

## Θεωρία ...

- ☛ **1. Πότε χρησιμοποιούμε την δομή επανάληψης; Ποιες είναι οι διάφορες εντολές (μορφές) της;**

**ΑΠΑΝΤΗΣΗ**

Η δομή επανάληψης χρησιμοποιείται όταν μια σειρά εντολών πρέπει να εκτελεστεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό. Επίσης δίνει την δυνατότητα ένα σύνολο εντολών να εκτελεσθεί πολλές φορές.

Για παράδειγμα, όλες οι τράπεζες κάθε εξάμηνο αποδίδουν τόκους των καταθέσεων ταμειυτηρίου. Ο υπολογισμός των τόκων πρέπει να γίνει για όλους τους λογαριασμούς της τράπεζας, άρα η πράξη

$$\text{τόκος} = \text{ποσό} * \text{επιτόκιο}$$

πρέπει να εκτελεσθεί για όλους τους τραπεζικούς λογαριασμούς.

Οι επαναληπτικές διαδικασίες μπορεί να έχουν διάφορες μορφές και συνήθως εμπεριέχουν και συνθήκες επιλογών

Υπάρχουν τρεις εντολές (μορφές) της δομής επανάληψης:

- 📖 **Όσο...επανάλαβε**
- 📖 **Μέχρις\_ότου**
- 📖 **Για...από...μέχρι**

- ☛ **2. Αναφερθείτε στην έννοια του βρόχου.**

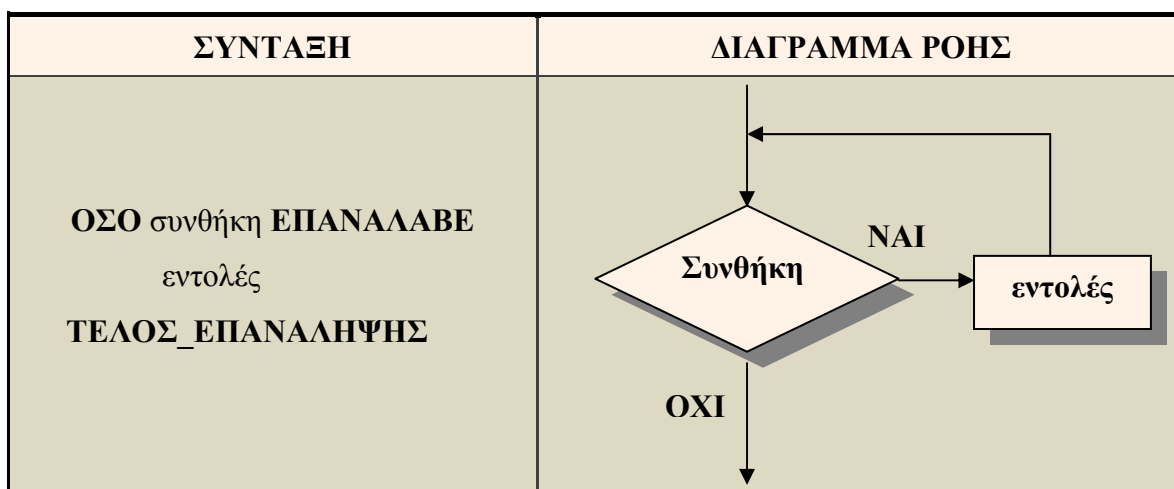
**ΑΠΑΝΤΗΣΗ**

Το τμήμα του αλγορίθμου (προγράμματος) που επαναλαμβάνεται ονομάζεται βρόχος.

