Πανεπιστήμιο Ιωαννίνων

Σχολή Θετικών Επιστημών

Τμήμα Πληροφορικής

ΠΛΥ106 - ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

3ο Εργαστήριο

Άσκηση 1. «Φιλικοί» Αριθμοί

Όταν ρώτησαν το Πυθαγόρα τι είναι φίλος, απάντησε: *«Αυτός που είναι ο άλλος σου εαυτός, όπως το 220 και το 284».* Δύο θετικοί αριθμοί *n*, *m* ονομάζονται «φίλοι» ή «φιλικοί» εάν καθένας από αυτούς ισούται με το άθροισμα των γνήσιων διαιρετών του άλλου. Έτσι, το ζεύγος αριθμών (220, 284) είναι φιλικοί, γιατί:

- οι διαιρέτες του 220 είναι 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284, και
- οι διαιρέτες του 284 είναι 1 + 2 + 4 + 71 + 142 = 220.

Ο Pierre Fermat το 1636 βρήκε το ζεύγος φιλικών αριθμών (17.296, 18.416).
Ο Renè Descartes βρήκε ένα τρίτο ζεύγος (9.363.584, 9.437.056).
Ο Nicolò Paganini, το 1866 βρήκε το ζεύγος (1.184, 1.210).

Να σημειωθεί ότι, δεν γνωρίζουμε σήμερα αν τα ζευγάρια των φίλων αριθμών είναι άπειρα ή πεπερασμένα. Επίσης, έχει παρατηρηθεί χωρίς να έχει αποδειχθεί ότι σε ένα ζεύγος φιλικών αριθμών και οι δύο είναι άρτιοι ή και οι δύο περιττοί αριθμοί.

Να γραφεί πρόγραμμα που να ελέγχει σε ένα εύρος τιμών $[r_1, r_2]$ που δίνεται από το χρήστη, ποια ζεύγη «φίλων» αριθμών υπάρχουν. Έτσι το πρόγραμμά σας:

α) Θα ζητά από το χρήστη ως είσοδο δύο θετικούς αριθμούς r_1, r_2, με $r_1 < r_2$. Σε κάθε άλλη περίπτωση θα του ζητά να επαναλάβει τη διαδικασία.

β) Θα βρίσκει στο διάστημα $[r_1, r_2]$ τα ζεύγη των «φίλων» αριθμών και θα τυπώνει ανάλογο μήνυμα (για παράδειγμα το μήνυμα: "Οι αριθμοί 220 και 284 είναι Φιλικοί αριθμοί").

γ) Να τροποποιηθεί κατάλληλα το πρόγραμμα σας έτσι ώστε το άθροισμα των διαιρετών του κάθε αριθμού να υπολογίζεται και να επιστρέφεται από τη συνάρτηση sum_divisors (προσοχή στα ορίσματα και στην τιμή που θα επιστρέφει η συνάρτηση).

Αποθηκεύστε το αρχείο ως *amicable.py*.

Άσκηση 2. Μέγιστος Κοινός Διαιρέτης

Να υπολογίσετε το Μέγιστο Κοινό Διαιρέτη (ΜΚΔ) δύο θετικών ακεραίων αριθμών x, y. Ο ΜΚΔ των x, y, τον οποίο θα συμβολίζουμε με ΜΚΔ(x, y) ορίζεται σύμφωνα με τον αλγόριθμο του Ευκλείδη ως εξής:

Αλγόριθμος του Ευκλείδη
Είσοδος: Δύο θετικοί ακέραιοι αριθμοί x, y.
Έξοδος: Ο Μέγιστος Κοινός Διαιρέτης των x, y.

1. Διαίρεσε τον x με τον y και υπολόγισε το υπόλοιπο p.
2. Αν το p είναι μηδέν, επέστρεψε την τιμή του p.
3. Αν το p δεν είναι μηδέν, ανάθεσε στο x την παλιά τιμή του y και στο y την τιμή του p, και μετά επανέλαβε ολόκληρη την διαδικασία.

α) Να γραφεί πρόγραμμα που υπολογίζει το ΜΚΔ δύο θετικών ακεραίων αριθμών με βάση τον αλγόριθμο του Ευκλείδη. Θα δημιουργήσετε μια συνάρτηση GCD(x,y), την οποία θα καλείτε για κάθε είσοδο του χρήστη.
Αποθηκεύστε το αρχείο ως *gcd1.py*.

β) Να γραφεί πρόγραμμα που να υπολογίζει το ΜΚΔ με αναδρομικό τρόπο. Για το σκοπό αυτό δημιουργείστε μια αναδρομική συνάρτηση GCD_recursive(x,y).
Αποθηκεύστε το αρχείο ως *gcd2.py*.

γ) Ποια η διαφορά στο χρόνο εκτέλεσης των προγραμμάτων που προέκυψαν από τα ερωτήματα α) και β); Χρησιμοποιήστε τη συνάρτηση time() για να υπολογίσετε το χρόνο εκτέλεσής τους.

