστερά m χαρακτήρες της μεταβλητής A\$. Αν A\$=‘ΚΕΙΜΕΝΟ’ τότε LEFT(A\$,2)=‘ΚΕ’
RIGHT(A\$,m)	Μας δίνει τους δεξιά m χαρακτήρες της μεταβλητής A\$. Αν A\$=‘ΚΕΙΜΕΝΟ’ τότε RIGHT(A\$,3)=‘ΕΝΟ’
MID(A\$,k,m)	Μας δίνει τους χαρακτήρες k,k+1,...,k+m-1 της μεταβλητής A\$. Αν A\$=‘ΚΕΙΜΕΝΟ’ τότε MID(A\$,2,3)=‘ΕΙΜ’

## Εντολές εισόδου – αντικατάστασης - εξόδου στην BASIC

Το παρακάτω μοντέλο χαρακτηρίζει τα περισσότερα προβλήματα που θα ασχοληθούμε :



Στο πρώτο στάδιο λοιπόν κάθε προγράμματος συνήθως δίνουμε κάποιες εντολές εισόδου με τις οποίες δίνουμε τα δεδομένα του προβλήματος μας, στην συνέχεια μέσω κάποιων εντολών αντικατάστασης ή ελέγχου ή διακλαδώσεων επεξεργαζόμαστε τα δεδομένα αυτά και στην συνέχεια τα αποτελέσματα που βρήκαμε τα εμφανίζουμε στην οθόνη ή στον εκτυπωτή μέσω κάποιων εντολών εξόδου. Στην συνέχεια θα δώσουμε κάποιες εντολές εισόδου-εξόδου και όσον αφορά το στάδιο επεξεργασίας θα παρουσιάσουμε εντολές αντικατάστασης.

### Η εντολή αντικατάστασης LET

Είναι η εντολή LET με την οποία δίνουμε στις μεταβλητές του προγράμματος μια συγκεκριμένη αριθμητική, αλφαριθμητική ή και λογική τιμή.

**Σύνταξη :** xx [LET] μεταβλητή=σταθερά ή μεταβλητή ή παράσταση

**Σχόλια :** Δημιουργεί μια θέση στην μνήμη για την μεταβλητή που βρίσκεται αριστερά του ίσον εάν δεν υπάρχει και τοποθετεί το αποτέλεσμα που βρίσκεται δεξιά του ίσον. Ουσιαστικά το (=) σημαίνει αντικατάσταση.

#### Παράδειγμα.

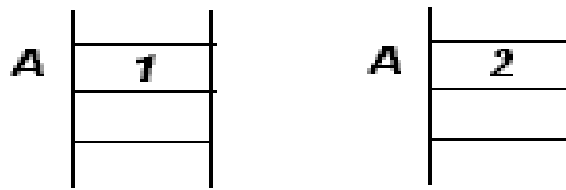
10 LET A=2

Δημιουργεί μια θέση στη μνήμη του Η/Υ για την μεταβλητή A και τοποθετεί μέσα την τιμή 2.

10 LET A=1

20 LET A=A+1

Η πρώτη εντολή δημιουργεί μια θέση στην μνήμη του Η/Υ για το A και τοποθετεί την τιμή 1.



Η δεύτερη εντολή βρίσκει το αποτέλεσμα δεξιά του (=) που είναι 1+1 (εφόσον η τιμή του A από την πρώτη εντολή ήταν 1) και το τοποθετεί στην ίδια θέση της μνήμης που είχε ανοίξει στο πρώτη εντολή για να στεγάσει την τιμή του A.

### Εντολές εισόδου

Οι εντολές της BASIC με τη βοήθεια των οποίων επιτυγχάνουμε να μεταφέρουμε τις τιμές των δεδομένων του προβλήματος στην μνήμη του Η/Υ ονομάζονται *εντολές εισόδου*.

Ο πρώτος και πιο απλός τρόπος να εισάγουμε δεδομένα στον Η/Υ είναι μέσω της εντολής **LET** που είδαμε προηγουμένως. Ένα σοβαρό μειονέκτημα είναι ότι πρέπει να δίνουμε η εντολές για την τοποθέτηση η δεδομένων και ότι η διόρθωση όλων των δεδομένων γίνεται μέσα από το πρόγραμμα και επηρεάζει όλες αυτές τις εντολές. Ένας πιο εύκολος τρόπος εισαγωγής δεδομένων στην περίπτωση που θέλουμε ο χρήστης να δίνει τα δεδομένα από το πληκτρολόγιο και όχι μέσα από το πρόγραμμα είναι η εντολή **INPUT**.

### A) Η εντολή INPUT.

**Σκοπός :** Η καταχώρηση τιμών από τον χρήστη, στις μεταβλητές της INPUT.

**Σύνταξη :** xx INPUT ['κείμενο',] λίστα μεταβλητών

Όπου λίστα μεταβλητών είναι ένα πλήθος μεταβλητών που διαχωρίζονται με κόμμα, ενώ 'κείμενο' είναι ένα προειδοποιητικό μήνυμα που εμφανίζεται στην οθόνη και βρίσκεται πάντα ανάμεσα σε εισαγωγικά (το 'κείμενο' δεν είναι υποχρεωτικό).

**Σχόλια :** Κατά την εκτέλεση της εντολής ο Η/Υ εμφανίζει το μήνυμα 'κείμενο' στην οθόνη και περιμένει να του δώσουμε τόσες τιμές όσες και οι μεταβλητές της εντολής. Εφόσον δώσουμε τις τιμές και πατήσουμε (ENTER) στην συνέχεια τις τοποθετεί στην μνήμη του Η/Υ κάνοντας την κατάλληλη αντιστοίχιση. Οι τιμές δίνονται είτε χωρισμένες με κόμμα είτε αφήνοντας κενό μεταξύ τους. Στο 'κείμενο' συνήθως τοποθετούμε μια υπενθύμιση για τον χρήστη όπως τα ονόματα των μεταβλητών, τι τύπου δεδομένα δέχονται και με ποια σειρά να μας δώσει τις τιμές. Εάν δώσουμε λιγότερα δεδομένα από ότι οι μεταβλητές εμφανίζεται το μήνυμα Redo from start. Θα πρέπει στην περίπτωση αυτή να ξαναδώσουμε τα δεδομένα μας. Θα πρέπει να υπάρχει συμφωνία (ένα προς ένα αντιστοίχιση) μεταξύ των μεταβλητών της INPUT και των τιμών που δίνει ο χρήστης.

#### **Παράδειγμα.**

10 INPUT "Δώσε μια τιμή για τον χρόνο : ";A

Ο Η/Υ εμφανίζει το παρακάτω μήνυμα στην οθόνη :

Δώσε μια τιμή για τον χρόνο :?

Και περιμένει να απαντήσουμε γράφοντας μια τιμή την οποία θέλουμε να πάρει το A και στην συνέχεια να πατήσουμε ENTER.

10 INPUT A

Ο Η/Υ εμφανίζει το παρακάτω μήνυμα στην οθόνη :

?

και περιμένει να απαντήσουμε γράφοντας μια τιμή την οποία θέλουμε να πάρει το A και στην συνέχεια να πατήσουμε ENTER. Βλέπουμε ότι στην δεύτερη περίπτωση δεν γνωρίζουμε αν πρέπει να δώσουμε μια ή πολλές τιμές καθώς και η τιμή που θα δώσουμε σε ποια μεταβλητή θα καταχωρηθεί. Επίσης δεν ξέρουμε αν πρέπει να δώσουμε αριθμητική ή αλφαριθμητική τιμή καθώς και σε ποια όρια θα κυμαίνεται αυτή η τιμή αν πρόκειται για αριθμητική τιμή. Είναι καλό λοιπόν να ακολουθούν την εντολή INPUT πάντα κάποιες οδηγίες στο 'κείμενο'.



## **B) Οι εντολές READ-DATA-RESTORE.**

### **Η εντολή READ.**

**Σύνταξη :** xx READ [μεταβλητές]

**Σχόλια :** Χρησιμοποιείται για να δώσουμε τιμές σε μεταβλητές που θα χρησιμοποιηθούν από το πρόγραμμα. Οι τιμές αυτές δεν δίνονται από τον χρήστη μέσω του πληκτρολογίου (και συνεπώς μπορούν να αλλάζουν σε κάθε εκτέλεση του προγράμματος) όπως με την εντολή INPUT αλλά είναι γνωστές από πριν και δίδονται μέσα στο πρόγραμμα από την εντολή DATA. Ως παράδειγμα θα μπορούσαμε να αναφέρουμε τα αποτελέσματα ενός πειράματος τα οποία τα γνωρίζουμε εκ των προτέρων και τα οποία είναι πολλά σε πλήθος. Η επιλογή της εντολής INPUT ως εντολή εισόδου θα είχε το εξής μειονέκτημα : μια εσφαλμένη πληκτρολόγηση σε ένα από τα δεδομένα θα απαιτούσε την επαναπληκτρολόγηση όλων των δεδομένων από την αρχή. Καμιά εντολή READ δεν μπορεί να χρησιμοποιηθεί αν δεν συνοδεύεται από μια ή περισσότερες DATA.

#### **Παράδειγμα.**

```
10 READ A, B$, N
```

(Θα δούμε στην συνέχεια πως συνδέονται οι READ-DATA).

### **Η εντολή DATA.**

**Σύνταξη :** xx DATA [σταθερές]

**Σχόλια :** Δίνει τιμές σε μεταβλητές που έχουν δοθεί στην READ. Οι τιμές αυτές πρέπει να δίνονται σε μια γραμμή και να διαχωρίζονται με κόμμα. Η DATA μπορεί να βρίσκεται σε κάθε σημείο του προγράμματος χωρίς να επηρεάζει την ροή του εφόσον πρόκειται για μη εκτελέσιμη εντολή. Κυρίως τοποθετείται ή στην αρχή ή στο τέλος του προγράμματος.

#### **Παράδειγμα.**

```
100 DATA 10, 'NIKOS', 30
```

(Θα δούμε στην συνέχεια πως συνδέονται οι READ-DATA).

### **Συνδυασμός READ-DATA.**

Ο Η/Υ μπορούμε θεωρητικά να φανταστούμε ότι προτού ξεκινήσει η εκτέλεση του προγράμματος έχει τοποθετημένο έναν δείκτη στην πρώτη

σταθερά της πρώτης DATA που υπάρχει στο πρόγραμμα. Όταν λοιπόν συναντήσει κατά την διάρκεια εκτέλεσης του προγράμματος την πρώτη READ, αντιστοιχεί την σταθερά αυτή στην οποία βρίσκεται ο δείκτης στην πρώτη μεταβλητή της READ. Εφόσον πάρει τιμή η πρώτη μεταβλητή της READ ο δείκτης αυτός τοποθετείται στην επόμενη σταθερά της DATA. Στην συνέχεια η δεύτερη μεταβλητή της READ δέχεται την σταθερά αυτή και η διαδικασία συνεχίζεται. Στην περίπτωση που οι σταθερές της συγκεκριμένης DATA έχουν τελειώσει τότε ο δείκτης τοποθετείται στην πρώτη σταθερά της επόμενης DATA που ακολουθεί στο πρόγραμμα. Το ίδιο συμβαίνει και με την εντολή READ δηλ. εάν πήραν τιμές όλες οι μεταβλητές της πρώτης READ τότε η επόμενη READ θα συνεχίζει να παίρνει τιμές από το σημείο στο οποίο έχει σταματήσει ο δείκτης. Στην περίπτωση που οι μεταβλητές της READ είναι περισσότερες από τις σταθερές που δίνονται στις DATA του προγράμματος το πρόγραμμα θα σταματήσει να εκτελείται και το ακόλουθο μήνυμα θα εμφανισθεί :

**READ OUT OF DATA**

Θα πρέπει να προσέξουμε ιδιαίτερα να υπάρχει αντιστοίχιση μεταξύ αριθμητικών μεταβλητών και αριθμών καθώς και αλφαριθμητικών μεταβλητών με αλφαριθμητικές σταθερές. Ο καλός προγραμματισμός απαιτεί κάθε εντολή DATA να συνοδεύει μόνο μια εντολή READ και ο αριθμός των μεταβλητών σε μια READ να είναι όσο το δυνατό πιο μικρός.

#### **Παράδειγμα. (Πλήρης αντιστοίχιση δεδομένων-μεταβλητών)**

```
10 READ A, B, C$
```

```
20 DATA 5,7,'NIKOS'
```

Ο νοητός δείκτης που αναφέραμε παραπάνω βρίσκεται στο 5, συνεπώς όταν ο Η/Υ εκτελέσει την πρώτη εντολή, το 5 θα τοποθετηθεί στην μεταβλητή A, ενώ εκείνη τη στιγμή ο δείκτης θα τοποθετηθεί στο 7. Στην συνέχεια η επόμενη μεταβλητή B θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ.7, ενώ ο δείκτης θα τοποθετηθεί στην επόμενη τιμή 'NIKOS'. Τέλος η τελευταία μεταβλητή C\$ θα πάρει την τιμή 'NIKOS' στην οποία βρίσκεται ο δείκτης.

#### **Παράδειγμα. (Περισσότερες εντολές READ από ότι DATA)**

```
10 READ A, B, C
```

20 READ D, E

30 DATA 10,20,30,40,50

Ο νοητός δείκτης βρίσκεται στο 10, συνεπώς όταν ο Η/Υ εκτελέσει την πρώτη εντολή, το 10 θα τοποθετηθεί στην μεταβλητή A, ενώ εκείνη τη στιγμή ο δείκτης θα τοποθετηθεί στο 20. Στην συνέχεια η επόμενη μεταβλητή B θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 20, ενώ ο δείκτης θα τοποθετηθεί στην επόμενη τιμή 30. Η επόμενη μεταβλητή C θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 30, ενώ ο δείκτης θα τοποθετηθεί στην επόμενη τιμή 40. Η επόμενη μεταβλητή D θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 40, ενώ ο δείκτης θα τοποθετηθεί στην επόμενη τιμή 50. Τέλος η τελευταία μεταβλητή E θα πάρει την τιμή 50 στην οποία βρίσκεται ο δείκτης.

**Παράδειγμα. (Περισσότερες εντολές DATA από ότι READ)**

10 READ A, B, C

20 DATA 10,20

30 DATA 30

Ο νοητός δείκτης βρίσκεται στο 10, συνεπώς όταν ο Η/Υ εκτελέσει την πρώτη εντολή, το 10 θα τοποθετηθεί στην μεταβλητή A, ενώ εκείνη τη στιγμή ο δείκτης θα τοποθετηθεί στο 20. Στην συνέχεια η επόμενη μεταβλητή B θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 20, ενώ ο δείκτης θα τοποθετηθεί στην επόμενη τιμή 30. Τέλος η επόμενη μεταβλητή C θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 30.

**Παράδειγμα. (Περισσότερες μεταβλητές από δεδομένα)**

10 READ A, B

20 DATA 10

Ο νοητός δείκτης βρίσκεται στο 10, συνεπώς όταν ο Η/Υ εκτελέσει την πρώτη εντολή, το 10 θα τοποθετηθεί στην μεταβλητή A. Στην συνέχεια η επόμενη μεταβλητή B δεν θα μπορέσει να πάρει κάποια τιμή λόγω του ότι ο δείκτης δεν μπορεί να τοποθετηθεί σε κάποια τιμή. Συνεπώς η εκτέλεση του προγράμματος θα σταματήσει και θα εμφανισθεί το μήνυμα :

READ OUT OF DATA

**Παράδειγμα. (Περισσότερα δεδομένα από μεταβλητές)**

```
10 READ A, B
```

```
20 DATA 10,20,30
```

Ο νοητός δείκτης βρίσκεται στο 10, συνεπώς όταν ο Η/Υ εκτελέσει την πρώτη εντολή, το 10 θα τοποθετηθεί στην μεταβλητή A, ενώ εκείνη τη στιγμή ο δείκτης θα τοποθετηθεί στο 20. Στην συνέχεια η επόμενη μεταβλητή B θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 20, ενώ ο δείκτης θα τοποθετηθεί στην επόμενη τιμή 30. Δεν υπάρχει πρόβλημα λοιπόν με την ύπαρξη περισσότερων τιμών στην DATA σε σχέση με τον αριθμό των μεταβλητών.

**Η εντολή RESTORE.**

**Σύνταξη :** xx RESTORE [αριθμός γραμμής]

**Σχόλια :** Χρησιμοποιείται για την επανατοποθέτηση των ιδίων δεδομένων που έχουν αναφερθεί στην εντολή DATA. Πιο συγκεκριμένα επαναφέρει τον νοητό δείκτη που είχαμε αναφέρει παραπάνω στην πρώτη σταθερά της DATA που βρίσκεται στον [αριθμό γραμμής]. Όταν η RESTORE δεν ακολουθείται από έναν συγκεκριμένο αριθμό γραμμής, τότε ο δείκτης επαναφέρεται στην πρώτη σταθερά της πρώτης κατά σειρά αρίθμησης DATA του προγράμματος.

**Παράδειγμα. (RESTORE χωρίς αριθμό γραμμής)**

```
10 READ A,B
```

```
20 RESTORE
```

```
30 READ C,D
```

```
40 DATA 10,20
```

Ο νοητός δείκτης βρίσκεται στο 10, συνεπώς όταν ο Η/Υ εκτελέσει την πρώτη εντολή, το 10 θα τοποθετηθεί στην μεταβλητή A, ενώ εκείνη τη στιγμή ο δείκτης θα τοποθετηθεί στο 20. Στην συνέχεια η επόμενη μεταβλητή B θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 20. Η εντολή RESTORE θα επαναφέρει τον δείκτη στην πρώτη τιμή της πρώτης DATA και συνεπώς στην τιμή 10. Η επόμενη μεταβλητή λοιπόν C θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ.10, ενώ ο δείκτης θα τοποθετηθεί στην τιμή 20. Τέλος η επόμενη μεταβλητή D θα πάρει την τιμή στην οποία βρίσκεται ο δείκτης δηλ. 20.

**Παράδειγμα. (RESTORE με αριθμό γραμμής)**

```
10 RESTORE 40
20 READ C,D
30 DATA 10,20
40 DATA 30,40
```

Ο νοητός δείκτης τοποθετείται στην πρώτη σταθερά της DATA που δίνεται από τον αριθμό της γραμμής που υπάρχει στην RESTORE δηλ. στον αριθμό 30 και όχι στον 10 που αναμενόταν. Συνεπώς σύμφωνα με τα παραπάνω οι μεταβλητές C και D θα πάρουν τις τιμές 30 και 40 αντίστοιχα.

**Εντολές εξόδου**

Με τις εντολές αυτές μπορούμε να δούμε τα αποτελέσματα της επεξεργασίας των δεδομένων μας είτε πάνω στην οθόνη είτε πάνω στον εκτυπωτή.

**A) Η εντολή PRINT.**

**Σύνταξη :** xx PRINT *λίστα* από [σταθερές ή/και μεταβλητές ή/και παραστάσεις]

όπου *λίστα* = {πλήθος στοιχείων διαχωριζόμενων από (,) ή (;)}

**Σχόλια :** Η εντολή PRINT χρησιμοποιείται για την εκτύπωση των παραπάνω όρων της εντολής. Οι όροι της εντολής διαχωρίζονται με κόμμα ή ερωτηματικό. Όταν χωρίζονται με κόμμα τυπώνονται ανά 14 θέσεις της οθόνης διαφορετικά (με ;) τυπώνονται η μια τιμή μετά την άλλη με ένα κενό ενδιάμεσα. Όταν ο τελευταίος όρος τελειώνει με (;) ή (,) τότε η επόμενη PRINT θα συνεχίσει να τυπώνει τις τιμές στην ίδια γραμμή. Η εντολή LPRINT έχει την ίδια σύνταξη με την μόνη διαφορά ότι εκτυπώνει τα αποτελέσματα όχι στην οθόνη αλλά στον εκτυπωτή.

**Παράδειγμα.**

```
10 LET A=2
20 LET B=3
30 PRINT 5,A,B,A+B
```

Το αποτέλεσμα της εκτύπωσης θα είναι

5            2            3            5

Οι όροι δηλαδή της PRINT εκτυπώνονται σε διαστήματα 14 θέσεων στην οθόνη. Εάν αντί της παραπάνω PRINT είχα την εξής :

```
30 PRINT 5,A,B,A+B
```

το αποτέλεσμα θα ήταν

```
5 2 3 5
```

λόγω του ότι αφήνει μόνο ένα κενό μεταξύ των αριθμών.

### Παράδειγμα

```
10 LET A=2 : LET B=3
```

```
20 PRINT 5,A,B,
```

```
30 PRINT "A+B=",A+B
```

Το αποτέλεσμα της εκτύπωσης θα είναι :

```
5            2            3            A+B=    5
```

Το (,) στο τέλος της εντολής 20 δίνει εντολή στον Η/Υ να συνεχίσει η εκτύπωση της επόμενης PRINT στην ίδια γραμμή.

## B) Η εντολή PRINT USING.

**Σύνταξη :** xx PRINT USING [προδιαγραφή] [σταθερές, μεταβλητές, εκφράσεις]

**Σχόλια :** Χρησιμοποιείται για τον καθορισμό του τρόπου με τον οποίο θα εμφανίζονται οι τιμές πάνω στην οθόνη. Στην [προδιαγραφή] μπορώ να έχω πολλούς και διάφορους χαρακτήρες. Εδώ θα αναφέρουμε μόνο τον χαρακτήρα '#' ο οποίος σημαίνει πως πρέπει να κρατηθεί μια θέση για την εμφάνιση ενός μονοψηφίου αριθμού στην οθόνη. Όταν έχουμε περισσότερους του ενός τέτοιους χαρακτήρες τότε θεωρούμε ότι έχουμε ένα αριθμητικό πεδίο και μπορούν να εμφανισθούν αριθμοί με περισσότερα ψηφία. Αν το πλήθος των ψηφίων του αριθμού είναι μικρότερος από το πλήθος των χαρακτήρων '#' τότε ο αριθμός εμφανίζεται στα δεξιά ενώ οι υπόλοιπες θέσεις μένουν κενές.

### Παράδειγμα.

```
10 PRINT USING '#####' 30
```

Εμφανίζεται το ακόλουθο μήνυμα

bbb30

όπου (b) συμβολίζει το κενό.

### Παράδειγμα.

```
10 PRINT USING '###.##' 30
```

Εμφανίζεται το ακόλουθο μήνυμα

```
b30.00
```

## Άλλες εντολές

### Η εντολή CLS

Αναφέρθηκε στην προηγούμενη ενότητα. Χρησιμοποιείται για τον καθαρισμό της οθόνης. Κυρίως χρησιμοποιείται στην αρχή των προγραμμάτων.

### Η εντολή STOP

Σύνταξη : xx STOP

**Σχόλια** : Μπορεί να βρεθεί σε περισσότερα από ένα σημεία του προγράμματος μας. Δηλώνει την αυτόματη διακοπή του προγράμματος. Χρησιμοποιείται κυρίως σε συνδιασμό με άλλες εντολές όπως η IF-THEN-ELSE. Κατά την εκτέλεση της εμφανίζεται ένα μήνυμα στην οθόνη Break in line xx.

### Παράδειγμα

```
10 INPUT A
```

```
20 B=A^2
```

```
30 PRINT A
```

```
40 STOP
```

```
50....
```

```
60 .....
```

### Η εντολή END

Σύνταξη : xx END

**Σχόλια** : Δηλώνει στον Η/Υ που τελειώνει το πρόγραμμα μας.

## Σχόλια μέσω της REM

**Σύνταξη :** xx REM κείμενο

**Σχόλια :** Όταν ο Η/Υ διαβάσει τη λέξη REM αγνοεί ότι ακολουθεί μετά και συνεχίζει με την εκτέλεση της επόμενης εντολής. Χρησιμοποιείται για την δημιουργία σχολίων συνήθως στην αρχή του προγράμματος. Τα σχόλια αυτά μπορεί να αναφέρουν την φύση του προβλήματος καθώς και το τι παριστά η κάθε μεταβλητή. Αντίστοιχα σχόλια μπορούν να υπάρχουν σε όλο το πρόγραμμα αναφέροντας κάθε φορά ποιο βήμα του αλγορίθμου εκτελείται. Η χρήση σχολίων βοηθάει τον προγραμματιστή να θυμηθεί την επίλυση ενός προβλήματος που είχε δημιουργήσει ο ίδιος πριν πολύ καιρό και το οποίο θέλει διόρθωση καθώς και ένα άλλο πρόσωπο που θα κληθεί να λύσει τα τυχόν λάθη του προγράμματος μας.

