ΚΕΦΑΛΑΙΟ 2 – Βασικές έννοιες αλγορίθμων

Αλγόριθμος : Είναι ένα σύνολο βημάτων, αυστηρά καθορισμένων κι εκτελέσιμων σε πεπερασμένο χρόνο, που οδηγούν στην επίλυση ενός προβλήματος.

Χαρακτηριστικά ενός σωστού αλγορίθμου (κριτήρια σωστού αλγορίθμου):

Είσοδος	Καμία, μία ή περισσότερες τιμές δεδομένων δίνονται ως είσοδοι.
Έξοδος	Πρέπει να δημιουργεί μία τιμή δεδομένων ως αποτέλεσμα.
Περατότητα	Να τελειώνει, φτάνοντας στο επιθυμητό αποτέλεσμα, σε πεπερασμένο αριθμό βημάτων ή χρόνο. Αν δεν συμβαίνει αυτό τότε είναι απλώς *υπολογιστική διαδικασία*.
Σαφήνεια	Τα επιμέρους βήματα να είναι απλά και σαφή.
Εκτελεσιμότητα	Τα βήματα να είναι έτσι διατυπωμένα ώστε να μπορούν να εκτελεστούν από τον Η/Υ.
Πληρότητα	Ο αλγόριθμος να προβλέπει κάθε πιθανό ενδεχόμενο κατά την εκτέλεση του (π.χ. μία διαίρεση με το μηδέν).
Αποτελεσματικότητα	Μετά το πέρας της εκτέλεσης των βημάτων να έχει επιτευχθεί ο στόχος (το ζητούμενο)

Τρόποι αναπαράστασης ενός αλγορίθμου

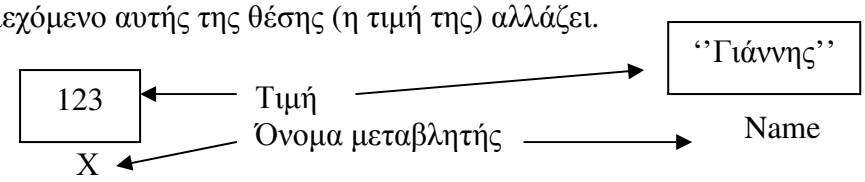
1) **Με ελεύθερο κείμενο :** Αποτελεί έναν αδόμητο τρόπο περιγραφής, που μπορεί να καταστρατηγήσει εύκολα τα κριτήρια του σωστού αλγορίθμου (ασάφειες, μη εκτελεσιμότητα κλπ)
2) **Με διαγραμματικές τεχνικές** : Αποτελεί έναν γραφικό τρόπο παρουσίασης των εντολών. Η πιο γνωστή τεχνική είναι το **διάγραμμα ροής** (flow chart) που παλαιότερα ήταν αρκετά δημοφιλής.
3) **Με φυσική γλώσσα κατά βήματα**: Χρειάζεται προσοχή διότι μπορεί να παραβιαστεί το κριτήριο της πληρότητας. Δεν είναι προτιμητέος τρόπος.
4) **Με κωδικοποίηση:** Στην περίπτωση αυτή, δημιουργούμε τις εντολές κατευθείαν σε μία γλώσσα προγραμματισμού.

Στα παρακάτω παραδείγματα χρησιμοποιούμε μία κωδικοποίηση σε μία μη υπαρκτή δομημένη γλώσσα προγραμματισμού, καλούμενη ως **ψευδογλώσσα**. Φτιάχνοντας τον αλγόριθμο με ψευδογλώσσα, μπορούμε, κατόπιν, να τον μετατρέψουμε χωρίς πολλές μετατροπές σε μία υπαρκτή γλώσσα.

Βασικά στοιχεία αλγορίθμου.

1) **Σταθερές :** Αφορούν ποσότητες που δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του αλγόριθμου. Υπάρχουν 3 ειδών :
   - *Αριθμητικές* π.χ. 123, -8, 3.14
   - *Αλφαριθμητικές* π.χ. ''Γιάννης'', ''Μακεδονίας 12'' . Λέγονται και λεκτικά ή συμβολοσειρές. Περικλείνονται σε ''
   - *Λογικές* π.χ. True, False.

2) **Μεταβλητές** : Αφορούν ποσότητες που μεταβάλλονται κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Χονδρικά, παριστάνουν μία θέση μνήμης που περιέχει μία τιμή. Το περιεχόμενο αυτής της θέσης (η τιμή της) αλλάζει.



