



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Προγραμματισμός Ι

Εισαγωγικά Θέματα

Α. Θεοδωρόπουλος, Γ. Λέπουρας

Ιανουάριος 2013

Πίνακας περιεχομένων

1. Η ΕΝΝΟΙΑ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ- ΑΛΓΟΡΙΘΜΟΙ	3
Πρόβλημα - Επίλυση ενός προβλήματος	3
Δεδομένα – Επεξεργασία – Πληροφορία	3
Στάδια αντιμετώπισης ενός προβλήματος	3
Επίλυση ενός προβλήματος στον υπολογιστή	4
Τι ονομάζεται αλγόριθμος;	4
Κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος	4
Τρόποι αναπαράστασης και περιγραφής ενός αλγορίθμου	5
Τι ορίζουμε ως δομή δεδομένων;	5
Τι είναι πρόγραμμα και τι προγραμματισμός;	5
Γλώσσες μηχανής	6
Στοιχειώδεις ενέργειες που μπορεί να κάνει ένας υπολογιστής;	6
Γλώσσες συμβολομετάφρασης (χαμηλού επιπέδου)	6
Γλώσσες υψηλού επιπέδου.....	6
Επιλογή μιας γλώσσας προγραμματισμού;.....	6
Από τι προσδιορίζεται μια γλώσσα προγραμματισμού;	7
2. ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ (ΔΡΔ)	8
Σύμβολα ΔΡΔ	8
Δομές ΔΡΔ (δομές ελέγχου)	9
Παραδείγματα δομημένων και μη δομημένων ΔΡΔ	11
Κανόνες για το σχηματισμό δομημένων ΔΡΔ	11
Παραδείγματα ΔΡΔ	12
Παράδειγμα 1_Απλή πρόσθεση δύο αριθμών	12
Παράδειγμα 2_Άθροισμα N ακέραιων αριθμών	13
Παράδειγμα 3_Άθροισμα N ακέραιων αριθμών μόνο θετικών	15
Παράδειγμα 4_Άθροισμα αγνώστου αριθμού θετικών, τερματίζει με το 0.....	17
Παράδειγμα 5_Εύρεση μεγαλύτερου από 3 αριθμούς.....	19
Παράδειγμα 6_Εύρεση μεγαλύτερου από σειρά αριθμών	21
Παράδειγμα 7_Εύρεση μέσου όρου N αριθμών	23
Παράδειγμα 8_Επίλυση πρωτοβάθμιας εξίσωσης.....	25
Παράδειγμα 9_Επίλυση δευτεροβάθμιας εξίσωσης.....	27
Παράδειγμα 10_Μενού επιλογών, τερματισμός με 0.....	30
Παράδειγμα 11_Διαβάζει βαθμολογία και τυπώνει αντίστοιχο αποτέλεσμα	33

Παράδειγμα 12_Διάβασμα μονοδιάστατου πίνακα	35
Παράδειγμα 13_Διάβασμα μονοδιάστατου πίνακα θετικών.....	37
Παράδειγμα 14_Μονοδιάστατος πίνακας, εκτύπωση άρθρων	39
Παράδειγμα 15_Πολλαπλασιασμός ακεραίου με διάνυσμα	41
Ασκήσεις για εξάσκηση	43

[Οι σημειώσεις αυτές αποτελούν ένα βοήθημα του μαθήματος Προγραμματισμός Ι, στοχεύοντας να καλύψουν τυχόν κενά σε φοιτητές χωρίς προηγούμενη εμπειρία στον προγραμματισμό (π.χ. θετική κατεύθυνση Δευτεροβάθμιας εκπαίδευσης). Παρουσιάζονται βασικές έννοιες αλγορίθμων και δίνεται έμφαση στις διαγραμματικές τεχνικές.]

1. Η ΕΝΝΟΙΑ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ- ΑΛΓΟΡΙΘΜΟΙ

Πρόβλημα - Επίλυση ενός προβλήματος

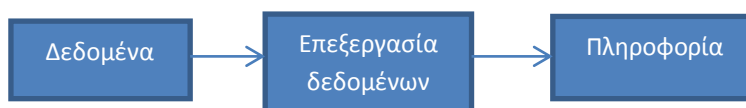
Με τον όρο πρόβλημα εννοούμε μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της δεν είναι γνωστή ούτε προφανής.

Η **επίλυση** ενός προβλήματος είναι η διαδικασία μέσω της οποίας βρίσκουμε το **ζητούμενο**, δηλαδή το σημείο στο οποίο πρέπει να φτάσουμε για να βγούμε από την «προβληματική» κατάσταση στην οποία βρισκόμαστε.

Η κατανόηση ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων, της **σωστής διατύπωσης** εκ μέρους του δημιουργού του και της αντίστοιχα **σωστής ερμηνείας** από τη μεριά εκείνου που καλείται να το αντιμετωπίσει. Όσον αφορά τη διατύπωση η άστοχη χρήση ορολογίας και η λανθασμένη σύνταξη είναι δύο στοιχεία που μπορούν να προκαλέσουν παρερμηνείες και παραπλανήσεις. Όσον αφορά τη σωστή ερμηνεία, σημαντικό ρόλο παίζει το επίπεδο της γνώσης και της αντίληψης του λύτη, το οποίο καθορίζεται από τις γενικές και τις ειδικές γνώσεις του πάνω στο αντικείμενο του προβλήματος, καθώς επίσης η εμπειρία, το ενδιαφέρον και η ευφυΐα του.

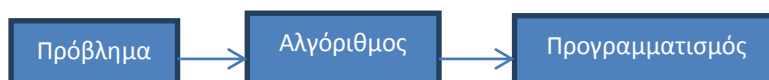
Δεδομένα – Επεξεργασία – Πληροφορία

- **Δεδομένα** είναι οποιαδήποτε στοιχεία μπορούν να γίνουν αντιληπτά από έναν παρατηρητή με μια από τις πέντε αισθήσεις του. Τα δεδομένα είναι τα στοιχεία πάνω στα οποία μπορούμε να στηριχτούμε για να λύσουμε ένα πρόβλημα.
- **Επεξεργασία δεδομένων** είναι η διαδικασία κατά την οποία ένας «μηχανισμός» δέχεται δεδομένα, τα επεξεργάζεται σύμφωνα με έναν προκαθορισμένο τρόπο και αποδίδει πληροφορίες.
- **Πληροφορία** αποτελεί οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.



Στάδια αντιμετώπισης ενός προβλήματος

- Κατανόηση
- Ανάλυση
- Επίλυση



Επίλυση ενός προβλήματος στον υπολογιστή

Οι λόγοι για τους οποίους αναθέτουμε την επίλυση ενός προβλήματος στον υπολογιστή σχετίζονται με:

- την πολυπλοκότητα των υπολογισμών,
- την επαναληπτικότητα των διαδικασιών,
- την ταχύτητα εκτέλεσης πράξεων,
- το μεγάλο πλήθος δεδομένων.

Τι ονομάζεται αλγόριθμος;

Αλγόριθμος είναι μια πεπερασμένη ακολουθία αυστηρά καθορισμένων εντολών, καθεμία από τις οποίες είναι εκτελέσιμη σε πεπερασμένο χρόνο, που στοχεύει στην επίλυση ενός συγκεκριμένου προβλήματος.

παραδείγματα αλγορίθμων:

- Ο υπολογισμός της σχολικής βαθμολογίας.
Βήμα 1. Πάρε τη βαθμολογία που έχει ο μαθητής στα γραπτά.
Βήμα 2. Πάρε επίσης τη βαθμολογία που έχει ο μαθητής στα προφορικά.
Βήμα 3. Πρόσθεσε τις δύο αυτές βαθμολογίες.
Βήμα 4. Υπολόγισε το μέσο όρο διαιρώντας το προηγούμενο αποτέλεσμα με 2.
Βήμα 5. Αν ο μέσος όρος είναι μεγαλύτερος του 9.5, τότε εμφάνισε το μήνυμα ότι ο μαθητής περνάει την τάξη, διαφορετικά εμφάνισε το μήνυμα ότι ο μαθητής δεν περνάει την τάξη.
- Η αγαπημένη σου συνταγή μαγειρικής.
Βήμα 1. Άνοιξε το μάτι της κουζίνας στο 2.
Βήμα 2. Βάλε 3 λίτρα νερό σε μια κατσαρόλα χωρητικότητας 4 λίτρων.
Βήμα 3. Τοποθέτησε την κατσαρόλα στο μάτι που έχεις ήδη ανάψει.
Βήμα 4. Πρόσθεσε στην κατσαρόλα μια κουταλιά αλάτι.
Βήμα 5. Περίμενε μέχρι να βράσει το νερό.
Βήμα 6. Βάλε τα μακαρόνια στην κατσαρόλα.
Βήμα 7. Ανακάτευε τα μακαρόνια για 10 λεπτά.

Κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος

Κάθε αλγόριθμος πρέπει να ικανοποιεί τα επόμενα κριτήρια:

- **Είσοδος (input):** Το σύνολο των τιμών που δέχεται ο αλγόριθμος ως δεδομένα για την επίλυση του προβλήματος. Ως είσοδος μπορεί να είναι το κενό σύνολο, όταν για παράδειγμα ο αλγόριθμος επεξεργάζεται δεδομένα που είτε τα παράγει ο ίδιος (π.χ. τυχαίους αριθμούς), είτε αποτελούν γνωστές τιμές από τη διατύπωση του ίδιου του προβλήματος.

- **Έξοδος (output):** Είναι το σύνολο των τιμών που δημιουργεί ο αλγόριθμος ως αποτέλεσμα (δεν μπορεί να είναι το κενό σύνολο).
- **Καθοριστικότητα (definiteness):** Κάθε εντολή ενός αλγορίθμου θα πρέπει να είναι επακριβώς και αυστηρώς καθορισμένη, έτσι που η εκτέλεσή τους να γίνεται χωρίς καμία αμφιβολία (π.χ. μια εντολή διαίρεσης δεν πρέπει να γίνει στην περίπτωση που ο διαιρέτης λαμβάνει μηδενική τιμή).
- **Περατότητα (finiteness):** Ο αλγόριθμος να τερματίζει έπειτα από πεπερασμένα βήματα εκτέλεσης των εντολών του.
- **Αποτελεσματικότητα (effectiveness):** Κάθε μεμονωμένη εντολή ενός αλγορίθμου πρέπει να είναι απλή. Αυτό σημαίνει ότι δεν πρέπει απλά να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

Τρόποι αναπαράστασης και περιγραφής ενός αλγορίθμου

- Ελεύθερο κείμενο
- Φυσική γλώσσα
- Διαγραμματικές τεχνικές (diagrammatic techniques)*
- Κωδικοποίηση (coding) – ψευδογλώσσα

Τι ορίζουμε ως δομή δεδομένων;

Δομή δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίσταται επεξεργασία από ένα σύνολο λειτουργιών (π.χ. ένας πίνακας).

Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα

Τι είναι πρόγραμμα και τι προγραμματισμός;

Πρόγραμμα είναι το σύνολο των εντολών που πρέπει να δοθούν στον υπολογιστή ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος. Το πρόγραμμα, το οποίο γράφεται σε κάποια γλώσσα προγραμματισμού, δεν είναι απλώς η υλοποίηση του αλγορίθμου, αλλά βασικό στοιχείο του είναι τα δεδομένα και οι δομές δεδομένων επί των οποίων ενεργεί.

Προγραμματισμός είναι η διεργασία που γίνεται για τη δημιουργία ενός προγράμματος.

Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία του ανθρώπου (προγραμματιστή) με τη μηχανή (υπολογιστής).

* Το παρόν εγχειρίδιο ασχολείται με διαγραμματικές τεχνικές και συγκεκριμένα με Διαγράμματα Ροής Δεδομένων (ΔΡΔ).

Γλώσσες μηχανής

Οι γλώσσες μηχανής είναι οι πρώτες γλώσσες προγραμματισμού στις οποίες οι εντολές για την υλοποίηση ενός προγράμματος έπρεπε να δοθούν από τον προγραμματιστή σε ακολουθίες 0 και 1, δηλαδή σε μορφή κατανοητή από τον υπολογιστή.

Ακόμα και στις σύγχρονες γλώσσες προγραμματισμού οι εντολές κάθε προγράμματος μετατρέπονται σε ακολουθίες 0 και 1, δηλαδή σε εντολές σε γλώσσα μηχανής.

