

Εισαγωγή στον Προγραμματισμό

Δομές Επανάληψης
Για (for)

Φροντιστήριο

Αριστείδης Ηλίας

ΓΙΑ <μεταβλητή> ΑΠΟ <αρχική τιμή> ΜΕΧΡΙ <τελική τιμή> ΜΕ_ΒΗΜΑ <μέγεθος βήματος>
<εντολή> ή <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

στη C

```
for(αρχική τιμή μεταβλητής ελέγχου; τελική τιμή μεταβλητής ελέγχου; βήμα μεταβολής ελέγχου){  
    <εντολή> ή <εντολές>  
}
```

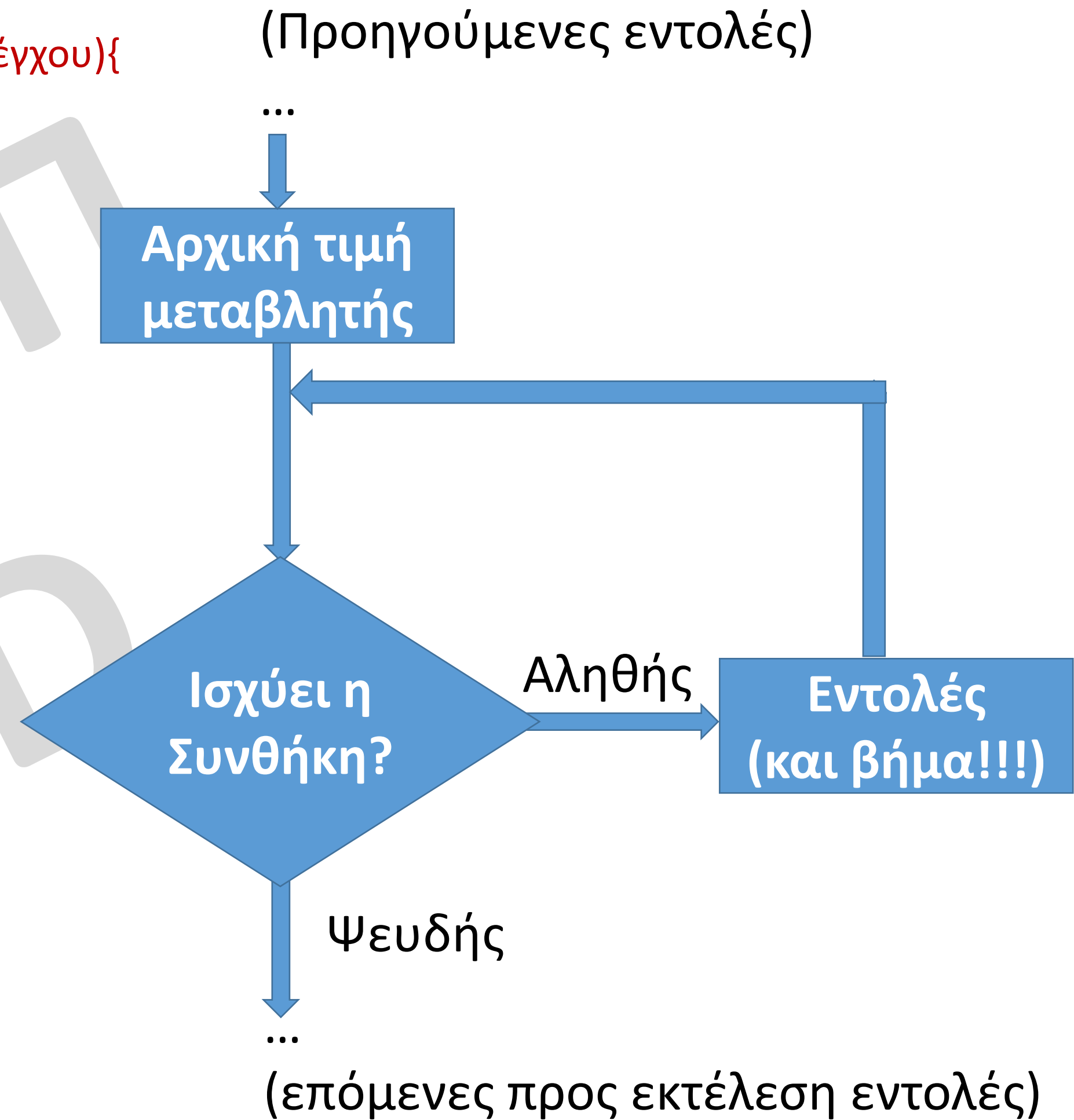
Η εντολή αυτή χειρίζεται μια <μεταβλητή> στην οποία καταχωρείται αρχικά η <αρχική τιμή>. Η τιμή που παίρνει η <μεταβλητή> συγκρίνεται με τη <τελική τιμή> και εφόσον είναι μικρότερη απ' αυτήν, τότε εκτελούνται οι εντολές που βρίσκονται στο βρόχο. Στη συνέχεια η μεταβλητή αυξάνεται κατά την τιμή <μέγεθος βήματος> που ορίζεται στο βήμα. Αν η νέα τιμή είναι μικρότερη της τελικής, τότε ο βρόχος (επανάληψη) εκτελείται ξανά. Η διαδικασία επαναλαμβάνεται μέχρι η <μεταβλητή> να γίνει μεγαλύτερη από την <τελική τιμή>, αλλιώς ο αλγόριθμος (ή το πρόγραμμα) συνεχίζει με την εντολή μετά το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ.

