Μέρος

3

ΕΙΣΑΓΩΓΗ

Στο πρώτο μέρος αναφερθήκαμε στην έννοια του αλγορίθμου της Δομής δεδομένων και γενικά τα βήματα που πρέπει να ακολουθήσουμε για την αντιμετώπιση ενός προβλήματος με αλγορίθμους. Στο κεφάλαιο αυτό στόχος μας είναι να διατυπώσουμε τους αλγορίθμους σε μορφή κατανοητή από τον Η/Υ. Ο Προγραμματισμός των Η/Υ ασχολείται με ακριβώς αυτή τη διαδικασία .

Γλώσσες προγραμματισμού υπάρχουν πολλές. Η κάθε μια έχει σχεδιαστεί με ιδιαίτερη έμφαση σε κάποια χαρακτηριστικά. Έτσι υπάρχουν γλώσσες προγραμματισμού εξειδικευμένες στα γραφικά , άλλες για χειρισμό μεγάλων όγκων δεδομένων , άλλες στη σχεδίαση μέσω Η/Υ κλπ. Σχεδόν όλες όμως οι γλώσσες προγραμματισμού έχουν κοινά χαρακτηριστικά και επεξεργάζονται σχεδόν τους ίδιους τύπους δεδομένων.

Η γλώσσα προγραμματισμού που παρουσιάζεται στα επόμενα ονομάζεται «ΓΛΩΣΣΑ» . Περιέχει τα κοινά στοιχεία πολλών γλωσσών προγραμματισμού όπως η Visual Basic , Pascal ,C , C++ κλπ. Και είναι σχεδιασμένη για εκπαιδευτικούς σκοπούς.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΓΛΩΣΣΑΣ

Όπως όλες οι γλώσσες προγραμματισμού είναι τεχνιτές γλώσσες και όχι φυσικές που σημαίνει ότι αφού έχουν κατασκευαστεί για συγκεκριμένο σκοπό , η αρχιτεκτονική τους είναι τέτοια που να εξυπηρετεί αυτό το σκοπό. Ένα σύνολο λέξεων και σημείων στίξεως αποτελούν τις εντολές δηλαδή τις οδηγίες για την εκτέλεση διαφόρων ενεργειών από τον Η/Υ. Αυτό το σύνολο λέξεων και σημείων στίξεως μαζί με κάποιους κανόνες αποτελούν τη γλώσσα προγραμματισμού. Είναι γεγονός ότι το οποιοδήποτε σύστημα κανόνων είναι καλύτερο από την ανυπαρξία κανόνων , διότι οι κανόνες μας βοηθούν να δουλέψουμε στην κατεύθυνση του πειθαρχημένου προγραμματισμού.

Μια σημαντική μέθοδος πειθαρχημένου προγραμματισμού είναι ο ***ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ*** . Με τη μέθοδο αυτή το πρόβλημα χωρίζεται σε επί μέρους προβλήματα . Έτσι ένα πρόβλημα αντιμετωπίζεται σαν άθροισμα μικρότερων προβλημάτων. Η μέθοδος αυτή λέγεται : ***"Μέθοδος από επάνω προς τα κάτω"*** . Η «ΓΛΩΣΣΑ» που θα παρουσιάσομε στη συνέχεια στηρίζεται σε αυτή τη μέθοδο και διευκολύνει πολύ τον προγραμματισμό.

ΓΕΝΙΚΗ ΜΟΡΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Ένα πρόγραμμα σε «ΓΛΩΣΣΑ» έχει την παρακάτω μορφή:

ΠΡΟΓΡΑΜΜΑ < όνομα προγράμματος>
ΣΤΑΘΕΡΕΣ
Όνομα-1=τιμή-1
Όνομα-2=τιμή-2
Όνομα-3=τιμή-3
……
Όνομα-ν=τιμή-ν
ΜΕΤΑΒΛΗΤΕΣ
Τύπος-1:Λίστα-1
Τύπος-2: Λίστα-2
……..
Τύπος-ν: Λίστα-ν
ΑΡΧΗ
<εντολές>
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Περιοχή δηλώσεων Προγράμματος

Κύριο μέρος προγράμματος

Εδώ πρέπει να αναφέρουμε μερικές βασικές παρατηρήσεις πάνω στην παραπάνω δομή προγράμματος τις οποίες πρέπει να προσέξουμε ιδιαίτερα.

☼ Κάθε πρόγραμμα έχει μια επικεφαλίδα : **ΠΡΟΓΡΑΜΜΑ** < όνομα προγράμματος> . Η λέξη ΠΡΟΓΡΑΜΜΑ είναι δεσμευμένη και δεν μπορεί να χρησιμοποιηθεί σαν όνομα μεταβλητής ή σταθεράς.

Δεσμευμένες λέξεις είναι και οι επικεφαλίδες ΜΕΤΑΒΛΗΤΕΣ, ΣΤΑΘΕΡΕΣ. Η περιοχή από τον τίτλο μέχρι τη λέξη ΑΡΧΗ είναι η περιοχή δηλώσεων μεταβλητών και σταθερών

☼ Το κύριο μέρος του προγράμματος αρχίζει με τη λέξη **ΑΡΧΗ** και τελειώνει με το **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ** . Εδώ τοποθετείται ο αλγόριθμος του προγράμματος. Οι εντολές γράφονται με ελεύθερο τρόπο μπορούμε να χρησιμοποιούμε κεφαλαία ή μικρά γράμματα.

Η δική μας πρόταση όλες οι δεσμευμένες λέξεις με κεφαλαία γράμματα και τα ονόματα των μεταβλητών με λατινικούς χαρακτήρες για να γίνεται διάκριση από τις δεσμευμένες λέξεις.

☼ Σχόλια μπορούμε να τοποθετήσουμε σε οποιοδήποτε σημείο και γράφονται με όποιο τρόπο θέλουμε αρκεί το πρώτο γράμμα της γραμμής να είναι το θαυμαστικό (!)

☼ Σε κάθε πρόγραμμα δεν είναι απαραίτητο να υπάρχουν όλα τα τμήματα του προγράμματος. Δηλαδή αν δεν υπάρχει σταθερά μέσα στο πρόγραμμα το τμήμα δηλώσεων των σταθερών παραλείπεται.

☼ Δεν υπάρχει πρόγραμμα χωρίς κύριο μέρος (ΑΡΧΗ-ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ)

Στα επόμενα θα εξεταστούν όλα τα τμήματα της παραπάνω μορφής του προγράμματος όπως αυτά τα υποστηρίζει η γλώσσα προγραμματισμού «ΓΛΩΣΣΑ»

