

242 - Εισαγωγή στους Η/Υ

Τμήμα Μαθηματικών,
Πανεπιστήμιο Ιωαννίνων

Ακαδημαϊκό Έτος 2018-2019

Άρτια Α.Μ. (0-2-4-6-8)

Νικόλαος Γλυνός

207α, Β' όροφος

Τηλ: 8251

e-mail: nglinos@uoi.gr

Διδασκαλία και Τρόπος Εξέτασης

- Κάθε εβδομάδα
 - 3ωρη διάλεξη: Θεωρία + κατανόηση εννοιών
 - 2ωρο εργαστήριο: πρακτική εξάσκηση μέσα από συγκεκριμένα προβλήματα
- Εργαστήρια:
 - 7~ περίπου εβδομαδιαία εργαστήρια (εξέταση εργαστηριακών ασκήσεων)
 - Σκοπός των εργαστηρίων είναι η εμπέδωση της ύλης, η εξάσκηση και η λύση αποριών
 - Η παρακολούθηση είναι υποχρεωτική και είναι το *διαβατήριο* για να λάβετε μέρος στις τελικές εξετάσεις
 - Μπορείτε να λείψετε σε **2 το πολύ** εργαστήρια για σοβαρό λόγο

Διδασκαλία και Τρόπος Εξέτασης

- Εργαστήριο Η/Υ: 1^ο όροφο δίπλα από το «Αναγνωστήριο»,
- Στα Εργαστήρια θα πρέπει:
 - Να κατανοείτε το πρόβλημα και να προσπαθείτε να σχεδιάσετε την λύση **πριν έλθετε στο εργαστήριο.**
 - Να συμμετέχετε ενεργά και να είστε προετοιμασμένοι να απαντάτε σε ερωτήσεις σχετικά με το μάθημα
- Για να είναι επιτυχής η παρακολούθηση των εργαστηρίων θα πρέπει:
 1. να μην έχετε παραπάνω από δύο απουσίες
 2. να έχετε προσπαθήσει ΟΛΕΣ τις εργασίες που θα σας ζητούνται στα εργαστήρια

Χρήσιμο Υλικό

- Διδακτικό υλικό, Βιβλία:
- C Προγραμματισμός, 7η Έκδοση, Abbey Deitel, Harvey Deitel.
- C++ ΓΙΑ ΜΑΘΗΜΑΤΙΚΟΥΣ: ΜΙΑ ΕΙΣΑΓΩΓΗ ΓΙΑ ΣΠΟΥΔΑΣΤΕΣ ΚΑΙ ΚΑΘΗΓΗΤΕΣ, EDWARD SCHEINERMAN.
- ΜΑΘΕΤΕ ΤΗ C++ ΑΠΟ ΤΟ ΜΗΔΕΝ, HERBERT SCHILDT.
- C++ Προγραμματισμός, 9η Εκδ., Paul J. Deitel, Harvey M. Deitel.
- C++, 9η Έκδοση, Savitch Walter.
- Πλήρης C++, Savitch Walter.
- Διαφάνειες και εργαστηριακές ασκήσεις:
ιστότοπος μαθήματος → users.uoi.gr/nglinos

Βαθμολογία

- 70% έχει βάρος η τελική εξέταση
- 30% έχει βάρος η βαθμολογία εργαστηρίων (test/εξέταση εργαστήριο)
- Προϋπόθεση: επιτυχή παρακολούθηση εργαστηρίων
- τελικός βαθμός = 0.7 (τ.ε) + 0.3 (εργ)

Τμήματα Εργαστηρίων

- Τα εργαστήρια θα αρχίσουν την

Τετάρτη 13/3/19

****Μην ξεχάσετε να δηλώσετε το μάθημα!!!*

Τμήματα Εργαστηρίων (με βάση τον Α.Μ. - ΑΡΤΙΟΙ):

Γίνονται Τετάρτη και Παρασκευή. Ενημερωθείτε για τις ώρες από την σελίδα του μαθήματος users.uoi.gr/nglinos

- Αλλαγές ΔΕΝ επιτρέπονται.
- Δύο το πολύ απουσίες.
- Παλαιότεροι φοιτητές: Πρέπει να συμμετάσχουν στην εξέταση του εργαστηρίου για να έχουν και τον αντίστοιχο βαθμό.

Στόχος Μαθήματος

- Εισαγωγή στην επιστήμη της πληροφορικής
- Βασικές αρχές συστηματικού και δομημένου προγραμματισμού
- Σχεδίαση, υλοποίηση λύσης προβλημάτων χρησιμοποιώντας δομημένο προγραμματισμό
 - Εκμάθηση δομημένου προγραμματισμού από την αρχή
- Διαγράμματα ροής προγράμματος, ψευδοκώδικας
- Αναπαράσταση, επεξεργασία δεδομένων

Στόχος Μαθήματος

- Το μέσο μας: Γλώσσα προγραμματισμού C/C++
 - Δυνατή γλώσσα προγραμματισμού
 - Κατάλληλη για όσους αρχίζουν να μαθαίνουν προγραμματισμό αλλά και για έμπειρους προγραμματιστές!
 - Γλώσσα κατάλληλη για όλες τις εφαρμογές αλλά κυρίως για υπολογιστικά απαιτητικές εφαρμογές
- Διαλέξεις (3 ώρες) και Εργαστήρια (2 ώρες)
- Η συμμετοχή στις εργαστηριακές ασκήσεις είναι υποχρεωτική

Τι είναι ο υπολογιστής;

- Υπολογιστής
 - Συσκευή για εκτέλεση υπολογισμών και λογικών αποφάσεων
 - Επεξεργασία *δεδομένων* χρησιμοποιώντας ομάδες εντολών που ονομάζονται *προγράμματα*.
- Hardware – Υλικό
 - Συσκευές που αποτελούν τον υπολογιστή
 - Πληκτρολόγιο, οθόνη, ποντίκι, δίσκοι, μνήμη, CD-ROM και μονάδες επεξεργασίας
- Software – Λογισμικό
 - Προγράμματα που τρέχουν στον υπολογιστή.

Οργάνωση του υπολογιστή

- Υπάρχουν έξι λογικές μονάδες σε κάθε υπολογιστή:
 - Μονάδα εισόδου
 - Παίρνει πληροφορίες από συσκευές εισόδου (πληκτρολόγιο, ποντίκι)
 - Μονάδα εξόδου
 - Στέλνει πληροφορίες (στην οθόνη, στον εκτυπωτή, για έλεγχο άλλων συσκευών)

Μονάδα μνήμης

- Άμεση πρόσβαση, χαμηλή χωρητικότητα, αποθηκεύει πληροφορίες εισόδου

Αριθμητική και λογική μονάδα (ALU)

- Αριθμητικοί υπολογισμοί και λογικές αποφάσεις

Κεντρική Μονάδα επεξεργασίας (CPU)

- Επιβλέπει και συντονίζει τα τμήματα του υπολογιστή

Μονάδα δευτερεύουσας αποθήκευσης

- Φτηνή, μακροχρόνια, υψηλής χωρητικότητας, αποθηκεύει ανενεργά προγράμματα και δεδομένα

Εξέλιξη των λειτουργικών συστημάτων

- Μαζική επεξεργασία (Batch processing)
 - Εκτέλεσε μια εργασία την φορά
- Λειτουργικά συστήματα
 - Διαχείριση μετάβασης ανάμεσα σε εργασίες
 - Αυξημένη απόδοση – ποσότητα εργασιών που επεξεργάζεται ο υπολογιστής
- Πολυπρογραμματισμός (Multiprogramming)
 - Πολλές εργασίες μοιράζονται τους πόρους του συστήματος
- Μοίρασμα χρόνου (Timesharing)
 - Εκτέλεση μικρού τμήματος από την διεργασία ενός χρήστη, μετάβαση στον επόμενο χρήστη
 - Τα προγράμματα φαίνεται να εκτελούνται ταυτόχρονα

Προσωπικός υπολογιστής, κατανεμημένοι υπολογιστές και εφαρμογές πελάτη/εξυπηρετητή (client/server)

- Προσωπικός υπολογιστής:
 - Διαδόθηκε στις πλατιές μάζες από την Apple Computer το 1977
 - Η IBM ακολούθησε το 1981 με τον IBM Personal Computer (το γνωστό «PC»)
 - Μικρό κόστος
 - Αυτόνομες και φορητές μονάδες
- Κατανεμημένος υπολογισμός
 - Ένα πανεπιστήμιο που έχει ένα τοπικό δίκτυο (Local Area Network – LAN)
 - Υπολογιστές που συνδέονται με αυτό
 - Ο υπολογισμός διαμοιράζεται στους υπολογιστές του LAN

Προσωπικός υπολογιστής, κατανεμημένοι υπολογιστές και εφαρμογές πελάτη/εξυπηρετητή (client/server)

- Το μοντέλο client/server
 - Οι εξυπηρετητές (servers) προσφέρουν αρχεία που χρειάζονται συχνότερα οι πελάτες (clients)
 - C και C++ αρκετά δημοφιλείς για τη δημιουργία λειτουργικών συστημάτων, δικτυακών εφαρμογών και εφαρμογών πελάτη/εξυπηρετητή
 - Μία άλλη γλώσσα είναι η Java η οποία χρησιμοποιείται συχνά για εφαρμογές στο Internet

Γλώσσες μηχανής, Γλώσσες Assembly και Γλώσσες Υψηλού Επιπέδου

- Κατηγορίες γλωσσών προγραμματισμού

1. Γλώσσες μηχανής

- Ακολουθίες αριθμών που δίνουν στον υπολογιστή συγκεκριμένες οδηγίες
- Παράδειγμα:

```
1010011100011100101010  
0100101001110001100110  
1001010101001100110101
```

