Προβλήματα εξάσκησης στην ύλη της Διαλέξης 5

Σε όλα τα παραπάνω προβλήματα δεν πρέπει να χρησιμοποιήσετε εντολές επανάληψης, όπως `while` *και* `for`*.*

1. Δώστε τον ορισμό της αναδρομικής συνάρτησης `print_many` όπου η κλήση `print_many(n, s)` εμφανίζει `n` φορές τη συμβολοσειρά `s`, π.χ.,
```
>>> print_many(3, 'Hello world')
Hello world
Hello world
Hello world
```

2. Συμπληρώστε τον ορισμό της αναδρομικής συνάρτησης `sum_naturals` η οποία υπολογίζει την τιμή του αθροίσματος $1 + 2 + \cdots + n$.

```
def sum_naturals(n):
    """ Computes 1 + 2 … + n.

    n – positive integer
    Returns 1 +2 + … + n.

    >>> sum_naturals(3)
    6
    """
    if __________:
        ____________
    else:
        ____________
```

3. Αλλάξτε τον ορισμό της συνάρτησης `sum_naturals` της άσκησης 1 ώστε να λειτουργεί με *tail recursion*, δηλαδή η τελευταία εντολή σε κάθε κλήση της `sum_naturals` να είναι η αναδρομική κλήση.

4. Χωρίς τη χρήση υπολογιστή, απαντήστε τι θα εμφανίσει στο τερματικό το πρόγραμμα: (*Αν σας βοηθάει, μπορείτε να σημειώνετε τα πλαίσια και τα περιεχόμενά τους καθώς εκτελείται κάθε εντολή.)*

```
def what_do_i_do(n):
    if n <= 1:
        print('*')
    else:
        what_do_i_do(n // 2)
        print(n * '*')
        what_do_i_do(n // 2)

what_do_i_do(8)
```

5. Δώστε τον ορισμό αναδρομικής συνάρτησης `contains(n, x)` όπου επιστρέφει `True` εάν ο αριθμός `n` περιέχει το ψηφίο `x`, αλλιώς `False`, π.χ.,

```
>>> contains(123456, 3)
True
>>> contains(511982, 5)
True
>>> contains(511982, 7)
False
```

6. Δίνεται ο ορισμός συναρτήσεων `foo` και `goo` που εκτελούν *αμοιβαία αναδρομή*:

```
def foo(n):
    if n == 0:
        return 0
    elif n % 2 == 0:
        return goo(n // 2)
    else:
        return goo(n - 1) + foo(n - 1)

def goo(n):
    if n <= 1:
        return 1 - n
    else:
        return 1 + foo(n - 1)
```

Αλλάξτε τον ορισμό της `foo` ώστε να εκτελεί απλή αναδρομή, δηλαδή χωρίς κλήση στη `goo`.

7. Στα προβλήματα 4, 5, 6 εξάσκησης στη Διάλεξη 3 γράψατε τρεις συναρτήσεις που εμφανίζουν στο τερματικό:

```
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
I promise to do my practice problems.
```

```
1. I promise to do my practice problems.
2. I promise to do my practice problems.
3. I promise to do my practice problems.
4. I promise to do my practice problems.
5. I promise to do my practice problems.
6. I promise to do my practice problems.
7. I promise to do my practice problems.
8. I promise to do my practice problems.
9. I promise to do my practice problems.
10. I promise to do my practice problems.
```

```
1. I promise to do my practice problems.
2. I promise to follow rule 1
3. I promise to follow rule 2
4. I promise to follow rule 3
5. I promise to follow rule 4
6. I promise to follow rule 5
7. I promise to follow rule 6
8. I promise to follow rule 7
9. I promise to follow rule 8
10. I promise to follow rule 9
```

Γράψτε νέες συναρτήσεις με τα ίδια ακριβώς αποτελέσματα χρησιμοποιώντας αναδρομικούς υπολογισμούς – χωρίς εντολές επανάληψης (π.χ, `while`) -.