- *Ατέρμων βρόχος ονομάζεται ο βρόχος που δεν τερματίζεται. Δηλαδή οι εντολές επαναλαμβάνονται συνεχώς και η εκτέλεση τους δεν σταματάει ποτέ.*

- ☛ **3. Να περιγραφεί η επαναληπτική δομή όσο...επανάλαβε και να δοθεί το διάγραμμα ροής.**

**ΑΠΑΝΤΗΣΗ**



### ✂ Λειτουργία

Ελέγχεται η συνθήκη και αν είναι Αληθής, εκτελούνται οι εντολές που βρίσκονται ανάμεσα στις **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ** και **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**. Στη συνέχεια ελέγχεται πάλι η συνθήκη και αν ισχύει, εκτελούνται πάλι οι ίδιες εντολές. Όταν η συνθήκη γίνει Ψευδής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.

### 📄 Παρατηρήσεις

- ☞ Η συνθήκη που ελέγχει την επανάληψη βρίσκεται στην αρχή της επανάληψης και ο βρόχος επαναλαμβάνεται συνεχώς, όσο η συνθήκη αυτή ισχύει.
- ☞ Με την Όσο... επανάλαβε μπορούν να εκφραστούν όλες οι επαναλήψεις και γι αυτό είναι η σημαντικότερη από όλες τις εντολές (μορφές) επανάληψης.
- ☞ Όταν η λογική συνθήκη είναι εξ'αρχής ψευδής, τότε προσπερνάμε τις εντολές μέσα στην επανάληψη και εκτελείται κατευθείαν η εντολή μετά το Τέλος\_επανάληψης. Οπότε μπορεί να μην γίνει ούτε μία επανάληψη εάν η συνθήκη είναι εξ'αρχής ψευδής.
- ☞ Ο αριθμός των επαναλήψεων δεν είναι γνωστός, ούτε μπορεί να υπολογιστεί πριν από την εκτέλεση του προγράμματος.
- ☞ Εφόσον μετά από κάθε επανάληψη ελέγχεται εκ νέου η συνθήκη, πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μία εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Σε αντίθετη περίπτωση η επανάληψη δε θα τερματίζεται και θα εκτελείται συνεχώς.
- ☞ Η χρήση τιμών για τον τερματισμό μίας επαναληπτικής διαδικασίας είναι συνήθης στον προγραμματισμό. Η τιμή αυτή ορίζεται από τον προγραμματιστή και αποτελεί μια σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή είναι τέτοια, ώστε να μην είναι λογικά σωστή για το πρόβλημα. Η τιμή αυτή συχνά αποκαλείται "τιμή φρουρός"

**ΠΑΡΑΔΕΙΓΜΑ**

Να γίνει αλγόριθμος που να εμφανίζει όλους τους ακέραιους αριθμούς από το 1 έως το 100.

*Αλγόριθμος K38*

$x \leftarrow 1$

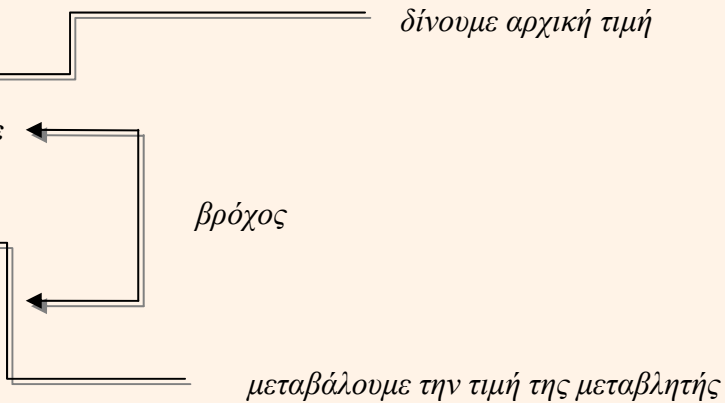
**Όσο  $x \leq 100$  επανάλαβε**

Εμφάνισε  $x$

$x \leftarrow x + 1$

**Τέλος\_επανάληψης**

**Τέλος K38**



☛ 4. Να περιγραφεί η επαναληπτική δομή Μέχρις\_ότου και να δοθεί το διάγραμμα ροής.