### Παράδειγμα

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΕΜΒΑΔΟΥ ΚΥΚΛΟΥ
20 REM R=ΑΚΤΙΝΑ ΠΙ=3.14 E=ΕΜΒΑΔΟ
30 INPUT 'ΑΚΤΙΝΑ=',R
40 ΠΙ=3.14
50 REM ΥΠΟΛΟΓΙΣΜΟΣ ΕΜΒΑΔΟΥ ΜΕΣΩ ΤΥΠΟΥ E=π*r^2
50 E=ΠΙ*R^2
60 PRINT E
70 END
```

### Παραδείγματα απλών προγραμμάτων

**Παράδειγμα 1.** Θεωρούμε ότι ο κορμός ενός δέντρου είναι κυλινδρικός. Συνεπώς ο όγκος του δέντρου V αν γνωρίζουμε την περίμετρο του P και το ύψος του H που είναι εύκολα μετρήσιμα θα δίνεται από τον τύπο :

$$V = \frac{P^2 \times H}{4 \times \pi} \quad \text{όπου } \pi = 3.14$$

Θέλουμε λοιπόν να γράψουμε ένα πρόγραμμα που να υπολογίζει τον όγκο αυτό αν γνωρίζουμε την περίμετρο και τα ύψος του δέντρου.

Το πρώτο βήμα για την επίλυση του παραπάνω προβλήματος είναι να διαχωρίσουμε ποια θα είναι η είσοδος και ποια η έξοδος του προβλήματος.



**A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Η περίμετρος P και το ύψος H του δέντρου.

*Έξοδος* – Ο όγκος του δέντρου V.

Στην συνέχεια πρέπει να λύσουμε ένα παράδειγμα με το χέρι, για να δούμε ποια είναι η διαδικασία επίλυσης.

**B) Παράδειγμα με το χέρι.**

Έστω H=3.14m και P=0.5m τότε εφαρμόζω τον τύπο και έχω

$$V = \frac{0.5^2 \times 3.14}{4 \times 3.14} = \frac{0.25}{4} = 0.0625m^3$$

Στην συνέχεια πρέπει να γράψουμε τον αλγόριθμο του προβλήματος δηλ. τα βήματα που θα κάνουμε για την επίλυση του γενικού προβλήματος που θα ισχύει για οποιαδήποτε περίμετρο και ύψος.

**Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζουμε τις τιμές του ύψους και της περιμέτρου.

*Βήμα 2.* Υπολογίζουμε τον όγκο.

*Βήμα 3.* Εκτυπώνουμε τα αποτελέσματα.

Τέλος ακολουθεί η υλοποίηση του παραπάνω αλγορίθμου στην γλώσσα BASIC.

**Δ) Πρόγραμμα**

```

10 REM ΠΡΟΓΡΑΜΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΟΓΚΟΥ ΔΕΝΤΡΟΥ
20 REM H=ΥΨΟΣ, P=ΠΕΡΙΜΕΤΡΟΣ, V=ΟΓΚΟΣ
30 REM ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΥΨΟΥΣ ΚΑΙ ΠΕΡΙΜΕΤΡΟΥ
40 INPUT 'ΠΕΡΙΜΕΤΡΟΣ=',P
50 INPUT 'ΥΨΟΣ=',H
60 REM ΒΗΜΑ 2. ΥΠΟΛΟΓΙΣΜΟΣ ΟΓΚΟΥ
70 V=P^2*H/(4*3.14)
80 REM ΒΗΜΑ 3. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

```

```
90 PRINT "ΠΕΡΙΜΕΤΡΟΣ=",P
100 PRINT "ΥΨΟΣ=",H
110 PRINT "ΟΓΚΟΣ=",V
120 END
```

**Παράδειγμα 2.** Έστω ότι έχουμε μια δασική έκταση εμβαδού  $A$ . Έστω επίσης ότι ο ρυθμός αναδάσωσης είναι  $E$ . Τότε το ποσό της έκτασης αυτής μετά από  $N$  χρόνια θα είναι  $T$  και θα δίνεται από τον τύπο :

$$T = A \times (1 + E)^N$$

Να γραφεί πρόγραμμα που να υπολογίζει την συνολική έκταση της δασικής περιοχής μετά από  $N$  χρόνια αν είναι γνωστός ο ρυθμός της αναδάσωσης και η αρχική δασική έκταση.

#### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Η αρχική έκταση  $A$ , ο ρυθμός αναδάσωσης  $E$  και τα χρόνια  $N$ .

*Έξοδος* – Η συνολική έκταση μετά από  $N$  χρόνια.

#### **B) Παράδειγμα με το χέρι.**

Έστω  $N=1$ χρόνος,  $E=0.02$  και  $A=14.000.000\text{m}^2$  τότε εφαρμόζω τον τύπο και έχω

$$T = 14.000.000 \times (1 + 0.02)^{10} = 17065921,87993\text{m}^2$$

#### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζουμε τις τιμές της αρχικής δασικής έκτασης, του ρυθμού αναδάσωσης και των χρόνων  $N$  μετά από τους οποίους ενδιαφερόμαστε να βρούμε την συνολική έκταση.

*Βήμα 2.* Υπολογίζουμε την τελική έκταση.

*Βήμα 3.* Εκτυπώνουμε τα αποτελέσματα.

#### **Δ) Πρόγραμμα**

```
10 REM ΠΡΟΓΡΑΜΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΑΝΑΔΑΣΩΤΕΑΣ ΕΚΤΑΣΗΣ
20 REM A=ΑΡΧΙΚΗ ΕΚΤΑΣΗ, E=ΡΥΘΜΟΣ ΑΝΑΔΑΣΩΣΗΣ
```

```

30 REM N=ΧΡΟΝΙΑ, T=ΤΕΛΙΚΗ ΕΚΤΑΣΗ
40 REM ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ
50 INPUT 'ΑΡΧΙΚΗ ΕΚΤΑΣΗ=',A
60 INPUT 'ΡΥΘΜΟΣ ΑΝΑΔΑΣΩΣΗΣ=',E
70 INPUT 'ΧΡΟΝΙΑ=',N
80 REM ΒΗΜΑ 2. ΥΠΟΛΟΓΙΣΜΟΣ ΤΕΛΙΚΗΣ ΕΚΤΑΣΗΣ
90 T=A*(1+E)^N
100 REM ΒΗΜΑ 3. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
110 PRINT 'ΤΕΛΙΚΗ ΕΚΤΑΣΗ=',T
120 END

```

**Παράδειγμα 3.** Εάν δανειστούμε ένα ποσό  $A$  με επιτόκιο  $T=E\%$  και υποχρεωθούμε να το ξεχρεώσουμε σε  $N$  χρόνια τότε το ποσό της μηνιαίας δόσης  $M$  θα δίνεται από τον τύπο :

$$M = \frac{A \times T}{12} \times \frac{(1+T)^N}{(1+T)^N - 1} \quad \text{όπου } T = E/100$$

α) Να γραφεί ένα πρόγραμμα υπολογισμού της μηνιαίας δόσης για διάφορες τιμές των  $A$ ,  $E$  και  $N$ .

β) Ποια η μορφή του προγράμματος αν  $A=10.000.000$ ,  $E=10\%$  και  $N=20$ .

### A) Περιγραφή εισόδου-εξόδου

*Είσοδος* – Το ποσό του δανείου  $A$ , το επιτόκιο  $E\%$  και τα χρόνια εξόφλησης  $N$ .

*Έξοδος* – Η μηνιαία δόση αποπληρωμής του δανείου  $M$ .

### B) Παράδειγμα με το χέρι.

Έστω  $N=15$  χρόνια,  $E=15.25\%$  και  $A=20.000.000$  δρχ. τότε εφαρμόζω τον τύπο και έχω

$$M = \frac{20.000.000 \times 0.1525}{12} \times \frac{1.1525^{15}}{1.1525^{15} - 1} \cong 288483 \text{ δρχ.}$$

### Γ) Αλγόριθμος προβλήματος

*Βήμα 1.* Διαβάζουμε τις τιμές του ποσού δανείου  $A$ , του επιτοκίου  $E\%$  και των χρόνων εξόφλησης.

*Βήμα 2.* Υπολογίζουμε την μηνιαία δόση βάσει του τύπου.

*Βήμα 3.* Εκτυπώνουμε τα αποτελέσματα.

#### **Δ) Πρόγραμμα**

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΗΝΙΑΙΑΣ ΔΟΣΗΣ ΔΑΝΕΙΟΥ
20 REM A=ΠΟΣΟ ΔΑΝΕΙΟΥ, N=ΧΡΟΝΙΑ ΕΞΟΦΛΗΣΗΣ
30 REM E=ΕΠΙΤΟΚΙΟ (%), M=ΜΗΝΙΑΙΑ ΔΟΣΗ
40 REM ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ
50 INPUT 'ΠΟΣΟ ΔΑΝΕΙΟΥ=',A
60 INPUT 'ΕΠΙΤΟΚΙΟ (%) =',E
70 INPUT 'ΧΡΟΝΙΑ ΕΞΟΦΛΗΣΗΣ=',N
80 T=E/100
90 REM ΒΗΜΑ 2. ΥΠΟΛΟΓΙΣΜΟΣ ΜΗΝΙΑΙΑΣ ΔΟΣΗΣ
100 M=(A*T/12)*((1+T)^N/((1+T)^N-1))
110 REM ΒΗΜΑ 3. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
120 PRINT 'ΜΗΝΙΑΙΑ ΔΟΣΗ =',M
130 END
```

Στην περίπτωση που  $A=10.000.000$ ,  $E=10\%$  και  $N=20$  θα μπορούσαμε να αντικαταστήσουμε τις εντολές 50, 60 και 70 με μια από τις ακόλουθες :

40 LET A=10000000	40 READ A, E, N
50 LET E=10	50 DATA 10000000,10,20
60 LET N=20	

### **Επαναληπτικές Ασκήσεις στις Εντολές Εισόδου-Εξόδου**

**Άσκηση 1.** Καταθέτουμε σε μια τράπεζα στην αρχή του κάθε έτους ένα σταθερό ποσό  $A$ . Εάν η κατάθεση γίνεται με επιτόκιο  $E\%$ , τότε το κεφάλαιο  $K$  μετά την  $n$ -οστή κατάθεση θα είναι :

$$K = \frac{A \times (1 + E) \times [(1 + E)^N - 1]}{E}$$

Να γραφεί πρόγραμμα υπολογισμού του  $K$  αν είναι γνωστά τα  $A$ ,  $E$  και  $N$ .

**Άσκηση 2.** Με τον όρο δυνητική εξατμισοδιαπνοή εννοούμε την συνολική ποσότητα ύδατος που εξατμίζεται για ένα ορισμένο χρονικό διάστημα από μια γήινη επιφάνεια στην οποία συνεχώς προσφέρεται νερό είτε από την ατμόσφαιρα σαν βροχή είτε από τα βαθύτερα στρώματα του εδάφους. Ένας

εμπειρικός τύπος για την δυνητική εξατμισοδιαπνοή ETP δίνεται από τον Turc και είναι ο εξής :

$$ETP = 0.4 \times \frac{T}{T + 15} \times (I + 50)$$

όπου ETP είναι η δυνητική εξατμισοδιαπνοή σε mm/μήνα, T η μέση μηνιαία θερμοκρασία και I είναι η ολική ακτινοβολία κατά μέρα (μέση του μήνα).

Να γραφεί πρόγραμμα που θα υπολογίζει την δυνητική εξατμισοδιαπνοή εάν είναι γνωστή η μέση μηνιαία θερμοκρασία T και η ολική ακτινοβολία κατά μέρα (μέση του μήνα) I.

**Άσκηση 3.** Είναι γνωστό ότι η μέση θερμοκρασία της ημέρας δίνεται αν διαιρέσουμε με τρία το άθροισμα των θερμοκρασιών που έχουμε στις 8:00 το πρωί, στις 16:00 το μεσημέρι και στις 24:00 το βράδυ. Να γραφεί πρόγραμμα που να υπολογίζει την μέση θερμοκρασία της ημέρας αν δίνονται οι παραπάνω τρεις θερμοκρασίες.

**Άσκηση 4.** Τα ετήσια ποσά της ολικής ηλιακής ακτινοβολίας  $Q_s$  στον Ελλαδικό χώρο, εκφρασμένα σε  $Mjm^{-2}yr^{-1}$  δίνονται με ικανοποιητική ακρίβεια από την εξίσωση :

$$Q_s = 10618,470 - 2,439\phi + 0,347\lambda - 0,301h$$

όπου :  $\phi$  και  $\lambda$  οι γεωγραφικές συντεταγμένες ενός τόπου και  $h$  (σε μέτρα το υψόμετρο του. Να γραφεί πρόγραμμα που να υπολογίζει την ολική ηλιακή ακτινοβολία  $Q_s$  αν δίνονται οι τιμές των  $\phi$ ,  $\lambda$  και  $h$ .

**Άσκηση 5.** Έστω  $d_0, d_1$  οι διάμετροι στην βάση και στο σημείο αποκορύφωσης ενός δέντρου. Αν το συνολικό μήκος του δέντρου είναι  $\ell$ , τότε ο όγκος  $V$  του δέντρου δίνεται από τον τύπο :

$$V = \frac{\pi}{4} \times \ell \times \left( \frac{d_0^2 + d_1^2}{2} \right)$$

Να γραφεί πρόγραμμα που να υπολογίζει τον όγκο του δέντρου αν  $d_0 = 0.5m$ ,  $d_1 = 0.1m$  και  $\ell = 25m$ .

**Άσκηση 6.** Ο υπολογισμός του ύψους  $H$  ενός φράγματος δίνεται από τον παρακάτω τύπο :

$$H = L \frac{J_{\phi} - J_{\alpha}}{100}$$

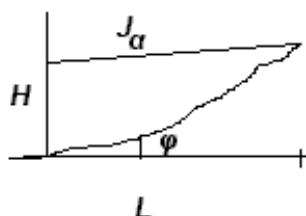
όπου :

$L$  : το μήκος του προς διευθέτηση τμήματος,

$J_{\phi}$  : η φυσική κλίση

$J_{\alpha}$  : η επιδιωκόμενη κλίση αντιστάθμισης

Να γραφεί πρόγραμμα υπολογισμού του ύψους φράγματος αν δίνονται από τον χρήστη οι τιμές των  $L, J_{\phi}, J_{\alpha}$ .



**Άσκηση 7.** Σε ένα δάσος ερυθρελάτης πήραμε ένα δείγμα 10 δέντρων για να υπολογίσουμε το μέσο ύψος. Τα ύψη των δέντρων είναι τα παρακάτω :

11.0, 13.0, 16.5, 12.0, 18.0, 21.5, 15.0, 16.0, 17.0, 13.5

Να γραφεί πρόγραμμα που να υπολογίζει τον μέσο όρο ύψους των παραπάνω δέντρων.

**Άσκηση 8.** Η ταχύτητα ενός ρεύματος κατά την οποία ένα στερεό υλικό αρχίζει να μετακινείται ή όταν κινείται να αποτίθεται, ονομάζεται οριακή ταχύτητα και δίδεται από τον παρακάτω τύπο :

$$U = \sqrt{\frac{p \times \beta \times (d - \gamma) \times (f \sigma \nu \nu(a) - \eta \mu(a))}{\gamma}}$$

όπου :

$U$  : Παραπυθμμένα ταχύτητα του χειμαρρώδους ρεύματος (m/sec)

$\rho$  : ένας συντελεστής που ισούται  $P = \frac{1}{0.076}$  γιατί στερεά υλικά δεν έχουν

κανονική μορφή

$\beta$  : Το μήκος του στερεού υλικού (διάσταση παράλληλη προς τον άξονα της κοίτης) (m)

$d$  : ειδικό βάρος στερεού υλικού ( $\text{kg}/\text{m}^3$ )

$\gamma$  : ειδικό βάρος νερού ( $\text{Kgr}/\text{m}^3$ )

$\alpha$  : γωνία κλίσεως του πυθμένα

$f$  : συντελεστής τριβής

Να γραφεί πρόγραμμα υπολογισμού της οριακής ταχύτητας  $U$ , αν δίδονται τα  $U, \rho, \beta, d, \gamma, \alpha, f$  από τον χρήστη.

**Άσκηση 9.** Να διαβαστούν οι κάθετες πλευρές  $A, B$  ενός ορθογωνίου τριγώνου και στην συνέχεια να υπολογιστεί η υποτείνουσα του τριγώνου  $C$  από τον τύπο :

$$C = \sqrt{A^2 + B^2}$$

**Άσκηση 10.** Να διαβαστεί η ακτίνα ενός κύκλου  $R$  και να υπολογιστεί το εμβαδόν  $E$  του κύκλου από τον τύπο :

$$E = \pi R^2 \quad \text{όπου } \pi = 3.14$$

### Εντολές ελέγχου και διακλαδώσεων.

Στο προηγούμενο κεφάλαιο παρατηρήσαμε ότι οι εντολές του προγράμματος εκτελούνται κατά σειρά. Εάν θέλουμε όμως εμείς φτάνοντας σε κάποια εντολή να αλλάξουμε την ροή του προγράμματος βάση κάποιας συνθήκης και όχι μόνο, τότε χρησιμοποιούμε μια σειρά από εντολές της BASIC που θα αναπτύξουμε παρακάτω.



## Η εντολή **GOTO**.

**Σκοπός :** Η αλλαγή της ροής (διακλάδωση ή ανακύκλωση) του προγράμματος.

**Σύνταξη :** GOTO αριθμός γραμμής

**Σχόλια :** Η επόμενη εντολή που θα εκτελεστεί θα είναι αυτή με τον συγκεκριμένο <αριθμό γραμμής>, καθώς και όσες την ακολουθούν.

**Παράδειγμα.** Θέλω να υπολογίσω τον μέσο όρο των βαθμών 5 φοιτητών οι οποίοι έχουν πάρει 5 μαθήματα στο Α' εξάμηνο. Να γραφεί πρόγραμμα που να εκτυπώνει τα ονοματεπώνυμα των φοιτητών και τους μέσους όρους αντίστοιχα αν γνωρίζω τα παρακάτω στοιχεία τους :

A/A	Ονοματεπώνυμο	Βαθ. 1	Βαθ. 2	Βαθ. 3	Βαθ. 4	Βαθ. 5
1	Νικολαΐδου Μάρθα	5	6	7	4	3
2	Μάλτεπε Φιλιώ	4	5	7	8	9
3	Τασκούδη Ολυμπία	5	5	6	6	7
4	Δουλούδη Λαμπρινή	4	5	5	6	9
5	Ρίτκου Άννα	5	8	8	9	10

### A) Περιγραφή εισόδου-εξόδου

*Είσοδος* – Τα ονοματεπώνυμα των φοιτητών και οι βαθμοί τους.

*Έξοδος* – Τα ονοματεπώνυμα των φοιτητών και οι μέσοι όροι των βαθμών τους.

### B) Παράδειγμα με το χέρι.

Διαβάζω το πρώτο όνομα της φοιτήτριας «Νικολαΐδου Μάρθα», αθροίζω τους βαθμούς της και διαιρώ με το 5 (πλήθος μαθημάτων) και έχω :

$$MO = \frac{5 + 6 + 7 + 4 + 3}{5} = 5$$



και συνεπώς εκτυπώνω το όνομα της και τον μέσο όρο :

Νικολαΐδου Μάρθα 5

Το ίδιο θα έπρεπε να κάνω και για τις 5 φοιτήτριες.

### Γ) Αλγόριθμος προβλήματος

*Βήμα 1.* Διαβάζω τα ονόματα των φοιτητών και τους βαθμούς τους. Υπάρχουν τρεις τρόποι ανάγνωσης των δεδομένων. Ο πρώτος είναι με την εντολή LET, αλλά θα χρειαστούν γύρω στις 30 εντολές LET, πράγμα που το αποκλείω αν και είναι δυνατό. Δεύτερος τρόπος υλοποίησης είναι με την εντολή INPUT, αλλά αν κατά την διάρκεια εισαγωγής δεδομένων κάνω ένα λάθος π.χ. στον 3<sup>ο</sup> βαθμό της 4<sup>ης</sup> φοιτήτριας, τότε θα πρέπει να ξαναδώσω όλα τα δεδομένα από την αρχή. Ο πιο σίγουρος τρόπος για να δώσω δεδομένα τα οποία τα γνωρίζω εκ των προτέρων είναι με την εντολή READ-DATA. Το ζήτημα είναι πως θα υλοποιήσω την READ-DATA.

*Βήμα 2.* Υπολογίζουμε τον μέσο όρο της φοιτήτριας.

*Βήμα 3.* Εκτυπώνουμε τα αποτελέσματα.

*Βήμα 4.* Πηγαίνω στο βήμα 1.

