242 - Εισαγωγή στους Η/Υ

Τμήμα Μαθηματικών,
Πανεπιστήμιο Ιωαννίνων

Ακαδημαϊκό Έτος 2015-2016

Άρτια Α.Μ. (0-2-4-6-8)

Αντιμετώπιση ενός προβλήματος

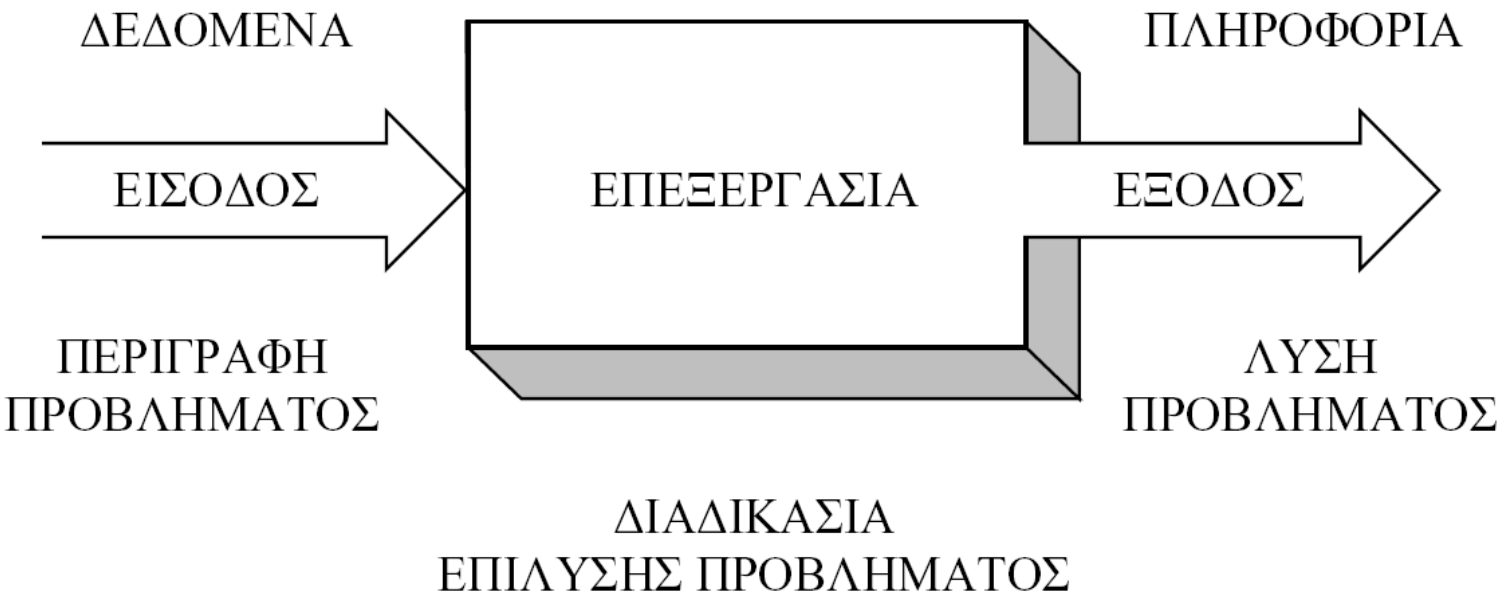
1. Εντοπισμός / προσδιορισμός του προβλήματος
2. Αναλυτική διατύπωση του προβλήματος
3. Ανάλυση και σχεδίαση της επίλυσης του προβλήματος
4. Εφαρμογή της διαδικασίας επίλυσης
5. Επαλήθευση ότι το πρόβλημα έχει αντιμετωπιστεί επιτυχώς
6. Καταγραφή εμπειρίας για μελλοντική χρήση

Αλγόριθμοι – Προγράμματα

- Η περιγραφή της διαδικασίας επίλυσης ενός προβλήματος καλείται **αλγόριθμος**.
- Ο μετασχηματισμός του αλγορίθμου σε μορφή κατανοητή από έναν υπολογιστή καλείται **προγραμματισμός**.
- Το αποτέλεσμα του προγραμματισμού είναι η **ανάπτυξη προγραμμάτων**, τα οποία αποτελούν το **λογισμικό (software)** των υπολογιστών.
- Ένα πρόγραμμα αποτελείται από ένα σύνολο **εντολών (commands)**, οι οποίες αποτελούν «οδηγίες» προς τον υπολογιστή για την εκτέλεση συγκεκριμένων ενεργειών.

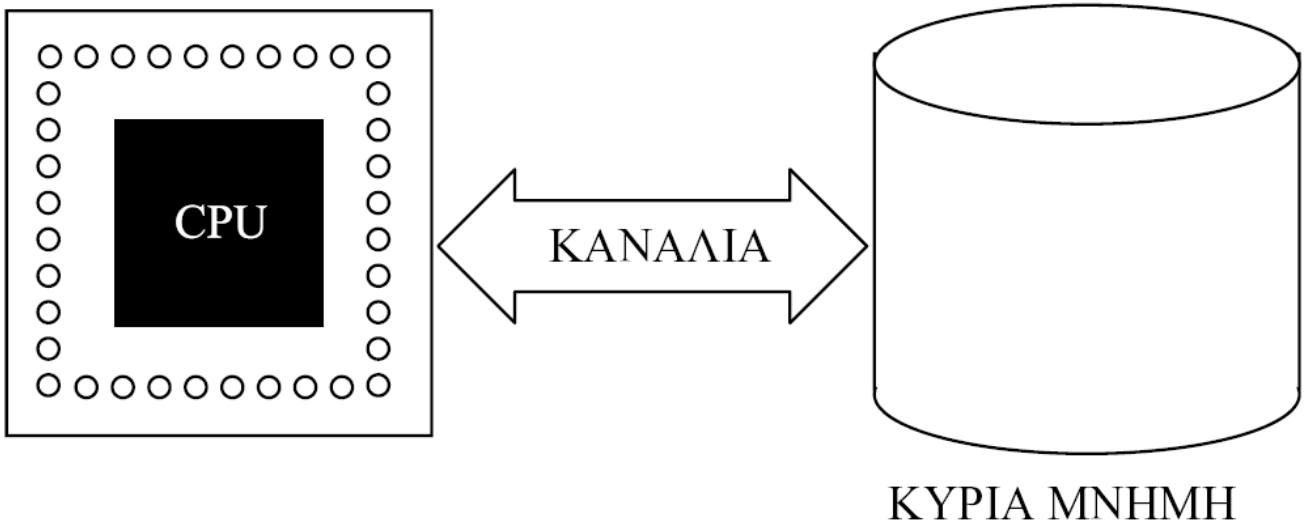
Σύστημα επεξεργασίας δεδομένων

- Στις **εισόδους (inputs)**, οι χρήστες του υπολογιστή του παρέχουν τις πληροφορίες που χρειάζεται (πρόκειται για τα δεδομένα του προβλήματος).
- Στις **εξόδους (outputs)**, ο υπολογιστής παράγει την απάντηση στο πρόβλημα που του τέθηκε.
- Μέσα στο κουτί συντελείται η **επεξεργασία (processing)** των δεδομένων ώστε να παραχθεί η πληροφορία εξόδου.