Το πρώτο μας πρόγραμμα

```
ΠΡΟΓΡΑΜΜΑ Πρώτο
ΑΡΧΗ
    ΓΡΑΨΕ «Αυτό είναι το πρώτο μου πρόγραμμα !!! Ζήτω !!!»
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Το πρόγραμμα αυτό θα τυπώσει στην οθόνη του Η/Υ τη φράση : Αυτό είναι το πρώτο μου πρόγραμμα !!! Ζήτω !!!

Για την εντολή **ΓΡΑΨΕ** θα αναφερθούμε παρακάτω. Όταν θα γράφουμε ένα πρόγραμμα όλες οι εντολές –δεσμευμένες λέξεις θα γράφονται με έντονα μπλε χρώμα

ΤΟ ΛΕΞΙΛΟΓΙΟ ΤΗΣ «ΓΛΩΣΣΑΣ»

Ένα πρόγραμμα σε ΓΛΩΣΣΑ αποτελείται από :
Γράμματα (Τα 24 του Ελληνικού και τα 26 του λατινικού αλφαβήτου)
Ψηφία (0 έως 9 τα αραβικά νούμερα)
Ειδικά σύμβολα
A. Πράξεων : +,-,*,/,div , mod
Β. Σχέσης : <,>,<=,>=,=,<>
Γ. Λογικών πράξεων : Η, ΚΑΙ , ΟΧΙ
Δ. Ειδικοί χαρακτήρες : , «,»,^, space

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ΠΟΥ ΥΠΟΣΤΗΡΙΖΕΙ Η «ΓΛΩΣΣΑ»

Για την σωστή αντιμετώπιση του θέματος θα αναφερθούμε συνοπτικά στην έννοια **«τύπος δεδομένων»** Τύπος δεδομένων είναι ο συνδυασμός από το σύνολο τιμών του δεδομένου και το σύνολο των πράξεων αυτών των τιμών

Για παράδειγμα λέμε ότι ένα δεδομένο είναι ΑΚΕΡΑΙΟΣ και εννοούμε ότι , μπορεί να πάρει ακέραιες τιμές και επί πλέον μπορούμε να κάνομε συγκεκριμένες πράξεις μεταξύ ακεραίων.

Η γλώσσα προγραμματισμού «ΓΛΩΣΣΑ» υποστηρίζει τέσσερεις τύπους δεδομένων

- **Τον Ακέραιο**
- **Τον Πραγματικό**
- **Τον Λογικό**
- **Τον Χαρακτήρα**

ΑΚΕΡΑΙΟΣ ΤΥΠΟΣ (Integer) –ΠΡΑΞΕΙΣ

Ο ακέραιος τύπος περιλαμβάνει όλους τους ακεραίους των μαθηματικών. Δηλαδή μπορεί να είναι θετικός ή αρνητικός.

Οι πράξεις που ορίζονται στον ακέραιο τύπο είναι :

ΤΕΛΕΣΤ	ΠΡΑΞΗ
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
Div	Ακέραιο Πηλίκο
Mod	Υπόλοιπο ακέραιας διαίρεσης

Η σύνταξη των Div και Mod είναι η εξής :

A DIV B και A Mod B όπου A και B ακέραιοι αριθμοί.

Μερικά παραδείγματα στις πράξεις με τους ακεραίους:

-10 + 3 → -7 , -3*5 → -15 , 25 Div 7 → 3 ,-25 Div 7 → -3 , 25 Div -7 →-3 , 25 Mod 7 → 4 , -25 Mod 7 → -4 , 25 Mod -7 → 4 , -25 Mod -7 → -4

ΠΡΑΓΜΑΤΙΚΟΣ ΤΥΠΟΣ (Real) –ΠΡΑΞΕΙΣ

Στη «ΓΛΩΣΣΑ» οι πραγματικοί αριθμοί αντιστοιχούν στους γνωστούς δεκαδικούς αριθμούς Οι πράξεις που ορίζονται στον πραγματικό τύπο είναι :

ΤΕΛΕΣΤ	ΠΡΑΞΗ
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση

Πρέπει να επισημάνουμε ότι :

☼ Η διαίρεση (/) δίνει σαν αποτέλεσμα **πάντα** πραγματικό αριθμό

☼ Σε μια πράξη που συνυπάρχουν Ακέραιος τύπος και πραγματικός τύπος το αποτέλεσμα θα είναι πραγματικός. Για παράδειγμα 2*3,5 = 7,0

ΛΟΓΙΚΟΣ ΤΥΠΟΣ (Boolean) –ΠΡΑΞΕΙΣ

Ο λογικός τύπος παίρνει μόνο δύο τιμές «ΑΛΗΘΗΣ» , «ΨΕΥΔΗΣ»

Μια πρόταση που μπορεί να χαρακτηριστεί «Αληθής» ή «Ψευδής» θα ονομάζεται **λογική πρόταση**. Για παράδειγμα προτάσεις όπως :

Προτάσεις	Χαρακτηρισ
120>7	Α
30<120	Ψ
«Σήμερα χιονίζει»	Α ή Ψ
Α=20	Α ή Ψ
Χ>=80	Α ή Ψ

Είναι λογικές προτάσεις γιατί μπορούν να χαρακτηριστούν «Αληθείς» ή «Ψευδείς»

Παρατηρούμε ότι για να σχηματίσομε μια λογική πρόταση απαιτείται ένας **τελεστής συσχέτισης**
Οι τελεστές συσχέτισης στη «ΓΛΩΣΣΑ» φαίνονται στον πίνακα :

Αλγεβρικός	**Τελεστής σε**
=	=
≠	< >
<	<
≤	<=
>	>
≥	>=

Οι πράξεις που υποστηρίζει η γλώσσα προγραμματισμού στον λογικό τύπο η «ΓΛΩΣΣΑ» είναι τρεις :

Λογική Πράξη	**τελεστή**	**Εναλλακτικός**
Διάζευξη	Η	OR
Σύζευξη	ΚΑΙ	AND
Άρνηση	ΟΧΙ	NOT

.

Λογική πράξη «Άρνηση» - ΟΧΙ

Πρότασ	**ΟΧ**
Α	Ψ
Ψ	Α

Παραδείγματα : ΟΧΙ (7>8) είναι Αληθής
ΟΧΙ (Α=Β) αν Α=5 και Β=12 είναι Αληθής
ΟΧΙ (Α+Β>10) Αν Α=5 και Β=7 είναι Ψευδής

Λογική πράξη «Σύζευξη»- ΚΑΙ

Πρότασ	**Πρότασ**	**Q**
Α	Α	Α
Α	Ψ	Ψ
Ψ	Α	Ψ
Ψ	Ψ	Ψ

Όπως φαίνεται από τον παραπάνω πίνακα δύο λογικές προτάσεις Q , P που συνδέονται με το λογικό ΚΑΙ δημιουργούν με νέα λογική πρόταση (Q ΚΑΙ P) η οποία είναι αληθής μόνο στην περίπτωση που και οι δύο προτάσεις που τη συνθέτουν είναι Αληθείς. Σε όλες τις άλλες περιπτώσεις η σύνθετη πρόταση είναι ψευδής
Παραδείγματα : (7<8) ΚΑΙ (6<12) Αληθής
(Α=15) ΚΑΙ (Β<20) Αν Α=6 , Β=10 είναι Ψευδής

Λογική πράξη «Διάζευξη»-Ή

Πρότασ	**Πρότασ**	**Q Η**
Α	Α	Α
Α	Ψ	Α
Ψ	Α	Α
Ψ	Ψ	Ψ

Όπως φαίνεται από τον παραπάνω πίνακα δύο λογικές προτάσεις Q , P που συνδέονται με το λογικό Η δημιουργούν με νέα λογική πρόταση (Q Η P) η οποία είναι Ψευδής μόνο στην περίπτωση που και οι δύο προτάσεις που τη συνθέτουν είναι Ψευδείς. Σε όλες τις άλλες περιπτώσεις η σύνθετη πρόταση είναι Αληθής

Παραδείγματα: (7>8) Η (6<12) Αληθής
(Α=15) Η (Β<20) Αν Α=6 , Β=10 είναι Αληθής

☼ Πρέπει να γνωρίζουμε ότι «ΑΛΗΘΗΣ» > «ΨΕΥΔΗΣ»

ΤΥΠΟΣ ΧΑΡΑΚΤΗΡΑΣ (STRING)- ΠΡΑΞΕΙΣ

Ο τύπος αυτός παίρνει τιμές ένα ή περισσότερους χαρακτήρες. Οι χαρακτήρες πρέπει υποχρεωτικά να βρίσκονται μέσα σε απλά εισαγωγικά. Για παράδειγμα 'Α', 'ΜΑΝΟΛΗΣ ' , ' Σήμερα είναι Πέμπτη' . Μέσα στα απλά εισαγωγικά μπορούμε να βάλουμε **οποιοδήποτε χαρακτήρα που παράγεται από το πληκτρολόγιο**. Επειδή τα δεδομένα αυτού του τύπου περιέχουν και αριθμητικούς και αλφαβητικούς χαρακτήρες ονομάζονται **αλφαριθμητικά**

Η **μοναδική πράξη** που μπορεί να γίνει σε αυτό τον τύπο είναι η πρόσθεση των αλφαριθμητικών , με την έννοια της **συνένωσης.**

Παράδειγμα 1 : 'ΧΤΕΣΙΝΟ' +'ΒΡΑΔΥΝΟ' θα έχει σαν αποτέλεσμα
'ΧΤΕΣΙΝΟΒΡΑΔΥΝΟ'

Παράδειγμα 2 : '12'+'14' θα έχει σαν αποτέλεσμα '1214'

Παράδειγμα 3 : Str1←'Καλή' , Str2 ← ' Μέρα'
Str3← Str1+Str2
Το Str3 θα έχει την τιμή ' Καλή Μέρα'
Παρατηρούμε ότι στο παράδειγμα 3 έχουμε εκχωρίσει στις μεταβλητές Str1 , Str2, Str3 αλφαριθμητικές σταθερές.

Σε πολλές εφαρμογές η διάταξη των χαρακτήρων παίζει σημαντικό ρόλο , ιδιαίτερα η αλφαβητική διάταξη διευκολύνει πολλές εργασίες . Η **σύγκριση** (Διάταξη) δύο αλφαριθμητικών σταθερών ή μεταβλητών μπορεί να γίνει χρησιμοποιώντας τους γνωστούς τελεστές συσχέτισης (= , <= , >= , < . >) αφού λάβουμε υπ' όψιν μας τα παρακάτω :

Για τα γράμματα ισχύουν 'Α' <'Β'<…….<'Ζ' για το αγγλικό αλφάβητο . Ανάλογα ισχύουν και για το Ελληνικό αλφάβητο.
Οι χαρακτήρες '0' έως '9' δεν είναι οι ίδιοι με τους αριθμούς 0 έως 9 ωστόσο ισχύει η παρακάτω διάταξη '0'<'1'<'2'<……<'9'

Έτσι η λογική έκφραση 'Α' <'Κ' είναι αληθής.