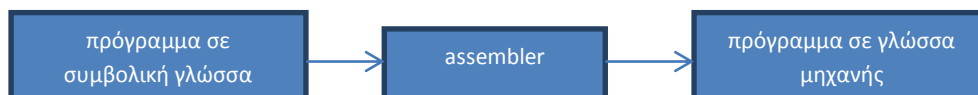
Στοιχειώδεις ενέργειες που μπορεί να κάνει ένας υπολογιστής;

Το μόνο που κάνει ένας υπολογιστής είναι στοιχειώδεις ενέργειες σε ακολουθίες 0 και 1 με ασύλληπτη ταχύτητα:

- αποθηκεύει στη μνήμη
- ανακτά από τη μνήμη,
- κάνει στοιχειώδεις αριθμητικές πράξεις
- και συγκρίνει.

Γλώσσες συμβολομετάφρασης (χαμηλού επιπέδου)

Οι γλώσσες αυτές είναι αποτέλεσμα προσπαθειών για τη δημιουργία μιας συμβολικής γλώσσας που θα ήταν πιο προσιτή στον άνθρωπο από τις ακολουθίες 0 και 1. Ο συμβολομεταφραστής (assembler) είναι ένα ειδικό πρόγραμμα που μεταφράζει τη συμβολική γλώσσα σε γλώσσα μηχανής.



Γλώσσες υψηλού επιπέδου

Με τη ραγδαία χρήση των υπολογιστών αναπτύχθηκαν οι γλώσσες υψηλού επιπέδου. Χρησιμοποιούν προγράμματα μετάφρασης που μετατρέπουν τα προγράμματα σε γλώσσα μηχανής και ονομάζονται μεταγλωττιστές (compilers). Π.χ. Fortran, Cobol, Algol, Lisp, Prolog, Basic, Pascal, C, C++, Java.

Επιλογή μιας γλώσσας προγραμματισμού;

Μπορούμε να ισχυριστούμε με βεβαιότητα ότι μια γλώσσα προγραμματισμού που να είναι αντικειμενικά καλύτερη από όλες τις άλλες δεν υπάρχει ούτε πρόκειται να υπάρξει.

Η επιλογή της γλώσσας για την ανάπτυξη μιας εφαρμογής εξαρτάται:

- από το είδος της εφαρμογής,

- το υπολογιστικό περιβάλλον στο οποίο θα εκτελεστεί,
- τα προγραμματιστικά περιβάλλοντα που διαθέτουμε
- και κυρίως τις γνώσεις και προτιμήσεις του προγραμματιστή.






Από τι προσδιορίζεται μια γλώσσα προγραμματισμού;

- Το αλφάβητο: το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα.
- Το λεξιλόγιο: ένα υποσύνολο όλων των ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου (τις λέξεις) που είναι δεκτές από τη γλώσσα.
- Τη γραμματική: αποτελείται από το τυπικό και το συντακτικό. Το τυπικό είναι το σύνολο των κανόνων που ορίζει τις μορφές που μια λέξη είναι αποδεκτή και το συντακτικό είναι το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνθεσης των λέξεων της γλώσσας για τη δημιουργία εντολών.
- Τη σημασιολογία: το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και κατ' επέκταση των εκφράσεων που χρησιμοποιούνται σε μια γλώσσα.

2. ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ (ΔΡΔ)

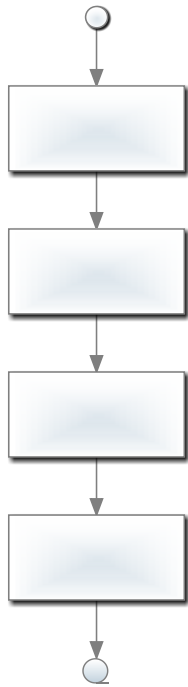
Σύμβολα ΔΡΔ

Ένα διάγραμμα ροής δεδομένων αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μια συγκεκριμένη ενέργεια ή λειτουργία. Τα κυριότερα γεωμετρικά σχήματα είναι τα εξής:

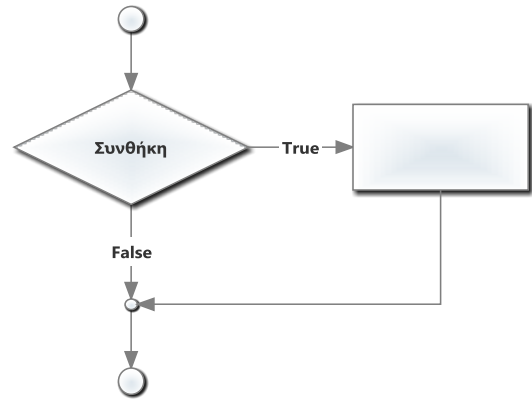
σύμβολο	επεξήγηση
	<i>έλλειψη, που δηλώνει την αρχή και το τέλος του κάθε αλγορίθμου</i>
	<i>πλάγιο παραλληλόγραμμο, που δηλώνει είσοδο ή έξοδο* στοιχείων</i>
	<i>ορθογώνιο, που δηλώνει την εκτέλεση μίας ή περισσότερων πράξεων</i>
	<i>ρόμβος, που δηλώνει ότι πρόκειται να ληφθεί μια απόφαση</i>
	<i>βέλη, που δείχνουν τη "ροή ελέγχου"</i>

* **Είσοδος:** πληκτρολόγιο, αρχείο. **Έξοδος:** οθόνη, εκτύπωση, αρχείο.

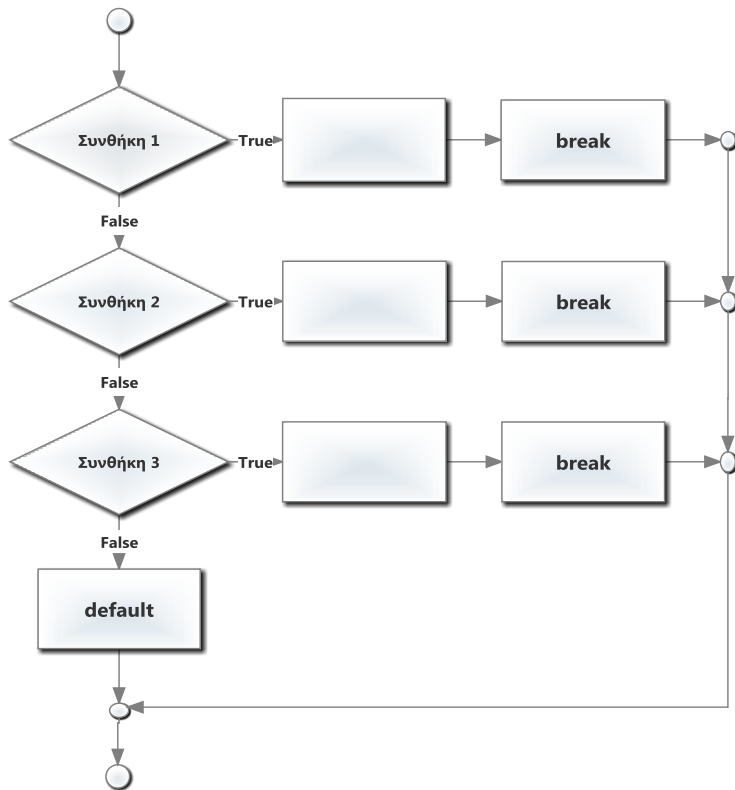
Δομές ΔΡΔ (δομές ελέγχου)



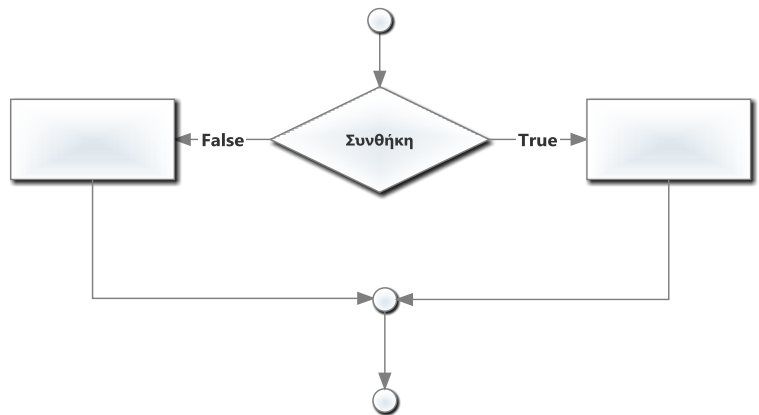
Σειρά (SEQUENCE structure)



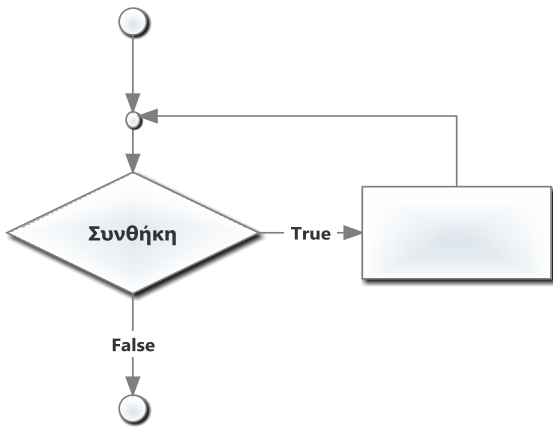
δομή if (μία επιλογή)



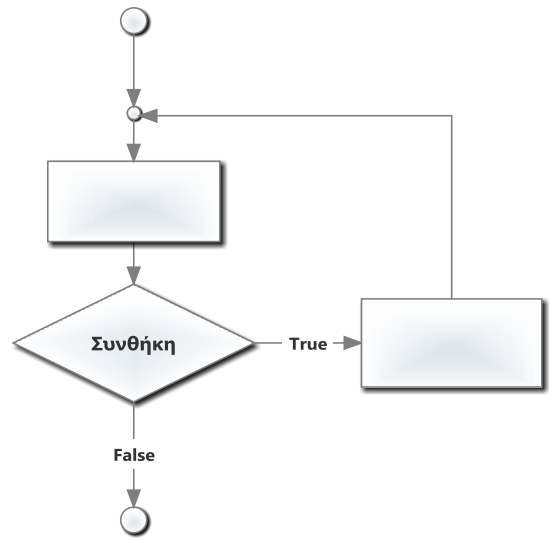
δομή switch (πολλές επιλογές)



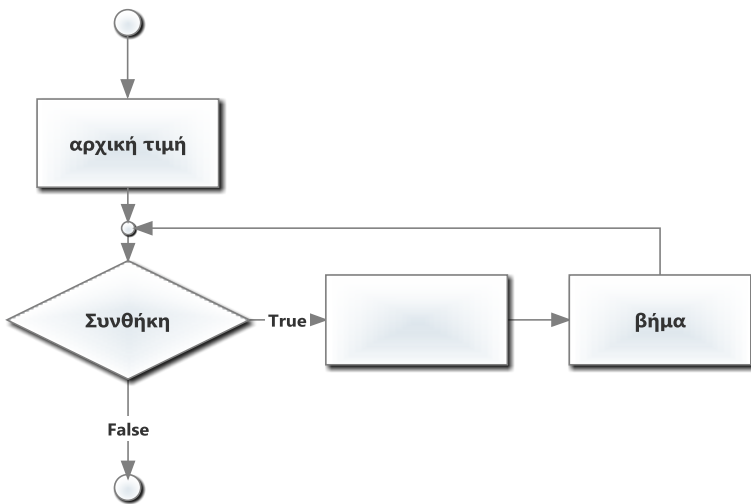
δομή if...else (διπλή επιλογή)



