**ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΚΑΛΑΜΠΑΚΑΣ - Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον**

**Δομή Επανάληψης - (2ο κεφ. 2.4.5, 8ο κεφ. 8.2)**

**Παραδείγματα με την εντολή ΓΙΑ**

Η δομή επανάληψης (loop = βρόχος = θηλιά) επιτρέπει την εκτέλεση μιας σειράς εντολών πολλές φορές. Θα μάθουμε τρεις εντολές που την υλοποιούν και τις συναντάμε και στις περισσότερες γλώσσες προγραμματισμού.

Η εντολή ΓΙΑ είναι η πιο εύκολη από τις εντολές επανάληψης. Χρησιμοποιείται όταν είναι γνωστός ο αριθμός των επαναλήψεων. Στηρίζεται σε μία ακέραια μεταβλητή η οποία ελέγχει τον αριθμό των επαναλήψεων. Μπορούνε να την ονομάσουμε **μεταβλητή ελεγκτής επαναλήψεων**. Η μεταβλητή αυτή παίρνει μία αρχική τιμή και στο τέλος κάθε επανάληψης αυξάνεται κατά ένα αριθμό–βήμα (ή μειώνεται κατά ένα αριθμό-βήμα). Κάθε φορά ελέγχεται αν η μεταβλητή ξεπέρασε κάποιο όριο και τότε τερματίζεται η επανάληψη.

1. Να εμφανίσετε στην οθόνη το όνομά σας 15 φορές. Η επίλυση αυτού του απλού προβλήματος μα οδηγεί στο συμπέρασμα ότι χρειαζόμαστε εργαλεία-εντολές για να επαναλαμβάνουμε την ίδια διαδικασία πολλές φορές.
	1. Συμπληρώστε τα κενά που λείπουν.
	2. Αν θέλουμε δίπλα σε κάθε όνομα να εμφανίζεται και ο αντίστοιχος αριθμός σειράς εκτύπωσής τότε τι θα προσθέταμε; Σκεφτείτε ποια μεταβλητή έχει ήδη αυτό που χρειαζόμαστε.
	3. Συμπληρώστε το ισοδύναμο διάγραμμα ροής. Τι παρατηρείται όσο αφορά την αναπαράσταση της δομής επανάληψης σε αυτό;

|  |  |  |
| --- | --- | --- |
| 1η λύση | 2η λύση (Ψευδογλώσσα) | 3η λύση (ΔΡ) |
| Αλγόριθμος Ονομα\_μουΕμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Εμφάνισε "Γρηγόρης"Τέλος Ονομα\_μου | Αλγόριθμος Ονομα\_μουΓια .... από .... μέχρι .... με βήμα ..... Εμφάνισε " Γρηγόρης"Τέλος\_επανάληψηςΤέλος Ονομα\_μου | ι←....ι←.....Εμφάνισε"Γρηγόρης"............. |

1. Να γράψετε τον αλγόριθμο (σε ψευδογλώσσα και διάγραμμα ροής) που να εμφανίζει τους αριθμούς 1..100 με την εντολή ΓΙΑ.
2. Να τροποποιήσετε τον παραπάνω αλγόριθμο ώστε να εμφανίζει του άρτιους αριθμούς στο διάστημα 1-100.
3. Να τροποποιήσετε τον παραπάνω αλγόριθμο ώστε να εμφανίζει του άρτιους αριθμούς στο διάστημα από 1 -100 αλλά με αντίστροφη σειρά, δηλαδή 100, 98, 96 …..4, 2.
4. Να γράψετε αλγόριθμο που να διαβάζει το όνομα και τους δύο βαθμούς ενός μαθητή και να εμφανίζει το όνομά του και τον μέσο όρο των δύο βαθμών. Η παραπάνω διαδικασία να επαναλαμβάνεται για τους 10 μαθητές μιας τάξης.
5. Να γράψετε αλγόριθμο που θα υπολογίζει τον ετήσιο τόκο 1000000 πελατών μίας τράπεζας. Ο αλγόριθμος για κάθε πελάτη θα διαβάζει το ποσό του λογαριασμού του και το επιτόκιο που έχει συμφωνηθεί με την τράπεζα και θα υπολογίζει τον τόκο = ποσό \* επιτόκιο.