☑ ΑΠΑΝΤΗΣΗ

ΣΥΝΤΑΞΗ	ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ
<p><b>ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ</b> εντολές <b>ΜΕΧΡΙΣ_ΟΤΟΥ</b> λογική-έκφραση</p>	

✂ Λειτουργία

Εκτελούνται οι εντολές μεταξύ των **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ**. Στη συνέχεια ελέγχεται η λογική έκφραση και αν δεν ισχύει (είναι ψευδής), τότε οι εντολές που βρίσκονται ανάμεσα στις **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ**, εκτελούνται πάλι.

Ελέγχεται ξανά η λογική έκφραση και αν δεν ισχύει, επαναλαμβάνεται η εκτέλεση των ίδιων εντολών. Όταν η λογική έκφραση γίνει Αληθής τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά από την **ΜΕΧΡΙΣ\_ΟΤΟΥ**.

### 📄 Παρατηρήσεις

- ☞ Η συνθήκη που ελέγχει την επανάληψη βρίσκεται στο τέλος της επανάληψης και ο βρόχος επαναλαμβάνεται συνεχώς, όσο η συνθήκη αυτή δεν ισχύει.
- ☞ Το σύνολο εντολών θα εκτελεσθεί τουλάχιστον μία φορά ακόμα και αν είναι η συνθήκη εξ' αρχής αληθής. Αυτό συμβαίνει διότι πρώτα εκτελείται το σύνολο των εντολών που βρίσκεται μέσα στην επαναληπτική δομή και μετά ελέγχεται η συνθήκη.
- ☞ Ο αριθμός των επαναλήψεων δεν είναι γνωστός, ούτε μπορεί να υπολογιστεί πριν από την εκτέλεση του προγράμματος.
- ☞ Πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μία εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Σε αντίθετη περίπτωση η επανάληψη δε θα τερματίζεται και θα εκτελείται συνεχώς.
- ☞ Σε περιπτώσεις όπου η επανάληψη θα συμβεί υποχρεωτικά μία φορά, είναι προτιμότερη η χρήση της Μέχρις\_ότου. Χαρακτηριστική περίπτωση όπου προτιμάται η εντολή Μέχρις\_ότου είναι στον έλεγχο αποδεκτών τιμών καθώς και στην επιλογή από προκαθορισμένες απαντήσεις ή μενού. Επίσης η Μέχρις\_ότου μπορεί να χρησιμοποιηθεί και σε όλες τις άλλες περιπτώσεις αλλά τότε δημιουργούνται πιο πολύπλοκοι αλγόριθμοι σε σχέση με την όσο...επανάλαβε.

### ΠΑΡΑΔΕΙΓΜΑ

Να γίνει αλγόριθμος που να εμφανίζει όλους τους ακέραιους αριθμούς από το 1 έως το 100.

#### Αλγόριθμος K38

$x \leftarrow 1$

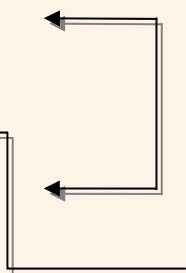
Αρχή\_επανάληψης

Εμφάνισε  $x$

$x \leftarrow x + 1$

Μέχρις\_ότου  $x > 100$

Τέλος K38



βρόχος

μεταβάλλουμε την τιμή της μεταβλητής

☛ **5. Ποιες είναι οι διαφορές μεταξύ της Όσο... επανάλαβε και της Μέχρις\_ότου.**