Τα συστατικά μέρη επεξεργασίας

- η κεντρική μονάδα επεξεργασίας (central processing unit - CPU) ή επεξεργαστή (processor)
- την κύρια μνήμη (main memory),
- τα κανάλια διακίνησης πληροφοριών (buses),



Πρόβλημα και Αλγόριθμος Επίλυσης

- Η επίλυση κάθε προβλήματος απαιτεί την εκτέλεση μιας σειράς ενεργειών, οι οποίες δίνουν στο τέλος ένα συγκεκριμένο αποτέλεσμα.

- Η επιτυχής επίλυση του προβλήματος δεν εξαρτάται μόνο από την εφαρμογή σωστών βημάτων στη σωστή σειρά, αλλά και
- από την εκμετάλλευση όλων των πληροφοριών που μας έχουν δοθεί για το πρόβλημα.

Παράδειγμα Μ.Κ.Δ.

- Πρόβλημα εύρεσης του Μέγιστου Κοινού Διαιρέτη (ΜΚΔ) δύο θετικών ακεραίων Χ και Ψ

- Βήμα 1. ΕΣΤΩ ζ = ΥΠΟΛΟΙΠΟ (χ/ψ)
- Βήμα 2. ΕΑΝ ζ = 0, ΤΟΤΕ
- Βήμα 2.1 ΜΚΔ = ψ. ΤΕΡΜΑΤΙΣΜΟΣ
- Βήμα 3. ΑΛΛΙΩΣ ΕΠΑΝΑΛΗΨΗ ΑΠΟ βήμα 1 ΜΕ (χ = ψ ΚΑΙ ψ = ζ)

Παράδειγμα Αυτόματης Συναλλαγής

- Βήμα 1. ΕΙΣΑΓΩΓΗ κάρτας συναλλαγών
- Βήμα 2. ΕΙΣΑΓΩΓΗ κωδικού χρήστη
- Βήμα 3. ΕΠΙΛΟΓΗ συναλλαγής ΑΠΟ σύνολο συναλλαγών
- Βήμα 4. ΕΑΝ συναλλαγή = ανάληψη ΤΟΤΕ
- Βήμα 4.1 ΕΙΣΑΓΩΓΗ ποσού ανάληψης
- Βήμα 4.2 ΠΑΡΑΛΑΒΗ χρημάτων
- Βήμα 4.3 ΠΑΡΑΛΑΒΗ απόδειξης
- Βήμα 5. ΑΛΛΙΩΣ ΕΑΝ συναλλαγή= κατάθεση ΤΟΤΕ
- Βήμα 5.1 ΕΙΣΑΓΩΓΗ ποσού κατάθεσης
- Βήμα 5.2 ΑΠΟΘΕΣΗ χρημάτων
- Βήμα 5.3 ΠΑΡΑΛΑΒΗ απόδειξης
- Βήμα 6. ΠΑΡΑΛΑΒΗ κάρτας συναλλαγών. ΤΕΡΜΑΤΙΣΜΟΣ

Χαρακτηριστικά Αλγορίθμων

- **Είναι πεπερασμένος**. αποτελείται από πεπερασμένο αριθμό βημάτων, και τερματίζει μετά από ένα πεπερασμένο αριθμό βημάτων.

- **Είναι καλά ορισμένος**. οι ενέργειες που θα εκτελεστούν σε αυτό πρέπει να περιγράφονται ρητά και αδιαμφισβήτητα.

- **Είσοδος**. Κάθε αλγόριθμος δέχεται μία ή περισσότερες εισόδους.

- **Εξοδος**. Κάθε αλγόριθμος παράγει μία ή περισσότερες εξόδους.

- **Αποτελεσματικότητα**. όλες οι ενέργειες που περιλαμβάνει να είναι αρκετά απλές ώστε να μπορούν κατ. αρχήν να εκτελεστούν με ακρίβεια σε πεπερασμένο χρόνο από έναν άνθρωπο (με χαρτί και μολύβι).

Τμήματα του Αλγόριθμου

- την **κεφαλή (header)** του αλγόριθμου,
  - η οποία περιλαμβάνει το όνομα του αλγόριθμου και τον κατάλογο των δεδομένων εισόδου/ εξόδου.
- το **τμήμα δηλώσεων των δεδομένων** με τα οποία εκτελούνται οι πράξεις που περιγράφει ο αλγόριθμος.
- το **κυρίως τμήμα του αλγορίθμου (body)**, στο οποίο περιγράφονται οι πράξεις που εκτελούνται πάνω στα δεδομένα.