Για να γίνει η σύγκριση σε δύο αλβαβητικές σταθερές συγκρίνουμε το πρώτο γράμμα της κάθε μιας και αν προκύψει διάταξη αυτή προσδιορίζει και τη διάταξη μεταξύ των σταθερών διαφορετικά προχωρούμε στη σύγκριση των δεύτερων χαρακτήρων κ.οκ

Για παράδειγμα 'ΑΝΝΑ' <'ΑΝΝΙΤΑ' διότι ισχύει 'Α'<'Ι'

ΔΗΛΩΣΗ ΣΤΑΘΕΡΩΝ – ΜΕΤΑΒΛΗΤΩΝ

Στα προηγούμενα περιγράψαμε βασικούς τύπους της «ΓΛΩΣΣΑΣ» .Παρακάτω θα δούμε πως δηλώνοντα σε ένα πρόγραμμα στην περιοχή δηλώσεων του προγράμματος.

Α. ΔΗΛΩΣΕΙΣ ΣΤΑΘΕΡΩΝ

Στη περιοχή δηλώσεων κάτω από τη λέξη ΣΤΑΘΕΡΕΣ γράφουμε το όνομα της <σταθερά> = <τιμή σταθεράς>

Για παράδειγμα :
Π=3.14
ΑΠΟΣΤΑΣΗ = 250
ΛΕΞΗ =’ΚΑΛΗΝΥΧΤΑ’
ΣΗΜΑΙΑ =ΨΕΥΔΗΣ

Στην περίπτωση που δεν απαιτούνται στο πρόγραμμα σταθερές η παράγραφος αυτή παραλείπεται. Στους δεκαδικούς αριθμούς σαν υποδιαστολή χρησιμοποιείται η τελεία «.»

Ένας γενικός κανόνας είναι όταν μια τιμή επαναλαμβάνεται μέσα στο πρόγραμμα να δηλώνεται σαν σταθερά.

Για παράδειγμα ένα πρόβλημα που μας τυπώνει τιμολόγια πιθανόν να χρειαστεί ο ΦΠΑ(18%) να χρησιμοποιηθεί πολλές φορές. Αν κάποια στιγμή αλλάξει ο ΦΠΑ είναι προτιμότερο να έχει δηλωθεί σαν σταθερά (ΦΠΑ=0.18) διότι έτσι θα διορθωθεί μία μόνο φορά.

Β. ΔΗΛΩΣΕΙΣ ΜΕΤΑΒΛΗΤΩΝ

Στη περιοχή δηλώσεων κάτω από τη λέξη ΜΕΤΑΒΛΗΤΕΣ γράφουμε τον τύπο των μεταβλητών (ΑΚΕΡΑΙΕΣ, ΠΡΑΓΜΑΤΙΚΕΣ, κλπ) ακολουθούμενο από τα ονόματα των μεταβλητών χωρισμένες με κόμμα “,”

Για παράδειγμα :

ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ : I,SUM
ΠΡΑΓΜΑΤΙΚΕΣ : X, MESOS_OROS
ΛΟΓΙΚΕΣ : ΣΗΜΑΙΑ, flag
ΧΑΡΑΚΤΗΡΕΣ : A ,ONOMA ,LastName

Στην περίπτωση που δεν απαιτούνται στο πρόγραμμα μεταβλητές η παράγραφος αυτή παραλείπεται

☼ Οι μεταβλητές και οι σταθερές απαγορεύεται να είναι δεσμευμένες λέξεις (π.χ ΠΡΟΓΡΑΜΜΑ, ΣΤΑΘΕΡΕΣ, ΜΕΤΑΒΛΗΤΕΣ , ΑΚΕΡΑΙΕΣ, ΠΡΑΓΜΑΤΙΚΕΣ, ΑΡΧΗ, ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ, και όλα τα ονόματα των εντολών που θα παρουσιαστούν στη συνέχεια)

☼ Δεν επιτρέπεται να υπάρχουν κενά στα ονόματα των μεταβλητών

☼ Δεν επιτρέπεται να χρησιμοποιείς το ίδιο όνομα μεταβλητής για διαφορετικούς σκοπούς.

☼ Να χρησιμοποιείτε ονόματα μεταβλητών λέξεις που έχουν σχέση με αυτό που εκφράζει η μεταβλητή. Για παράδειγμα αν θέλουμε να χρησιμοποιήσουμε μια μεταβλητή που θα δέχεται για τιμές επώνυμα καλό είναι να δώσουμε σ'αυτή το όνομα Eponymo , ή LName ή LastName ή Last_Name ή ΕΠΩΝΥΜΟ

Μετά από την παρουσίαση όλων των τύπων δεδομένων καθώς και των πράξεων που υποστηρίζει κάθε τύπος στη «ΓΛΩΣΣΑ» θα δούμε πως δημιουργούμε σύνθετες αριθμητικές παραστάσεις και σύνθετες λογικές εκφράσεις.

Δημιουργία αριθμητικών εκφράσεων (παραστάσεων)

Οι αριθμητικές παραστάσεις περιέχουν σταθερές , μεταβλητές και αριθμητικούς τελεστές, όπως και στην άλγεβρα.

Στην περίπτωση που οι σταθερές και οι μεταβλητές είναι τύπου ΑΚΕΡΑΙΟΣ σε μια αριθμητική παράσταση και **δεν υπάρχει ο τελεστής της διαίρεσης** , τότε η τιμή της παράστασης είναι ακέραιος. αριθμός , **σε όλες τις άλλες περιπτώσεις η τιμή της είναι ένας ΠΡΑΓΜΑΤΙΚΟΣ αριθμός**.

Μπορούμε να κατασκευάσουμε πολύπλοκες αλγεβρικές παραστάσεις στη «ΓΛΩΣΣΑ» χρησιμοποιώντας παρενθέσεις. Επίσης η σειρά της εκτέλεσης των πράξεων διέπεται από ορισμένους κανόνες που εμφανίζονται στο πίνακα.:

Σύμβολ	Σ
( )	1
^	2
* , /,Div,	3
+ , -	4

Είναι φανερό ότι με τη χρήση της παρένθεσης αλλάζει η σειρά εκτέλεσης των πράξεων και κατά συνέπεια η τιμή του αποτελέσματος.

Παράδειγμα :

Αλγεβρική παράσταση	**Παράσταση σε «ΓΛΩΣΣΑ»**
$b^2 - 4ac$	b^2-4*a*c
$\frac{a+b}{x-y}$	(a+b)/(x-y)
$\frac{1}{1+x^2}$	1/(1+X^2)

Οι τελεστές DIV, MOD μπορεί να χρησιμοποιηθούν μόνο για ακεραίου τύπου μεταβλητές , σταθερές, ή παραστάσεις.

Παράδειγμα α) 2 mod 7=2 , 2 div 7 = 0 , 12 mod 3 =0 , 12 div 3 = 4 , 12 div 9 = 1 ,12 mod 9 =3

Παράδειγμα β) $N \bmod 2 = \begin{cases} 1, αν Ν περιττός \\ 0, αν Ν άρτιος \end{cases}$

Παράδειγμα γ) Η αριθμητική παράσταση Z-(A+B div 2)+W*Y υπολογίζεται ως εξής (Η υπογράμμιση σημαίνει ήδη εκτελεσμένες πράξεις)

Α) Η διαίρεση B Div 2 Β) Η Πρόσθεση στην A+B Div 2

Γ) Ο Πολλαπλασιασμός στην W*Y Δ) Η αφαίρεση στην Z- (A+B Div 2)

Ε) Η Πρόσθεση στην Z- (A+B Div 2) +W*Y

Δημιουργία σύνθετων λογικών εκφράσεων (παραστάσεων)

Όπως στις αριθμητικές πράξεις χρησιμοποιούμε τους αριθμητικούς τελεστές '+' , '-' , '*' ,'/',div , mod και κατασκευάζουμε συνθετότερες αριθμητικές ή αλγεβρικές παραστάσεις , έτσι και με τις λογικές πράξεις μπορούμε να σχηματίσουμε χρησιμοποιήσουμε τους τελεστές συσχέτισης , λογικούς τελεστές Η , ΚΑΙ, ΟΧΙ ,σταθερές , ή μεταβλητές για να κατασκευάσουμε συνθετότερες λογικές προτάσεις..

Ανάλογα με τις τιμές των σταθερών και μεταβλητών σε μια λογική παράσταση , η τιμή της παράστασης μπορεί να χαρακτηριστεί Αληθής ή Ψευδής.

Επειδή μέσα στις λογικές παραστάσεις μπορεί να υπάρχουν και αριθμητικές παραστάσεις γιαυτό θα πρέπει να υπάρχει μια σειρά εκτέλεσης των πράξεων. Έτσι θα πρέπει **πρώτα να εκτελούνται οι αριθμητικές πράξεις και μετά οι λογικές.** Έτσι η σειρά των τελεστών φαίνεται στον παρακάτω πίνακα :

Τελεστές-	Σ
( )	1
ΟΧΙ	2
* , /,Div, Mod,	3
+ , - , Η	4
=,<	5

Παράδειγμα 1) Να εκφραστούν οι παρακάτω σχέσεις σε «ΓΛΩΣΣΑ»

α.) 0<X<3.5 β. X≤ Υ≤ Z

Απάντηση α.) (0<X) ΚΑΙ (X<3.5)
β) (X<=Y) ΚΑΙ (Υ<=Z)

Παράδειγμα 2) Αν X=3 , Υ=4 , Z=2 και Flag = Ψευδής τότε οι παραστάσεις

(X>Z) ΚΑΙ (Υ>Z) (Αληθής)
(X+Υ/Z) <= 3.5 (Ψευδής)
(Z>X) Η (Z>Υ) (Ψευδής)
ΟΧΙ Flag (Αληθής)
(ΟΧΙ Flag) Η ((Υ+Z) >= (X-Υ)) (Αληθής)

ΕΙΣΟΔΟΣ- ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ

Όπως σε όλες τις γλώσσες προγραμματισμού έτσι και στη «ΓΛΩΣΣΑ» για να αποδώσουμε σε μια μεταβλητή τιμή χρησιμοποιούμε την εντολή εκχώρησης τιμής «←»

Γενική Μορφή : <Μεταβλητή> ← <Παράσταση>

Παράδειγμα 1 Met1 ← 5 , Met2 ← A*(B-C^2) , FLAG1 ←Αληθής , Flag2 ← (X>=0)

☼ **Προσοχή !!! Ο τύπος της παράστασης στο δεύτερο μέλος και εκείνος της μεταβλητής στο πρώτο πρέπει να είναι ο ίδιος.** . Δηλαδή αν η παράσταση έχει αριθμητική τιμή ΑΚΕΡΑΙΟ θα πρέπει και η μεταβλητή στο πρώτο μέλος να δηλωθεί Ακέραια , Αν η παράσταση έχει τιμή ΛΟΓΙΚΗ θα πρέπει και η μεταβλητή να δηλωθεί λογική.

☼ Σε μια μεταβλητή που έχει δηλωθεί Πραγματική, μπορεί να εκχωρηθεί ακέραια τιμή. Το αντίθετο απαγορεύεται

Παράδειγμα 2