...Δομές Επανάληψης... ...Για (for)...

ΓΙΑ <μεταβλητή> ΑΠΟ <αρχική τιμή> ΜΕΧΡΙ <τελική τιμή> ΜΕ_ΒΗΜΑ <μέγεθος βήματος>
<εντολή> ή <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

στη C

```
for(αρχική τιμή μεταβλητής ελέγχου; τελική τιμή μεταβλητής ελέγχου; βήμα μεταβολής μεταβλητής ελέγχου){  
    <εντολή> ή <εντολές>  
}
```



...Δομές Επανάληψης...

...Για (for)...

ΓΙΑ <μεταβλητή> ΑΠΟ <αρχική τιμή> ΜΕΧΡΙ <τελική τιμή> ΜΕ_ΒΗΜΑ <μέγεθος βήματος>
<εντολή> ή <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

στη C

```
for(αρχική τιμή μεταβλητής ελέγχου; τελική τιμή μεταβλητής ελέγχου; βήμα μεταβολής ελέγχου){  
    <εντολή> ή <εντολές>  
}
```

Ο αριθμός του βήματος μπορεί να είναι θετικός ή αρνητικός και αυτό καθορίζει τον έλεγχο για την επανάληψη.
Έτσι έχουμε τις περιπτώσεις:

1. Βήμα = 1 : παραλείπεται το ΜΕ_ΒΗΜΑ 1 και η μεταβλητή αυξάνεται κατά +1.
2. Βήμα > 0 : τότε πρέπει να ισχύει $\text{τιμή1} \leq \text{τιμή2}$ για να εκτελεστεί τουλάχιστον μια φορά ο βρόχος.
3. Βήμα < 0 : τότε πρέπει να ισχύει $\text{τιμή1} \geq \text{τιμή2}$ για να εκτελεστεί τουλάχιστον μια φορά ο βρόχος.
4. Βήμα = 0 : τότε ο βρόχος είναι ατέρμων.

...Δομές Επανάληψης...

...Για (for)...

ΓΙΑ <μεταβλητή> ΑΠΟ <αρχική τιμή> ΜΕΧΡΙ <τελική τιμή> ΜΕ_ΒΗΜΑ <μέγεθος βήματος>
<εντολή> ή <εντολές>
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

στη C

```
for(αρχική τιμή μεταβλητής ελέγχου; τελική τιμή μεταβλητής ελέγχου; βήμα μεταβολής ελέγχου){  
    <εντολή> ή <εντολές>  
}
```

Παρατηρήσεις

1. Μέσα στην εντολή ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ... δεν επιτρέπεται να αλλάζετε την τιμή της <μεταβλητής> του ΓΙΑ, γιατί τότε δεν λειτουργεί σωστά η εντολή επανάληψης.
2. Το βήμα δεν είναι απαραίτητο να είναι ακέραιος αριθμός. Μπορεί να πάρει οποιαδήποτε πραγματική τιμή.
3. Παρατηρούμε ότι στο διάγραμμα ροής δεν υπάρχει ειδικό σύμβολο για τη δομή Για. Η υλοποίησή της απαιτεί τη χρήση μιας σειράς εντολών. Αυτές είναι:
 - a. Αρχικοποίηση της μεταβλητής (π.χ. εντολή εκχώρησης $i \leftarrow 1$)
 - b. Συνθήκη ελέγχου τερματισμού (π.χ. $i \leq 100$)
 - c. Εντολή εκχώρησης μεταβολής (π.χ. $i \leftarrow i + 1$)

...Δομές Επανάληψης...

