

9. Οδηγός μελέτης – Δομή Επανάληψης ΟΣΟ...ΕΠΑΝΑΛΑΒΕ (Θ ΚΕΦ. 2.4.5, 8.2) (Ο 3.1)

9.1 Εισαγωγή στην δομή επανάληψης

Ας υποθέσουμε ότι μας ζητάνε να φτιάξουμε ένα αλγόριθμο που εμφανίζει στην οθόνη του υπολογιστή 100 φορές την λέξη **HELLO**. Με βάση αυτά που γνωρίζουμε ως τώρα θα έπρεπε να γράψουμε 100 φορές την εντολή...

```
ΓΡΑΨΕ 'HELLO'  
ΓΡΑΨΕ 'HELLO'
```

.....

Κάτι το οποίο είναι πολύ χρονοβόρο μέχρι και αδύνατο αν ο αριθμός των επαναλήψεων αυξηθεί υπερβολικά. Σε αυτές τις περιπτώσεις την λύση μας την προσφέρει η δομή επανάληψης.

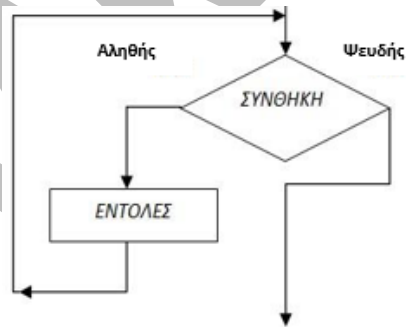


Η δομή επανάληψης χρησιμοποιείται στην περίπτωση που πρέπει να εκτελέσουμε πολλές φορές κάποιες εντολές.

Δομή επανάληψης ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ

Η πρώτη και πιο βασική δομή επανάληψης είναι η **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ** που εκτελεί τις εντολές που περιλαμβάνει - δηλαδή τις εντολές που βρίσκονται μεταξύ του **ΟΣΟ** και του **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ** - συνέχεια μέχρι η συνθήκη να γίνει **ΨΕΥΔΗΣ**

```
ΟΣΟ (ΣΥΝΘΗΚΗ) ΕΠΑΝΑΛΑΒΕ  
  ΓΡΑΨΕ 'HELLO'  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



Έστω ότι θέλουμε να δημιουργήσουμε μια επανάληψη που εκτελείται όσες φορές θέλουμε. Θα βάλουμε τον υπολογιστή να μετράει από το 1 μέχρι τον αριθμό που θέλουμε. Για να το καταφέρουμε αυτό θα χρησιμοποιήσουμε μια μεταβλητή ως **ΜΕΤΡΗΤΗ**.

Εκχωρούμε στον μετρητή ως αρχική τιμή το 1

Αυξάνουμε σε κάθε επανάληψη την τιμή του μετρητή κατά +1 οπότε η μεταβλητή M (μετρητής) θα πάρει διαδοχικά τις τιμές.
1 2 3 4 5 6 ...

```
Μ <-- 1  
ΟΣΟ Μ <= 100 ΕΠΑΝΑΛΑΒΕ  
  ΓΡΑΨΕ 'HELLO'  
  Μ <-- Μ + 1  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Συνθήκη τερματισμού

Στο **ξεκίνημα** καθώς και **κάθε φορά που τελειώνουν** οι εντολές που βρίσκονται μέσα στην επανάληψη ο υπολογιστής ελέγχει την συνθήκη. Αν η συνθήκη είναι **ΑΛΗΘΗΣ** εκτελεί τις εντολές ενώ αν η συνθήκη είναι **ΨΕΥΔΗΣ** τερματίζετε η επανάληψη και συνεχίζει με τις εντολές που βρίσκονται μετά το τέλος της επανάληψης.



Κάθε δομή επανάληψης έχει μια συνθήκη τερματισμού. Επαναληπτική δομή που δεν τερματίζετε παραβιάζει το κριτήριο της περατότητας!

Ατέρμων βρόγχος

```
α <-- 1  
Όσο α <> 6 επανάλαβε  
  α <-- α + 2  
τέλος_επανάληψης
```

Παράδειγμα 1

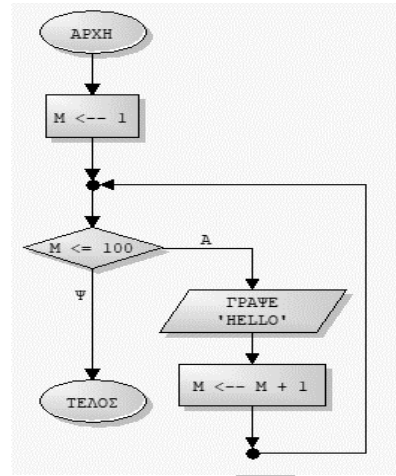
Να γίνει το διάγραμμα ροής του παρακάτω αλγόριθμου

Αλγόριθμος Π2

```

M ← 1
Όσο M ≤ 100 επανάλαβε
  Γράψε 'HELLO'
  M ← M + 1
Τέλος_επανάληψης
Τέλος Π2

```



9.2 Χρήση της δομής επανάληψης

1. Εργασία που επαναλαμβάνετε Χ φορές.