2. Γλώσσες Assembly

- Συντομογραφίες που αναπαριστούν βασικές πράξεις του υπολογιστή (οι οποίες μεταφράζονται σε ακολουθίες αριθμών – γλώσσα μηχανής – από τους assemblers)
- Παράδειγμα:

```
LOAD    BASEPAY  
ADD     OVERPAY  
STORE   GROSSPAY
```

Γλώσσες μηχανής, Γλώσσες Assembly και Γλώσσες Υψηλού Επιπέδου

- Κατηγορίες γλωσσών προγραμματισμού
 3. Γλώσσες υψηλού επιπέδου
 - Μοιάζει με φυσική γλώσσα (έχει επικρατήσει η αγγλική γλώσσα) και χρησιμοποιεί μαθηματικές αναπαραστάσεις (οι οποίες μεταφράζονται σε γλώσσα μηχανής από τους compilers)
 - Παράδειγμα:
grossPay = basePay + overTimePay

Ιστορία της C++

- Η C++ εξελίχθηκε από την C
 - C εξελίχθηκε από 2 προηγούμενες γλώσσες προγραμματισμού, τις BCPL and B
 - ANSI C καθιέρωσε παγκόσμια standards για προγραμματισμό σε C
- C++ αναβαθμίζει C
 - Δίνει δυνατότητες για αντικειμενοστραφή προγραμματισμό
 - Αντικείμενα – επαναχρησιμοποιούμενα συστατικά λογισμικού που μοντελοποιούν κομμάτια του πραγματικού κόσμου
 - Τα αντικειμενοστραφή προγράμματα είναι εύκολα στην κατανόηση, διόρθωση και μεταφορά τους σε άλλο υπολογιστικό περιβάλλον

Ιστορία της Java

- Java
 - Βασίστηκε στην C και στην C++
 - Αναπτύχθηκε το 1991 για «έξυπνες» ηλεκτρονικές συσκευές
 - Επειδή δεν αναπτύχθηκε η αγορά, το project κινδύνευσε να ακυρωθεί
 - Με την έκρηξη του Internet το 1993, το project σώθηκε
 - Η Java χρησιμοποιήθηκε για την δημιουργία σελίδων με δυναμικό περιεχόμενο
 - Η Java ανακοινώθηκε επίσημα το 1995
 - Πλέον χρησιμοποιείται για δημιουργία σελίδων με διαδραστικό περιεχόμενο, για αναβάθμιση ιστοτόπων web, για εφαρμογές σε διάφορες συσκευές (tablets, κινητά)...

Άλλες γλώσσες υψηλού προγραμματισμού

- Κάποιες άλλες γλώσσες που έχουν γίνει ευρύτερα αποδεκτές είναι οι εξής:
 - FORTRAN (FORmula TRANslator)
 - Για επιστημονικές εφαρμογές και εφαρμογές για μηχανικούς
 - COBOL (COmmon Business Oriented Language)
 - Για διαχείριση μεγάλου όγκου δεδομένων σε τραπεζικές και εμπορικές εφαρμογές
 - Pascal
 - Κυρίως για ακαδημαϊκή χρήση και εκμάθηση δομημένου προγραμματισμού
 - BASIC
 - Αναπτύχθηκε το 1965
 - Απλή γλώσσα που μπορεί να βοηθήσει τους αρχάριους να κατανοήσουν βασικές προγραμματιστικές αρχές

Δομημένος προγραμματισμός

- Δομημένος προγραμματισμός
 - Πειθαρχημένη προσέγγιση στην συγγραφή προγραμμάτων
 - Ξεκάθαρο πρόγραμμα, εύκολο στον έλεγχο και την αποσφαλμάτωση (debugging)
 - Η Pascal σχεδιάστηκε με σκοπό να διδάξει τον δομημένο προγραμματισμό
 - Δεν χρησιμοποιείται σε βιομηχανικές ή εμπορικές εφαρμογές
- Multitasking (πολυεπεξερασία)
 - Πολλές διεργασίες εκτελούνται παράλληλα
 - Η standard C δεν έχει threads (νήματα) αλλά κάποιες βιβλιοθηκες για αυτό υπάρχουν
 - Η C++ και η Java επιτρέπουν multithreading (πολυνηματικές)
 - Διεργασίες μπορούν να εκτελούνται παράλληλα

Το Internet και το World Wide Web (Παγκόσμιος Ιστός)

- Το Internet
 - Αναπτύχθηκε πριν από 30 χρόνια, από το Αμερικάνικο Υπουργείο Άμυνας
 - Αρχικά σχεδιάστηκε για να συνδέσει πανεπιστήμια
 - Πλέον προσβάσιμο από δεκάδες εκατομμύρια υπολογιστών
- World Wide Web
 - Διαχείριση εγγράφων με multimedia (πολυμέσα)
 - Έκρηξη του Internet
 - Ανάμιξη υπολογιστών και επικοινωνιών
 - Αλλαγή στον τρόπο λειτουργίας των επιχειρήσεων
 - Οι πληροφορίες είναι άμεσα διαθέσιμες
 - Θα ασχοληθούμε με εφαρμογές Java που χρησιμοποιούν το Internet

Πληροφορική

- Πληροφορική

Ηλεκτρονικοί Υπολογιστές

Μαθηματικά



Σχεδίαση και Κατασκευή

Θεωρία και Αναλυτική
Μέθοδος

- Κεντρική έννοια:
 - υπολογισμός (computation)
- Η επιστήμη της επεξεργασίας πληροφοριών, δηλαδή των μεθόδων καταγραφής, χειρισμού και ανάκτησης πληροφοριών

Επιστήμη της Πληροφορικής

- Πληροφορική: Επίλυση προβλήματος
 - Απαιτήσεις, περιορισμοί προβλήματος
 - Κατανόηση ζητούμενων
 - Οργάνωση δεδομένων
 - Σχεδίαση αποδοτικού αλγορίθμου για την επίλυση του προβλήματος
 - Υλοποίηση λύσης
 - Διεξοδικός έλεγχος σωστής λειτουργίας
 - Υπολογισμός πολυπλοκότητας και θεωρητική απόδειξη ορθότητας
 - Βελτιστοποίηση
 - Ανάλυση πολυπλοκότητας
 - Τεκμηρίωση - Συντήρηση

Επιστήμη της Πληροφορικής

- Σκοπός: να μάθεις να σκέφτεσαι
 - Η Μαθηματική επιστήμη είναι βασικό εργαλείο. Λογική, διακριτά μαθηματικά, στατιστική, πιθανότητες, θεωρία γράφων, και άλλες περιοχές συνεισφέρουν στην θεωρητική θεμελίωση της Πληροφορικής
 - Σκοπός είναι η ανάπτυξη της ικανότητας μαθηματικής επίλυσης απλών αλλά και σύνθετων πρωτότυπων προβλημάτων χωρίς να βασιζόμαστε σε αποστήθιση κανόνων

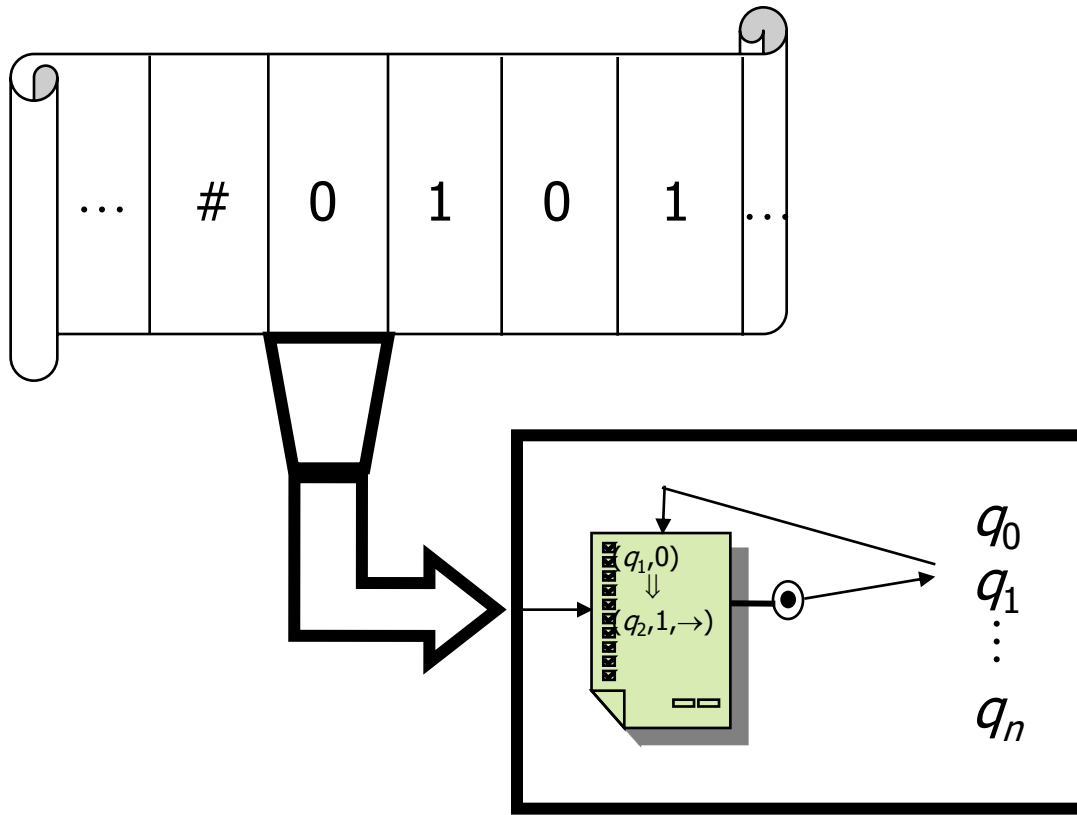
Επιστήμη της Πληροφορικής

- Τριτοβάθμια εκπαίδευση:
 - Η τεχνολογία αλλάζει αέναα και γρήγορα
τα θεμέλια όμως μένουν
 - Αυτά τα θεμέλια είναι η ραχοκοκαλιά στην τριτοβάθμια εκπαίδευση: έμφαση στην αλγοριθμική σκέψη σε αντιδιαστολή με τις τεχνολογικές δεξιότητες
 - Dijkstra:
η Επιστήμη των Υπολογιστών έχει τόση σχέση με τους υπολογιστές όση και η Αστρονομία με τα τηλεσκόπια