Τμήματα Προγράμματος

<ΑΡΧΗ>

Δήλωση μεταβλητών

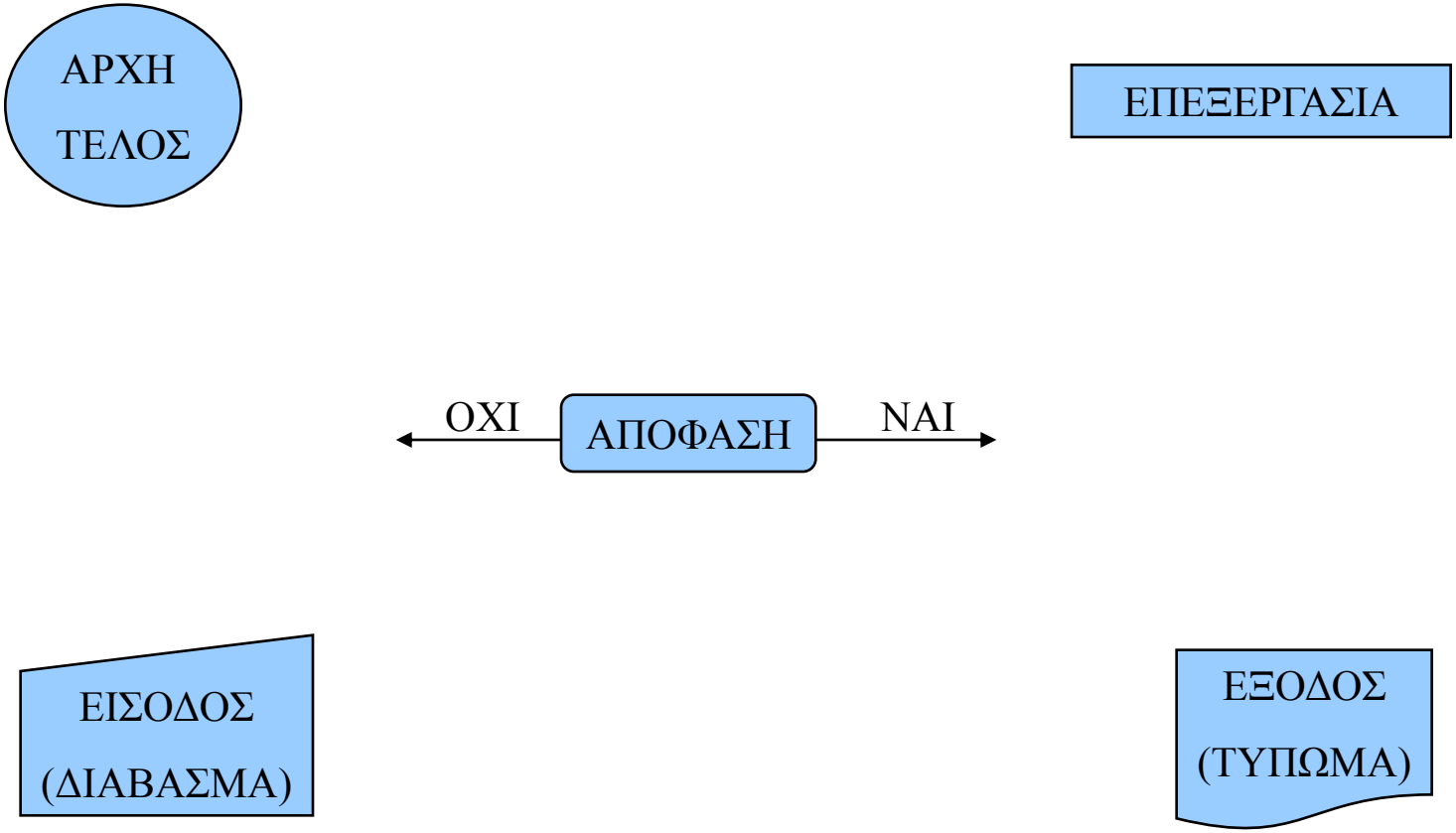
Διάβασμα

Επεξεργασία

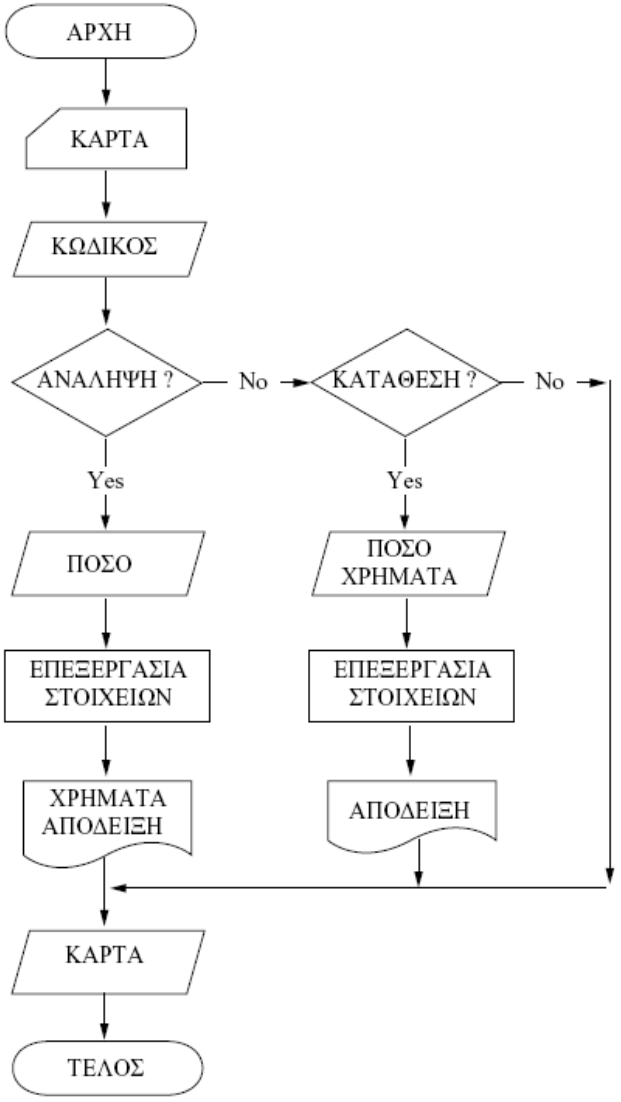
Τύπωση

<ΤΕΛΟΣ>

Διάγραμμα Ροής Προγράμματος



ΔΡΠ για ΑΤΜ



ΑΡΧΗ

ΔΙΑΒΑΣΕ (ΚΑΡΤΑ) ΑΠΟ ΣΧΙΣΜΗ-ΚΑΡΤΑΣ

ΔΙΑΒΑΣΕ (ΚΩΔΙΚΟΣ) ΑΠΟ ΚΑΡΤΑ

ΔΙΑΒΑΣΕ (ΣΥΝΑΛΛΑΓΗ) ΑΠΟ ΜΕΝΟΥ

ΕΑΝ (ΣΥΝΑΛΛΑΓΗ=ανάληψη) ΤΟΤΕ

    ΔΙΑΒΑΣΕ (ΠΟΣΟ) ΑΠΟ ΜΕΝΟΥ

    ΥΠΟΛΟΓΙΣΕ ΕΠΕΞΕΡΓΑΣΙΑ-ΣΤΟΙΧΕΙΩΝ

    ΓΡΑΨΕ (ΧΡΗΜΑΤΑ) ΣΕ ΣΧΙΣΜΗ-ΧΡΗΜΑΤΩΝ

    ΓΡΑΨΕ (ΑΠΟΔΕΙΞΗ) ΣΕ ΣΧΙΣΜΗ- ΑΠΟΔΕΙΞΗΣ

ΑΛΛΙΩΣ

    ΕΑΝ (ΣΥΝΑΛΛΑΓΗ=κατάθεση) ΤΟΤΕ

        ΔΙΑΒΑΣΕ (ΠΟΣΟ, ΧΡΗΜΑΤΑ) ΑΠΟ ΜΕΝΟΥ

        ΥΠΟΛΟΓΙΣΕ ΕΠΕΞΕΡΓΑΣΙΑ-ΣΤΟΙΧΕΙΩΝ

        ΓΡΑΨΕ (ΑΠΟΔΕΙΞΗ) ΣΕ ΣΧΙΣΜΗ–ΑΠΟΔΕΙΞΗΣ

    ΕΑΝ-ΤΕΛΟΣ

ΕΑΝ-ΤΕΛΟΣ

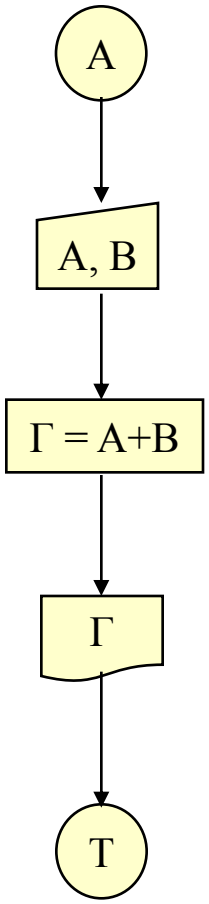
ΓΡΑΨΕ (ΚΑΡΤΑ) ΣΕ ΣΧΙΣΜΗ–ΚΑΡΤΑΣ

ΤΕΛΟΣ

Δομές ελέγχου ροής προγράμματος στον Δομημένο προγραμματισμό

- Ανάθεση τιμής σε μεταβλητή.
- Εντολές επιλογής:
  - If (συνθήκη) {εντολές;}
  - If (συνθήκη) {εντολές1} else {εντολές2;}
  - Switch …..
- Εντολές άνακύκλωσης
- For (αρχικοπ, συνθήκη τερμ, αύξηση μεταβλ. Ελέγχου) {εντολές;}
- While (συνθήκη) {εντολές;}
- Do {εντολές;} while (συνθήκη)

Παράδειγμα: άθροισμα δύο αριθμών



διαδικασια ΑΘΡΟΙΣΜΑ_1
δηλωση (Α, Β, Γ) **ακερ**;
αρχη
    διαβασε Α, Β;
    Γ ← Α+Β;
    γραψε Γ;
τελος ΑΘΡΟΙΣΜΑ_1

```
// όνομα αρχείου Ath.cpp
#include <iostream>
main()
{
   int A, B, C;
   cin >> A >> B;
   C = A + B;
   cout << C;
}
```

