

**ΠΛΥ106 - ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ**

**2<sup>ο</sup> Εργαστήριο**

**Άσκηση 1. Πρώτοι Αριθμοί**

Στα μαθηματικά πρώτος αριθμός (ή απλά πρώτος) είναι ένας φυσικός αριθμός μεγαλύτερος της μονάδας με την ιδιότητα ότι οι μόνοι φυσικοί διαιρέτες του είναι η μονάδα και ο εαυτός του.

Για τον υπολογισμό των πρώτων αριθμών υπάρχουν διάφοροι αλγόριθμοι.

**Αλγόριθμος 1**

*Είσοδος:* Ένας ακέραιος αριθμός  $N$ .

*Έξοδος:* ΝΑΙ, εάν ο αριθμός  $N$  είναι πρώτος, άλλως ΟΧΙ.

1. Εξετάζουμε διαδοχικά όλους τους ακέραιους  $M$ , όπου  $M = 2, 3, \dots, N-1$ .
2. Εάν κάποιος ακέραιος  $M$  είναι διαιρέτης του  $N$ , τότε σταματάμε διότι ο αριθμός  $N$  δεν είναι πρώτος και επιστρέφουμε ΟΧΙ.
3. Αν εξαντληθούν όλοι οι ακέραιοι  $M$  χωρίς να βρεθεί διαιρέτης του  $N$ , τότε ο αριθμός  $N$  είναι πρώτος και επιστρέφουμε ΝΑΙ.

Είναι γνωστό ότι αν ένας αριθμός  $N$  δεν είναι πρώτος τότε έχει (τουλάχιστον) δύο διαιρέτες μεγαλύτερους από  $1$ , και ότι τουλάχιστον ένας διαιρέτης είναι μικρότερος ή ίσος από την τετραγωνική ρίζα του αριθμού.

**Αλγόριθμος 2**

*Είσοδος:* Ένας ακέραιος αριθμός  $N$ .

*Έξοδος:* ΝΑΙ, εάν ο αριθμός  $N$  είναι πρώτος, άλλως ΟΧΙ.

1. Εξετάζουμε διαδοχικά όλους τους ακέραιους  $M$ , όπου  $M = 2, 3, \dots, \sqrt{N}$ .
2. Εάν κάποιος ακέραιος  $M$  είναι διαιρέτης του  $N$ , τότε σταματάμε διότι ο αριθμός  $N$  δεν είναι πρώτος και επιστρέφουμε ΟΧΙ.
3. Αν εξαντληθούν όλοι οι ακέραιοι  $M$  (από το  $2$  έως την τετραγωνική ρίζα του  $N$ ) χωρίς να βρεθεί διαιρέτης του  $N$ , τότε ο αριθμός  $N$  είναι πρώτος και επιστρέφουμε ΝΑΙ.

- α) Να γραφεί ένα πρόγραμμα που θα δέχεται ως είσοδο έναν αριθμό και θα υπολογίζει εάν ο αριθμός είναι πρώτος, υλοποιώντας τον Αλγόριθμο 1. Το πρόγραμμά σας, θα ενημερώνει το χρήστη αν ο αριθμός είναι ή όχι πρώτος με κατάλληλο μήνυμα. Αποθηκεύστε το αρχείο ως *prime1.py*.
- β) Να γραφεί ένα πρόγραμμα που θα υπολογίζει εάν ένας αριθμός είναι πρώτος χρησιμοποιώντας τον Αλγόριθμο 2. Το πρόγραμμα θα δέχεται τον αριθμό ως είσοδο από το χρήστη και θα τον ενημερώνει με κατάλληλο μήνυμα. Αποθηκεύστε το αρχείο ως *prime2.py*.
- γ) Να γραφεί πρόγραμμα που να υπολογίζει και να τυπώνει όλους τους πρώτους αριθμούς μεταξύ 2 και  $n$ :
- Δημιουργήστε μια συνάρτηση `main()`, η οποία για είσοδο  $n$  που θα δίνεται από το χρήστη, θα ελέγχει κάθε έναν από τους αριθμούς 2 έως  $n$ , ποιοι είναι πρώτοι καλώντας τη συνάρτηση `isprime(n)` (δείτε γ-ii) και θα τυπώνει τους αριθμούς αυτούς. Εάν ο χρήστης δίνει λανθασμένη είσοδο τότε θα πρέπει το πρόγραμμά σας να επιστρέφει κατάλληλο μήνυμα λάθους.
  - Δημιουργήστε μία συνάρτηση `isprime(n)` που θα ελέγχει αν ένας αριθμός είναι πρώτος και θα επιστρέφει `True` ή `False`. Υλοποιήστε τη συνάρτηση `isprime(n)` χρησιμοποιώντας τον Αλγόριθμο 1 ή τον Αλγόριθμο 2. Αποθηκεύστε το αρχείο ως *prime3.py*.
- δ) Τροποποιήστε κατάλληλα το πρόγραμμά σας έτσι ώστε ο χρήστης δίνοντας ως είσοδο έναν ακέραιο θετικό αριθμό να τον αναλύει σε γινόμενο πρώτων παραγόντων. Αποθηκεύστε το αρχείο ως *prime4.py*.

## Άσκηση 2. Fibonacci Αριθμοί

Η ακολουθία των αριθμών Fibonacci έχει τη μορφή  $(f_0, f_1, \dots, f_{i-2}, f_{i-1}, f_i, \dots)$  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  $f_0 = 0, f_1 = 1, f_2 = 1$ , 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.

- Να γραφεί ένα πρόγραμμα που να υπολογίζει όλους τους αριθμούς Fibonacci που είναι μικρότεροι από έναν αριθμό  $n$ , που δίδεται ως είσοδος από το χρήστη.
- Να γραφεί ένα πρόγραμμα που να υπολογίζει και τυπώνει τους Fibonacci αριθμούς που είναι πρώτοι αριθμοί.

### Άσκηση 3. Ύψωση αριθμού σε δύναμη (Προερατικό)

Η ύψωση σε δύναμη είναι μαθηματική πράξη, που συμβολίζεται ως  $a^n$  και περιλαμβάνει δύο αριθμούς, την βάση  $a$  και τον εκθέτη  $n$ .

Να γραφεί πρόγραμμα που να δέχεται ως είσοδο δύο αριθμούς  $a$ ,  $n$  και να υπολογίζει την δύναμη  $a^n$ , όπου  $n \leq 0$  ή  $n \geq 0$  και  $a \leq 0$  ή  $a \geq 0$ .