Στις μεταβλητές δίνουμε ένα όνομα (αναγνωριστικό) και αυτό χρησιμοποιούμε στον αλγόριθμο. Η τιμή της αλλάζει με μία **εντολή εκχώρησης τιμής** π.χ. X ← 123 ή Name ← ''Γιάννης''.

Κι εδώ, ανάλογα με το είδος δεδομένου που εκχωρείται η μεταβλητή διακρίνεται σε :

- *Αριθμητική* π.χ X ← 123
- *Αλφαριθμητική* π.χ. Name ← ''Γιάννης''
- *Λογική* π.χ. Έγγαμος ← False.

Επιπλέον, οι αριθμητικές διακρίνονται σε
Ακέραιες , αν δέχονται ακέραια τιμή π.χ. 5 , -12
Πραγματικές, αν δέχονται πραγματική τιμή π.χ. 3.14 , -4.56

3) **Τελεστές.** Είναι τα γνωστά σύμβολα. Είναι 3 ειδών:
   - *Αριθμητικοί* : + , - , * (πολλ/σμός), / (διαίρεση) , ^ (ύψωση σε δύναμη), DIV (ακέραια διαίρεση), MOD (ακέραιο υπόλοιπο). Π.χ. X ←5 * 2, X ← 6^2 (=36), X ← 5 DIV 2 (=2 πηλίκο), X ← 5 MOD 2 (=1 υπόλοιπο)
   - *Λογικοί* : AND, OR, NOT . Π.χ. Έγκυρος ← (X >=1) AND (X <=20)
   - *Συγκριτικοί* : > , >= , < , <= , <> (διάφορο)

4) **Εφράσεις** : Συνδυάζουν όλα τα παραπάνω : Π.χ. δες (λογική έκφραση)

5) **Βασικές εντολές** :
   - *Εντολή εισόδου* : **Διάβασε** . Π.χ. Διάβασε X . Στη μεταβλητή X εισάγεται μία τιμή.
   - *Εντολή εξόδου* : **Τύπωσε ή Εμφάνισε** Π.χ. Τύπωσε X
   - *Εντολή εκχώρησης τιμής* : Μορφή : **Μεταβλητή ← Έκφραση**. Αριστερά βάζουμε το όνομα της μεταβλητής και δεξιά μία τιμή ή έκφραση Π.χ. X ← (5 * 2) / 100. Το αποτέλεσμα εκχωρείται στη μεταβλητή.

Με δύο τρόπους λοιπόν, τοποθετούμε τιμές σε μεταβλητές:

I. Με εντολή εισόδου Διάβασε.
II. Κατευθείαν με μία εντολή εκχώρησης,

Αλγοριθμικές δομές

Εννοούμε τον τρόπο που θέτουμε τα βήματα (εντολές) στον αλγόριθμο ώστε να έχουμε σωστή και αυστηρή σύνταξη (δομημένη) και να πληρούνται τα χαρακτηριστικά του.

Είναι 3 :

1. **Ακολουθιακή** : Οι εντολές εκτελούνται η μία μετά την άλλη.
2. **Επιλογής :** Μία ομάδα εντολών εκτελούνται ανάλογα με την τιμή μίας συνθήκης (απόφαση).
3. **Επαναληπτική :** Μία ομάδα εντολών εκτελούνται συνεχώς επαναληπτικά ανάλογα με την τιμή μίας συνθήκης (βρόχος –Loop).

Ακολουθιακή δομή

Παράδειγμα: Να γραφεί αλγόριθμος που διαβάζει τη βάση και το ύψος ενός τριγώνου και υπολογίζει και τυπώνει το Εμβαδόν του.

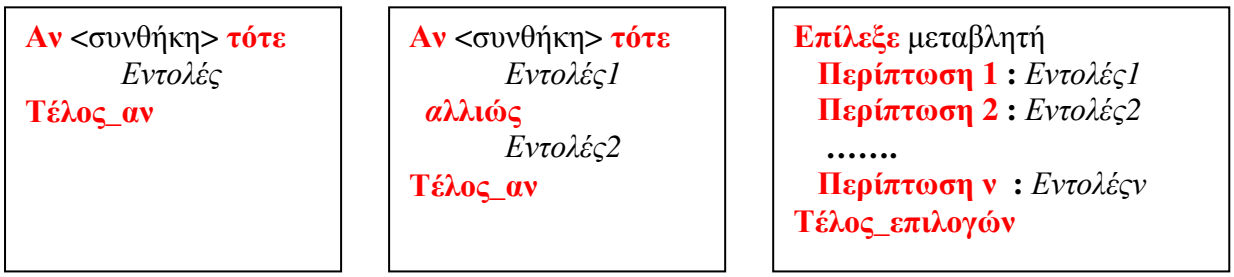
Εδώ θα εισαχθούν δύο αριθμητικά δεδομένα (Βάση και ύψος) και θα εξαχθεί σαν αποτέλεσμα (ζητούμενο) το Εμβαδόν. Για την αποθήκευση των τιμών θέλουμε αντίστοιχα τρεις μεταβλητές.