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 την παύλα (‘-’) και την κάτω παύλα (‘_’), αλλά αρχίζει πάντα από γράμμα. Π.χ. X, MAX2, MAX-2.
- ΕΑΓ: πάντα με ΚΕΦΑΛΑΙΑ
- C++: δεν παίζει ρόλο (καλύτερα με κεφαλαία)

Τύπος Μεταβλητής ΕΑΓ και C++

- Τα μεγέθη διαφέρουν ως προς τις ιδιότητες και το είδος των τιμών που παίρνουν.
- Υπάρχουν τρεις βασικοί (ή πρωτογενείς) τύποι μεταβλητών:

Επεξήγηση	ΕΑΓ	C++	π.χ.
ακέραιος	**ακερ**	`int`	5, 18
πραγματικός	**πραγματικ**	`float`	3.1, 24.6
χαρακτήρας	**χαρακτηρ**	`char`	“A”, “7”

Δηλώσεις μεταβλητών

Η δήλωση μιας μεταβλητής γίνεται ως εξής:

ΕΑΓ: **δηλωση** (<λίστα μεταβλητών>) <τύπος> ;

Π.χ. **δηλωση** (Α, Β, Γ) **ακερ** ;
δηλωση (Δ) **πραγματικ** ;
δηλωση (Τ) **χαρακτηρ** ;

C++: <τύπος> <λίστα μεταβλητών> ;

Π.χ.
```
int A, B, C;
float D;
char T;
```

Είσοδος/Έξοδος C++

- Είσοδος/Εξοδος: τερματικό παράθυρο
- Η είσοδος (διάβασμα δεδομένων) θα θεωρήσουμε ότι γίνεται από το πληκτρολόγιο
- Διάβασμα: `cin >> x;`
  - Περιμένει μια πληκτρολόγηση δεδομένων
  - εκτελείται η εντολή όταν πατηθεί «RETURN» (enter) και τότε αναθέτει στην x τα δεδομένα
- Η έξοδος (εκτύπωση δεδομένων) θα θεωρήσουμε ότι γίνεται στο τερματικό παράθυρο
- Εκτύπωση:

```
cout << "Μάθημα 1";
cout << x;
cout << "Η x έχει τιμή" << x;
```

Είσοδος/Έξοδος ΕΑΓ

- ΔΙΑΒΑΣΕ
- ΤΥΠΩΣΕ

Παράδειγμα: Φαρενάιτ (F) σε βαθμούς Κελσίου (C)

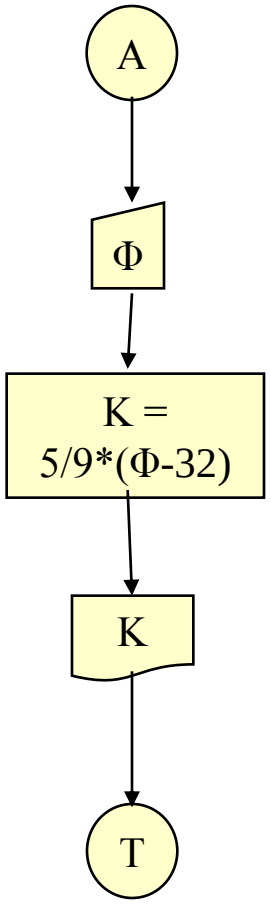
- Να περιγραφεί αλγόριθμος που μετατρέπει βαθμούς Φαρενάιτ (F) σε βαθμούς Κελσίου (C) χρησιμοποιώντας την εξίσωση:

$$C=5/9 * (F-32)$$

- Σκεφτείτε πρώτα ΕΙΣΟΔΟΣ - ΕΞΟΔΟΣ

- Στη συνέχεια διαχωρίστε τις μεταβλητές
  - Τύπος κάθε μεταβλητής
  - Δήλωση κάθε μεταβλητής

Παράδειγμα: Φαρενάιτ (F) σε βαθμούς Κελσίου (C)



διαδικασια ΦΑΡ_ΚΕΛ
δηλωση (Φ, Κ) **πραγματικ**;
αρχη
    διαβασε Φ;
    Κ ← (5/9) * (Φ-32);
    τυπωσε Κ;
τελος ΦΑΡ_ΚΕΛ

```
// όνομα αρχείου F-C.cpp
#include <iostream>
main()
{
   float F, C;
   cin >> F;
   C = (5/9) * (F - 32);
   cout << C;
}
```

Τελεστές

- Είναι σύμβολα που παριστάνουν κάποια στοιχειώδη λειτουργία/επεξεργασία.

  - Αριθμητικοί τελεστές:

    + - * / % (υπόλοιπο διαίρεσης)

    - Γράφονται μεταξύ των στοιχείων που συνδυάζουν
    - Π.χ. A+B, ((K + 3)/4) * L
    - C++: Μπορεί να γραφτούν και πριν ή μετά τις μεταβλητές
    - Π.χ. I++, ++I, K--

  - Τελεστές σύγκρισης ή Συσχετιστικοί τελεστές:

    >=, <=, <, >, !=, ==

  - Λογικοί τελεστές: **|| (η) , && (και)**

Εκφράσεις

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

Αριθμητικές: Π.χ. (5 * X + Y),

((K + 3)/4) * L, K % 2.

Σύγκρισης: Π.χ. X >= Y, A = B, K < L και X != Y

είναι εκφράσεις σύγκρισης. Η τιμή που επιστρέφει μια έκφραση σύγκρισης είναι τύπου BOOLEAN (0/1 δηλαδή ψευδές / αληθές).

Λογικές: Π.χ. (X > Y) && (X < 10),

(A == B) || (A >= 5),

είναι λογικές εκφράσεις. Η τιμή που επιστρέφει μια λογική έκφραση είναι επίσης τύπου BOOLEAN

Τελεστές & Εκφράσεις

```
if (A > B) && (B > C)
    cout << A << B << C;
else
    τι τιμή έχει εδώ η B?
```

⇐ $B \geq A$ ή $B \leq C$

```
if (A > B) || (B > C)
    cout << A << B << C;
else
    τι τιμή έχει εδώ η B?
```