Να γραφτεί αλγόριθμος που διαβάζει 20 θετικούς ακέραιους αριθμούς και εμφανίζει για κάθε αριθμό αν είναι άρτιος ή περιττός.



Όταν θέλουμε μια εργασία να την επαναλάβουμε πολλές φορές την τοποθετούμε εντός μιας επαναληπτικής δομής.

```

M ← 1
ΟΣΟ M ≤ 20 ΕΠΑΝΑΛΑΒΕ
  [ ]
  M ← M + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

```

ΔΙΑΒΑΣΕ A
ΑΝ A MOD 2 = 0 ΤΟΤΕ
  ΓΡΑΨΕ 'ΑΡΤΙΟΣ'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'ΠΕΡΙΤΤΟΣ'
ΤΕΛΟΣ_ΑΝ

```

Άρα τον διπλανό αλγόριθμο θα τον τοποθετήσουμε μέσα σε μια επανάληψη που θα εκτελεστεί 20 φορές.

```

M ← 1
ΟΣΟ M ≤ 20 ΕΠΑΝΑΛΑΒΕ
  ΔΙΑΒΑΣΕ A
  ΑΝ A mod 2 = 0 ΤΟΤΕ
    ΓΡΑΨΕ 'ΑΡΤΙΟΣ'
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'ΠΕΡΙΤΤΟΣ'
  ΤΕΛΟΣ_ΑΝ
  M ← M + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

Η σωστή αρχική και τελική τιμή του Μετρητή θα μας βοηθήσει να σταματήσουμε την επανάληψη μετά από 20 φορές.

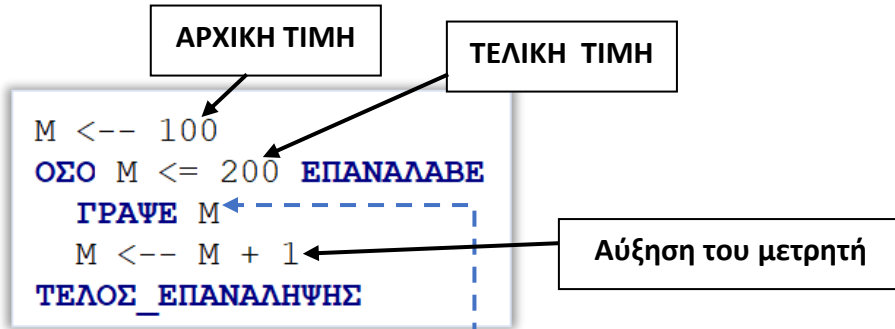
Σε κάθε επανάληψη διαβάζουμε έναν νέο αριθμό, τον τοποθετούμε στην μεταβλητή A και ελέγχουμε αν ο αριθμός είναι άρτιος ή περιττός.



Μέσα σε μια δομή επανάληψης μπορούμε να έχουμε άλλες εμφωλευμένες δομές. Η εμφωλευμένη δομή θα εκτελείτε κάθε φορά που επαναλαμβάνονται οι εντολές της επανάληψης.

2. Επανάληψη στην οποία χρησιμοποιούμε τον μετρητή στην εργασία.

Να γραφτεί τμήμα που εμφανίζει όλους τους ακέραιους αριθμούς από το 100 μέχρι και το 200.

