



Οικονομικό Πανεπιστήμιο Αθηνών, τμήμα Πληροφορικής  
Μάθημα: Εισαγωγή στον Προγραμματισμό Υπολογιστών  
Ακαδημαϊκό έτος: 2019–20  
Διδάσκων: Α. Δημάκης

## Επαναληπτική Εξέταση Σεπτεμβρίου: διάρκεια 1 ½ ώρα

Σύνολο μονάδων: 10, Άριστα: 10

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ!**

Επίθετο: \_\_\_\_\_ Όνομα: \_\_\_\_\_ ΑΜ: \_\_\_\_\_

**1<sup>η</sup> άσκηση. (2)** Συμπληρώστε τα κενά. (Γράψτε «ΜΗΝΥΜΑ ΛΑΘΟΥΣ» εάν πρόκειται να εμφανιστεί κάποιο μήνυμα λάθους, ή «ΚΕΝΟ» αν δεν εμφανιστεί κάτι. Εάν εμφανιστούν πολλαπλές γραμμές δώστε τις στον διαθέσιμο χώρο.)

**Απάντηση:**

```
>>> 1+1 * 'world'
ΜΗΝΥΜΑ ΛΑΘΟΥΣ
>>> ls = [1, [2, 3], 4]
>>> len(ls)
3
>>> ls2 = ls
>>> ls[2] = -4
>>> ls
[1, [2, 3], -4]
>>> ls2
[1, [2, 3], -4]
>>> ls3 = ls[:]
>>> ls3[0] = -1
>>> ls[0]
1
>>> ls3[1][1] = -3
>>> ls[0]
1
>>> ls[1]
[2, -3]
>>> def x(y):
    def y(x):
        return x
    return y
>>> x(lambda x:x+1)(1)
1
>>> x(x(lambda x:x+1))(1)
1
>>> for x in (2*x for x in (3*x for x in 'ba')):
    print(x)
bbbbbb
aaaaaa
```

**2<sup>η</sup> άσκηση.** Στην άσκηση αυτή θα δώσετε δύο υλοποιήσεις της συνάρτησης `mix` έτσι ώστε η `mix(s)` να επιστρέφει συμβολοσειρά που αποτελείται (από αριστερά προς δεξιά) από τον 1<sup>ο</sup> χαρακτήρα της συμβολοσειράς `s`, τον 1<sup>ο</sup> από δεξιά (δηλ. τον τελευταίο), τον 2<sup>ο</sup> από αριστερά, τον 2<sup>ο</sup> από δεξιά,... κτλ. έως τον `n`-οστό από αριστερά και τον `n`-οστό από δεξιά, όπου `n` το μήκος της συμβολοσειράς.

Για παράδειγμα,

```
>>> mix('0123456789')
'09182736455463728190'
>>> mix('abcdefgh')
'ahbgcfdfeedfcgbha'
>>> mix('abc')
'acbbca'
>>> mix('eg')
'egge'
>>> mix('e')
'ee'
```

a) **(2)** Υλοποίηση με επαναληπτικό υπολογισμό ή *comprehensions*\*: Συμπληρώστε τα κενά του κώδικα που ακολουθεί χρησιμοποιώντας εντολές `while` ή `for` – χωρίς αναδρομικές κλήσεις.

\* *comprehensions* είναι εκφράσεις της μορφής `[x*x for x in range(1, 10)]`, `(x+1 for x in [1,2,3] if x % 1 == 1)`, κτλ.

**Απάντηση:**

```
def mix(s):
    """ΥΛΟΠΟΙΗΣΗ ΜΕ ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ while ή for """
    """ΣΥΜΠΛΗΡΩΣΤΕ ΤΟ ΣΩΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ."""

    from functools import reduce
    from operator import add

    return reduce(add, (x+y for (x,y) in zip(s, s[::-1])))
```

- b) **(3)** Υλοποίηση με αναδρομή: Συμπληρώστε τα κενά του κώδικα που ακολουθεί χρησιμοποιώντας αναδρομικές κλήσεις συναρτήσεων – χωρίς εντολές επανάληψης `while`, `for` και `comprehensions`.

Απάντηση:

```
def mix(s):
    """ΥΛΟΠΟΙΗΣΗ ΜΕ ΑΝΑΔΡΟΜΗ """
    """ΣΥΜΠΛΗΡΩΣΤΕ ΤΟ ΣΩΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ."""

    if len(s) <= 2:
        return s + s[::-1]

    return s[0]+s[-1] + mix(s[1:-1]) + s[-1]+s[0]
```

**3<sup>η</sup> άσκηση. (3)** Συμπληρώστε στο παρακάτω πλαίσιο τον ορισμό της συνάρτησης `gather` όπου η κλήση `gather(ls)` συγκεντρώνει όλα τα στοιχεία της λίστας `ls` με ίσες τιμές ώστε να τοποθετούνται σε διαδοχικές θέσεις. Η `gather(ls)` επιστρέφει `None`.

Για παράδειγμα,

```
>>> a = [2, 1, 'hello', 3, 10, 'hello', 10, 2, 10]
>>> gather(a)
>>> a
[2, 2, 10, 10, 10, 'hello', 'hello', 3, 1]
```

Η ακριβής θέση των ίσων στοιχείων δεν έχει σημασία, π.χ, στο παραπάνω παράδειγμα εξίσου σωστό αποτέλεσμα θα ήταν:

```
[2, 2, 1, 'hello', 'hello', 3, 10, 10, 10]
```

Απάντηση:

```
"""ΔΩΣΤΕ ΤΟΝ ΟΡΙΣΜΟ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ gather."""  
  
def gather(ls):  
    for i in range(len(ls)):  
        nexti = -1  
        for j in range(i+1, len(ls)):  
            if ls[j] == ls[i]:  
                nexti = j  
                break  
        if i + 1 < len(ls):  
            ls[i+1], ls[j] = ls[j], ls[i+1]
```