⇐ $B \geq A$ και $B \leq C$

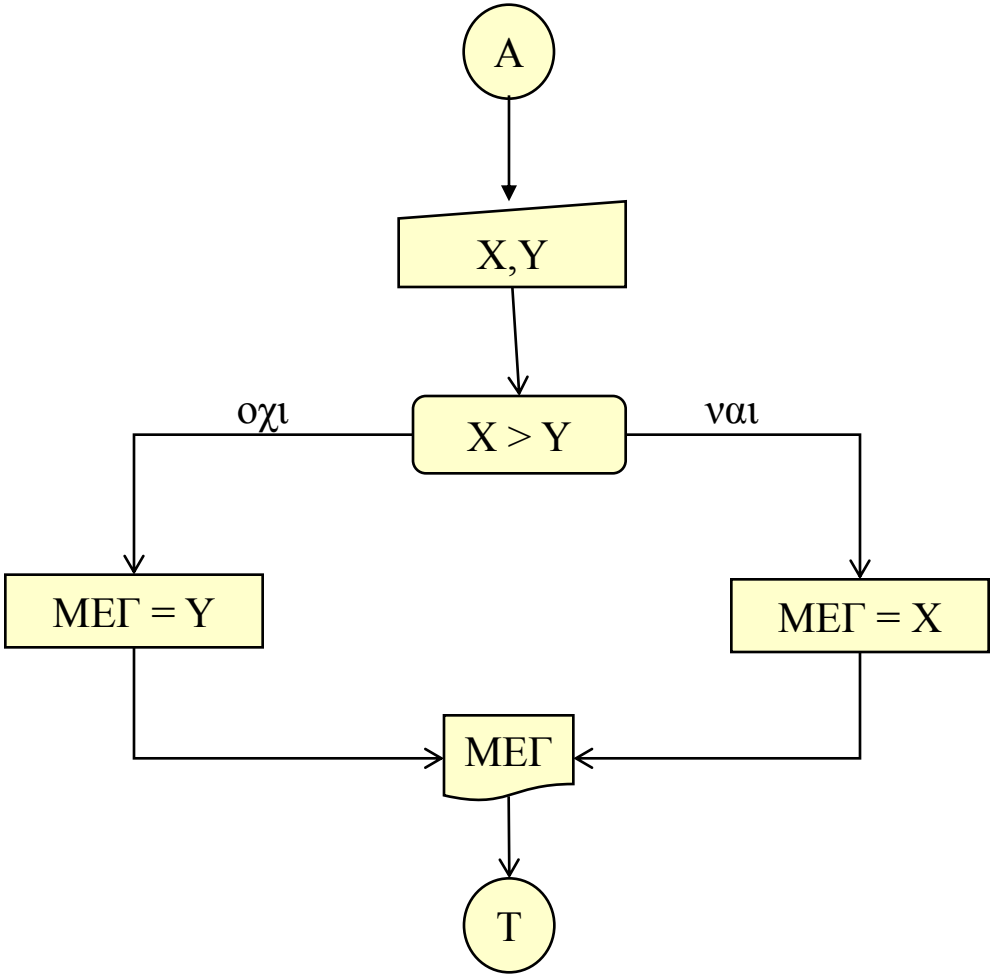
δηλ. $C \geq B \geq A$

Παράδειγμα: MAX-XY

```
διαδικασια ΕΥΡΜΕΓ1
δηλωση (Χ, Υ, ΜΕΓ) ακερ;
αρχη
    διαβασε Χ, Υ;
    εαν  Χ > Υ
          τοτε      ΜΕΓ ← Χ;
          αλλως     ΜΕΓ ← Υ;
    τυπωσε ΜΕΓ;
τελος ΕΥΡΜΕΓ1
```

```
// όνομα αρχείου MAX1.cpp
#include <iostream>
main()
{
   int X, Y, MAX;
   cin >> X >> Y;
   if (X > Y)
       MAX = X;
   else
       MAX = Y;
   cout << MAX;
}
```

Παράδειγμα ΔΡΠ : ΜΑΧ-ΧΥ



Σώμα εντολών

- Κάθε ομάδα εντολών πρέπει να διαχωρίζεται με κάποιο τρόπο.
- ΕΑΓ:
  - Οι αρχικές εντολές (πρώτου επιπέδου) βρίσκονται μεταξύ **αρχη** και **τελος** .
  - Όλες οι άλλες εντολές περικλείονται μεταξύ (και)
  - Εάν υπάρχει μια μόνο εντολή δεν είναι απαραίτητη η χρήση ( )
- C++:
  - Όλες οι ομάδες εντολών περικλείονται μεταξύ { και }
  - Εάν υπάρχει μια μόνο εντολή δεν είναι απαραίτητη η χρήση { }

- Για κάθε αριστερή (ή { υπάρχει ΠΑΝΤΑ και μια δεξιά) ή }

Παράδειγμα: MAX-XYZ

```
διαδικασια ΕΥΡΜΕΓ2
δηλωση (Χ, Υ, Ζ, ΜΕΓ) ακερ;
αρχη
    διαβασε Χ, Υ, Ζ;
    εαν  Χ > Υ
        τοτε
                εαν Χ > Ζ
                        τοτε    ΜΕΓ ← Χ;
                        αλλως   ΜΕΓ ← Ζ;
        αλλως
                εαν Υ > Ζ
                        τοτε    ΜΕΓ ← Υ;
                        αλλως   ΜΕΓ ← Ζ;
    τυπωσε ΜΕΓ;
τελος ΕΥΡΜΕΓ2
```

```
// όνομα αρχείου MAX2.cpp
#include <iostream>
main()
{
   int X, Y, Z MAX;
   cin >> X >> Y >> Z;
   if (X > Y)
        if (X > Z)
                MAX = X;
        else
                MAX = Z;
   else
        if (Y > Z)
                MAX = Y;
        else
                MAX = Z;
   cout << MAX;
}
```

Παράδειγμα: MAX-XYZ

```
// όνομα αρχείου MAX2.cpp
#include <iostream>
main()
{
   int X, Y, Z MAX;
   cin >> X >> Y >> Z;
   if (X > Y)
        if (X > Z)
                MAX = X;
        else
                MAX = Z;
   else
        if (Y > Z)
                MAX = Y;
        else
                MAX = Z;
   cout << MAX;
}
```

είναι το ίδιο

Θα προτιμάτε:

```
// όνομα αρχείου MAX2.cpp
#include <iostream>
main()
{
   int X, Y, Z MAX;
   cin >> X >> Y >> Z;
   if (X > Y)
   {
        if (X > Z)
                MAX = X;
        else
                MAX = Z;
   }
   else
   {
        if (Y > Z)
                MAX = Y;
        else
                MAX = Z;
   }
   cout << MAX;
}
```

Δεσμευμένες λέξεις C++ 1/2

Επεξήγηση	C++
αρχή διαδικασίας	`main()`
τέλος διαδικασίας	
αρχή ομάδας οδηγιών	{
τέλος ομάδας οδηγιών	}
Καταχώρηση σε μεταβλητή	=
Διαχωριστής εντολών	;
Είσοδος δεδομένων	`cin >> …`
Εξοδος δεδομένων	`cout << …`
Σχόλια	// ή /*....*/

```
// όνομα αρχείου Ath.cpp

#include <iostream>

main()
{
   int A, B, C;

   cin >> A >> B;

   C = A + B;

   cout << C;
}
```

Δεσμευμένες λέξεις C++ 2/2

Επεξήγηση	C++

Απόφαση	`if(...)`
	`{…}`
	`[else]`
	`{…}`

Επανάληψη με μετρητή	`for(..;..;..)` `{…}`
Επανάληψη με συνθήκη στην αρχή	`while(..)` `{…}`

Επανάληψη με συνθήκη στο τέλος	`do`
	`{…}`
	`while(..);`

```
// όνομα αρχείου MAX1.cpp
#include <iostream>
main()
{
   int X, Y, MAX;
   cin >> X >> Y;
   if (X > Y)
   {
        MAX = X;
   }
   else
   {
        MAX = Y;
   }
   cout << MAX;
}
```

Δεσμευμένες λέξεις ΕΑΓ και C++ 1/3

- ΕΑΓ: άτονα, ελληνικά, μικρά έντονα (υπογραμμισμένα) γράμματα
- C++: αγγλικά

Επεξήγηση	ΕΑΓ	C++
αρχή διαδικασίας	**διαδικασια** ΟΝΟΜΑ	`main()`
τέλος διαδικασίας	**τελος** ΟΝΟΜΑ	
αρχή ομάδας οδηγιών	**(**	{
τέλος ομάδας οδηγιών	**)**	}
Καταχώρηση σε μεταβλητή	←	=
Διαχωριστής εντολών	;	**;**