```
ΠΡΟΓΡΑΜΜΑ Test_Εμβαδό
ΣΤΑΘΕΡΕΣ
        Π=3.14
ΜΕΤΑΒΛΗΤΕΣ
      ΑΚΕΡΑΙΕΣ : ΑΚΤΙΝΑ
      ΠΡΑΓΜΑΤΙΚΕΣ : ΕΜΒΑΔΟ
ΑΡΧΗ
      ΑΚΤΙΝΑ ← 5
      ΕΜΒΑΔΟ ← Π*ΑΚΤΙΝΑ^2
      ΓΡΑΨΕ ΕΜΒΑΔΟ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Προφανώς το παραπάνω πρόγραμμα βρίσκει το εμβαδό του κύκλου. Αυτό που πρέπει να παρατηρήσομε ε'ναι ότι ενώ η ακτίνα έχει δηλωθεί ακέραιος επειδή το Π είναι πραγματικός η μεταβλητή ΕΜΒΑΔΟ πρέπει να δηλωθεί πραγματικός.

ΕΙΣΟΔΟΣ ΔΕΔΟΜΕΝΩΝ (ΔΙΑΒΑΣΜΑ)

Η είσοδος των δεδομένων από το πληκτρολόγιο ενώ εκτελείται το πρόγραμμα γίνεται με την εντολή ΔΙΑΒΑΣΕ . Τα δεδομένα καταχωρούνται σε μεταβλητές και ακολουθεί επεξεργασία τους . Η γενική μορφή της εντολής είναι:

Γενική μορφή: ΔΙΑΒΑΣΕ μεταβλητή1,μεταβλητή2,…

☼ **Προσοχή !!! Θα πρέπει τα δεδομένα που θα δώσουμε από το πληκτρολόγιο να είναι του ίδιου τύπου με τη δήλωση της μεταβλητής.** Διαφορετικά θα προκύψει πρόβλημα στην είσοδο δεδομένων

Παράδειγμα 1: ΔΙΑΒΑΣΕ ΜΗΚΟΣ , ΔΙΑΒΑΣΕ ΟΝΟΜΑ

Παράδειγμα 2

ΠΡΟΓΡΑΜΜΑ Υπολογισμός_ΦΠΑ
ΣΤΑΘΕΡΕΣ
Σ_ΦΠΑ=0,18
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ : ΤΙΜΗ,ΦΠΑ
ΧΑΡΑΚΤΗΡΕΣ : Name
ΑΡΧΗ
ΔΙΑΒΑΣΕ Name
ΔΙΑΒΑΣΕ ΤΙΜΗ
ΦΠΑ ← ΤΙΜΗ*Σ_ΦΠΑ
ΓΡΑΨΕ Name
ΓΡΑΨΕ ΦΠΑ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Μια καλή τακτική κατά την είσοδο των δεδομένων στο πρόγραμμα , με την εντολή ΔΙΑΒΑΣΕ , είναι κάθε σε μεταβλητή να χρησιμοποιούμε μια εντολή ΔΙΑΒΑΣΕ

ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

Η εντολή εισόδου μας επιτρέπει να εισάγουμε δεδομένα στο πρόγραμμα , τα οποία αφού επεξεργαστούν , με την βοήθεια της εντολής εξόδου λαμβάνουμε τα αποτελέσματα. Η εντολή η οποία ανακατευθύνει τα αποτελέσματα σε συσκευή εξόδου του Η/Υ είναι η ΓΡΑΨΕ . Η συσκευή εξόδου μπορεί να είναι : Η Οθόνη , ο εκτυπωτής , η η βοηθητική Μνήμη.

Γενική μορφή : ΓΡΑΨΕ X1,X2,X3,…

Στη παραπάνω γενική μορφή την εντολή ΓΡΑΨΕ ακολουθούν χ1,χ2,…. Χωρισμένα με κόμμα , που μπορεί να είναι σταθερές ή μεταβλητές.. Το αποτέλεσμα της εντολής είναι να εμφανίσει τις τιμές των σταθερών και μεταβλητών στην οθόνη του Η/Υ .

Παράδειγμα : Με τη χρήση της εντολής ΓΡΑΨΕ μπορούμε να έχουμε βελτιωμένη είσοδο δεδομένων και επίσης βελτιωμένη έξοδο αποτελεσμάτων. Έτσι το πρόγραμμα που παρουσιάστηκε προηγούμενα με τον ΦΠΑ θα μπορούσε να γίνει :

ΠΡΟΓΡΑΜΜΑ Υπολογισμός_ΦΠΑ
ΣΤΑΘΕΡΕΣ
Σ_ΦΠΑ=0,18
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ : ΤΙΜΗ, ΦΠΑ,SUM

```
        ΧΑΡΑΚΤΗΡΕΣ : Name
ΑΡΧΗ
        ΓΡΑΨΕ  ‘ Δώσε όνομα προϊόντος’
        ΔΙΑΒΑΣΕ Name
        ΓΡΑΨΕ ‘ Δώσε τιμή του προϊόντος’
        ΔΙΑΒΑΣΕ  ΤΙΜΗ
        ΦΠΑ ← ΤΙΜΗ*Σ_ΦΠΑ
        SUM ← ΤΙΜΗ+ΦΠΑ
        ΓΡΑΨΕ ‘ ---------- ΑΠΟΤΕΛΕΣΜΑΤΑ -------
        ΓΡΑΨΕ  ‘ Ονομα :   ‘,Name
        ΓΡΑΨΕ  ‘Τιμή     :   ‘,ΤΙΜΗ
        ΓΡΑΨΕ ‘ΦΠΑ     :   ‘, ΦΠΑ
        ΓΡΑΨΕ  ‘Σύνολο  :   ‘,SUM
        ΓΡΑΨΕ ‘----------------------------------------------
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Το πρόγραμμα αυτό στην είσοδο των δεδομένων μας τυπώνει ένα πληροφοριακό μήνυμα πριν ο χρήστης εισάγει δεδομένα. Έτσι η επικοινωνία χρήστη με πρόγραμμα γίνεται πιο φιλική. Επίσης τα αποτελέσματα θα έχουν την παρακάτω μορφή αν δώσουμε στην είσοδο Name=Βιβλίο Πληροφορικής , ΤΙΜΗ=10000 . Παρατηρούμε ότι η εντολή ΓΡΑΨΕ τυπώνει την τιμή μιας αλφαριθμητικής σταθεράς και την τιμή μιας μεταβλητής.

```
---------- ΑΠΟΤΕΛΕΣΜΑΤΑ ----------
 ‘Ονομα :   Βιβλίο Πληροφορικής
 Τιμή     :   10000
ΦΠΑ     :    1800
Σύνολο   :  11800
 ----------------------------------------------
```

Δεν είναι καλύτερα !!!!!

ΔΟΜΕΣ ΑΛΓΟΡΙΘΜΩΝ

Επειδή οι δομές αλγορίθμων έχουν παρουσιαστεί στο πρώτο μέρος , εδώ γίνεται μια συνοπτική ανακεφαλαίωση όλων των δομών. ΑΚΟΛΟΥΘΙΑ – ΕΠΙΛΟΓΗ - ΕΠΑΝΑΛΗΨΗ . Ισχύουν όλα όπως έχουν παρουσιαστεί στο μέρος 1

1η Μορφή Δομής Επιλογής

```
ΑΝ <συνθήκη> ΤΟΤΕ
        ομάδα εντολών
ΤΕΛΟΣ_ΑΝ
```

2η Μορφή Δομής Επιλογής

```
ΑΝ <συνθήκη> ΤΟΤΕ
            Ομάδα εντολών 1
          ΑΛΛΙΩΣ
            Ομάδα εντολών 2
ΤΕΛΟΣ_ΑΝ
```

3η Μορφή Δομής επιλογής

```
ΕΠΙΛΕΞΕ έκφραση
ΠΕΡΙΠΤΩΣΗ 1
        Ομάδα εντολών 1
ΠΕΡΙΠΤΩΣΗ 2
        Ομάδα εντολών 2
            .....
 ΠΕΡΙΠΤΩΣΗ N
        Ομάδα εντολών ν
         ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
                Ομάδα εντολών
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
```

4η Μορφή – παραλλαγή της 3ης μορφής επιλογής

```
ΑΝ <συνθήκη 1> ΤΟΤΕ
      Ομάδα εντολών 1
      ΑΛΛΙΩΣ_ΑΝ <συνθήκη 2> ΤΟΤΕ
                  Ομάδα εντολών 2
                  ΑΛΛΙΩΣ_ΑΝ <συνθήκη 3 > ΤΟΤΕ
                        Ομάδα εντολών 3
                  ……..
          ΑΛΛΙΩΣ
                  Ομάδα εντολών ν
ΤΕΛΟΣ_ΑΝ
```

ΕΜΦΩΛΕΥΜΕΝΗ ΕΠΙΛΟΓΗ

```
ΑΝ <συνθήκη1> ΤΟΤΕ
        Ομάδα εντολών 1
        ΑΝ <συνθήκη2> ΤΟΤΕ
                  Ομάδα εντολών 2
         ΤΕΛΟΣ_ΑΝ
  ΑΛΛΙΩΣ
        Ομάδα εντολών 2
         ΑΝ <συνθήκη> ΤΟΤΕ
                  Ομάδα εντολών 3
              ΑΛΛΙΩΣ
                        ομάδα εντολών 4
          ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
```

1η Μορφή Επαναληπτικής δομής (ΓΙΑ - ΑΠΟ - ΜΕΧΡΙ)