Επιστήμη της Πληροφορικής

- Κλάδοι της Πληροφορικής:
 - Γλώσσες Προγραμματισμού
 - Σχεδίαση και Ανάλυση Αλγορίθμων
 - Δομές δεδομένων
 - Θεωρία Γραφημάτων
 - Θεωρία Πολυπλοκότητας
 - Επεξεργασία Φυσικής Γλώσσας
 - Αρχιτεκτονική υπολογιστών και δικτύων
 - Αριθμητικοί και συμβολικοί υπολογισμοί
 - Λειτουργικά συστήματα
 - Μεθοδολογία – τεχνολογία λογισμικού
 - Βάσεις δεδομένων και διαχείριση πληροφοριών
 - Τεχνητή νοημοσύνη και ρομποτική
 - Επικοινωνία ανθρώπου – υπολογιστή (γραφικά)

Η μηχανή Turing: το μαθηματικό μοντέλο του Η/Υ!



A
L
A
N

T
U
R
I
N
G



- Μία άπειρα εκτεινόμενη ταινία χωρισμένη σε κελιά
- Κάθε κελί αποθηκεύει ένα σύμβολο, συνήθως δυαδικό ψηφίο (0 ή 1) ή το κενό (#)
- Μία κεφαλή που διαβάζει το περιεχόμενο ενός κελιού – κίνηση δεξιά/αριστερά
- Μηχανισμός «λήψης αποφάσεων»

Υπολογίζοντας με μία μηχανή Turing!

Το παρακάτω «πρόγραμμα» (είναι ένας πίνακας μεταβάσεων) υπολογίζει τη διαφορά μεταξύ δύο θετικών ακεραίων m και n (μόνο εάν $m > n$, αλλιώς επιστρέφει το 0) που δίνονται στην μορφή $0^m 10^n$ στην ταινία της μηχανής Turing:

	q_0	q_1	q_2	q_3	q_4	q_5	q_6
0	$(q_1, \#, \Delta)$	$(q_1, 0, \Delta)$	$(q_3, 1, A)$	$(q_3, 0, A)$	$(q_4, 0, A)$	$(q_5, \#, \Delta)$	- (σταματά)
1	$(q_5, \#, \Delta)$	$(q_2, 1, \Delta)$	$(q_2, 1, \Delta)$	$(q_3, 1, A)$	$(q_4, \#, A)$	$(q_5, \#, \Delta)$	- (σταματά)
#	- (κρεμά)		$(q_4, \#, A)$	$(q_0, \#, \Delta)$	$(q_6, 0, \Delta)$	$(q_6, \#, \Delta)$	- (σταματά)

Τι είναι ο «αλγόριθμος»;

Μία «λογική» για το πώς θα λυθεί ένα πρόβλημα.

Σκέφτομαι συγκεκριμένα βήματα που έχουν αρχή και τέλος και με οδηγούν σε λύση (ή στον εντοπισμό ότι δεν υπάρχει λύση).

Μπορώ να τον περιγράψω με πολλούς τρόπους π.χ. Καθομιλουμένη γλώσσα!

Πώς περιγράψω τον αλγόριθμο;

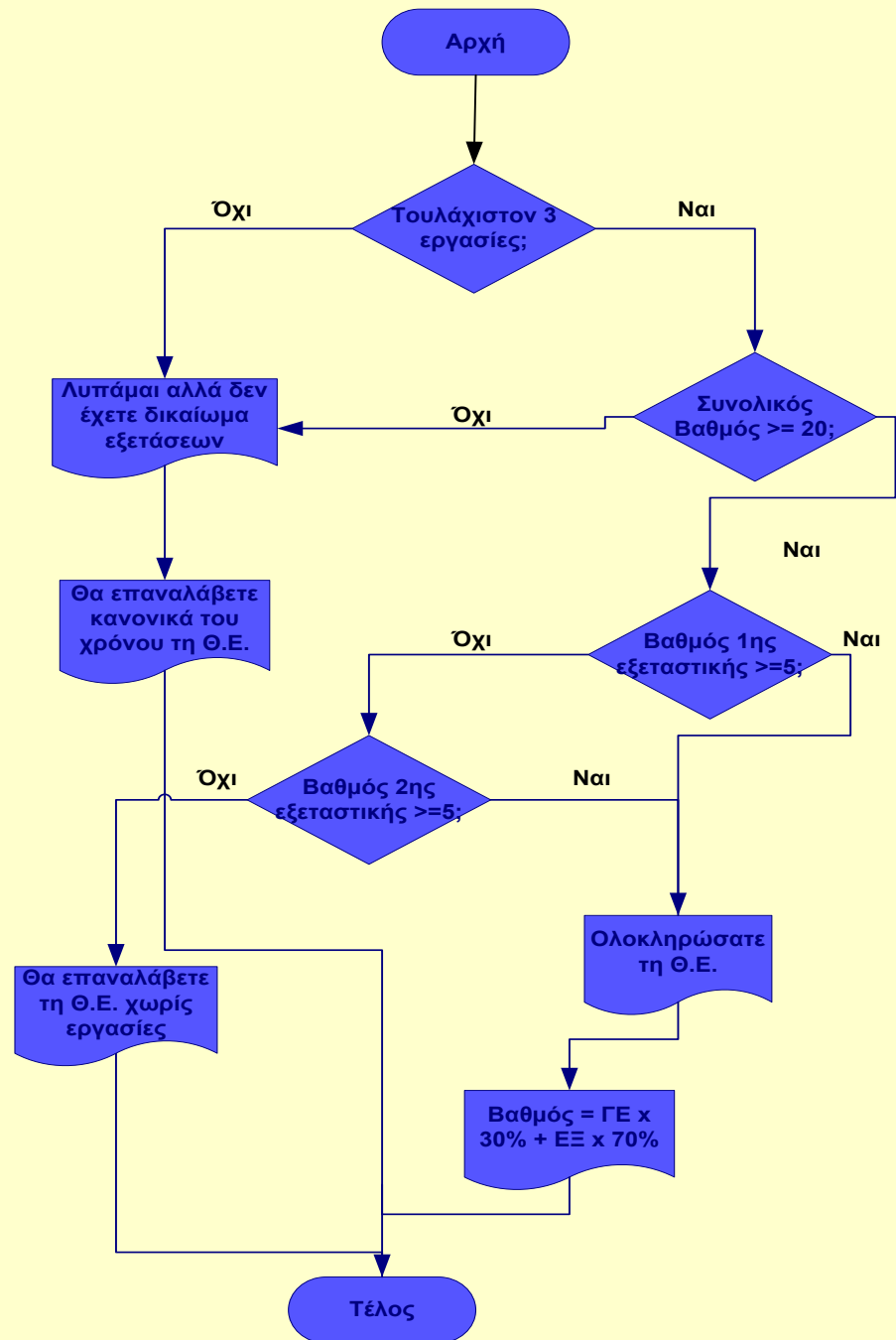
- Η φυσική (καθομιλουμένη γλώσσα) είναι μία λύση, αλλά όχι πάντα πρακτική.
- Χρειάζομαι κάτι πιο δομημένο όπως θα δούμε αργότερα και ο ίδιος ο κώδικας (C++) .
- **Ψευδοκώδικας**: Είναι μία «σύμβαση». Μία δομημένη γλώσσα με συγκεκριμένους κανόνες και σύνταξη.

Ουσία!

