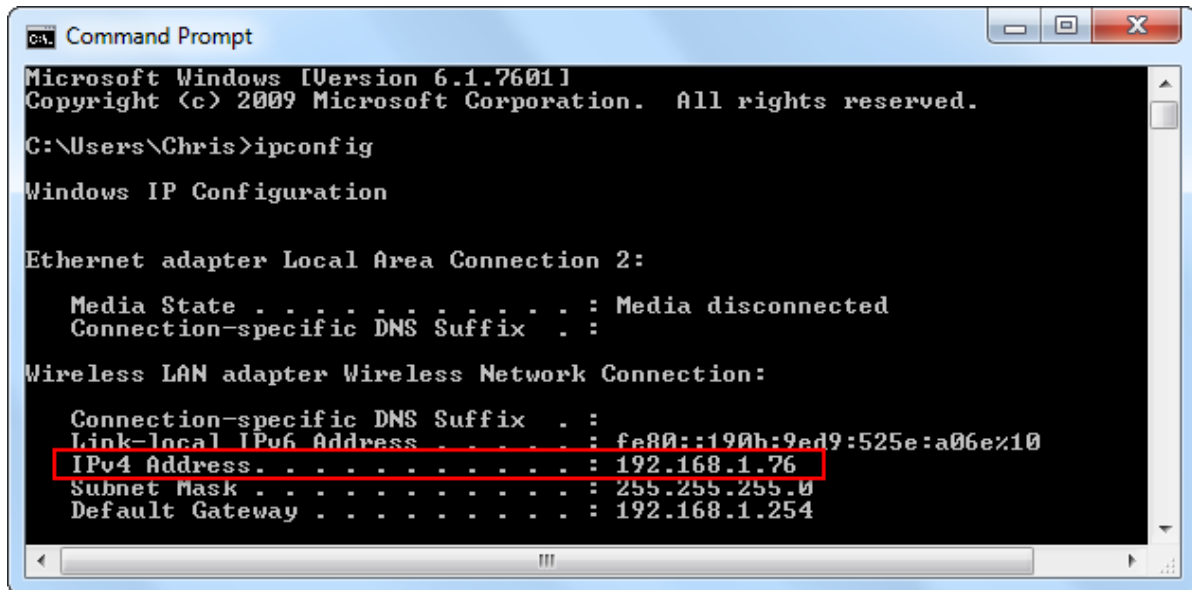


## IP Address (ip)

Marco is trying to hack the network in his university. In order to do so, he needs to send a malicious packet to each computer that might be connected to the same network.

Just to be sure, Marco decided to send the packet to a range of *IP addresses*. An IP address is a set of four “octets” separated by dots (each “octet” is an integer in the 0 to 255 inclusive range). For example, 127.0.0.1, 192.168.2.3 and 255.255.255.255 are valid IP addresses, while something like 10.20.300.4000 is not.



```

C:\> Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Chris>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:


    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::190b:9ed9:525e:a06e%10
    IPv4 Address. . . . . : 192.168.1.76
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.254
  
```

Figure 1: Using the ipconfig command on Windows to obtain the IP.

Given an IP address, we can *increment* it and go to the next address in this way: if the last octet is less than 255 then we increment it by 1, otherwise we set it to 0 and increment the second-last octet (repeating this in a similar way to the left if necessary). This is similar to what happens when we increment a number by 1: if the last digit is 9 then we need to set it to 0 and increment the second-last digit (which could be 9 as well and so on).

For example, if Marco wanted to send the malicious packet to all IPs in the range that goes from 192.168.1.0 to 192.168.1.255, he would need to send a total of 256 packets. If the range was 192.168.1.0 to 192.168.2.0, instead, he would have to send 257 packets.

Calculate, given a valid range of addresses, how many packets Marco needs to send.

 Among the attachments of this task you may find a template file `ip.*` with a sample incomplete implementation.

## Input

The first line contains the first IP address. The second line contains the second IP address.

## Output





You need to write a single line with an integer: the number of packets that Marco should send.

## Constraints

- Both addresses in input are valid IP addresses.
- The first address is lower than or equal to the second.

## Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** (0 points)      Examples.  

- **Subtask 2** (20 points)      The IP endpoints only differ in the last octet (e.g. 192.168.0.21 to 192.168.0.147)  

- **Subtask 3** (50 points)      The given IP range spans at most 1 000 000 addresses.  

- **Subtask 4** (30 points)      No additional limitations.  


## Examples

input	output
192.168.1.0 192.168.1.255	256
192.168.1.0 192.168.2.0	257
0.1.2.3 200.201.202.203	3368601801