### Δ) Πρόγραμμα

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ ΦΟΙΤΗΤΩΝ
20 REM ΟΝΟΜΑ$=ΟΝΟΜΑΤΕΠΩΝΥΜΟ  Β1,Β2,Β3,Β4,Β5 =ΒΑΘΜΟΙ
30 REM ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ
40 READ ΟΝΟΜΑ$,Β1,Β2,Β3,Β4,Β5
50 REM ΒΗΜΑ 2. ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ
60 ΜΟ=(Β1+Β2+Β3+Β4+Β5)/5
70 REM ΒΗΜΑ 3. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
80 PRINT ΟΝΟΜΑ$, ΜΟ
90 REM ΒΗΜΑ 4. ΠΗΓΑΙΝΕ ΣΤΟ ΠΡΩΤΟ ΒΗΜΑ
100 GOTO 30
110 DATA Νικολαΐδου Μάρθα,5,6,7,4,3
120 DATA Μάλτεπε Φιλιώ,4,5,7,8,9
130 DATA Τασκούδη Ολυμπία,5,5,6,6,7
140 DATA Δουλούδη Λαμπρινή,4,5,5,6,9

```

150 DATA Ρίτκου Άννα,5,8,8,9,10

160 END

Κατά την εκτέλεση του προγράμματος θα εκτελεστούν οι εντολές από 10 έως 90 και θα εκτυπώσουν τα αποτελέσματα για τον πρώτο φοιτητή. Η εντολή 100 θα στείλει την ροή του προγράμματος στην εντολή 30. Θα εκτελεστούν οι εντολές 30-90 και θα εκτυπώσουν τα αποτελέσματα του δεύτερου φοιτητή κ.ο.κ. έως ότου διαβαστούν και τα στοιχεία του πέμπτου φοιτητή και βρεθεί ο μέσος όρος του. Στην συνέχεια ο Η/Υ θα πάει στην εντολή 30-40 θα προσπαθήσει να διαβάσει καινούργιες τιμές για τις μεταβλητές ΟΝΟΜΑ\$,B1,B2,B3.B4,B5 αλλά πλέον ο δείκτης στην DATA θα έχει φτάσει στο τέλος και δεν θα μπορέσει να αντιστοιχίσει άλλες τιμές και συνεπώς ο Η/Υ θα τελειώσει με το μήνυμα OUT OF DATA. Για τον λόγο αυτό όπως θα δούμε παρακάτω χρησιμοποιούμε την εντολή GOTO συνήθως με συνδυασμό άλλων εντολών όπως η IF-THEN-ELSE.

## Η εντολή IF-THEN-ELSE.

**Σκοπός :** Η εκτέλεση εντολής (ή/και εντολών) αν μια συνθήκη ικανοποιείται.

### Σύνταξη :

*1<sup>η</sup> μορφή :* IF <συνθήκη> <εντολή>

*Σχόλια :* Εάν η <συνθήκη> είναι αληθής τότε θα εκτελεστεί η <εντολή>.

*2<sup>η</sup> μορφή :* IF <συνθήκη> THEN <εντολή>

*Σχόλια :* Εάν η <συνθήκη> είναι αληθής τότε θα εκτελεστεί η <εντολή>.

*3<sup>η</sup> μορφή :* IF <συνθήκη> THEN <εντολή 1> ELSE <εντολή 2>

*Σχόλια :* Εάν η <συνθήκη> είναι αληθής τότε θα εκτελεστεί η <εντολή 1> διαφορετικά (δηλ. η συνθήκη είναι ψευδής) θα εκτελεστεί η <εντολή 2>.

Η πιο απλή μορφή της συνθήκης έχει την εξής μορφή :

Σταθερά Μεταβλητή Παράσταση	<b>Τελεστής Σύγκρισης</b>	Σταθερά Μεταβλητή Παράσταση
-----------------------------------	-------------------------------	-----------------------------------

Όπου τελεστής σύγκρισης είναι ένας από τους παρακάτω :

<	Μικρότερο
>	Μεγαλύτερο
=	Ίσο
>= ή =>	Μεγαλύτερο ίσο
<= ή =<	Μικρότερο ίσο
<> ή ><	Διάφορο

**Παράδειγμα.**

50 IF 5<7 GOTO 100

Εάν η συνθήκη (5<7) είναι αληθής (που είναι αληθής) τότε πήγαινε στην εντολή με αριθμό γραμμής 100.

**Παράδειγμα.**

40 A=30

50 IF A>15 THEN PRINT A

Εάν η συνθήκη (A>15) είναι αληθής (που είναι αληθής λόγω του ότι το A είναι 30) τότε εκτύπωσε την τιμή του A (δηλ. 30).

**Παράδειγμα.**

40 B=40

50 IF B-20<>10 THEN PRINT B ELSE GOTO 100

Εάν η συνθήκη (B-20<>10) είναι αληθής (που είναι λόγω του ότι B-20=20 και συνεπώς διάφορο του 10) τότε εκτύπωσε την τιμή του B, διαφορετικά πήγαινε στην εντολή με αριθμό γραμμής 100.

Στον όρο <συνθήκη> όμως μπορεί να έχω και συνδυασμό συνθηκών. Ο συνδυασμός αυτός γράφεται με την χρήση των παρακάτω λογικών τελεστών :

<b><u>NOT</u></b>	OXI
AND	ΚΑΙ
OR	Ή

Η προτεραιότητα των λογικών τελεστών είναι αυτή που δίδεται στο παραπάνω σχήμα δηλ. πρώτα εκτελείται η NOT μετά η AND και τέλος η OR.

Οι πίνακες αληθείας των παραπάνω λογικών τελεστών είναι οι εξής :

<b>&lt;Πρόταση 1&gt;</b>	<b>NOT &lt;Πρόταση 1&gt;</b>
Αληθής	Ψευδής
Ψευδής	Αληθής

<b>&lt;Πρόταση 1&gt;</b>	<b>&lt;Πρόταση 2&gt;</b>	<b>&lt;Πρόταση 1&gt; AND &lt;Πρόταση 2&gt;</b>
Αληθής	Αληθής	Αληθής
Αληθής	Ψευδής	Ψευδής
Ψευδής	Αληθής	Ψευδής
Ψευδής	Ψευδής	Ψευδής

<b>&lt;Πρόταση 1&gt;</b>	<b>&lt;Πρόταση 2&gt;</b>	<b>&lt;Πρόταση 1&gt; OR &lt;Πρόταση 2&gt;</b>
Αληθής	Αληθής	Αληθής
Αληθής	Ψευδής	Αληθής
Ψευδής	Αληθής	Αληθής
Ψευδής	Ψευδής	Ψευδής

### Παράδειγμα.

20 A=30 : B=40

30 IF A>20 AND B<30 THEN PRINT A+B ELSE GOTO 70

Είναι φανερό ότι η συνθήκη A>20 είναι αληθής ενώ η συνθήκη B<50 είναι ψευδής. Συνεπώς από τον πίνακα αληθείας της AND η όλη συνθήκη μας είναι ψευδής και συνεπώς δεν θα εκτυπωθεί το A+B αλλά θα εκτελεστεί η εντολή με αριθμό γραμμής 70.

**Παράδειγμα.**

20 A=20 : K=20

30 IF A<>30 OR K>30 THEN GOTO 50 ELSE PRINT A+B+K

Η συνθήκη A<>30 είναι αληθής, ενώ η συνθήκη K>30 είναι ψευδής. Συνεπώς από τον πίνακα αληθείας της OR η όλη συνθήκη θα είναι αληθής.

**Παράδειγμα.**

20 A=20 : B=10 : K=30

30 IF NOT (A>30) AND (B>20) OR (K<>10) THEN PRINT A+B+C ELSE GOTO 70

Επειδή έχουμε συνδυασμό λογικών τελεστών είχαμε αναφέρει ότι σε αυτήν την περίπτωση τηρούμαι την σειρά προτεραιότητας NOT,AND,OR. Άρα πρώτα εξετάζω την συνθήκη NOT (A>30) που είναι αληθής. Στην συνέχεια εκτελείται η AND που εξετάζει την παρακάτω λογική παράσταση (αληθής) AND (B>20) που είναι ψευδής. Τέλος εκτελείται η OR που εξετάζει την λογική παράσταση (ψευδής OR (K<>10)) που είναι αληθής. Άρα η λογική μας παράσταση είναι αληθής και συνεπώς θα εκτυπωθεί το A+B+K.

Μια παράσταση η οποία συνδέει λογικές προτάσεις με τους συνδέσμους NOT, AND, OR ονομάζεται *λογική παράσταση* σε αντίθεση με την *αριθμητική παράσταση* που συνδυάζει αριθμούς ή μεταβλητές με τα σύμβολα των πράξεων.

Χρήσιμες συνθήκες είναι οι εξής :

Το x βρίσκεται στο διάστημα [A,B]	$X \geq A \text{ AND } X \leq B$
Το x βρίσκεται εκτός του διαστήματος [A,B]	$X < A \text{ OR } X > B$
Το x είναι πολλαπλάσιο του a	$X \text{ MOD } A = 0$
Το x είναι θετικό (αρνητικό)	$X > 0 \text{ (} X < 0 \text{)}$
Το x είναι άρτιο (περιττό)	$X \text{ MOD } 2 = 0 \text{ (} X \text{ MOD } 2 <> 0 \text{)}$

**Παράδειγμα.** Ο διαχωρισμός της ροής του νερού σε ποτάμια ή χειμαρρώδη δίνεται παρακάτω :

<u>ΑΝ</u>	<u>ΤΟΤΕ</u>
$U < \sqrt{g \times h}$	Ποτάμια ροή
$U = \sqrt{g \times h}$	Οριακή κατάσταση
$U > \sqrt{g \times h}$	Χειμαρρώδη ροή

όπου :

$U$  = μέση ταχύτητα ροής του νερού (m/sec),

$g$ =η επιτάχυνση της βαρύτητας σε m/sec<sup>2</sup>,

$h$ =βάθος του υδάτινου ρεύματος (m)

Να γραφεί πρόγραμμα που να υπολογίζει το είδος της ροής του νερού αν δίνεται η μέση ταχύτητα ροής του νερού  $U$ , το βάθος του υδάτινου ρεύματος  $h$ , και είναι γνωστό ότι  $g=9.81$ m/sec<sup>2</sup>.

### A) Περιγραφή εισόδου-εξόδου

*Είσοδος* – Η μέση ταχύτητα του νερού  $U$ , το βάθος του υδάτινου ρεύματος  $h$ , και η επιτάχυνση της βαρύτητας  $g=9.81$ m/sec<sup>2</sup>.

*Εξοδος* – Μια από τις παρακάτω ενδείξεις : «Ποτάμια ροή», «Οριακή κατάσταση», «Χειμαρρώδης ροή».

### B) Παράδειγμα με το χέρι.

Έστω ότι  $U=10$ m/sec,  $h=5$ m και  $g=9.81$ m/sec<sup>2</sup>. Τότε υπολογίζω το

$$\sqrt{g \times h} = \sqrt{9.81 \times 5} \approx 7$$

και κάνω έλεγχο αν η ταχύτητα ροής του νερού είναι μεγαλύτερη από 7. Επειδή  $U=10$ m/sec, άρα η ροή του νερού είναι χειμαρρώδης.

### Γ) Αλγόριθμος προβλήματος

*Βήμα 1.* Διαβάζω τις τιμές των  $U, h$  με την βοήθεια της INPUT έτσι ώστε το πρόγραμμα μου να μπορεί να εκτελεστεί και με διαφορετικά δεδομένα. Θέτω επίσης  $g=9.81$ .

*Βήμα 2.* Υπολογίζουμε την ποσότητα  $\sqrt{g \times h}$ .

*Βήμα 3.* Ελέγχω αν η παραπάνω ποσότητα είναι μεγαλύτερη, ίση ή μικρότερη από την μέση ταχύτητα ροής του νερού και εμφανίζω το αντίστοιχο μήνυμα.

**Δ) Πρόγραμμα**

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΟΡΦΗΣ ΡΟΗΣ ΝΕΡΟΥ
20 REM U=ΜΕΣΗ ΤΑΧΥΤΗΤΑ ΝΕΡΟΥ, G=9.81
30 H=ΥΨΟΣ ΥΔΑΤΙΝΟΥ ΡΕΥΜΑΤΟΣ
40 REM ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ
50 INPUT 'ΤΑΧΥΤΗΤΑ ΡΟΗΣ ΝΕΡΟΥ (M/SEC)=';U
60 INPUT 'ΥΨΟΣ ΥΔΑΤΙΝΟΥ ΡΕΥΜΑΤΟΣ (M)=';H
70 G=9.81
80 REM ΒΗΜΑ 2. ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΡΙΖΑΣ ΤΟΥ GxH
90 UOR=SQR(G*H)
100 REM ΒΗΜΑ 3. ΕΛΕΓΧΟΣ ΜΟΡΦΗΣ ΡΟΗΣ ΝΕΡΟΥ
110 IF U>UOR THEN PRINT 'ΧΕΙΜΑΡΡΩΔΗΣ ΡΟΗ'
120 IF U=UOR THEN PRINT 'ΟΡΙΑΚΗ ΚΑΤΑΣΤΑΣΗ'
130 IF U<UOR THEN PRINT 'ΠΟΤΑΜΙΑ ΡΟΗ'
140 END

```

Οι εντολές 100-130 θα μπορούσαν να αντικατασταθούν και από την εξής πιο πολύπλοκη εντολή :

```

100 IF U>SQR(G*H) THEN PRINT 'ΧΕΙΜΑΡΡΩΔΗΣ ΡΟΗ' ELSE IF
U=SQR(G*H) THEN PRINT 'ΟΡΙΑΚΗ ΚΑΤΑΣΤΑΣΗ' ELSE PRINT 'ΠΟΤΑΜΙΑ
ΡΟΗ'

```

**Παράδειγμα.** Η μέτρηση της μέγιστης παροχής νερού ενός ρεύματος δίνεται από τον εξής τύπο του Υδρογραφικού Ινστιτούτου της Βιέννης :

Για $F < 25 \text{Km}^2$	$Q_{\max} = a \times 16.67$
Για $F \geq 25 \text{Km}^2$	$Q_{\max} = a \times \left( 1.16 + \frac{187.7}{F} \right)$

όπου :

$Q_{\max}$  = μέγιστη παροχή ( $\text{m}^3/\text{sec}$ )

F=έκταση λεκάνης απορροής ( $\text{Km}^2$ )

a=συντελεστής απορροής με μέση τιμή 0.6

Να γραφεί πρόγραμμα υπολογισμού της μέγιστης παροχής νερού ρεύματος  $Q_{\max}$  αν είναι γνωστή η έκταση λεκάνης απορροής F και το a=0.6.

### Α) Περιγραφή εισόδου-εξόδου

Είσοδος – Η έκταση λεκάνης απορροής F και το a=0.6.

Εξοδος – Η μέγιστη παροχή νερού  $Q_{\max}$ .

### Β) Παράδειγμα με το χέρι.

Έστω ότι F=30 Km<sup>2</sup> και a=0.6. Τότε επειδή F>25 άρα η μέγιστη παροχή νερού θα είναι :

$$Q_{\max} = 0.6 \times \left( 1.16 + \frac{387.7}{30} \right) = 8.45 \text{ Km}^3 / \text{sec}$$

### Γ) Αλγόριθμος προβλήματος

Βήμα 1. Διαβάζω την τιμή του F και θέτω a=0.6.

Βήμα 2. Ελέγχω την τιμή του F σε σχέση με το 25 και αντίστοιχα υπολογίζω την μέγιστη παροχή νερού.

Βήμα 3. Εκτυπώνω το αποτέλεσμα.

### Δ) Πρόγραμμα

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΓΙΣΤΗΣ ΠΑΡΟΧΗΣ ΝΕΡΟΥ
20 REM F=ΕΚΤΑΣΗ ΛΕΚΑΝΗΣ ΑΠΟΡΡΟΗΣ , A=0.6
30 REM QMAX=ΜΕΓΙΣΤΗ ΠΑΡΟΧΗ ΝΕΡΟΥ
40 REM ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ
50 INPUT 'ΕΚΤΑΣΗ ΛΕΚΑΝΗΣ ΑΠΟΡΡΟΗΣ (ΚΜ^2)=';F
60 A=0.6
70 REM ΒΗΜΑ 2. ΕΛΕΓΧΟΣ ΤΟΥ F
80 IF F>=25 THEN QMAX=A*16.67 ELSE QMAX=A*(1.16+387.7/F)
90 REM ΒΗΜΑ 3. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΟΣ
100 PRINT 'ΜΕΓΙΣΤΗ ΠΑΡΟΧΗ ΝΕΡΟΥ=';QMAX
110 END
```



## Επαναληπτικές Ασκήσεις στις Εντολές Ελέγχου και Διακλάδωσης

**Άσκηση 1.** Να γραφεί πρόγραμμα που να διαβάζει δύο αριθμούς. Εάν οι αριθμοί αυτοί είναι ίσοι, να διαβάζονται δύο καινούργιοι αριθμοί. Εάν οι αριθμοί είναι διάφοροι να εμφανίζεται στην οθόνη ο μεγαλύτερος. Μπορείτε να επεκτείνετε το πρόγραμμα ούτως ώστε να καλύπτει την περίπτωση τριών αριθμών.

**Άσκηση 2.** Να γραφεί πρόγραμμα που να διαβάζει μια τιμή για την μεταβλητή X. Εάν η τιμή αυτή δεν ανήκει στο κλειστό διάστημα  $[-10,10]$  τότε να διαβάζει το πρόγραμμα μια νέα τιμή για το X, διαφορετικά : α) να εμφανίζει το 0 εάν το X είναι 0 ή β) το  $1/X$  όταν το X είναι διάφορο του μηδέν.

**Άσκηση 3.** Να γραφεί πρόγραμμα που να διαβάζει τρεις αριθμούς A, B, C. Στην συνέχεια να υπολογίζει το r και E όπου :

$$r = \frac{A + B + C}{2}$$

$$E = \sqrt{r(r - A)(r - B)(r - C)}$$

**Σημείωση :** Να γίνεται έλεγχος αν η υπόριζη ποσότητα είναι θετική πριν τον υπολογισμό της. Ο παραπάνω τύπος είναι γνωστός και ως τύπος του Ήρων του Αλεξανδρείας και υπολογίζει το εμβαδόν ενός τριγώνου E αν είναι γνωστά τα μήκη των πλευρών του τριγώνου A,B,C.

**Άσκηση 4.** Να διαβαστούν τρεις αριθμοί οι οποίοι παριστούν τα μήκη των πλευρών ενός τριγώνου. Στην συνέχεια να διαπιστωθεί αν το τρίγωνο είναι :

- α) ισοσκελές,
- β) ισόπλευρο,
- γ) ορθογώνιο,
- δ) σκαληνό.

**Άσκηση 5.** Να διαβαστούν τρεις αριθμοί οι οποίοι παριστούν τις γωνίες ενός τριγώνου. Στην συνέχεια να διαπιστωθεί αν το τρίγωνο είναι :

- α) ισοσκελές,
- β) ισόπλευρο,

- γ) ορθογώνιο,
- δ) οξυγώνιο,
- ε) αμβλυγώνιο.

**Άσκηση 6.** Εάν  $A=4$ ,  $B=1$  και  $C=7$  ποια από τις παρακάτω προτάσεις είναι αληθής :

$B < C$  AND ( $A < B$  OR  $A > C$ )

NOT ( $A > B$  OR NOT ( $B > C$ ))

**Άσκηση 7.** Για ποια από τις παρακάτω τιμές του  $X$  {1, 15, 5, 25} είναι η παρακάτω πρόταση αληθής :

$X > 10$  AND  $X < 20$  OR  $X = 1$

### **Ανακύκλωση και χρήση μετρητών σε προγράμματα.**

Πολλές φορές χρειαζόμαστε μια διαδικασία να την επαναλάβουμε συγκεκριμένο αριθμό φορών π.χ. θέλουμε να βρούμε τον μέσο όρο των βαθμών από 10 φοιτητές που έχουν πάρει 5 μαθήματα, θέλουμε να βρούμε το πλήθος από ένα σύνολο αριθμών που ικανοποιούν κάποια ιδιότητα κ.τ.λ. Η επανάληψη μιας διαδικασίας παραπάνω από μια φορές ονομάζεται ανακύκλωση. Προκειμένου να λύσουμε το πρόβλημα της ανακύκλωσης χρησιμοποιούμε μια καινούργια μεταβλητή  $I$  η οποία αυξάνει κατά ένα κάθε φορά που επαναλαμβάνεται η διαδικασία μας. Επίσης είναι απαραίτητη η ύπαρξη στο πρόγραμμα μιας εντολής IF-THEN-ELSE που θα ελέγχει αν αυτό το  $I$ , δηλαδή ο αριθμός των επαναλήψεων της διαδικασίας έχει ξεπεράσει το όριο που θέλουμε. Για να γίνουμε πιο κατανοητοί θα δούμε ένα παράδειγμα που αναφέραμε στην εντολή GOTO και στην συνέχεια θα δώσουμε μια γενική μορφή επίλυσης των προβλημάτων αυτών.

**Παράδειγμα.** Θέλω να υπολογίσω τον μέσο όρο των βαθμών 5 φοιτητών οι οποίοι έχουν πάρει 5 μαθήματα στο Α' εξάμηνο. Να γραφεί πρόγραμμα που να εκτυπώνει τα ονοματεπώνυμα των φοιτητών και τους μέσους όρους αντίστοιχα αν γνωρίζω τα παρακάτω στοιχεία τους :

A/A	Όνοματεπώνυμο	Βαθ. 1	Βαθ. 2	Βαθ. 3	Βαθ. 4	Βαθ. 5
1	Νικολαΐδου Μάρθα	5	6	7	4	3
2	Μάλτεπε Φιλιώ	4	5	7	8	9
3	Τασκούδη Ολυμπία	5	5	6	6	7
4	Δουλούδη Λαμπρινή	4	5	5	6	9
5	Ρίτκου Άννα	5	8	8	9	10

Με την επίλυση του προβλήματος αυτού είχαμε ασχοληθεί στην ενότητα της εντολής GOTO και είχαμε διαπιστώσει ότι το πρόγραμμα τελειώνει απότομα και μάλιστα με το μήνυμα OUT OF DATA. Μηνύματα όπως αυτό δεν επιτρέπονται από προγραμματιστικής απόψεως. Για τον σκοπό αυτό θα χρησιμοποιήσουμε έναν μετρητή I που θα μετράει το πλήθος των ατόμων που έχουμε εξετάσει. Ο μετρητής αυτός ξεκινάει με την τιμή 1, για να δηλώσει τον πρώτο φοιτητή. Στην συνέχεια εφόσον έχουμε τελειώσει την διαδικασία εύρεσης του μέσου όρου του πρώτου φοιτητή, αυξάνουμε τον μετρητή αυτόν κατά ένα ( $I=I+1$ ) έτσι ώστε να πάμε στον επόμενο φοιτητή. Πριν πάμε στον επόμενο φοιτητή ελέγχουμε αν αυτή η τιμή του I έχει ξεπεράσει τον αριθμό 5 που είναι και το πλήθος των φοιτητών μου. Σηριζόμενοι στο παραπάνω σκεπτικό μπορούμε να διατυπώσουμε τον αλγόριθμο του προβλήματος :

### A) Αλγόριθμος προβλήματος

*Βήμα 1.* Διαβάζω το πλήθος των φοιτητών έστω N.

*Βήμα 2.* Θέτω  $I=1$ .

*Βήμα 3.* Διαβάζω τα στοιχεία του I φοιτητή.

*Βήμα 4.* Υπολογίζω τον μέσο όρο του.

*Βήμα 5.* Εκτυπώνω το ονοματεπώνυμο και τον μέσο όρο του.

*Βήμα 6.* Αυξάνω το I κατά ένα ( $I=I+1$ ).

*Βήμα 7.* Εάν το I δεν έχει ξεπεράσει το N πηγαίνω στο βήμα 3.

*Βήμα 8.* Τέλος

### B) Πρόγραμμα

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ ΦΟΙΤΗΤΩΝ
```

```
20 REM ΟΝΟΜΑ$=ΟΝΟΜΑΤΕΠΩΝΥΜΟ  Β1,Β2,Β3,Β4,Β5 =ΒΑΘΜΟΙ
```

```
30 ΒΗΜΑ 1. ΕΙΣΑΓΩΓΗ ΠΛΗΘΟΥΣ ΦΟΙΤΗΤΩΝ
40 N=5
50 REM ΒΗΜΑ 2. ΘΕΤΩ I=1
60 I=1
70 REM ΒΗΜΑ 3. ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ I ΦΟΙΤΗΤΗ
80 READ ΟΝΟΜΑ$,B1,B2,B3,B4,B5
90 REM ΒΗΜΑ 4. ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ
100 ΜΟ=(B1+B2+B3+B4+B5)/5
110 REM ΒΗΜΑ 5. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
120 PRINT ΟΝΟΜΑ$, ΜΟ
130 REM ΒΗΜΑ 6. ΑΥΞΑΝΩ ΤΟ I ΚΑΤΑ Ένα
140 I=I+1
150 REM ΒΗΜΑ 7. ΕΑΝ ΤΟ I ΔΕΝ ΞΕΠΕΡΑΣΕ ΤΟ Ν ΠΗΓΑΙΝΕ ΣΤΟ ΒΗΜΑ 3
160 IF I<=N THEN GOTO 70
110 DATA Νικολαΐδου Μάρθα,5,6,7,4,3
120 DATA Μάλτεπε Φιλιώ,4,5,7,8,9
130 DATA Τασκούδη Ολυμπία,5,5,6,6,7
140 DATA Δουλούδη Λαμπρινή,4,5,5,6,9
150 DATA Ρίτκου Άννα,5,8,8,9,10
160 END
```

### Διατύπωση γενικού αλγορίθμου

**Βήμα 1.** Ζητώ να μάθω πόσες φορές θα επαναληφθεί η διαδικασία. Έστω N.

Σε συγκεκριμένες ασκήσεις ζητώ να μάθω το πλήθος των αριθμών που θα επεξεργαστώ. Εάν ο αριθμός αυτός είναι γνωστός εκ των προτέρων τον τοποθετώ στο πρόγραμμα με την εντολή LET (π.χ. LET N=10), διαφορετικά τον ζητώ από τον χρήστη με την εντολή INPUT (π.χ. INPUT ‘‘Δώσε πλήθος :’,N)

**Βήμα 2.** Αρχικοποίηση παραμέτρων.

Θέτω :

I=1 (μετρητής, που χρησιμοποιείται για να μετράει το πλήθος των ανακυκλώσεων ή καλύτερα των αριθμών που επαναλαμβάνεται μια διαδικασία)

S=0 (αθροιστής, που χρησιμοποιείται όταν μας ζητούν άθροισμα αριθμών)

$P=1$  (πολλαπλασιαστής, που χρησιμοποιείται όταν μας ζητούν γινόμενο αριθμών)

$PL=0$  (μετρητής πλήθους, που χρησιμοποιείται όταν μας ζητούν το πλήθος αριθμών που ικανοποιούν κάποια συνθήκη)

Σε περίπτωση που μας ζητούν παραπάνω από ένα αθροίσματα, γινόμενα ή πλήθη μπορούμε να χρησιμοποιήσουμε παραπάνω από έναν αθροιστές ( $S1, S2, \dots$ ) ή πολλαπλασιαστές ( $P1, P2, \dots$ ) ή μετρητές ( $PL1, PL2, \dots$ )

**Βήμα 3.** Αναζήτηση αριθμού ή δεδομένων προς επεξεργασία. Η αναζήτηση αυτή γίνεται με την εντολή INPUT εάν δεν είναι γνωστά εκ των προτέρων τα δεδομένα (π.χ. INPUT 'Δώσε αριθμό :', X), ενώ με την εντολή READ-DATA στην αντίθετη περίπτωση (π.χ. READ X).

**Βήμα 4.** Επεξεργασία των δεδομένων, είτε βάσει κάποιας συνθήκης είτε χωρίς συνθήκη.

*Χωρίς συνθήκη :* Εάν χρειάζομαι το άθροισμα κάποιων αριθμών χρησιμοποιώ την εντολή  $S=S+X$  ενώ για το γινόμενο των αριθμών που θέλω να επεξεργαστώ χρησιμοποιώ την εντολή  $P=P*X$ .

*Με συνθήκη :* Εάν χρειάζομαι το άθροισμα κάποιων αριθμών που ικανοποιούν κάποια «συνθήκη» χρησιμοποιώ την εντολή

IF «συνθήκη» THEN  $S=S+X$

Εάν χρειάζομαι το γινόμενο κάποιων αριθμών που ικανοποιούν κάποια «συνθήκη» χρησιμοποιώ την εντολή

IF «συνθήκη» THEN  $P=P*X$

Εάν χρειάζομαι το πλήθος κάποιων αριθμών που ικανοποιούν κάποια «συνθήκη» χρησιμοποιώ την εντολή

IF «συνθήκη» THEN  $PL=PL+1$

Υπάρχει επίσης περίπτωση να μην μου ζητάν άθροισμα ή γινόμενο αλλά ένα άλλο είδος επεξεργασίας δεδομένων όπως αυτό της προηγούμενης άσκησης (υπολογισμός μέσου όρου)

**Βήμα 5.** Αυξάνω τον μετρητή του πλήθους I κατά ένα.

**Βήμα 6.** Ελέγχω αν η διαδικασία έχει επαναληφθεί N φορές. Εάν όχι πηγαίνω στο βήμα 3 διαφορετικά συνεχίζω με το επόμενο βήμα.

IF  $I \leq N$  THEN GOTO "Βήμα 3"

**Βήμα 7.** Εμφανίζω τα αποτελέσματα.

	ΧΩΡΙΣ ΣΥΝΘΗΚΗ		ΜΕ ΣΥΝΘΗΚΗ	
	Εκτός loop	Εντός loop	Εκτός loop	Εντός loop
Άθροισμα	S=0	S=S+X	S=0	IF «συνθήκη» THEN S=S+X
Γινόμενο	P=1	P=P*X	P=1	IF «συνθήκη» THEN P=P*X
Πλήθος	PL=0	PL=PL+1	PL=0	IF «συνθήκη» THEN PL=PL+1

**Σημείωση :** Μια επιπλέον συνθήκη που μπορεί να ζητηθεί είναι η εύρεση του μέσου όρου ενός πλήθους στοιχείων που ικανοποιούν κάποια συνθήκη. Στην περίπτωση αυτή θα πρέπει να βρω : το πλήθος των στοιχείων που ικανοποιούν την συνθήκη έστω PL, το άθροισμα των στοιχείων αυτών έστω S και στην συνέχεια να υπολογίσω τον μέσο όρο MO που θα είναι S/PL εάν το PL είναι διάφορο του μηδενός ή 0 διαφορετικά (IF PL=0 THEN MO=0 ELSE MO=S/PL).