Υπολογισμός αθροισμάτων

Για τον υπολογισμό αθροισμάτων πολλών αριθμών στον προγραμματισμό χρησιμοποιούμε μία μεταβλητή που παίζει το ρόλο του αθροιστή π.χ. Σ, καθώς και επανάληψη. Σε κάθε επανάληψη προστίθεται στον αθροιστή ένας νέος αριθμός έτσι ώστε σταδιακά να υπολογιστεί το άθροισμα. Αυτό επιτυγχάνεται με εντολή της μορφής Σ🡨Σ + αριθμός. Ο αθροιστής αρχικοποιείται συνήθως στην τιμή 0.

|  |  |
| --- | --- |
| Υπολογισμός αθροίσματος 1+2+3+…+99+100 (αριθμητική πρόοδος) | Άθροισμα 100 αριθμών που διαβάζονται από το πληκτρολόγιο |
| Σ🡨0Για ι από 1 μέχρι 100 Σ🡨Σ+ιΤέλος\_επανάληψηςΕναλλακτικά υπολογισμός με τον τύπο της αριθμητικής προόδου \Sigma_\nu=\frac{\nu(\alpha_1+\alpha_\nu)}{2}Σ🡨100\*(1+100)/2 | Σ🡨0Για ι από 1 μέχρι 100 Διάβασε αριθμός Σ🡨 Σ + αριθμόςΤέλος\_επανάληψης |

1. Να γραφτεί αλγόριθμος που θα διαβάζει τα γκολ ενός ποδοσφαιριστή σε 18 αγώνες και θα υπολογίζει το συνολικό αριθμό των γκολ καθώς και το μέσο όρο τους σε κάθε παιχνίδι.
2. Να γραφτεί αλγόριθμος που θα διαβάζει 10 αριθμούς και υπολογίζει το άθροισμα των άρτιων και το άθροισμα των περιττών (Υπ: χρειάζονται δύο αθροιστές).
3. Να αναπτυχθεί αλγόριθμος που θα διαβάζει έναν αριθμό Ν (θετικό ακέραιο) και να υπολογίζει τη σειρά: 11 + 22 + 33+ …… + ΝΝ
4. Να γράψετε αλγόριθμο που να υπολογίζει το γινόμενο 1\*2\*3\*…\*50.
5. Πόσες φορές θα εκτελεστεί ο βρόχος στις παρακάτω εντολές:
	1. Για κ από 1 μέχρι 5
	2. Για κ από 1 μέχρι 5 με\_βήμα 2
	3. Για κ από 0 μέχρι 100 με\_βήμα 5
	4. Για κ από 100 μέχρι 4 με\_βήμα -4
	5. Για κ από 5 μέχρι 5
	6. Για κ από 5 μέχρι 1
	7. Για χ από 0 μέχρι 1 με\_βήμα 0,2

Μπορείτε να βγάλετε ένα γενικό τύπο υπολογισμού του αριθμού επαναλήψεων για την εντολή ΓΙΑ με βάση την αρχική\_τιμή, την τελική\_τιμή και το βήμα.

Μεταβλητή Μετρητής

Σε ένα αλγόριθμο πολλές φορές χρειάζεται να μετρήσουμε το πλήθος των δεδομένων που διαβάζονται ή πόσες φορές ισχύει μία συνθήκη π.χ. πόσοι μαθητές έχουνε μο >18. Για το σκοπό αυτό χρησιμοποιούμε μεταβλητές που μπορούμε να τις ονομάσουμε **μετρητές**. Οι μεταβλητές αυτές αρχικοποιούνται και συνήθως αυξάνονται κατά ένα όταν χρειάζεται. Στην επαναληπτική εντολή Για η μεταβλητή που ελέγχει τον αριθμό των επαναλήψεων (ελεγκτής επαναλήψεων) είναι ένα είδος μεταβλητής μετρητή.

|  |
| --- |
| Αλγόριθμος που μετράει πόσοι από τους 50 αριθμούς ξεπερνούν το 18 |
| μ🡨0Για ι από 1 μέχρι 50 Διάβασε χ Αν χ>18 τότε μ🡨μ+1Τέλος\_ανΤέλος\_επανάληψηςΕμφάνισε μ |