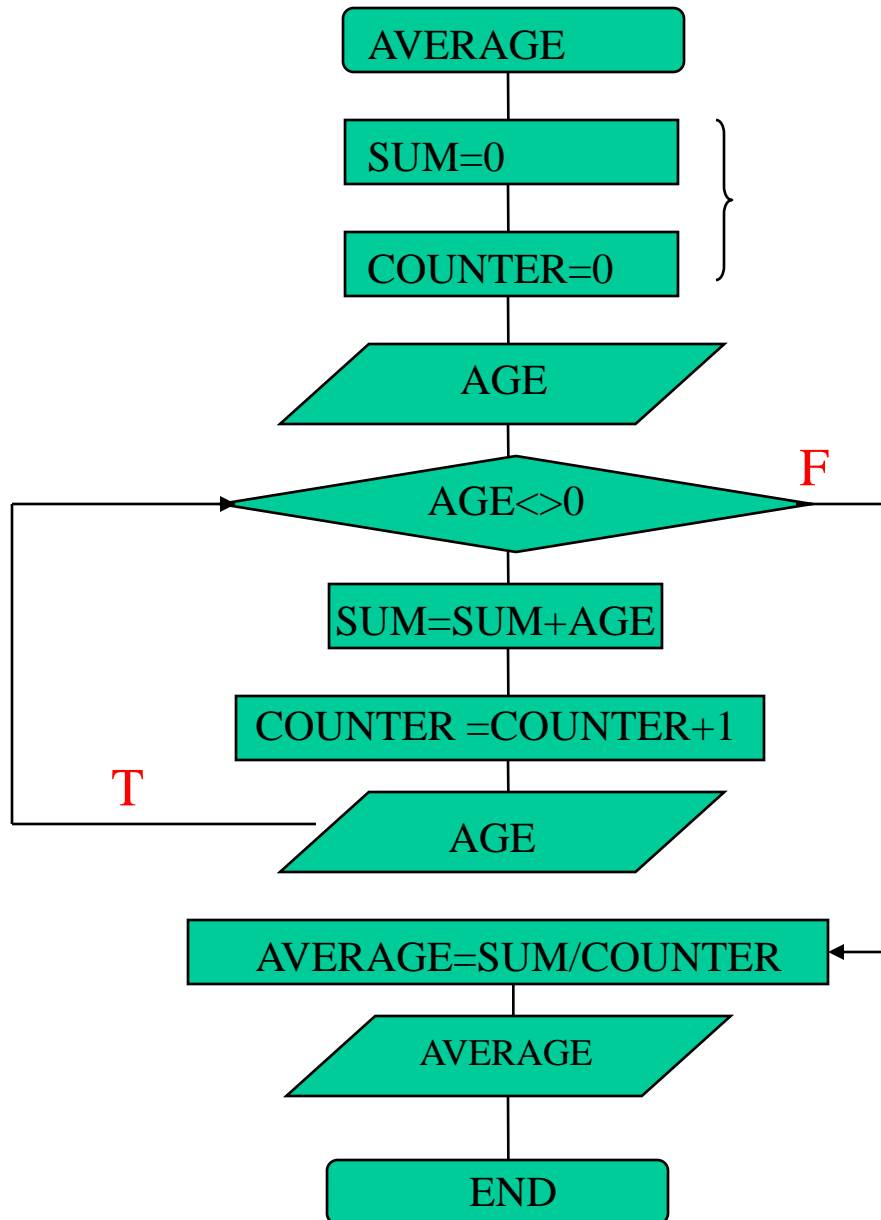
Η ουσία είναι να μπορώ να σκεφτώ ένα σωστό αλγόριθμο.

- Ο **ψευδοκώδικας** είναι μία συμβατική μορφή αναπαράστασης του αλγορίθμου. Η φυσική γλώσσα έκφρασης του **ψευδοκώδικα** μπορεί να είναι η Αγγλική αλλά και η Ελληνική (ΕΑΓ: Ελληνική Αλγοριθμική Γλώσσα).
- Μία **γλώσσα προγραμματισμού** (π.χ. C, C++) είναι ένα εργαλείο για να υλοποιηθεί ο αλγόριθμος στον υπολογιστή.
 - Θα πρέπει να μπορώ να χρησιμοποιώ πολλά εργαλεία... (πολλές γλώσσες)

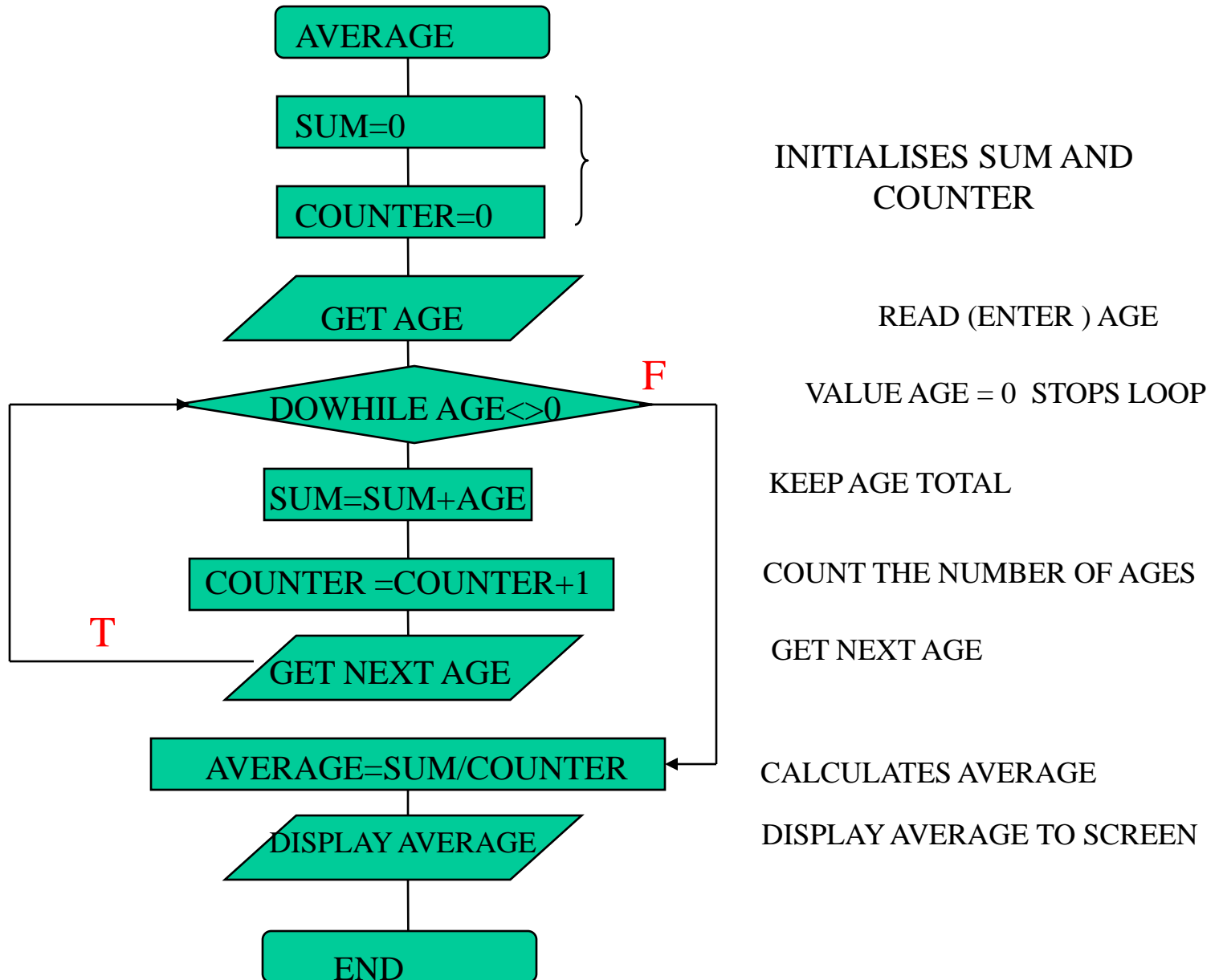
Ένα παράδειγμα Διαγράμματος Ροής (ΔΡΠ)



Ποια η λειτουργία του παρακάτω διαγράμματος ροής προγράμματος;



Ποια η λειτουργία του;



ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

- Ο ψευδοκώδικας είναι μία άτυπη υψηλού επιπέδου περιγραφή των λειτουργιών/βημάτων ενός αλγορίθμου. Είναι λιγότερο αυστηρή από μια γλώσσα προγραμματισμού. Κάνει πιο ελεύθερη περιγραφή ενός αλγορίθμου ώστε αυτός να γίνει άμεσα κατανοητός από τον άνθρωπο αλλά όχι από μια μηχανή. Παραλείπει τις λεπτομέρειες μια γλώσσας προγραμματισμού και κρατάει εκείνα τα στοιχεία που είναι απαραίτητα για την περιγραφή των βημάτων ενός αλγόριθμου.
- Η γλώσσα του ψευδοκώδικας μπορεί να είναι η αγγλική (πιο γνωστός είναι εκείνος που προέρχεται από την PASCAL) και χρησιμοποιεί όρους και λέξεις από τις γλώσσες προγραμματισμού.
- **ΕΑΓ- Ελληνική Αλγοριθμική Γλώσσα:** είναι μια γλώσσα περιγραφής αλγορίθμων στην Ελληνική που κατά βάση προήλθε από την γλώσσα PASCAL. Κατά την διάρκεια του μαθήματος θα δούμε αρκετά παραδείγματα.
- Παρακάτω περιγράφεται γρήγορα ένας ψευδοκώδικας με αγγλικούς όρους που όπως θα δείτε μοιάζει πολύ με αυτούς της γλώσσας προγραμματισμού που θα χρησιμοποιήσουμε για την υλοποίηση των προγραμμάτων μας.

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Μεταβλητή

- Κάθε μέγεθος (έννοια) του προβλήματος που εμπλέκεται στη λύση του πρέπει να αντιπροσωπεύεται από μια μεταβλητή στον αλγόριθμο.
- Κάθε μεταβλητή μπορεί να μην έχει σταθερή τιμή, αλλά να παίρνει διάφορες τιμές, κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου (προγράμματος).
- Το όνομα μιας μεταβλητής περιέχει γράμματα, αριθμούς και την παύλα ('-'), αλλά αρχίζει πάντα από γράμμα. Π.χ. X, MAX2, MAX-2.

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ Τύπος Μεταβλητής

- Τα μεγέθη διαφέρουν ως προς τις ιδιότητες και το είδος των τιμών που παίρνουν.
- Υπάρχουν τέσσερις βασικοί (ή πρωτογενείς) τύποι μεταβλητών:
 - INTEGER π.χ. 5, 18
 - REAL π.χ. 3.1, 24.6
 - CHAR π.χ. “A”, “7”, “DEKA”
 - BOOLEAN : TRUE ή FALSE

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Σταθερές

- Είναι υπολογιστικές οντότητες που έχουν μια σταθερή τιμή, η οποία δεν αλλάζει κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
- Τα ονόματά τους σχηματίζονται με τους ίδιους κανόνες όπως και των μεταβλητών.
- Δεν χρειάζεται να δηλώσουμε τον τύπο μιας σταθεράς, διότι αυτός προκύπτει από την τιμή της.

