



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

# Εισαγωγή στην πληροφορική

**Βασίλειος Βεσκούκης**

Δρ. Ηλεκτρολόγος Μηχανικός &  
Μηχανικός Υπολογιστών ΕΜΠ  
[v.vescoukis@cs.ntua.gr](mailto:v.vescoukis@cs.ntua.gr)

**Ρωμύλος Κορακίτης**

Αstroφυσικός  
Αναπλ. Καθηγητής ΕΜΠ  
[romylos@survey.ntua.gr](mailto:romylos@survey.ntua.gr)

*Ροή εργασιών προγραμματισμού*

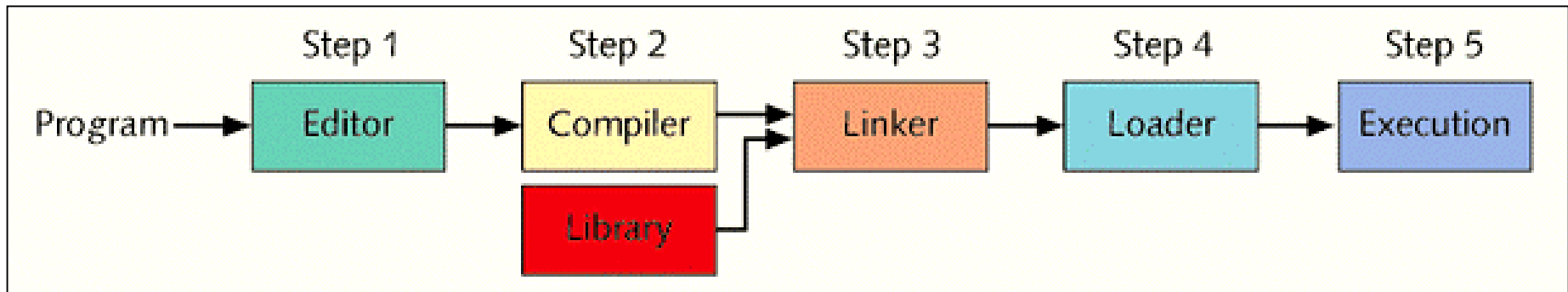
*Εισαγωγή στο περιβάλλον της γλώσσας C++*