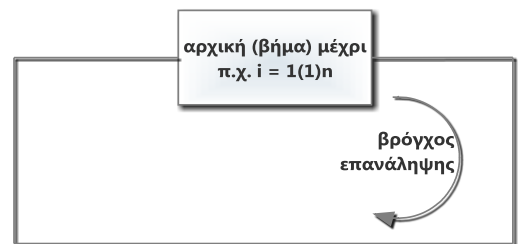
δομή while



δομή do...while

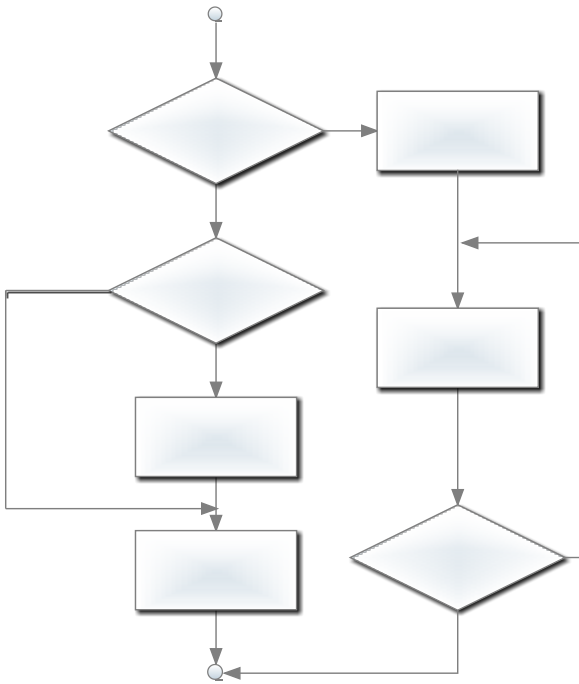


δομή for (I)

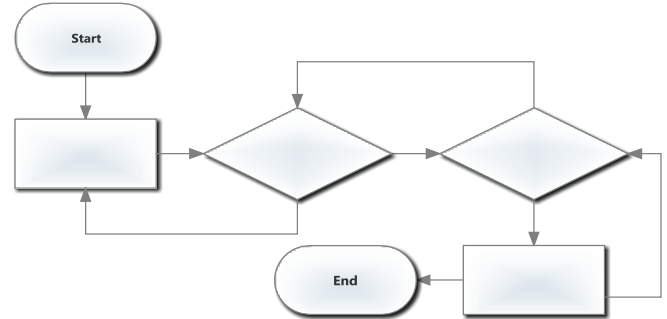


δομή for (II)

Παραδείγματα δομημένων και μη δομημένων ΔΡΔ



Δομημένο (Structured)



Μη δομημένο (Unstructured)

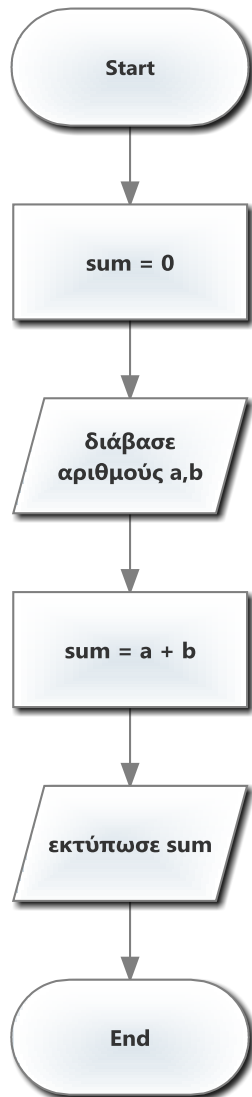
- σύνδεση συμβόλων με αυθαίρετο τρόπο
- δομές ελέγχου με περισσότερες από μία είσοδο/έξοδο

Κανόνες για το σχηματισμό δομημένων ΔΡΔ

- ✓ Αρχίστε με το απλούστερο δυνατό διάγραμμα ροής.
- ✓ Οποιοδήποτε ορθογώνιο (ενέργεια) μπορεί να αντικατασταθεί με δύο ορθογώνια (ενέργειες) στη σειρά.
- ✓ Οποιοδήποτε ορθογώνιο (ενέργεια) μπορεί να αντικατασταθεί από οποιαδήποτε δομή ελέγχου (σειράς, if, if...else, switch, while, do...while ή for).

Παραδείγματα ΔΡΔ

Παράδειγμα 1_Απλή πρόσθεση δύο αριθμών



ΔΡΔ

```
#include <stdio.h>

int main(void)
{
    int a,b;
    int sum = 0;

    printf("enter number a:");
    scanf("%d", &a);

    printf("enter number b:");
    scanf("%d", &b);

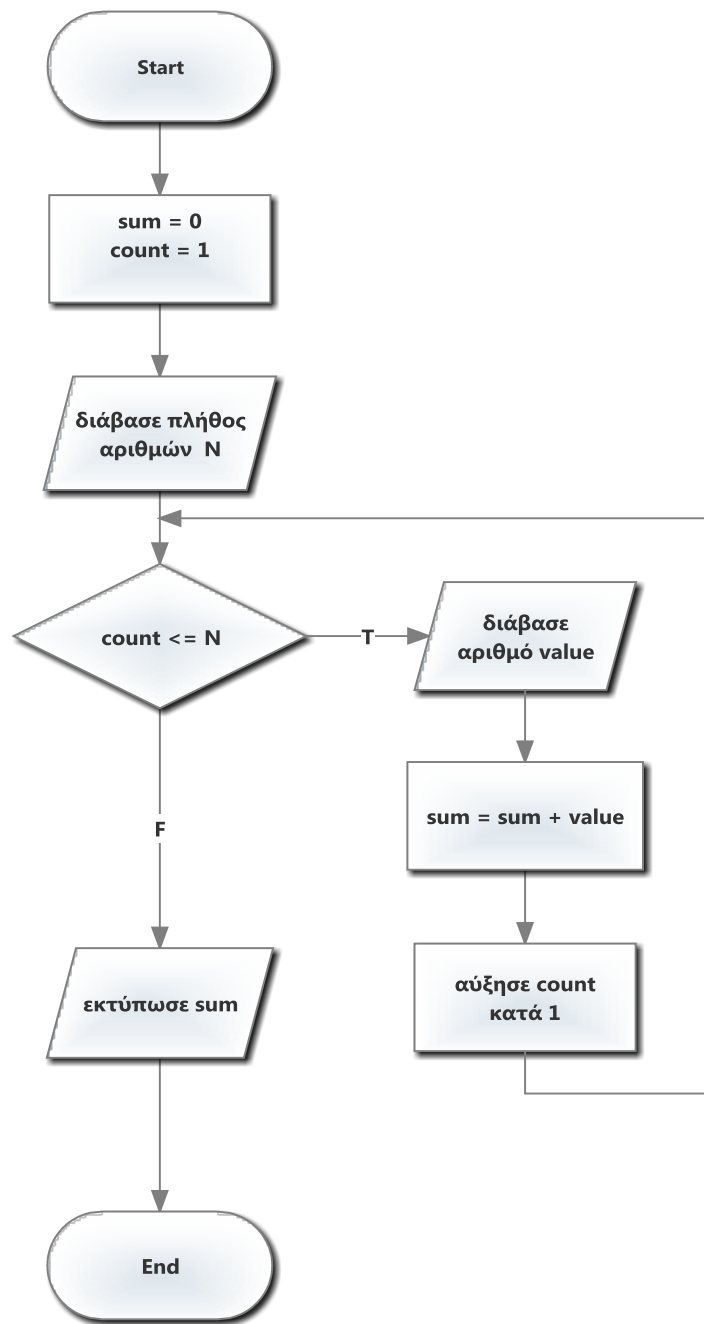
    sum = a + b;

    printf("The summary is: %d\n" , sum);

    return 0;
}
```

Κώδικας σε C

Παράδειγμα 2_Άθροισμα N ακέραιων αριθμών



ΔΡΔ

Κώδικας σε C (με 2 τρόπους)

```
/* Παράδειγμα 2α (με while)
   Άθροισμα N ακέραιων αριθμών */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int N, val; /* N: το πλήθος των αριθμών, val: η τιμή ενός αριθμού */
    int count = 1, sum = 0; /* count: μετρητής, sum: συνολικό άθροισμα */

    printf("How many numbers to add? ");
    scanf("%d", &N);

    while (count <= N)
    {
        printf("Enter number %d: ", count);
        scanf("%d", &val);
        sum += val;
        count +=1;
    } /* τέλος της while */

    printf("\nSummary of the given numbers is: sum = %d\n", sum);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

```
/* Παράδειγμα 2b (με for)
   Άθροισμα N ακέραιων αριθμών */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int N, val; /* N: το πλήθος των αριθμών, val: η τιμή ενός αριθμού */
    int count, sum = 0; /* count: μετρητής, sum: συνολικό άθροισμα */

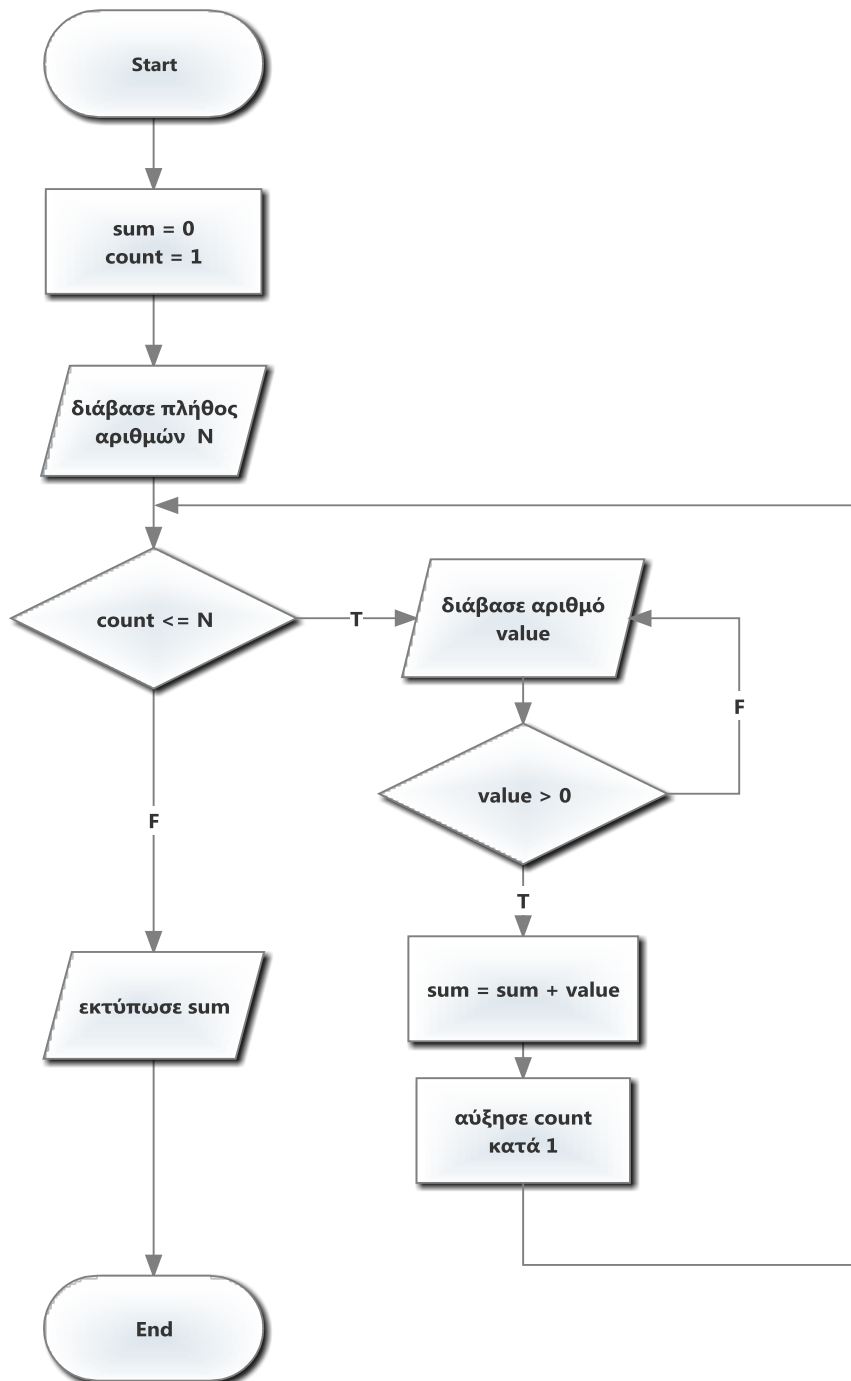
    printf("How many numbers to add? ");
    scanf("%d", &N);

    for (count = 1; count <=N; count++)
    {
        printf("Enter number %d: ", count);
        scanf("%d", &val);
        sum += val;
    } /* end of for */

    printf("\nSummary of the given numbers is: sum = %d\n", sum);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 3_Άθροισμα N ακέραιων αριθμών μόνο θετικών