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Τελεστές

- Είναι σύμβολα που παριστάνουν κάποια στοιχειώδη λειτουργία/επεξεργασία.
 - Αριθμητικοί τελεστές: +, -, *, /, DIV, MOD
 - Τελεστές σύγκρισης ή Συσχετιστικοί τελεστές: = >, >=, <, <=, <>
 - Λογικοί τελεστές: NOT, OR, AND

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Εκφράσεις

Οι εκφράσεις είναι στοιχεία του ψευδοκώδικα που παράγονται από τον συνδυασμό τελεστών και μεταβλητών ή τιμών:

Αριθμητικές: Π.χ. $(5 * X + Y)$, $((K + 3)/4) * L$, $K \text{ DIV } 3$.

Σύγκρισης: Π.χ. $X \geq Y$, $A = B$, $K < L$ και $X \langle \rangle Y$ είναι εκφράσεις σύγκρισης. Η τιμή που επιστρέφει μια έκφραση σύγκρισης είναι τύπου **BOOLEAN**.

Λογικές: Π.χ. $(X > Y) \text{ AND } (X < 10)$, $(A \langle \rangle B) \text{ OR } (A \geq 5)$, $\text{NOT } (X < Y)$ είναι λογικές εκφράσεις. Η τιμή που επιστρέφει μια λογική έκφραση είναι επίσης τύπου **BOOLEAN**

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Δηλώσεις μεταβλητών

Η δήλωση μιας μεταβλητής γίνεται ως εξής:

<λίστα μεταβλητών> : <τύπος> ;

Π.χ.

I, J : INTEGER;

X, Y, Z : REAL;

C : CHAR;

TELOS : BOOLEAN;

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Τύπος πίνακα

- Ο τύπος πίνακα είναι ένας σύνθετος τύπος που αντιπροσωπεύει σύνολα τιμών.
- Ο τύπος πίνακα χρησιμοποιείται όταν πρέπει να χρησιμοποιήσουμε μεταβλητές για να παραστήσουμε σχετικά μεγάλο αριθμό από ομοειδή μεγέθη.

Π.χ. BATHMOS : **ARRAY** [1..20] **OF** INTEGER;

- Η πρόσβαση στις τιμές (ή στοιχεία) ενός πίνακα γίνεται μέσω του ονόματος του πίνακα και ενός δείκτη που δείχνει τη θέση της τιμής στον πίνακα.
- Οι BATHMOS[3], BATHMOS[I] ονομάζονται *μεταβλητές με δείκτη*.

ΠΑΡΟΥΣΙΑΣΗ ΨΕΥΔΟΚΩΔΙΚΑ

Βασικές εντολές ψευδοκώδικα

Εντολή καταχώρησης

Με την εντολή αυτή καταχωρούμε τιμές σε μεταβλητές.

Π.χ. **X := 12** , **X := Y** , **X := (Y + 10)/4** είναι εντολές καταχώρησης.

Εντολή για είσοδο δεδομένων

Για να εισάγονται δεδομένα στον αλγόριθμο (πρόγραμμα) από το πληκτρολόγιο χρησιμοποιείται μια εντολή εισόδου.

Π.χ. **READ (X)**, **READ (X, Y)**

Εντολή για έξοδο αποτελεσμάτων

Για να εκτυπώνονται αποτελέσματα στην οθόνη χρησιμοποιείται μια εντολή εξόδου.

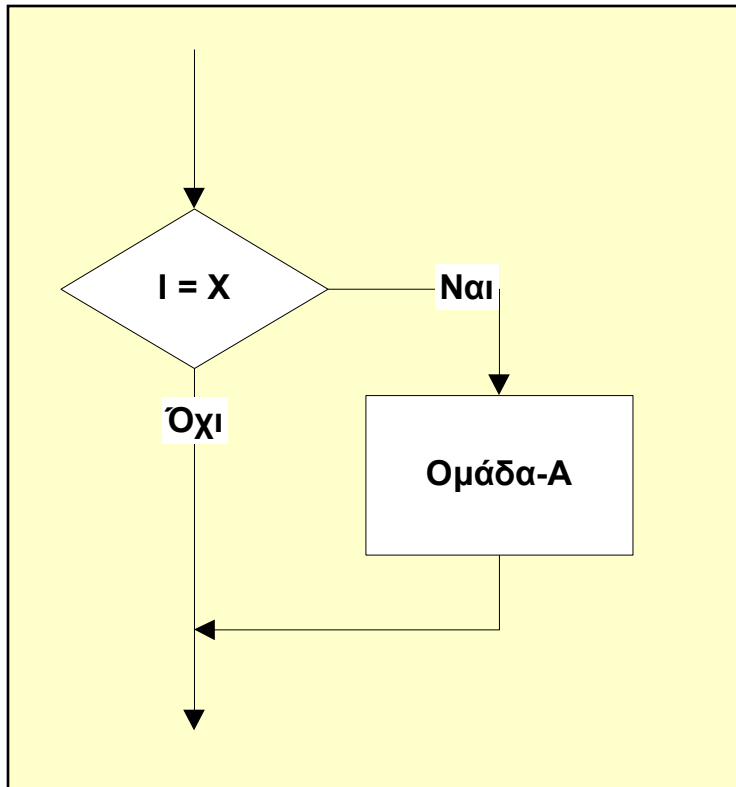
Π.χ. **WRITE («ΤΟ ΕΜΒΑΔΟΝ ΕΙΝΑΙ: », E)**, **WRITE («X = », Y*(Z+2))**, **WRITE («ΤΟ ΕΜΒΑΔΟΝ ΕΙΝΑΙ: », E, «ΚΑΙ Ο ΟΓΚΟΣ: », V)**

Εντολές ελέγχου ροής προγράμματος – παρακάτω if, for, while, do while

- ΕΜΕΙΣ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΣΟΥΜΕ ΤΗΝ C/C++ ΓΙΑ ΝΑ ΥΛΟΠΟΙΗΣΟΥΜΕ ΑΛΓΟΡΙΘΜΟΥΣ.

Δομή Ελέγχου Ροής (if)

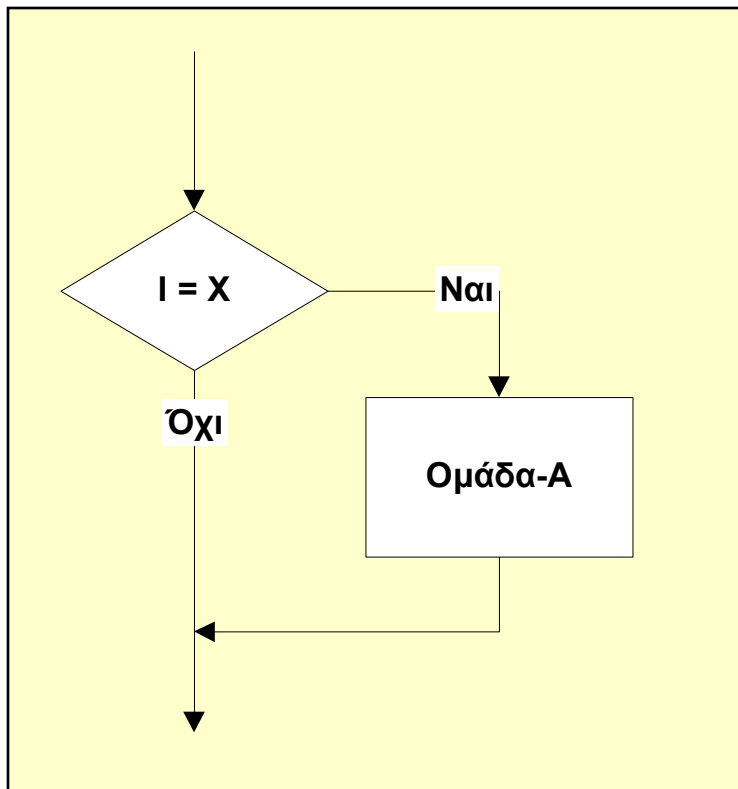
- Η εντολή IF χρησιμοποιείται όταν θέλουμε να ελέγξουμε την ισχύ μίας συνθήκης και ανάλογα να πράξουμε



```
if (I==X)
{
    Ομάδα-A;
}
```

Δομή Ελέγχου Ροής (IF)

