



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

## ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

---

ΜΥΥ-105 / ΠΛΥ-106: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

3ο ΕΡΓΑΣΤΗΡΙΟ

(ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2013-2014)

Διδάσκοντες

Α. Κόντης, Γ. Μανής

Υπεύθυνη Εργαστηρίου

Μαρία Χρόνη

---

### 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 Rene Descartes βρήκε ένα τρίτο ζεύγος (9.363.584, 9.437.056).

Ο Nicolo 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. Πιο συγκεκριμένα θα αναπτύξετε ένα πρόγραμμα το οποίο:

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 υπολογίζει το άθροισμα των διαιρετών του κάθε αριθμού μέσω μιας συνάρτησης την οποία θα ονομάσετε *sumDivisors*. Η συνάρτηση θα δέχεται σαν όρισμα έναν ακέραιο αριθμό και θα επιστρέφει το άθροισμα των διαιρετών του.

γ) Θα βρίσκει στο διάστημα  $[r_1, r_2]$  όλα τα δυνατά ζεύγη των “φίλων” αριθμών και θα τυπώνει ανάλογο μήνυμα (για παράδειγμα το μήνυμα: “Οι αριθμοί 220 και 284 είναι Φιλικοί αριθμοί” ή “Δεν υπάρχουν φίλοι αριθμοί στο διάστημα  $[200, 250]$ ”).

Αποθηκεύστε το αρχείο ως `amicable.py`.

## Άσκηση 2. Λίστες

Στο τμήμα Μηχανικών ΗΥ και Πληροφορικής τα στοιχεία του κάθε φοιτητή στα μαθήματα “Εισαγωγή στους Υπολογιστές” και “Εισαγωγή στο Προγραμματισμό”, είναι αποθηκευμένα σε μία λίστα ως εξής:

$$[firstname_1, lastname_1, grade_{1a}, grade_{1b}].$$

Τα στοιχεία όλων των φοιτητών του πρώτου έτους στα μαθήματα αυτά, είναι επίσης αποθηκευμένα σε μία λίστα ως εξής:

$$students = [ [firstname_1, lastname_1, grade_{1a}, grade_{1b}], [firstname_2, lastname_2, grade_{2a}, grade_{2b}], \dots, [firstname_N, lastname_N, grade_{Na}, grade_{Nb}] ].$$

Μπείτε στη σελίδα του μαθήματος στο `ecourse` και κατεβάστε το αρχείο `grades.py` στο οποίο θα προσθέσετε τον κώδικά σας. Να γραφεί πρόγραμμα που να υλοποιεί τα παρακάτω.

α) Εισάγετε στη λίστα `students`, στις θέσεις 2 και 7 τους παρακάτω φοιτητές, χρησιμοποιώντας μεθόδους των λιστών της `python`.

$$['Michelle', 'Osborn', 8, 8] \text{ και } ['Yen', 'Zhao', 10, 10]$$

β) Τροποποιήστε τη λίστα `students`, έτσι ώστε οι τιμές των βαθμών του κάθε μαθητή και στα δύο μαθήματα, να επιστρέφονται από τη συνάρτηση παραγωγής τυχαίων αριθμών `randint(a, b)`.

γ) Υπολογίστε το μέσο όρο (MO) των βαθμών του κάθε μαθητή και εισάγετε τον αριθμό αυτό στα στοιχεία του συγκεκριμένου μαθητή.

$$students = [[firstname_1, lastname_1, grade_{1a}, grade_{1b}, MO_1], [firstname_2, lastname_2, grade_{2a}, grade_{2b}, MO_2], \dots, [firstname_N, lastname_N, grade_{Na}, grade_{Nb}, MO_N]].$$

δ) Υπολογίστε το μέσο όρο των βαθμών στο μάθημα “Εισαγωγή στο Προγραμματισμό” ( $MO_{Pr}$ ) καθώς και το μέσο όρο στο μάθημα της “Εισαγωγή στους υπολογιστές” ( $MO_Y$ ).

ε) Αποθηκεύστε σε μία λίστα (έστω  $L_{Pr}[]$ ) το πλήθος των φοιτητών για κάθε βαθμό στο μάθημα “Εισαγωγή στο Προγραμματισμό”. Στην πρώτη θέση του πίνακα θα είναι το πλήθος των φοιτητών που έχουν βαθμό 0, στη δεύτερη θέση το πλήθος των φοιτητών με βαθμό 1, ..., στη δέκατη θέση του πίνακα το πλήθος των φοιτητών με βαθμό 10.

στ) Επαναλάβετε τη διαδικασία του ερωτήματος (γ), για το μάθημα “Εισαγωγή στους υπολογιστές” (έστω  $L_Y[]$  η λίστα που θα προκύψει).

ζ) Δημιουργήστε ένα ιστόγραμμα, προκειμένου να απεικονίσετε διαγραμματικά πως κυμάνθηκαν οι βαθμοί των φοιτητών και στα δύο μαθήματα. Συμβολίστε με “\*” το μάθημα “Εισαγωγή στους υπολογιστές” και με “+” το μάθημα “Εισαγωγή στο Προγραμματισμό”. Στο ερώτημα αυτό θα χρησιμοποιήσετε τις λίστες ( $L_{Pr}[]$  και  $L_Y[]$ ) που προέκυψαν από τα ερωτήματα (ε) και (στ). Για παράδειγμα, αν  $L_{Pr} = [0, 0, 2, 5, 8, 14, 6, 3, 2, 1]$ , τότε ένα παραδοσιακό ιστόγραμμα θα είχε τη μορφή του Ιστογράμματος Α της παρακάτω εικόνας. Προκειμένου όμως να το απεικονίσετε στον υπολογιστή είναι ευκολότεροι να το περιστρέψετε κατά  $90^\circ$ , όπως φαίνεται στο Ιστόγραμμα Β.

η) Υπολογίστε και τυπώστε στην οθόνη τα στοιχεία του φοιτητή με το μεγαλύτερο βαθμό στο μάθημα “Εισαγωγή στους Υπολογιστές” καθώς και στο μάθημα “Εισαγωγή στον Προγραμματισμό”.

θ) Στο τέλος του προγράμματός σας, γράψτε την εντολή: `students.sort()` και τυπώστε ξανά τη λίστα `students`. Τι παρατηρείτε; Ταξινομήστε τη λίστα `students` ως προς το βαθμό του κάθε μαθητή κατά αύξουσα σειρά και αποθηκεύστε τη σε μία νέα λίστα.

ι) Ταξινομήστε τη λίστα `students` αλφαβητικά ως προς το επίθετο του κάθε μαθητή και αποθηκεύστε τη σε μία νέα λίστα.

Αποθηκεύστε το αρχείο ως `grades.py`.