```
ΓΙΑ <μτ> ΑΠΌ <ατ> ΜΈΧΡΙ <ττ> ΜΕ ΒΉΜΑ <μβ>

<ομάδα εντολών>

ΤΈΛΟΣ_ΕΠΑΝΆΛΗΨΗΣ
```

2η Μορφή Επαναληπτικής Δομής (ΟΣΟ - ΕΠΑΝΑΛΑΒΕ)

ΌΣΟ <συνθήκη> **ΕΠΑΝΆΛΑΒΕ**

<ομάδα εντολών>

ΤΈΛΟΣ_ΕΠΑΝΆΛΗΨΗΣ

3η Μορφή Επαναληπτικής δομής (ΜΕΧΡΙΣ_ΟΤΟΥ)

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
<ομάδα εντολών>
ΜΕΧΡΙΣ_ΟΤΟΥ <συνθήκη τέλους>

Για τις μετατροπές από τη μια μορφή σε άλλη , στις δομές επανάληψης ισχύουν αυτά που έχουν αναφερθεί στους αλγορίθμους

ΑΣΚΗΣΗ Να γραφεί αλγόριθμος ο οποίος να υπολογίζει τις αποδοχές ενός υπαλλήλου ως εξής: Για μικτό μισθό μέχρι 200.000 δρχ. το ποσοστό των κρατήσεων είναι 12,5% και για μεγαλύτερο μισθό 15%. Ο αλγόριθμος θα διαβάζει το πλήθος των ωρών που εργάστηκε ο υπάλληλος και το ύψος του ωρομισθίου και θα εμφανίζει τα εξής:

ΣΥΝΟΛΟ ΩΡΩΝ : ..... ΩΡΟΜΙΣΘΙΟ : .....

ΣΥΝΟΛΙΚΕΣ ΑΠΟΔΟΧΕΣ : .....

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ

Άσκηση 1 Να δειχθεί ότι οι προτάσεις α.) ΟΧΙ (Ρ ΚΑΙ Q) β) (ΟΧΙ Ρ) Η (ΟΧΙ Q) δίνουν τα ίδια αποτελέσματα. (Νόμος του Morgan)

Λύση της άσκησης 1
Είναι χρήσιμο σε τέτοιες περιπτώσεις που δεν γνωρίζουμε την τιμή των προτάσεων Ρ και Q να κατασκευάζουμε τον ***πίνακα αλήθειας*** των παραπάνω προτάσεων σε όλες τις δυνατές περιπτώσεις.

Ρ	Q	ΟΧΙ Ρ	ΟΧΙ Q	Ρ ΚΑΙ Q	ΟΧΙ(Ρ ΚΑΙQ)	(ΟΧΙ Ρ) Η (ΟΧΙ Q)
Α	Α	Ψ	Ψ	Α	Ψ	Ψ
Α	Ψ	Ψ	Α	Ψ	Α	Α
Ψ	Α	Α	Ψ	Ψ	Α	Α
Ψ	Ψ	Α	Α	Ψ	Α	Α

Οι δύο πρώτες στήλες είναι όλες οι δυνατές περιπτώσεις για τις προτάσεις Ρ και Q . Οι δύο

τελευταίες στήλες είναι τα αποτελέσματα για τις ζητούμενες προτάσεις. Παρατηρούμε ότι είναι τα ίδια άρα οι προτάσεις δίνουν τα ίδια αποτελέσματα.

Άσκηση 2 **Δίνονται οι προτάσεις P : Ο αριθμός χ είναι ρητός**
Q : Ο αριθμός χ είναι φυσικός
α) Να διατυπωθούν με λογικές εκφράσεις οι παρακάτω προτάσεις
1. « Ο χ δεν είναι ρητός» 2. «Ο χ δεν είναι φυσικός» 3. «Ο χ είναι ρητός και όχι φυσικός»
β) Να διατυπωθούν με λόγια οι παρακάτω λογικές εκφράσεις
1. P H Q 2. P ΚΑΙ Q 3. (ΟΧΙ P) ΚΑΙ (ΟΧΙ Q) 4. P ΚΑΙ (ΟΧΙ Q)

Άσκηση 3 **Να γραφεί η συνθήκη αν δίδονται οι παρακάτω προτάσεις :**
Α : αδικαιολόγητες απουσίες
Δ :δικαιολογημένες απουσίες
S : Σύνολο απουσιών (Α+Δ)
Ο μαθητής έχει δικαίωμα να δώσει εξετάσεις τον Ιούνιο αν:
Το σύνολο των απουσιών είναι μικρότερο ή ίσο με 64 ή όταν το σύνολο των απουσιών του είναι μικρότερο ή ίσο με 114 με την προϋπόθεση ότι οι πάνω (επιπλέον) από τις 64 απουσίες είναι δικαιολογημένες

Απάντηση άσκησης 3 : Shmaia← (S<=64) H ((S<=114) ΚΑΙ (Δ>=(S-64)))

Άσκηση 4 **Να γραφεί η συνθήκη αν δίδονται οι παρακάτω προτάσεις :**
ΑΑ : αδικαιολόγητες απουσίες
ΔΑ :δικαιολογημένες απουσίες
S : Σύνολο απουσιών (ΑΑ+ΔΑ)
ΜΟ : Μέσος όρος προφορικής βαθμολογίας
Ο μαθητής έχει δικαίωμα να δώσει εξετάσεις τον Ιούνιο αν:
α. έχει μέχρι 64 απουσίες ή
β. έχει μέχρι 114 απουσίες από τις οποίες οι αδικαιολόγητες δεν ξεπερνούν τις 64 ή
γ. έχει μέχρι 164 απουσίες από τις οποίες οι αδικαιολόγητες δεν ξεπερνούν τις 64 και ο μέσος όρος στα προφορικά του είναι πάνω από 15.

Απάντηση άσκησης 4
flag1← (S<=64) H ((S <=114) ΚΑΙ (ΑΑ<=64)) H ((S<=164) ΚΑΙ (ΑΔ <=64) ΚΑΙ (ΜΟ>=15))

Άσκηση 5 **Να γραφεί η συνθήκη αν δίδονται οι παρακάτω προτάσεις :**
ΑΑ : αδικαιολόγητες απουσίες
ΔΑ :δικαιολογημένες απουσίες
S : Σύνολο απουσιών (ΑΑ+ΔΑ)
ΜΟ : Μέσος όρος προφορικής βαθμολογίας
Ο μαθητής παραπέμπεται για ολική εξέταση το Σεπτέμβριο αν:
α. έχει πάνω από 64 και μέχρι 114 απουσίες και οι αδικαιολόγητες ξεπερνούν τις 64 ή

β. έχει πάνω από 114 και μέχρι 164 απουσίες από τις οποίες οι αδικαιολόγητες δεν ξεπερνούν τις 64 αλλά ο μέσος όρος στα προφορικά του δεν είναι πάνω από 15.

Απάντηση άσκησης 5

Flag2← ((S>=64) ΚΑΙ (S<=114) ΚΑΙ (ΑΔ>64)) Η ((S>114) ΚΑΙ (S<164) ΚΑΙ (ΑΔ<=64) ΚΑΙ (ΜΟ<=15))

Άσκηση 6 **Σχετικά με τις απουσίες ενός μαθητή σε Λύκειο, ισχύουν τα παρακάτω:**
Ο μαθητής έχει δικαίωμα να δώσει εξετάσεις τον Ιούνιο αν:
α. έχει μέχρι 64 απουσίες ή
β. έχει μέχρι 114 απουσίες από τις οποίες οι αδικαιολόγητες δεν ξεπερνούν τις 64 ή
γ. έχει μέχρι 164 απουσίες από τις οποίες οι αδικαιολόγητες δεν ξεπερνούν τις 64 και ο μέσος όρος στα προφορικά του είναι πάνω από 15.
Ο μαθητής παραπέμπεται για ολική εξέταση το Σεπτέμβριο αν:
α. έχει πάνω από 64 και μέχρι 114 απουσίες και οι αδικαιολόγητες ξεπερνούν τις 64 ή
β. έχει πάνω από 114 και μέχρι 164 απουσίες από τις οποίες οι αδικαιολόγητες δεν ξεπερνούν τις 64 αλλά ο μέσος όρος στα προφορικά του δεν είναι πάνω από 15.
Σε κάθε άλλη περίπτωση ο μαθητής επαναλαμβάνει τη χρονιά.
Να γίνει πρόγραμμα που να διαβάζει τον αριθμό αδικαιολόγητων απουσιών και δικαιολογημένων απουσιών καθώς και το μέσο προφορικό βαθμό και να εμφανίζει την περίπτωση όπου ανήκει ο μαθητής

Απάντηση άσκησης 6

```
ΠΡΟΓΡΑΜΜΑ Φοίτηση
ΜΕΤΑΒΛΗΤΕΣ
      ΑΚΕΡΑΙΕΣ : ΑΑ, ΔΑ , Σ
      ΠΡΑΓΜΑΤΙΚΕΣ : ΜΟ
ΑΡΧΗ
ΓΡΑΨΕ 'Δώσε αριθμό Αδικαιολόγητων απουσιών'
ΔΙΑΒΑΣΕ ΑΑ

ΓΡΑΨΕ ' Δώσε αριθμό δικαιολογημένων απουσιών'
ΔΙΑΒΑΣΕ ΔΑ

ΓΡΑΨΕ ' Δώσε Μέσο όρο '
ΔΙΑΒΑΣΕ ΜΟ
Σ←ΑΑ+ΔΑ

ΑΝ (Σ<=64) Η ((Σ<=114) ΚΑΙ ((Σ-64)<=ΔΑ)) ΤΟΤΕ
            Φ←'Επαρκής'
      ΑΛΛΙΩΣ
            Φ← 'Ανεπαρκής'
ΤΕΛΟΣ_ΑΝ
```

```
ΓΡΑΨΕ ' Η φοίτηση χαρακτηρίζεται : ', Φ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Άσκηση 7 **Να γίνει πρόγραμμα που να υπολογίζει την ενίσχυση υπαλλήλων μιας εταιρίας ανάλογα με τον αριθμό των παιδίων που έχει κάθε εργαζόμενος και συγκεκριμένα:**

Αριθμός παιδιών	Ενίσχυση σε €
0	0
1	150
2	320
3	480
>3	650

Λύση της άσκησης 7

```
ΠΡΟΓΡΑΜΜΑ ενίσχυση_υπαλλήλων
! Πρόγραμμα υπολογισμού της επιδότησης ν υπαλλήλων μιας
! εταιρείας ανάλογα με τον αριθμό παιδιών του καθένα
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: πλήθος, μετρητής, αμοιβή, παιδιά
ΑΡΧΗ
! Εισαγωγή δεδομένων
ΓΡΑΨΕ «Πόσους υπαλλήλους έχει η εταιρεία;»
ΔΙΑΒΑΣΕ  πλήθος
! Εγκυρότητα δεδομένων εισόδου
ΓΙΑ μετρητής ΑΠΟ 1 ΜΕΧΡΙ πλήθος
    ΕΠΑΝΑΛΑΒΕ
    ΓΡΑΨΕ  'Δώστε τον αριθμό των παιδιών του', μετρητής , 'υπαλλήλου'
    ΔΙΑΒΑΣΕ παιδιά
ΜΕΧΡΙΣ_ΟΤΟΥ παιδιά >=0

! Υπολογισμοί
ΑΝ παιδιά = Ο ΤΟΤΕ
    αμοιβή ←Ο
    ΑΛΛΙΩΣ_ΑΝ παιδιά = 1 ΤΟΤΕ
       αμοιβή ← 150
       ΑΛΛΙΩΣ_ΑΝ παιδιά = 2 ΤΟΤΕ
       αμοιβή ← 320
       ΑΛΛΙΩΣ_ΑΝ  παιδιά = 3 ΤΟΤΕ
       αμοιβή ← 480
          ΑΛΛΙΩΣ
```

```
        αμοιβή ← 650
ΤΕΛΟΣ ΑΝ
! Εμφάνιση αποτελεσμάτων
ΑΝ αμοιβή = Ο ΤΟΤΕ
      ΓΡΑΨΕ «Δε δικαιούστε χρήματα»
      ΑΛΛΙΩΣ
      ΓΡΑΨΕ «Τα χρήματα που θα πάρετε είναι», αμοιβή, «ευρώ»
   ΤΕΛΟΣ_ΑΝ
   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ ενίσχυση_υπαλλήλων
```

Άσκηση 8 **Να γίνει πρόγραμμα που θα ζητάει το ΑΦΜ και το εισόδημα μισθωτών, μέχρι που να διαβάσει αρνητικό ΑΦΜ, και να εμφανίζει το φόρο που πρέπει να πληρώσει ο κάθε μισθωτός.**
Αν το εισόδημα είναι αρνητικός αριθμός το πρόγραμμα να ξαναζητάει το εισόδημα από τον χρήστη.
Ο υπολογισμός του φόρου ενός μισθωτού υπολογίζεται σύμφωνα με τον παρακάτω πίνακα:

