ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΗΝ ΠΡΑΞΗ ΜΕ ΤΗΝ ΓΛΩΣΣΑ LOGO

1.1 Εισαγωγή στη Γλώσσα Προγραμματισμού LOGO

Η Logo είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, που προέρχεται από την Lisp. Όπως κάθε άλλη γλώσσα προγραμματισμού έχει αλφάβητο, λεξιλόγιο και συντακτικό.

Στην Logo υπάρχουν δυο κατηγορίες λέξεων: α) λέξεις που ήδη γνωρίζει η Logo επειδή είναι ενσωματωμένες σε αυτή και εκτελούν μια συγκεκριμένη ενέργεια. Οι λέξεις αυτές ονομάζονται **πρωτογενείς διαδικασίες** β) λέξεις τις οποίες ορίζουμε εμείς και λέγονται απλά **διαδικασίες**. Στις λέξεις που εμείς δημιουργούμε δεν μπορούμε να χρησιμοποιήσουμε τις ενσωματωμένες λέξεις γιατί αυτές είναι δεσμευμένες από την γλώσσα. Πάντως και οι πρωτογενείς διαδικασίες και οι απλές διαδικασίες λειτουργούν με τον ίδιο τρόπο.

Όλες οι λέξεις της Logo μπορούν να αποτελούν εντολές και να εκτελούν κάποια ενέργεια. Για να εκτελεστεί όμως μια ενέργεια πρέπει να ακολουθήσουμε τους γραμματικούς και συντακτικούς κανόνες της Logo. Συγκεκριμένα πρέπει να δίνουμε σωστές οδηγίες γράφοντας πάντα πρώτα την **εντολή** που θέλουμε να εκτελέσουμε και αμέσως μετά τα **ορίσματά** της, δηλαδή τις εισόδους που χρειάζεται η εντολή για να ενεργήσει. Πχ στην οδηγία forward 100, η λέξη forward είναι η εντολή, ενώ ο αριθμός 100 το όρισμά της. Η οδηγία έχει σαν αποτέλεσμα την μετακίνηση της χελώνας προς τα εμπρός κατά 100 «βήματα».

Η Logo χειρίζεται τρεις τύπους δεδομένων: **αριθμούς**, **λέξεις** και **λίστες**. Στους αριθμούς δεν υπάρχει διαχωρισμός ανάμεσα σε ακέραιους και πραγματικούς. Ένας αριθμός αποτελείται από πρόσημο, ακέραιο μέρος, υποδιαστολή και δεκαδικό μέρος. Το πρόσημο μπορεί να παραληφθεί αν ο αριθμός είναι θετικός. Επίσης η υποδιαστολή μπορεί να παραληφθεί αν δεν υπάρχει δεκαδικό μέρος. Προσοχή στο αρνητικό πρόσημο, που πρέπει να γραφεί ακριβώς αριστερά από τον αριθμό, αλλιώς η Logo θα το εκλάβει σαν το σύμβολο της αφαίρεσης.

Μια λέξη αποτελείται από συνδυασμό χαρακτήρων του αλφαβήτου της γλώσσας. Επομένως μπορεί να περιλαμβάνει γράμματα, και αριθμητικά ψηφία με οποιαδήποτε σειρά και συχνότητα. Οι λέξεις ξεχωρίζουν μεταξύ τους με την παρεμβολή ενός κενού διαστήματος. Μια λέξη που ξεκινά με " λέει στη Logo πως αποτελεί όρισμα μιας εντολής. Τα " δεν απαιτούνται και στο τέλος της λέξης.

Λίστα είναι μια σειρά από λέξεις που περικλείονται σε αγκύλες. Σε μια λίστα κάθε λέξη μπορεί να είναι μια άλλη λίστα.

Μεταβλητή της Logo είναι μια θέση της μνήμης για να αποθηκεύουμε δεδομένα για να μπορούμε να τα επεξεργαστούμε κατά την διάρκεια εκτέλεσης του προγράμματος. Η Logo υποστηρίζει τον **δομημένο προγραμματισμό**, δηλαδή διαθέτει δομές ακολουθίας, επανάληψης και επιλογής. Στη δομή ακολουθίας οι εντολές εκτελούνται με την σειρά που είναι γραμμένες, η μια μετά την άλλη. Στη δομή επανάληψης οι εντολές επαναλαμβάνονται με την σειρά που είναι γραμμένες πολλές φορές. Στη δομή επιλογής η εκτέλεση των εντολών εξαρτάται από κάποια συνθήκη.

Για να εκτελέσουμε ένα πρόγραμμα γράφουμε τις εντολές στο **κέντρο εντολών** και τις εκτελούμε μια προς μια. Εναλλακτικά μπορούμε να τις γράψουμε όλες στην καρτέλα **Διαδικασίες** και να τις εκτελέσουμε από το κέντρο εντολών γράφοντας απλά το όνομα της διαδικασίας.