Δεσμευμένες λέξεις ΕΑΓ και C++ 2/3

- ΕΑΓ: άτονα, ελληνικά, μικρά έντονα (υπογραμμισμένα) γράμματα
- C++: αγγλικά

Επεξήγηση	ΕΑΓ	C++
Είσοδος δεδομένων	**διαβασε**	`cin >> ...`
Εξοδος δεδομένων	**τυπωσε** ή **γράψε**....	`cout << ...`
Σχόλια	// ή /*....*/	// ή /*....*/

Δεσμευμένες λέξεις ΕΑΓ και C++ 3/3

- ΕΑΓ: άτονα, ελληνικά, μικρά έντονα (υπογραμμισμένα) γράμματα
- C++: αγγλικά

Επεξήγηση	ΕΑΓ	C++
Απόφαση	**εαν**... **τοτε**... (**αλλως**)	`if...(else)`
Επανάληψη με μετρητή	**για** .. **εως** .. **επαναλαβε**	`for(..;..;)`
Επανάληψη με συνθήκη στην αρχή	**εφοσον** .. **επαναλαβε**	`while(..)`
Επανάληψη με συνθήκη στο τέλος	**επαναλαβε** ..... **εως** ...;	`do …` `while(..);`

Παράδειγμα: ΜΚΔ

- Θέλουμε να βρούμε τον ΜΚΔ μεταξύ δύο ακεραίων (Χ και Ψ).
- Σύμφωνα με τον αλγόριθμο του Ευκλείδη:
  1. αν οι δύο αριθμοί είναι ίσοι, ο μέγιστος κοινός διαιρέτης ισούται μ' αυτούς (Χ = Ψ).
  2. αν δεν είναι ίσοι, αντικαθιστούμε τον μεγαλύτερο με τη διαφορά τους (Χ-Ψ ή Ψ-Χ) και επαναλαμβάνουμε τη σύγκριση

Έστω Χ=36, Ψ=84:

36 84 Δεν είναι ίσοι. Το 84 αντικαθίσταται με τη διαφορά 84-36=48
36 48 Δεν είναι ίσοι. Το 48 αντικαθίσταται με τη διαφορά 48-36=12
36 12 Δεν είναι ίσοι. Το 36 αντικαθίσταται με τη διαφορά 36-12=24
24 12 Δεν είναι ίσοι. Το 24 αντικαθίσταται με τη διαφορά 24-12=12
12 12 Είναι ίσοι. Ο μέγιστος κοινός διαιρέτης των 36 και 84 είναι το 12.

Κατά βήμα εκλέπτυνση

- Διάσπαση των σχεδιαστικών αποφάσεων σε στοιχειώδη επίπεδα
- Απομόνωση σχεδιαστικών τμημάτων που στην πραγματικότητα δεν εξαρτώνται από άλλα.
- Αναβολή όσο το δυνατόν περισσότερο των αποφάσεων αναπαράστασης.
- Προσεκτική απόδειξη ότι κάθε διαδοχικό βήμα εκλέπτυνσης είναι πιστή επέκταση προηγούμενων βημάτων.
- Βελτίωση του προκύπτοντος αλγορίθμου.

Παράδειγμα: ΜΚΔ

- Θα το αναλύσουμε σε C++:

```
// όνομα αρχείου MKD1.cpp
#include <iostream>
main()
{
   int A, B, MKD;
   cin >> A >> B;
   while ( A != B)
   {
        MAX(A,B) = A-B ή
        MAX(A,B) = B-A
   }
   MKD = A;

   cout << MKD;
}
```

```
// όνομα αρχείου MKD1.cpp
#include <iostream>
main()
{
   int A, B, MKD;
   cin >> A >> B;
   while( A != B)
   {
        if( A > B )
                A = A-B;
        else
                B = B-A;
   }
   MKD = A; // ή MKD = B;
   cout << MKD;
}
```

Παράδειγμα: ΜΚΔ2

```
while( A != B)
{
     if( A > B )
            A = A-B;
     else
            B = B-A;
}
```

- Επεκτείνετε το πρόγραμμα έτσι ώστε να εκτυπώνετε και το πλήθος των επαναλήψεων που χρειάστηκαν μέχρι να βρούμε το μέγιστο κοινό διαιρέτη.

Παράδειγμα: ΜΚΔ2

```
// όνομα αρχείου MKD2.cpp
#include <iostream>
main()
{
   int A, B, MKD, EPAN;
   cin >> A >> B;
   EPAN = 0;
   while( A != B)
   {
        if( A > B )
                A = A-B;
        else
                B = B-A;
        EPAN = EPAN + 1;
   }
   MKD = A;
   cout << MKD << EPAN;
}
```

διαδικασια ΜΚΔ2
δηλωση (Α, Β, ΕΠΑΝ, ΜΚΔ) **ακερ**;
αρχη
    διαβασε Α, Β;
    ΕΠΑΝ ← 0;
    εφοσον Α ≠ Β **επαναλαβε**
    (
        εαν Α > Β
            τοτε Α ← Α – Β;
            αλλως Β ← Β - Α;
        ΕΠΑΝ ← ΕΠΑΝ+1;
)
    ΜΚΔ ← Α;
    τυπωσε ΜΚΔ, ΕΠΑΝ;
τελος ΜΚΔ2

Μέγιστο, Ελάχιστο, Μέσο όρο, από Ν αριθμούς (γνωστό ή άγνωστο πλήθος)

- Ξεχωρίζουμε δύο περιπτώσεις:
  1. Όταν το πλήθος Ν των αριθμών είναι γνωστό
  2. Όταν το πλήθος Ν των αριθμών είναι άγνωστο

- Και στις δύο περιπτώσεις πρέπει να έχουμε ένα μετρητή Ι για την διάσχιση των Ν αριθμών.
  - Ι = 0 (αρχικοποίηση) και
  - Ι = Ι + 1 (αύξηση)

Μέγιστο, Ελάχιστο, Μέσο όρο, από Ν αριθμούς (γνωστό ή άγνωστο πλήθος)

- Στην πρώτη περίπτωση πρέπει να διαβάσουμε το πλήθος Ν των αριθμών.
  - Εκτελούμε Ν φορές συγκεκριμένες εντολές
  - Γνωστή επανάληψη ⇒

    για .. **εως** .. **επαναλαβε** `for(..;..;..)`

- Στην δεύτερη περίπτωση (άγνωστο Ν) ζητάμε έναν «τερματιστή» από τον χρήστη.
  - Όταν ένας από τους αριθμούς που δίνει ο χρήστης είναι ίσος με τον «τερματιστή» τότε σταματάμε την εκτέλεση.
  - Άγνωστη επανάληψη ⇒

    εφοσον .. **επαναλαβε** `while(..)`
  - Όσο ο «τερματιστής» είναι διαφορετικός από τον αριθμό που διάβασες...

Εύρεση Αθροίσματος

- Να περιγραφεί η διαδικασία αθροίσματος, αγνώστου πλήθους, ακεραίων αριθμών, χρησιμοποιώντας
  (i) την ΕΑΓ, (ii) το ΔΡΠ, και (iii) την C++.