...Για (for)...

Σε 10 σχολεία της περιφέρειας έχουν εγκατασταθεί πειραματικά 10 ηλεκτρονικοί υπολογιστές (εξυπηρετητές) που περιέχουν πληροφοριακές «σελίδες» του Internet και μπορεί να προσπελάσει κανείς την πληροφορία τους μέσα από οποιοδήποτε ηλεκτρονικό υπολογιστή στον κόσμο. Να γραφτεί ένας αλγόριθμος που θα διαβάσει το συνολικό αριθμό των προσπελάσεων που πραγματοποιήθηκε σε κάθε έναν από τους εξυπηρετητές αυτούς για διάστημα μίας ημέρας. Να βρεθεί ο εξυπηρετητής με το μικρότερο αριθμό προσπελάσεων καθώς και ο εξυπηρετητής με το μεγαλύτερο αριθμό προσπελάσεων

Αλγόριθμος Εξυπηρετητής

Διάβασε προσπελάσεις

ελάχιστος_εξ ← 1

ελάχιστος ← προσπελάσεις

μέγιστος_εξ ← 1

μέγιστος ← προσπελάσεις

Για i από 2 μέχρι 10

Διάβασε προσπελάσεις

Αν προσπελάσεις < ελάχιστος **τότε**

ελάχιστος ← προσπελάσεις

ελάχιστος_εξ ← i

Τέλος_Αν

Αν προσπελάσεις > μέγιστος **τότε**

μέγιστος ← προσπελάσεις

μέγιστος_εξ ← i

Τέλος Αν

Τέλος_Επανάληψης

Εμφάνισε ελάχιστος_εξ, μέγιστος_εξ

Τέλος Εξυπηρετητής

Αρχικά μέγιστος/ ελάχιστος είναι η πρώτη τιμή που διαβάζεται

Έχει σημασία που τοποθετούμε την εκτύπωση για ορθότητα στο ζητούμενο αποτέλεσμα

...Δομές Επανάληψης...

...Για (for)...

Να αναπτύξετε αλγόριθμο ο οποίος θα υπολογίζει και θα εκτυπώνει το άθροισμα των διψήφιων περιττών αριθμών, δηλαδή το άθροισμα $11 + 13 + 15 + \dots + 95 + 97 + 99$

Αλγόριθμος Άθροισμα_Περιττών

$S \leftarrow 0$

Για i **από** 11 **μέχρι** 99 **με_βήμα** 2

$S \leftarrow S + i$

Τέλος_Επανάληψης

Εκτύπωσε "Το άθροισμα είναι ", S

Τέλος Άθροισμα_Περιττών

...Δομές Επανάληψης...

...Για (for)...

Να αναπτύξετε αλγόριθμο ο οποίος θα εντοπίζει και θα εκτυπώνει όλους τους τετραψήφιους αριθμούς που μπορούν να διαβαστούν και ανάποδα (καρκινική ιδιότητα για παράδειγμα 1331, 7447, 9229 κ.ο.κ.).

Αλγόριθμος Καρκινικοί

Για X **από** 1000 **μέχρι** 9999

χιλιάδες $\leftarrow X \text{ div } 1000$

βοηθητική $\leftarrow X \text{ mod } 1000$

εκατοντάδες $\leftarrow \text{βοηθητική} \text{ div } 100$

βοηθητική $\leftarrow \text{βοηθητική} \text{ mod } 100$

δεκάδες $\leftarrow \text{βοηθητική} \text{ div } 10$

μονάδες $\leftarrow \text{βοηθητική} \text{ mod } 10$

Αν (χιλιάδες=μονάδες) **ΚΑΙ** (εκατοντάδες=δεκάδες) **τότε**

Εκτύπωσε "Ο αριθμός", X, " ικανοποιεί την εκφώνηση"