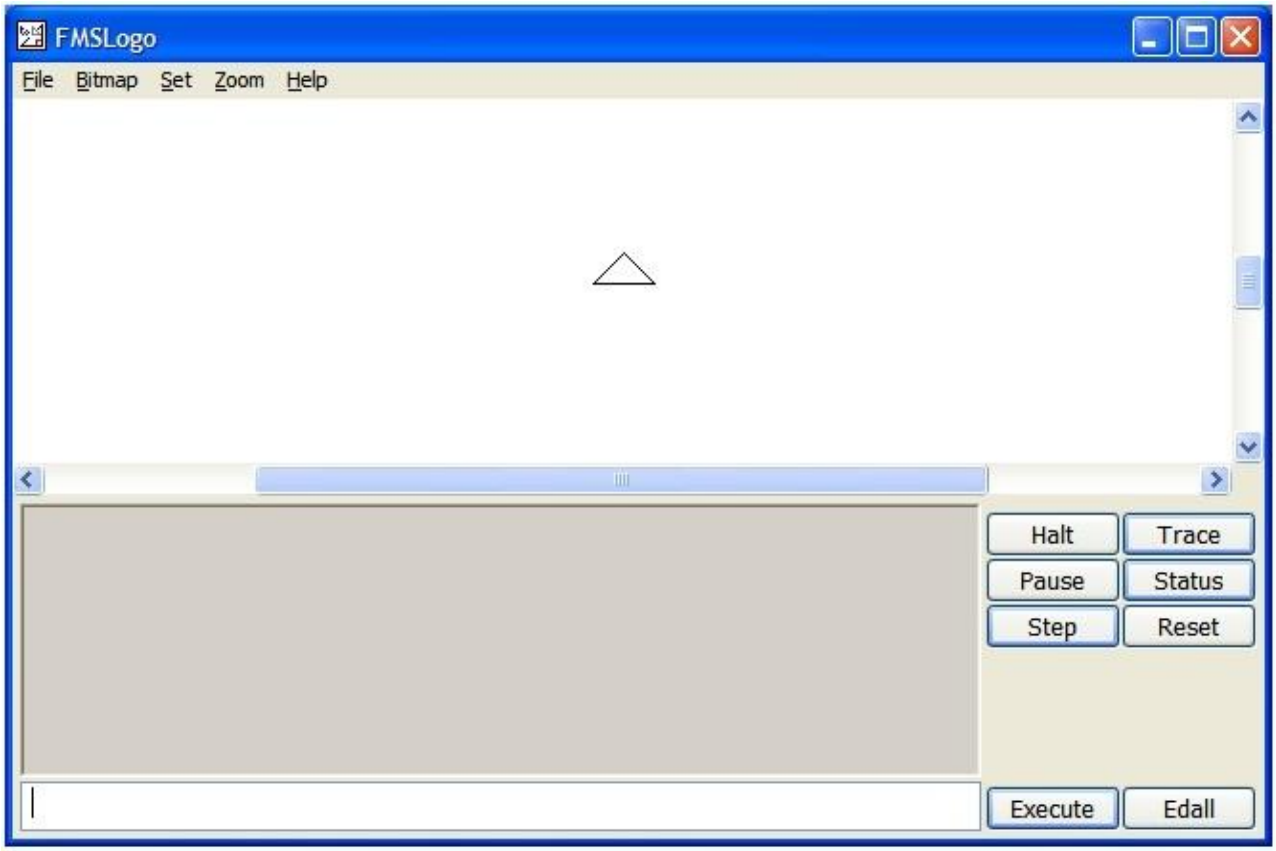
1.2 Το Περιβάλλον Προγραμματισμού FmsLogo

Με την εκκίνηση του προγράμματος εμφανίζονται στην οθόνη η γνωστή γραμμή μενού καθώς και η γραμμή κατάστασης όπως και σε κάθε άλλη παραθυρική εφαρμογή. Η υπόλοιπη οθόνη χωρίζεται σε δυο μέρη, στην επιφάνεια εργασίας και στο κέντρο εντολών. Στην επιφάνεια εργασίας εμφανίζεται και η «χελώνα» με μορφή τριγώνου.

Για να ξαναδουλέψουμε μια εργασία που έχουμε φτιάξει παλαιότερα επιλέγουμε από το μενού: Αρχείο – Φόρτωση (File – Load). Στο παράθυρο που θα ανοίξει επιλέγουμε φάκελο και όνομα εργασίας και πατάμε το πλήκτρο Άνοιγμα. Έτσι μπορούμε στην υπάρχουσα εργασία να προσθέσουμε εντολές και μεταβλητές.

Προσοχή! Αν επιλέξουμε από το μενού Αρχείο – Άνοιγμα (File – Open) τότε διαγράφεται η προηγούμενη εργασία και ανοίγει νέα εργασία.

Επιλέγοντας Αρχείο –Αποθήκευση (File – Save As) αποθηκεύουμε την εργασία που κάναμε σε κάποιο φάκελο με κάποιο όνομα.



1.3 Εντολές Εισόδου-Εξόδου και Αριθμητικές Πράξεις

Με τις εντολές εισόδου-εξόδου μπορούμε να εισάγουμε δεδομένα αλλά και να εξάγουμε τα αποτελέσματα της επεξεργασίας τους. Οι εντολές αυτές είναι:

Εντολή **show**: εμφανίζει μια λέξη ή μια λίστα στο κέντρο εντολών. Επίσης μπορεί να εμφανίσει το αποτέλεσμα αριθμητικών πράξεων ή παραστάσεων. Παραδείγματα:

- Show 5
- Show "Καλημέρα
- Show [σήμερα είναι Τετάρτη]
- Show 5 + 12
- Show 28 – 13
- Show 12 * 13
- Show 12 / (2 * 3)
- Show sum 5 12
- Show difference 28 17
- Show product 10 12
- Show quotient 9 3
- Show remainder 31 10
- Show sqrt 100
- Show power 3 4
- Show random 15
- Show (Sentence "A "sentence "is "simply "a "list "of "words)
- Show (Sentence "A "sentence [is simply a] "list [of words])

Εντολή **messagebox**: εμφανίζει μηνύματα στην επιφάνεια εργασίας υπό μορφή πλαισίων και συντάσσεται ως εξής: messagebox [Τίτλος] [Μήνυμα]. Η εντολή ολοκληρώνεται πατώντας σε κάποιο από τα πλήκτρα OK και Άκυρο (Cancel). Αν πατηθεί το Άκυρο τότε διακόπτεται η εκτέλεση του προγράμματος, σε περίπτωση βέβαια που η εντολή είναι μέρος ενός προγράμματος.
Παραδείγματα:

- Messagebox [Χαιρετισμός] [Καλημέρα!]
- Messagebox [This is the title] [This is the message. Press OK.]

Εντολή **questionbox**: ανοίγει ένα πλαίσιο διαλόγου στην οθόνη όπου εμφανίζεται η ερώτηση που ακολουθεί την εντολή. Το πλαίσιο περιέχει ένα πεδίο που μπορούμε να πληκτρολογήσουμε την απάντησή μας στην ερώτηση. Η εντολή συντάσσεται ως εξής: questionbox [Τίτλος] [Ερώτηση]. Η εντολή ολοκληρώνεται πατώντας σε κάποιο από τα πλήκτρα OK και Άκυρο (Cancel). Αν πατηθεί το Άκυρο τότε διακόπτεται η εκτέλεση του προγράμματος, σε περίπτωση βέβαια που η εντολή είναι μέρος ενός προγράμματος.
Παραδείγματα:

- show questionbox [Ερώτηση] [Τι μέρα είναι;]
- show questionbox [This is the title] [This is the question. Answer and Press OK.]

Εντολή **readword**: εμφανίζει ένα πλαίσιο διαλόγου εν αναμονή μιας απάντησης. Την απάντηση που θα δοθεί μπορούμε να την καταχωρήσουμε σε μια μεταβλητή με χρήση της εντολής **make**.