*Έλεγχος - αποσφαλμάτωση προγραμμάτων*

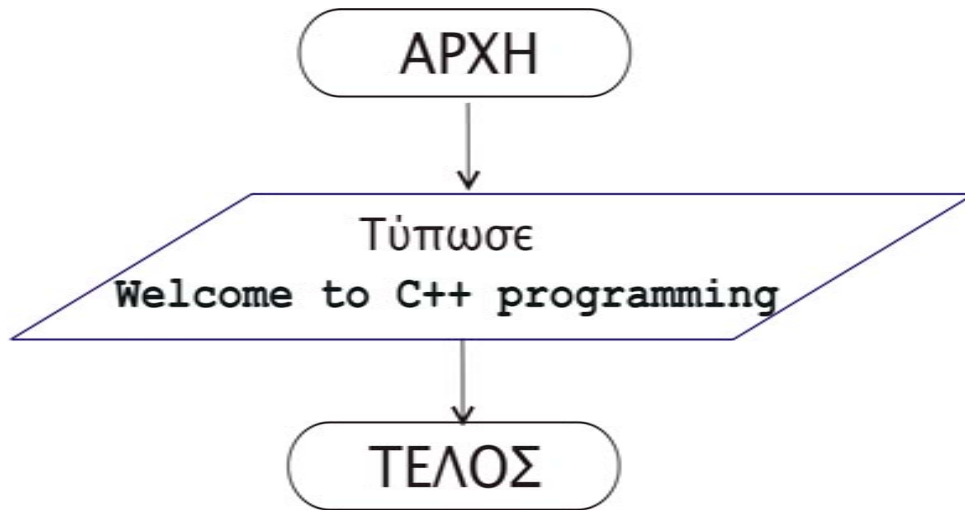
*Παραδείγματα*

## Ροή εργασιών προγραμματισμού

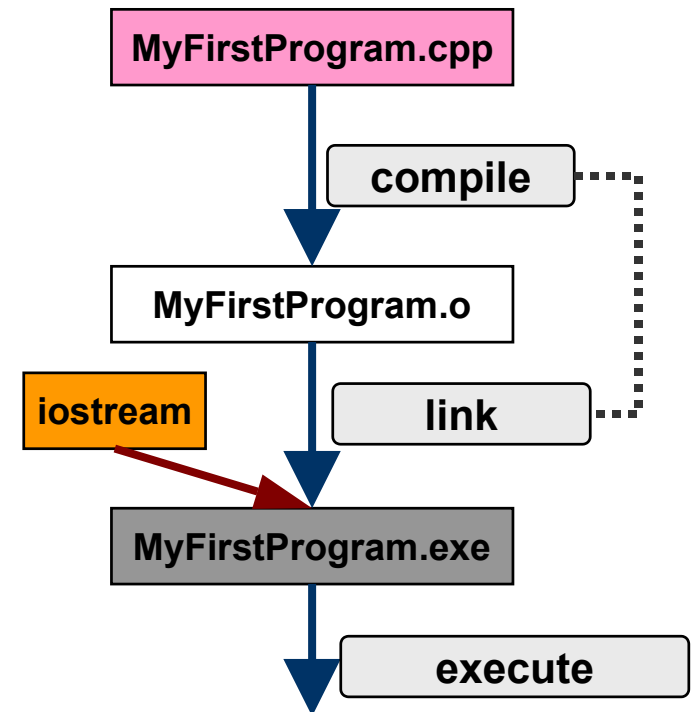
1. Με χρήση ενός προγράμματος **συντάκτη κειμένου (editor)** γράφουμε το πηγαίο πρόγραμμα (source program) σε μια γλώσσα προγραμματισμού (λ.χ. C++)
2. Με χρήση του μεταφραστή της C++ (**compiler**) ελέγχουμε αν το πηγαίο πρόγραμμα τηρεί τους κανόνες σύνταξης της γλώσσας και εφόσον το κάνει, το μεταφράζουμε σε μια ισοδύναμη μορφή που λέγεται object program (μη αναγνώσιμη από τον άνθρωπο)  
**(Μεταγλώττιση)**
3. Με χρήση του "συνδυαστή" (**linker**) συνδυάζουμε το object program που δημιουργήσαμε με τυχόν συστατικά στοιχεία έτοιμων object προγραμμάτων, τα οποία διατίθενται με τη μορφή "βιβλιοθήκης», ώστε να δημιουργηθεί το εκτελέσιμο αρχείο  
**(Σύνδεση)**
4. Με τη βοήθεια του λειτουργικού συστήματος, φορτώνουμε το εκτελέσιμο πρόγραμμα στη μνήμη του υπολογιστή, όπου εκτελείται



# Ένα πρώτο πρόγραμμα



```
//MyFirstProgram.cpp  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    cout<<"Welcome to C++ programming";  
    return 0;  
}
```



**Welcome to C++ Programming**

# Στοιχεία προγράμματος C++

## Αναγνωριστικά ονόματα (identifiers)

- Μεταβλητές μνήμης
- Σταθερές

## Τελεστές

- Αριθμητικοί
- Λογικοί
- Σύγκρισης

## Εκφράσεις, εντολές

- Δεσμευμένες λέξεις
- Αριθμητικές και λογικές εκφράσεις
- Έλεγχος ροής προγράμματος (δομές)

## Συναρτήσεις

## Σχόλια

## Ένα πρώτο πρόγραμμα - αναγνώριση στοιχείων

```
//MyFirstProgram.cpp
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
cout << "Welcome to C++ Programming";
```

```
return 0;
```

```
}
```

Δήλωση χρήσης βιβλιοθήκης

Όνομα ενσωματωμένης βιβλιοθήκης

Συνάρτηση

Αναγνωριστικό του καθιερωμένου ρεύματος εξόδου της βιβλιοθήκης `iostream`

Τελεστής εξόδου

Αλφαριθμητική σταθερά

**ΕΞΟΔΟΣ:**

**Welcome to C++ Programming**

## Παράδειγμα 1

### Υπολογισμός της περιμέτρου και του εμβαδού ενός παραλληλογράμμου

Γνωρίζουμε ότι

- $\text{Περίμετρος} = 2 * [ (\text{μήκος πλευράς 1}) + (\text{μήκος πλευράς 2}) ]$
- $\text{Εμβαδόν} = (\text{μήκος πλευράς 1}) * (\text{μήκος πλευράς 2})$

Όταν και μόνον όταν

- Τα μήκη είναι εκπεφρασμένα στις ίδιες μονάδες

Στοιχεία **εισόδου**

- Μήκος πλευράς 1 (a)
- Μήκος πλευράς 2 (b)

Στοιχεία **εξόδου**

- Περίμετρος (perimeter)
- Εμβαδόν (area)

