Informatics 2, 2nd midterm (2017-04-10)

The answers should fit next to the questions, if you used a separate paper let us know clearly!

1. Write a python class for representing lines on the plane. Define three methods: *(6 points)*

constructor store two lists, both of length 2: the vector of the normal vector and one point on the line. If the normal vector os a null vector then raise a `ValueError` exception.

repr method which returns the string representation of the equation of the line in the following format: $2*x+y=3$

rotate this method should return a new line which is the rotated version of the original line by $+90°$ (counter-clockwise).

```python
class Line(object):
    def __init__(self, n, P):




    def __repr__(self):




    def rotate(self):
```

2. Theoretical questions *(3 points)*

a) How to write a function with variadic number of arguments?

b) What are the parameters of the `map` function. What type is the returned value?

c) How to inpherit from a class (what to write and where)?

3. What happens if we iterate over the following class? *(3 points)*

```python
class myclass(object):
    def __init__(self, l, n):
        self.l = l
        self.n = n
    def __iter__(self):
        self.index1 = 0
        self.index2 = -1
        return self
    def next(self):
        self.index2 += 1
        if self.index2 >= len(self.l):
            self.index1 += 1
            self.index2 = 0
            if self.index1 >= self.n:
                raise StopIteration
        return self.l[self.index2]
```

4.

a) What will be printed after these commands? *(2 points)*

```python
l = range(10)
f = lambda x: x%2==0
print filter(f, l)
```

b) Write a functional (lambda) expression which returns a two element list for a list input. The result should contain the sum of positive numbers and the absolute value of the sum of negative numbers. *(2 points)*

Example: `[1, -2, 3, -4] ↦ [4, 6]`

5. In the following code there are 4 mistakes, find those! *(4 points)*

```python
class A:
    def __init__(x, y):
        self.value = x + y
    def add(self,x):
        return self.x + x
    def __repr__(self):
        return self.value
    self.x = 0
a=A(1,2)
print a.add(4)
print a
```