ΔΡΔ

Κώδικας σε C

```
/* Παράδειγμα 3
   Άθροισμα N ακέραιων αριθμών μόνο θετικών */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int N, val; /* N: το πλήθος των αριθμών, val: η τιμή ενός αριθμού */
    int count = 1, sum = 0; /* count: μετρητής, sum: συνολικό άθροισμα */
    int valid = 0; /* μεταβλητή για έλεγχο αρνητικού */

    printf("How many numbers to add? ");
    scanf("%d", &N);

    while (count <= N)
    {
        printf("Enter number %d: ", count);
        scanf("%d", &val);

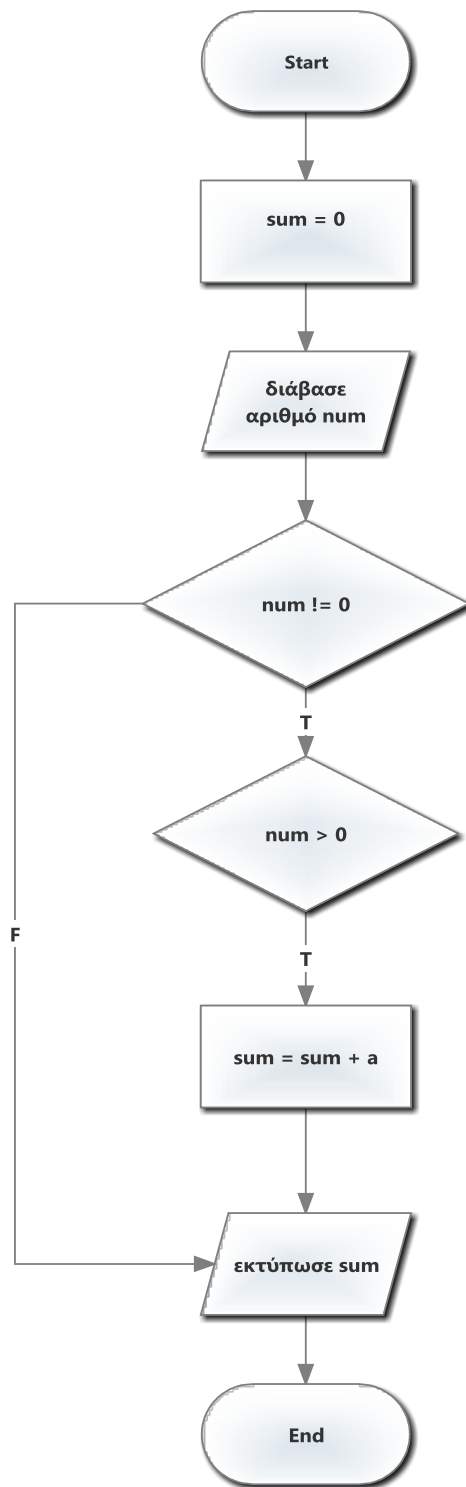
        if (val < 0)
        {
            printf("Number is negative. Please re-enter\n");
        } /* τέλος if */
        else
        {
            sum += val; /* προσθέτει την τιμή του δοθέντος αριθμού στο
                        συνολικό άθροισμα */
            count +=1; /* αυξάνουμε το μετρητή κατά 1 για να
                       διαβάσουμε τον επόμενο αριθμό */
        } /* τέλος else */

    } /* τέλος while */

    printf("\nSummary of the given numbers is: sum = %d\n", sum);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 4_Άθροισμα αγνώστου αριθμού θετικών, τερματίζει με το 0



ΔΡΔ

Κώδικας σε C

```
/* Παράδειγμα 4
   Άθροισμα N ακέραιων θετικών αριθμών, τερματισμός με 0 */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int sum = 0, num;

    printf("Enter a number (0 to end): ");
    scanf("%d", &num);

    /* το πρόγραμμα τερματίζει όταν ο χρήστης δώσει 0 */
    while (num != 0)
    {
        if (num > 0) /* έλεγχος για θετικούς */
        {
            sum += num; /* πρόσθεσε τον αριθμό στη sum */
            printf("Enter a number (0 to end): "); /* δώσε επόμενο
                                                    αριθμό */
            scanf("%d", &num);
        } /* τέλος if */

        else

        {
            printf("Only positive numbers are allowed!\n");

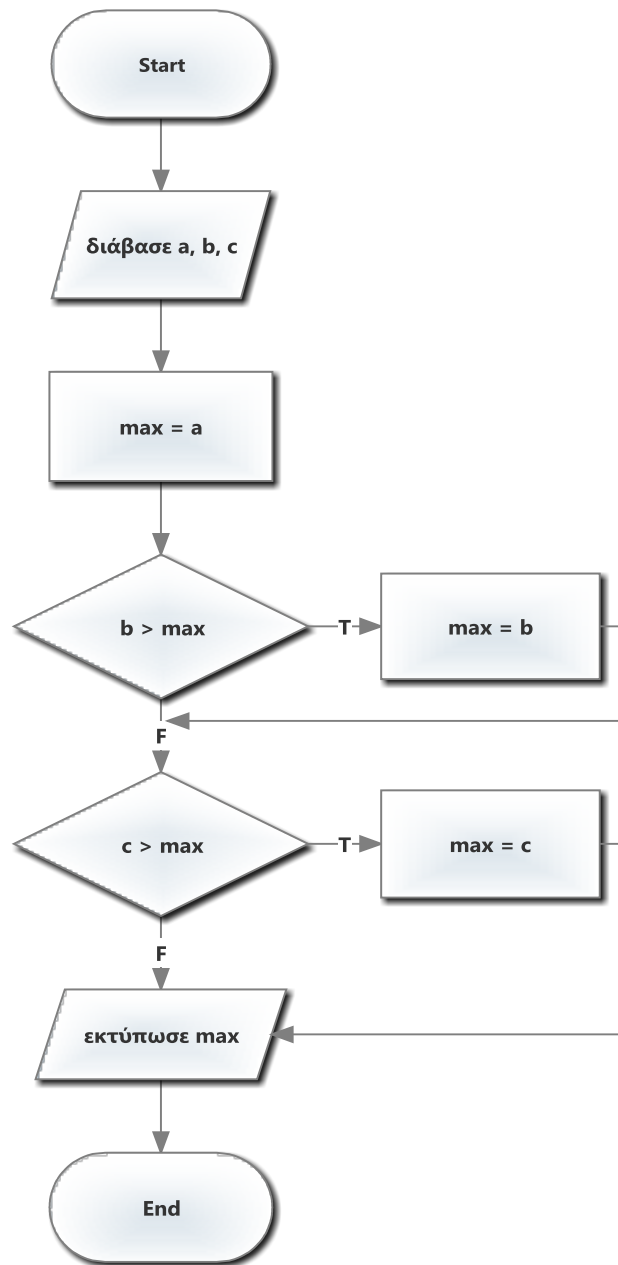
            printf("Enter a number (0 to end): ");
            scanf("%d", &num);
        } /* τέλος else */

    } /* τέλος while */

    printf("Summary of given numbers is: %d\n", sum);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 5_Εύρεση μεγαλύτερου από 3 αριθμούς



ΔΡΔ

Κώδικας σε C

```
/* Παράδειγμα 5
   Εύρεση μεγαλύτερου από 3 αριθμούς */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int a, b, c, max;

    printf("Enter number a: ");
    scanf("%d", &a);

    printf("Enter number b: ");
    scanf("%d", &b);

    printf("Enter number c: ");
    scanf("%d", &c);

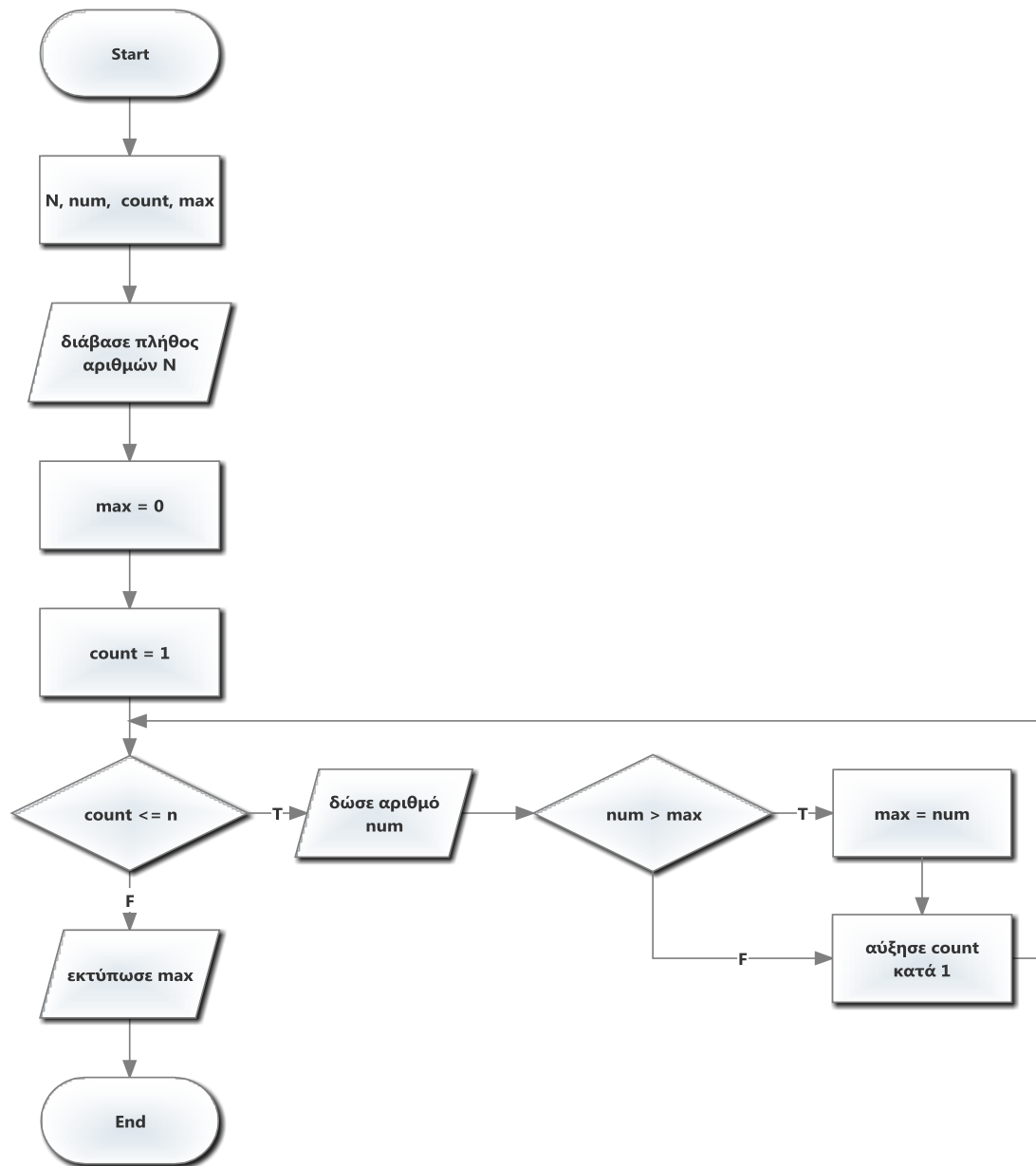
    max = a; /* ορίζουμε μέγιστο τον πρώτο */

    if(b > max)
        max = b;
    if (c > max)
        max = c;

    printf("\nThe largest number is: %d\n", max);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 6_Εύρεση μεγαλύτερου από σειρά αριθμών



ΔΡΔ

Κώδικας σε C (με 2 τρόπους)