Υπολογισμοί

- $\text{perimeter} = 2 * (a + b)$  [μονάδα μήκους]
- $\text{area} = a * b$  [μονάδα μήκους]<sup>2</sup>

## Παράδειγμα 1 (συνέχεια)

Σημεία προσοχής:

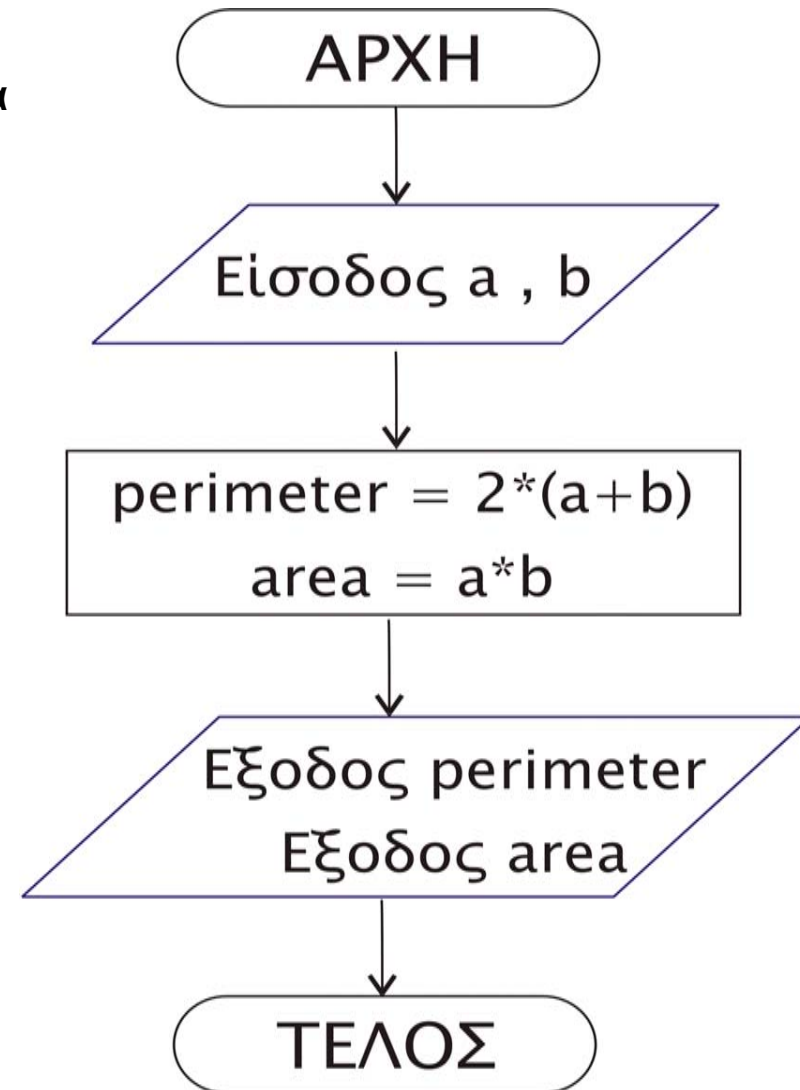
- Να διαβάσουμε τα δεδομένα εισόδου και «να τα αποθηκεύσουμε κάπου»
- Να δεσμεύσουμε χώρο για τα αποτελέσματα
- Να κάνουμε σωστά τους υπολογισμούς

Αλγόριθμος 1 (εκδοχή α)

- ΑΡΧΗ
- Διάβασε το **a**
- Διάβασε το **b**
- Υπολόγισε το **perimeter**
- Υπολόγισε το **area**
- Εκτύπωσε το **perimeter**
- Εκτύπωσε το **area**
- ΤΕΛΟΣ