Αλγόριθμος Εμβαδόν_τριγώνου
Μεταβλητές
Ακέραιοι: βάση, ύψος
Πραγματικός: Εμβαδόν
Αρχή
Διάβασε βάση
Διάβασε ύψος
Εμβαδόν ← (βάση * ύψος) / 2
Τύπωσε Εμβαδόν

Τέλος Εμβαδόν_τριγώνου

Δομή Επιλογής

Έχει τρεις μορφές :



Μορφή 1 Μορφή 2 Πολλαπλή επιλογή

- Στην πρώτη μορφή, αν η συνθήκη δώσει True (αληθής) τότε εκτελείται η ομάδα των εντολών.

  Παράδειγμα :
  Να γραφεί αλγόριθμος που διαβάζει έναν ακέραιο αριθμό κι αν αυτός είναι θετικός να βγάζει το μήνυμα «Είναι θετικός».

  Αλγόριθμος Θετικός
  Μεταβλητές
  Ακέραιος: X *!ο αριθμός θα εισαχθεί στην μεταβλητή με όνομα X*
  Αρχή
  Διάβασε X
  Αν X > 0 **τότε**
  Τύπωσε ''Είναι θετικός''
  Τέλος_αν
  Τέλος Θετικός

- Στην δεύτερη μορφή, αν η συνθήκη δώσει True (αληθής) τότε εκτελείται η ομάδα των Εντολών1 ενώ αν δώσει False εκτελείται η ομάδα των Εντολών2

  Παράδειγμα :
  Να γραφεί αλγόριθμος που διαβάζει έναν ακέραιο αριθμό κι αν αυτός είναι θετικός να βγάζει το μήνυμα «Είναι θετικός» ενώ αν δεν είναι να βγάζει το μήνυμα «Δεν είναι θετικός»

  Αλγόριθμος Έλεγχος_αριθμού
  Μεταβλητές
  Ακέραιος: X *!ο αριθμός θα εισαχθεί στην μεταβλητή με όνομα X*
  Αρχή
  Διάβασε X
  Αν X > 0 **τότε**
  Τύπωσε ''Είναι θετικός''
  αλλιώς
  Τύπωσε ''Δεν είναι θετικός''
  Τέλος_αν

  Τέλος Έλεγχος_αριθμού

- Στην πολλαπλή επιλογή εξετάζεται η τιμή μίας μεταβλητής μέσα σε ένα πλήθος τιμών. Αν ισούται με κάποια από τις περιπτώσεις τότε εκτελείται η αντίστοιχη ομάδα εντολών.

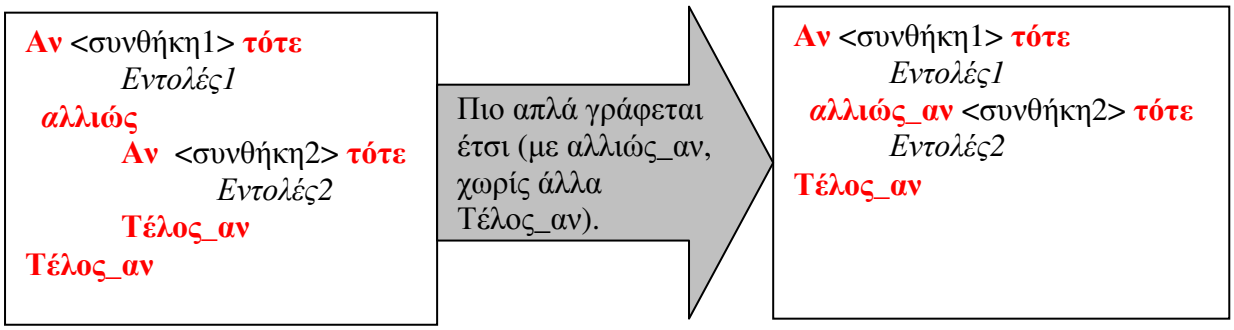
Παράδειγμα:
Γράψτε έναν αλγόριθμο που διαβάζει τον αριθμό της ημέρας και τυπώνει την αντίστοιχη περιγραφή.