- Το άθροισμα μπορούμε να το κωδικοποιήσουμε:
  - ΑΘΡ ← 0; // αρχικοποίηση
    ΑΘΡ ← ΑΘΡ + ΑΡΙΘΜ // όπου ΑΡΙΘΜ είναι ο αριθμός που
    // διαβάζουμε από τον χρήστη

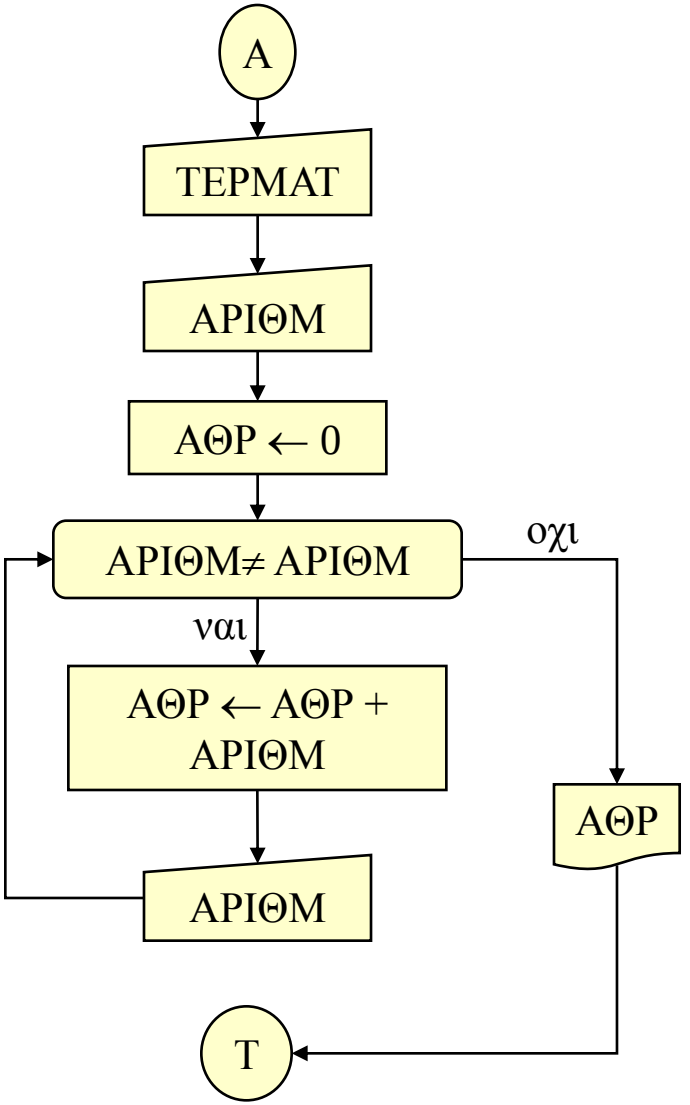
- Θα εκτελέσουμε κατά βήμα εκλέπτυνση..

Εύρεση Αθροίσματος - ΕΑΓ

```
διαδικασια ΑΘΡΟΙΣΜΑ2
δηλωση (ΤΕΡΜΑΤ, ΑΡΙΘΜ, ΑΘΡ) ακερ;
αρχη
    διαβασε ΤΕΡΜΑΤ, ΑΡΙΘΜ;
    ΑΘΡ ← 0;
    εφοσον  ΤΕΡΜΑΤ ≠ ΑΡΙΘΜ  επαναλαβε
    (
            ΑΘΡ ← ΑΘΡ + ΑΡΙΘΜ;
            διαβασε ΑΡΙΘΜ;
    )
    τυπωσε ΑΘΡ;
τελος ΑΘΡΟΙΣΜΑ2
```

Εύρεση Αθροίσματος – ΕΑΓ, ΔΡΠ

```
διαδικασια ΑΘΡΟΙΣΜΑ2
δηλωση (ΤΕΡΜΑΤ, ΑΡΙΘΜ, ΑΘΡ) ακερ;
αρχη
    διαβασε ΤΕΡΜΑΤ, ΑΡΙΘΜ;
    ΑΘΡ ← 0;
    εφοσον  ΤΕΡΜΑΤ ≠ ΑΡΙΘΜ  επαναλαβε
    (
            ΑΘΡ ← ΑΘΡ + ΑΡΙΘΜ;
            διαβασε ΑΡΙΘΜ;
    )
    τυπωσε ΑΘΡ;
τελος ΑΘΡΟΙΣΜΑ2
```



Εύρεση Αθροίσματος – ΕΑΓ, ΔΡΠ, C++

```
διαδικασια ΑΘΡΟΙΣΜΑ2
δηλωση (ΤΕΡΜΑΤ, ΑΡΙΘΜ, ΑΘΡ) ακερ;
αρχη
    διαβασε ΤΕΡΜΑΤ, ΑΡΙΘΜ;
    ΑΘΡ ← 0;
    εφοσον  ΤΕΡΜΑΤ ≠ ΑΡΙΘΜ  επαναλαβε
    (
            ΑΘΡ ← ΑΘΡ + ΑΡΙΘΜ;
            διαβασε ΑΡΙΘΜ;
    )
    τυπωσε ΑΘΡ;
τελος ΑΘΡΟΙΣΜΑ2
```

```
// όνομα αρχείου ATHR2.cpp
#include <iostream>
main()
{
   int TERMAT, ARITHM, ATHR;
   cin >> TERMAT >> ARITHM;
   ATHR = 0;
   while(TERMAT != ARITHM)
   {
        ATHR = ATHR + ARITHM;
        cin >> ARITHM;
   }
   cout << ATHR;
}
```

Εύρεση Μεγίστου

- Να περιγραφεί η διαδικασία εύρεσης μεγίστου, γνωστού πλήθους, ακεραίων αριθμών, χρησιμοποιώντας
  (i) την ΕΑΓ, (ii) το ΔΡΠ, και (iii) την C++

- Για την εύρεση μεγίστου μπορούμε :

```
MAX ← -10000000;          // αρχικοποίηση
    εαν  ΑΡΙΘΜ > MAX      // όπου ΑΡΙΘΜ είναι ο αριθμός που
       τοτε  MAX ← ΑΡΙΘΜ  // διαβάζουμε από τον χρήστη
```