Παραδοχές

- Ιδίες μονάδες
- «Τρέχει» μόνο μία φορά



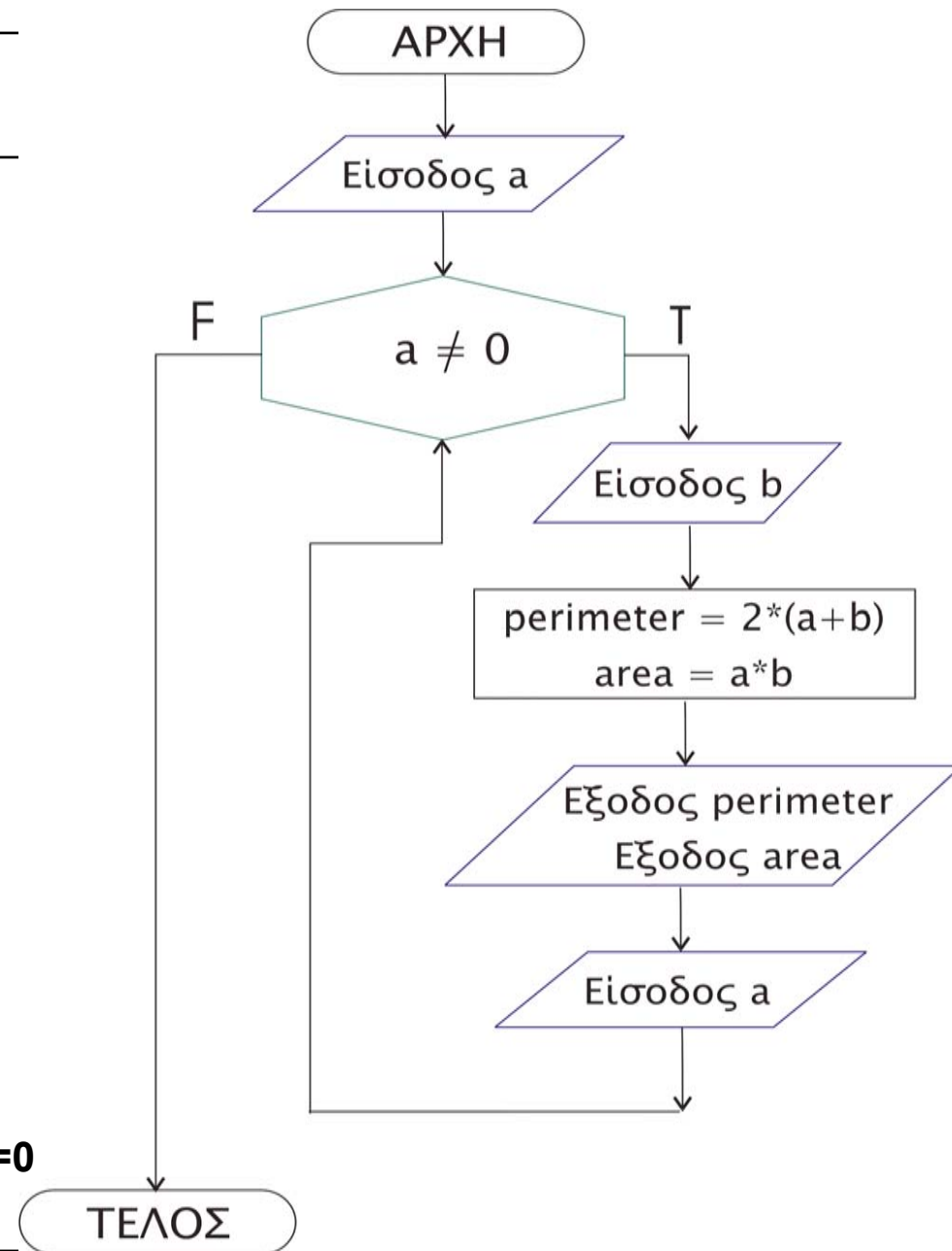
## Παράδειγμα 1 (συνέχεια)

### Αλγόριθμος 1 (εκδοχή β)

- ΑΡΧΗ
- Διάβασε το **a**
- Όσο  $a \neq 0$ 
  - Διάβασε το **b**
  - Υπολόγισε το **perimeter**
  - Υπολόγισε το **area**
  - Εκτύπωσε το **perimeter**
  - Εκτύπωσε το **area**
  - Διάβασε το **a**
- Τέλος επανάληψης
- ΤΕΛΟΣ