Παρακάτω θα δώσουμε ορισμένα παραδείγματα εφαρμογής του παραπάνω αλγόριθμου.

### Παράδειγμα. (Χωρίς συνθήκη)

Δίνονται N αριθμοί. Να υπολογιστεί :

- A) το άθροισμα τους,
- B) το γινόμενο τους.

### A) Περιγραφή εισόδου-εξόδου

*Είσοδος* – Οι N αριθμοί.

*Έξοδος* – Το άθροισμα τους και το γινόμενο τους.

### B) Παράδειγμα με το χέρι.

Εάν έχω 5 αριθμούς : 2,3,4,2,1. Τότε το άθροισμα είναι 12 ενώ το γινόμενο 48. Παρατηρώ ότι στο άθροισμα διατηρώ πάντα μια θέση στο μυαλό μου για να προσθέτω τον καινούργιο αριθμό, χωρίς να με ενδιαφέρουν οι προηγούμενοι. Δηλαδή τοποθετώ στην αρχή στο μυαλό μου τον αριθμό 2, στην συνέχεια προσθέτω το 3 και πλέον σβήνω από το μυαλό μου τον 2 και πλέον στην θέση του έχω τον 5 κ.ο.κ. έως του φτάσω στον 1 και συνεπώς έχω άθροισμα 12. Το ίδιο συμβαίνει και με τον πολλαπλασιασμό. Την θέση

αυτή που διατηρώ κάθε φορά στο μυαλό μου για το άθροισμα ή το γινόμενο την παίζει αντίστοιχα στο πρόγραμμα μου ο αθροιστής S και ο πολλαπλασιαστής P που δεν είναι τίποτα άλλο παρά θέσεις της μνήμης του Η/Υ.

### Γ) Αλγόριθμος προβλήματος

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών : α) Εάν το πλήθος είναι γνωστό, έστω 20 τότε χρησιμοποιώ την εντολή LET N=20 β) Εάν το πλήθος δεν είναι γνωστό όπως στην άσκηση μου τότε χρησιμοποιώ την εντολή INPUT “Δώσε πλήθος :”;N.

*Βήμα 2.* Αρχικοποίηση των παραμέτρων.

Θέτω I=1 για να δηλώσω ότι ακολουθεί ο πρώτος αριθμός που θα διαβάσω. Διαβάζω καλά την άσκηση και σημειώνω τις λέξεις κλειδιά, από αυτά που μου ζητάει. Δηλαδή στο συγκεκριμένο πρόβλημα σημειώνω την λέξη άθροισμα στο A και γινόμενο στο B ζήτημα αντίστοιχα. Οι λέξεις αυτές θα με βοηθήσουν να επιλέξω το είδος των παραμέτρων που θα χρησιμοποιήσω εκτός και εντός της ανακύκλωσης. Στο πρόβλημα λοιπόν αυτό θα θέσω έναν αθροιστή S=0 επειδή ζητάει η άσκηση άθροισμα και έναν πολλαπλασιαστή P=1 επειδή η άσκηση ζητάει γινόμενο.

*Βήμα 3.* Αναζήτηση αριθμού ή δεδομένων προς επεξεργασία.

Εάν οι αριθμοί ήταν γνωστοί εκ των προτέρων χρησιμοποιώ την εντολή READ X-DATA διαφορετικά όπως στην περίπτωση της άσκησης αυτής την εντολή INPUT “Δώσε αριθμό :”;X

*Βήμα 4.* Επεξεργασία των δεδομένων χωρίς συνθήκη.

Επειδή θέλουμε άθροισμα θέτουμε  $S=S+X$

Επειδή θέλουμε γινόμενο θέτουμε  $P=P*X$

*Βήμα 5.* Αυξάνω τον μετρητή του πλήθους I κατά ένα ( $I=I+1$ )

*Βήμα 6.* Ελέγχω αν η διαδικασία έχει επαναληφθεί N φορές. Εάν όχι πηγαίνω στο βήμα 3 διαφορετικά συνεχίζω με το επόμενο βήμα.

IF  $I \leq N$  THEN GOTO “Βήμα 3”

*Βήμα 7.* Εκτυπώνω τα αποτελέσματα (το άθροισμα και το γινόμενο των αριθμών).

### Δ) Πρόγραμμα

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ & ΓΙΝΟΜΕΝΟΥ ΑΡΙΘΜΩΝ
20 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ Χ=ΑΡΙΘΜΟΣ
30 REM S=ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ P=ΓΙΝΟΜΕΝΟ ΑΡΙΘΜΩΝ
40 REM ΒΗΜΑ 1. ΔΙΑΒΑΖΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
50 INPUT "ΔΩΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ=";N
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
70 I=1
80 S=0
90 P=1
100 REM ΒΗΜΑ 3. ΑΝΑΖΗΤΗΣΗ ΑΡΙΘΜΟΥ
110 INPUT "ΔΩΣΕ ΑΡΙΘΜΟ =";X
120 REM ΒΗΜΑ 4. ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ
130 S=S+X
140 P=P*X
150 ΒΗΜΑ 5. ΑΥΞΗΣΗ ΤΟΥ ΜΕΤΡΗΤΗ ΠΛΗΘΟΥΣ ΚΑΤΑ ΕΝΑ
160 I=I+1
170 REM ΒΗΜΑ 6. ΕΛΕΓΧΟΣ ΑΝ ΕΧΟΥΝ ΤΕΛΕΙΩΣΕΙ ΟΙ ΑΡΙΘΜΟΙ
180 IF I<=N THEN GOTO 100
190 REM ΒΗΜΑ 7. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
200 PRINT "ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ =";S
210 PRINT "ΓΙΝΟΜΕΝΟ ΑΡΙΘΜΩΝ=";P
220 END
```

Η παραπάνω άσκηση θα μπορούσε να δοθεί με την εξής μορφή :

**Παράδειγμα.** Δίδονται οι παρακάτω 10 αριθμοί :

45, 23, 4, 34, 67, 89, 12, 26, 90, 12

Να βρεθεί το άθροισμα και το γινόμενο τους.

Η μόνη διαφορά με την παραπάνω άσκηση είναι ότι επειδή το πλήθος τους είναι γνωστό θα διαβαστεί με την εντολή LET (δες εντολή 50) και επειδή οι αριθμοί είναι επίσης γνωστοί θα διαβαστούν με την εντολή READ-DATA (δες εντολές 110,220)



**Πρόγραμμα**

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ & ΓΙΝΟΜΕΝΟΥ ΑΡΙΘΜΩΝ
20 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ Χ=ΑΡΙΘΜΟΣ
30 REM S=ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ Ρ=ΓΙΝΟΜΕΝΟ ΑΡΙΘΜΩΝ
40 REM ΒΗΜΑ 1. ΔΙΑΒΑΖΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
50 LET N=10
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
70 I=1
80 S=0
90 P=1
100 REM ΒΗΜΑ 3. ΑΝΑΖΗΤΗΣΗ ΑΡΙΘΜΟΥ
110 READ X
120 REM ΒΗΜΑ 4. ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ
130 S=S+X
140 P=P*X
150 ΒΗΜΑ 5. ΑΥΞΗΣΗ ΤΟΥ ΜΕΤΡΗΤΗ ΠΛΗΘΟΥΣ ΚΑΤΑ ΕΝΑ
160 I=I+1
170 REM ΒΗΜΑ 6. ΕΛΕΓΧΟΣ ΑΝ ΕΧΟΥΝ ΤΕΛΕΙΩΣΕΙ ΟΙ ΑΡΙΘΜΟΙ
180 IF I<=N THEN GOTO 100
190 REM ΒΗΜΑ 7. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
200 PRINT "ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ =";S
210 PRINT "ΓΙΝΟΜΕΝΟ ΑΡΙΘΜΩΝ=";P
220 DATA 45, 23, 4, 34, 67, 89, 12, 26, 90, 12
230 END

```

**Παράδειγμα. (Με συνθήκη)**

Δίνονται N αριθμοί. Να υπολογιστεί

- α) το άθροισμα των αριθμών που είναι μέσα στο κλειστό διάστημα  $[-10, 10]$ ,
- β) το γινόμενο των αριθμών που είναι εκτός του διαστήματος  $[-1, 1]$ ,
- γ) το πλήθος των αριθμών που είναι άρτιοι.

**A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Οι N αριθμοί.

Έξοδος – Το άθροισμα τους και το γινόμενο και το πλήθος των αριθμών που ικανοποιούν αντίστοιχα τις συνθήκες (α,β,γ).

### **Β) Παράδειγμα με το χέρι.**

Εάν έχω 5 αριθμούς : -20, 3, 7, 0.5, 1. Τότε παίρνω τον πρώτο αριθμό το -20, α) επειδή βρίσκεται εκτός του διαστήματος [-10,10] δεν τον αθροίζω, β) επειδή βρίσκεται εκτός του [-1,1] τον πολλαπλασιάζω με το 1 και έχω -20 και γ) επειδή είναι άρτιος θέτω ως πλήθος των αρτίων την 1. Παίρνω τον δεύτερο αριθμό τον 3, α) επειδή βρίσκεται εντός του διαστήματος [-10,10] τον αθροίζω και έχω ως τώρα άθροισμα 3, β) επειδή βρίσκεται εκτός του [-1,1] τον πολλαπλασιάζω με το -20 που είχα από πριν και έχω  $-20*3=-60$  και γ) επειδή δεν είναι άρτιος το πλήθος των αρτίων παραμένει 1 κ.ο.κ. Παρατηρώ δηλ. ότι για κάθε αριθμό κάνω τρεις ελέγχους σύμφωνα με τις συνθήκες (α,β,γ) και αν η (α) συνθήκη ικανοποιείται προσθέτω σε έναν αρχικό μετρητή (με αρχική τιμή 0) που είχα τον αριθμό αυτό, αν η (β) συνθήκη ικανοποιείται πολλαπλασιάζω σε έναν πολλαπλασιαστή (με αρχική τιμή 1) τον αριθμό αυτό και τέλος αν η συνθήκη (γ) ικανοποιείται προσθέτω την μονάδα σε έναν μετρητή (με αρχική τιμή το 0). Η διαδικασία αυτή επαναλαμβάνεται για όλους τους αριθμούς.

### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών. Εφόσον η άσκηση δεν προσδιόρισε τον αριθμό αυτό θα χρησιμοποιήσω την εντολή INPUT (π.χ. INPUT “Δώσε πλήθος :”;N).

*Βήμα 2.* Αρχικοποίηση των παραμέτρων.

Θέτω  $I=1$  για να δηλώσω ότι ακολουθεί ο πρώτος αριθμός που θα διαβάσω. Διαβάζω καλά την άσκηση και σημειώνω τις λέξεις κλειδιά, από αυτά που μου ζητάει. Δηλαδή στο συγκεκριμένο πρόβλημα σημειώνω την λέξη άθροισμα στο A, γινόμενο στο B και πλήθος στο Γ ζήτημα αντίστοιχα. Οι λέξεις αυτές θα με βοηθήσουν να επιλέξω το είδος των παραμέτρων που θα χρησιμοποιήσω εκτός και εντός της ανακύκλωσης. Στο πρόβλημα λοιπόν αυτό θα θέσω έναν αθροιστή  $S=0$  επειδή ζητάει η άσκηση άθροισμα, έναν πολλαπλασιαστή  $P=1$  επειδή η άσκηση ζητάει γινόμενο και έναν μετρητή  $PL=0$  επειδή η άσκηση ζητάει και πλήθος.

*Βήμα 3.* Αναζήτηση αριθμού ή δεδομένων προς επεξεργασία.

Επειδή οι αριθμοί δεν είναι γνωστοί εκ των προτέρων θα χρησιμοποιήσουμε την εντολή INPUT “Δώσε αριθμό :”;X

*Βήμα 4.* Επεξεργασία των δεδομένων με συνθήκη.

Επειδή θέλουμε άθροισμα με την συνθήκη το X να ανήκει στο [-10,10] χρησιμοποιούμε την εντολή :

```
IF X<=10 AND X>=-10 THEN S=S+X
```

Επειδή θέλουμε γινόμενο με την συνθήκη το X να μην ανήκει στο [-1,1] χρησιμοποιούμε την εντολή :

```
IF X>1 OR X<-1 THEN P=P*X
```

Επειδή θέλουμε πλήθος με την συνθήκη το X να είναι άρτιος χρησιμοποιούμε την εντολή :

```
IF X MOD 2=0 THEN PL=PL+1
```

*Βήμα 5.* Αυξάνω τον μετρητή του πλήθους I κατά ένα ( $I=I+1$ )

*Βήμα 6.* Ελέγχω αν η διαδικασία έχει επαναληφθεί N φορές. Εάν όχι πηγαίνω στο βήμα 3 διαφορετικά συνεχίζω με το επόμενο βήμα.

```
IF I<=N THEN GOTO “Βήμα 3”
```

*Βήμα 7.* Εκτυπώνω τα αποτελέσματα (το άθροισμα, το γινόμενο και το πλήθος των αριθμών).

#### **Δ) Πρόγραμμα**

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜ., ΓΙΝΟΜΕΝΟΥ & ΠΛΗΘΟΥΣ ΑΡΙΘΜΩΝ
20 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ X=ΑΡΙΘΜΟΣ
30 REM S=ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ P=ΓΙΝΟΜΕΝΟ ΑΡΙΘΜΩΝ
40 REM PL=ΠΛΗΘΟΣ ΑΡΤΙΩΝ ΑΡΙΘΜΩΝ
50 REM ΒΗΜΑ 1. ΔΙΑΒΑΖΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
60 INPUT “ΔΩΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ=”;N
70 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
80 I=1
90 S=0
100 P=1
110 PL=0
120 REM ΒΗΜΑ 3. ΑΝΑΖΗΤΗΣΗ ΑΡΙΘΜΟΥ
130 INPUT “ΔΩΣΕ ΑΡΙΘΜΟ =”;X
```

```
140 REM ΒΗΜΑ 4. ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ
150 IF X>=-10 AND X<=10 THEN S=S+X
160 IF X>1 OR X<-1 THEN P=P*X
170 IF X MOD 2=0 THEN PL=PL+1
180 ΒΗΜΑ 5. ΑΥΞΗΣΗ ΤΟΥ ΜΕΤΡΗΤΗ ΠΛΗΘΟΥΣ ΚΑΤΑ ΕΝΑ
190 I=I+1
200 REM ΒΗΜΑ 6. ΕΛΕΓΧΟΣ ΑΝ ΕΧΟΥΝ ΤΕΛΕΙΩΣΕΙ ΟΙ ΑΡΙΘΜΟΙ
210 IF I<=N THEN GOTO 120
220 REM ΒΗΜΑ 7. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
230 PRINT "ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ ΣΤΟ [-10,10] =";S
240 PRINT "ΓΙΝΟΜΕΝΟ ΑΡΙΘΜΩΝ ΕΚΤΟΣ ΤΟΥ [-1,1] =";P
250 PRINT "ΠΛΗΘΟΣ ΑΡΤΙΩΝ ΑΡΙΘΜΩΝ =";PL
260 END
```

Τα παραπάνω παραδείγματα μπορούν να γίνουν πιο ρεαλιστικά όπως φαίνεται παρακάτω. Παρ' όλα αυτά η επίλυση των προβλημάτων παραμένει η ίδια.

**Παράδειγμα.** Η κατανομή της έκτασης των δασών της χώρας μας φαίνεται στον παρακάτω πίνακα :

Μορφές εδαφοπονίας	Έκταση στρεμ.
1. Δάση	25.124.180
2. Μερικώς δασοσκεπείς εκτάσεις	32.421.400
3. Φρυγανότοποι	2.773.135
4. Αλπικές εκτάσεις	4.400.577
5. Χορτολίβαδα	17.555.073
6. Έλη-Λίμνες-Ποταμοί	2.728.620
7. Άγωνα	7.348.513
8. Γεωργικές καλλιέργειες	39.638.500

Να γραφεί πρόγραμμα που να υπολογίζει την συνολική έκταση των δασών στην χώρα μας.

**Απάντηση.**

Παρατηρούμε ότι η εκφώνηση του παραπάνω προβλήματος μπορεί κάλλιστα να διαμορφωθεί ως εξής :

«Δίνονται οι εξής 8 αριθμοί : 25.124.180, 32.421.400, 2.773.135, 4.400.577, 17.555.073, 2.728.620, 2.728.620, 7.348.513, 39.638.500. Να βρεθεί το γινόμενο τους.»

οπότε και η επίλυση της σύμφωνα με προηγούμενο παράδειγμα θα είναι η εξής :

### **Πρόγραμμα**

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ ΑΡΙΘΜΩΝ
20 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ Χ=ΑΡΙΘΜΟΣ
30 REM S=ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ
40 REM ΒΗΜΑ 1. ΔΙΑΒΑΖΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
50 LET N=8
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
70 I=1
80 S=0
90 REM ΒΗΜΑ 3. ΑΝΑΖΗΤΗΣΗ ΑΡΙΘΜΟΥ
100 READ X
110 REM ΒΗΜΑ 4. ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ
120 S=S+X
130 ΒΗΜΑ 5. ΑΥΞΗΣΗ ΤΟΥ ΜΕΤΡΗΤΗ ΠΛΗΘΟΥΣ ΚΑΤΑ ΕΝΑ
140 I=I+1
150 REM ΒΗΜΑ 6. ΕΛΕΓΧΟΣ ΑΝ ΕΧΟΥΝ ΤΕΛΕΙΩΣΕΙ ΟΙ ΑΡΙΘΜΟΙ
160 IF I<=N THEN GOTO 90
170 REM ΒΗΜΑ 7. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
180 PRINT "ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ =" ;S
190 DATA 25124180, 32421400, 2773135, 4400577
210 DATA 17555073, 2728620, 2728620, 7348513, 39638500
220 END

```

Παρατηρήστε ότι δεν τοποθετούμε (.) στον διαχωρισμό χιλιάδων και εκατομμυρίων.

Κατά την εκτέλεση του προγράμματος θα πάρουμε ως αποτέλεσμα 131.990.000 στρέμματα που αποτελεί το 19% της συνολικής έκτασης της

χώρας μας, ποσοστό σχετικά φτωχό σε σχέση με άλλα δάση σε χώρες της Ευρώπης.

**Παράδειγμα.** Έγινε μια δειγματοληπτική έρευνα σε ένα δάσος Ελάτης. Το θέμα της έρευνας ήταν να μετρήσουμε τις διαμέτρους 20 δέντρων ελάτης και να βρούμε το πλήθος εκείνων :

A) που η διάμετρος τους ανήκει στο διάστημα [10-21εκ.]

B) που η διάμετρος τους ανήκει στο διάστημα ]21εκ.-35εκ.]

Γ) που η διάμετρος τους ανήκει στο διάστημα ]35-& άνω]

Έστω ότι τα αποτελέσματα της έρευνας έδειξαν τις παρακάτω διαμέτρους :

15, 25, 22, 40, 34, 12, 13, 34, 40, 29, 19, 38, 46, 24, 27, 29, 40, 19, 17, 30

Να γραφεί πρόγραμμα που να επιλύει τα ζητήματα (α,β,γ).

### Απάντηση

Το παραπάνω πρόβλημα ανάγεται στα παραπάνω προβλήματα ανακύκλωσης με συνθήκη. Επειδή χρειάζομαι να μετρήσω το πλήθος των δέντρων που ικανοποιούν μια από τις συνθήκες (α,β,γ) θα χρησιμοποιήσουμε τρεις μετρητές έστω PL1,PL2 και PL3 οι οποίες αρχικά θα έχουν την τιμή 0, ενώ θα αυξάνονται κατά ένα στην διάρκεια της ανακύκλωσης εάν κάποια από τις συνθήκες τις οποίες αντιπροσωπεύουν ικανοποιείται.

### Πρόγραμμα

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜΑΤΟΣ & ΓΙΝΟΜΕΝΟΥ ΑΡΙΘΜΩΝ
```

```
20 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ Χ=ΑΡΙΘΜΟΣ
```

```
30 REM PL1,PL2,PL3=ΜΕΤΡΗΤΕΣ ΠΛΗΘΟΥΣ ΓΙΑ Α,Β,Γ
```

```
40 REM ΒΗΜΑ 1. ΔΙΑΒΑΖΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
```

```
50 LET N=20
```

```
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
```

```
70 I=1
```

```
80 PL1=0 : PL2=0 : PL3=0
```

```
90 REM ΒΗΜΑ 3. ΑΝΑΖΗΤΗΣΗ ΑΡΙΘΜΟΥ
```

```
100 READ X
```

```
110 REM ΒΗΜΑ 4. ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ
```

```
120 IF X>=10 AND X<=21 THEN PL1=I+1
```

```
130 IF X>21 AND X<=35 THEN PL2=PL2+1
```

```

140 IF X>35 THEN PL3=PL3+1
150 ΒΗΜΑ 5. ΑΥΞΗΣΗ ΤΟΥ ΜΕΤΡΗΤΗ ΠΛΗΘΟΥΣ ΚΑΤΑ ΕΝΑ
160 I=I+1
170 REM ΒΗΜΑ 6. ΕΛΕΓΧΟΣ ΑΝ ΕΧΟΥΝ ΤΕΛΕΙΩΣΕΙ ΟΙ ΑΡΙΘΜΟΙ
180 IF I<=N THEN GOTO 90
190 REM ΒΗΜΑ 7. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
200 PRINT "ΠΛΗΘΟΣ ΔΕΝΤΡΩΝ ΜΕ ΔΙΑΜΕΤΡΟ ΜΕΤΑΞΥ 10 ΚΑΙ 21 =";PL1
210 PRINT "ΠΛΗΘΟΣ ΔΕΝΤΡΩΝ ΜΕ ΔΙΑΜΕΤΡΟ ΜΕΤΑΞΥ 21 ΚΑΙ 35 =";PL2
220 PRINT "ΠΛΗΘΟΣ ΔΕΝΤΡΩΝ ΜΕ ΔΙΑΜΕΤΡΟ ΜΕΓΑΛΥΤ. ΑΠΟ 35 =";PL3
230 DATA 15, 25, 22, 40, 34, 12, 13, 34, 40, 29
240 DATA , 19, 38, 46, 24, 27, 29, 40, 19, 17, 30
250 END

```

## Επαναληπτικές ασκήσεις στην ανακύκλωση και χρήση μετρητών.

**Άσκηση 1.** Η κατανομή των ενεργών δασικών συνεταιρισμών σε γεωγραφικά διαμερίσματα (1982) δίδεται από τον παρακάτω πίνακα :

A/A	Γεωγραφικά Διαμερίσματα	Αριθμός Δασικών Συν/μων
1	ΘΡΑΚΗ	33
2	ΑΝΑΤΟΛΙΚΗ ΜΑΚΕΔΟΝΙΑ	68
3	ΚΕΝΤΡΙΚΗ ΜΑΚΕΔΟΝΙΑ	62
4	ΔΥΤΙΚΗ ΜΑΚΕΔΟΝΙΑ	35
5	ΗΠΕΙΡΟΣ	60
6	ΘΕΣΣΑΛΙΑ	87
7	ΣΤΕΡΕΑ ΕΛΛΑΔΑ	91
8	ΠΕΛΟΠΟΝΝΗΣΟΣ	10
9	ΝΗΣΙΑ	0

Να γραφεί πρόγραμμα που να υπολογίζει τον συνολικό αριθμό των δασικών συνεταιρισμών στην Ελλάδα.

**Άσκηση 2.** Στον παρακάτω πίνακα παρατηρούμε τα μέγιστα ύψη βροχής που σημειώθηκαν ανά υδρολογικό έτος στην Βρετανία :

Υδρολογικό Έτος	Ύψη σε mm
1960	22.35
1961	35.56
1962	40.64
1963	54.10
1964	48.77
1965	41.91
1966	55.88
1967	49.53
1968	65.53
1969	33.53

Να γράψετε πρόγραμμα που θα υπολογίζει :

- την μέση μέγιστη βροχόπτωση (το άθροισμα όλων των παραπάνω δια 10),
- το πλήθος των υδρολογικών ετών που η μέγιστη βροχόπτωση ήταν στο διάστημα [45mm-55mm]

### Εντολές Επανάληψεων

Με τις εντολές αυτές μπορούμε να επαναλάβουμε μια σειρά εντολών στην BASIC παραπάνω από μια φορές. Οι εντολές αυτές είναι δύο και όπως θα δούμε στη συνέχεια α) η πρώτη (FOR-NEXT) χρησιμοποιείται για την επανάληψη μιας διαδικασίας για συγκεκριμένο αριθμό φορών ενώ β) η δεύτερη (WHILE-WEND) όταν δεν γνωρίζουμε το πλήθος των επαναλήψεων αλλά μια συνθήκη η οποία θα σταματήσει την επανάληψη της διαδικασίας. Για να γίνουμε πιο σαφής θεωρείστε το παρακάτω παράδειγμα :

**Παράδειγμα.** Να γραφεί πρόγραμμα που να τυπώνει την λέξη ΝΙΚΟΣ 20 φορές.

**Απάντηση 1<sup>η</sup>.** Μπορούμε να γράψουμε 100 φορές την παρακάτω εντολή :



### PRINT “ΝΙΚΟΣ”

Το μειονέκτημα του παραπάνω προγράμματος θα ήταν το μεγάλο πλήθος εντολών. Θα ήταν πολύ μεγαλύτερο το μειονέκτημα να θέλαμε να τυπώσουμε το παραπάνω όνομα όχι 100 αλλά 10000 φορές.

**Απάντηση 2<sup>η</sup>.** Μπορούμε να γράψουμε το παρακάτω πρόγραμμα :

```
10 PRINT “ΝΙΚΟΣ”
20 GOTO 20
```

Μειονέκτημα του παραπάνω προγράμματος είναι ότι θα τυπώσει το παραπάνω όνομα άπειρες φορές. Το συγκεκριμένο πρόγραμμα δεν θα σταματήσει παρά μόνο αν πατήσω τον συνδυασμό πλήκτρων Ctrl και C.

**Απάντηση 3<sup>η</sup>.** Μπορώ να χρησιμοποιήσουμε τους μετρητές που αναφέραμε στο προηγούμενο κεφάλαιο.