1. Να γράψετε αλγόριθμο που να διαβάζει 30 αριθμούς και να βρίσκει πόσοι από αυτούς είναι άρτιοι και πόσοι περιττοί (Υπ: χρησιμοποιείστε 2 μετρητές).
2. Να γραφτεί αλγόριθμος που θα διαβάζει σε ποια τάξη ανήκει ένας μαθητής (Α, Β, Γ) και να εμφανίζει πόσοι μαθητές είναι σε κάθε τάξη (Σύνολο μαθητών 25).
3. Να γραφτεί αλγόριθμος που θα διαβάζει 100 αριθμούς και θα βρίσκει τον μέσο όρο των αριθμών που είναι ανάμεσα σε 15 και 20. (Υπ: θα χρειαστούμε μετρητή και αθροιστή)

Μικρότερος - Μεγαλύτερος

Έχουμε 30 αριθμούς και θέλουμε τον μικρότερο. Έστω min ο πρώτος. Σύγκρινε τους υπόλοιπους 29 ως εξής. Αν ο δεύτερος είναι μικρότερος του min κράτα αυτόν για min. Αν ο τρίτος είναι μικρότερος του min κράτα αυτόν για μικρότερο. Συνεχίζουμε με αυτό τον τρόπο εξετάζοντας όλους τους αριθμούς.

Διάβασε αριθμός

min🡨 αριθμός ! Έστω min ο πρώτος

Για i από 2 μέχρι 30

Διάβασε αριθμός

Αν αριθμός<min τότε

 min🡨αριθμός

Τέλος\_αν

Τέλος\_επανάληψης

1. Να γραφτεί αλγόριθμος που να διαβάζει τα γκολ 30 ποδοσφαιριστών και να εμφανίζει τον πρώτο σκόρερ και πόσα γκολ πέτυχε. (Εδώ θέλουμε και την θέση του μεγαλύτερου. Υποθέστε ότι δεν έχουμε ισοβαθμία.)
2. Να διαβασθούν τα στοιχεία 50 μαθητών ενός σχολείου: το όνομα, η τάξη (a, b, c) και ο βαθμός του καθενός και να βρεθούν για κάθε τάξη το σύνολο των μαθητών και ο μέσος όρος βαθμολογίας κάθε τάξης. Να εμφανίζεται επίσης ποια τάξη έχει το μεγαλύτερο μέσο όρο.
3. Άσκηση ΔΤ8 (σελίδα 23)
4. Άσκηση ΔΣ1 (σελίδα 24)
5. Άσκηση ΔΤ4 (σελίδα 78)

Από την ΓΙΑ στην ΟΣΟ:

Όλα τα προβλήματα της επανάληψης μπορούν να επιλυθούν με την επαναληπτική εντολή ΟΣΟ. Αυτή μπορεί να χρησιμοποιηθεί όταν γνωρίζουμε αλλά και όταν δεν γνωρίζουμε το αριθμό των επαναλήψεων.

|  |  |
| --- | --- |
| Άθροισμα 100 αριθμών που διαβάζονται από το πληκτρολόγιο | Άθροισμα θετικών αριθμών που διαβάζονται από το πληκτρολόγιο μέχρι να δοθεί αρνητικός αριθμός ή μηδέν. |
| Σ🡨0Για ι από 1 μέχρι 100 Διάβασε αριθμός Σ🡨 Σ + αριθμόςΤέλος\_επανάληψηςΕμφάνισε Σ | Σ🡨0ι🡨1Όσο ι<=100 επανάλαβε Διάβασε αριθμός Σ🡨 Σ + αριθμός ι🡨ι+1Τέλος\_επανάληψηςΕμφάνισε Σ | Σ🡨0Διάβασε χΌσο χ>0 επανάλαβε Σ🡨 Σ + αριθμός Διάβασε χΤέλος\_επανάληψηςΕμφάνισε Σ |

Παρατηρώντας τα παραπάνω παραδείγματα βλέπουμε πως μία ΓΙΑ μπορεί να γραφτεί ισοδύναμα με την ΟΣΟ. Όταν επιλύουμε ένα πρόβλημα με την ΟΣΟ πρέπει να σκεφτούμε ποια θα είναι *η συνθήκη τερματισμού της επανάληψης*, όπως στο τρίτο παράδειγμα που η επανάληψη επαναλαμβάνεται όσο οι αριθμοί που διαβάζονται είναι θετικοί. Η συνθήκη πρέπει κάποια στιγμή να γίνει ψευδής, ώστε να τερματιστεί ο βρόχος και να μην έχουμε *ατέρμονη* επανάληψη. Επίσης, αν η συνθήκη είναι εξαρχής ψευδής τότε ο βρόχος δεν εκτελείται καμία φορά.