```
/* Παράδειγμα 6α (με for)
   Εύρεση μεγαλύτερου από N αριθμούς */

#include <stdio.h>
/* έναρξη συνάρτησης main */
int main(void)
{
    int N, val, count, max;

    printf("How many numbers to enter: ");
    scanf("%d", &N);
    printf("Enter number 1: ");
    scanf("%d", &val);

    max = val; /* ορίζουμε μέγιστο τον πρώτο */

    /* εισαγωγή N ακέραιων αριθμών και εύρεση μεγίστου με for */
    for (count = 2; count <= N; count++)
    {
        printf("Enter the number %d: ", count);
        scanf("%d", &val);
        if(val > max)
            max = val;
    } /* τέλος for */

    printf("The Largest Number is: %d\n", max);
    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

```
/* Παράδειγμα 6β (με while)
   Εύρεση μεγαλύτερου από N αριθμούς */

#include <stdio.h>
/* έναρξη συνάρτησης main */
int main(void)
{
    int a, max;

    printf("Enter a number: ");
    scanf("%d",&a);

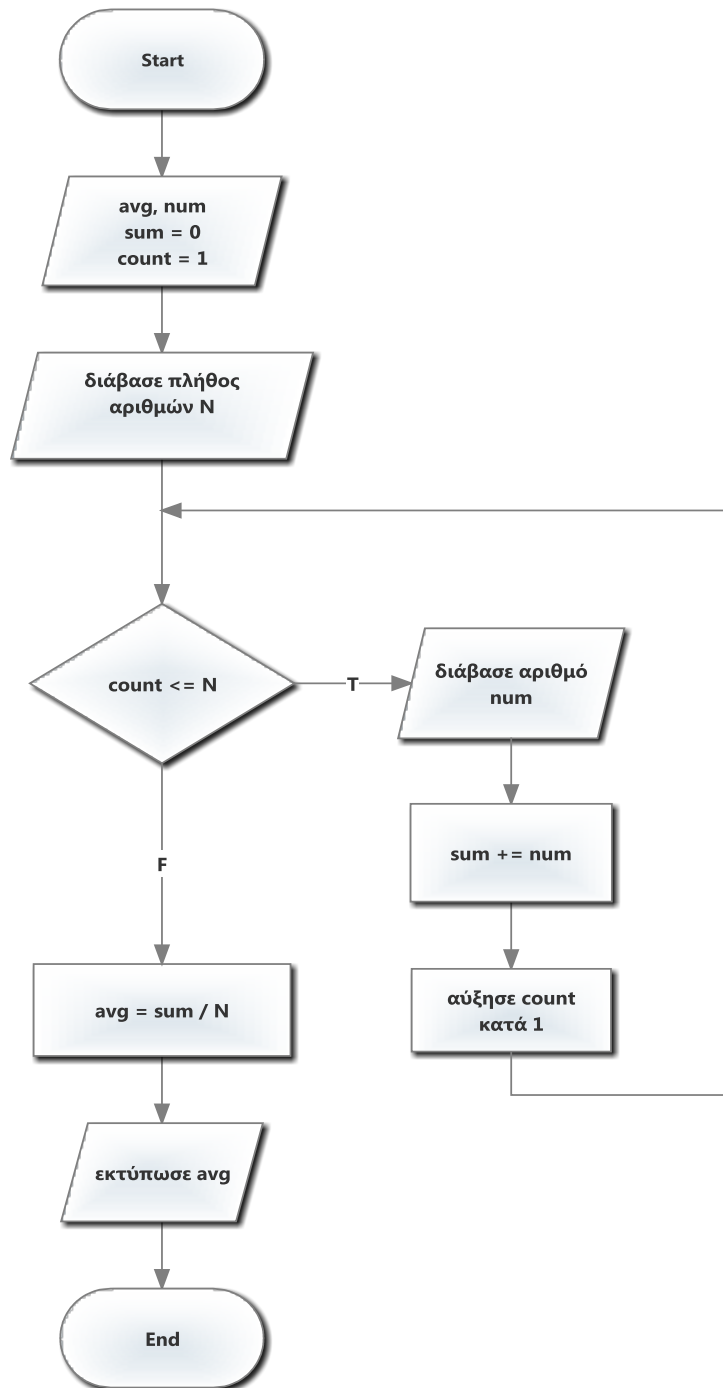
    max = a;

    /* εισαγωγή N ακέραιων θετικών αριθμών και εύρεση μεγίστου με while */
    while (a != -1)
    {
        printf("Enter new number, (-1 to end): ");
        scanf("%d",&a);
        if(a > max)
            max = a;
    } /* end while */

    printf("The largest number is: %d \n", max);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 7_Εύρεση μέσου όρου N αριθμών



ΔΡΔ

Κώδικας σε C

```
/* Παράδειγμα 7
   Εύρεση μέσου όρου N ακέραιων αριθμών */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int N, count;
    float sum = 0, num, avg; /* τύπου float για πραγματικούς */

    printf("Enter How Many Numbers: ");
    scanf("%d", &N);

    for (count = 1; count <= N; count++)
    {
        printf("Enter number %d: ", count);
        scanf("%f", &num);

        sum += num;
    } /* τέλος for */

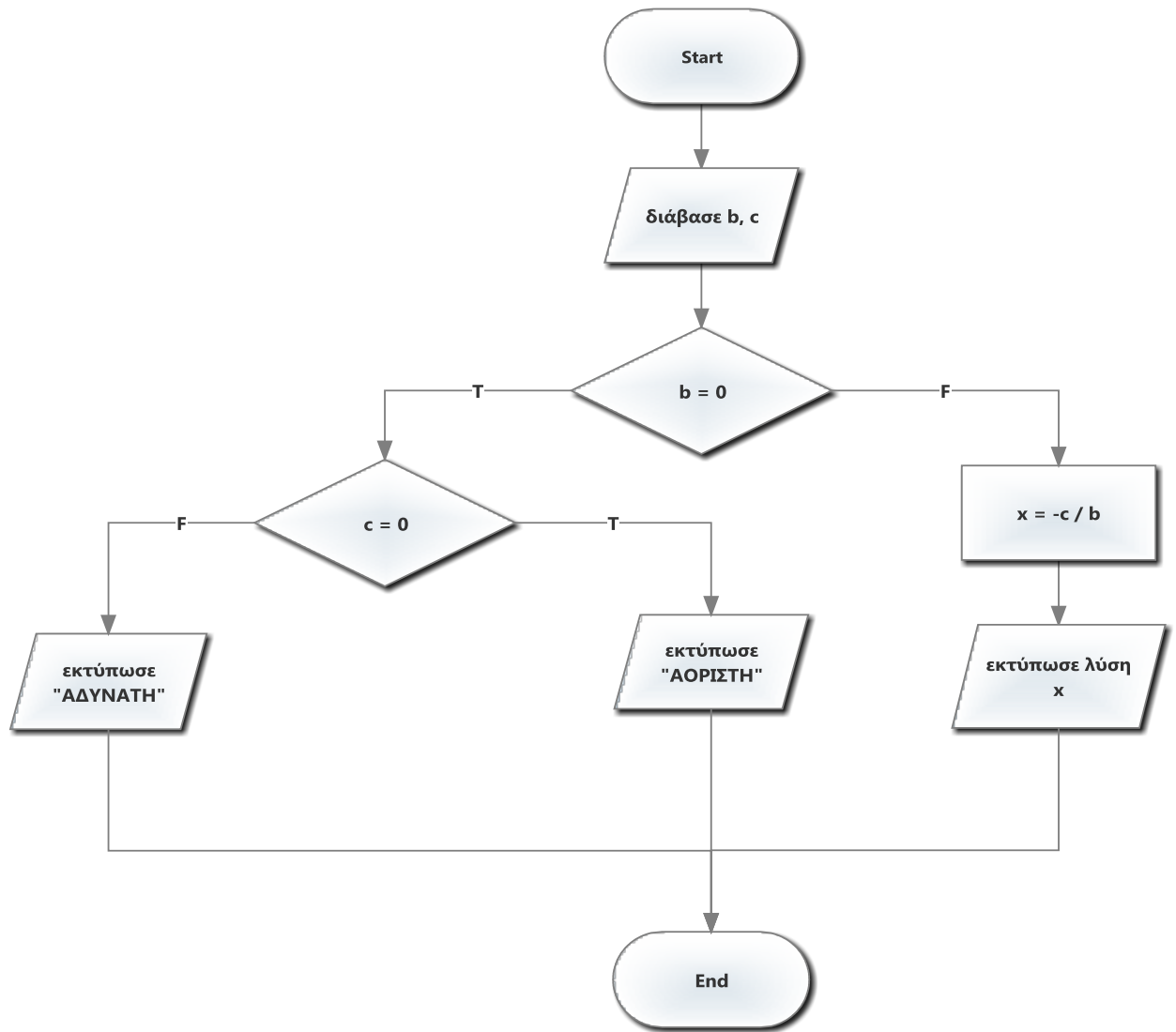
    avg = sum / N;

    printf("The Average of Numbers is : %.2f\n", avg);
    /* το προσδιοριστικό 0.2f θα τυπώσει τα 2 πρώτα δεκαδικά ψηφιά */

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 8_Επίλυση πρωτοβάθμιας εξίσωσης

$$y = bx + c$$



ΔΡΔ

Κώδικας σε C

```
/* Παράδειγμα 8
   Επίλυση Πρωτοβάθμιας Εξίσωσης */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    float x, c, b; /* τύπου float για πραγματικούς */

    printf("The primary equation is b*x + c = 0\n");

    printf("Enter number for b: ");
    scanf("%f", &b);

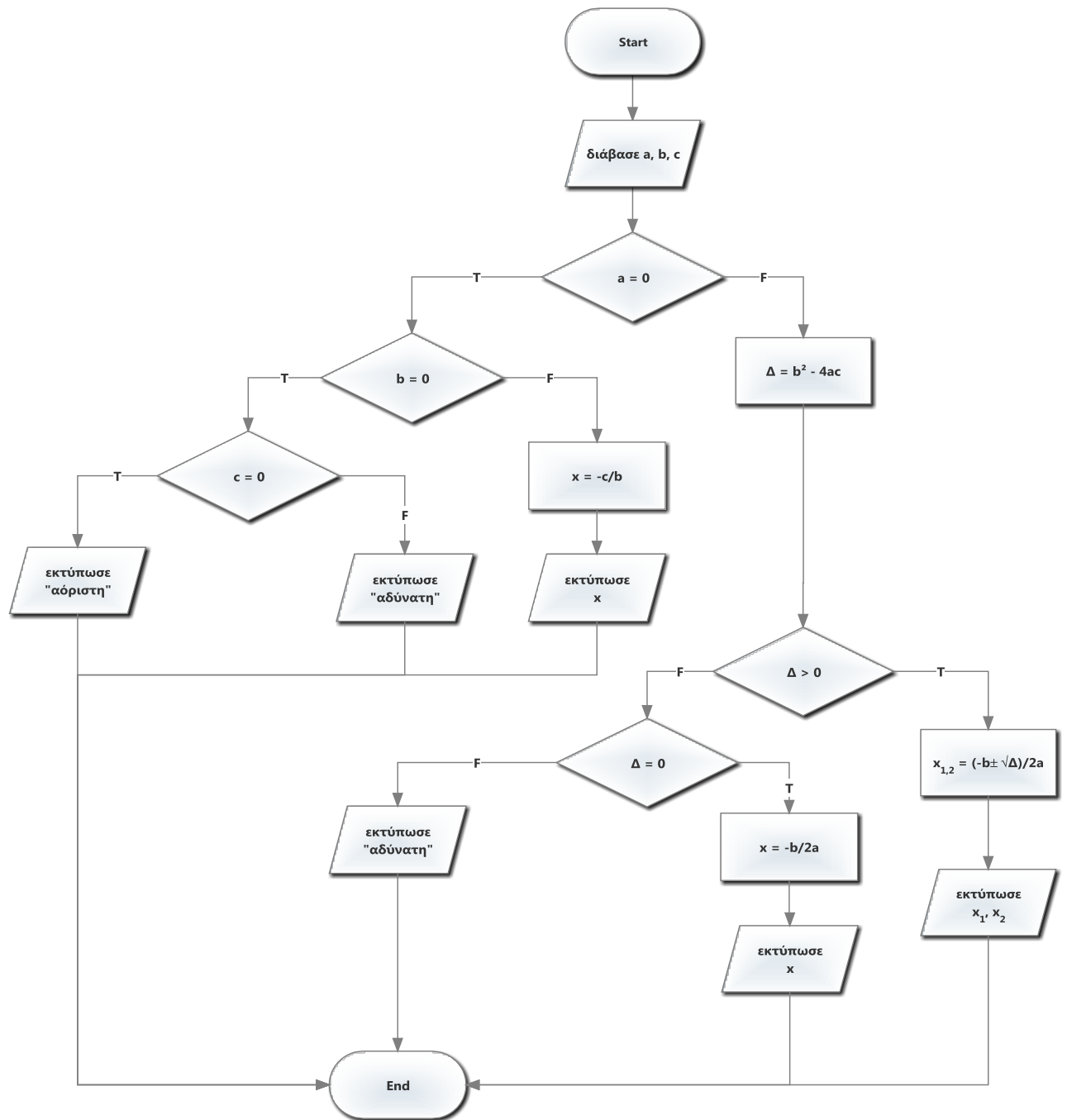
    printf("Enter number for c: ");
    scanf("%f", &c);

    if (b == 0)
    {
        if (c == 0)
            printf("Indeterminate equation: x can take any value\n");
        else
            printf("Equation is impossible: There in no solution\n");
    } /* τέλος if */

    else
    {
        x = -c / b;
        printf("The solution is: x = %.2f\n", x);
    } /* τέλος else */

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 9_Επίλυση δευτεροβάθμιας εξίσωσης
 $y = ax^2 + bx + c$



ΔΡΔ

Κώδικας σε C (με 2 τρόπους)