```
10 N=100
20 I=1
30 PRINT “ΝΙΚΟΣ”
40 I=I+1
50 IF I<=N THEN GOTO 30
60 END
```

Το παραπάνω πρόγραμμα θέτει ως *αρχική τιμή* του I το 1, ενώ κάθε φορά που εκτυπώνει το «ΝΙΚΟΣ» αυξάνετε το I με *βήμα* 1 και πάντοτε ελέγχει αν το I έχει ξεπεράσει την *τελική τιμή* που θέλω να είναι το N.

Συμπεραίνουμε λοιπόν από το παραπάνω παράδειγμα αλλά και από το προηγούμενο κεφάλαιο ότι θα ήταν σκόπιμο να υπάρχει μια εντολή η οποία θα θέτει αρχική τιμή σε μια μεταβλητή, στην συνέχεια να αφήνει να μεσολαβήσει κάποια διαδικασία, έπειτα να αυξάνει την τιμή της μεταβλητής κατά κάποιο βήμα και να ελέγχει αν η τιμή της μεταβλητής αυτής έχει ξεπεράσει κάποια τελική τιμή που έχουμε θέσει. Η εντολή αυτή είναι η FOR-NEXT που αναλύουμε παρακάτω :

## Η εντολή FOR-NEXT

**Σκοπός :** Η επανάληψη ενός πλήθους εντολών έναν συγκεκριμένο αριθμό φορών που καθορίζεται από τις παραμέτρους της εντολής.

**Σύνταξη :**

```
XX FOR I=A TO T STEP B
```

Διάφορες εντολές της BASIC

```
XX NEXT I
```

Που σημαίνει ότι το I παίρνει αρχική τιμή A, στην συνέχεια εκτελούνται οι εντολές της BASIC που ακολουθούν, η εντολή NEXT I αυξάνει την τιμή του I κατά το βήμα δηλ.  $I=I+A$ , στην συνέχεια η ροή του προγράμματος πηγαίνει στην εντολή FOR όπου και γίνεται έλεγχος αν το I έχει ξεπεράσει την τελική του τιμή που είναι το B. Αν δεν την έχει ξεπεράσει τότε εκτελούνται οι εντολές της BASIC μετά την FOR κ.ο.κ ενώ διαφορετικά εκτελούνται οι εντολές μετά την NEXT.

Το A λέγεται αρχική τιμή της μεταβλητής I.

Το T λέγεται τελική τιμή της μεταβλητής I.

Το N λέγεται βήμα της μεταβολής των τιμών της μεταβλητής I.

Το A,T,B μπορούν να είναι αριθμητικές σταθερές, μεταβλητές ή παραστάσεις.

**Παράδειγμα.** Να εκτυπωθούν οι αριθμοί 1,2,3,... 100.

**Απάντηση.**

Επειδή θέλω να επαναληφθεί μια διαδικασία 100 φορές (συγκεκριμένος αριθμός φορών) θα χρησιμοποιήσω την εντολή FOR-NEXT. Παρατηρώ ότι θέλω να εκτυπώνω την μεταβλητή I που έχει ως αρχική τιμή 1, τελική τιμή 100 και βήμα 1, άρα το πρόγραμμα που θα χρησιμοποιήσω θα είναι το εξής :

```
10 FOR I=1 TO 100 STEP 1
```

```
20 PRINT I
```

```
30 NEXT I
```

```
40 END
```

Η διαδικασία ροής του προγράμματος και το περιεχόμενο της μνήμης φαίνεται παρακάτω :

Εντολή	Εμφάνιση στην οθόνη	Τιμή της μεταβλητής I στην μνήμη
10		1
20	1	1
30	1	2
10	1	2 (<=100)
20	2	2
30	2	3
10	2	3 (<=100)
...	...	...
20	99	99
30	99	100
10	99	100 (<=100)
20	100	100
30	100	110
10	100	110 (>100)
40	100	110

- Στην περίπτωση που το βήμα είναι 1 το [STEP 1] μπορεί να παραληφθεί και αντίστροφα αν το [STEP B] παραλείπεται θεωρείτε ότι το βήμα είναι 1.

**Παράδειγμα.** Να εκτυπωθούν οι αριθμοί 100,99,98,... 1.

**Απάντηση.** Παρατηρώ τώρα ότι θέλω να ξεκινήσω με αρχική τιμή 100 και να φτάσω ως το 1. Συνεπώς το βήμα θα πρέπει να είναι αρνητικό και πιο συγκεκριμένα  $-1$ . Άρα το πρόγραμμα που υλοποιεί αυτό που μου ζητάει η άσκηση θα είναι το εξής :

```
10 FOR I=100 TO 1 STEP -1
20 PRINT I
30 NEXT I
40 END
```

### Παρατηρήσεις.

- Το βήμα θα πρέπει πάντα να είναι διάφορο του μηδενός.
- Αν η αρχική τιμή A είναι μικρότερη από την τελική τιμή T τότε το βήμα B θα πρέπει να είναι θετικό.  
Αν η αρχική τιμή A είναι μεγαλύτερη από την τελική τιμή T τότε το βήμα B θα πρέπει να είναι αρνητικό.
- Η μεταβλητή I της FOR-NEXT δεν πρέπει να αλλάζει τιμή μέσα στο loop ή αλλιώς να βρίσκεται αριστερά του (=). Είναι λάθος για παράδειγμα το παρακάτω πρόγραμμα :

```

10 FOR I=1 TO 10
20 I=I+1
30 NEXT I
    
```

- Μπορούμε να βγούμε μέσα από μια FOR-NEXT με την εντολή GOTO ή με τον συνδυασμό των εντολών IF-THEN-ELSE και GOTO, αλλά δεν μπορούμε να μπούμε σε μια FOR-NEXT χωρίς αναγκαστικά να περάσουμε από την εντολή FOR.

ΣΩΣΤΟ	ΛΑΘΟΣ
<pre> 10 FOR I=1 TO 3 20 INPUT "ΚΩΔΙΚΟΣ=",KEY 30 IF KEY=999 THEN GOTO 60 40 NEXT I 50 STOP 60 PRINT "ΕΠΙΤΥΧΗΣ ΠΡΟΣΒΑΣΗ" 70 END                     </pre>	<pre> 10 GOTO 30 20 FOR I=1 TO 10 30 PRINT I 40 NEXT I 50 END                     </pre>
<p>Το παραπάνω πρόγραμμα σας ζητάει να του δώσετε ένα νούμερο. Αν αυτό είναι το 999 σας απαντάει με το μήνυμα «ΕΠΙΤΥΧΗΣ ΠΡΟΣΒΑΣΗ» διαφορετικά αν δεν απαντήσετε σωστά σε τρεις προσπάθειες το πρόγραμμα σταματάει απότομα. Βλέπουμε λοιπόν ότι μπορούμε να βγούμε από το loop προτού ολοκληρωθούν και οι τρεις επαναλήψεις.</p>	<p>Είναι λάθος να μπούμε μέσα σε μια FOR-NEXT με μια GOTO.</p>

- Το παρακάτω loop χρησιμοποιείται για να καθυστερήσει το πρόγραμμά μας και για αυτό λέγεται βρόγχος καθυστέρησης :

```
FOR I=1 TO 1000 : NEXT I
```

Χρησιμοποιείται όταν θέλω να ακινητοποιήσω για λίγο τα αποτελέσματα που έχουν εμφανισθεί στην οθόνη προτού εμφανισθούν κάποια άλλα μηνύματα.

- Το πρόγραμμα που βρίσκεται μεταξύ της FOR-NEXT είναι αισθητικά και προγραμματιστικά καλύτερο όταν τοποθετείται μια παράγραφος πιο μέσα, όπως φαίνεται στα παραπάνω παραδείγματα. Αυτό γίνεται για να ξεχωρίζει αυτός που διαβάζει το πρόγραμμα πιο κομμάτι του προγράμματος επαναλαμβάνεται στην FOR-NEXT.
- Ως όνομα μεταβλητής στην FOR μπορούμε να χρησιμοποιήσουμε όποιο θέλουμε, αλλά θα πρέπει να έχουμε το ίδιο όνομα και στην NEXT.

**ΛΑΘΟΣ**

```
10 FOR J=1 to 10
20 PRINT J
30 NEXT K
```

- Ο αριθμός των επαναλήψεων μιας FOR-NEXT δίνεται από τον τύπο :

$$\left[ \frac{T - A}{B} \right] + 1$$

όπου T=Τελική τιμή, A=Αρχική τιμή, B=Βήμα και [x] δηλώνει το ακέραιο μέρος του x π.χ.

10 FOR N=5 TO 15 STEP 4 : ..... : NEXT N

Θα εκτελεστεί :

$$\left[ \frac{15 - 5}{4} \right] + 1 = [2.5] + 1 = 2 + 1 = 3 \text{ φορές}$$

**Παράδειγμα.** Να διαβαστούν τα ονοματεπώνυμα N φοιτητών και οι βαθμοί τους σε 4 μαθήματα. Στην συνέχεια να τυπωθεί το ονοματεπώνυμο τους και ο μέσος όρος στα 4 αυτά μαθήματα.

**Απάντηση.**

**A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Ο αριθμός των φοιτητών έστω N. Τα ονοματεπώνυμα που θα καταχωρηθούν στην μεταβλητή ONOMA\$ και οι βαθμοί στα 4 μαθήματα που θα καταχωρηθούν στις μεταβλητές B1,B2,B3 και B4 αντίστοιχα.

Έξοδος – Ο μέσος όρος κάθε φοιτητή που θα καταχωρηθεί στην μεταβλητή ΜΟ.

### **Β) Παράδειγμα με το χέρι.**

Παίρνω τα στοιχεία του πρώτου φοιτητή και υπολογίζω τον μέσο όρο και εκτυπώνω το ονοματεπώνυμο και τον μέσο όρο. Επαναλαμβάνω την διαδικασία αυτή Ν φορές, όπου το Ν μου το έχουν δηλώσει στην αρχή.

### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζω το πλήθος των φοιτητών. Εφόσον η άσκηση δεν προσδιόρισε τον αριθμό αυτό θα χρησιμοποιήσω την εντολή INPUT (π.χ. INPUT “Δώσε πλήθος :”;N).

*Βήμα 2.* Η παρακάτω διαδικασία επαναλαμβάνεται Ν φορές.

*Βήμα 2.1* Αναζήτηση στοιχείων του φοιτητή.

*Βήμα 2.2* Υπολογισμός μέσου όρου.

*Βήμα 2.3* Εκτύπωση των αποτελεσμάτων.

### **Δ) Πρόγραμμα**

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ ΒΑΘΜΩΝ ΦΟΙΤΗΤΩΝ
20 REM ΟΝΟΜΑ$=ΟΝΟΜ/ΜΟ, Β1,Β2,Β3,Β4=ΒΑΘΜΟΙ ΣΕ 4 ΜΑΘΗΜΑΤΑ
30 REM ΜΟ=ΜΕΣΟΣ ΟΡΟΣ, Ν=ΠΛΗΘΟΣ ΦΟΙΤΗΤΩΝ
40 REM ΒΗΜΑ 1. ΥΠΟΛΟΓΙΣΜΟΣ ΠΛΗΘΟΥΣ ΦΟΙΤΗΤΩΝ
50 INPUT “ΔΩΣΕ ΠΛΗΘΟΣ ΦΟΙΤΗΤΩΝ=”,N
60 REM ΒΗΜΑ 2. ΕΠΑΝΑΛΗΨΗ Ν ΦΟΡΩΝ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ
70 FOR I=1 TO N
80     REM ΒΗΜΑ 2.1 ΑΝΑΖΗΤΗΣΗ ΣΤΟΙΧΕΙΩΝ ΦΟΙΤΗΤΗ
90     INPUT “ΟΝΟΜΑΤΕΠΩΝΥΜΟ=”,ΟΝΟΜΑ$
100    INPUT “ΒΑΘΜΟΙ (Β1,Β2,Β3,Β4) =”,Β1,Β2,Β3,Β4
110    REM ΒΗΜΑ 2.2 ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ
120    ΜΟ=(Β1+Β2+Β3+Β4)/4
130    REM ΒΗΜΑ 2.3 ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
140    PRINT ΟΝΟΜΑ$,ΜΟ
150 NEXT I
160 END
```

**Παράδειγμα.** Δίνονται  $N$  αριθμοί. Να υπολογιστεί :

- α) το άθροισμα των αριθμών που ανήκουν στο διάστημα  $[-20,20]$ ,
- β) το γινόμενο των αριθμών που είναι πολλαπλάσια του 3,
- γ) το πλήθος των μηδενικών αριθμών.

**Απάντηση**

### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών  $N$  καθώς και οι αριθμοί που θα δίνονται μέσω της μεταβλητής  $X$ .

*Εξοδος* – Το άθροισμα τους και το γινόμενο και το πλήθος των αριθμών που ικανοποιούν αντίστοιχα τις συνθήκες  $(\alpha, \beta, \gamma)$ .

### **B) Παράδειγμα με το χέρι.**

Εάν έχω 5 αριθμούς : -25, 3, 7, 0.5, 1. Τότε παίρνω τον πρώτο αριθμό το -20, α) επειδή βρίσκεται εκτός του διαστήματος  $[-20,20]$  δεν τον αθροίζω, β) επειδή δεν είναι πολλαπλάσιο του 3 δεν τον πολλαπλασιάζω με το 1 και γ) επειδή δεν είναι μηδέν δεν αυξάνω το πλήθος των μηδενικών αριθμών κατά ένα 1. Συνεχίζω την ίδια διαδικασία με τους υπόλοιπους 4 αριθμούς. Βλέπω λοιπόν ότι η ίδια διαδικασία επαναλαμβάνεται 5 φορές και συνεπώς η εντολή FOR-NEXT είναι ιδανική για την επανάληψη της διαδικασίας αυτής.

### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών. Εφόσον η άσκηση δεν προσδιόρισε τον αριθμό αυτό θα χρησιμοποιήσω την εντολή INPUT (π.χ. INPUT “Δώσε πλήθος :”;N).

*Βήμα 2.* Αρχικοποίηση των παραμέτρων.

Διαβάζω καλά την άσκηση και σημειώνω τις λέξεις κλειδιά, από αυτά που μου ζητάει. Δηλαδή στο συγκεκριμένο πρόβλημα σημειώνω την λέξη άθροισμα στο A, γινόμενο στο B και πλήθος στο Γ ζήτημα αντίστοιχα. Οι λέξεις αυτές θα με βοηθήσουν να επιλέξω το είδος των παραμέτρων που θα χρησιμοποιήσω εκτός και εντός της ανακύκλωσης. Στο πρόβλημα λοιπόν αυτό θα θέσω έναν αθροιστή  $S=0$  επειδή ζητάει η

άσκηση άθροισμα, έναν πολλαπλασιαστή  $P=1$  επειδή η άσκηση ζητάει γινόμενο και έναν μετρητή  $PL=0$  επειδή η άσκηση ζητάει και πλήθος.

*Βήμα 3.* Επανάληψη της παρακάτω διαδικασίας  $N$  φορές :

*Βήμα 3.1* Αναζήτηση αριθμού ή δεδομένων προς επεξεργασία.

Επειδή οι αριθμοί δεν είναι γνωστοί εκ των προτέρων θα χρησιμοποιήσουμε την εντολή INPUT "Δώσε αριθμό :";X

*Βήμα 3.2.* Επεξεργασία των δεδομένων με συνθήκη.

Επειδή θέλουμε άθροισμα με την συνθήκη το  $X$  να ανήκει στο  $[-20,20]$  χρησιμοποιούμε την εντολή :

```
IF X<=-20 AND X>=-20 THEN S=S+X
```

Επειδή θέλουμε γινόμενο με την συνθήκη το  $X$  να είναι πολλαπλάσιο του 3 (το υπόλοιπο της διαίρεσης του με το 3 είναι μηδέν) χρησιμοποιούμε την εντολή :

```
IF X MOD 3=0 THEN P=P*X
```

Επειδή θέλουμε πλήθος με την συνθήκη το  $X$  να είναι μηδέν χρησιμοποιούμε την εντολή :

```
IF X=0 THEN PL=PL+1
```

*Βήμα 4.* Εκτυπώνω τα αποτελέσματα (το άθροισμα, το γινόμενο και το πλήθος των αριθμών).

#### **Δ) Πρόγραμμα**

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜ., ΓΙΝΟΜΕΝΟΥ & ΠΛΗΘΟΥΣ ΑΡΙΘΜΩΝ
20 REM S=ΑΘΡΟΙΣΤΗΣ, P=ΠΟΛΛΑΠΛΑΣΙΑΣΤΗΣ, PL=ΜΕΤΡ. ΠΛΗΘΟΥΣ
30 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
40 REM ΒΗΜΑ 1. ΑΝΑΖΗΤΗΣΗ ΠΛΗΘΟΥΣ ΑΡΙΘΜΩΝ
50 INPUT "ΔΩΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ=",N
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
70 S=0 : P=1 : PL=0
80 REM ΒΗΜΑ 3. ΕΠΑΝΑΛΗΨΗ ΤΗΣ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑΣ N ΦΟΡΕΣ
90 FOR I=1 TO N
100  ΒΗΜΑ 3.1 ΔΙΑΒΑΣΕ ΤΟΝ I ΑΡΙΘΜΟ
110  INPUT "ΔΩΣΕ ΑΡΙΘΜΟ=",X
120  ΒΗΜΑ 3.2 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΒΑΣΗ ΣΥΝΘΗΚΗΣ
130  IF X>=-20 AND X<=20 THEN S=S+X
```



```

140 IF X MOD 3=0 THEN P=P*X
150 IF X=0 THEN PL=PL+1
160 NEXT I
170 REM ΒΗΜΑ 4. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
180 PRINT "ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ ΣΤΟ [-20,20] =",S
190 PRINT "ΓΙΝΟΜΕΝΟ ΠΟΛΛΑΠΛΑΣΙΩΝ ΤΟΥ 3 =", P
200 PRINT "ΠΛΗΘΟΣ ΜΗΔΕΝΙΚΩΝ =",PL
210 END

```

**Παράδειγμα.** Δίνονται οι παρακάτω αριθμοί :

15, 25, 30, 40, 60, -20, -30, -12, 0, 5

Να υπολογιστεί :

- α) το άθροισμα των αριθμών που δεν ανήκουν στο διάστημα [-25,25],
- β) το γινόμενο των περιττών,
- γ) το πλήθος των άρτιων αριθμών.

### **Απάντηση**

#### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών  $N=10$  καθώς και οι αριθμοί που θα δίνονται μέσω της μεταβλητής  $X$ .

*Έξοδος* – Το άθροισμα τους και το γινόμενο και το πλήθος των αριθμών που ικανοποιούν αντίστοιχα τις συνθήκες (α,β,γ).

#### **B) Παράδειγμα με το χέρι.**

Παίρνω τον πρώτο αριθμό τον 15, α) επειδή βρίσκεται εντός του διαστήματος [-25,25] δεν τον αθροίζω, β) επειδή είναι περιττός τον πολλαπλασιάζω με το 1 και γ) επειδή δεν είναι άρτιος δεν αυξάνω το πλήθος των μηδενικών αριθμών κατά ένα 1. Συνεχίζω την ίδια διαδικασία με τους υπόλοιπους αριθμούς. Βλέπω λοιπόν ότι η ίδια διαδικασία επαναλαμβάνεται 10 φορές και συνεπώς η εντολή FOR-NEXT είναι ιδανική για την επανάληψη της διαδικασίας αυτής.

#### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Θέτω ως πλήθος των αριθμών  $N=10$ . Εφόσον η άσκηση προσδιόρισε τον αριθμό αυτό θα χρησιμοποιήσω την εντολή LET (π.χ. LET  $N=10$ ).

*Βήμα 2.* Αρχικοποίηση των παραμέτρων.

Διαβάζω καλά την άσκηση και σημειώνω τις λέξεις κλειδιά, από αυτά που μου ζητάει. Δηλαδή στο συγκεκριμένο πρόβλημα σημειώνω την λέξη άθροισμα στο A, γινόμενο στο B και πλήθος στο Γ ζήτημα αντίστοιχα. Οι λέξεις αυτές θα με βοηθήσουν να επιλέξω το είδος των παραμέτρων που θα χρησιμοποιήσω εκτός και εντός της ανακύκλωσης. Στο πρόβλημα λοιπόν αυτό θα θέσω έναν αθροιστή  $S=0$  επειδή ζητάει η άσκηση άθροισμα, έναν πολλαπλασιαστή  $P=1$  επειδή η άσκηση ζητάει γινόμενο και έναν μετρητή  $PL=0$  επειδή η άσκηση ζητάει και πλήθος.

*Βήμα 3.* Επανάληψη της παρακάτω διαδικασίας N (όπου N το έχω ορίσει παραπάνω) φορές :

*Βήμα 3.1* Αναζήτηση αριθμού ή δεδομένων προς επεξεργασία.

Επειδή οι αριθμοί είναι γνωστοί εκ των προτέρων και ταυτόχρονα είναι πολλοί θα χρησιμοποιήσουμε την εντολή READ-DATA

*Βήμα 3.2.* Επεξεργασία των δεδομένων με συνθήκη.

Επειδή θέλουμε άθροισμα με την συνθήκη το X να μην ανήκει στο [-25,25] χρησιμοποιούμε την εντολή :

```
IF X>25 OR X<-25 THEN S=S+X
```

Επειδή θέλουμε γινόμενο με την συνθήκη το X να είναι περιττός αριθμός (το υπόλοιπο της διαίρεσης του με το 2 είναι διάφορο του μηδέν) χρησιμοποιούμε την εντολή :

```
IF X MOD 2<>0 THEN P=P*X
```

Επειδή θέλουμε πλήθος με την συνθήκη το X να είναι άρτιος χρησιμοποιούμε την εντολή :

```
IF X MOD 2=0 THEN PL=PL+1
```

*Βήμα 4.* Εκτυπώνω τα αποτελέσματα (το άθροισμα, το γινόμενο και το πλήθος των αριθμών).

#### **Δ) Πρόγραμμα**

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡΟΙΣΜ., ΓΙΝΟΜΕΝΟΥ & ΠΛΗΘΟΥΣ ΑΡΙΘΜΩΝ  
20 REM S=ΑΘΡΟΙΣΤΗΣ, P=ΠΟΛΛΑΠΛΑΣΙΑΣΤΗΣ, PL=ΜΕΤΡ. ΠΛΗΘΟΥΣ
```

```

30 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
40 REM ΒΗΜΑ 1. ΘΕΤΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
50 LET N=10
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
70 S=0 : P=1 : PL=0
80 REM ΒΗΜΑ 3. ΕΠΑΝΑΛΗΨΗ ΤΗΣ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑΣ Ν ΦΟΡΕΣ
90 FOR I=1 TO N
100  ΒΗΜΑ 3.1 ΔΙΑΒΑΣΕ ΤΟΝ Ι ΑΡΙΘΜΟ
110  READ X
120  ΒΗΜΑ 3.2 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΒΑΣΗ ΣΥΝΘΗΚΗΣ
130  IF X>25 OR X<-25 THEN S=S+X
140  IF X MOD 2<>0 THEN P=P*X
150  IF X MOD 2=0 THEN PL=PL+1
160 NEXT I
170 REM ΒΗΜΑ 4. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
180 PRINT "ΑΘΡΟΙΣΜΑ ΑΡΙΘΜΩΝ ΕΚΤΟΣ ΤΟΥ [-25,25] =",S
190 PRINT "ΓΙΝΟΜΕΝΟ ΠΕΡΙΤΤΩΝ =", P
200 PRINT "ΠΛΗΘΟΣ ΑΡΤΙΩΝ =",PL
210 DATA 15, 25, 30, 40, 60, -20, -30, -12, 0, 5
220 END

```

**Παράδειγμα.** Δίνονται οι παρακάτω 30 θερμοκρασίες που σημειώθηκαν κατά τον μήνα Νοέμβριο του 1997 :

10, 9, 8, 4, 7, 10, 12, 14, 18, 19, 20, ....., 15

Να βρεθεί το πλήθος των θερμοκρασιών που ανήκουν στα διαστήματα [0-10], (10-20] και (20,40).