### Παρατηρήσεις

- Επαναληπτική εκτέλεση μέχρις ότου  $a=0$

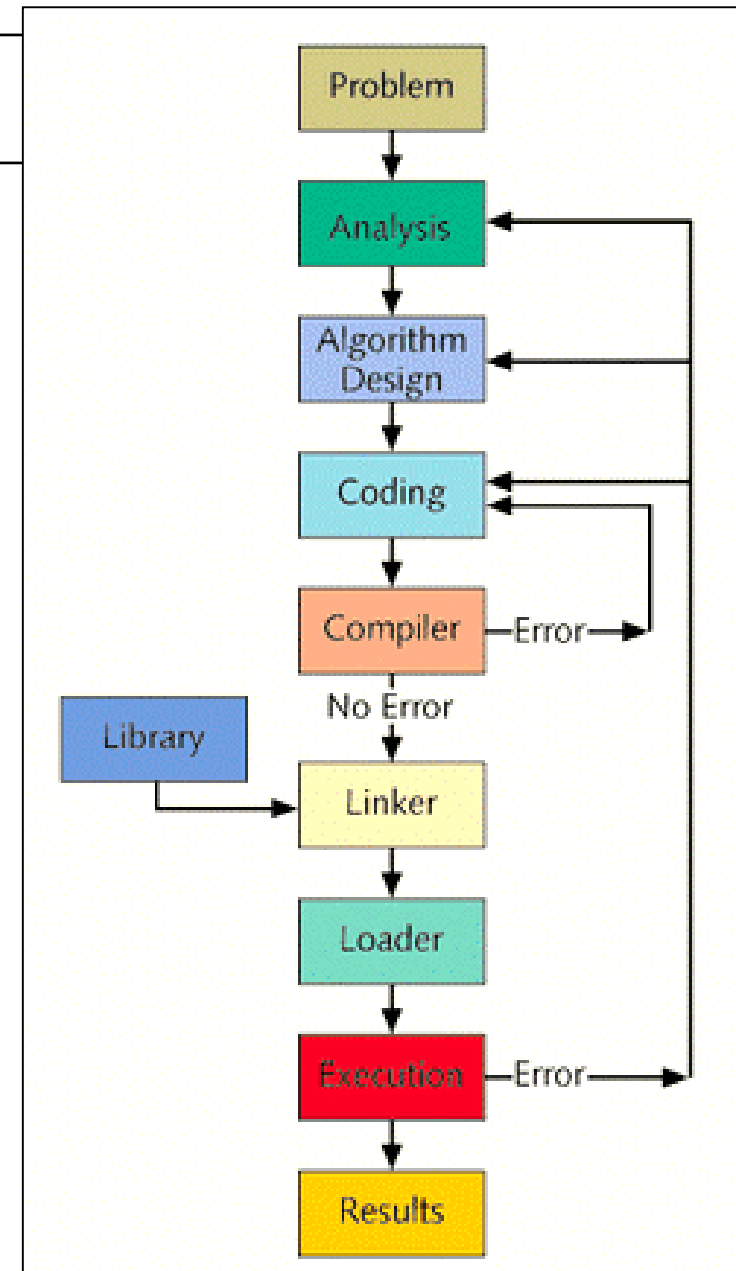




## Έλεγχος του προγράμματος

Ο έλεγχος του προγράμματος γίνεται με σειρά αντίστροφη από την δημιουργία του. Οι τρεις φάσεις οπισθοδρόμησης στην διαδικασία του προγραμματισμού είναι:

- Διόρθωση **συντακτικών σφαλμάτων** : ανιχνεύονται από τον compiler και απαιτούν διορθώσεις στην κωδικοποίηση του αλγορίθμου (πηγαίος κώδικας)
- Διόρθωση **λογικών σφαλμάτων**: μερικά μπορεί να ανιχνευθούν επειδή προκαλούν σφάλμα κατά την εκτέλεση του προγράμματος (π.χ. διαίρεση με 0). Άλλα λογικά σφάλματα εντοπίζονται με εξαντλητικό έλεγχο της συμπεριφοράς του προγράμματος με διάφορες τιμές των δεδομένων εισόδου. Αντιμετωπίζονται με τις κατάλληλες τροποποιήσεις στην σχεδίαση και κωδικοποίηση του αλγορίθμου.
- Διόρθωση **σφαλμάτων αρχής**: παρόλο που το πρόγραμμα εκτελείται κανονικά, μπορεί τα αποτελέσματα να μην ανταποκρίνονται στις αρχικές επιθυμίες. Αυτό σημαίνει ότι υπάρχει σφάλμα στην σύλληψη ή την διατύπωση του προβλήματος, που αντιμετωπίζεται με νέα διατύπωση και ανάλυση του προβλήματος και επανασχεδιασμό του αλγορίθμου.