Παράδειγμα: make "x readword
show :x

1.4 Σχεδιασμός Γεωμετρικών Σχημάτων

Για να σχεδιάσουμε σχήματα στην επιφάνεια εργασίας χρησιμοποιούμε εντολές που αναγκάζουν την χελώνα να μετακινηθεί αφήνοντας το ίχνος της. Οι εντολές αυτές είναι:

- Εντολή **forward** ή **fd**: μετακινεί την χελώνα προς τα εμπρός (δηλαδή προς την κατεύθυνση που κοιτάει η μύτη του τριγώνου). Ο αριθμός που ακολουθεί δηλώνει την απόσταση που θα μετακινηθεί η χελώνα σε pixels (εικονοστοιχεία). Π.χ. fd 100
- Εντολή **back** ή **bk**: μετακινεί την χελώνα προς τα πίσω (δηλαδή προς την κατεύθυνση την αντίθετη από αυτή που κοιτάει η μύτη του τριγώνου). Ο αριθμός που ακολουθεί δηλώνει την απόσταση που θα μετακινηθεί η χελώνα σε pixels (εικονοστοιχεία). Π.χ. bk 100
- Εντολή **right** ή **rt**: αλλάζει την κατεύθυνση της χελώνας στρίβοντάς την προς τα δεξιά. Ο αριθμός που ακολουθεί δηλώνει τις μοίρες που θα στρίψει η χελώνα προς τα δεξιά. Π.χ. rt 90
- Εντολή **left** ή **lt**: αλλάζει την κατεύθυνση της χελώνας στρίβοντάς την προς τα αριστερά. Ο αριθμός που ακολουθεί δηλώνει τις μοίρες που θα στρίψει η χελώνα προς τα αριστερά. Π.χ. lt 90
- Εντολή **penup** ή **pu**: αναγκάζει την χελώνα να μην αφήνει ίχνος κατά την κίνησή της
- Εντολή **pendown** ή **pd**: αναγκάζει την χελώνα να αφήνει ίχνος κατά την κίνησή της
- Εντολή **clearscreen** ή **cs**: σβήνει ότι έχουμε σχεδιάσει στην επιφάνεια εργασίας και επαναφέρει την χελώνα στο κέντρο με κατεύθυνση προς τα επάνω
- Εντολή **cleartext** ή **ct**: σβήνει όλες τις εντολές που έχουμε γράψει στο κέντρο εντολών
- Εντολή **setscreencolor**: αλλάζει το φόντο στην επιφάνεια εργασίας. Οι 3 αριθμοί που ακολουθούν μέσα σε αγκύλες είναι ο συνδυασμός των τριών βασικών χρωμάτων RGB. Κάθε απόχρωση αντιστοιχεί σε διαφορετικό συνδυασμό. Καθένας από τους τρεις αριθμούς παίρνει τιμές από 0 μέχρι 255. Π.χ. setscreencolor [0 0 255] (χρώμα μπλε). Οι 3 αριθμοί για τους συνδυασμούς RGB μπορούν να αντικατασταθούν με ένα αριθμό (δείκτη) από 0 μέχρι 15 για τα βασικά χρώματα.
- Εντολή **setfloodcolor**: αλλάζει το χρώμα της επιφάνειας ενός κλειστού σχήματος. Οι 3 αριθμοί που ακολουθούν μέσα σε αγκύλες είναι ο συνδυασμός των τριών βασικών χρωμάτων RGB.. Π.χ. setfloodcolor [155 96 59] (χρώμα καφέ). Προσοχή! για να ενεργήσει η εντολή θα πρέπει η χελώνα να βρίσκεται μέσα στο κλειστό σχήμα του οποίου το χρώμα θέλουμε να αλλάξουμε και στη συνέχεια να δοθεί η εντολή **fill** π.χ.
    - CLEARSCREEN
    - REPEAT 4 [FORWARD 100 RIGHT 90]
    - RIGHT 45
    - PENUP
    - FORWARD 20
    - SETFLOODCOLOR [0 255 0]
    - FILL
- Εντολή **setpencolor**: αλλάζει το χρώμα του ίχνους που αφήνει η χελώνα. Οι 3 αριθμοί που ακολουθούν μέσα σε αγκύλες είναι ο συνδυασμός των τριών βασικών χρωμάτων RGB.. Π.χ. setpencolor [255 163 0] (χρώμα πορτοκαλί)
- Εντολή **setpensize**: αλλάζει το πάχος του ίχνους που αφήνει η χελώνα. Ο αριθμός που ακολουθεί καθορίζει το πάχος της γραμμής.
- Εντολές **setX**, **setY**, **setXY**: τοποθετούν την χελώνα σε συγκεκριμένο σημείο στην επιφάνεια εργασίας. Μπορούμε να μετακινήσουμε την χελώνα μόνο οριζόντια (άξονας Χ) ή μόνο κατακόρυφα (άξονας Υ) ή ταυτόχρονα και στους δυο άξονες. Π.χ. setXY -465 -305 (τοποθέτηση της χελώνας κοντά στην κάτω αριστερά γωνία), setXY 0 0 (τοποθέτηση της χελώνας στο κέντρο). Οι συντεταγμένες της κάτω αριστερά γωνίας είναι [-500 -500], όχι όμως για όλες τις οθόνες
- Εντολή **fill**: γεμίζει ένα κλειστό σχήμα ή ολόκληρη την οθόνη με το χρώμα που είναι επιλεγμένο. Για να λειτουργήσει η εντολή πρέπει να έχουμε τοποθετήσει την χελώνα μέσα στο σχήμα
- Εντολή **circle**: σχεδιάζει κύκλο με κέντρο την τρέχουσα θέση της χελώνας. Ο αριθμός που ακολουθεί είναι η ακτίνα του κύκλου. Προσοχή! η χελώνα δεν μετακινείται κατά την εκτέλεση της εντολής. Π.χ. circle 100
- Εντολή **circle2**: σχεδιάζει κύκλο μετακινώντας την χελώνα κατά μήκος της περιφέρειας του κύκλου που θα σχεδιασθεί, κατά την φορά των δεικτών του ρολογιού. Η χελώνα θα επιστρέψει στην αρχική της θέση μετά την ολοκλήρωση της εντολής. Π.χ. circle2 100
- Εντολή **wait**: καθυστερεί την εκτέλεση της επόμενης εντολής για περίπου 1/60 του δευτερολέπτου. Π.χ. wait 600 (καθυστέρηση 10 δευτερολέπτων)

1.5 Δομή Επανάληψης

Ένα από τα πλεονεκτήματα του υπολογιστή έναντι του ανθρώπου είναι η επαναληπτικότητα των διαδικασιών. Μπορεί δηλαδή να επαναλαμβάνει κάποιες διαδικασίες χωρίς να κουράζεται ή να μπερδεύεται. Πλήθος προβλημάτων μπορούν να λυθούν εκτελώντας την ίδια ή περισσότερες ενέργειες πολλές φορές. Π,χ. για τον σχεδιασμό ενός τετραγώνου πρέπει να δώσουμε τις εξής εντολές:

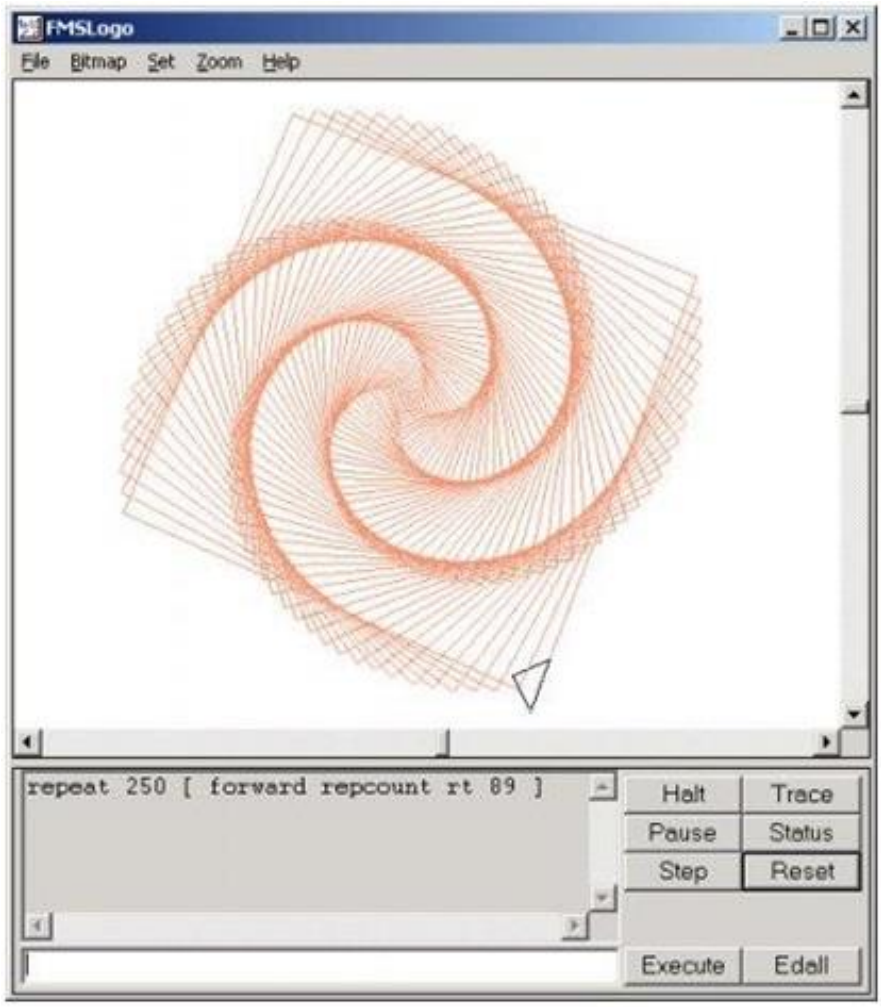
```
fd 100 rt 90
fd 100 rt 90
fd 100 rt 90
fd 100 rt 90
```

Όπως φαίνεται οι οδηγίες fd 100 rt 90 εκτελούνται 4 φορές, μια για κάθε πλευρά του τετραγώνου. Με την εντολή **repeat** η Logo μπορεί να εκτελέσει ένα σύνολο εντολών που επαναλαμβάνονται, όσες φορές θέλουμε εμείς. Έτσι ισοδύναμα μπορούμε να γράψουμε: **repeat 4 [fd 100 rt 90].** Ο αριθμός που γράφεται μετά την λέξη repeat δηλώνει το πλήθος των επαναλήψεων, ενώ μέσα στις αγκύλες γράφουμε την λίστα των εντολών που θέλουμε να επαναληφθούν. Φυσικά είναι δυνατόν μέσα στην αγκύλες να υπάρχει μια άλλη εντολή repeat. Εύκολα αντιλαμβανόμαστε ότι για το τετράγωνο η χελώνα στρίβει 4 φορές, κατά 90° την φορά, και έτσι συμπληρώνει μια πλήρη περιστροφή (4 X 90° = 360°). Συνεπώς για το τετράγωνο χωρίζω το 360 δια 4, για το πεντάγωνο θα πρέπει να χωρίσω δια 5, για το τρίγωνο δια 3 κ.ο.κ.

Σημείωση: με την εντολή repeat 360 [fd 1 rt 1] κατασκευάζουμε σχήμα με 360 πλευρές, με κάθε πλευρά να διαφέρει κατά 1ο από την προηγούμενη. Πρακτικά αυτό σημαίνει ότι φτιάχνουμε κύκλο.

Η εντολή repeat ελέγχεται εσωτερικά από την LOGO με τον μετρητή **repcount**. Ο μετρητής αυτός κρατά τον αριθμό των επαναλήψεων που έχουν γίνει μέχρι και την τρέχουσα, δηλαδή στην 1η επανάληψη έχει την τιμή 1, στην 2η την τιμή 2 κ.ο.κ. Η τρέχουσα τιμή του μετρητή μπορεί να χρησιμοποιηθεί σαν όρισμα σε εντολές της LOGO και να μας βοηθήσει να φτιάξουμε εντυπωσιακά σχήματα όπως αυτό της παρακάτω εικόνας:

```
repeat 250 [fd repcount rt 89]
```



Δοκιμάστε τις παρακάτω δομές επανάληψης και σχεδιάστε μερικά ωραία σχήματα:

```
repeat 55 [ repeat 360 [ fd 1 rt 1] rt 10 ]
repeat 12 [ repeat 4 [ fd 100 rt 90] rt 30 ]
repeat 7 [fd 100 rt 90  repeat 3 [ fd 50 rt 90] bk 50 rt 51 ]
```

Προσπαθήστε να εξηγήσετε τι συμβαίνει στην τελευταία εντολή repeat.

1.6 Διαδικασίες

Η Logo μας δίνει την δυνατότητα να αποθηκεύσουμε ένα σύνολο εντολών για μελλοντική χρήση, δίνοντας του ένα όνομα. Η δυνατότητα αυτή είναι η δημιουργία των διαδικασιών.