Εισόδημα (ευρώ)	Φόρος
0 <= εισόδημα <= 6000	0%
6000 < εισόδημα <= 8000	5%
8000 < εισόδημα <= 12500	12%
Εισόδημα > 12500	25%

Λύση της άσκησης 8

```
ΠΡΟΓΡΑΜΜΑ φορολογία_μισθωτών
! Πρόγραμμα υπολογισμού της φορολογίας μισθωτών

ΜΕΤΑΒΛΗΤΕΣ ΑΚΕΡΑΙΕΣ: ΑΦΜ
  ΠΡΑΓΜΑΤΙΚΕΣ: εισόδημα, φόρος

ΑΡΧΗ
  ΓΡΑΨΕ «Δώστε το ΑΦΜ του φορολογούμενου:»
  ΔΙΑΒΑΣΕ ΑΦΜ
   ΟΣΟ ΑΦΜ >= Ο ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ «Δώστε το εισόδημα του φορολογούμενου:»
  ΔΙΑΒΑΣΕ εισόδημα
  ΟΣΟ εισόδημα < Ο ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ «Μη αποδεκτός αριθμός εισοδήματος»
  ΓΡΑΨΕ «Δώστε το εισόδημα του φορολογούμενου:»
  ΔΙΑΒΑΣΕ εισόδημα
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΝ εισόδημα <= 6000 ΤΟΤΕ
    φόρος<—Ο
    ΑΛΛΙΩΣ_ΑΝ εισόδημα <= 8000 ΤΟΤΕ
    Φόρος ← (εισόδημα - 6000)*0.05
     ΑΛΛΙΩΣ_ΑΝ εισόδημα <= 12500 ΤΟΤΕ
    φόρος ← 2000*0.05 + (εισόδημα - 8000)*0.12
     ΑΛΛΙΩΣ
    φόρος ← 2000*0.05 + 4500*0.12 + (εισόδημα -12500)*0.25
     ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ « φορολογία του μισθωτού είναι», φόρος, «ευρώ»
 ΓΡΑΨΕ «Δώστε το ΑΦΜ του φορολογούμενου:»
ΔΙΑΒΑΣΕ ΑΦΜ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟ Σ_Π ΡΟ ΓΡΑΜΜΑΤΟΣ φορολογία_μισθωτών
```

ΣΥΝΘΕΤΟΙ ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ – ΠΙΝΑΚΕΣ

Ότι έχει παρουσιαστεί στο δεύτερο μέρος για τις σύνθετες δομές δεδομένων ισχύουν και στον προγραμματισμό. Στα επόμενα θα δούμε πως δηλώνονται οι πίνακες σ' ένα πρόγραμμα . Κατά τα άλλα οι τεχνικές επεξεργασίας στοιχείων πίνακα , αναζητήσεις , ταξινομήσεις , που έχουμε μάθει ισχύουν τα ίδια.

Ο πίνακας είναι μια μεταβλητή που αναφέρεται σε πολλά στοιχεία , του ίδιου τύπου. Θα πρέπει το πρόγραμμα να γνωρίζει τον τύπο των στοιχείων αυτών και το μέγεθος του πίνακα από την αρχή του προγράμματος (Στατική δομή δεδομένων) . Έτσι λοιπόν αν σε ένα πίνακα θέλουμε να καταχωρίσουμε τις θερμοκρασίες ενός μήνα , θα πρέπει στο τμήμα δηλώσεων μεταβλητών του προγράμματος να γράψουμε

```
ΜΕΤΑΒΛΗΤΕΣ
        ΠΡΑΓΜΑΤΙΚΕΣ : THERMO[30]
```

Αυτή η δήλωση σημαίνει 1) Ότι όλα τα στοιχεία του πίνακα θα είναι πραγματικοί αριθμοί
2) Μπορούμε να καταχωρίσουμε 30 θερμοκρασίες

Απαιτείται ιδιαίτερη προσοχή στους πίνακες διότι είναι μια στατική δομή δεδομένων και δεν έχουμε τη δυνατότητα μέσα στο πρόγραμμα (μετά τη δήλωσή του) να αλλάξουμε ούτε τον τύπο των στοιχείων του , ούτε το μέγεθός του.

Στο 2ο μέρος παρουσιάστηκαν οι τυπικές επεξεργασίες πινάκων : Υπολογισμός αθροισμάτων στοιχείων πίνακα -Εύρεση μέγιστου , ελάχιστου στοιχείου πίνακα.
Στην αναζήτηση και ταξινόμηση στοιχείων πίνακα έγινε η παρουσίαση της γραμμικής –σειριακής μεθόδου (αναζήτηση) και η ταξινόμηση ευθείας ανταλλαγής- φυσσαλίδας .Οι μέθοδοι αυτές δεν

είναι οι καλύτερες δυνατές. Υπάρχουν αλγόριθμοι στις αναζητήσεις και ταξινομήσεις που είναι πάρα πολύ γρήγορες αλλά η παρουσίαση αυτών των θεμάτων είναι εκτός πνεύματος του σχολικού εγχειριδίου.

Ασκήσεις στους πίνακες

Άσκηση **Να γραφεί πρόγραμμα το οποίο να διαβάζει έναν πίνακα Α διαστάσεων 100×100 και στη συνέχεια να υπολογίζει και να εμφανίζει το μέγιστο στοιχείο της κάθε στήλης, καθώς επίσης και το άθροισμά τους (Μεγίστων).**

Λύση

```
ΠΡΟΓΡΑΜΜΑ Μέγιστα
ΜΕΤΑΒΛΗΤΕΣ
      ΑΚΕΡΑΙΕΣ: I, J
      ΠΡΑΓΜΑΤΙΚΕΣ: A [100, 100], SUM, MAX
ΑΡΧΗ
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 100
          ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 100
                    ΔΙΑΒΑΣΕ A[I, J]
          ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

     SUM ← 0
ΓΙΑ J ΑΠΟ 1 ΜΕΧΡΙ 100
          MAX ← A[1, J]
          ΓΙΑ I ΑΠΟ 2 ΜΕΧΡΙ 100
               ΑΝ A[I, J] > MAX ΤΟΤΕ
                              MAX ← A[I, J]
               ΤΕΛΟΣ_ΑΝ
          ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
          ΓΡΑΨΕ 'Το μέγιστο στοιχείο της ', J, 'ης στήλης είναι ίσο με ', MAX
          SUM ← SUM + MAX
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Το άθροισμα των μεγίστων στοιχείων ανά στήλη είναι ίσο με', SUM
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Άσκηση **Στα πλαίσια μιας εξέτασης 100 διαγωνιζόμενοι πρόκειται να υποβληθούν σε προφορικές και γραπτές εξετάσεις .Οι προφορικοί τους βαθμοί τοποθετούνται σ ένα μονοδιάστατο πίνακα με την ονομασία PROF ενώ οι γραπτοί τους σ ένα μονοδιάστατο πίνακα με την ονομασία GRAPTOS. Οι τελικοί τους βαθμοί προκύπτουν από το άθροισμα των προφορικών και των γραπτών και τοποθετούνται στο μονοδιάστατο πίνακα TELIKOS.**
Να δοθεί πρόγραμμα που
α) θα διαβάζει τους πίνακες PROF [100] και GRAPTOS[100]
β) θα υπολογίζει τον πίνακα TELIKOS[100]
γ) θα υπολογίζει το μέσο όρο για τους 100 υποψηφίους

δ) θα υπολογίζει το πλήθος των υποψήφιων με τελικό βαθμό πάνω από τον μέσο όρο καθώς και το πλήθος των υποψηφίων με βαθμό κάτω από το μέσο όρο.

```
ΠΡΟΓΡΑΜΜΑ εξετάσεις
ΜΕΤΑΒΛΗΤΕΣ
ΠΡΑΓΜΑΤΙΚΕΣ   PROF [ 100 ], GRAPTOS [ 100], TELIKOS[100], Σ, ΜΟ
ΑΚΕΡΑΙΕΣ      Ι, Π, Κ


ΑΡΧΗ
ΓΙΑ   Ι ΑΠΟ  1   ΜΕΧΡΙ  Ν
       ΓΡΑΨΕ 'δώσε   τον ', Ι ,' προφορικό βαθμό'
       ΔΙΑΒΑΣΕ   PROF [ Ι ]
       ΓΡΑΨΕ 'δώσε   τον', Ι ,' γραπτό βαθμό'
       ΔΙΑΒΑΣΕ   GRAPTOS [ Ι ]
      TELIKOS[ Ι ] ← PROF[ Ι ] + GRAPTOS[ Ι ]
       Σ ← Σ + TELIKOS [ Ι ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ←Σ/100
     Π←0
      Κ←0
    ΓΙΑ   Ι   ΑΠΟ   1 ΜΕΧΡΙ   Ν
                    ΑΝ    TELIKOS[ Ι ] >   ΜΟ ΤΟΤΕ
                          Π←Π+1
                           ΑΛΛΙΩΣ
                               Κ ← Κ+1
                ΤΕΛΟΣ_ΑΝ
       ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
       ΓΡΑΨΕ «Ο μέσος όρος είναι» , ΜΟ
      ΓΡΑΨΕ «Πλήθος υποψηφίων με βαθμό πάνω από το ΜΟ», Π
      ΓΡΑΨΕ  «Πλήθος υποψηφίων με βαθμό κάτω από το ΜΟ» , Κ
ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

Τμηματικός προγραμματισμός σημαίνει το 'σπάσιμο' ενός προγράμματος , που μπορεί να είναι αρκετά περίπλοκο , σε επιμέρους απλούστερους υποαλγορίθμους ή υποπρογράμματα. . Θα έχετε ήδη αντιληφθεί ότι η αντιμετώπιση ενός προβλήματος αποτελείται από τυποποιημένες διαδικασίες ή ενέργειες (π.χ. είσοδο δεδομένων, επεξεργασίες, έξοδο αποτελεσμάτων). Κάθε μία από αυτές τις διαδικασίες μπορεί να αποτελεί ένα υποπρόγραμμα (υποαλγόριθμο) ή να 'σπάει' πάλι σε επιμέρους υποπρογράμματα. Δηλαδή ένα **υποπρόγραμμα** είναι ένα αυτόνομο τμήμα προγράμματος , που ς γράφεται ξεχωριστά από το υπόλοιπο πρόγραμμα. Στο τέλος του προγράμματος. Το αρχικό πρόγραμμα (κύριο πρόγραμμα) με κάποιο τρόπο καλεί κάθε υποπρόγραμμα για να το ενεργοποιήσει.