Αλγόριθμος Μέρα_εβδομάδας
Μεταβλητές
Ακέραιος: μέρα
Αρχή
Διάβασε μέρα
Επίλεξε μέρα
Περίπτωση 1 : Τύπωσε ''Είναι Κυριακή''
Περίπτωση 2 : Τύπωσε ''Είναι Δευτέρα''
Περίπτωση 3 : Τύπωσε ''Είναι Τρίτη''
Περίπτωση 4 : Τύπωσε ''Είναι Τετάρτη''
Περίπτωση 5 : Τύπωσε ''Είναι Πέμπτη''
Περίπτωση 6 : Τύπωσε ''Είναι Παρασκευή''
Περίπτωση 7 : Τύπωσε ''Είναι Σάββατο''
Τέλος_επιλογών

Τέλος Μέρα_εβδομάδας.

Εμφωλευμένες δομές ελέγχου

Πρόκειται για τις περιπτώσεις που έχουμε μία δομή Αν…τότε.. μέσα σε μία άλλη. Π.χ.



Μπορούμε να έχουμε πολλά **αλλιώς_αν** καταλήγοντας σε μία μορφή *πολλαπλής επιλογής*, όπου είναι καλύτερα να εκφραστεί με την **εντολή Επίλεξε…Τέλος_επιλογών.**

Παράδειγμα : Να γραφεί αλγόριθμος που διαβάζει το βαθμό ενός μαθητή και τυπώνει το χαρακτηρισμό του (π.χ. αν διαβάσει το 15 να τυπώσει «Καλά»).

Αλγόριθμος Χαρακτηρισμός_βαθμού
Μεταβλητές
Πραγματικός: Μ *!ο βαθμός που θα διαβαστεί.*
Αρχή
Διάβασε Μ
Αν Μ<10 **τότε**
Τύπωσε ''Κακώς''
αλλιώς_αν Μ>=10 AND Μ<12,5 **τότε**
Τύπωσε ''Μέτρια''
αλλιώς_αν Μ>=12,5 AND Μ<15,5 **τότε**
Τύπωσε ''Καλά''
αλλιώς_αν Μ>=15,5 AND Μ<18,5 **τότε**

```
        Τύπωσε ‘’Πολύ καλά’’
αλλιώς_αν  M>=18,5 AND M<=20 τότε
        Τύπωσε ‘’Αριστα’’
Τέλος_αν
```

Τέλος Χαρακτηρισμός_βαθμού

Η παραπάνω εμφωλευμένη επιλογή μπορεί να γραφτεί ισοδύναμα και πιο σύντομα με την εντολή Επίλεξε…Τέλος_επιλογών. (Πώς;)

Επαναληπτική δομή

Έχει 3 μορφές :

Όσο <συνθήκη> **επανέλαβε**
Εντολές
Τέλος_επανάληψης

Μορφή 1

Αρχή_επανάληψης
Εντολές
Μέχρις ότου <συνθήκη>

Μορφή 2

Για <μετρητής> **από** <αρχική τιμή> **μέχρι** <τελική τιμή> [**με βήμα** βήμα]
Εντολές
Τέλος_επανάληψης

Μορφή 3

Να σημειωθεί ότι στις περισσότερες περιπτώσεις (και πάντα στη μορφή 3) χρειαζόμαστε μία **μεταβλητή-μετρητής**. Ο μετρητής (συνήθως τον ονομάζουμε i) μετρά τον αριθμό των επαναλήψεων.

- Στην πρώτη μορφή, όσο η συνθήκη δίνει True οι Εντολές εκτελούνται συνεχώς Αν δώσει false τότε σταματάει η επανάληψη. Να σημειωθεί ότι πρώτα ελέγχεται η συνθήκη. Συνεπώς οι εντολές μπορούν να εκτελεστούν 0, μία ή περισσότερες φορές.

Παράδειγμα 1 :
Να γραφεί αλγόριθμος που διαβάζει 10 αριθμούς και υπολογίζει το άθροισμά τους.