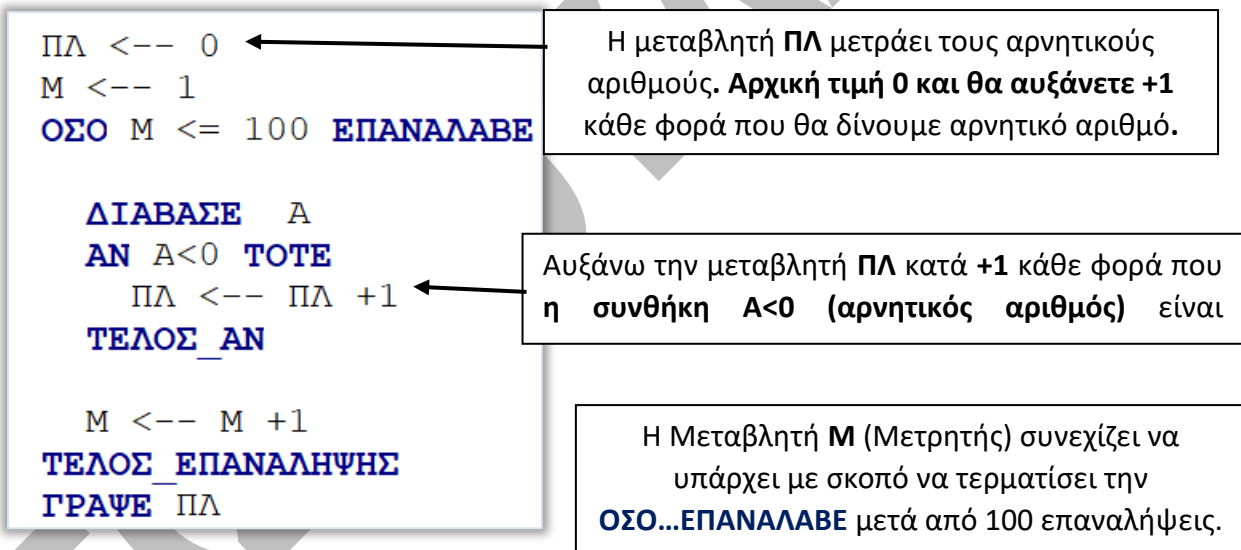


Συχνά τον μετρητή μιας επανάληψης τον χρησιμοποιούμε, εσωτερικά μέσα στην επανάληψη, για να επεξεργαστούμε την τιμή του. Σε αυτές τις περιπτώσεις προσαρμόζουμε το εύρος τιμών του έτσι ώστε να μας βοηθάει όχι μόνο στον τερματισμό της επανάληψης αλλά και στην επίλυση του

3. Τερματισμός επανάληψης με βάση συνθήκη της εκφώνησης

Εύρεση πλήθους

Να γίνει ο αλγόριθμος που διαβάζει 100 αριθμούς και εμφανίζει το πλήθος των αρνητικών.



Δεν ξεχνάμε να θέτουμε **αρχική τιμή 0** στην μεταβλητή του **πλήθους**. Ο υπολογιστής δεν μπορεί να εκτελέσει **την πρώτη φορά** την εντολή $ΠΛ \leftarrow ΠΛ + 1$ αν η μεταβλητή ΠΛ δεν αρχική τιμή.

Απαγορεύεται να χρησιμοποιούμε μεταβλητές αν δεν έχουν τιμή!

Εύρεση αθροίσματος

Να γράψετε τον αλγόριθμο που υπολογίζει το άθροισμα $\Sigma = 1 + 2 + 3 + 4 + \dots + 100$

Επανάληψη	ΑΘ	M
	0	1
1 ^η	1	2
2 ^η	3	3
3 ^η	6	4
4 ^η	10	5
5 ^η	15	6
...		

```

ΑΘ <-- 0
Μ <-- 1
ΟΣΟ Μ <= 100 ΕΠΑΝΑΛΑΒΕ

    ΑΘ <-- ΑΘ + Μ
    Μ <-- Μ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ΑΘ
    
```

Παράδειγμα επανάληψης με συνθήκη στην εκφώνηση.

Να διαβάσεις αριθμούς μέχρι το άθροισμά τους να ξεπεράσει το 1000. Στην συνέχεια να εμφανίσουμε πόσους (πλήθος) αριθμούς δώσαμε.

```

Πλήθος
Π <-- 0
S <-- 0
ΟΣΟ S <= 1000 ΕΠΑΝΑΛΑΒΕ
    ΔΙΑΒΑΣΕ Α
    S <-- S + Α
    Π <-- Π + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Π
    
```

Δεν έχουμε μετρητή! Η επανάληψη εκτελείτε όσο το άθροισμα είναι ακόμα μικρότερο ή ίσο του 1000 ($S \leq 1000$)



Η Εντολή εκχώρησης, για να υπολογίσουμε ένα άθροισμα, είναι:

ΑΘΡΟΙΣΜΑ ← ΑΘΡΟΙΣΜΑ + ΤΙΜΗ

Προσθέτουμε στο **ΑΘΡΟΙΣΜΑ** την **ΤΙΜΗ** και στην συνέχεια αυτό που θα βρούμε το εκχωρούμε ως νέα τιμή στην ίδια μεταβλητή **ΑΘΡΟΙΣΜΑ**

4. Τερματισμός επανάληψης με συγκεκριμένη τιμή (τιμή φρουρός).