Υπάρχουν δύο κατηγορίες υποπρογραμμάτων οι **διαδικασίες** (procedures) και οι **συναρτήσεις** (functions).

Η διαδικασία είναι η γενικότερη και ισχυρότερη μορφή υποπρογραμμάτων. Μπορούμε να τη θεωρήσουμε ως μικρό πρόγραμμα διότι είναι δυνατό να εκτελέσει οποιαδήποτε λειτουργία.

Διαδικασίες

Ένα πρόγραμμα σε «ΓΛΩΣΣΑ» που χρησιμοποιεί υποπρογράμματα(διαδικασίες) θα μπορούσε να έχει την παρακάτω μορφή :

```
ΠΡΟΓΡΑΜΜΑ < όνομα προγράμματος>
ΣΤΑΘΕΡΕΣ
    Όνομα-1=τιμή-1
    Όνομα-2=τιμή-2
    Όνομα-3=τιμή-3
     ……
    Όνομα-ν=τιμή-ν
ΜΕΤΑΒΛΗΤΕΣ
     Τύπος-1:Λίστα-1
     Τύπος-2: Λίστα-2
      …….
      Τύπος-ν: Λίστα-ν
ΑΡΧΗ
ΚΑΛΕΣΕ ΥΠΟΠΡΟΓΡΑΜΑ_1
ΚΑΛΕΣΕ ΥΠΟΠΡΟΓΡΑΜΜΑ_2
……..
ΚΑΛΕΣΕ ΥΠΟΠΡΟΓΡΑΜΜΑ_Κ
   <εντολές>
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΥΠΟΠΡΟΓΡΑΜΜΑ_1
ΜΕΤΑΒΛΗΤΕΣ
ΣΤΑΘΕΡΕΣ
ΑΡΧΗ
<εντολές>
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΥΠΟΠΡΟΓΡΑΜΜΑ_2
ΜΕΤΑΒΛΗΤΕΣ
ΣΤΑΘΕΡΕΣ
ΑΡΧΗ
<εντολές>
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Εδώ καλείται ! Το υποπρόγραμμα 1 Η εντολή **που** χρησιμοποιούμε είναι η ΚΑΛΕΣΕ <Όνομα υποπρογράμματος>

Το υποπρόγραμμα (Διαδικασία) έχει την ίδια δομή με το πρόγραμμα

Ας κάνουμε ένα συγκεκριμένο παράδειγμα : θα κατασκευάσουμε ένα υποπρόγραμμα που κάνει μια γραμμή με αστεράκια που θα καλείται από το κύριο πρόγραμμα¨

```
ΠΡΟΓΡΑΜΜΑ ΔΟΚΙΜΗ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΡΧΗ
```

ΚΑΛΕΣΕ γραμμή
<εντολές>
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ γραμμή
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ : i

ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 50
ΓΡΑΨΕ «*»
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Το πρόγραμμα αυτό καλεί το υποπρόγραμμα γραμμή που τυπώνει 50 αστεράκια που δημιουργούν μια γραμμή.
Θα είχε ενδιαφέρον να μπορούσαμε να προσδιορίσομε το μήκος της γραμμής (Πλήθος από αστεράκια) μέσα από το κυρίως πρόγραμμα.

Παράμετροι Διαδικασιών

Η παρακάτω μορφή Διαδικασίας «παίρνει» μια πληροφορία από το κυρίως πρόγραμμα για το πόσα αστεράκια θα τυπώσει. Έτσι μέσα από το κυρίως πρόγραμμα αποφασίζουμε το μήκος της γραμμής.

ΠΡΟΓΡΑΜΜΑ ΔΟΚΙΜΗ2
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ : x
ΑΡΧΗ
Χ←70
ΚΑΛΕΣΕ γραμμή(x)
Χ←100
ΚΑΛΕΣΕ γραμμή(χ)
<εντολές>
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Η τιμή της μεταβλητής χ (70) μεταβιβάζεται στην μεταβλητή – φορέα της Διαδικασίας plithos

ΔΙΑΔΙΚΑΣΙΑ γραμμή(Plithos)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ : i ,Plithos
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Plithos
ΓΡΑΨΕ «*»
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Η μεταβλητή– φορέα της Διαδικασίας plithos δέχεται την τιμή 70

Το παραπάνω πρόγραμμα θα τυπώσει δύο γραμμές η μία με 70 αστεράκια και η άλλη με 100 αστεράκια.
Εδώ πρέπει να παρατηρήσουμε ότι η τιμή της μεταβλητής Χ μεταβιβάζεται στην μεταβλητή Plithos της διαδικασίας.. Με αυτό τον τρόπο η **μεταβλητή Plithos είναι ο φορέας που μεταβιβάζεται η πληροφορία στο υποπρόγραμμα «γραμμή»**
Ένας τέτοιος φορέας που μεταβιβάζει πληροφορίες :

- Από το κυρίως πρόγραμμα στο υποπρόγραμμα (Διαδικασία)

- Από το υποπρόγραμμα στο κυρίως πρόγραμμα

Ονομάζεται **παράμετρος.** Το παραπάνω παράδειγμα χρησιμοποιεί τη διαδικασία «γραμμή» με μία παράμετρο . Θα μπορούσαμε να χρησιμοποιήσουμε περισσότερες παραμέτρους και μάλιστα διαφορετικού τύπου. (Ακέραιες, Πραγματικές, Χαρακτήρες, Λογικές)

```
ΠΡΟΓΡΑΜΜΑ  ΔΟΚΙΜΗ3
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ : x
        ΧΑΡΑΚΤΗΡΕΣ : y
     ΑΡΧΗ
Χ←70
Υ←' ='
ΚΑΛΕΣΕ γραμμή(x,y)
Χ←100
Υ←' +'
ΚΑΛΕΣΕ γραμμή(χ,y)
<εντολές>
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Μεταβιβάζονται δύο τιμές διαφορετικού τύπου στο χ η τιμή 00, στο y η τιμή '='

```
ΔΙΑΔΙΚΑΣΙΑ γραμμή(Plithos,xarakt)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ : i ,Plithos
ΧΑΡΑΚΤΗΡΕΣ : xarakt
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ  Plithos
      ΓΡΑΨΕ xarakt
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Οι μεταβλητές– φορείς της Διαδικασίας plithos δέχονται τις τιμές που διαβιβάζονται από το κυρίως πρόγραμμα

Έτσι το παραπάνω πρόγραμμα θα τυπώσει δύο γραμμές η μία με 70 αστεράκια και η άλλη με 100 «=»

Με τα παραπάνω παραδείγματα παρατηρούμε ότι :

Η γενική μορφή μιας διαδικασίας είναι:

ΔΙΑΔΙΚΑΣΙΑ όνομα (λίστα παραμέτρων)
Τμήμα δηλώσεων μεταβλητών
ΑΡΧΗ
….
εντολές
….

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Να μπούν και άλλα παραδείγματα !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Συναρτήσεις

Η συνάρτηση είναι η πιο περιορισμένη μορφή υποπρογραμμάτων. Ο συναρτήσεις χωρίζονται σε δύο κατηγορίες :

- Ενσωματωμένες συναρτήσεις
- Οριζόμενες από τον χρήστη

Οι ενσωματωμένες συναρτήσεις υπάρχουν σε μια βιβλιοθήκη της «ΓΛΩΣΣΑΣ» και είναι διαθέσιμες .

Αυτές είναι :

Συνάρτηση	**Επιστρέφει Τιμή**	**Συνάρτηση**	**Επιστρέφει Τιμή**
ΗΜ(χ)	Ημίτονο του χ	ΛΟΓ(χ)	Φυσικό λογάριθμο του χ
ΣΥΝ(χ)	Συνημίτονο του χ	Ε(χ)	e^x
ΕΦ(χ)	Εφαπτομένη του χ	Α_Μ(χ)	Ακέραιο μέρος του χ
Τ_Ρ(Χ)	Τετραγωνική ρίζα του χ	Α_Τ(χ)	Απόλυτη τιμή του χ

Μια από τις παραπάνω συναρτήσεις μπορεί να χρησιμοποιηθεί σε μια αλγεβρική παράσταση (έκφραση).

Για παράδειγμα :

1) Χ←2*π/3

Π← ΗΜ(χ) Εδώ καλείται η συνάρτηση ΗΜ(χ) στη παράμετρο χ περνάμε την τιμή 2*π/3 και επιστρέφει στο σημείο που κλήθηκε το αποτέλεσμα που αποδίδεται στη μεταβλητή Π

2) Ο τύπος επίλυσης της δευτεροβάθμιας εξίσωσης

$$R = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Θα πρέπει να γραφεί σύμφωνα με το παρακάτω τμήμα προγράμματος

```
D←b^2-4*a*c
ΑΝ D>=0 ΤΟΤΕ
      R←(-b+Τ_Ρ(D))/(2*a)
ΤΕΛΟΣ_ΑΝ
```

Γίνεται κλήση της ενσωματωμένης συνάρτησης Τ_Ρ(Χ) της διαβιβάζουμε σαν παράμετρο την τιμή της D και επιστρέφει την τετραγωνική ρίζα της D

Συναρτήσεις οριζόμενες από τον χρήστη

Εκτός από τις ενσωματωμένες συναρτήσεις η «ΓΛΩΣΣΑ» υποστηρίζει και συναρτήσεις που τις κατασκευάζει ο χρήστης

Οι ενσωματωμένες συναρτήσεις έχουν ένα χαρακτηριστικό : **Δέχονται σαν όρισμα μία ή περισσότερες τιμές και επιστρέφουν στο όνομά τους μία και μοναδική τιμή**

Υπάρχει η δυνατότητα ο χρήστης να κατασκευάσει τις δικιές του συναρτήσεις που να συμπεριφέρονται με τον ίδιο ακριβώς τρόπο.

Μια συνάρτηση που ορίζεται από τον χρήστη έχει την παρακάτω δομή:

ΣΥΝΑΡΤΗΣΗ όνομα (λίστα παραμέτρων) : τύπος συνάρτησης
Τμήμα δηλώσεων μεταβλητών
ΑΡΧΗ
….
όνομα ← έκφραση
…
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Έτσι λοιπόν μπορούμε να πούμε ότι συνάρτηση είναι ένα σύντομο πρόγραμμα **που το όνομα της επιστέφει μία και μόνο τιμή** στο σημείο που κλήθηκε .

Παράδειγμα 1 **Να γίνει συνάρτηση δέχεται ως παράμετρο έναν ακέραιο αριθμό (τη μεταβλητή Ar) και επιστρέφει το τετράγωνό του.**