Πλεονεκτήματα χρήσης διαδικασιών:

- Διευκόλυνση ανάπτυξης προγραμμάτων
- Διευκόλυνση κατανόησης και διόρθωσης του προγράμματος
- Ξόδεμα λιγότερου χρόνου και κόπου για την συγγραφή προγραμμάτων
- Επέκταση των δυνατοτήτων των γλωσσών προγραμματισμού

Για να γράψουμε μια διαδικασία πατάμε το πλήκτρο **Edall** (βρίσκεται κάτω δεξιά στην οθόνη), οπότε ανοίγει το παράθυρο του editor (συντάκτης). Κάθε διαδικασία ξεκινά με την λέξη to και ακολουθεί το όνομά της. Το όνομα δεν μπορεί να περιέχει κενά και προφανώς δεν πρέπει να είναι μια από τις δεσμευμένες λέξεις της Logo. Αλλάζουμε γραμμή πατώντας το πλήκτρο Enter. Γράφουμε τις εντολές που περιέχει η διαδικασία, κατά προτίμηση μια εντολή σε κάθε σειρά. Ολοκληρώνουμε την διαδικασία γράφοντας την λέξη end. Από το μενού του editor επιλέγουμε File/Save and Exit, οπότε επιστρέφουμε στο παράθυρο της Logo. Για να καλέσουμε την διαδικασία, απλά πληκτρολογούμε το όνομά της στο κέντρο εντολών. Οι διαδικασίες αποθηκεύονται μαζί με την εργασία μας και φυσικά μπορούμε να τις μεταφέρουμε και σε άλλες εργασίες.

Παράδειγμα:

```
to tetragono
    repeat 4 [fd 50 rt 90]
end
```

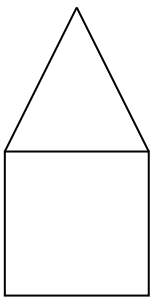
Η διαδικασία αυτή δημιουργεί τετράγωνο πλευράς 50 pixels, γράφοντας απλά την λέξη tetragono.

Είναι προφανές ότι μέσα από μια διαδικασία μπορούμε να καλούμε και άλλες διαδικασίες που έχουμε ήδη γράψει και αποθηκεύσει στην εργασία μας. Για παράδειγμα, αν θέλουμε να κατασκευάσουμε το σπίτι της παρακάτω εικόνας, παρατηρούμε ότι αποτελείται από ένα τετράγωνο και ένα τρίγωνο. Θα μπορούσαμε λοιπόν να γράψουμε στον editor τις εξής διαδικασίες:

```
to tetragono
    repeat 4 [fd 50 rt 90]
end
to trigono
    repeat 3 [fd 50 rt 120]
end
to spiti
   tetragono
   fd 50
   rt 30
   trigono
end
```

και να σχεδιάσουμε το σπίτι γράφοντας:

```
spiti
```



1.7 Μεταβλητές

Μεταβλητή είναι ένας χώρος της μνήμης του υπολογιστή που χρησιμοποιείται για να αποθηκεύουνε δεδομένα και να μπορούμε να τα επεξεργαστούμε κατά την διάρκεια εκτέλεσης του προγράμματος. Μπορούμε να φανταστούμε την μεταβλητή σαν ένα μονοθέσιο αυτοκίνητο που απέξω έχει μια ετικέτα (το **όνομα** της μεταβλητής) και μέσα χωρά μόνο ένας επιβάτης (το **περιεχόμενο** της μεταβλητής) για να μεταφερθεί εκεί που επιθυμεί. Για να μετακινηθεί ένας επιβάτης πρέπει να κατεβάσει τον προηγούμενο. Οι μεταβλητές διατηρούν τις τιμές τους, μέχρις ότου τις διαγράψουμε ή εγκαταλείψουμε την εφαρμογή.

Οι μεταβλητές στη LOGO χρησιμοποιούνται πάντα σε συνδυασμό με δυο σύμβολα, την άνω-κάτω τελεία (**:**) και τα εισαγωγικά (**"**). Όταν θέλουμε να χρησιμοποιήσουμε το όνομα της μεταβλητής, τότε πριν το όνομα βάζουμε τα **"**, ενώ όταν θέλουμε να χρησιμοποιήσουμε την τιμή της, βάζουμε πριν από το όνομα την **:**.

Οι μεταβλητές μας επιτρέπουν να φτιάχνουμε **παραμετρικές διαδικασίες**, δηλαδή να δίνουμε διαφορετική τιμή στην μεταβλητή κάθε φορά που θα καλούμε την διαδικασία. Η τιμή που θα δίνουμε κάθε φορά θα είναι αυτή που θα χρησιμοποιείται μέσα στην διαδικασία και έτσι η ίδια διαδικασία θα εκτελείται διαφορετικά κάθε φορά. Παράδειγμα: έστω ότι θέλουμε να κατασκευάσουμε ένα τετράγωνο με πλευρά 50 pixels. Θα γράφαμε τότε την ακόλουθη διαδικασία:

```
to tetragono
  repeat 4 [fd 50 rt 90]
end
```

Αν τώρα θέλαμε να κατασκευάσουμε τετράγωνο πλευράς 100 pixels θα γράφαμε:

```
to tetragono
  repeat 4 [fd 100 rt 90]
end
```

Ας δούμε τώρα πως μπορούμε να γράψουμε μια παραμετρική διαδικασία:

```
to neo_tetragono :mikos
  repeat 4 [fd :mikos rt 90]
end
```

Παρατηρούμε ότι αρκεί να αναφέρουμε δίπλα στο όνομα της διαδικασίας την μεταβλητή που πρόκειται να χρησιμοποιηθεί (και στο συγκεκριμένο παράδειγμα έχει όνομα **mikos**) και μετά σε όποιο σημείο της διαδικασίας πρόκειται να χρησιμοποιηθεί η τιμή της, απλά την αναφέρουμε. Προκειμένου βέβαια να δουλέψει με επιτυχία η διαδικασία θα πρέπει όταν την καλούμε να αναφέρουμε και την τιμή που θέλουμε να πάρει η μεταβλητή. έτσι πχ αν θέλουμε να κατασκευάσουμε τετράγωνο πλευράς 50 pixels θα γράψουμε:

```
neo_tetragono 50
```

ενώ αν θέλουμε να κατασκευάσουμε τετράγωνο πλευράς 100 pixels θα γράψουμε:

```
neo_tetragono 100
```

Αν θέλουμε να κατασκευάσουμε ορθογώνιο παραλληλόγραμμο, τότε επειδή πρέπει να προσδιορίσουμε δυο τιμές, μια για το μήκος και μια για το πλάτος, θα μπορούσαμε να γράψουμε την εξής διαδικασία:

```
to parallhlogrammo :x :y
  repeat 2 [fd :x rt 90 fd :y rt 90]
end
```

Εδώ χρησιμοποιούμε δυο μεταβλητές (με ονόματα **x** και **y)** και θα καλέσουμε την διαδικασία γράφοντας:

```
parallhlogrammo 120 70
```

και θα κατασκευάζαμε παραλληλόγραμμο με μήκος 120 pixels και πλάτος 70 pixels.