```
/* Παράδειγμα 9α
   Δευτεροβάθμια Εξίσωση */

#include <stdio.h>
#include <math.h> /* ενσωμάτωση της βιβλιοθήκης math */

/* έναρξη συνάρτησης main */
int main(void)
{
    /* μεταβλητές που θα χρησιμοποιηθούν στην εξίσωση */
    int a, b, c;
    float x, x1, x2, disc; /* x,x1,x2: οι λύσεις και disc: η Διακρίνουσα*/

    printf("The Quadratic equation is (a*x^2 + b*x + c = 0)\n");
    printf("Enter numbers for a, b, c (separated by commas): ");
    scanf("%d, %d, %d", &a, &b, &c);

    if(a == 0)
    {
        /* όμοια με Πρωτοβάθμια εξίσωση (b*x + c = 0)*/
        if(b == 0)
        {
            if(c == 0)
                printf("Indeterminate equation: x can take any\
value\n");
            else
                printf("Equation is impossible: There in no\
solution\n");
        } /* τέλος if b*/
        else
        {
            x = (-c) / b;
            printf("The solution is: x = %.2f\n", x);
        } /* τέλος else */
    } /* τέλος if a*/
    else
    {
        disc = (b*b) - (4*a*c); /* υπολογισμός Διακρίνουσας */
        if(disc > 0) /* ρίζες της Διακρίνουσας */
        {
            x1 = ((-b) + sqrt(disc)) / (2*a);
            x2 = ((-b) - sqrt(disc)) / (2*a);
            printf("There are two possible solutions: x1 = %.2f and\
x2 = %.2f\n", x1, x2);
        } /* τέλος if */
        else
        {
            if(disc == 0) /* Διακρίνουσα ίση με 0 */
            {
                x = (-b)/(2*a);
                printf("The solution is: x = %.2f\n", x);
            }
            else /* Διακρίνουσα < 0 */
                printf("Equation is impossible: There in no\
solution\n");
        } /* τέλος else */
    } /* τέλος εξωτερικού else */

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

```

/* Παράδειγμα 9β
   Δευτεροβάθμια εξίσωση - Υπολογισμός Ριζών με Διακρίνουσα */

#include <stdio.h>
#include <math.h> /* ενσωμάτωση της βιβλιοθήκης math */

/* έναρξη συνάρτησης main */
int main(void)
{
    int a, b, c;
    float x1, x2, disc; /* x1,x2: οι λύσεις και disc: η Διακρίνουσα*/

    printf("Enter value of a : ");
    scanf("%d", &a);
    printf("Enter value of b : ");
    scanf("%d", &b);
    printf("Enter value of c : ");
    scanf("%d", &c);

    disc=(b * b) - (4 * a * c); /* υπολογισμός Διακρίνουσας */

    if(disc > 0)
    {
        printf("\n ROOTS ARE REAL ROOTS\n");
        x1 = ((-b) + sqrt(disc)) / (2*a); /* χρήση της συνάρτησης math */
        x2 = ((-b) - sqrt(disc)) / (2*a);
        printf("\n ROOTS : %f, %f\n", x1, x2);
    } /* τέλος if */

    else if(disc == 0)
    {
        printf("\n ROOTS ARE EQUAL\n");
        x1 = (-b/(2*a));
        printf("\n ROOT IS...: %f\n", x1);
    } /* τέλος else if */

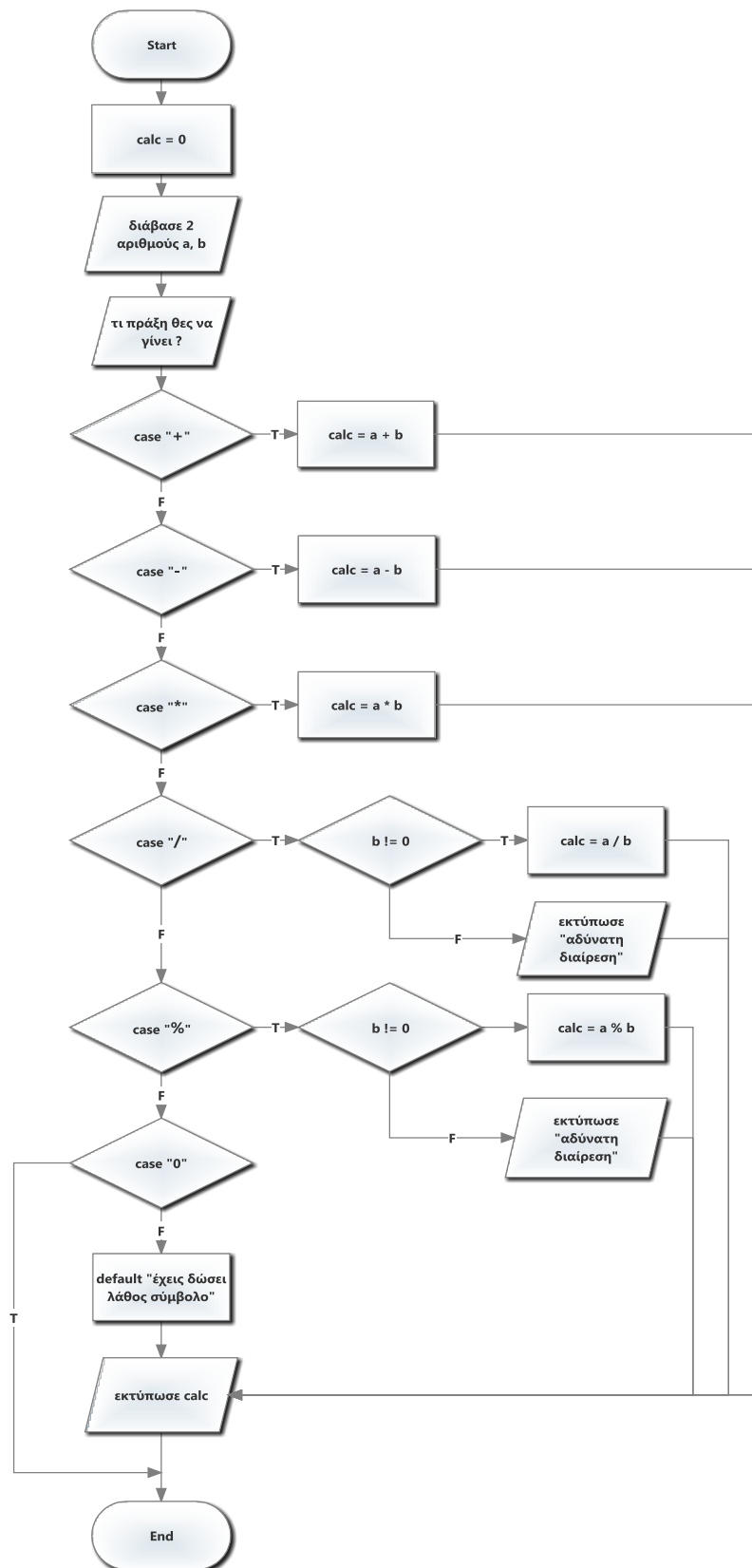
    else
        printf("\n ROOTS ARE IMAGINARY ROOTS\n");

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */

```

Παράδειγμα 10_Μενού επιλογών, τερματισμός με 0

άθροισμα, αφαίρεση, πολλαπλασιασμός, διαίρεση, υπόλοιπο ακέραιας διαίρεσης



Κώδικας σε C

```
/* Παράδειγμα 10
   Υπολογιστής τσέπης, με χρήση switch */

#include <stdio.h>
#include <stdlib.h> /* ενσωμάτωση της βιβλιοθήκης stdlib */

/* έναρξη συνάρτησης main */
int main(void)
{
    int a,b,res,ch;

    /* Τελεστές +,-,*, /, % */
    printf("\n\t*****");
    printf("\n\tMENU\n");
    printf("\n\t*****");
    printf("\n\t(1)ADDITION"); /* επιλογή για πρόσθεση */
    printf("\n\t(2)SUBTRACTION"); /* επιλογή για αφαίρεση */
    printf("\n\t(3)MULTIPLICATION"); /* επιλογή για πολλαπλασιασμό */
    printf("\n\t(4)DIVISION"); /* επιλογή για διαίρεση */
    printf("\n\t(5)REMAINDER"); /* επιλογή για υπόλοιπο ακέραιας διαίρεσης */
    printf("\n\t(0)EXIT"); /* επιλογή για τερματισμό */
    printf("\n\t*****");
    printf("\n\n\tEnter your choice: ");
    scanf("%d",&ch);

    if(ch <= 5 && ch > 0)
    {
        printf("\n\tEnter value for number 1: ");
        scanf("%d", &a);
        printf("\tEnter value for number 2: ");
        scanf("%d", &b);
    }
    switch(ch)
    {
        case 1:
            res= a + b;
            printf("\n\t Addition: %d\n",res);
            break;
        case 2:
            res= a - b;
            printf("\n\t Subtraction: %d\n",res);
            break;
        case 3:
            res= a * b;
            printf("\n\t Multiplication: %d\n",res);
            break;
        case 4:
            res= a / b; /* υπόθεση ότι b != 0 */
            printf("\n\t Division: %d\n",res);
            break;
        case 5:
            res= a % b; /* υπόθεση ότι b != 0 */
            printf("\n\t Remainder: %d\n",res);
            break;
        case 0:
            printf("\n\t Choice Terminated\n");
            exit(1); /* χρήση της συνάρτησης exit της βιβλιοθήκης
                       stdlib */
            break;
    }
}
```



```
    default:
        printf("\n\t Invalid Choice\n");
} /* end switch */

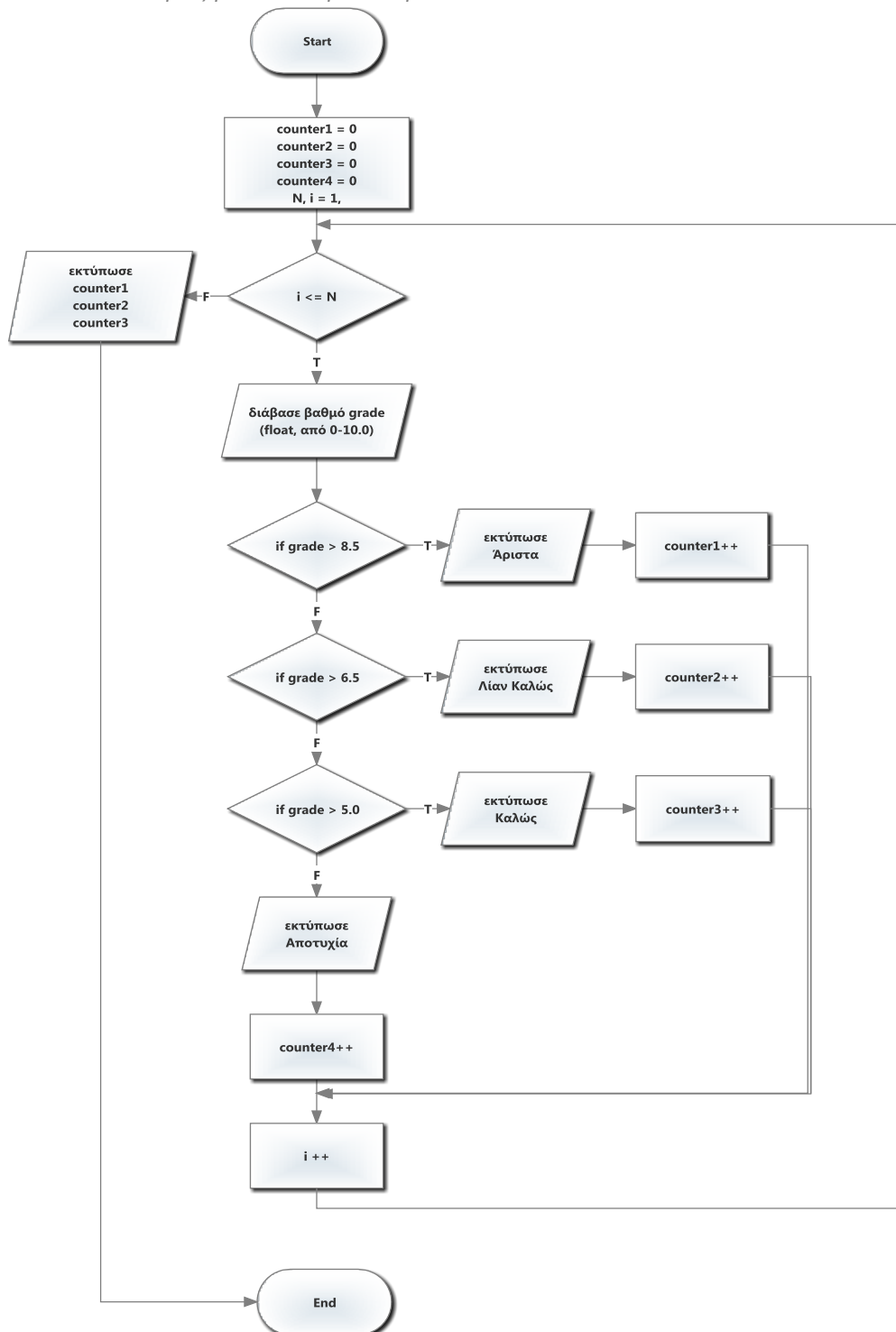
return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 11_Διαβάζει βαθμολογία και τυπώνει αντίστοιχο αποτέλεσμα

εισάγει τον βαθμό μαθήματος (δεκαδικός αριθμός μεταξύ του 0.0 και του 10.0) N φοιτητών και χαρακτηρίζει ως εξής:

- a) "Άριστα", εάν ο βαθμός είναι μεγαλύτερος ή ίσος του 8.50
- b) "Λίαν Καλώς", εάν ο βαθμός είναι μικρότερος του 8.50 και μεγαλύτερος ή ίσος του 6.50
- c) "Καλώς", εάν ο βαθμός είναι μικρότερος του 6.50 και μεγαλύτερος ή ίσος του 5.00 και
- d) "Αποτυχία", εάν ο βαθμός είναι μικρότερος του 5.00.

Να τυπώνεται το πλήθος για κάθε περίπτωση.



