

**Ενότητα: Λίστες**

**Σχέδιο Δραστηριότητας: Το κόσκινο του Ερατοσθένη**

Οι πρώτοι αριθμοί είναι οι ακέραιοι αριθμοί που δεν έχουν άλλους διαιρέτες εκτός από το 1 και τον εαυτό τους, π.χ. 2,3,5,7,11.

**Δραστηριότητα 1**

**Βήμα 1:** Ανοίξτε ένα νέο αρχείο με όνομα `primes.py` και συμπληρώστε τον κώδικα της παρακάτω συνάρτησης, έτσι ώστε να επιστρέφει `True` όταν ο αριθμός είναι πρώτος αλλιώς `False`. Θα πρέπει να ελέγξετε αν υπάρχει αριθμός μικρότερος του `n` ο οποίος να τον διαιρεί ακριβώς. Αφού φορτώσετε το αρχείο στον διερμηνευτή πειραματιστείτε με τη συνάρτηση για διάφορους αριθμούς.

```
def isPrime(n):
    for i in range(____, ____):
        if (____):
            return _____
    return _____
```

**Βήμα 2:** Χρησιμοποιώντας την `isPrime` να γράψετε μια συνάρτηση `allPrimes( n )` η οποία να επιστρέφει το πλήθος των πρώτων αριθμών μέχρι τον αριθμό `n`.

Να υπολογίσετε πόσο χρόνο θα κάνει η `allPrimes` για `n = 1000, 10000, 100000`, με χρήση της συνάρτησης `timer` της βιβλιοθήκης `cs`.

**timer( function, data )** : υπολογίζει και επιστρέφει τον χρόνο εκτέλεσης της συνάρτησης με όρισμα τα συγκεκριμένα δεδομένα

Θα πρέπει να συμπεριλάβετε την `timer` με τη δήλωση `from cs import timer` αφού πρώτα κατεβάσετε στον υπολογιστή σας το αρχείο `cs.py`.

**Βήμα 3:** Ένα πολύ χρήσιμο θεώρημα μας λέει ότι αρκεί να ελέγξουμε μέχρι τη ρίζα του αριθμού `n` για να σιγουρευτούμε αν είναι πρώτος. Τροποποιήστε την `isPrime` έτσι ώστε να ελέγχει μέχρι τη ρίζα. Στη συνέχεια να υπολογίσετε πάλι την απόδοση της `allPrimes` για `n=1000, 10000, 100000`. Τι παρατηρείτε;

**Δραστηριότητα 2**

**Βήμα 1:** Δώστε τις παρακάτω εντολές στον διερμηνευτή της Python και σχολιάστε τα αποτελέσματα:

```
>>> array = 10*[0]
>>> array[0] = array[9] = 28
>>> print( array )
>>> array = array + [6] + [496]
>>> print( array )
>>> array.append( 8128 )
>>> print( len( array ) )
```

**Βήμα 2:** Να δημιουργήσετε έναν πίνακα λογικών τιμών (τύπου `bool`) 100 θέσεων. Στη συνέχεια να θέσετε σε όλα τα πολλαπλάσια του 2 την τιμή `False`.

### Δραστηριότητα 3

Ένας εξαιρετικά δημοφιλής αλγόριθμος για εύρεση πρώτων αριθμών είναι το κόσκινο του Ερατοσθένη. Η ιδέα έχει ως εξής:

1. Αρχικά δημιουργούμε μια λίστα (κόσκινο) με όλους τους αριθμούς που θέλουμε να ελέγξουμε εκτός από το 0 και το 1. Στη δική μας υλοποίηση πρόκειται για μια λίστα λογικών τιμών όπου το στοιχείο  $a[i]$  έχει την λογική τιμή True αν είναι πρώτος και False αν δεν είναι. Αρχικά όλοι οι αριθμοί εκτός των 0,1 είναι εν δυνάμει πρώτοι, άρα True.
2. Ξεκινάμε με τον αριθμό 2 ο οποίος γνωρίζουμε ότι είναι πρώτος.
3. Θέτουμε την τιμή/ένδειξη False σε όλα τα πολλαπλάσια του 2, που σημαίνει ότι δεν είναι πρώτοι.
4. Κάνουμε το ίδιο και για το 3. Το 4 έχει ήδη αφαιρεθεί ως πολλαπλάσιο του 2 (είναι False) άρα ο επόμενος αριθμός στη λίστα είναι το 5.

Η παραπάνω διαδικασία συνεχίζεται μέχρι να μη μείνουν άλλα στοιχεία στην αρχική λίστα.

Αφού ολοκληρώσετε την ανάπτυξη μιας συνάρτησης `primeSieve( n )` η οποία επιστρέφει όλους τους πρώτους αριθμούς από το 2 έως το  $n$  υλοποιώντας το κόσκινο του Ερατοσθένη να την συγκρίνετε με την `allPrimes` όσον αφορά την ορθότητά και την αποδοτικότητά της.

**Υπόδειξη:** Στο τέλος οι θέσεις στις οποίες έχει μείνει η τιμή True είναι πρώτοι αριθμοί.