Να και μια διαδικασία κατασκευής κύκλου συγκεκριμένης ακτίνας:

```
to kyklos :x
  repeat 360 [fd :x/57.5 rt 1]
end
```

Έτσι για να φτιάξουμε κύκλο με ακτίνα 50 pixels θα γράψουμε:

```
kyklos 50
```

1.8 Δημιουργία Μεταβλητών

Η LOGO διαθέτει δυο εντολές για να μπορούμε να δημιουργούμε μεταβλητές και να τους δίνουμε διάφορες τιμές. Οι εντολές αυτές είναι:

- Εντολή **make**: η εντολή αναφέρθηκε και στην παράγραφο 2.3 σε συνδυασμό με την εντολή readword. Συντάσσεται ως εξής: make όνομα τιμή Π.χ. make "x 16 (δημιουργία μεταβλητής με όνομα x και τιμή τον αριθμό 16) make "paidi "Peter (δημιουργία μεταβλητής με όνομα paidi και τιμή τους χαρακτήρες Peter). Παρατηρήστε ότι χρησιμοποιήσαμε τα " για να αποθηκεύσουμε τους αλφαριθμητικούς χαρακτήρες.
  make "car "Mercedes
  show :car
  Mercedes
  Με την εντολή make δημιουργήσαμε μια μεταβλητή με όνομα car και τιμή (περιεχόμενο) Mercedes. Με την εντολή show ζητάμε από την LOGO να μας εμφανίσει την τιμή της μεταβλητής car.
- Εντολή **name**: Συντάσσεται ως εξής: name τιμή όνομα Π.χ. name 16 "x (δημιουργία μεταβλητής με όνομα x και τιμή τον αριθμό 16) name "Peter "paidi (δημιουργία μεταβλητής με όνομα paidi και τιμή τους χαρακτήρες Peter)

Παρατηρήστε τις επόμενες εντολές.

```
make "a 1
show :a
repeat 9 [ make "a :a + 1 show :a]
```

Το αποτέλεσμα της εκτέλεσης αυτών των εντολών είναι να εμφανιστούν οι αριθμοί από το 1 μέχρι το 10. Η μεταβλητή με όνομα a παίρνει αρχικά την τιμή 1 και στη συνέχεια σε κάθε επανάληψη της repeat αυξάνεται κατά μια μονάδα.

1.9 Σύνθετες εφαρμογές με τη LOGO

1.9.1 Δημιουργία παράθυρου

Για να φτιάξουμε ένα καινούργιο παράθυρο χρησιμοποιούμε την εντολή **windowcreate**. Το παράθυρο είναι ένα πλαίσιο μέσα στο οποίο μπορούμε να προσθέσουμε διάφορα αντικείμενα (π.χ. κουμπιά, μεταβολείς, αναδιπλούμενες λίστες κλπ). Η εντολή συντάσσεται ως εξής:

WINDOWCREATE p1 p2 p3 p4 p5 p6 p7 p8

όπου p1 το παράθυρο μέσα στο οποίο θα δημιουργηθεί το δικό μας νέο παράθυρο
p2 το όνομα του νέου παράθυρου
p3 ο τίτλος του νέου παράθυρου
p4 η συντεταγμένη του οριζόντιου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία του νέου παράθυρου
p5 η συντεταγμένη του κατακόρυφου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία του νέου παράθυρου
p6 το μήκος του νέου παράθυρου σε pixels
p7 το πλάτος του νέου παράθυρου σε pixels
p8 οι εντολές ή η διαδικασία που θα εκτελούνται με την δημιουργία του νέου παράθυρου

Π.χ. **windowcreate "main "mywindow "Αριθμομηχανή 10 10 300 300 []**

το παράθυρο μέσα στο οποίο θα δημιουργηθεί το δικό μας παράθυρο λέγεται **main** και είναι το παράθυρο της εφαρμογής της LOGO
το παράθυρό μας ονομάζεται **mywindow**
ο τίτλος του παράθυρού μας είναι **Αριθμομηχανή**
οι συντεταγμένες της πάνω αριστερά γωνίας του παράθυρού μας είναι (10, 10) (**Προσοχή!** το σημείο με συντεταγμένες (0,0) είναι η πάνω αριστερά γωνία της οθόνης. Πάντως ανεξάρτητα που θα πρωτοδημιουργήσουμε το παράθυρο, μπορούμε στη συνέχεια να το μετακινήσουμε με κλικ πάνω στον τίτλο του παράθυρου και σύρσιμο του)
οι διαστάσεις του παράθυρού μας είναι: μήκος 300 και πλάτος 300 pixels
προς το παρόν δεν έχουμε καθορίσει τις εντολές ή τα αντικείμενα που θα περιέχει το παράθυρο, για αυτό και ανάμεσα στις αγκύλες δεν έχουμε γράψει τίποτα. Μπορούμε να το κάνουμε στη συνέχεια.
Αν θέλουμε ο τίτλος του παράθυρου να είναι μια φράση πρέπει να χρησιμοποιήσουμε αγκύλες αντί για διπλά εισαγωγικά για τον τίτλο του παράθυρου.

Π.χ. **windowcreate "main "mywindow [Το παράθυρό μου] 10 10 300 300 []**

Το νέο παράθυρο που δημιουργήσαμε μπορούμε να το διαγράψουμε (κλείσουμε) με την εντολή **windowdelete**, που συντάσσεται:

WINDOWDELETE p1

όπου p1 το όνομα του παράθυρου.

Π.χ. **windowdelete "mywindow**

Προσοχή! όταν διαγράφουμε ένα παράθυρο, διαγράφουμε ταυτόχρονα και όλα τα αντικείμενα που υπήρχαν μέσα σε αυτό.