```

/* Παράδειγμα 11
   Χαρακτηρισμός Βαθμών */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    float grade;
    int counter1=0; /* μετρητής για "Άριστα" */
    int counter2=0; /* μετρητής για "Λίαν Καλώς" */
    int counter3=0; /* μετρητής για "Καλώς" */
    int counter4=0; /* μετρητής για "Αποτυχία" */
    int i, N;

    printf("Enter # of students: ");
    scanf("%d", &N);

    for (i=1; i<=N; i++)
    {
        printf("Give me a grade for student %d (values between 0.00 and
10.00): ", i);
        scanf("%f",&grade);
        if (grade > 8.50)
        {
            printf("ARISTA\n");
            counter1++;
        } /* τέλος if */

        else if (grade > 6.50)
        {
            printf("LIAN KALWS\n");
            counter2++;
        } /* τέλος else if */

        else if (grade > 5.00)
        {
            printf("KALWS\n");
            counter3++;
        } /* τέλος else if */

        else
        {
            printf("APOTYXIA\n");
            counter4++;
        } /* τέλος else */
    } /* τέλος for */

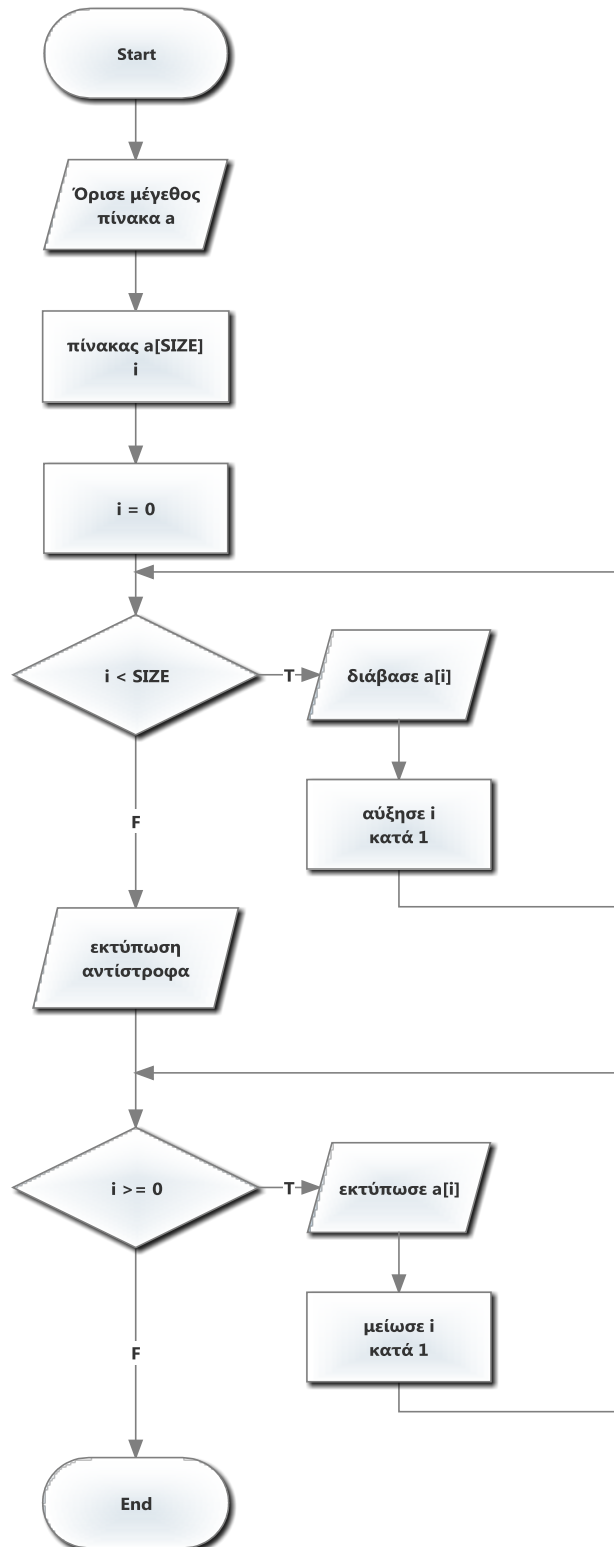
    printf("*****\n");
    printf("%d students phran KALWS \n",counter1);
    printf("%d students phran LIAN KALWS \n",counter2);
    printf("%d students phran ARISTA \n",counter3);
    printf("%d students failed \n",counter4);
    printf("*****\n");

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */

```

Παράδειγμα 12_Διάβασμα μονοδιάστατου πίνακα

γέμισμα μονοδιάστατου πίνακα ακεραίων και εκτύπωση αντίστροφα



Κώδικας σε C

```
/* Παράδειγμα 12
   Απόδοση τιμών σε πίνακα από πληκτρολόγιο και εκτύπωση αντίστροφα */

#include <stdio.h>
#define SIZE 10 /* δήλωση συμβολικής σταθεράς για μέγιστο μέγεθος πίνακα */

/* έναρξη συνάρτησης main */
int main(void)
{
    int a[SIZE]; /* δήλωση πίνακα a ακεραίων με SIZE στοιχεία */
    int i; /* μετρητής για τις επαναλήψεις στον πίνακα*/

    /* ορισμός τιμών */
    for (i = 0; i < SIZE; i++)
    {
        printf("Enter value for the %d element of the array: ", i);
        scanf("%d", &a[i]);
    } /* τέλος for */

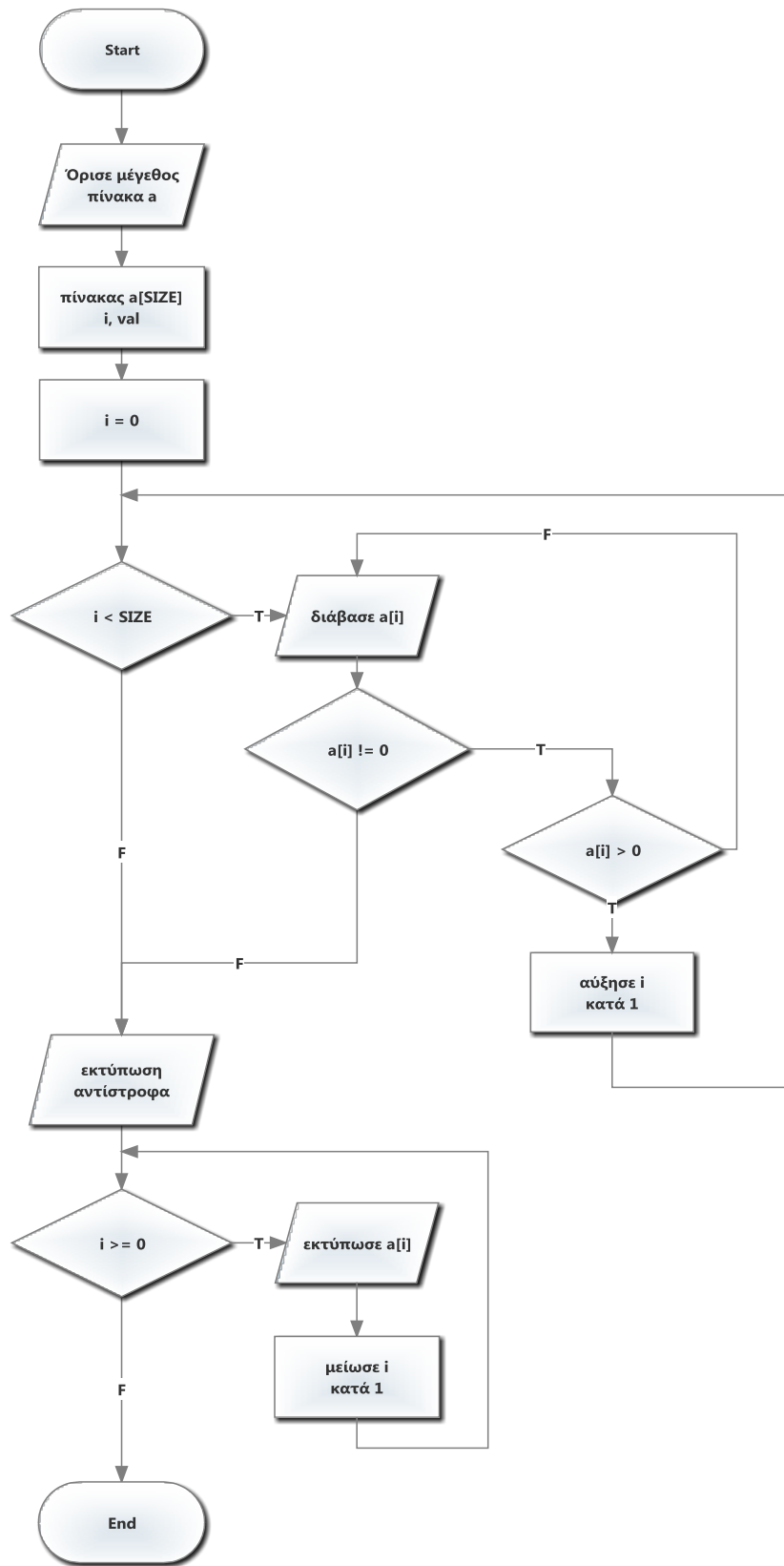
    /* εκτύπωση συμβολοσειρών για καλύτερη επισκόπηση πίνακα */
    printf("\nReverse print of the array\n");
    printf("%s%13s\n", "Element", "Value");

    /* εκτύπωση αντίστροφα */
    for(i = SIZE-1; i >= 0; i--)
        printf("%7d%13d\n", i, a[i]);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 13_Διάβασμα μονοδιάστατου πίνακα θετικών

γέμισμα μονοδιάστατου πίνακα με θετικούς, τερματίζει στο 0 ή στο μέγιστο πλήθος και εκτύπωση αντίστροφα.