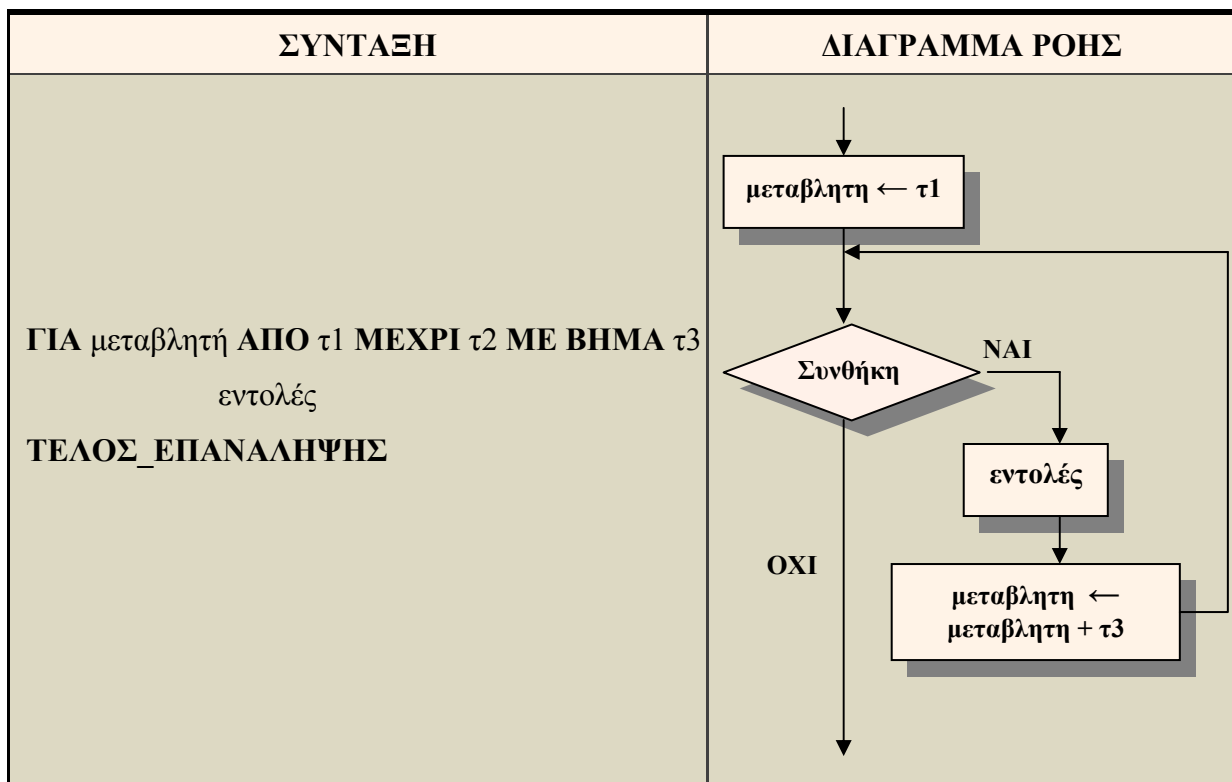
☑ **ΑΠΑΝΤΗΣΗ**

Οι κυριότερες διαφορές είναι:

- ☰ Ο έλεγχος της συνθήκης στην Όσο...επανάλαβε γίνεται στην αρχή του βρόχου ενώ της Μέχρις\_ότου στο τέλος.
- ☰ Στην Μέχρις\_ότου το σύνολο εντολών θα εκτελεσθεί τουλάχιστον μία φορά δηλαδή θα γίνει σίγουρα μια επανάληψη ενώ στην Όσο...επανάλαβε μπορεί να μην γίνει ούτε μία επανάληψη εάν εξ' αρχής η συνθήκη είναι ψευδής.
- ☰ Όσο η συνθήκη είναι αληθής συνεχίζει η Όσο...επανάλαβε ενώ όσο η συνθήκη είναι ψευδής συνεχίζει η Μέχρις\_ότου.
- ☰ Στις περιπτώσεις όπου η επανάληψη θα συμβεί υποχρεωτικά μία φορά, είναι προτιμότερη η χρήση της Μέχρις\_ότου.

☛ **6. Να περιγραφεί η επαναληπτική δομή για...από...μέχρι και να δοθεί το διάγραμμα ροής.**

☑ **ΑΠΑΝΤΗΣΗ**



### ✖ Λειτουργία

Οι εντολές του βρόχου εκτελούνται για όλες τις τιμές της μεταβλητής από την αρχική τιμή μέχρι και την τελική τιμή, μεταβαλλόμενες όπως καθορίζει η τιμή του βήματος..