Τέλος_Αν

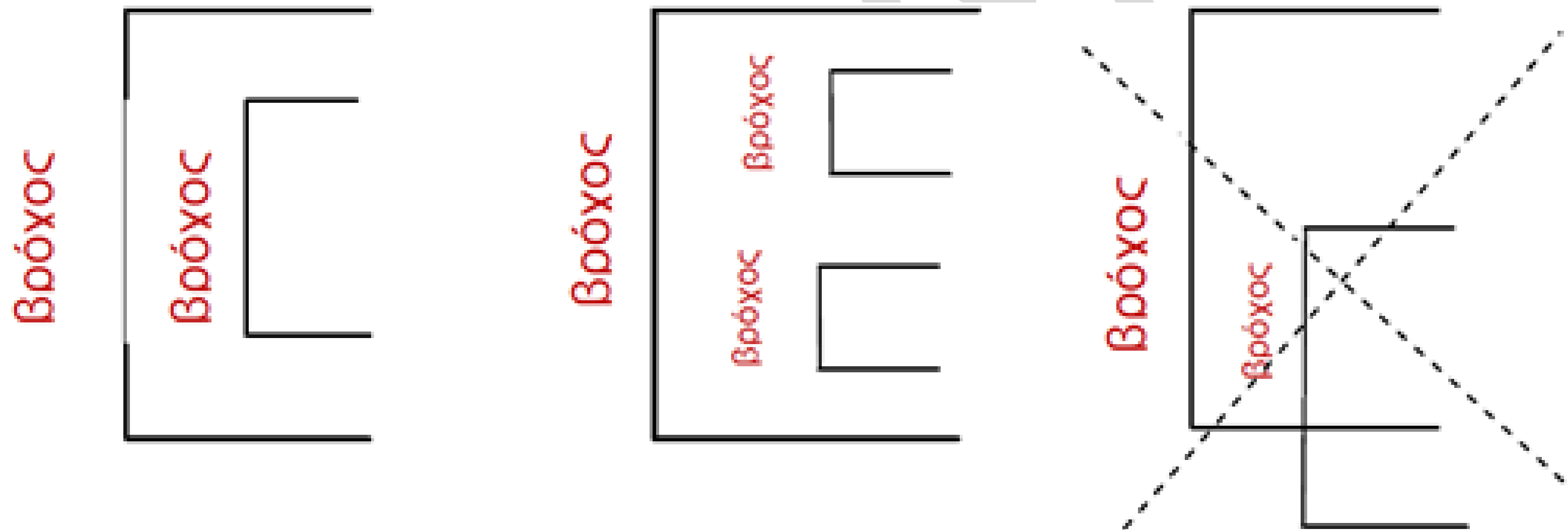
Τέλος_Επανάληψης

Τέλος Καρκινικοί

Μπορούμε να χρησιμοποιήσουμε την ίδια μεταβλητή, γιατί δεν μας ενδιαφέρει να κρατήσουμε τις προγενέστερες τιμές και μας το επιτρέπει η... εμβέλεια

...Δομές Επανάληψης...
Εμφωλευμένοι Βρόχοι [Για (for)]...

- Στην περίπτωση αυτή ο ένας βρόχος βρίσκεται μέσα στον άλλο. Για τη σωστή λειτουργία των εμφωλευμένων βρόχων πρέπει να ισχύουν συγκεκριμένοι κανόνες:
 - Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό
 - Οι επαναλήψεις του εσωτερικού βρόγχου πρέπει να ολοκληρώνονται για όλες τις τιμές της μεταβλητής του εσωτερικού και έπειτα να μεταβάλλεται το βήμα της εξωτερικής και πάλι να επαναλαμβάνεται ο εσωτερικός βρόγχος για όλες τις τιμές της εσωτερικής και στο τέλος να μεταβάλλεται πάλι το βήμα της εξωτερικής.



- Η είσοδος σε βρόχο υποχρεωτικά γίνεται από την αρχή του.
 - Δεν πρέπει να χρησιμοποιηθεί η ίδια <μεταβλητή> ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

...Δομές Επανάληψης...

...Για (for)...

Να γίνει αλγόριθμος που να εμφανίζει όλους τους αριθμούς από το 1 έως το 100 που έχουν ακριβώς 5 διαιρέτες (προφανώς μικρότερους ή ίσους από τον αριθμό για τον οποίο αναζητάμε τους διαιρέτες του).

Αλγόριθμος Διαιρέτες

```
Για x από 1 μέχρι 100  
    πλήθος_διαιρετών ← 0
```

```
    Για y από 1 μέχρι x
```

```
        Αν  $x \bmod y = 0$  τότε
```

```
            πλήθος_διαιρετών ← πλήθος_διαιρετών + 1
```

```
        Τέλος_αν
```

```
    Τέλος_επανάληψης
```

```
    Αν πλήθος_διαιρετών = 5 τότε
```

```
        Εμφάνισε x
```

```
    Τέλος_αν
```

```
Τέλος επανάληψης
```

Τέλος Διαιρέτες

Ευχαριστώ

Ερωτήσεις?