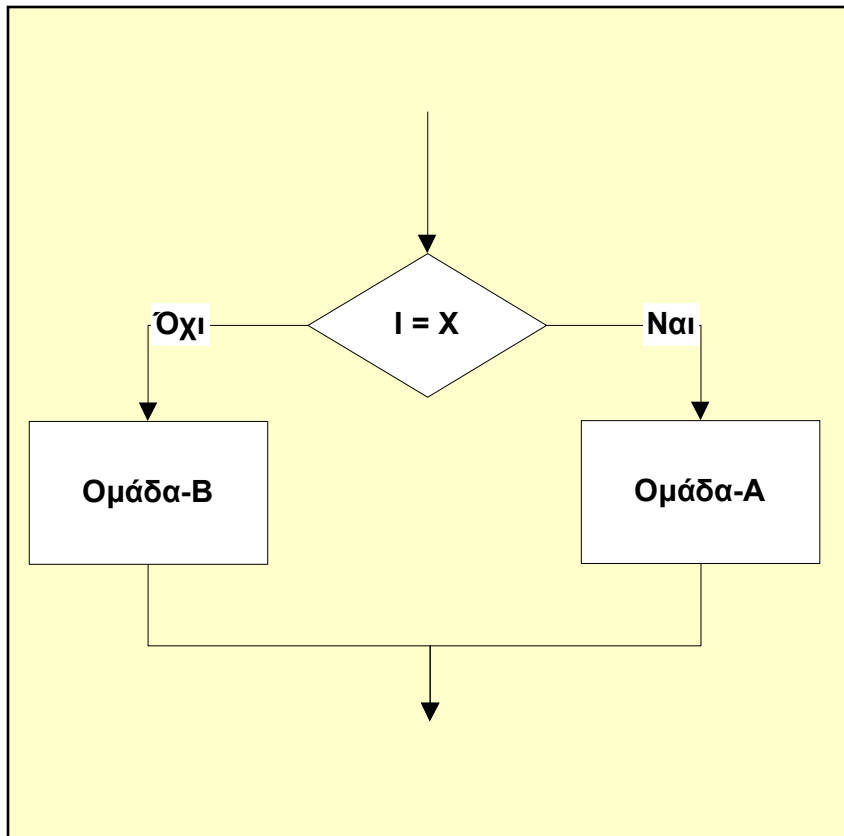
- Η εντολή IF χρησιμοποιείται όταν θέλουμε να ελέγξουμε την ισχύ μίας συνθήκης και ανάλογα να πράξουμε



ΕΑΝ (I = X) ΤΟΤΕ
Ομάδα-A;
ΕΑΝ-ΤΕΛΟΣ;

Δομή Ελέγχου Ροής (if - else)

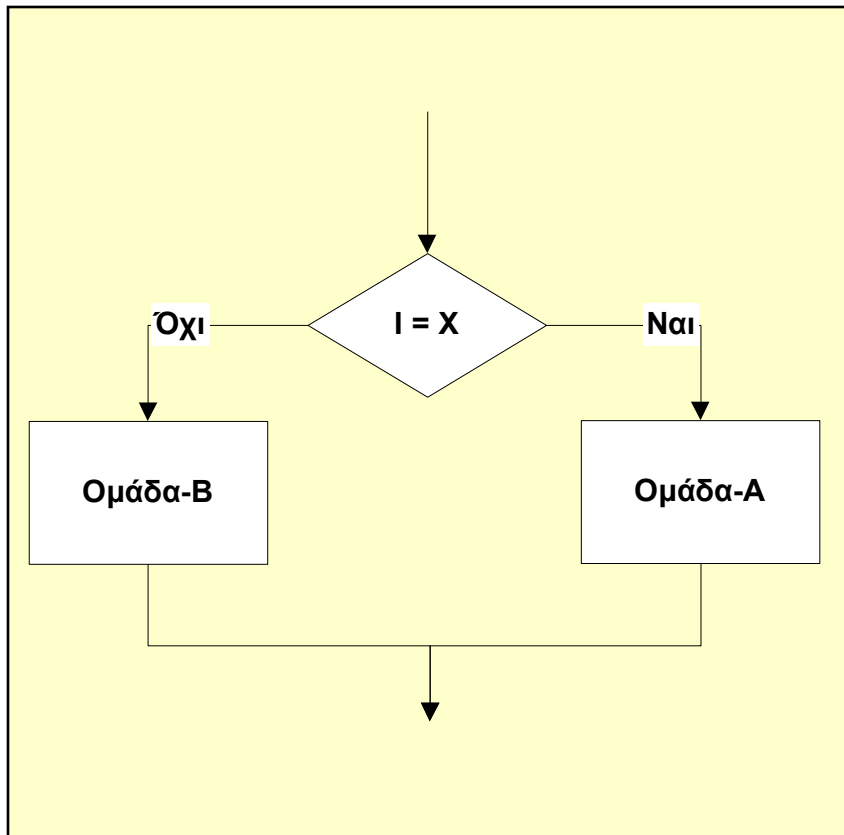
- Η εντολή IF - ELSE χρησιμοποιείται με παρόμοιο τρόπο με την εντολή IF, με τη διαφορά ότι εκτελούμε μία (διαφορετική) ομάδα εντολών και στην περίπτωση που δεν ισχύει η συνθήκη



```
if (I==X)
{
    Ομάδα-A;
}
else
{
    Ομάδα-B;
}
```

Δομή Ελέγχου Ροής (IF - ELSE)

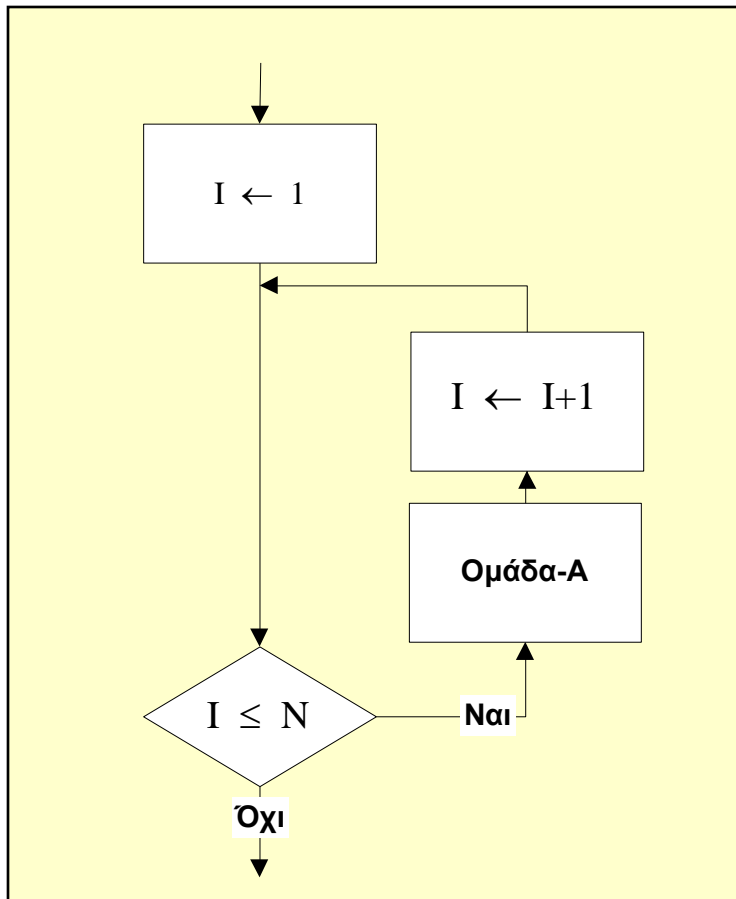
- Η εντολή IF - ELSE χρησιμοποιείται με παρόμοιο τρόπο με την εντολή IF, με τη διαφορά ότι εκτελούμε μία (διαφορετική) ομάδα εντολών και στην περίπτωση που δεν ισχύει η συνθήκη



ΕΑΝ (I = X) ΤΟΤΕ
Ομάδα-A;
ΑΛΛΙΩΣ
Ομάδα-B;
ΕΑΝ-ΤΕΛΟΣ;

Επαναληπτική Δομή (for)

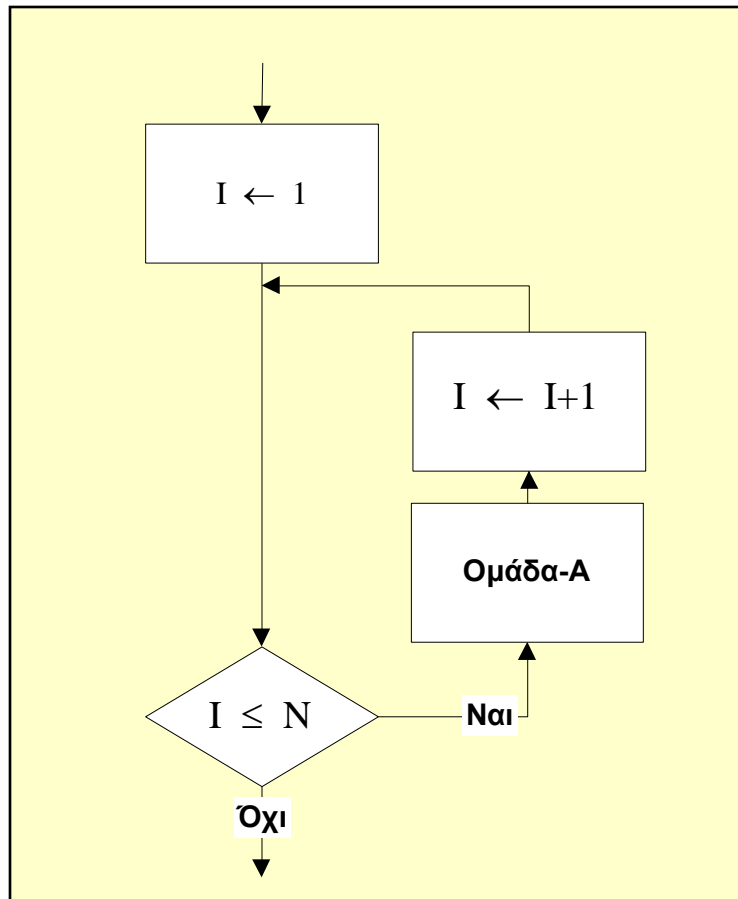
- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Στη FOR ο αριθμός των επαναλήψεων είναι **σταθερός**.



```
for (I=1; I<=N; I++)  
{  
    Ομάδα-A;  
}
```

Επαναληπτική Δομή (FOR)

- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Στη FOR ο αριθμός των επαναλήψεων είναι **σταθερός**.