Αναλυτικότερα:

1. Αρχικά στην μεταβλητή εκχωρείται η αρχική τιμή  $\tau_1$ .
2. Η τιμή της μεταβλητής συγκρίνεται με την τελική τιμή ( $\tau_2$ ) και αν ισχύει η συνθήκη **μεταβλητή**  $\leq \tau_2$  (όταν το βήμα είναι θετικό) ή **μεταβλητή**  $\geq \tau_2$  (όταν το βήμα είναι αρνητικό) τότε εκτελούνται οι εντολές που βρίσκονται στο βρόχο (ανάμεσα στις εντολές **ΓΙΑ** και **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**).
3. Στη συνέχεια η μεταβλητή μεταβάλλεται κατά την τιμή που ορίζει το **ΒΗΜΑ**  $\tau_3$ .
4. Ελέγχουμε αν ισχύει η συνθήκη για την νέα τιμή της μεταβλητής. Αν αυτό συμβαίνει τότε ο βρόχος εκτελείται ξανά.
5. Η διαδικασία αυτή επαναλαμβάνεται συνεχώς μέχρι η συνθήκη να γίνει ψευδής οπότε τερματίζεται η επανάληψη και το πρόγραμμα συνεχίζει με την εντολή που ακολουθεί το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.

### ΠΑΡΑΔΕΙΓΜΑ

Να γίνει αλγόριθμος που να εμφανίζει όλους τους ακέραιους αριθμούς από το 1 έως το 100.

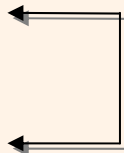
**Αλγόριθμος K38**

**Για**  $x$  από 1 μέχρι 100

**Εμφάνισε**  $x$

**Τέλος\_επανάληψης**

**Τέλος** K38



βρόχος

### 📄 Παρατηρήσεις

- 📄 Η μορφή αυτή μπορεί να χρησιμοποιηθεί **μόνο** όταν είναι γνωστός από την αρχή ο αριθμός των επαναλήψεων.
- 📄 Αν το βήμα είναι ίσο με 1, τότε παραλείπεται.
- 📄 Οι τιμές  $\tau_1$ ,  $\tau_2$  και βήμα ( $\tau_3$ ) δεν είναι απαραίτητο να είναι ακέραιες. Μπορούν λάβουν οποιαδήποτε πραγματική τιμή. Για παράδειγμα:

**Για**  $x$  από 0,2 μέχρι 1 με βήμα 0,001

☰ Το βήμα δεν μπορεί να είναι μηδέν, γιατί τότε ο βρόχος εκτελείται επ' άπειρον. Είναι δυνατόν όμως το βήμα να είναι αρνητικός αριθμός, αρκεί η αρχική τιμή να είναι μεγαλύτερη ή ίση της τελικής τιμής, όπως για παράδειγμα στην επόμενη εντολή:

*Για k από 100 μέχρι 0 με\_βήμα -1*

☰ Υπάρχει η περίπτωση, το σύνολο των εντολών της επαναληπτικής δομής να μην εκτελεσθεί **ούτε μια φορά**. Αυτό θα συμβεί εάν η συνθήκη είναι ψευδής εξ' αρχής. Για παράδειγμα στις επόμενες εντολές δεν γίνεται η επανάληψη καμία φορά:

*Για x από 1 μέχρι -10 με\_βήμα 2.,    Για x από 10 μέχρι 20 με\_βήμα -3*

☛ **7. Ποιά είναι η βασική διαφορά της Για...από...μέχρι από τις άλλες δύο εντολές επανάληψης;**

☑ **ΑΠΑΝΤΗΣΗ**

Η Για...από...μέχρι χρησιμοποιείται για προκαθορισμένο αριθμό επαναλήψεων, ενώ για τις άλλες δύο δεν είναι γνωστός εκ των προτέρων και εξαρτάται από την συνθήκη.