Κώδικας σε C

```
/* Παράδειγμα 13
   Αρχικοποίηση στοιχείων πίνακα με θετικούς και εκτύπωση αντίστροφα */

#include <stdio.h>
#define SIZE 10 /* μέγιστο μέγεθος πίνακα */

/* έναρξη συνάρτησης main */
int main(void)
{
    int a[SIZE]; /* δήλωση πίνακα a ακεραίων με SIZE στοιχεία */
    int i; /* μετρητής για τις επαναλήψεις στον πίνακα*/

    /* ορισμός τιμών */
    for (i = 0; i < SIZE; i++)
    {
        printf("Enter value for the %d element of the array: ", i);
        scanf("%d", &a[i]);

        if(a[i] < 0)
        {
            printf("You entered a negative value, re-enter\n");
            i -=1;
        } /* end if */
        else if (a[i] == 0) /* όταν δοθεί 0 τερματίζει */
            break;
    } /* τέλος for */

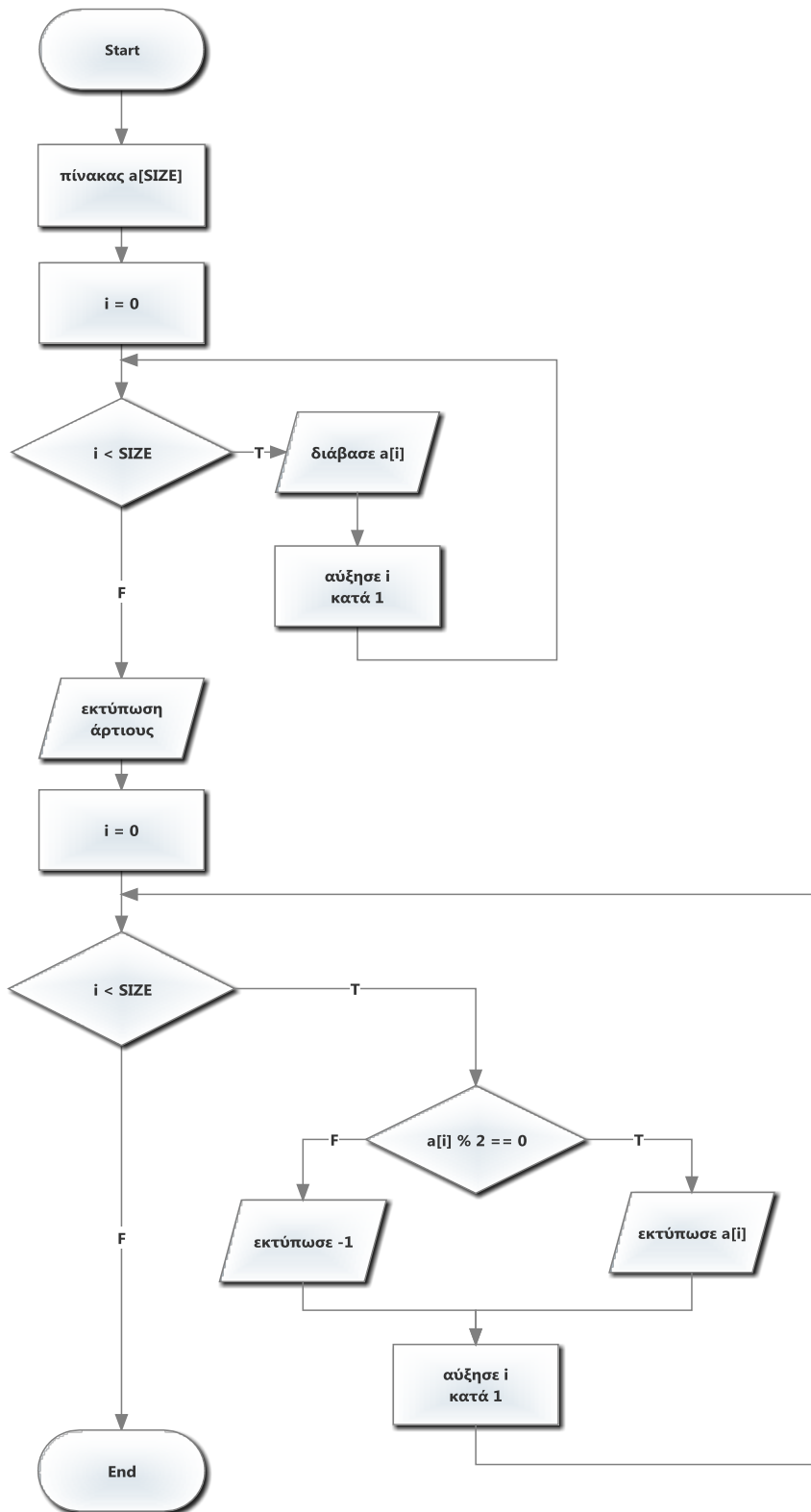
    /* εκτύπωση συμβολοσειρών για καλύτερη επισκόπηση πίνακα */
    printf("\nReverse print of the array\n");
    printf("%s%13s\n", "Element", "Value");

    /* εκτύπωση αντίστροφα */
    for (i = SIZE-1; i >= 0; i--)
    {
        printf("%7d%13d\n", i, a[i]);
    } /* τέλος for */

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 14_Μονοδιάστατος πίνακας, εκτύπωση άρτιων

γέμισμα μονοδιάστατου πίνακα 10 θέσεων ακεραίων έλεγχος για άρτιους και περιττούς και εκτύπωση των άρτιων τιμών αλλιώς -1.



Κώδικας σε C

```
/* Παράδειγμα 14
   Γέμισμα πίνακα και εκτύπωση των άρτιων αλλιώς εκτύπωση -1*/

#include <stdio.h>
#define SIZE 10 /* μέγιστο μέγεθος πίνακα */

/* έναρξη συνάρτησης main */
int main(void)
{
    int a[SIZE]; /* δήλωση πίνακα a ακεραίων με SIZE στοιχεία */
    int i; /* για τις επαναλήψεις στον πίνακα*/
    int evens = 0, odds = 0; /* για τις επαναλήψεις στον πίνακα*/

    /* ορισμός τιμών πίνακα */
    for (i = 0; i < SIZE; i++)
    {
        printf("Enter value for the %d element of the array: ", i);
        scanf("%d", &a[i]);

        if (a[i] % 2 == 0) /* έλεγχος για άρτιο */
        {
            printf("The element is even number\n");
            a[i] = a[i]; /* όταν άρτιος τύπωσε την τιμή του */
        } /* τέλος if */

        else
        {
            printf("The element is odd number (replace with -1)\n");
            a[i] = -1; /* όταν περιττός τύπωσε -1 */
        } /* τέλος else */

    } /* τέλος for */

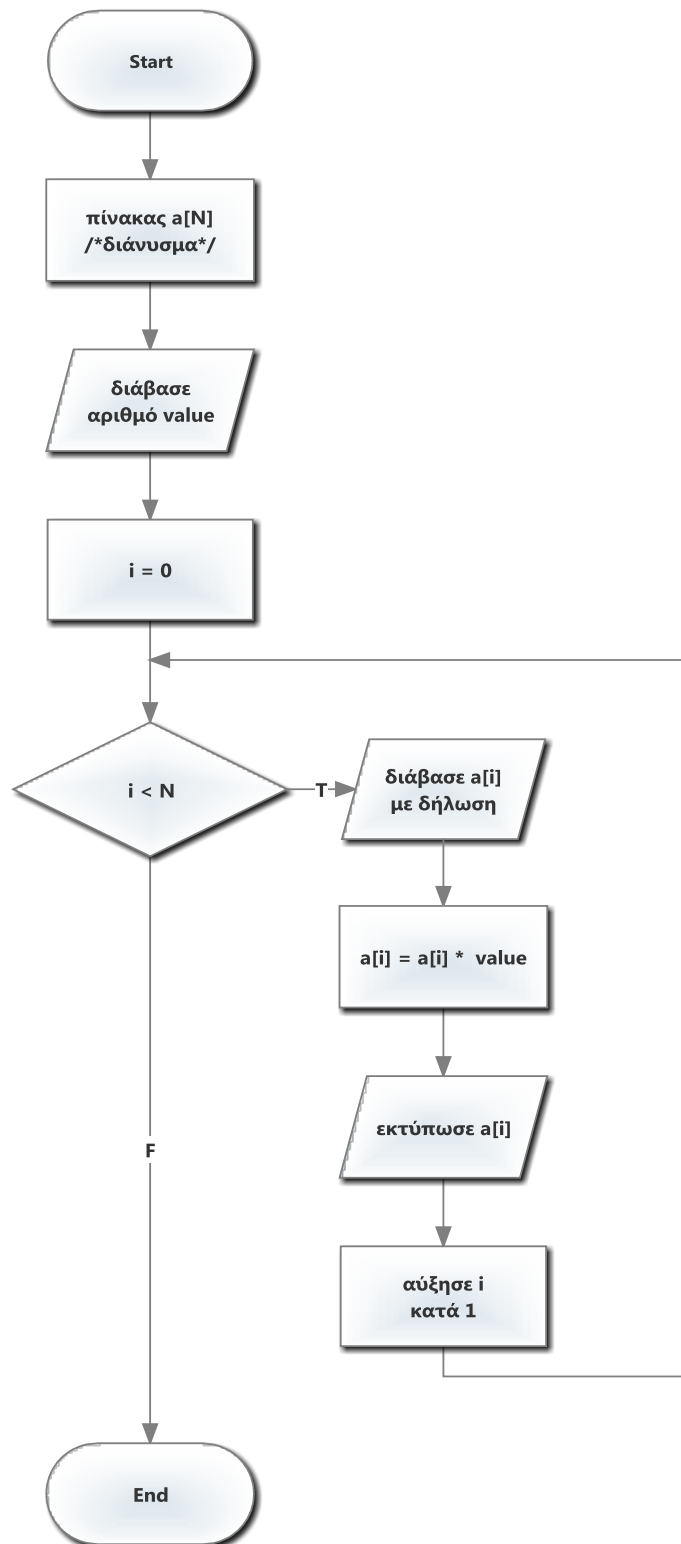
    /* εκτύπωση συμβολοσειρών για καλύτερη επισκόπηση πίνακα */
    printf("%s%13s\n", "Element", "Value");

    /* εκτύπωση πίνακα */
    for(i = 0; i < SIZE; i++)
        printf("%7d%13d\n", i, a[i]);

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Παράδειγμα 15_Πολλαπλασιασμός ακεραίου με διάνυσμα

απόδοση τιμών σε πίνακα από ορισμό και πολλαπλασιασμός ακεραίου αριθμού με μονοδιάστατο πίνακα και εκτύπωση



Κώδικας σε C

```
/* Παράδειγμα 15
   Πολλαπλασιασμός αριθμού με διάνυσμα */

#include <stdio.h>

/* έναρξη συνάρτησης main */
int main(void)
{
    int value, i;

    /* απόδοση αρχικών τιμών σε πίνακα με μια δήλωση */
    int a[5] = {32, 24, 18, 9, 77};

    printf("Give a number: ");
    scanf("%d", &value);

    printf("%s%11s\n", "Element", "Vector");

    /* πολλαπλασιασμός στοιχείων πίνακα με αριθμό */
    for (i = 0; i < 5; i++)
    {
        printf("%d", a[i]); /* εκτύπωση αρχικού πίνακα */
        a[i] = a[i]*value;

        printf("%13d\n", a[i]); /* εκτύπωση τελικού πίνακα */
    }

    return 0; /* για επιτυχή τερματισμό */
} /* τέλος συνάρτησης main */
```

Ασκήσεις για εξάσκηση

1. Διάβασμα δισδιάστατου πίνακα
διάβασμα δισδιάστατου πίνακα ακεραίων $M \times M$ και εκτύπωση της κύριας διαγωνίου ($i=j$).
2. Πρόσθεση δισδιάστατου πίνακα
πρόσθεση των στοιχείων δισδιάστατου πίνακα $M \times N$ και εκτύπωση πίνακα.
3. Πολλαπλασιασμός δισδιάστατου πίνακα
πολλαπλασιασμός των στοιχείων δισδιάστατου πίνακα $M \times N$ ανά γραμμή.
4. Παιχνίδι ναυμαχία
γέμισμα δισδιάστατου πίνακα $M \times M$ με 0 και 1 και να βρίσκει και να μετράει όλα τα 1.
5. Χ φοιτητές Υ μαθήματα Μ.Ο μαθημάτων, Μ.Ο φοιτητή
να δίνει τους βαθμούς Υ μαθημάτων που έχουν πάρει Χ φοιτητές και να υπολογίζει το μέσο όρο του κάθε φοιτητή σε όλα τα μαθήματα καθώς και το μέσο όρο του κάθε μαθήματος από όλους τους φοιτητές.