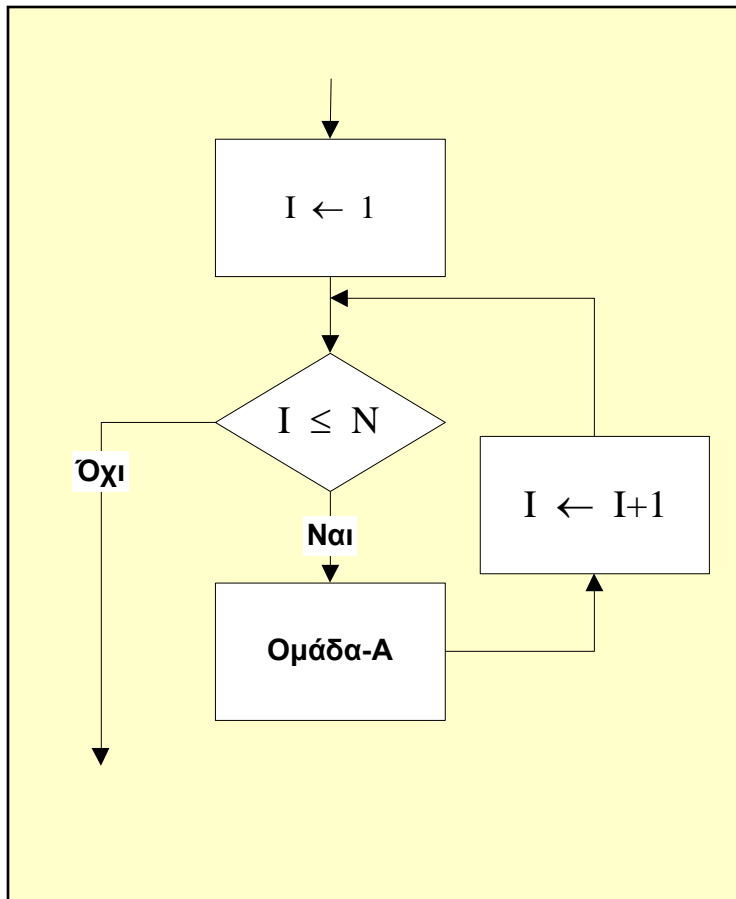
ΓΙΑ $I := 1$ ΕΩΣ N ΕΠΑΝΑΛΑΒΕ

Ομάδα-A;

ΓΙΑ-ΤΕΛΟΣ;

Επαναληπτική Δομή (while)

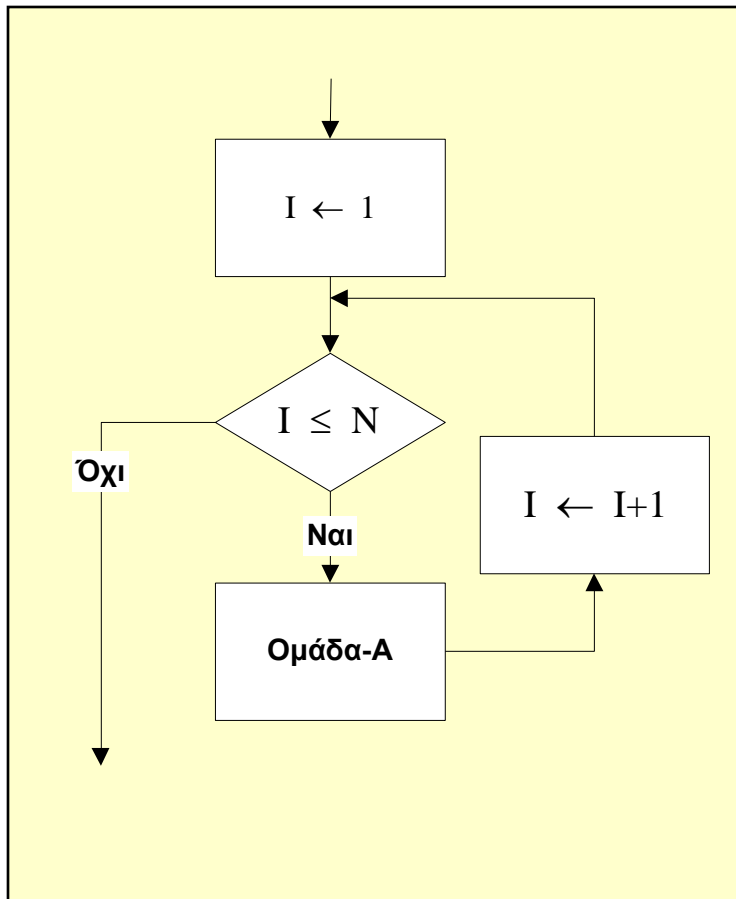
- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Στη WHILE ο αριθμός των επαναλήψεων εξαρτάται από κάποια συνθήκη.



```
I = 1;  
while (I<=N)  
{  
    Ομάδα-A;  
    I = I + 1;  
}
```

Επαναληπτική Δομή (WHILE)

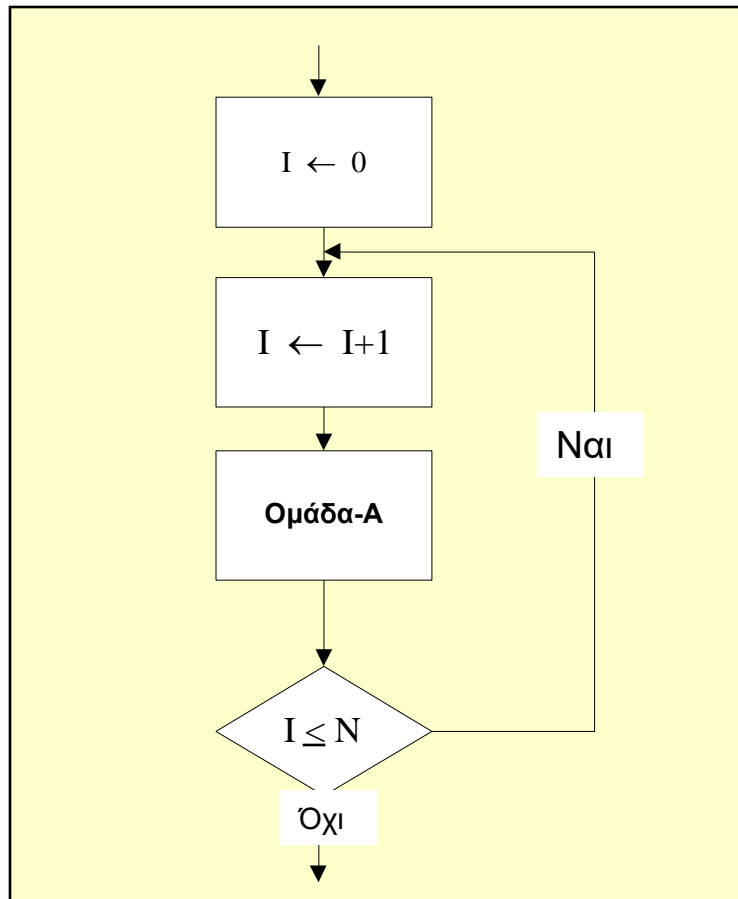
- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Στη WHILE ο αριθμός των επαναλήψεων εξαρτάται από κάποια συνθήκη.



```
I:=1;  
ΕΝΟΣΩ ( I <= N ) ΕΠΑΝΕΛΑΒΕ  
    Ομάδα-A;  
    I := I + 1;  
ΕΝΟΣΩ-ΤΕΛΟΣ;
```

Επαναληπτική Δομή (do-while)

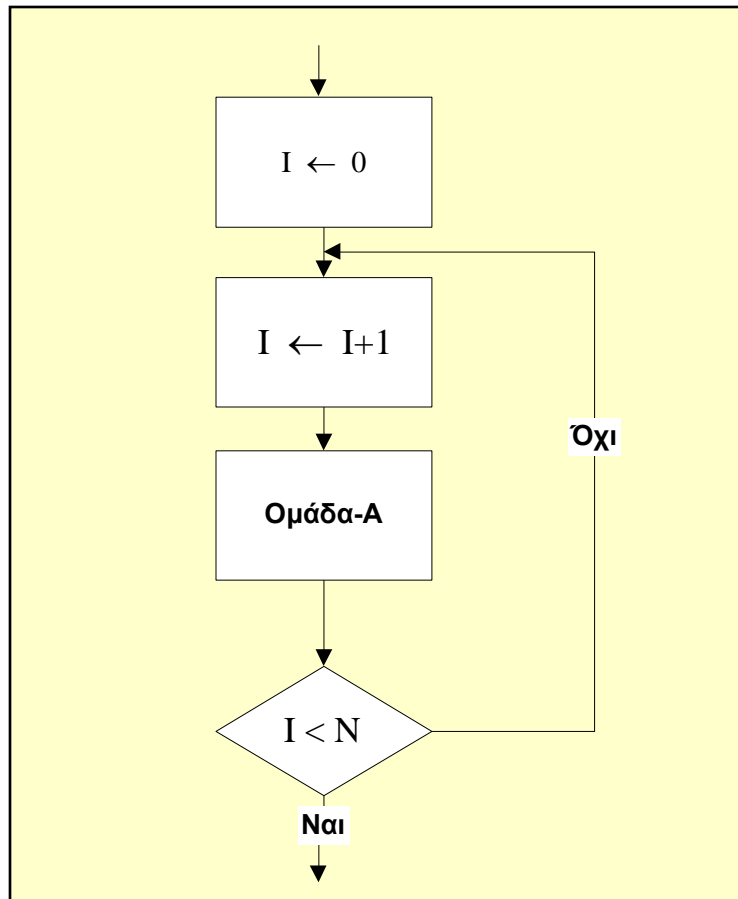
- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Η επισυναπτόμενη ομάδα εντολών εκτελείται **πάντοτε** μία τουλάχιστον φορά.



```
I = 0;  
do  
{  
    I = I + 1;  
    Ομάδα-A;  
}  
while (I ≤ N);
```

Επαναληπτική Δομή (do-while)

- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Η επισυναπτόμενη ομάδα εντολών εκτελείται **πάντοτε** μία τουλάχιστον φορά.



