INFORMATICS 2 - 3RD MIDTERM (EXERCISES)

Write the solutions in a textfile called `NNNNNN.py`, where `NNNNNN` is your neptun code, and send this file to the address `sa42bme@gmail.com` at most 40 minutes after the start!

Exercise 1. Define a function `apply()` that can be called with n arguments, where $n > 0$. The first argument will be a function of $n-1$ arguments, and `apply()` applies this function to the rest of the arguments, and returns with the result of this application. For example:

```
>>> apply(lambda x,y : x+y,2,3), apply(lambda: 42)
(5, 42)
```

Exercise 2. Define a function `appall()` that can be called with any number of arguments, which, apart from the first, are all functions of one argument.

`appall(c, f1, f2,..., fn)`

should return the list

`[f1(c), f2(c),..., fn(c)]`

For example:

```
>>> appall(5,lambda x: 2*x, lambda x: x**2, lambda x: -x)
[10, 25, -5]
```

Exercise 3. Define a function `deeprev()` whose only argument will be list, whose members will be numbers and lists whose members are integers and lists, whose members are numbers and lists, etc. `deeprev()` should return the reverse list but in the extended sense that every list that occurs in this list, at any level, is also reversed. For example:

```
>>> deeprev([8,3,[1,3, [5,6,7],6],4,2,1])
[1, 2, 4, [6, [7, 6, 5], 3, 1], 3, 8]
```

1

```
def apply(fn,*args):
    return fn(*args)
```

2

```
def appall(arg, *fncs):
    return [f(arg) for f in fncs]
```

3

```
def deeprev(x):
    if not isinstance(x,list) or x == []: return x
    return [deeprev(i) for i in x][::-1]
```