



Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής
Μάθημα: Εισαγωγή στον Προγραμματισμό Υπολογιστών
Ακαδημαϊκό έτος: 2018–19
Διδάσκων: Α. Δημάκης

Τελική Εξέταση: διάρκεια 2 ώρες

Όλες οι ασκήσεις είναι βαθμολογικά ισοδύναμες

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!

Επίθετο: _____ Όνομα: _____ ΑΜ: _____

1^η άσκηση. Συμπληρώστε τα κενά. Συμπληρώστε ΜΗΝΥΜΑ ΛΑΘΟΥΣ εάν πρόκειται να εμφανιστεί κάποιο μήνυμα λάθους, ή KENO αν δεν εμφανιστεί κάτι.

```
>>> min = 5
KENO
>>> def f(x):
    return x / 3 * 3 + 1
>>> f(9)
10.0
>>> x, f = f, 9
>>> x(f)
10.0
>>> def f(g, x):
    def h(y):
        return g(x, y)
    return h
>>> h = f(max, 3)
>>> h(2)
3
>>> h(4)
4
>>> def sum_items(f, ls):
    return sum(f(x) for x in ls)
>>> sum_items(lambda x: x*x, [2, -2])
8
>>> # πλήθος εμφανίσεων του 'apple' στη λίστα ['apple', 'orange', 'apple']:
>>> sum_items(lambda x: x == 'apple', ['apple', 'orange', 'apple'])
2
```

2^η άσκηση. Συμπληρώστε το κενό στον κώδικα της `most_frequent_item` η οποία επιστρέφει την τιμή που περιέχεται τις περισσότερες φορές στη λίστα που δίνεται στο όρισμα. Για παράδειγμα, η κλήση `most_frequent_item([3, 2, 3, -2])` επιστρέφει 3. (Αν υπάρχουν πολλαπλές τέτοιες τιμές, επιλέξτε οποιαδήποτε από αυτές.)

```
>>> def most_frequent_item(ls):  
    """ΣΥΜΠΛΗΡΩΣΤΕ ΤΟ ΣΩΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ."""  
    return max(ls, key = lambda x: sum(item == x for item in ls))
```

3^η άσκηση. Συμπληρώστε τα κενά στον κώδικα. Θα πρέπει να χρησιμοποιήσετε αναδρομή – χωρίς εντολές επανάληψης και *comprehensions*.

```
def sum_items(f, ls):  
    """Επιστρέφει την τιμή της έκφρασης sum(f(x) for x in ls).  
  
    f -- συνάρτηση που δέχεται ένα όρισμα  
    ls -- λίστα (list)  
  
    Παραδείγματα:  
>>> sum_items(lambda x:x, [1, 2, 3]) # άθροισμα στοιχείων  
6  
>>> sum_items(lambda x: x%2 != 0, [1, 2, 3]) # πλήθος περιττών  
2  
    """  
    """ΣΥΜΠΛΗΡΩΣΤΕ ΤΟ ΣΩΜΑ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ."""
```

```
return 0 if len(ls) == 0 else f(ls[0]) + sum_items(f, ls[1:])
```

4^η άσκηση. Συμπληρώστε τα κενά του κώδικα που ακολουθεί έτσι ώστε να παράγεται το αναγραφόμενο αποτέλεσμα, με την ελάχιστη δυνατή επανάληψη παρόμοιου κώδικα.

```
>>> class Robot:
    """ΣΥΜΠΛΗΡΩΣΤΕ ΤΟΝ ΟΡΙΣΜΟ ΤΗΣ ΤΑΞΗΣ Robot."""
    def __init__(self, name):
        self.name = name
    def identify_yourself(self):
        return 'Hello, my name is '+ self.name + '.'
```

```
>>> r2d2 = Robot('R2-D2')
>>> c3po = Robot('C3-PO')
>>> r2d2.identify_yourself()
'Hello, my name is R2-D2.'
>>> c3po.identify_yourself()
'Hello, my name is C3-PO.'
```

```
>>> # ΣΥΜΠΛΗΡΩΣΤΕ ΚΩΔΙΚΑ ΣΤΟ ΚΕΝΟ:
>>> class TalkingRobot(Robot):
    def __init__(self, name, phrases):
        Robot.__init__(self, name)
    def phrase_iterator(ls):
        i = 0
        while True:
            yield ls[i]
            i = (i + 1) % len(ls)
    self.iterator = phrase_iterator(list(phrases))
def talk(self):
    return next(self.iterator)

>>> t1000 = TalkingRobot('T1000', ["I'll be back.", 'Asta la vista
baby.'])
>>> t1000.identify_yourself()
'Hello, my name is T1000.'
>>> t1000.talk()
"I'll be back."
>>> t1000.talk()
'Asta la vista baby.'
>>> r2d2 = TalkingRobot('R2D2', ['piou-piou', 'dou-rou-dou-dit'])
>>> r2d2.talk()
'piou-piou'
>>> r2d2.talk()
'dou-rou-dou-dit'
```