$I:=0;$

ΕΠΑΝΕΛΑΒΕ

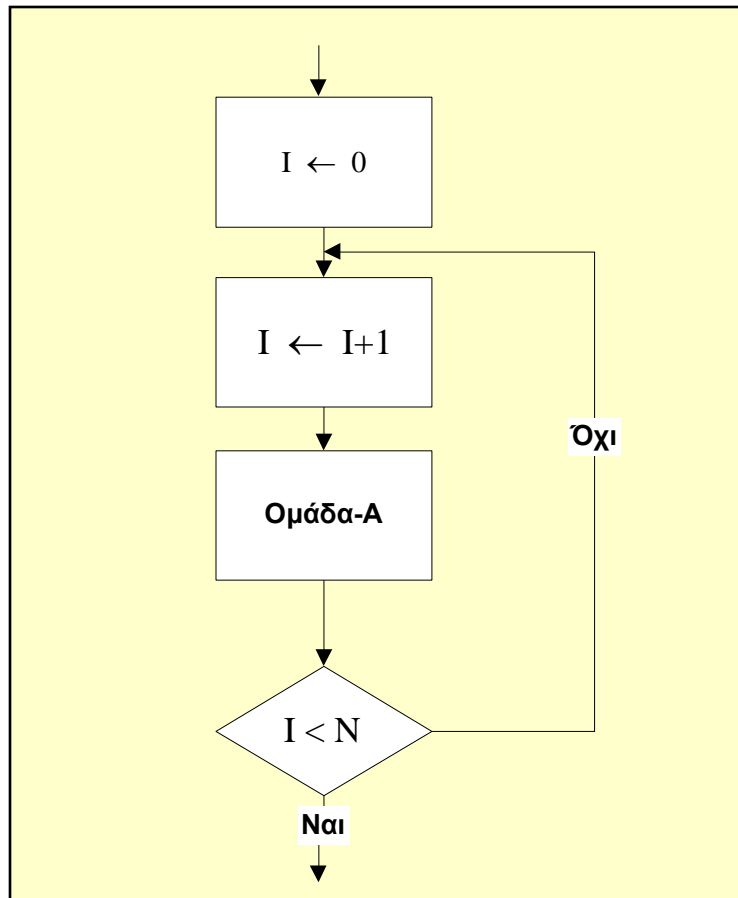
$I:= I + 1;$

Ομάδα-A;

ΕΝΟΣΩ ($I < N$);

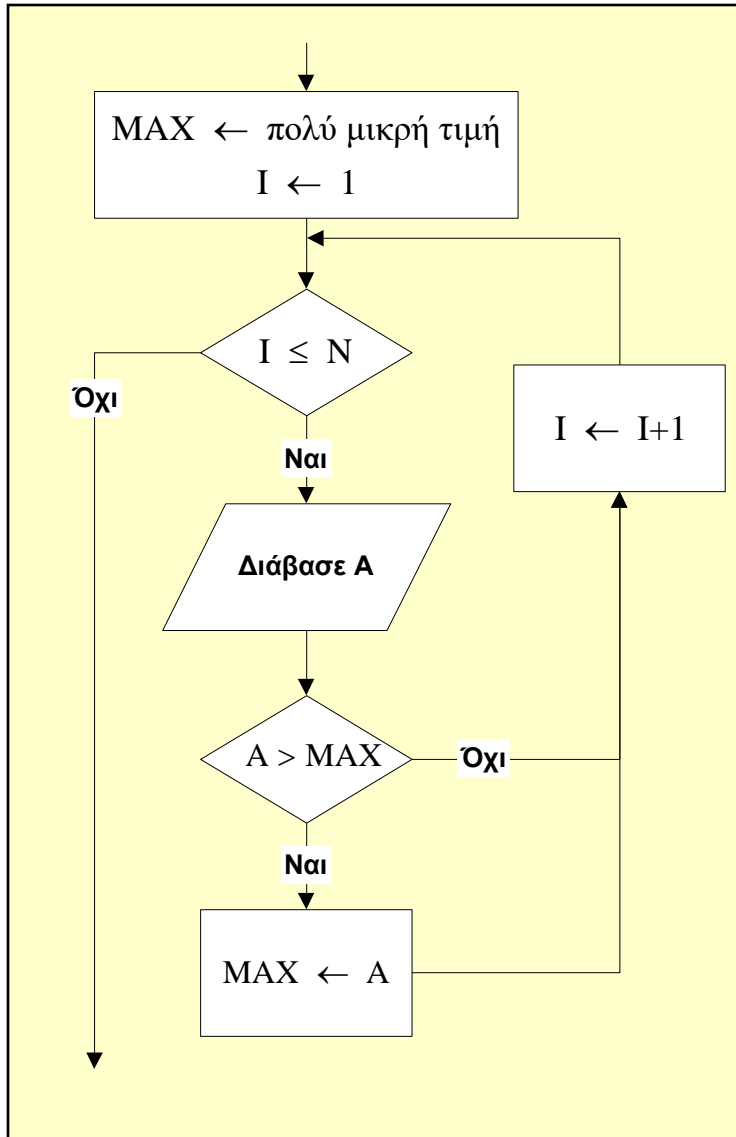
Επαναληπτική Δομή (REPEAT UNTIL)

- Χρησιμοποιείται όταν θέλουμε να εκτελέσουμε μία ομάδα εντολών περισσότερες από μία φορές. Στη REPEAT UNTIL ο αριθμός των επαναλήψεων εξαρτάται από κάποια συνθήκη, αλλά η επισυναπτόμενη ομάδα εντολών εκτελείται **πάντοτε** μία τουλάχιστον φορά.



```
I:=0;  
ΕΠΑΝΕΛΑΒΕ  
    I:= I + 1;  
    Ομάδα-A;  
ΜΕΧΡΙ ( I < N );
```

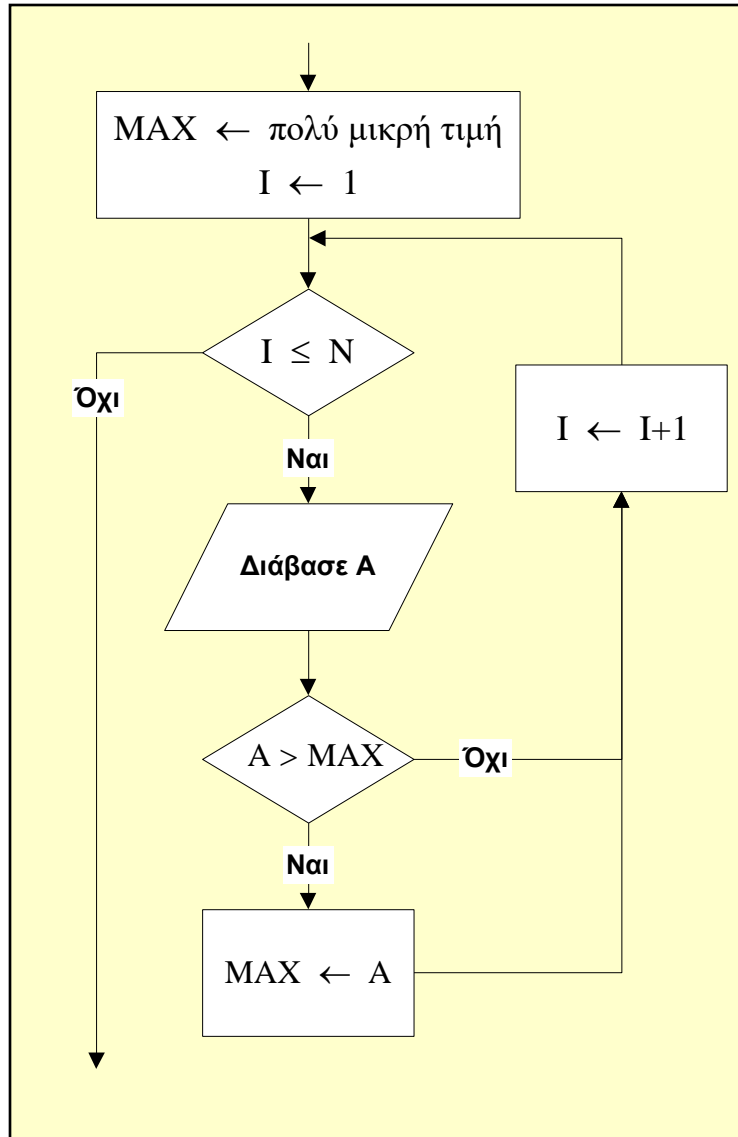
Αλγόριθμος εύρεσης Μέγιστου N αριθμών



```

MAX = -1000000;
I = 1;
while (I <= N)
{
    cin >> A;
    if (A > MAX)
    {
        MAX = A;
    }
    I = I + 1;
}
  
```

Αλγόριθμος εύρεσης Μέγιστου N αριθμών



MAX := -1000000000;

I := 1;

ΕΝΟΣΩ (I ≤ N) ΕΠΑΝΕΛΑΒΕ

ΔΙΑΒΑΣΕ (A);

ΕΑΝ (A > MAX) ΤΟΤΕ

MAX := A;

ΕΑΝ-ΤΕΛΟΣ;

I := I + 1;

ΕΝΟΣΩ-ΤΕΛΟΣ;