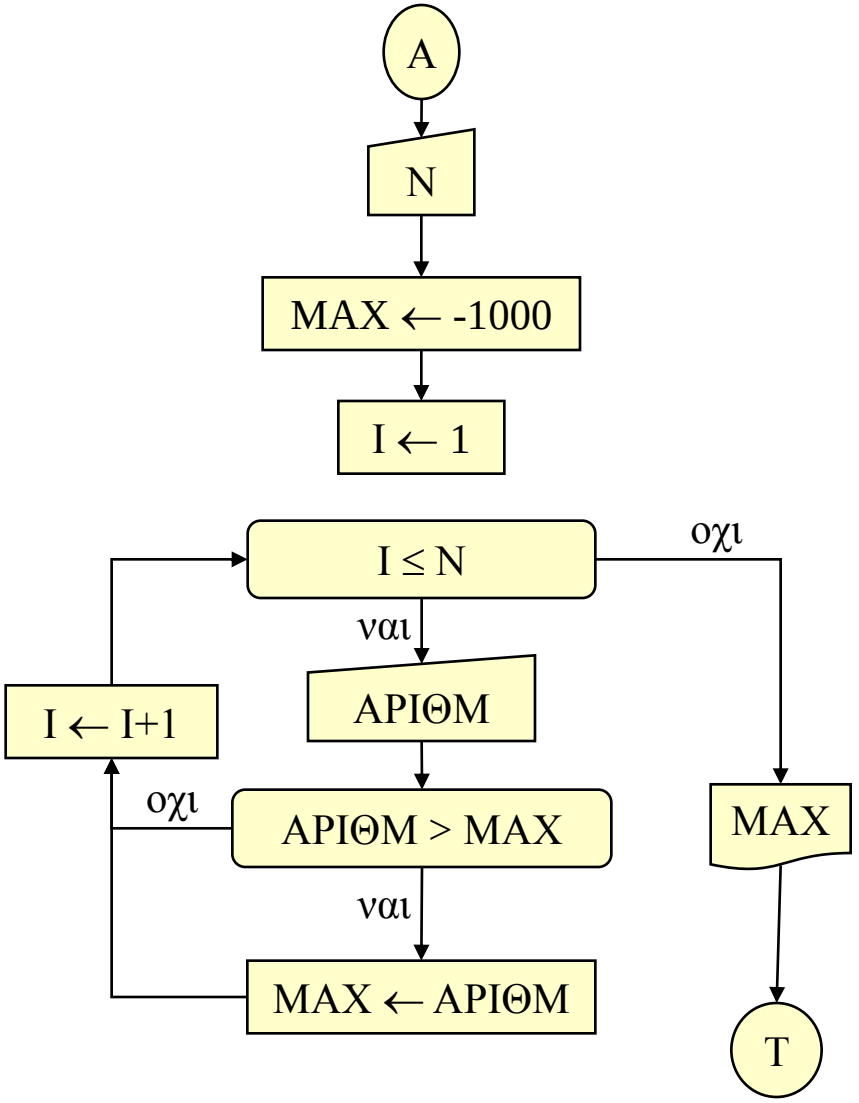
- Το πλήθος Ν των στοιχείων μας το δίνει ο χρήστης. (δηλαδή μπορούμε να το διαβάσουμε).

Εύρεση Μέγιστου - ΕΑΓ

```
διαδικασια ΜΕΓΙΣΤΟ2
δηλωση (N, ΑΡΙΘΜ, I, MAX) ακερ;
αρχη
    διαβασε N;
    MAX ← -1000000000;
    για I ← 1 εως N επαναλαβε
    (
        διαβασε ΑΡΙΘΜ;
        εαν  ΑΡΙΘΜ > MAX
                τοτε   MAX ← ΑΡΙΘΜ;
    )
    τυπωσε MAX;
τελος ΜΕΓΙΣΤΟ2
```

Εύρεση Μέγιστου – ΕΑΓ, ΔΡΠ

```
διαδικασια ΜΕΓΙΣΤΟ2
δηλωση (N, ΑΡΙΘΜ, I, MAX) ακερ;
αρχη
    διαβασε N;
    MAX ← -1000000000;
    για I ← 1 εως N επαναλαβε
    (
        διαβασε ΑΡΙΘΜ;
        εαν  ΑΡΙΘΜ > MAX
                τοτε   MAX ← ΑΡΙΘΜ;
    )
    τυπωσε MAX;
τελος ΜΕΓΙΣΤΟ2
```



Εύρεση Μέγιστου – ΕΑΓ, ΔΡΠ, C++

```
διαδικασια ΜΕΓΙΣΤΟ2
δηλωση (N, ΑΡΙΘΜ, I, MAX) ακερ;
αρχη
    διαβασε N;
    MAX ← -1000000000;
    για I ← 1 εως N επαναλαβε
    (
        διαβασε ΑΡΙΘΜ;
        εαν  ΑΡΙΘΜ > MAX
                τοτε   MAX ← ΑΡΙΘΜ;
    )
    τυπωσε MAX;
τελος ΜΕΓΙΣΤΟ2
```

```
// όνομα αρχείου MAX2.cpp
#include <iostream>
main()
{
   int N, ARITHM, I, MAX;
   cin >> N;
   MAX = -100000000;
   for (I=0; I<=N; I++)
   {
        cin >> ARITHM;
        if (ARITHM > MAX)
              MAX = ARITHM;
   }
   cout << MAX;
}
```

Γνωστό και Άγνωστο πλήθος

```
διαβασε N;
..
για I ← 1 εως N επαναλαβε
(
        ...
        διαβασε ΑΡΙΘΜ;
)
```

```
cin >> N;
...
for (I=0; I<=N; I++)
{
    cin >> ARITHM;
    ...
}
```

```
διαβασε ΤΕΡΜΑΤ, ΑΡΙΘΜ;
...
εφοσον ΤΕΡΜΑΤ ≠ ΑΡΙΘΜ επαναλαβε
(
        ....
        διαβασε ΑΡΙΘΜ;
)
```

```
cin >> TERMAT >> ARITHM;
...
while(TERMAT != ARITHM)
{
    ...
    cin >> ARITHM;
}
```

Εύρεση Μέσου όρου

- Να περιγραφεί η διαδικασία εύρεσης μέσου όρου, αγνώστου πλήθους, ακεραίων αριθμών, χρησιμοποιώντας (i) την ΕΑΓ, (ii) το ΔΡΠ, και (iii) την C++.

- Μήπως μπορούμε να τροποποιήσουμε κάποια γνωστή διαδικασία;

- Μέσος_Όρος = Άθροισμα / Πλήθος

- Χρειάζεται όμως προσοχή...

```
διαδικασια ΜΕΣΟΣΟΡΟΣ1
δηλωση (ΤΕΡΜΑΤ, ΑΡΙΘΜ, ΑΘΡ, Ν) ακερ;
δηλωση (ΜΕΣΟΡ) πραγματικ;
αρχη
    διαβασε ΤΕΡΜΑΤ, ΑΡΙΘΜ;
    ΑΘΡ ← 0;
     Ν ← 1;
    εφοσον  ΤΕΡΜΑΤ ≠ ΑΡΙΘΜ  επαναλαβε
    (
            Ν ← Ν+1;
            ΑΘΡ ← ΑΘΡ + ΑΡΙΘΜ;
            διαβασε ΑΡΙΘΜ;
    )
     ΜΕΣΟΡ ← ΑΘΡ / Ν;
    τυπωσε ΜΕΣΟΡ ;
τελος ΜΕΣΟΣΟΡΟΣ1
```

Εύρεση Μέσου όρου

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(int argc, char *argv[])
{  system("chcp 1253");
   int  number;           /* ακέραιοι */
   int  total = 0;        /* άθροισμα */
   int  count = 0;        /* πλήθος   */
   cout<<"Δώσε επόμενο ακέραιο - (0 για τέλος): ";
   cin>>number;
   while (number != 0)
   {  total += number;
      count=count + 1;
      cout<<"Δώσε επόμενο ακέραιο - (0 για τέλος): ";
      cin>>number;
      }
   cout<<"Τέλος εισόδου.\n";
```

Εύρεση Μέσου όρου

- cout<<"Άθροισμα -> "<<total<<"\n";
- cout<<"Πλήθος -> "<<count<<"\n";
- if (count > 0) {cout<<"Μέσος όρος (ακέραιος) -> "<<total / count<<endl;
- cout<<"Μέσος όρος (πραγματικός) -> "<<(float) total / count<<endl;}
- else cout<<"Μηδενικό πλήθος αριθμών"<<endl;
-
- system("PAUSE");
- return 0;
- }