☛ **8. Αναφερθείτε στην εμφωλευμένη επανάληψη.**

☑ **ΑΠΑΝΤΗΣΗ**

Πολύ συχνά για την επίλυση των προβλημάτων απαιτείται η χρήση εμφωλευμένων βρόχων. Σε αυτή την περίπτωση ο ένας βρόχος βρίσκεται μέσα στον άλλο.

Στη χρήση των εμφωλευμένων βρόχων ισχύουν συγκεκριμένοι κανόνες που πρέπει να ακολουθούνται αυστηρά για την σωστή λειτουργία των προγραμμάτων. Συγκεκριμένα:

☰ Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Δηλαδή ο βρόχος που ξεκινάει τελευταίος, πρέπει να ολοκληρώνεται πρώτος.

☰ Η είσοδος σε κάθε βρόχο υποχρεωτικά γίνεται από την αρχή του.

**ΠΑΡΑΔΕΙΓΜΑ**

*Αλγόριθμος K38*

*Για x από 1 μέχρι 100*

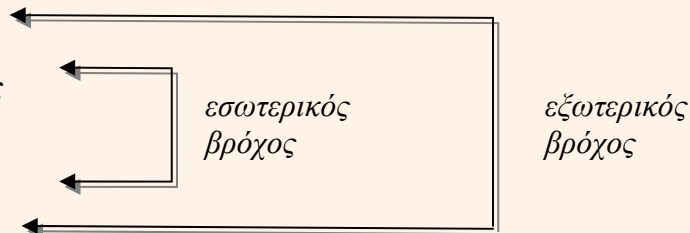
*Για y από 3 μέχρι 10*

*Εμφάνισε y*

*Τέλος\_επανάληψης*

*Τέλος\_επανάληψης*

*Τέλος K38*



☰ Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

☛ 9. Τι είναι ο πολλαπλασιασμός αλά ρωσικά;

☑ ΑΠΑΝΤΗΣΗ

Είναι ένας τρόπος με τον οποίο γίνεται η πράξη του πολλαπλασιασμού μεταξύ δύο ακεραίων αριθμών. Αυτός ο τρόπος χρησιμοποιείται από τον υπολογιστή.

Στην συνέχεια χωρίς βλάβη της γενικότητας θεωρούμε ότι οι ακεραίοι είναι **θετικοί** (μεγαλύτεροι του μηδενός), αλλά η μέθοδος μπορεί εύκολα να μετατραπεί, ώστε να περιγράψει και την περίπτωση των αρνητικών ακεραίων.

☛ 10. Πως λειτουργεί ο πολλαπλασιασμός αλά ρωσικά;

☑ ΑΠΑΝΤΗΣΗ

Πως ακριβώς λειτουργεί η μέθοδος, θα φανεί με το επόμενο παράδειγμα, όπου περιγράφεται ο αλγόριθμος με ελεύθερο κείμενο.

Εστω, λοιπόν, ότι δίνονται δύο θετικοί ακεραίοι αριθμοί, οι αριθμοί 45 και 19.

- Κατασκευάζουμε τρεις στήλες. Στην πρώτη βάζουμε τον ένα αριθμό (45) στην δεύτερη στήλη τον άλλο αριθμό (19) και στην τρίτη βάζουμε τον αριθμό που υπάρχει στην πρώτη στήλη μόνο όταν ο αριθμός της δεύτερης στήλης είναι **περιττός**. Αν δεν είναι περιττός τότε αφήνουμε κενή την τρίτη στήλη.
- Έπειτα διπλασιάζουμε τον αριθμό της πρώτης στήλης ( $45 \times 2$ ) και υποδιπλασιάζουμε τον αριθμό της δεύτερης στήλης αγνοώντας το δεκαδικό μέρος ( $19 \bmod 2$ ).
- Στο σχήμα παρουσιάζεται η επαναλαμβανόμενη διαδικασία, που συνεχίζεται μέχρις ότου στη δεύτερη στήλη να προκύψει μονάδα.
- Τελικώς, το γινόμενο των δύο ακεραίων ισούται με το άθροισμα των στοιχείων της τρίτης στήλης.