### Απάντηση

#### Α) Περιγραφή εισόδου-εξόδου

*Είσοδος* – Το πλήθος των αριθμών  $N=30$  καθώς και οι αριθμοί που θα δίνονται μέσω της μεταβλητής  $X$ .

*Έξοδος* – Το πλήθος των αριθμών που ανήκουν αντίστοιχα στα διαστήματα [0-10], (10-20] και (20,40).

## **B) Παράδειγμα με το χέρι.**

Παίρνω τον πρώτο αριθμό τον 10, επειδή βρίσκεται εντός του διαστήματος [0,10] αυξάνω τον μετρητή πλήθους για αυτό το διάστημα κατά ένα. Στη συνέχεια παίρνω τον δεύτερο αριθμό τον 9, επειδή βρίσκεται εντός του διαστήματος [0,10] αυξάνω τον μετρητή πλήθους για αυτό το διάστημα κατά ένα κ.ο.κ. Παρατηρώ ότι θα χρειαστώ τρεις μετρητές πλήθους για τα 3 αυτά διαστήματα.

## **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Θέτω ως πλήθος των αριθμών  $N=30$ . Εφόσον η άσκηση προσδιόρισε τον αριθμό αυτό θα χρησιμοποιήσω την εντολή LET (π.χ. LET  $N=30$ ).

*Βήμα 2.* Αρχικοποίηση των παραμέτρων.

Διαβάζω καλά την άσκηση και σημειώνω τις λέξεις κλειδιά, από αυτά που μου ζητάει. Δηλαδή στο συγκεκριμένο πρόβλημα σημειώνω την λέξη πλήθος. Επειδή όμως ενδιαφέρομαι για τρία πλήθη θα έχω και τρεις μετρητές πλήθους με αρχικές τιμές μηδέν δηλ.  $PL1=0$ ,  $PL2=0$ ,  $PL3=0$ .

*Βήμα 3.* Επανάληψη της παρακάτω διαδικασίας  $N$  (όπου  $N$  το έχω ορίσει παραπάνω) φορές :

*Βήμα 3.1* Αναζήτηση αριθμού ή δεδομένων προς επεξεργασία.

Επειδή οι αριθμοί είναι γνωστοί εκ των προτέρων και ταυτόχρονα είναι πολλοί θα χρησιμοποιήσουμε την εντολή READ-DATA

*Βήμα 3.2.* Επεξεργασία των δεδομένων με συνθήκη.

Επειδή θέλουμε το πλήθος των αριθμών που ανήκουν στο [0,10] θα χρησιμοποιούμε την εντολή :

```
IF  $X \geq 0$  AND  $X \leq 10$  THEN  $PL1=PL1+1$ 
```

Επειδή θέλουμε το πλήθος των αριθμών που ανήκουν στο (10,20] θα χρησιμοποιούμε την εντολή :

```
IF  $X > 10$  AND  $X \leq 20$  THEN  $PL2=PL2+1$ 
```

Επειδή θέλουμε το πλήθος των αριθμών που ανήκουν στο (20,40) θα χρησιμοποιούμε την εντολή :

```
IF  $X > 20$  AND  $X < 40$  THEN  $PL3=PL3+1$ 
```

*Βήμα 4.* Εκτυπώνω τα αποτελέσματα (τα πλήθη των αριθμών που ανήκουν στα αντίστοιχα διαστήματα).

#### **Δ) Πρόγραμμα**

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΠΛΗΘΟΥΣ ΑΡΙΘΜΩΝ ΠΟΥ ΑΝΗΚΟΥΝ ΣΕ ΔΙΑΣΤ.
20 REM PL1,PL2,PL3=ΜΕΤΡ. ΠΛΗΘΟΥΣ
30 REM N=ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
40 REM ΒΗΜΑ 1. ΘΕΤΩ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
50 LET N=30
60 REM ΒΗΜΑ 2. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
70 PL1=0 : PL2=0 : PL3=0
80 REM ΒΗΜΑ 3. ΕΠΑΝΑΛΗΨΗ ΤΗΣ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑΣ Ν ΦΟΡΕΣ
90 FOR I=1 TO N
100  ΒΗΜΑ 3.1 ΔΙΑΒΑΣΕ ΤΟΝ Ι ΑΡΙΘΜΟ
110  READ X
120  ΒΗΜΑ 3.2 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΒΑΣΗ ΣΥΝΘΗΚΗΣ
130  IF X>=0 AND X<=10 THEN PL1=PL1+1
140  IF X>10 AND X<=20 THEN PL2=PL2+1
150  IF X>20 AND X<40 THEN PL3=PL3+1
160 NEXT I
170 REM ΒΗΜΑ 4. ΕΚΤΥΠΩΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
180 PRINT "ΠΛΗΘΟΣ ΘΕΡΜ. ΕΝΤΟΣ ΤΟΥ [0,10] =",PL1
190 PRINT "ΠΛΗΘΟΣ ΘΕΡΜ. ΕΝΤΟΣ ΤΟΥ (10,20] =",PL2
200 PRINT "ΠΛΗΘΟΣ ΘΕΡΜ. ΕΝΤΟΣ ΤΟΥ (20,40) =",PL3
210 DATA 10, 9, 8, 4, 7, 10, 12, 14, 18, 19, 20, ....., 15
220 END

```

**Σημείωση :** Παρατηρώ ότι είναι δυνατό να χρησιμοποιώ στο ίδιο πρόγραμμα παραπάνω από έναν μετρητές πλήθους όπως παρόμοια θα μπορούσαμε να χρησιμοποιήσουμε παραπάνω από έναν αθροιστές και πολλαπλασιαστές. Στις περιπτώσεις αυτές δίπλα στο όνομα του μετρητή πλήθους για παράδειγμα τοποθετώ και ένα νούμερο για να ξεχωρίσω τον έναν μετρητή από τον άλλο όπως και κάναμε στην προηγούμενη άσκηση.

**Παράδειγμα.** Δίνονται οι παρακάτω 12 μέσες μηνιαίες τιμές της βροχόπτωσης που παρατηρήθηκε στον νομό Δράμας κατά το έτος 1996:

50, 45, 60, 20, 10, 30, 40, 10, 60, 70, 100, 120

Να βρεθεί η ελάχιστη και η μέγιστη βροχόπτωση.

### **Απάντηση**

#### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών  $N=12$  καθώς και οι αριθμοί που θα δίνονται μέσω της μεταβλητής  $X$ .

*Εξοδος* – Η ελάχιστη και η μέγιστη τιμή των 12 αυτών αριθμών.

#### **B) Παράδειγμα με το χέρι.**

Παίρνω τον πρώτο αριθμό τον 50 και τον θεωρώ ως μέγιστο. Στην συνέχεια παίρνω τον δεύτερο αριθμό και ελέγχω αν είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο δηλ. τον 50. Δεν είναι και συνεπώς παίρνω τον επόμενο που είναι το 60. Ελέγχω πάλι αν είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο δηλ. τον 50. Επειδή είναι, θεωρώ από δω και στο εξής τον 60 ως μέγιστο και συνεχίζω την ίδια διαδικασία με τον επόμενο αριθμό κ.ο.κ. Παρόμοια δουλεύω για να βρω το ελάχιστο.

#### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Θέτω ως πλήθος των αριθμών  $N=12$ . Εφόσον η άσκηση προσδιόρισε τον αριθμό αυτό θα χρησιμοποιήσω την εντολή LET (π.χ. LET  $N=12$ ).

*Βήμα 2.* Διαβάζω τον πρώτο αριθμό  $X$  με READ-DATA επειδή οι αριθμοί είναι γνωστοί.

*Βήμα 3.* Θέτω ως αρχική τιμή του μέγιστου τον  $X$  και ως αρχική τιμή του ελάχιστου τον  $X$ .

*Βήμα 4.* Επαναλαμβάνω  $N-1$  φορές την παρακάτω διαδικασία :

*Βήμα 4<sup>α</sup>.* Διαβάζω τον καινούργιο αριθμό.

*Βήμα 4β.* Αν είναι μικρότερος από αυτόν που θεωρούσα ως ελάχιστο θέτω στην θέση του ελάχιστου τον αριθμό αυτό.

*Βήμα 4γ.* Αν είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο θέτω στην θέση του μέγιστου τον αριθμό αυτό.

*Βήμα 5.* Εκτυπώνω τον μέγιστο και τον ελάχιστο.

#### **Δ) Πρόγραμμα**

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΓΙΣΤΟΥ ΕΛΑΧΙΣΤΟΥ
20 REM MEG=ΜΕΓΙΣΤΟΣ, ELAX=ΕΛΑΧΙΣΤΟΣ, Χ=ΑΡΙΘΜΟΣ
30 REM ΒΗΜΑ 1. ΘΕΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
40 LET N=12
50 REM ΒΗΜΑ 2. ΔΩΣΕ ΠΡΩΤΟ ΑΡΙΘΜΟ
60 READ X
70 REM ΒΗΜΑ 3. ΘΕΣΕ ΩΣ ΜΕΓΙΣΤΟ ΚΑΙ ΕΛΑΧΙΣΤΟ ΤΟΝ Χ
80 MEG=X
90 ELAX=X
100 REM ΒΗΜΑ 4. ΕΠΑΝΕΛΑΒΕ ΤΗΝ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑ Ν-1 ΦΟΡΕΣ
110 FOR I=2 TO N
120  REM ΒΗΜΑ 4Α. ΔΩΣΕ ΝΕΟ ΑΡΙΘΜΟ
130  READ X
140  REM ΒΗΜΑ 4Β-Γ. ΕΛΕΓΞΕ ΑΝ ΕΙΝΑΙ ΜΕΓ. Η ΜΙΚΡ. ΤΟΥ ΜΕΓ Ή
    ELAX
150  IF X<ELAX THEN ELAX=X
160  IF X>MEG THEN MEG=X
170 NEXT I
180 REM ΒΗΜΑ 5. ΕΚΤΥΠΩΣΕ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ
190 PRINT "ΕΛΑΧΙΣΤΟΣ=",ELAX
200 PRINT "ΜΕΓΙΣΤΟΣ=",MEG
210 DATA 50, 45, 60, 20, 10, 30, 40, 10, 60, 70, 100, 120
220 END

```

#### **Η εντολή WHILE-WEND**

**Σκοπός :** Η επανάληψη μιας σειράς εντολών έως ότου μια συνθήκη ικανοποιηθεί.

**Σύνταξη :**

XX WHILE <έκφραση>

### Διάφορες εντολές της BASIC

#### XX WEND

Που σημαίνει ότι οι εντολές που βρίσκονται μεταξύ της WHILE-WEND θα εκτελούνται όσο η έκφραση είναι αληθής ή διάφορη του μηδενός. Η παραπάνω εντολή χρησιμοποιείται όταν θέλω να επαναλάβω μια διαδικασία παραπάνω από μια φορές αλλά δεν ξέρω εκ των προτέρων τον αριθμό των επαναλήψεων. Θα πρέπει οι μεταβλητές που βρίσκονται στην <έκφραση> να παίρνουν τιμές πριν την εκτέλεση της WHILE-WEND αλλά και στις εντολές μεταξύ των WHILE-WEND.

**Παράδειγμα.** Δίνεται το παρακάτω πρόγραμμα :

```
10 INPUT "ΔΩΣΕ ΑΡΙΘΜΟ",KEY
20 WHILE KEY<>7
30 INPUT "ΔΩΣΕ ΑΡΙΘΜΟ",KEY
40 WEND
50 PRINT "ΕΠΙΤΥΧΗΣ ΠΡΟΣΒΑΣΗ»
```

Ποιο το αποτέλεσμα της εκτέλεσης του προγράμματος ;

#### **Απάντηση.**

Η πρώτη εντολή εμφανίζει στην οθόνη το μήνυμα «ΔΩΣΕ ΑΡΙΘΜΟ» και περιμένει να δώσει ο χρήστης μια τιμή για την αριθμητική μεταβλητή KEY. Έστω ότι ο χρήστης έδωσε την τιμή 35. Η εντολή 20 ελέγχει αν η πρόταση KEY<>7 είναι αληθής. Επειδή είναι αληθής θα συνεχίσει η εκτέλεση των εντολών μεταξύ 20-40. Συνεπώς θα εμφανισθεί το μήνυμα «ΔΩΣΕ ΑΡΙΘΜΟ» σύμφωνα με την εντολή 30 και θα περιμένει το πρόγραμμα μου να δώσει ο χρήστης μια τιμή για την αριθμητική μεταβλητή KEY. Αν η καινούργια του KEY είναι 45 τότε η εντολή 40 επαναφέρει τον έλεγχο στην εντολή 20 η οποία ελέγχει πάλι αν η πρόταση KEY<>7 είναι αληθής. Αν και πάλι είναι με την καινούργια τιμή του KEY θα εκτελεστούν ξανά οι εντολές μεταξύ των 20-40. Οι ενδιάμεσες εντολές δεν θα εκτελεστούν αν ο χρήστης δώσει ως τιμή για το KEY την 7 οπότε η πρόταση KEY<>7 είναι ψευδής και συνεπώς θα εκτελεστεί η εντολή που ακολουθεί την WEND δηλ. η εντολή 50 (θα εμφανισθεί το μήνυμα "ΕΠΙΤΥΧΗΣ ΠΡΟΣΒΑΣΗ» στην οθόνη).



**Παράδειγμα.** Να βρεθεί το ελάχιστο κοινό πολλαπλάσιο (ΕΚΠ) δύο αριθμών.

**Απάντηση.**

**A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Δύο αριθμοί A,B.

*Έξοδος* – Το Ε.Κ.Π. των δύο αριθμών.

**B) Παράδειγμα με το χέρι.**

Έστω οι αριθμοί 9,6. Θεωρώ ως ΕΚΠ τον μεγαλύτερο δηλ. το 9. Ελέγχω αν το 9 είναι πολλαπλάσιο του 6. Αν ναι τότε το 9 είναι το ΕΚΠ. Αν όχι τότε αυξάνω το υποτιθέμενο ΕΚΠ κατά τον μεγαλύτερο αριθμό δηλ. το 9 γίνεται 9+9 δηλ. 18. Ελέγχω αν το 18 είναι πολλαπλάσιο του 6. Επειδή είναι άρα το ΕΚΠ των δύο αριθμών θα είναι το 18.

**Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Ζητώ δύο αριθμούς A,B.

*Βήμα 2.* Αν ο A είναι μικρότερος του B τους αλλάζω σειρά. Η διαδικασία αυτή γίνεται με έναν από τους παρακάτω δύο τρόπους :

1 <sup>ος</sup> τρόπος K=A A=B B=K	2 <sup>ος</sup> τρόπος SWAP A,B
---	------------------------------------

*Βήμα 3.* Θέτω ως ΕΚΠ τον μεγαλύτερο δηλ. τον A (ΕΚΡ=A)

*Βήμα 4.* Όσο το ΕΚΠ δεν είναι πολλαπλάσιο του B εκτελώ τα παρακάτω βήματα :

*Βήμα 4<sup>α</sup>.* Αυξάνω το ΕΚΠ κατά τον μεγαλύτερο αριθμό δηλ. κατά A (ΕΚΡ=ΕΚΡ+A).

*Βήμα 5.* Εκτυπώνω τα αποτελέσματα.

**Δ) Πρόγραμμα**

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΕΚΠ ΔΥΟ ΑΡΙΘΜΩΝ.

20 REM A,B=ΔΥΟ ΑΡΙΘΜΟΙ, ΕΚΡ=ΤΟ ΕΚΠ ΤΩΝ A,B

30 REM ΒΗΜΑ 1. ΖΗΤΩ ΤΑ A,B

40 INPUT "ΔΩΣΕ A =",A

50 INPUT "ΔΩΣΕ B=",B

```
60 REM ΒΗΜΑ 2. ΤΟΠΟΘΕΤΩ ΣΤΟΝ Α ΤΟΝ ΜΕΓΑΛΥΤΕΡΟ
70 IF A<B THEN SWAP A,B
80 REM ΒΗΜΑ 3. ΘΕΤΩ ΤΟΝ Α ΩΣ ΤΟ ΕΚΠ
90 ΕΚΡ=Α
100 REM ΒΗΜΑ 4. ΟΣΟ ΤΟ ΕΚΠ ΔΕΝ ΕΙΝΑΙ ΠΟΛΛΑΠΛΑΣΙΟ ΤΟΥ Β
110 REM ΕΚΤΕΛΩ ΤΑ ΠΑΡΑΚΑΤΩ
120 WHILE ΕΚΡ MOD Β<>0
130 REM ΒΗΜΑ 4Α. ΑΥΞΗΣΕ ΤΟ ΕΚΠ ΚΑΤΑ Α
140 ΕΚΡ=ΕΚΡ+Α
150 WEND
160 REM ΒΗΜΑ 5. ΕΚΤΥΠΩΣΕ ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ
170 PRINT "ΕΚΠ ΤΩΝ";Α;" ";Β;"ΕΙΝΑΙ ΤΟ";ΕΚΡ
180 END
```

## Επαναληπτικές Ασκήσεις στις εντολές FOR-NEXT και WHILE-WEND

**Άσκηση 1.** α) Ποια από τα παρακάτω προγράμματα είναι συντακτικά σωστά :

10 N=2 20 FOR X=N TO N+10 ..... 90 NEXT N	10 FOR I=1 TO 4 20 FOR J=1 TO 3 30 IF I+J>5 THEN GOTO 60 40 NEXT I 50 NEXT J 60 PRINT I,J
--	--

**Άσκηση 2.** Δίδεται ο παρακάτω πίνακας μέσης θερμοκρασίας το έτος 1996 :

Μήνας	Μέση Θερμοκρασία
Ιανουάριος	8
Φεβρουάριος	8
Μάρτιος	12

Απρίλιος	14
Μάιος	15
Ιούνιος	23
Ιούλιος	28
Αύγουστος	28
Σεπτέμβριος	27
Οκτώβριος	20
Νοέμβριος	12
Δεκέμβριος	9

α) Να βρεθούν οι μήνες κατά τους οποίους είχαμε την ελάχιστη και την μέγιστη μέση θερμοκρασία.

β) Να βρεθεί η μέση θερμοκρασία του έτους 1996.

**Άσκηση 3.** Γνωρίζουμε ότι εάν στην αρχή κάθε έτους τοποθετούμε στην τράπεζα ένα σταθερό ποσό  $A$  με επιτόκιο  $E$ , τότε το κεφάλαιο μετά την  $N$ -οστή κατάθεση θα είναι :

$$K = \frac{A \times (1 + E) \left( (1 + E)^N - 1 \right)}{E}$$

Να βρεθεί το κεφάλαιο  $K$  για  $N=1,2,3,\dots,30$  δεδομένου ότι  $E=13.5\%$  και  $A=120.000\delta\rho\chi.$ . Θα μπορούσατε να επαναλάβετε το ίδιο για τις τιμές του  $A=130.000\delta\rho\chi., 140.000\delta\rho\chi., \dots, 360.000\delta\rho\chi.$

**Άσκηση 4.** Τι περιμένετε να εκτυπώσουν τα παρακάτω προγράμματα :

<pre> 10 V=0 20 FOR I1=1 TO 4 30  FOR I2=2 TO 3 40    V=V+1 50  NEXT I2 60 NEXT I1 70 PRINT V </pre>	<pre> 10 V=0 20 FOR I=10 TO 1 STEP -1 30  V=V+1 40  I=I+1 50 NEXT I 60 PRINT V </pre>
--	---

**Άσκηση 5.** Ο όγκος ενός κυλινδρικού κορμοτεμαχίου δίνεται από τον τύπο :

$$V = \frac{\pi D^2 H}{4}$$

όπου

V= ο όγκος του κυλινδρικού κορμοτεμαχίου

D= η διάμετρος του κορμοτεμαχίου

H= το ύψος του κορμοτεμαχίου

$$\pi=3.14159$$

Αν το μήκος (ύψος) του κορμοτεμαχίου είναι σταθερό και ίσο με 15m να υπολογίσουμε και να εμφανίσουμε τον όγκο του κορμοτεμαχίου όταν η διάμετρος του μεταβάλλεται από 25cm έως 58cm με βήμα 2cm.

**Άσκηση 6.** Ο τύπος που δίνει το εμβαδόν D της αναδασωτέας έκτασης αρχικού εμβαδού E μετά από N χρόνια με ποσοστό αναδάσωσης P είναι ο εξής :

$$D = E(1 + P)^N$$

Να βρεθεί μετά από N=1,2,...,10 χρόνια το εμβαδόν της αναδασωτέας έκτασης αν E=1000 και P=0.2%.

**Άσκηση 7.** Ο όγκος ενός κυλινδρικού κορμοτεμαχίου δίνεται από τον τύπο :

$$V = \frac{\pi D^2 H}{4}$$

όπου

V= ο όγκος του κυλινδρικού κορμοτεμαχίου

D= η διάμετρος του κορμοτεμαχίου

H= το ύψος του κορμοτεμαχίου

$$\pi=3.14159$$

Αν το ύψος παίρνει τις εξής τιμές :

$$10, 13, 16, \dots, 40$$

και η διάμετρος για κάθε μια από τις τιμές παίρνει τιμές :

$$20, 30, 40, \dots, 120$$

να εμφανιστούν οι όγκοι που βρίσκονται στο διάστημα [100-200].

**Άσκηση 8.** Να διαβαστούν  $N$  αριθμοί και να τοποθετηθούν σε έναν πίνακα. Μετά να βρεθεί και να εμφανιστεί ο μέσος όρος των μη μηδενικών αριθμών που βρίσκονται στο διάστημα [-10,10].

**Άσκηση 9.** Δίδονται οι 10 αριθμοί :

$$15, 30, 45, \dots$$

Να γραφεί πρόγραμμα το οποίο βρίσκει και εμφανίζει τον μεγαλύτερο αριθμό που βρίσκεται στο διάστημα [50,150].

Σημείωση - Το πρόγραμμα να μην εμφανίζει το μήνυμα OUT OF DATA.

**Άσκηση 10.** Η μηνιαία δόση  $M$  αποπληρωμής ενός στεγαστικού δανείου κεφαλαίου  $A$  με επιτόκιο  $E$ , σε  $N$  χρόνια δίνεται από τον τύπο :

$$M = \frac{A \times E}{12} \times \frac{(1 + E)^N}{(1 + E)^N - 1}$$

Να υπολογιστεί η μηνιαία δόση για  $E=12\%$  και  $N=1,2,3,\dots,25$  και  $A=1.000.000, 2.000.000, \dots, 40.000.000$ .

Κεφάλαιο\Χρόνια	1	2	....	25
<b>1.000.000</b>	93.333	49.308	...	10.625
<b>2.000.000</b>	186.668	98.616	...	21.250
....	...	...	...	...
<b>40.000.000</b>	3.733.333	1.972.327	...	425.000

**Άσκηση 11.** Ο όγκος ενός κυλινδρικού κορμοτεμαχίου δίνεται από τον τύπο :

$$V = \frac{\pi D^2 H}{4}$$

όπου

$V$ = ο όγκος του κυλινδρικού κορμοτεμαχίου

$D$ = η διάμετρος του κορμοτεμαχίου

$H$ = το ύψος του κορμοτεμαχίου

$\pi=3.14159$

Αν η διάμετρος του κορμοτεμαχίου μεταβάλλεται από 25cm έως 115cm κατά 2cm και για κάθε μια τιμή της διαμέτρου το ύψος του κορμοτεμαχίου παίρνει τις τιμές 2m, 4m, 6m, ..., 12m να βρεθεί το πλήθος των κορμοτεμαχίων που ο όγκος τους βρίσκεται μεταξύ  $4m^3$  και  $6m^3$ .

**Άσκηση 12.** Σε μια καταμέτρηση θερμοκρασιών τον μήνα Δεκέμβριο στην πόλη της Δράμας είχαμε τα εξής αποτελέσματα :

10,12,13,12,14,16,17,19,13,9,8,7,10,14,16,19,10,8,7,5,4,2,2,4,5,6,9,8,6,9,9

Να γραφεί πρόγραμμα που να υπολογίζει α) την μέση θερμοκρασία της πόλης και β) την ελάχιστη και μέγιστη θερμοκρασία τον μήνα Δεκέμβριο.

**Άσκηση 13.** Γνωρίζουμε ότι ο όγκος ενός κορμού δίνεται από τον τύπο :

$$V = \frac{S^2 h}{4\pi}$$

όπου  $S$  είναι η περίμετρος του κορμού,  $h$  το ύψος του κορμού και  $\pi=3.14$ . Να βρεθεί ο όγκος του κορμού για τις εξής τιμές του ύψους  $h$  : 100cm, 105cm, 110cm, 115cm, ..., 200cm δεδομένου ότι η περίμετρος  $S=30$ cm. Θα μπορούσατε να επαναλάβετε το ίδιο για τις τιμές του  $S=35$ cm, 40cm, 45cm, ..., 100cm.