Αλγόριθμος Άθροισμα_αριθμών
Μεταβλητές
Ακέραιος: X *!ο αριθμός που διαβάζεται*
Ακέραιος: SUM *!εδώ θα αποθηκεύεται το άθροισμα*
Ακέραιος: i *!ο μετρητής*
Αρχή
i ←1 *!ο μετρητής αρχικοποιείται πριν ξεκινήσει η επανάληψη*
SUM ← 0 *!Το ίδιο και για το άθροισμα*
Όσο i <= 10 **επανέλαβε**
Διάβασε X *!κάνε εισαγωγή του αριθμού στη X*
SUM ← SUM + X *!πρόσθεσέ τον στο άθροισμα*
i ← i +1 *!αυξάνουμε πάντα τον μετρητή πριν γυρίσει επάνω. Γιατί;*
τέλος_επανάληψης
Τύπωσε ‘’Το άθροισμα είναι ‘’, SUM
τέλος Άθροισμα_αριθμών

Εδώ ο αριθμός των επαναλήψεων ήταν γνωστός (10). Όταν ο αριθμός των επαναλήψεων είναι γνωστός (π.χ. πόσοι αριθμοί θα διαβαστούν) προτιμότερη είναι η τρίτη μορφή Για..από..μέχρι…

Πότε είναι κατάλληλη η Οσο..επανέλαβε…; Όταν δεν γνωρίζουμε τον αριθμό των επαναλήψεων.

Παράδειγμα 2 : **(επαναληπτική είσοδος αγνώστου πλήθους στοιχείων)**
Να γραφεί αλγόριθμος που διαβάζει μία ακολουθία αριθμών όπου ο τελευταίος είναι 0 και υπολογίζει το άθροισμά τους.

Αλγόριθμος Άθροισμα_αριθμών
Μεταβλητές
Ακέραιος: X *!ο αριθμός που διαβάζεται*
Ακέραιος: SUM *!εδώ θα αποθηκεύεται το άθροισμα*
!δεν χρειαζόμαστε μετρητή αφού δεν ξέρουμε το πλήθος των αριθμών. Το μόνο που !γνωρίζουμε είναι ότι ο τελευταίος είναι 0. Άρα θα διαβάζει ο Η/Υ μέχρι να συναντήσει το !0.
Αρχή
SUM ← 0 *!Αρχικοποίηση για το άθροισμα*
Διάβασε X *!ο πρώτος διαβάζεται πριν μπει στην επανάληψη*
Όσο X<>0 **επανέλαβε**
SUM ← SUM + X *!πρόσθεσέ τον στο άθροισμα*
Διάβασε X *!διάβασε τον επόμενο*
τέλος_επανάληψης
Τύπωσε ‘’Το άθροισμα είναι ‘’, SUM
τέλος Άθροισμα_αριθμών

- Στην δεύτερη μορφή, εκτελείται πρώτα η ομάδα των εντολών και μετά ελέγχεται η συνθήκη. Συνεπώς οι εντολές εκτελούνται μία ή περισσότερες φορές.
Εδώ, όσο η συνθήκη δίνει False η επανάληψη εκτελείται. Αν δώσει True σταματά. (το αντίθετο με την Οσο..επανέλαβε).

Για το ίδιο παράδειγμα2 παραπάνω, με την μορφή αυτή θα γράφαμε:

Αλγόριθμος Άθροισμα_αριθμών
Μεταβλητές
Ακέραιος: X *!ο αριθμός που διαβάζεται*
Ακέραιος: SUM *!εδώ θα αποθηκεύεται το άθροισμα*
!δεν χρειαζόμαστε μετρητή αφού δεν ξέρουμε το πλήθος των αριθμών. Το μόνο που !γνωρίζουμε είναι ότι ο τελευταίος είναι 0. Άρα θα διαβάζει ο Η/Υ μέχρι να συναντήσει το !0.
Αρχή
SUM ← 0 *!αρχικοποίηση για το άθροισμα*
Αρχή_επανάληψης
Διάβασε X *!διάβασε τον αριθμό*
SUM ← SUM + X *!πρόσθεσέ τον στο άθροισμα*
μέχρις_ότου X=0
Τύπωσε ‘’Το άθροισμα είναι ‘’, SUM

τέλος Άθροισμα_αριθμών

- Η τρίτη μορφή, είναι η καταλληλότερη κι ενδείκνυται όταν γνωρίζουμε τον αριθμό των επαναλήψεων. Χρησιμοποιεί απαραίτητα μετρητή.
Η μορφή αυτή ενσωματώνει :
  - Την αρχικοποίηση του μετρητή.
  - Τον έλεγχο της συνθήκης, αν φτάσαμε στην τελική τιμή.
  - Την αύξηση του μετρητή. Αν δεν καθορίσουμε βήμα τότε εννοείται βήμα 1.