1 <sup>η</sup> στήλη	2 <sup>η</sup> στήλη	3 <sup>η</sup> στήλη
45	19	45
90	9	90
180	4	
360	2	
720	1	720
άθροισμα		855



☛ **11. Ποιος είναι ο αλγόριθμος του πολλαπλασιασμού αλά ρωσικά;**

☑ **ΑΠΑΝΤΗΣΗ**

📖 Ο αλγόριθμος του πολλαπλασιασμού αλά ρωσικά με φυσική γλώσσα κατά βήματα.

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

📖 Ο αλγόριθμος του πολλαπλασιασμού αλά ρωσικά με ψευδοκώδικα

**Αλγόριθμος πολ\_αλά\_ρώσικα**

**Διάβασε**  $M1, M2$

$P \leftarrow 0$

**Όσο**  $M2 > 0$  **επανάλαβε**

**Αν**  $M2 \bmod 2 = 1$  **τότε**

$P \leftarrow P + M1$

**Τέλος\_αν**

$M1 \leftarrow M1 * 2$

$M2 \leftarrow M2 \text{ div } 2$

**Τέλος\_επανάληψης**

**Εμφάνισε**  $P$

**Τέλος** πολ\_αλά\_ρώσικα

☛ **12. Τι είναι η ολίσθηση;**

☑ **ΑΠΑΝΤΗΣΗ**

Ο υπολογιστής αποθηκεύει τα δεδομένα σε δυαδική μορφή δηλαδή συνδυασμούς από 0 και 1. Παράδειγμα ενός δυαδικού αριθμού είναι ο 00010001 που αντιστοιχεί στον αριθμό 17 του δεκαδικού συστήματος.

Αν ολισθήσουμε (μετακινήσουμε) τα ψηφία του αριθμού 00010001 κατά μία θέση προς τα αριστερά, αν δηλαδή προσθέσουμε το 0 στο τέλος του αριθμού και αφαιρέσουμε το αρχικό μηδέν θα προκύψει ο αριθμός 00100010 του δυαδικού συστήματος ο οποίος αντιστοιχεί στον αριθμό 34 του δεκαδικού συστήματος. Άρα **η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί δύο.**

Αν ολισθήσουμε (μετακινήσουμε) τα ψηφία του αριθμού 00010001 κατά μία θέση προς τα δεξιά, αν δηλαδή αφαιρέσουμε το 1 στο τέλος του αριθμού και προσθέσουμε στην αρχή το μηδέν θα προκύψει ο αριθμός 00001000 του δυαδικού συστήματος ο οποίος αντιστοιχεί στον αριθμό 8 του δεκαδικού συστήματος. Άρα **η ολίσθηση προς τα δεξιά ισοδυναμεί με ακέραια διαίρεση δια δύο.**

☛ **13. Ποιά η πρακτική σημασία του αλγορίθμου του πολλαπλασιασμού αλά ρωσικά;**

☑ **ΑΠΑΝΤΗΣΗ**

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 αντίθεση η γνωστή μας διαδικασία πολλαπλασιασμού απαιτεί πολλαπλασιασμό με οποιοδήποτε ακέραιο και πρόσθεση. Στον υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση δια δύο μπορούν να υλοποιηθούν ταχύτατα με μία απλή εντολή ολίσθησης (shift), σε αντίθεση με τον πολλαπλασιασμό με οποιοδήποτε ακέραιο που θεωρείται πιο χρονοβόρα διαδικασία.*

*Το τελευταίο γεγονός είναι ο λόγος που ο πολλαπλασιασμός αλά ρωσικά είναι προτιμότερος απ' ό τι ο χειρωνακτικός τρόπος πολλαπλασιασμού δύο ακεραίων.*