**Άσκηση 14.** Έστω  $d_0, d_1$  οι διάμετροι στην βάση και στο σημείο αποκορύφωσης ενός δέντρου. Αν το συνολικό μήκος του δέντρου είναι  $\ell$ , τότε ο όγκος  $V$  του δέντρου δίνεται από τον τύπο :

$$V = \frac{\pi}{4} \times \ell \times \left( \frac{d_0^2 + d_1^2}{2} \right)$$

Να γραφεί πρόγραμμα που να υπολογίζει τον όγκο του δέντρου για  $d_0 = 0.1m, 0.2m, \dots, 1m$ , ενώ  $d_1 = 0.1m$  και  $\ell = 25m$ . Μπορείτε να λύσετε το παραπάνω πρόβλημα για  $d_1 = 0.2m, 0.4m, 0.6m, \dots, 1m$ .

**Άσκηση 15.** Να διαβαστούν δύο αριθμοί και στη συνέχεια να υπολογιστεί ο μέγιστος κοινός διαιρέτης των δύο αριθμών.

## ΠΙΝΑΚΕΣ

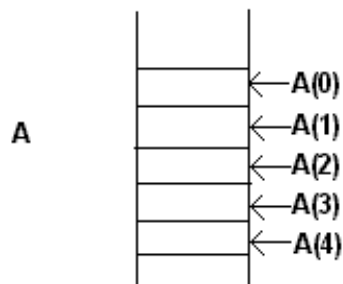
Είχαμε συναντήσει στα προηγούμενα κεφάλαια την έννοια της απλής μεταβλητής και είχαμε αναφέρει ότι αποτελεί μια θέση στην μνήμη του Η/Υ με ένα συγκεκριμένο όνομα η οποία δέχεται μια μόνο σταθερά. Τα είδη των απλών μεταβλητών τα είχαμε χωρίσει σε αριθμητικές και αλφαριθμητικές μεταβλητές. Η BASIC εκτός από τις απλές μεταβλητές έχει και *σύνθετες μεταβλητές*, μεταβλητές δηλαδή που καταλαμβάνουν παραπάνω από μια θέση στην μνήμη του Η/Υ, έχουν ένα συγκεκριμένο όνομα και δέχονται έναν συγκεκριμένο τύπο δεδομένων. Τα ονόματα των μεταβλητών αυτών ακολουθούν τους ίδιους κανόνες με αυτά των απλών μεταβλητών. Οι σύνθετες αυτές μεταβλητές χωρίζονται επίσης σε αριθμητικές και αλφαριθμητικές ανάλογα με τον τύπο των δεδομένων που δέχονται και ο διαχωρισμός τους γίνεται με τον ίδιο τρόπο δηλ. το όνομα των αλφαριθμητικών σύνθετων μεταβλητών ακολουθεί το σύμβολο του δολαρίου (\$).

### Μεταβλητές

Απλές		Σύνθετες	
Αριθμητικές	Αλφαριθμητικές	Αριθμητικές	Αλφαριθμητικές

Για να ξεχωρίσω όμως τα στοιχεία της σύνθετης μεταβλητής χρησιμοποιώ επιπλέον και δείκτες. Έτσι εάν έχω μια σύνθετη αριθμητική μεταβλητή με το όνομα A που καταλαμβάνει 5 συνεχόμενες θέσεις στην μνήμη του Η/Υ, θα αναφέρομε στο πρώτο στοιχείο της μεταβλητής αυτής με το όνομα A(0), στο δεύτερο με το όνομα A(1) κ.ο.κ. Για να μην υπάρξει σύγχυση από εδώ και στο εξής θα θεωρούμε ότι το πρώτο στοιχείο της σύνθετης μεταβλητής θα είναι το A(1) και όχι το A(0), το δεύτερο το A(2) και όχι το A(1) κ.ο.κ. Τις σύνθετες αυτές μεταβλητές τις ονομάζουμε και *πίνακες*. Ο δείκτης μιας σύνθετης μεταβλητής μπορεί να είναι μια σταθερά ή μεταβλητή ή παράσταση που

παίρνει ακέραιες τιμές και φυσικά δεν θα πρέπει να ξεπερνά τα όρια της διάστασης του πίνακα π.χ.  $A(3)$ ,  $A(K)$ ,  $A(2*L-3)$ .



Συνήθεις λόγοι για την χρήση των πινάκων είναι οι εξής :

1. Η χρησιμοποίηση των δεδομένων παραπάνω από μια φορά.
2. Η αποθήκευση των δεδομένων πριν την χρησιμοποίησή τους.

Εάν λοιπόν ένας από τους παραπάνω λόγους συντρέχει στο πρόβλημα σας, τότε θα πρέπει να χρησιμοποιήσετε πίνακες για την επίλυση του προβλήματος σας.

Η συνήθης επίλυση προβλημάτων που περιλαμβάνουν πίνακες απαρτίζεται από τρία βήματα :

1. Ορισμός πίνακα και εισαγωγή στοιχείων στον πίνακα.
2. Επεξεργασία στοιχείων πίνακα (εύρεση πλήθους, αθροίσματος και γινομένου στοιχείων που ικανοποιούν κάποια συνθήκη (ή όχι), εύρεση μικρότερου-μεγαλύτερου, ταξινόμηση στοιχείων σε αύξουσα ή φθίνουσα σειρά κ.α.)
3. Εμφάνιση αποτελεσμάτων και στοιχείων του πίνακα.

Με την υλοποίηση των παραπάνω βημάτων θα ασχοληθούμε στην συνέχεια.

## Ορισμός Πίνακα

Εάν θέλω να χρησιμοποιήσω έναν πίνακα μέσα στο πρόγραμμα μου θα πρέπει πρώτα να τον ορίσω. Ο ορισμός του πίνακα γίνεται πριν την χρήση του μέσα στο πρόγραμμα από την εντολή :



xx DIM όνομα πίνακα(μέγιστη διάσταση πίνακα-1)

**Παράδειγμα.** Θέλω να δημιουργήσω τον πίνακα ο οποίος παρουσιάζεται στην παραπάνω εικόνα.

**Απάντηση.** Η εντολή που θα χρησιμοποιήσω θα είναι η εξής :

```
10 DIM A(4)
```

**Σημείωση :** Για λόγους απλούστευσης εάν θέλω να δημιουργήσω έναν πίνακα A που θα δέχεται N στοιχεία θα δίνω την εντολή :

```
xx DIM A(N)
```

**Παράδειγμα.** Να ορίσετε έναν πίνακα που θα δέχεται τα ονόματα των 12 μηνών του έτους.

**Απάντηση.** Λαμβάνοντας υπ' όψιν την παραπάνω σημείωση η εντολή που θα χρησιμοποιήσω θα είναι η εξής :

```
10 DIM M$(12)
```

### **Εισαγωγή τιμών σε έναν πίνακα**

Συνήθως η καταχώρηση των τιμών σε έναν πίνακα γίνεται με συνδυασμό των εντολών FOR-NEXT και της εντολής INPUT εάν δεν είναι γνωστά τα δεδομένα εκ των προτέρων ή της εντολής READ-DATA εάν τα δεδομένα είναι ήδη γνωστά.

**Παράδειγμα.** Να διαβαστούν N αριθμοί και να καταχωρηθούν σε έναν πίνακα A. Οι τιμές θα δίδονται από τον χρήστη.

**Απάντηση.**

Επειδή δεν γνωρίζουμε το πλήθος των αριθμών N, θα το ζητήσουμε από τον χρήστη με την εντολή INPUT. Επειδή επίσης δεν γνωρίζουμε τους αριθμούς θα τους ζητήσουμε και αυτούς με την εντολή INPUT.

```
10 INPUT "N=",N
```

```
20 DIM A(N)
```

```
20 FOR I=1 TO N
```

```
30 PRINT "A(“;I;”)=“;
```

40 INPUT A(I)

50 NEXT I

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι :

Εντολή	Μνήμη	Οθόνη
10	Καταχώρηση μιας τιμής για το N από τον χρήστη.	N=?
20	Δημιουργία N+1 θέσεων για τον πίνακα A	
30-50	Καταχώρηση τιμών στα στοιχεία A(I) του πίνακα.	A(I)=?

**Παράδειγμα.** Δίδονται οι παρακάτω 10 αριθμοί :

20, 15, 25, 32, ...,23

Να καταχωρηθούν σε έναν πίνακα A.

**Απάντηση.** Επειδή το πλήθος των τιμών είναι γνωστό θα χρησιμοποιήσουμε την εντολή N=10 αντί της INPUT του προηγούμενου παραδείγματος. Επειδή επίσης οι τιμές του πίνακα είναι γνωστές, θα διαβαστούν με την εντολή READ-DATA.

10 N=10

20 DIM A(N)

30 FOR I=1 TO N

40 READ A(I)

50 NEXT I

60 DATA 20, 15, 25, 32, ...,23

Εντολή	Μνήμη	Οθόνη
10	Καταχώρηση της τιμής 10 για το N	Δεν εμφανίζεται τίποτα στην οθόνη.

20	Δημιουργία 11 θέσεων για τον πίνακα A	Δεν εμφανίζεται τίποτα στην οθόνη.
30-60	Καταχώρηση τιμών στα στοιχεία A(I) του πίνακα από τις τιμές της DATA	Δεν εμφανίζεται τίποτα στην οθόνη.

Συνεπώς η είσοδος των δεδομένων στον πίνακα μου θα εξαρτάται από το αν γνωρίζω εκ των προτέρων ή όχι τα δεδομένα μου. Αν δεν τα γνωρίζω θα εφαρμόζω το πρόγραμμα του πρώτου παραδείγματος διαφορετικά του δευτέρου παραδείγματος.

### Εμφάνιση τιμών ενός πίνακα

Εμφάνιση των τιμών ενός πίνακα γίνεται με συνδυασμό των εντολών FOR-NEXT και της εντολής PRINT.

**Παράδειγμα.** Έστω ότι σε ένα πρόγραμμα έχουν καταχωρηθεί οι 10 τιμές σε έναν πίνακα A. Να συνεχίσετε το πρόγραμμα αυτό με την εμφάνιση των τιμών του πίνακα.

#### Απάντηση.

10 N=10

..... (πρόγραμμα που μεσολαβεί)

100 FOR I=1 TO N

110 PRINT "A(;"I;")=";

120 PRINT A(I)

130 NEXT I

### Επεξεργασία των στοιχείων του πίνακα.

Στην ενότητα αυτή θα αναφερθούμε σε συγκεκριμένες εφαρμογές που μπορούμε να αντιμετωπίσουμε.

## **A) Εύρεση πλήθους – γινομένου - αθροίσματος στοιχείων του πίνακα που ικανοποιούν κάποια συνθήκη (ή όχι).**

**Παράδειγμα.** Να διαβαστούν N αριθμοί και να τοποθετηθούν σε έναν πίνακα A. Να βρεθούν και να εμφανισθούν :

- α) το πλήθος των αριθμών που βρίσκονται έξω από το διάστημα  $[-10,10]$ ,
- β) το γινόμενο των αριθμών που είναι διάφοροι του μηδενός και βρίσκονται στο διάστημα  $[-1,1]$ ,
- γ) το άθροισμα των αριθμών που είναι πολλαπλάσια του 5,
- δ) ο μέσος όρος των άρτιων αριθμών.

**Απάντηση.**

### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών N καθώς και οι N αριθμοί που θα τοποθετηθούν στον πίνακα A.

*Έξοδος* – Τα ζητήματα α,β,γ,δ.

### **B) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών. Επειδή το πλήθος δεν είναι γνωστό χρησιμοποιώ την εντολή INPUT “Δώσε πλήθος :”;N.

*Βήμα 2.* Ορισμός πίνακα A.

*Βήμα 3.* Εισαγωγή αριθμών του πίνακα. Επειδή οι αριθμοί αυτοί δεν είναι γνωστοί εκ των προτέρων θα χρησιμοποιήσουμε την εντολή INPUT A(I).

*Βήμα 4.* Αρχικοποίηση των παραμέτρων.

Διαβάζω καλά την άσκηση και σημειώνω τις λέξεις κλειδιά, από αυτά που μου ζητάει. Δηλαδή στο συγκεκριμένο πρόβλημα σημειώνω την λέξη πλήθος στο (α), γινόμενο στο (β), άθροισμα στο (γ) και μέσος όρος στο (δ) που απευθείας σημαίνει ένα καινούργιο άθροισμα και πλήθος. Οι λέξεις αυτές θα με βοηθήσουν να επιλέξω το είδος των παραμέτρων που θα χρησιμοποιήσω εκτός και εντός της ανακύκλωσης. Στο πρόβλημα λοιπόν αυτό θα θέσω έναν μετρητή πλήθους PL1=0 για το (α) ζήτημα, έναν πολλαπλασιαστή P=1 για το (β) ζήτημα, έναν αθροιστή S1=0 για το (γ) ζήτημα και τέλος έναν αθροιστή S2=0 και έναν μετρητή πλήθους PL2=0 για το (δ) ζήτημα.

*Βήμα 5.* Καθώς το I παίρνει τιμές από το 1 έως το N επανέλαβε τα παρακάτω βήματα :

*Βήμα 5<sup>α</sup>.* Εάν το A(I) δεν ανήκει στο διάστημα [-10,10] αύξησε τον μετρητή πλήθους PL1 κατά ένα.

*Βήμα 5<sup>β</sup>.* Εάν το A(I) ανήκει στο κλειστό διάστημα [-1,1] και είναι διάφορο του μηδενός πολλαπλασίασε τον πολλαπλασιαστή επί A(I).

*Βήμα 5<sup>γ</sup>.* Εάν το A(I) είναι πολλαπλάσιο του 5 τότε αύξησε τον αθροιστή S1 κατά A(I).

*Βήμα 5<sup>δ</sup>.* Εάν το A(I) είναι άρτιος τότε αύξησε τον μετρητή πλήθους PL2 κατά ένα και τον αθροιστή S2 κατά A(I).

*Βήμα 6.* Εύρεση του μέσου όρου. Εάν το PL2 είναι 0 τότε ο μέσος όρος είναι μηδέν, διαφορετικά ο μέσος όρος είναι S2/PL2.

*Βήμα 7.* Εκτυπώνω τα αποτελέσματα.

### Γ) Πρόγραμμα.

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΑΘΡ. ΠΛ. ΚΑΙ ΓΙΝ. ΣΤΟΙΧ. ΠΙΝΑΚΑ
20 REM S1,S2=ΑΘΡΟΙΣΤΕΣ ΓΙΑ (Γ),(Δ) ΖΗΤΗΜΑ
30 REM PL1,PL2=ΜΕΤΡΗΤΕΣ ΠΛΗΘΟΥΣ ΓΙΑ (Α),(Δ) ΖΗΤΗΜΑ
40 REM P=ΠΟΛΛΑΠΛΑΣΙΑΣΤΗΣ ΓΙΑ (Β) ΖΗΤΗΜΑ
50 REM ΒΗΜΑ 1. ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
60 INPUT "ΔΩΣΕ ΠΛΗΘΟΣ ΤΙΜΩΝ =",N
70 REM ΒΗΜΑ 2. ΟΡΙΣΜΟΣ ΠΙΝΑΚΑ
80 DIM A(N)
90 REM ΒΗΜΑ 3. ΕΙΣΑΓΩΓΗ ΤΙΜΩΝ ΣΤΟΝ ΠΙΝΑΚΑ Α
100 FOR I=1 TO N
110 PRINT "Α(“;I;”)=“;
120 INPUT A(I)
130 NEXT I
140 REM ΒΗΜΑ 4. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
150 PL1=0 : P=0 : S1=0 : PL2=0 : S2=0
160 REM ΒΗΜΑ 5. ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΙΝΑΚΑ
170 FOR I=1 TO N
180 REM ΒΗΜΑ 5Α. ΕΥΡΕΣΗ ΠΛΗΘΟΥΣ ΣΤΟΙΧ. ΕΚΤΟΣ ΤΟΥ [-10,10]
190 IF A(I)<-10 OR A(I)>10 THEN PL1=PL1+1

```

```
200 REM ΒΗΜΑ 5Β. ΕΥΡΕΣΗ ΓΙΝΟΜΕΝΟΥ ΜΗ ΜΗΔ. ΣΤΟΙΧ. ΣΤΟ [-1,1]
210 IF A(I)<>0 AND A(I)>=-1 AND A(I)<=1 THEN P=P*A(I)
220 REM ΒΗΜΑ 5Γ. ΕΥΡΕΣΗ ΑΘΡ. ΠΟΛΛΑΠΛΑΣΙΩΝ ΤΟΥ 5
230 IF A(I) MOD 5=0 THEN S1=S1+A(I)
240 REM ΒΗΜΑ 5Δ. ΕΥΡΕΣΗ ΠΛ. ΚΑΙ ΑΘΡ. ΑΡΤΙΩΝ ΑΡΙΘΜΩΝ
250 IF A(I) MOD 2=0 THEN PL2=PL2+1 : S2=S2+A(I)
260 NEXT I
270 REM ΒΗΜΑ 6. ΕΥΡΕΣΗ ΜΕΣΟΥ ΟΡΟΥ
280 IF PL2=0 THEN MO=0 ELSE MO=S2/PL2
290 REM ΒΗΜΑ 7. ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
300 PRINT "ΠΛΗΘΟΣ ΣΤΟΙΧ. ΕΚΤΟΣ ΤΟΥ [-10,10] =",PL1
310 PRINT "ΓΙΝΟΜΕΝΟ ΜΗ ΜΗΔ. ΣΤΟΙΧ. ΣΤΟ [-1,1] =",P
320 PRINT "ΑΘΡΟΙΣΜΑ ΠΟΛΛΑΠΛΑΣΙΩΝ ΤΟΥ 5 =",S1
330 PRINT "ΜΕΣΟΣ ΟΡΟΣ ΑΡΤΙΩΝ =",MO
340 END
```

**Παράδειγμα.** Δίδονται οι παρακάτω 30 αριθμοί :

20, -10, 37, ...,89

Να τοποθετηθούν σε έναν πίνακα A. Να βρεθεί και να εμφανισθεί ο μέσος όρος των στοιχείων του πίνακα.

**Απάντηση.**

#### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών  $N=30$  καθώς και οι  $N$  αριθμοί που θα τοποθετηθούν στον πίνακα A.

*Έξοδος* – Ο μέσος όρος των στοιχείων του πίνακα.

#### **B) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών. Επειδή το πλήθος είναι γνωστό χρησιμοποιώ την εντολή  $N=30$ .

*Βήμα 2.* Ορισμός πίνακα A.

*Βήμα 3.* Εισαγωγή αριθμών στον πίνακα. Επειδή οι αριθμοί αυτοί είναι γνωστοί εκ των προτέρων θα χρησιμοποιήσουμε την εντολή  $READ A(I)$ .

*Βήμα 4.* Αρχικοποίηση των παραμέτρων.

Επειδή η άσκηση ζητάει μέσο όρο, άρα θα χρειαστώ έναν αθροιστή  $S=0$ , ενώ μετρητής πλήθους δεν χρειάζεται γιατί το πλήθος των στοιχείων του πίνακα είναι  $N$  (όλα δηλαδή και όχι αυτά που πληρούν κάποια συνθήκη).

*Βήμα 5.* Καθώς το  $I$  παίρνει τιμές από το 1 έως το  $N$  επανέλαβε το παρακάτω βήμα :

*Βήμα 5α.* Αύξησε τον αθροιστή  $S$  κατά  $A(I)$ .

*Βήμα 6.* Εύρεση του μέσου όρου  $MO=S/N$

*Βήμα 7.* Εκτυπώνω τα αποτελέσματα.

### Γ) Πρόγραμμα.

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ ΣΤΟΙΧ. ΠΙΝΑΚΑ
20 REM S=ΑΘΡΟΙΣΤΗΣ
30 REM ΒΗΜΑ 1. ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
40 N=30
50 REM ΒΗΜΑ 2. ΟΡΙΣΜΟΣ ΠΙΝΑΚΑ
60 DIM A(N)
70 REM ΒΗΜΑ 3. ΕΙΣΑΓΩΓΗ ΤΙΜΩΝ ΣΤΟΝ ΠΙΝΑΚΑ A
80 FOR I=1 TO N
90 READ A(I)
100 NEXT I
110 REM ΒΗΜΑ 4. ΑΡΧΙΚΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ
120 S=0
130 REM ΒΗΜΑ 5. ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΙΝΑΚΑ
140 FOR I=1 TO N
150 REM ΒΗΜΑ 5Α. ΕΥΡΕΣΗ ΑΘΡ. ΣΤΟΙΧ. ΠΙΝΑΚΑ
160 S=S+A(I)
170 NEXT I
180 REM ΒΗΜΑ 6. ΕΥΡΕΣΗ ΜΕΣΟΥ ΟΡΟΥ
190 MO=S/N
200 REM ΒΗΜΑ 7. ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
210 PRINT "ΜΕΣΟΣ ΟΡΟΣ =",MO
220 DATA 20, -10, 37, ...,89
230 END

```

## **B) Εύρεση ελαχίστου-μεγίστου.**

**Παράδειγμα.** Να διαβαστούν  $N$  αριθμοί και εφόσον τοποθετηθούν σε έναν πίνακα  $A$  να βρεθεί ο ελάχιστος και ο μέγιστος των στοιχείων αυτών.

**Απάντηση.**

### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών  $N$  καθώς και οι αριθμοί που θα τοποθετηθούν στον πίνακα  $A$ .

*Έξοδος* – Η ελάχιστη και η μέγιστη τιμή των  $N$  αυτών τιμών του πίνακα  $A$ .

### **B) Παράδειγμα με το χέρι.**

Παίρνω τον πρώτο αριθμό του πίνακα  $A$  ( $A(1)$ ) έστω τον 50 και τον θεωρώ ως μέγιστο. Στην συνέχεια παίρνω τον δεύτερο αριθμό  $A(2)$  και ελέγχω αν είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο δηλ. τον 50. Δεν είναι και συνεπώς παίρνω τον επόμενο που είναι το 60. Ελέγχω πάλι αν είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο δηλ. τον 50. Επειδή είναι, θεωρώ από δω και στο εξής τον 60 ως μέγιστο και συνεχίζω την ίδια διαδικασία με τον επόμενο αριθμό κ.ο.κ. Παρόμοια δουλεύω για να βρω το ελάχιστο.

## **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών  $N$ . Εφόσον το  $N$  δεν είναι γνωστό χρησιμοποιώ την εντολή INPUT.

*Βήμα 2.* Ορίζω τον πίνακα  $A$ .

*Βήμα 3.* Διαβάζω τις τιμές του πίνακα  $A$  με την εντολή INPUT εφόσον δεν είναι γνωστές εκ των προτέρων.

*Βήμα 4.* Θέτω ως αρχική τιμή του μέγιστου τον  $A(1)$  και ως αρχική τιμή του ελάχιστου τον  $A(1)$ .

*Βήμα 5.* Επαναλαμβάνω την παρακάτω διαδικασία καθώς το  $I$  παίρνει τιμές από 2 έως  $N$  :

*Βήμα 5α.* Αν ο  $A(I)$  είναι μικρότερος από αυτόν που θεωρούσα ως ελάχιστο θέτω στην θέση του ελάχιστου τον αριθμό αυτό.



*Βήμα 5β.* Αν ο  $A(I)$  είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο θέτω στην θέση του μέγιστου τον αριθμό αυτό.

*Βήμα 6.* Εκτυπώνω τον μέγιστο και τον ελάχιστο.

#### **Δ) Πρόγραμμα**

```

10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΓΙΣΤΟΥ ΕΛΑΧΙΣΤΟΥ
20 REM MEG=ΜΕΓΙΣΤΟΣ, ELAX=ΕΛΑΧΙΣΤΟΣ
30 REM ΒΗΜΑ 1. ΘΕΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
40 INPUT "Δώσε πλήθος αριθμών =",N
50 REM ΒΗΜΑ 2. ΟΡΙΣΜΟΣ ΠΙΝΑΚΑ
60 DIM A(N)
70 REM ΒΗΜΑ 3. ΔΙΑΒΑΣΜΑ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΙΝΑΚΑ
80 FOR I=1 TO N
90 PRINT "A(“;I;”)=“;
100 INPUT A(I)
110 NEXT I
120 REM ΒΗΜΑ 4. ΘΕΣΕ ΩΣ ΜΕΓΙΣΤΟ ΚΑΙ ΕΛΑΧΙΣΤΟ ΤΟΝ A(1)
130 MEG=A(1)
140 ELAX=A(1)
150 REM ΒΗΜΑ 5. ΕΠΑΝΕΛΑΒΕ ΤΗΝ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑ N-1 ΦΟΡΕΣ
160 FOR I=2 TO N
170 REM ΒΗΜΑ 5Α. ΕΛΕΓΞΕ ΑΝ Ο A(I) ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟΣ ΤΟΥ ΕΛΑΧ.
180 IF A(I)<ELAX THEN ELAX=A(I)
190 REM ΒΗΜΑ 5Β. ΕΛΕΓΞΕ ΑΝ Ο A(I) ΕΙΝΑΙ ΜΕΓΑΛΥΤ. ΤΟΥ ΜΕΓΙΣΤΟΥ
200 IF A(I)>MEG THEN MEG=A(I)
210 NEXT I
220 REM ΒΗΜΑ 6. ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
230 PRINT "ΕΛΑΧΙΣΤΟΣ=",ELAX
240 PRINT "ΜΕΓΙΣΤΟΣ=",MEG
250 END

```

**Παράδειγμα.** Δίνεται ο παρακάτω πίνακας μέσης μηνιαίας θερμοκρασίας κατά το 1996.

Μήνας	Θερμοκρασία
Ιανουάριος	5
Φεβρουάριος	10
Μάρτιος	12
Απρίλιος	17
Μάιος	19
Ιούνιος	20
Ιούλιος	29
Αύγουστος	29
Σεπτέμβριος	24
Οκτώβριος	19
Νοέμβριος	15
Δεκέμβριος	5

Να γραφεί πρόγραμμα που να υπολογίζει την ελάχιστη και μέγιστη θερμοκρασία καθώς και του μήνες κατά τους οποίους εμφανίσθηκαν οι τιμές αυτές.