1.9.2 Δημιουργία κουμπιού

Για να φτιάξουμε ένα κουμπί μέσα σε ένα παράθυρο χρησιμοποιούμε την εντολή **buttoncreate**. Η εντολή συντάσσεται ως εξής:

BUTTONCREATE p1 p2 p3 p4 p5 p6 p7 p8

όπου p1 το παράθυρο μέσα στο οποίο θα δημιουργηθεί το κουμπί
p2 το όνομα του κουμπιού
p3 η ετικέτα του κουμπιού
p4 η συντεταγμένη του οριζόντιου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία του κουμπιού p5 η συντεταγμένη του κατακόρυφου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία του κουμπιού
p6 το μήκος του κουμπιού σε pixels
p7 το πλάτος του κουμπιού σε pixels
p8 οι εντολές ή η διαδικασία που θα εκτελούνται με το πάτημα του κουμπιού

Π.χ. **buttoncreate " mywindow "myleft "Left 25 25 25 25 [forward 2 left 1]**

το παράθυρο μέσα στο οποίο θα δημιουργηθεί το κουμπί λέγεται **mywindow**
το κουμπί ονομάζεται **myleft**
η ετικέτα του κουμπιού είναι **Left**
οι συντεταγμένες της πάνω αριστερά γωνίας του κουμπιού μας είναι (25,25) (**Προσοχή!** η πάνω αριστερά γωνία του παράθυρου θεωρείται ότι έχει συντεταγμένες (0,0))
οι διαστάσεις του κουμπιού είναι: μήκος 25 και πλάτος 25 pixels
με το πάτημα του κουμπιού η χελώνα θα προχωρήσει 2 pixels μπροστά και θα στρίψει αριστερά κατά μια μοίρα.

Αν θέλουμε η ετικέτα του κουμπιού να είναι μια φράση πρέπει να χρησιμοποιήσουμε αγκύλες αντί για διπλά εισαγωγικά για τον ετικέτα του κουμπιού.

Το κουμπί που δημιουργήσαμε μπορούμε να το διαγράψουμε με την εντολή **buttondelete**, που συντάσσεται:

BUTTONDELETE p1

όπου p1 το όνομα του κουμπιού.

Π.χ. **buttondelete "myleft**

Δοκιμάστε τις παρακάτω εντολές στο περιβάλλον της LOGO

windowcreate "main "mywindow [Το παράθυρό μου] 0 0 200 200 []
buttoncreate "mywindow "myleft "Left 25 25 25 25 [forward 2 left 1]
buttoncreate "mywindow "myright "Right 75 25 25 25 [forward 2 right 1]
buttoncreate "mywindow "close "΄Εξοδος 50 75 25 25 [windowdelete "mywindow]
Πατήστε τα κουμπιά left και right και δείτε το αποτέλεσμα της κίνησης της χελώνας
Πατήστε τα κουμπί Έξοδος για να κλείσει το παράθυρο

1.9.3 Δημιουργία ετικέτας

Για να φτιάξουμε μια ετικέτα μέσα σε ένα παράθυρο χρησιμοποιούμε την εντολή **staticcreate**. Η ετικέτα απλά εμφανίζει ένα κείμενο. Το κείμενο αυτό βέβαια μπορεί δυναμικά να αλλάξει. όση ώρα το παράθυρο είναι ανοικτό. Η εντολή συντάσσεται ως εξής:

STATICCREATE p1 p2 p3 p4 p5 p6 p7

όπου p1 το παράθυρο μέσα στο οποίο θα δημιουργηθεί η ετικέτα
p2 το όνομα της ετικέτας
p3 η το κείμενο που θα εμφανίζεται πάνω στην ετικέτα
p4 η συντεταγμένη του οριζόντιου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία της ετικέτας
p5 η συντεταγμένη του κατακόρυφου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία της ετικέτας
p6 το μήκος της ετικέτας σε pixels
p7 το πλάτος της ετικέτας σε pixels

Π.χ. **staticcreate " mywindow "mystatic "tampela 25 25 50 25**

το παράθυρο μέσα στο οποίο θα δημιουργηθεί το κουμπί λέγεται **mywindow**
η ετικέτα ονομάζεται **mystatic**
το κείμενο πάνω στην ετικέτα είναι **tampela**
οι συντεταγμένες της πάνω αριστερά γωνίας του κουμπιού μας είναι (25,25) (**Προσοχή!** η πάνω αριστερά γωνία του παράθυρου θεωρείται ότι έχει συντεταγμένες (0,0))
οι διαστάσεις του κουμπιού είναι: μήκος 50 και πλάτος 25 pixels
Αν θέλουμε το κείμενο πάνω στην ετικέτα να είναι μια φράση πρέπει να χρησιμοποιήσουμε αγκύλες αντί για διπλά εισαγωγικά για το κείμενο.
Την ετικέτα που δημιουργήσαμε μπορούμε να τη διαγράψουμε με την εντολή **staticdelete**, που συντάσσεται:

STATICDELETE p1

όπου **p1** το όνομα της ετικέτας.

Π.χ. **staticdelete "mystatic**

Αν θέλουμε να αλλάξουμε το κείμενο πάνω στην ετικέτα χρησιμοποιούμε την εντολή **staticupdate**, που συντάσσεται:

STATICUPDATE p1 p2

όπου p1 το όνομα της ετικέτας
p2 το καινούργιο κείμενο που θέλουμε να δείχνει η ετικέτα

Π.χ. **staticupdate "mystatic "nea_tampela**

η ετικέτα πλέον θα γράφει επάνω της το κείμενο nea_tampela

Δοκιμάστε τις παρακάτω εντολές στο περιβάλλον της LOGO

```
windowcreate "main "mywindow "mytitle 0 0 100 100 []
staticcreate "mywindow "mystatic [heading=0] 25 25 50 25
repeat 72 [right 5 staticupdate "mystatic sentence [heading=] heading wait 60]
buttoncreate "mywindow "close "΄Εξοδος 50 50 25 25 [windowdelete "mywindow
]
```

Παρατηρήστε τη χρήση των αγκυλών προκειμένου να εμφανιστεί φράση στο κείμενο της ετικέτας, καθώς και τη χρήση της εντολής sentence προκειμένου το κείμενο να σχηματισθεί τμηματικά. Η εντολή heading δείχνει τον προσανατολισμό της χελώνας σε μοίρες.

1.9.4 Μεταβολέας

Το αντικείμενο **μεταβολέας** είναι ένα αντικείμενο που μας παρέχεται από το προγραμματιστικό περιβάλλον της LOGO και μας δίνει την δυνατότητα να δίνουμε τιμές σε μεταβλητές αυτόματα. Έχει την μορφή:

Αντί να δίνουμε κάθε φορά τιμή σε μια μεταβλητή συνδέουμε την μεταβλητή με τον μεταβολέα και η μεταβλητή παίρνει αυτόματα τις τιμές που θέλουμε. Η εντολή για την δημιουργία του μεταβολέα είναι η **scrollbarcreate** και συντάσσεται ως εξής:

SCROLLBARCREATE p1 p2 p3 p4 p5 p6 p7

όπου p1 το παράθυρο μέσα στο οποίο θα δημιουργηθεί ο μεταβολέας
p2 το όνομα του μεταβολέα
p3 η συντεταγμένη του οριζόντιου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία του μεταβολέα
p4 η συντεταγμένη του κατακόρυφου άξονα όπου θα τοποθετηθεί η πάνω αριστερά γωνία του μεταβολέα
p5 το μήκος του μεταβολέα
p6 το πλάτος του μεταβολέα
p7 οι εντολές ή η διαδικασία που θα εκτελούνται με την αλλαγή του μεταβολέα

Π.χ. **scrollbarcreate "mywindow "myscroll 15 50 150 25 []**

το παράθυρο μέσα στο οποίο θα δημιουργηθεί ο μεταβολέας λέγεται **mywindow** (θα δούμε παρακάτω πως θα το δημιουργήσουμε αυτό το παράθυρο)
ο μεταβολέας ονομάζεται **myscroll**
οι συντεταγμένες της πάνω αριστερά γωνίας του μεταβολέα είναι (15, 50) (**Προσοχή!** η πάνω αριστερά γωνία του παράθυρου θεωρείται ότι έχει συντεταγμένες (0,0))
οι διαστάσεις του μεταβολέα είναι: μήκος 150 και πλάτος 25 pixels (επειδή το μήκος είναι μεγαλύτερο από το πλάτος ο μεταβολέας θα έχει οριζόντιο προσανατολισμό)
προς το παρόν δεν έχουμε καθορίσει τι θα συμβεί όταν αλλάξει ο μεταβολέας, για αυτό και ανάμεσα στις αγκύλες δεν έχουμε γράψει εντολές.
Για να δουλέψει σωστά ο μεταβολέας πρέπει να του δώσουμε αρχικές τιμές. Αυτό γίνεται με την εντολή **scrollbarset** η οποία συντάσσεται ως εξής:

SCROLLBARSET p1 p2 p3 p4

όπου p1 το όνομα του μεταβολέα που θα ρυθμιστεί
p2 η ελάχιστη τιμή του μεταβολέα
p3 η μέγιστη τιμή του μεταβολέα
p4 η αρχική τιμή του μεταβολέα

Π.χ. **scrollbarset "myscroll 0 100 50**

Η εντολή που παίρνει την τιμή από τον μεταβολέα είναι η **scrollbarget** και συντάσσεται ως εξής:

SCROLLBARGET p1

όπου p1 το όνομα του μεταβολέα
Χρησιμοποιώντας συνδυαστικά τις εντολές make και scrollbarget μπορούμε να δώσουμε σε μια μεταβλητή την τιμή που έχει ο μεταβολέας: **make "a scrollbarget ""myscroll**
Επειδή είναι πιθανό να χρησιμοποιούμε πολλές μεταβλητές, αν θέλουμε να εμφανίζεται μια ετικέτα κάτω από τον μεταβολέα που θα μας δείχνει σε ποια μεταβλητή πρόκειται να αποδοθεί η τιμή του μεταβολέα, χρησιμοποιούμε την εντολή staticcreate.

Δοκιμάστε τις παρακάτω εντολές στο περιβάλλον της LOGO

windowcreate "main "mywindow [Απόδοση τιμής σε μεταβλητή gr] 10 50 100 100 []
staticcreate "mywindow "still [Μεταβλητή] 40 40 40 30
scrollbarcreate "mywindow "myscroll 30 15 50 25 [staticupdate "still sentence [Μεταβλητή] scrollbarget "myscroll make "gr scrollbarget "myscroll]
scroolbarset "myscroll 0 300 50

1.9.5 Δομή Επιλογής

Με την δομή επιλογής μπορούμε να καθοδηγήσουμε το πρόγραμμα στην εκτέλεση μίας ή περισσοτέρων εντολών, με βάση το αν ισχύει ή όχι μια συγκεκριμένη συνθήκη. Η δομή επιλογής υποστηρίζεται στην Logo με τις εντολές **if** και **ifelse**. Η γενική τους μορφή είναι:

IF Συνθήκη [Εντολές]
IFELSE Συνθήκη [Εντολές1] [Εντολές2]

όπου Συνθήκη μια παράσταση, αποτελούμενη από αριθμούς, λέξεις, λίστες, αριθμητικές πράξεις και **ένα μόνο** συγκριτικό τελεστή (οι τελεστές που χρησιμοποιούνται στην Logo είναι οι =, Equal?, >, Greater?, <, less?, number?, word?, list?, empty?, Member?). Οι τελεστές-λέξεις σύγκρισης τελειώνουν όλες με ? και δέχονται πάντα δυο εισόδους στα δεξιά τους, ενώ οι τελεστές-σύμβολα δέχονται μία είσοδο από αριστερά και μία από δεξιά. Μία συνθήκη είτε είναι ΑΛΗΘΗΣ (δηλαδή θα ισχύει) ή θα είανι ΨΕΥΔΗΣ (δηλαδή δεν θα ισχύει), ανάλογα βέβαια με τις τιμές των μεταβλητών που συμμετέχουν στην συνθήκη. Στην εντολή **IF** οι εντολές εκτελούνται μόνο εφόσον η συνθήκη είναι **ΑΛΗΘΗΣ**. Στην εντολή **IFELSE** θα εκτελεστεί η πρώτη ομάδα εντολών εφόσον η συνθήκη είναι **ΑΛΗΘΗΣ**, ενώ θα εκτελεστεί η δεύτερη ομάδα εντολών εφόσον η συνθήκη είναι **ΨΕΥΔΗΣ**.

Παραδείγματα συνθηκών

```
:X = 0
Equal? "Γεώργιος "γεώργιος
equal? :X 0
:X > 0
greater? :x 0
:X < 0
less? :x 0
number? 14
word? "Γεώργιος
list? [Ανοίξτε τα βιβλία σας]
empty? "
```

Παραδείγματα εντολών

```
IF :X >9.5 [show "Προάγεται]

Ifelse  :X > 9.5 [show "Προάγεται] [show "Απορρίπτεται]
```

Παράδειγμα προγράμματος

Το παρακάτω πρόγραμμα υπολογίζει την διακρίνουσα μιας δευτεροβάθμιας εξίσωσης και ελέγχει το πλήθος των ριζών της.

```
make "a 3
make "b 2
make "c 5
make "d (power :b 2) -  (4*a*c)
if :d >0 [show [Η εξίσωση έχει δυο ρίζες]]
if :d =0 [show [Η εξίσωση έχει μία ρίζα]]
if :d >0 [show [Η εξίσωση δεν έχει ρίζες]]
```

Τροποποιήστε κατάλληλα το πρόγραμμα ώστε να υπολογίζει και να εμφανίζει την/τις ρίζα/ρίζες της εξίσωσης, όταν φυσικά υπάρχουν.