Να γραφεί πρόγραμμα το οποίο δέχεται από το πληκτρολόγιο θετικούς αριθμούς. Η διαδικασία σταματά με την είσοδο αρνητικού αριθμού ή του 0. Τέλος, τυπώνει το άθροισμα των θετικών αριθμών που έχουν πληκτρολογηθεί.

```

ΠΡΟΓΡΑΜΜΑ Άθροισμα
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: sum, x
ΑΡΧΗ
sum <- 0
ΔΙΑΒΑΣΕ x
ΟΣΟ x > 0 ΕΠΑΝΑΛΑΒΕ
    sum <- sum + x
    ΔΙΑΒΑΣΕ x
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ sum
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Άθροισμα
    
```

```

ΔΙΑΒΑΣΕ x
ΟΣΟ (έλεγχος του X) ΕΠΑΝΑΛΑΒΕ

    ΔΙΑΒΑΣΕ x
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    
```

Όταν η εκφώνηση αναφέρει κάποια τιμή, **τιμή φρουρός**, για το τέλος της επανάληψης ο τρόπος επίλυσης είναι η δημιουργία της επανάληψης όπως βλέπουμε παραπάνω.

Προσοχή: Υποχρεωτικά πρέπει να διαβάσουμε την πρώτη τιμή του X πριν την **ΟΣΟ** αφού δεν επιτρέπεται να χρησιμοποιούμε μεταβλητές χωρίς να τους έχουμε δώσει τιμές!



9.3 Πολλαπλασιασμός αλά ρωσικά (Θεωρία)

Η πράξη του πολλαπλασιασμού δεν εκτελείται από τον υπολογιστή με τον κλασικό τρόπο. Πιο συγκεκριμένα, ο χρησιμοποιούμενος τρόπος είναι ο λεγόμενος **πολλαπλασιασμός αλά ρωσικά**.

45	19	45
90	9	90
180	4	
360	2	
720	1	720
Άθροισμα = 855		

Έστω, λοιπόν, ότι δίνονται δύο θετικοί ακέραιοι αριθμοί, οι αριθμοί 45 και 19.

Οι αριθμοί γράφονται δίπλα-δίπλα και ο πρώτος διπλασιάζεται, ενώ ο δεύτερος υποδιπλασιάζεται αγνοώντας το δεκαδικό μέρος. Τελικώς, το γινόμενο ισούται με το άθροισμα των στοιχείων της πρώτης στήλης, όπου αντίστοιχα στη δεύτερη στήλη υπάρχει περιττός αριθμός.

Η μέθοδος αυτή χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ό,τι ο γνωστός μας χειρωνακτικός τρόπος πολλαπλασιασμού. Πιο συγκεκριμένα, απαιτεί πολλαπλασιασμό επί δύο, διαίρεση διά δύο και πρόσθεση. Σε επίπεδο, λοιπόν, κυκλωμάτων υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση διά δύο μπορούν να υλοποιηθούν ταχύτατα με μία απλή **εντολή ολίσθησης (shift)**. Άρα η **ολίσθηση ενός δυαδικού αριθμού προς τα αριστερά** με προσθήκη ενός 0 ισοδυναμεί με πολλαπλασιασμό επί δύο, ενώ η ολίσθηση προς τα **δεξιά** με αποκοπή του τελευταίου ψηφίου ισοδυναμεί με την ακέραια διαίρεση διά δύο.

Παράδειγμα

Ο αριθμός 7 στο δυαδικό γράφεται **111**

Ολίσθηση προς τα δεξιά, δηλαδή αποκοπή του τελευταίου ψηφίου, έχω **11** = 3 (7 div 2)

Ολίσθηση προς τα αριστερά, με προσθήκη ενός μηδέν στο τέλος, έχω **1110** = 14 (7 * 2)

Αλγόριθμος για τον πολλαπλασιασμό αλά ρωσικά

Δίνεται ο παρακάτω αλγόριθμος πολλαπλασιασμού αλά ρωσικά δύο θετικών ακεραίων αριθμών M1 και M2 σε φυσική γλώσσα κατά βήματα Να γράψετε στο τετράδιό σας την κωδικοποίηση των παραπάνω βημάτων σε ΓΛΩΣΣΑ

Αλγόριθμος: Πολλαπλασιασμός δύο θετικών ακεραίων (αλά ρωσικά)	
Είσοδος:	Δύο ακέραιοι M1 και M2, όπου M1, M2 > 1
Έξοδος:	Το γινόμενο P=M1*M2
Βήμα 1	Θέσε P=0
Βήμα 2	Αν M2>0, τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7
Βήμα 3	Αν ο M2 είναι περιττός, τότε θέσε P=P+M1
Βήμα 4	Θέσε M1=M1*2
Βήμα 5	Θέσε M2=M2/2 (θεώρησε μόνο το ακέραιο μέρος)
Βήμα 6	Πήγαινε στο Βήμα 2
Βήμα 7	Τύπωσε τον P.

```

Διαβάσε M1, M2
P <-- 0
Όσο M2 > 0 επανάλαβε
    Αν M2 mod 2 = 1 τότε
        P <-- P + M1
    Τελος_αν
    M1 <-- M1*2
    M2 <-- M2 div 2
Τέλος_επανάληψης
Γραψε P
    
```