#### **Απάντηση.**

##### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των μηνών  $N=12$ , οι  $N$  μήνες και οι  $N$  θερμοκρασίες.

*Έξοδος* – Η ελάχιστη και η μέγιστη τιμή των  $N$  αυτών μηνών καθώς και οι μήνες κατά τους οποίους εμφανίσθηκαν.

##### **B) Παράδειγμα με το χέρι.**

Όπως και στο προηγούμενο παράδειγμα πρώτα θα βρω την ελάχιστη (π.χ.5) και την μέγιστη τιμή (π.χ.29) των 12 μηνών. Στην συνέχεια πρώτα θα ψάξω να βρω σε ποιους μήνες εμφανίζεται η ελάχιστη τιμή (π.χ. Δεκέμβριος, Ιανουάριος) και σε ποιους μήνες εμφανίζεται η μέγιστη τιμή (π.χ. Ιούλιος, Αύγουστος).

##### **Γ) Αλγόριθμος προβλήματος**

*Βήμα 1.* Θέτω ως πλήθος των μηνών  $N=12$ .

*Βήμα 2.* Ορίζω τον πίνακα M\$ που θα δέχεται τα ονόματα των μηνών και T τον πίνακα που δέχεται τις αντίστοιχες θερμοκρασίες.

*Βήμα 3.* Διαβάζω τις τιμές των πινάκων M\$ και Tα με την εντολή READ-DATA εφόσον οι γνωρίζω εκ των προτέρων τους μήνες και τις θερμοκρασίες του κάθε μήνα.

*Βήμα 4.* Θέτω ως αρχική τιμή του μέγιστου τον T(1) και ως αρχική τιμή του ελάχιστου τον T(1).

*Βήμα 5.* Επαναλαμβάνω την παρακάτω διαδικασία καθώς το I παίρνει τιμές από 2 έως N :

*Βήμα 5α.* Αν ο T(I) είναι μικρότερος από αυτόν που θεωρούσα ως ελάχιστο θέτω στην θέση του ελάχιστου τον αριθμό αυτό.

*Βήμα 5β.* Αν ο T(I) είναι μεγαλύτερος από αυτόν που θεωρούσα ως μέγιστο θέτω στην θέση του μέγιστου τον αριθμό αυτό.

*Βήμα 6.* Εμφανίζω την τιμή του ελάχιστου.

*Βήμα 7.* Επαναλαμβάνω την παρακάτω διαδικασία καθώς το I παίρνει τιμές από 2 έως N :

*Βήμα 7α.* Αν το T(I) είναι ίσο με το ελάχιστο που βρήκα παραπάνω εμφανίζω την τιμή M\$(I) του αντίστοιχου μήνα.

*Βήμα 8.* Εμφανίζω την μέγιστη τιμή.

*Βήμα 9.* Επαναλαμβάνω την παρακάτω διαδικασία καθώς το I παίρνει τιμές από 2 έως N :

*Βήμα 9α.* Αν το T(I) είναι ίσο με το μέγιστο που βρήκα παραπάνω εμφανίζω την τιμή M\$(I) του αντίστοιχου μήνα.

#### **Δ) Πρόγραμμα**

```
10 REM ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΓΙΣΤΗΣ ΕΛΑΧΙΣΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ
```

```
20 REM MEG=ΜΕΓΙΣΤΟΣ, ELAX=ΕΛΑΧΙΣΤΟΣ
```

```
30 REM ΒΗΜΑ 1. ΘΕΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
```

```
40 N=12
```

```
50 REM ΒΗΜΑ 2. ΟΡΙΣΜΟΣ ΠΙΝΑΚΩΝ
```

```
60 DIM M$(N), T(N)
```

```
70 REM ΒΗΜΑ 3. ΔΙΑΒΑΣΜΑ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΩΝ ΠΙΝΑΚΩΝ
```

```
80 FOR I=1 TO N
```

```
90 READ M$(I), T(I)
```

```
100 NEXT I
110 REM ΒΗΜΑ 4. ΘΕΣΕ ΩΣ ΜΕΓΙΣΤΟ ΚΑΙ ΕΛΑΧΙΣΤΟ ΤΟΝ T(1)
120 MEG=T(1)
130 ELAX=T(1)
140 REM ΒΗΜΑ 5. ΕΠΑΝΕΛΑΒΕ ΤΗΝ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑ N-1 ΦΟΡΕΣ
150 FOR I=2 TO N
160 REM ΒΗΜΑ 5A. ΕΛΕΓΞΕ ΑΝ Ο T(I) ΕΙΝΑΙ ΜΙΚΡΟΤΕΡΟΣ ΤΟΥ ΕΛΑΧ.
170 IF T(I)<ELAX THEN ELAX=T(I)
180 REM ΒΗΜΑ 5B. ΕΛΕΓΞΕ ΑΝ Ο T(I) ΕΙΝΑΙ ΜΕΓΑΛΥΤ. ΤΟΥ ΜΕΓΙΣΤΟΥ
190 IF T(I)>MEG THEN MEG=T(I)
200 NEXT I
210 REM ΒΗΜΑ 6. ΕΜΦΑΝΙΣΗ ΕΛΑΧΙΣΤΟΥ
220 PRINT "ΕΛΑΧΙΣΤΗ ΘΕΡΜΟΚΡΑΣΙΑ =",ELAX
230 REM ΒΗΜΑ 7. ΕΠΑΝΕΛΑΒΕ ΤΗΝ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑ N ΦΟΡΕΣ
240 FOR I=1 TO N
250 REM ΒΗΜΑ 7A. ΕΛΕΓΞΕ ΑΝ Ο T(I) ΕΙΝΑΙ ΙΣΟΣ ΜΕ ΤΟΝ ΕΛΑΧΙΣΤΟ
260 IF T(I)=ELAX THEN PRINT M$(I)
270 NEXT I
280 REM ΒΗΜΑ 8. ΕΜΦΑΝΙΣΗ ΜΕΓΙΣΤΟΥ
290 PRINT "ΜΕΓΙΣΤΟΣ =",MEG
300 REM ΒΗΜΑ 9. ΕΠΑΝΕΛΑΒΕ ΤΗΝ ΠΑΡΑΚΑΤΩ ΔΙΑΔΙΚΑΣΙΑ N ΦΟΡΕΣ
310 FOR I=1 TO N
320 REM ΒΗΜΑ 9A. ΕΛΕΓΞΕ ΑΝ ΤΟ T(I) ΕΙΝΑΙ ΙΣΟ ΜΕ ΤΟΝ ΜΕΓΙΣΤΟ
330 IF T(I)=MEG THEN PRINT M$(I)
340 NEXT I
350 DATA "Ιανουάριος",5
360 DATA "Φεβρουάριος",10
370 DATA "Μάρτιος",12
380 DATA "Απρίλιος",17
390 DATA "Μάϊος",19
400 DATA "Ιούνιος",20
410 DATA "Ιούλιος",29
420 DATA "Αύγουστος",29
430 DATA "Σεπτέμβριος",24
```

440 DATA "Οκτώβριος",19  
 450 DATA "Νοέμβριος",15  
 460 DATA "Δεκέμβριος",5  
 470 END

### Γ) Ταξινόμηση στοιχείων πίνακα.

**Παράδειγμα.** Να διαβαστούν  $N$  αριθμοί και στην συνέχεια εφόσον τοποθετηθούν σε έναν πίνακα  $A$  να τοποθετηθούν σε αύξουσα σειρά.

**Απάντηση.**

#### A) Περιγραφή εισόδου-εξόδου

*Είσοδος* – Το πλήθος των αριθμών  $N$  καθώς και οι αριθμοί που θα τοποθετηθούν στον πίνακα  $A$ .

*Εξοδος* – Τα στοιχεία του πίνακα  $A$  ταξινομημένα κατά αύξουσα σειρά.

#### B) Παράδειγμα με το χέρι.

Ξεκινώ με ελάχιστο τον  $A(1)$  και ψάχνω να βρω τον πρώτο αριθμό από το  $A(2)$  έως το  $A(N)$  που είναι μικρότερος από αυτόν. Εάν βρω έναν τον μεταθέτω με τον  $A(1)$ . Συνεχίζω με τους υπόλοιπους αριθμούς έως το  $A(N)$  και κάνω το ίδιο. Μετά από αυτήν την διαδικασία έχει τοποθετηθεί στην θέση  $A(1)$  ο μικρότερος αριθμός. Επαναλαμβάνω την διαδικασία αυτή ξεκινώντας τώρα από το  $A(2)$  δηλ. ψάχνω να βρω τον πρώτο αριθμό από το  $A(3)$  έως το  $A(N)$  που είναι μικρότερος από αυτόν. Εάν βρω έναν τον μεταθέτω με τον  $A(2)$ . Συνεχίζω με τους υπόλοιπους αριθμούς έως το  $A(N)$  και κάνω το ίδιο. Μετά από αυτήν την διαδικασία έχει τοποθετηθεί στην θέση  $A(2)$  ο μικρότερος αριθμός από το  $A(2)$  έως το  $A(N)$ . Επαναλαμβάνω την διαδικασία αυτή ξεκινώντας τώρα από το  $A(3)$  κ.ο.κ.

#### Γ) Αλγόριθμος προβλήματος (Bubble sort)

*Βήμα 1.* Διαβάζω το πλήθος των αριθμών  $N$ . Εφόσον το  $N$  δεν είναι γνωστό χρησιμοποιώ την εντολή INPUT.

*Βήμα 2.* Ορίζω τον πίνακα  $A$ .

*Βήμα 3.* Διαβάζω τις τιμές του πίνακα  $A$  με την εντολή INPUT εφόσον δεν είναι γνωστές εκ των προτέρων.

*Βήμα 4.* Καθώς το I παίρνει τιμές από 1 έως και N-1 επαναλαμβάνω την παρακάτω διαδικασία :

*Βήμα 4<sup>α</sup>.* Ελέγχω αν τα στοιχεία A(J) όπου το J παίρνει τιμές από I+1 έως N είναι μικρότερο του A(I). Αν ναι τότε αλλάζω θέση στα A(I) και A(J) είτε με την εντολή SWAP A(I),A(J) είτε με τις εντολές (K=A(I):A(I)=A(J):A(J)=K).

*Βήμα 6.* Εκτυπώνω τον πίνακα A.

#### **Δ) Πρόγραμμα**

```
10 REM ΤΑΞΙΝΟΜΗΣΗ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ
20 REM ΒΗΜΑ 1. ΘΕΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
30 INPUT "Δώσε πλήθος αριθμών =",N
40 REM ΒΗΜΑ 2. ΟΡΙΣΜΟΣ ΠΙΝΑΚΑ
50 DIM A(N)
60 REM ΒΗΜΑ 3. ΔΙΑΒΑΣΜΑ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΙΝΑΚΑ
70 FOR I=1 TO N
80 PRINT "A(;"I;")=";
90 INPUT A(I)
100 NEXT I
110 REM ΒΗΜΑ 4. ΕΠΑΝΕΛΑΒΕ ΓΙΑ I=1 ΕΩΣ N-1
120 FOR I=1 TO N-1
130 REM ΒΗΜΑ 4Α. ΕΠΑΝΕΛΑΒΕ ΓΙΑ J=I+1 ΕΩΣ N
140 FOR J=I+1 TO N
150 IF A(J)<A(I) THEN SWAP A(I),A(J)
160 NEXT J
170 NEXT I
180 REM ΒΗΜΑ 5. ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
190 FOR I=1 TO N
200 PRINT A(I)
210 NEXT I
220 END
```

**Σημείωση :** Εάν η άσκηση ζητούσε την ταξινόμηση κατά φθίνουσα σειρά η εντολή 150 θα έπρεπε να αντικατασταθεί από την εντολή :

150 IF A(J)>A(I) THEN SWAP A(I),A(J)

**Σημείωση :** Το παραπάνω πρόγραμμα ταξινόμησης μπορεί να χρησιμοποιηθεί και για πίνακες που δέχονται αλφαριθμητικές σταθερές και όχι μόνο αριθμούς όπως για παράδειγμα η ταξινόμηση σε αύξουσα αλφαβητική σειρά 30 ονομάτων. Η παραπάνω μέθοδος λέγεται και μέθοδος της φυσαλίδας, διότι ως φυσαλίδες το ένα ελάχιστο στοιχείο μετά το άλλο αναδύονται στις θέσεις του πίνακα.

**Παράδειγμα.** Η κατανομή των ενεργών δασικών συνεταιρισμών σε γεωγραφικά διαμερίσματα (1982) δίδεται από τον παρακάτω πίνακα :

A/A	Γεωγραφικά Διαμερίσματα	Αριθμός Δασικών Συν/μων
1	ΘΡΑΚΗ	33
2	ΑΝΑΤΟΛΙΚΗ ΜΑΚΕΔΟΝΙΑ	68
3	ΚΕΝΤΡΙΚΗ ΜΑΚΕΔΟΝΙΑ	62
4	ΔΥΤΙΚΗ ΜΑΚΕΔΟΝΙΑ	35
5	ΗΠΕΙΡΟΣ	60
6	ΘΕΣΣΑΛΙΑ	87
7	ΣΤΕΡΕΑ ΕΛΛΑΔΑ	91
8	ΠΕΛΟΠΟΝΝΗΣΟΣ	10
9	ΝΗΣΙΑ	0

Να γραφεί πρόγραμμα που θα ταξινομεί τα παραπάνω γεωγραφικά διαμερίσματα σε φθίνουσα σειρά ως προς τον αριθμό των Δασικών Συνεταιρισμών.

**Απάντηση.**

#### **A) Περιγραφή εισόδου-εξόδου**

*Είσοδος* – Το πλήθος των αριθμών των γεωγραφικών διαμερισμάτων  $N=9$  καθώς και τα ονόματα των γεωγραφικών διαμερισμάτων και το αντίστοιχο πλήθος των δασικών συνεταιρισμών.

*Έξοδος* – Τα παραπάνω στοιχεία του πίνακα ταξινομημένα κατά φθίνουσα σειρά ως προς τα πλήθος των δασικών συνεταιρισμών ανά γεωγραφικό διαμέρισμα.

### **B) Παράδειγμα με το χέρι.**

Θα πρέπει να εφαρμόσω τον ίδιο αλγόριθμο ταξινόμησης με το προηγούμενο παράδειγμα με την μόνη διαφορά ότι : α) θα χρησιμοποιήσω έναν πίνακα για τα γεωγραφικά διαμερίσματα και έναν για το αντίστοιχο πλήθος των δασικών συνεταιρισμών και β) όταν θα αλλάζω τις θέσεις των στοιχείων του ενός πίνακα θα εκτελώ την ίδια αλλαγή και στα αντίστοιχα στοιχεία του άλλου πίνακα.

### **Γ) Αλγόριθμος προβλήματος (Bubble sort)**

*Βήμα 1.* Το πλήθος των γεωγραφικών διαμερισμάτων είναι  $N=9$ .

*Βήμα 2.* Ορίζω τον πίνακα  $O\$$  που θα δεχθεί τα ονόματα των γεωγραφικών διαμερισμάτων και  $A$  τον πίνακα που θα δεχθεί τα αντίστοιχα πλήθη των δασικών συνεταιρισμών.

*Βήμα 3.* Διαβάζω τις τιμές των πινάκων  $O\$$  και  $A$  μέσω της εντολής READ-DATA διότι τις τιμές των πινάκων τις γνωρίζω εκ των προτέρων.

*Βήμα 4.* Καθώς το  $I$  παίρνει τιμές από 1 έως και  $N-1$  επαναλαμβάνω την παρακάτω διαδικασία :

*Βήμα 4<sup>α</sup>.* Ελέγχω αν τα στοιχεία  $A(J)$  όπου το  $J$  παίρνει τιμές από  $I+1$  έως  $N$  είναι μεγαλύτερο του  $A(I)$ . Αν ναι τότε αλλάζω θέση στα  $A(I)$  και  $A(J)$  αλλά ταυτόχρονα αλλάζω θέσεις και στα  $O\$(I)$  και  $O\$(J)$ .

*Βήμα 6.* Εκτυπώνω τους πίνακες  $O\$$  και  $A$ .

### **Δ) Πρόγραμμα**

```
10 REM ΤΑΞΙΝΟΜΗΣΗ ΣΤΟΙΧΕΙΩΝ ΔΥΟ ΠΙΝΑΚΩΝ
20 REM ΒΗΜΑ 1. ΘΕΣΕ ΠΛΗΘΟΣ ΑΡΙΘΜΩΝ
30 N=9
40 REM ΒΗΜΑ 2. ΟΡΙΣΜΟΣ ΠΙΝΑΚΩΝ
50 DIM O$(N), A(N)
60 REM ΒΗΜΑ 3. ΔΙΑΒΑΣΜΑ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΠΙΝΑΚΑ
70 FOR I=1 TO N
```



```
80 READ O$(I), A(I)
90 NEXT I
100 REM ΒΗΜΑ 4. ΕΠΑΝΕΛΑΒΕ ΓΙΑ I=1 ΕΩΣ N-1
110 FOR I=1 TO N-1
120 REM ΒΗΜΑ 4Α. ΕΠΑΝΕΛΑΒΕ ΓΙΑ J=I+1 ΕΩΣ N
130 FOR J=I+1 TO N
140 IF A(J)>A(I) THEN SWAP A(I),A(J) : SWAP O$(I),O$(J)
150 NEXT J
160 NEXT I
170 REM ΒΗΜΑ 5. ΕΜΦΑΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ
180 FOR I=1 TO N
190 PRINT O$(I), A(I)
200 NEXT I
210 DATA "ΘΡΑΚΗ", 33
.....
290 DATA "ΝΗΣΙΑ",0
300 END
```

## Επαναληπτικές Ασκήσεις στους Πίνακες

**Άσκηση 1.** Αν ο αριθμός των βροχομετρικών σταθμών σε μια περιοχή είναι N. Να διαβαστούν τα ύψη βροχής που μετρήθηκαν σε κάθε σταθμό σε μια ορισμένη χρονική περίοδο και να υπολογιστεί το μέσο ύψος βροχής της περιοχής.

**Άσκηση 2.** Εφόσον τοποθετηθούν τα παρακάτω δεδομένα σε πίνακες :

Υδρολογικό Έτος	Ύψη βροχόπτωσης σε mm
1960	22.35
1961	35.56
1962	40.64
1963	54.10
1964	48.77
1965	41.91
1966	55.88;
1967	49.53
1968	65.53
1969	33.53

να γραφεί πρόγραμμα που να ταξινομεί τα παραπάνω στοιχεία σε φθίνουσα σειρά ως προς το ύψος βροχόπτωσης σε mm.

**Άσκηση 3.** Το ποσοστό της υγρασίας εδάφους % (του ξηρού βάρους) σε ένα δάσος ερυθρελάτης δίνεται σε σχέση με το βάθος του εδάφους ως εξής :

Βάθος εδάφους σε cm	Υγρασία εδάφους %
0	12.7
10	10.0
20	9.4
30	9.1
40	9.5
60	9.2
120	10.5

Να βρεθεί ο μέσος όρος της υγρασίας εδάφους.

**Άσκηση 4.** Η κατανομή της έκτασης των δασών της χώρας μας φαίνεται στον παρακάτω πίνακα :

Μορφές εδαφοπονίας	Έκταση στρεμ.
1. Δάση	25.124.180
2. Μερικώς δασοσκεπείς εκτάσεις	32.421.400
3. Φρυγανότοποι	2.773.135
4. Αλπικές εκτάσεις	4.400.577
5. Χορτολίβαδα	17.555.073
6. Έλη-Λίμνες-Ποταμοί	2.728.620
7. Άγωνα	7.348.513
8. Γεωργικές καλλιέργειες	39.638.500

Να γραφεί πρόγραμμα που να τοποθετεί τα παραπάνω στοιχεία σε δύο πίνακες D\$ και E και στην συνέχεια :

- A) να υπολογίζει την συνολική έκταση των δασών στην χώρα μας,
- B) να εμφανίζει τα δασοπονικά είδη με την ελάχιστη και μέγιστη έκταση.
- Γ) να ταξινομεί τα παραπάνω δασοπονικά είδη σε φθίνουσα σειρά ως προς την έκταση που καταλαμβάνουν

**Άσκηση 5.** Έστω  $d_0, d_1$  οι διάμετροι στην βάση και στο σημείο αποκορύφωσης ενός δέντρου. Αν το συνολικό μήκος του δέντρου είναι  $\ell$ , τότε ο όγκος  $V$  του δέντρου δίνεται από τον τύπο :

$$V = \frac{\pi}{4} \times \ell \times \left( \frac{d_0^2 + d_1^2}{2} \right)$$

Να διαβαστούν οι τιμές  $d_0$  και  $d_1$  και  $\ell$  από 20 δέντρα και να τοποθετηθούν σε 3 πίνακες D0, D1 και L αντίστοιχα. Στην συνέχεια να βρεθεί ο όγκος για το κάθε δέντρο και να τοποθετηθεί σε έναν καινούργιο πίνακα V. Τέλος να υπολογιστούν τα εξής :

- A) ο μέσος όγκος των δέντρων,
- B) να ταξινομηθεί ο πίνακας V σε φθίνουσα σειρά.

**Άσκηση 6.** Να διαβαστούν N τιμές και να τοποθετηθούν σε έναν πίνακα A. Στην συνέχεια να εμφανισθούν :

- α) το άθροισμα των άρτιων αριθμών του A, που ανήκουν στο κλειστό διάστημα  $[-10, 10]$ ,
- β) το πλήθος των αριθμών που είναι περιττοί,

- γ) το γινόμενο των αριθμών που δεν ανήκουν στο διάστημα  $[-1, 1]$ ,
- δ) ο μέσος όρος των θετικών αριθμών που είναι πολλαπλάσια του 5.

**Άσκηση 7.** Να διαβαστούν  $N$  αριθμοί και να τοποθετηθούν σε έναν πίνακα. Μετά να βρεθεί και να εμφανιστεί ο μέσος όρος των αριθμών που βρίσκονται στο διάστημα  $[-50, 50]$ .

**Άσκηση 8.** Να διαβαστούν  $N$  αριθμοί και να τοποθετηθούν σε έναν πίνακα  $A$ . Μετά να βρεθούν και εμφανιστούν τα εξής :

- α) Ο μεγαλύτερος αριθμός.
- β) Το πλήθος των ζυγών αριθμών που βρίσκονται στις πρώτες θέσεις του πίνακα.

**Άσκηση 9.** Να διαβαστούν  $N$  αριθμοί και να τοποθετηθούν σε έναν πίνακα  $A$ . Μετά να διαβαστεί ένας αριθμός από έξω και να βρεθεί αν υπάρχει ή όχι ο αριθμός αυτός στον πίνακα καθώς και σε ποια θέση αν υπάρχει.

**Άσκηση 10.** Να διαβαστούν τα ονοματεπώνυμα και οι βαθμοί ενός μαθήματος  $N$  μαθητών και να τοποθετηθούν με την σειρά σε δύο πίνακες  $B$  και  $O\Phi$ . Μετά να εμφανισθούν τα ονόματα των μαθητών που πέρασαν το μάθημα.

**Άσκηση 11.** Να διαβαστούν  $N$  αριθμοί και να εμφανιστούν ταξινομημένοι κατά αύξουσα σειρά.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Λάζος Κ., BASIC και MS-DOS, Θεσσαλονίκη 1994.
2. Etter D. M., Structured Fortran 77 for Engineering and Scientists, The Benjamin/Cummings Publishing Company Inc., 1993.
3. Βουτυράς Γ., Μυλωνάκης Γ. και Ρουβάς Α., Γλώσσα Προγραμματισμού BASIC, ΟΕΔΒ.
4. Καρακός Α., Μικροπολογιστές και η γλώσσα προγραμματισμού BASIC με παραδείγματα., 1986.
5. Αραμπατζόγλου Λ., Μαθήματα Δενδρομετρίας I & II (Θεωρία και Εργαστήριο), ΤΕΙ Καβάλας, Δράμα 1996.
6. Κατσιαμπούρα Δ.Α., Σημειώσεις Διευθέτησης Ορεινών Υδάτων, ΤΕΙ Λάρισας, Καρδίτσα 1996.
7. Εμμανουλούδης Δ. & Εμμανουηλίδης Δ., Διευθέτηση χειμαρρικών ρευμάτων II (Θεωρία-Εργαστήριο), ΤΕΙ Καβάλας, Δράμα 1992.
8. Μέρου Θ., Σημειώσεις δασικής υδρολογίας, ΤΕΙ Καβάλας, Δράμα 1996.