

Task 1: Special objects

Can you fill in the blanks?

```
# write code here
print(a) # prints 1
print(b) # prints 1
print(a-b) # prints 2
```

Hint 1: var `a` and `b` are not int's

Hint 2: Give subtraction another meaning

Task 2: CIDR subtraction

With the knowledge of assignment 1, can you fill in the blanks?

```
# write code here
print(a) # prints 192.168.1.0/24
print(b) # prints 192.168.1.16/29
print(a-b) # prints ['192.168.1.0/28', '192.168.1.24/29', '192.168.1.32/27', '192.168.1.64/26', '192.168.1.128/25']
```

In case you need help ..

If you don't need it just skip this section. Please read if you are not familier with CIDRs: <http://software77.net/cidr-101.html>

Also let's consider example:

```
...
print(a) # prints 192.168.1.0/29
print(b) # prints 192.168.1.0/31
print(a-b) # prints ['192.168.1.2/31', '192.168.1.4/30']
```

First CIDR to range (assume 192.168 is x.x for simplicity)

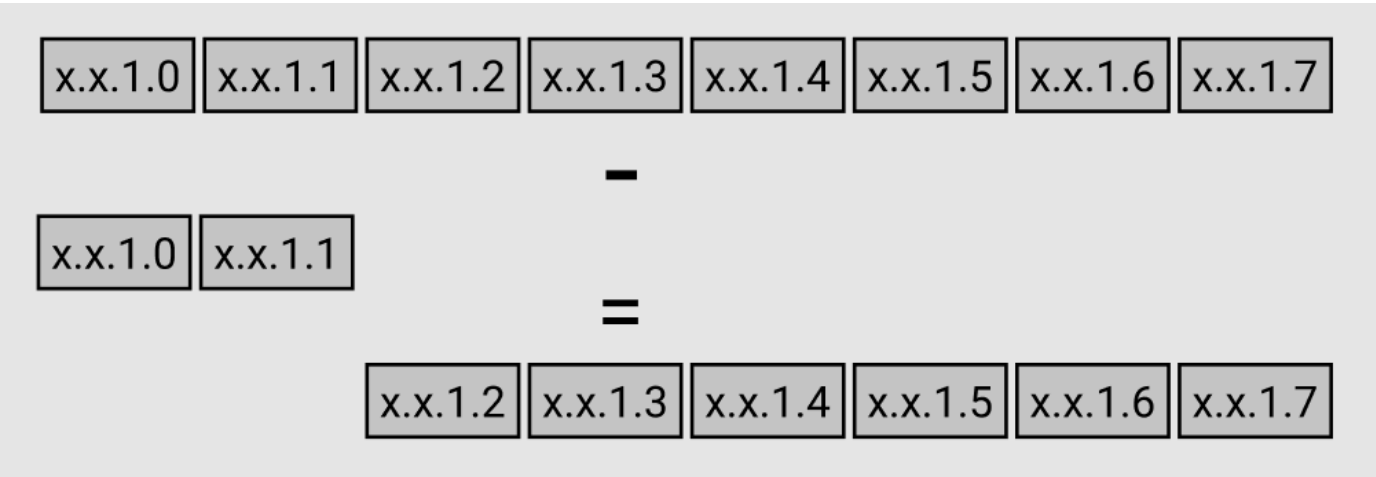
x.x.00000001.00000000 / 29 192.168.1.0/29 x.x.11111111.11111000 mask: / 29 x.x.1.0 - x.x.1.7

Subtracted CIDR to range:

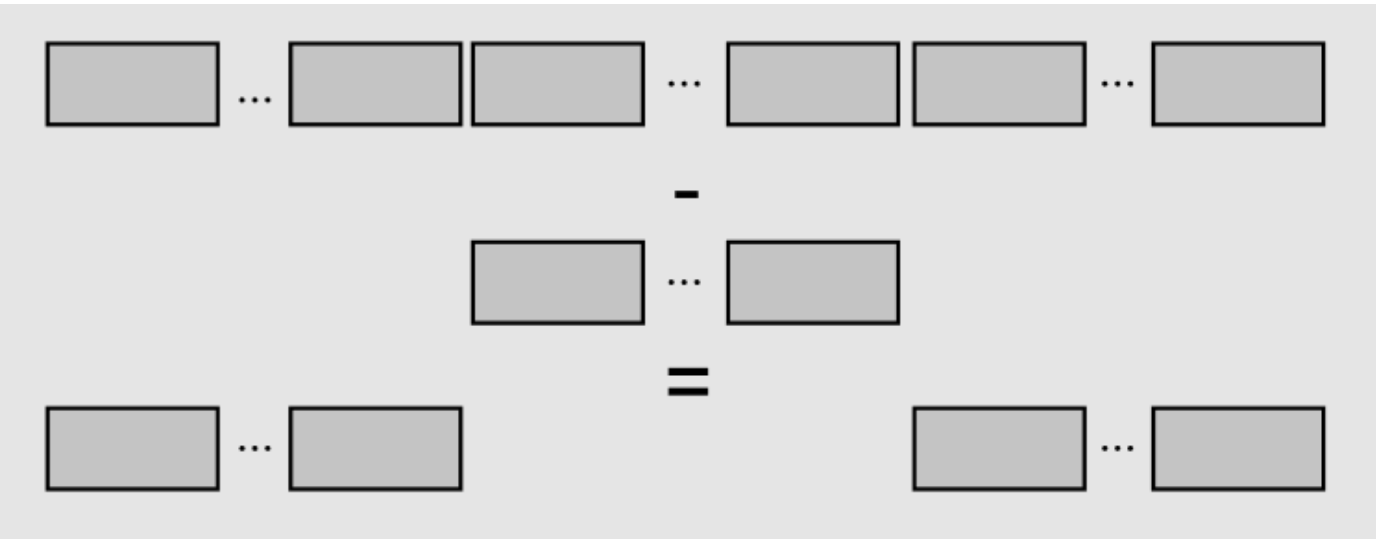
x.x.00000001.00000000 / 31 192.168.1.0/31 x.x.11111111.11111110 mask: / 31 x.x.1.0 - x.x.1.1

Subtract ranges to get result range: x.x.1.0 - x.x.1.7 - x.x.1.0 - x.x.1.1 = x.x.1.2 - x.x.1.7

NOTE: this subtraction produces one range



In another example you will face with situation which will produce 2 ranges:



Convert range to CIDR:

```
x.x.1.00000010 \__ x.x.1.2/31
x.x.1.00000011 /
x.x.1.00000100 \
x.x.1.00000101 \__ x.x.1.4/30
x.x.1.00000110 /
x.x.1.00000111 /
```