```
ΣΥΝΑΡΤΗΣΗ Τετράγωνο(Ar): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
      ΠΡΑΓΜΑΤΙΚΕΣ: Ar
ΑΡΧΗ
Τετράγωνο ← Ar ^ 2
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Παράδειγμα 2 **Να γραφτεί ένα πρόγραμμα που να υπολογίζει το εμββαδό τριγώνων με δεδομένα τη βάση και το ύψος.**

Επειδή η λειτουργία που θα εκτελεί είναι ένας υπολογισμός και συνεπώς θα επιστρέφει μία τιμή είναι καταλληλότερο να χρησιμοποιηθεί συνάρτηση. Η τιμή που επιστρέφει είναι ένας πραγματικός αριθμός.

```
ΠΡΟΓΡΑΜΜΑ Υπολογισμός_εμβαδών_Τριγώνων
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ :  Β,Υ
        ΠΡΑΓΜΑΤΙΚΕΣ :  Ε
ΑΡΧΗ
        ! Τα δεδομένα για το 1ο τρίγωνο
        Β ←12
        Υ ← 4
        Ε ← Εμβαδόν_τριγώνου (Β,Υ)
        ΓΡΑΨΕ "Το εμβαδόν είναι ", Ε

        ! Τα δεδομένα για το 2ο τρίγωνο
        Β ← 18
        Υ ← 6
        Ε ← Εμβαδόν_τριγώνου (Β,Υ)
        ΓΡΑΨΕ "Το εμβαδόν είναι ", Ε
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Εμβαδόν_τριγώνου (βάση, ύψος) : ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣς
        ΑΚΕΡΑΙΕΣ :  βάση, ύψος          ! οι ειδικές μεταβλητές για τις παραμέτρους.
ΑΡΧΗ
        Εμβαδόν_τριγώνου ← (βάση * ύψος) / 2
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Εδώ καλείται ! Περνά τις τιμές των *Β,Υ* του κύριου προγράμματος στις αντίστοιχες παραμέτρους *βάση, ύψος* του υποπρογράμματος. Κατόπιν, η συνάρτηση υπολογίζει το εμβαδόν και επιστρέφει την τιμή μέσω του ονόματός της. Αυτή, εκχωρείται στην Ε

Κι εδώ καλείται δεύτερη φορά ,όπου περνά άλλες τιμές.

Παράδειγμα 3 **Να γίνει πρόγραμμα που για κάθε θετικό αριθμό που θα δίνουμε να υπολογίζει το τετράγωνό του. Ο υπολογισμός θα γίνει με συνάρτηση . Η Είσοδος και η έξοδος των αποτελεσμάτων με διαδικασία.**

Το κυρίως πρόγραμμα έχει ως εξής;

```
ΠΡΟΓΡΑΜΜΑ Υπολογισμός_τετραγώνου
ΜΕΤΑΒΛΗΤΕΣ
        ΑΚΕΡΑΙΕΣ: x, y
ΑΡΧΗ
ΚΑΛΕΣΕ Είσοδος_δεδομένων(x)
      y ← Τετράγωνο(x)
ΚΑΛΕΣΕ Εμφάνιση_αποτελεσμάτων(y)
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
!-----------------------------------------------------------

ΔΙΑΔΙΚΑΣΙΑ Είσοδος_δεδομένων(Αριθμός)
ΜΕΤΑΒΛΗΤΕΣ
          ΑΚΕΡΑΙΕΣ: Αριθμός
ΑΡΧΗ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Δώστε ένα θετικό ακέραιο: '
            ΔΙΑΒΑΣΕ Αριθμός
ΜΕΧΡΙΣ_ΟΤΟΥ Αριθμός > 0
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

!-----------------------------------------------------------

ΔΙΑΔΙΚΑΣΙΑ Εμφάνιση_αποτελεσμάτων(Αριθμός)
ΜΕΤΑΒΛΗΤΕΣ
       ΑΚΕΡΑΙΕΣ: Αριθμός
ΑΡΧΗ
     ΓΡΑΨΕ 'Το αποτέλεσμα είναι: ', Αριθμός
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

!-----------------------------------------------------------

ΣΥΝΑΡΤΗΣΗ Τετράγωνο(Ar): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
       ΠΡΑΓΜΑΤΙΚΕΣ: Ar
ΑΡΧΗ
Τετράγωνο ← Ar ^ 2
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Εξασφάλιση ότι ο αριθμός είναι θετικός (Έλεγχος εγκυρότητας)

Άσκηση 1 **Να γίνει πρόγραμμα το οποίο:**
α) Να διαβάζει 2 πραγματικούς αριθμούς α, β.
β) Να λύνει την εξίσωση αx+β=0.

Παρατηρήσεις :
- Η επίλυση της εξίσωσης να γίνεται με ΔΙΑΔΙΚΑΣΙΑ της μορφής Επίλυση_εξίσωσης(α, β, x, είδος)
όπου:
α, β: Οι τιμές των συντελεστών της εξίσωσης (είσοδοι ΔΙΑΔΙΚΑΣΙΑΣ)
χ: Η λύση της εξίσωσης (έξοδος ΔΙΑΔΙΚΑΣΙΑΣ)

είδος: ΑΚΕΡΑΙΑ μεταβλητή (έξοδος ΔΙΑΔΙΚΑΣΙΑΣ)
που περιγράφει το είδος της εξίσωσης ως
εξής:
1 = Η εξίσωση έχει μια λύση
2 = Η εξίσωση είναι ΑΟΡΙΣΤΗ
3 = Η εξίσωση είναι ΑΔΥΝΑΤΗ

- Η εμφάνιση της τιμής του x (στο κυρίως πρόγραμμα) να γίνεται μόνο αν εξίσωση έχει μία λύση. Αλλιώς να εμφανίζεται ενημερωτικό μήνυμα.

```
ΠΡΟΓΡΑΜΜΑ Επίλυση_πρωτοβάθμιας_εξίσωσης

ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: α, β, x
  ΑΚΕΡΑΙΕΣ: είδος

ΑΡΧΗ

  ΓΡΑΨΕ 'Δώσε το συντελεστή α:'
  ΔΙΑΒΑΣΕ α
  ΓΡΑΨΕ 'Δώσε το συντελεστή β:'
  ΔΙΑΒΑΣΕ β

  ΚΑΛΕΣΕ Επίλυση_εξίσωσης(α, β, x, είδος)

  ΕΠΙΛΕΞΕ είδος
        ΠΕΡΙΠΤΩΣΗ 1
                ΓΡΑΨΕ 'Η εξίσωση έχει λύση την τιμή x=', x
        ΠΕΡΙΠΤΩΣΗ 2
                ΓΡΑΨΕ 'Η εξίσωση είναι ΑΟΡΙΣΤΗ'
        ΠΕΡΙΠΤΩΣΗ 3
                ΓΡΑΨΕ 'Η εξίσωση είναι ΑΔΥΝΑΤΗ'
  ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
!-------------------------------------------------------------

ΔΙΑΔΙΚΑΣΙΑ Επίλυση_εξίσωσης(α, β, x, είδος)
ΜΕΤΑΒΛΗΤΕΣ
   ΠΡΑΓΜΑΤΙΚΕΣ: α, β, x
   ΑΚΕΡΑΙΕΣ: είδος
ΑΡΧΗ
 ΑΝ α<>0 ΤΟΤΕ
    x ← -β/α
    είδος ← 1    ! Δηλ. η εξίσωση έχει μια λύση
  ΑΛΛΙΩΣ _ ΑΝ β=0 ΤΟΤΕ
      είδος ← 2  ! Δηλ. η εξίσωση είναι αόριστη
    ΑΛΛΙΩΣ
      είδος ← 3  ! Δηλ. η εξίσωση είναι αδύνατη
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

Άσκηση 2 **Ταξινόμηση φυσαλίδας Να γίνει πρόγραμμα που να διαβάζει 10 ακεραίους αριθμούς τους οποίους θα αποθηκεύει σε έναν πίνακα Π. Στη συνέχεια να ταξινομεί αυτούς τους αριθμούς σε αύξουσα σειρά με τη μέθοδο της φυσσαλίδας.**

ΣΗΜΕΙΩΣΗ:
- Η αντιμετάθεση που απαιτεί η μέθοδος να γίνεται με χρήση ΔΙΑΔΙΚΑΣΙΑΣ.

```
ΠΡΟΓΡΑΜΜΑ Ταξινόμηση_φυσσαλίδας
ΣΤΑΘΕΡΕΣ
  N=10
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: i, j, Π[N]
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
    ΓΡΑΨΕ 'Δώσε το ', i,'ο στοιχείο του πίνακα:'
    ΔΙΑΒΑΣΕ Π[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N
    ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1
      ΑΝ Π[j-1]>Π[j] ΤΟΤΕ
        ΚΑΛΕΣΕ Αντιμετάθεσε(Π[j-1],Π[j])
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

  ΓΡΑΨΕ 'Ακολουθεί ο πίνακας ταξινομημένος σε αύξουσα σειρά:'
  ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
    ΓΡΑΨΕ i,'ο στοιχείο: ', Π[i]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

!-------------------------------------------------------------
ΔΙΑΔΙΚΑΣΙΑ Αντιμετάθεσε(αριθμός1, αριθμός2)
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: αριθμός1, αριθμός2, temp
ΑΡΧΗ
  temp ←αριθμός1
  αριθμός1←αριθμός2
  αριθμός2 ←temp
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
!-------------------------------------------------------------
```