1. Να κάνετε το διάγραμμα ροής του τρίτου παραδείγματος καθώς και τον πίνακα τιμών, αν ως είσοδος δοθούνε οι αριθμοί: 10, 20,75,15, -2

|  |  |
| --- | --- |
| Διάγραμμα Ροής | Πίνακας τιμών |
|  |  |

1. Να γραφτεί αλγόριθμος που θα διαβάζει θετικούς αριθμούς και θα εμφανίζει τον μικρότερο. Ο αλγόριθμος σταματά όταν δώσουμε αρνητικό αριθμό σαν είσοδο.
2. ΘΕΜΑ 3ο (2007): Ένας συλλέκτης γραμματοσήμων επισκέπτεται στο διαδίκτυο το αγαπημένο του ηλεκτρονικό κατάστημα φιλοτελισμού προκειμένου να αγοράσει γραμματόσημα. Προτίθεται να ξοδέψει μέχρι 1500 ευρώ.

Να αναπτύξετε αλγόριθμο ο οποίος:

α. Για κάθε γραμματόσημο, να διαβάζει την τιμή και την προέλευσή του (ελληνικό/ξένο) και να επιτρέπει την αγορά του, εφόσον η τιμή του δεν υπερβαίνει το διαθέσιμο υπόλοιπο χρημάτων. Διαφορετικά να τερματίζει τυπώνοντας το μήνυμα «ΤΕΛΟΣ ΑΓΟΡΩΝ».

ΣΗΜΕΙΩΣΗ: Δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου. *Μονάδες 10*

β. Να τυπώνει:

1. Το συνολικό ποσό που ξόδεψε ο συλλέκτης. *Μονάδες 2*

2. Το πλήθος των ελληνικών και το πλήθος των ξένων γραμματοσήμων που αγόρασε. *Μονάδες 4*

3. Το ποσό που περίσσεψε, εφόσον υπάρχει, διαφορετικά το μήνυμα «ΕΞΑΝΤΛΗΘΗΚΕ ΟΛΟ ΤΟ ΠΟΣΟ». *Μονάδες 4*

Επαναληπτική εντολή Μέχρις\_ότου, Τουλάχιστον μία επανάληψη:

Όταν κάποιες εντολές πρέπει να επαναληφθούν τουλάχιστον μία φορά τότε υπάρχει μία πιο βολική επαναληπτική εντολή από την Όσο η εντολή Μέχρις\_ότου. Στην εντολή αυτή ο έλεγχος της συνθήκης γίνεται στο τέλος του βρόχου οπότε αυτός εκτελείται πάντα μία φορά. Η εντολή χρησιμοποιείται πολύ συχνά για τον έλεγχο της ορθότητας της εισαγωγής δεδομένων. Π.χ.

Αρχή\_επανάληψης

 Διάβασε Βαθμός

Μέχρις\_ότου Βαθμός >0 ΚΑΙ Βαθμός<=20

1. Να γράψετε αλγόριθμο σε ψευδογλώσσα και διάγραμμα ροής που να εμφανίζει τους αριθμούς 1..100 με την εντολή Μέχρις\_ότου
2. Να λύσετε την άσκηση με τα γραμματόσημα χρησιμοποιώντας την εντολή Μέχρις\_ότου
3. Παράδειγμα (Παν. Εξετάσεις 2005)

∆ίνεται το παρακάτω τμήμα αλγορίθμου:

S ← 0

**Για** I **από** 2 **μέχρι** 100 **με\_βήμα** 2

 S← S + I

**Τέλος\_επανάληψης**

1. Να μετατραπεί σε ισοδύναμο με χρήση της δομής *Όσο … Επανάλαβε.* Μονάδες 5

2. Να μετατραπεί σε ισοδύναμο με χρήση της δομής *Αρχή\_επανάληψης .. μέχρις\_ότου*. Μονάδες 5

1. Γράψτε τη σύνταξη των τριών εντολών επανάληψης και σχολιάστε τη χρησιμότητα και τις διαφορές των εντολών επανάληψης.:

|  |  |  |
| --- | --- | --- |
| **Όσο… Επανέλαβε:**Επαναληπτικό σχήμα με έλεγχο επανάληψης στην αρχή. | **Αρχή\_επανάληψης .. Μέχρις\_΄Οτου**:Επαναληπτικό σχήμα με έλεγχο επανάληψης στο τέλος. | **Για … Από … Μέχρι:**Επαναληπτικό σχήμα ορισμένων φορών επανάληψης. |

1. Τι θα εμφανίσει ο παρακάτω αλγόριθμος;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Αλγόριθμος** ΑριθμοίΑ← 1Β← 1Ν← 0Μ← 2**Όσο** Β < 6 **επανάλαβε** Χ← Α + Β **Αν** Χ **mod** 2 = 0 **τότε** Ν← Ν + 1 **Αλλιώς** Μ← Μ + 1 **Τέλος\_Αν**Α← ΒΒ← Χ**Εμφάνισε** Ν, Μ, Β**Τέλος\_επανάληψης**   **Εμφάνισε** Χ**Τέλος** Αριθμοί | Στο τέλος κάθε επανάληψης εμφανίζονται:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ν | Μ | Β |
| 1η Επανάληψη | 1 | 2 | 2 |
| 2η Επανάληψη |  |  |  |
| 3η Επανάληψη |  |  |  |
| 4η Επανάληψη |  |  |  |

Στο τέλος του αλγορίθμου τυπώνεται **……………** ως τιμή της μεταβλητής Χ. |

1. Να γραφτεί αλγόριθμος που να υπολογίζει τον Μέγιστο Κοινό Διαιρέτη δύο αριθμών.

Ας δούμε πως υπολογίζουμε τον ΜΚΔ των αριθμών 150, 35 καθώς και των αριθμών 57, 24 σύμφωνα με τον Ευκλείδη.

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | z=(x mod y) |  |
| 150 | 35 | 10 |  |
| 35 | 10 | 5 |  |
| 10 | 5 | 0 |  |
| ΜΚΔ(150,35)= | 5 |  |  |
|  |  |  |  |
|  |  |  |  |
| 57 | 24 | 9 |  |
| 24 | 9 | 6 |  |
| 9 | 6 | 3 |  |
| 6 | 3 | 0 |  |
| ΜΚΔ(57,24)= | 3 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Εντοπίστε ποιες ενέργειες επαναλαμβάνονται και γράψτε δίπλα τον αλγόριθμο.

Εμφωλιασμός επαναληπτικών δομών

Μερικά προβλήματα για να επιλυθούν απαιτούνε μια επαναληπτική δομή μέσα σε μία άλλη. Για παράδειγμα αναφέρεται στο βιβλίο στο 8ο κεφάλαιο, σελ. 179, ο αλγόριθμος εμφάνισης της προπαίδειας.

1. Να γράψετε αλγόριθμο που θα διαβάζει τα ονόματα ποδοσφαιριστών μέχρι να δοθεί το όνομα τέλος. Για κάθε ποδοσφαιριστή να διαβάζει τα γκολ που πέτυχε σε 14 αγώνες και να εμφανίζει το συνολικό αριθμό τερμάτων του.
2. Υπάρχει κάποιο λάθος στα παρακάτω τμήματα αλγορίθμων;

|  |  |  |
| --- | --- | --- |
| **Α** | **Β** | **Γ** |
|  S ← 0 **Για** i **από** -3 **μέχρι** 3 **Για** j **από** 10 **μέχρι** 20 **με\_βήμα** i  S ← S + 1 **Τέλος\_Επανάληψης** **Τέλος\_Επανάληψης** **Εκτύπωσε** S |  S ← 0 **Για** i **από** -1 **μέχρι** -3 **Για** j **από** 18 **μέχρι** 13 **με\_βήμα** i  S ← S + i \* j **Τέλος\_Επανάληψης** **Τέλος\_Επανάληψης** **Εκτύπωσε** S |  S ← 0 **Για** i **από** 2 **μέχρι** 5 **Για** j **από** 14 **μέχρι** i  S ← S + 2 **Τέλος\_Επανάληψης** **Τέλος\_Επανάληψης** **Εκτύπωσε** S |

1. Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί τελικά;

**Αλγόριθμος** Άσκηση6

 α ← 0

 **Όσο** (α <= 22) **επανάλαβε**

 **Για** i **από** 1 **μέχρι** 3

 α ← α + i

 **Τέλος\_Επανάληψης**

 α ← α + 5

 **Τέλος\_Επανάληψης**

 **Εκτύπωσε** α

**Τέλος** Άσκηση6

Ασκήσεις με μαθηματικά

1. Να γραφτεί αλγόριθμος που να υπολογίζει το Ν! (ν παραγοντικό). Το παραγοντικό n ! ορίζεται από τον τύπο n !=1\*2\*3\*...\*n . Κατά σύμβαση το παραγοντικό του μηδέν θεωρείται 1 ( 0 !=1 ).
2. Η ακολουθία Fibonacci είναι μια αναδρομική ακολουθία ακέραιων αριθμών. Το κάθε νέο μέλος της ακολουθίας ορίζεται ως το άθροισμα των δύο προηγούμενων μελών. Αν δηλαδή δύο συνεχόμενα μέλη της ακολουθίας είναι F (n−1) και F (n−2) τότε ο επόμενος αριθμός της ακολουθίας είναι το άθροισμα

F (n−1) + F (n−2 ) . Σε αναδρομικό τύπο: F n=F n−1+ F n−2. Να γραφτεί αλγόριθμος που να διαβάζει ένα ακέραιο θετικό αριθμό Ν και να εμφανίζει την ακολουθία των αριθμών Fibonacci μέχρι τον αριθμό αυτό. Ισχύει εξ ορισμού F 0=0 και F 1=1.

1. Το παρακάτω πρόβλημα είναι γνωστό ως Εικασία του Collatz από το όνομα του γερμανού μαθηματικού που το πρότεινε. Για οποιοδήποτε φυσικό αριθμό εκτελούμε μια από τις δύο παρακάτω πράξεις.

(i) Αν είναι άρτιος τον διαιρούμε δια δύο.

(ii) Αν είναι περιττός τον πολλαπλασιάζουμε με 3 και προσθέτουμε 1. Σε μορφή τύ-

που:

Η εικασία υποστηρίζει ότι η ακολουθία καταλήγει πάντα στο 1. Για παράδειγμα αν ο αρχικός αριθμός είναι 7, έχουμε την ακολουθία (α) 7 είναι περιττό 3⋅7+1=22 , (β) 22 είναι άρτιο 22 /2=11 , (γ) 11 είναι άρτιο 3⋅11+1=34 ... και συνεχίζουμε μέχρι να φτάσουμε το 1.

Το πρόβλημα είναι ότι μέχρι τώρα δεν έχει βρεθεί κανείς αριθμός που να μην καταλήγει στο 1 αλλά ταυτόχρονα κανείς δεν μπόρεσε να αποδείξει ότι αυτό ισχύει για όλους τους αριθμούς. Μπορεί τελικά να υπάρχει τέτοιος αριθμός οπότε η εικασία του Collatz να αποδειχθεί λάθος.

Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο:

(α) Να ζητάει έναν φυσικό αριθμό από τον χρήστη μέχρι να δοθεί ένας έγκυρος φυσικός μεγαλύτερος από μηδέν.

(β) Να υπολογίζει τα βήματα της ακολουθίας Collatz γράφοντας το βήμα που εκτελεί και το αποτέλεσμα του κάθε βήματος.

(γ) Να σταματά όταν φτάσει στο 1. Αν δεν φτάσει ποτέ στο 1 τότε έχουμε βρει τον αριθμό που καταρρίπτει την εικασία! Είναι δυνατόν να καταρρίψουμε την εικασία με έναν τέτοιο αλγόριθμο;