## Παράδειγμα 2

### Πρόβλημα:

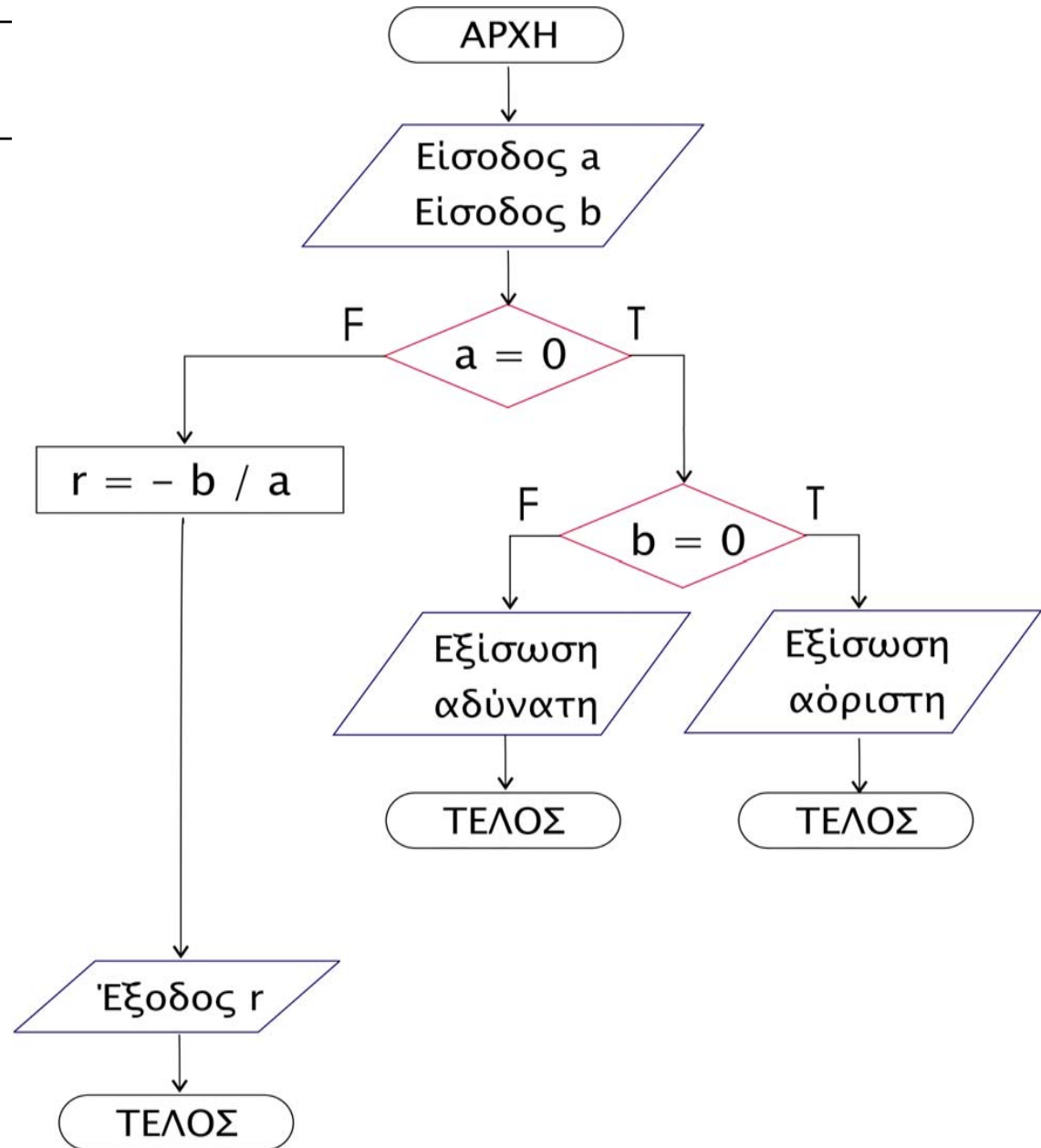
Να σχεδιαστεί ένα πρόγραμμα που θα λύνει την πρωτοβάθμια εξίσωση της μορφής :  $ax+b = 0$

Δεδομένα εισόδου:  $a, b$

Δεδομένα εξόδου: η τιμή της ρίζας  $r$

Μετασχηματισμοί:  $r = -b/a$

Περιορισμοί:  $a \neq 0$



## Παράδειγμα 3

### Πρόβλημα:

Να σχεδιαστεί ένα πρόγραμμα που θα μετατρέψει τις ορθογώνιες συντεταγμένες  $(x,y)$  ενός σημείου του επιπέδου στις αντίστοιχες πολικές  $(r,a)$

Δεδομένα εισόδου:  $x, y$

Δεδομένα εξόδου:  $r, a$

Μετασχηματισμοί:  $r = (x^2 + y^2)^{1/2}$

$a = \arctan(y/x)$

Περιορισμοί:  $0 \leq a < 2\pi$