Το παράδειγμα1 παραπάνω ισοδύναμα γράφεται :

Αλγόριθμος Άθροισμα_αριθμών
Μεταβλητές
Ακέραιος: X *!ο αριθμός που διαβάζεται*
Ακέραιος: SUM *!εδώ θα αποθηκεύεται το άθροισμα*
Ακέραιος: i *!ο μετρητής*
Αρχή
SUM ← 0 *!Αρχικοποίηση για το άθροισμα*
Για i **από** 1 **μέχρι** 10
Διάβασε X *!κάνε εισαγωγή του αριθμού στη X*
SUM ← SUM + X *!πρόσθεσέ τον στο άθροισμα*
τέλος_επανάληψης
Τύπωσε ‘’Το άθροισμα είναι ‘’, SUM

τέλος Άθροισμα_αριθμών

Σύνδεση λογικών προτάσεων

Στις δομές ελέγχου και στις επαναληπτικές δομές χρησιμοποιούμε <συνθήκη> για να καθοδηγήσουμε τον Η/Υ προς ένα ή άλλο μονοπάτι εκτέλεσης εντολών. Η <συνθήκη> είναι μία λογική πρόταση που δίνει true ή false και μπορεί να είναι :

Απλή : π.χ. i < 10 ή
Σύνθετη : π.χ. M>=18,5 AND M<=20.

Στην δεύτερη περίπτωση, χρησιμοποιούμε λογικές προτάσεις που συνδέονται μεταξύ τους με τους λογικούς τελεστές **AND** ή **OR**. Επίσης, χρησιμοποιούμε και το **NOT** (όχι) για την αντιστροφή της τιμής μίας πρότασης.

πρόταση **A**	πρόταση **B**	A **OR** B	A **AND** B	**NOT** A
True	True	True	True	False
True	False	True	False	False
False	True	True	False	True
False	False	False	False	True

Γενικά, στην περίπτωση του OR, όταν μία πρόταση είναι αληθής (true) τότε βγαίνει συνολικά αληθής. Στην περίπτωση του AND, όταν μία πρόταση είναι ψευδής (False) τότε βγαίνει συνολικά ψευδής..

Πολλ/σμός αλά ρωσικά

Είναι μία μέθοδος πολλαπλασιασμού δύο αριθμών. Συγκεκριμένα, αν έχουμε δύο αριθμούς Χ και Υ και θέλουμε το Χ*Υ τότε αυτό γίνεται ως εξής:

Βήμα 1 Ο Χ διπλασιάζεται και ο Υ υποδιπλασιάζεται.

Βήμα 2 Συνεχίζουμε το Βήμα 1 μέχρι να φτάσει ο Υ σε 1.

Βήμα 3 Το γινόμενο θα είναι το άθροισμα των στοιχείων Χ στα βήματα που το Υ είναι περιττός.

Π.χ. Χ = 45, Υ = 19

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

Αλγόριθμος Πολλ/σμός_αλά_ρωσικά
Μεταβλητές
Ακέραιοι: Χ, Υ *!οι αριθμοί*
Ακέραιος: Ρ *!το αποτέλεσμα-γινόμενο*
Αρχή
Ρ ← 0 *!Αρχικοποίηση γινομένου*
Οσο Υ>0 **επανέλαβε**
Αν (Υ MOD 2) = 1 **τότε** *!αν το υπόλοιπο της διαίρεσης του Υ με το 2 είναι 1*
Ρ ← Ρ + Χ *!πρόσθεσε το Χ στο άθροισμα*
τέλος_αν
Χ ← Χ * 2 *!διπλασιασμός του Χ*
Υ ← Υ DIV 2 *!υποδιπλασιασμός του Υ. Παίρνουμε το πηλίκο του Υ με το 2*
τέλος_επανάληψης
Τύπωσε ''Το γινόμενο είναι '', Ρ
τέλος_Πολλ/σμός_αλά_ρωσικά

Ποια η πρακτική σημασία της μεθόδου αυτής ; Διότι στους Η/Υ υλοποιείται, σε επίπεδο κυκλωμάτων, πολύ απλά και ταχύτατα με την εντολή της ολίσθησης (shift).