Άσκηση 3. Οι πύργοι του Ανόι

Ο μύθος λέει ότι στους καλόγερους του ναού Μπράμα δόθηκε το ακόλουθο πρόβλημα, γνωστό ως Πύργοι του Ανόι. Αν έχετε μια βάση με τρεις χρυσούς στύλους σε έναν από τους οποίους έχουν τοποθετηθεί 64 χρυσοί δίσκοι, τότε το τέλος του κόσμου θα έρθει όταν τοποθετήσετε τους δίσκους σε έναν άλλο στύλο σύμφωνα με τους παρακάτω κανόνες:

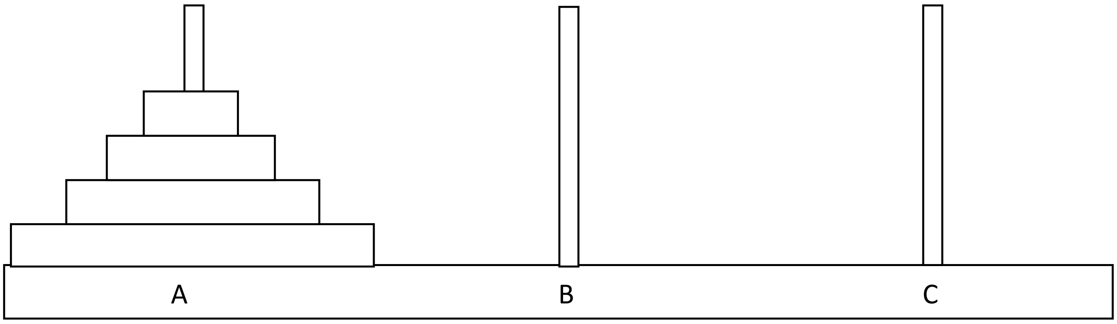
1. Όταν μεταφέρεται ένας δίσκος πρέπει να τοποθετείται σε ένα από τους τρεις στύλους.

2. Μόνο ένας δίσκος μπορεί να μεταφερθεί κάθε φορά και πρέπει να είναι αυτός που βρίσκεται στην κορυφή.

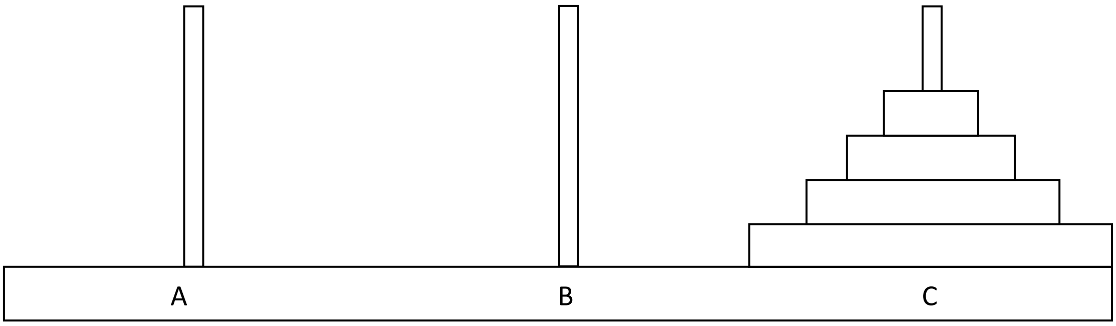
3. Ένας μεγαλύτερος δίσκος δεν μπορεί να τοποθετηθεί πάνω σε ένα μικρότερο.

Να γραφεί ένα πρόγραμμα που να επιλύει το πρόβλημα των πύργων του Ανόι.

Πιο συγκεκριμένα, στο πρόβλημα αυτό θα πρέπει οι δίσκοι (Σχήμα 1) από το στύλο A να μεταφερθούν στο στύλο C, χρησιμοποιώντας το στύλο B ως βοηθητικό (Σχήμα 2). Για την επίλυση του προβλήματος αυτού θα χρειαστεί να δημιουργήσετε μια αναδρομική συνάρτηση Hanoi_rec().
Αποθηκεύστε το αρχείο ως *hanoi.py*.



Σχήμα 1



Σχήμα 2

Άσκηση 4. Fibonacci Αριθμοί (Προαιρετικό)

Η ακολουθία των αριθμών Fibonacci έχει τη μορφή (f_0, f_1, …, f_{i-2}, f_{i-1}, f_i, …) όπου κάθε αριθμός μετά τους δύο πρώτους, που είναι οι αριθμοί 0 και 1 αντίστοιχα, είναι το άθροισμα των δύο προηγούμενων αριθμών. Έτσι, σε μια ακολουθία Fibonacci $f_0 = 0$, $f_0 = 1$, $f_2 = 1$, και γενικά $f_i = f_{i-1} + f_{i-2}$, $i > 1$. Για παράδειγμα, η ακολουθία 0, 1, 1, 2, 3, 5, 8, 13, 21 είναι μια ακολουθία Fibonacci με 9 όρους.

α) Η ακολουθία Fibonacci παράγεται από τη αναδρομική $f(n) = f(n\text{-}1) + f(n\text{-}2)$, με σχέση $f(0) = 0$ και $f(1) = 1$. Να δημιουργήσετε ένα πρόγραμμα υπολογισμού της ακολουθίας των αριθμών Fibonacci χρησιμοποιώντας την παραπάνω αναδρομική σχέση.

β) Στο προηγούμενο εργαστήριο είχατε υλοποιήσει τον μη αναδρομικό τρόπο υπολογισμού της ακολουθίας αριθμών Fibonacci. Ποια είναι η διαφορά στο χρόνο εκτέλεσης των προγραμμάτων με τον αναδρομικό και μη αναδρομικό τρόπο;