COMMAND TO RUN THIS PROGRAM:'

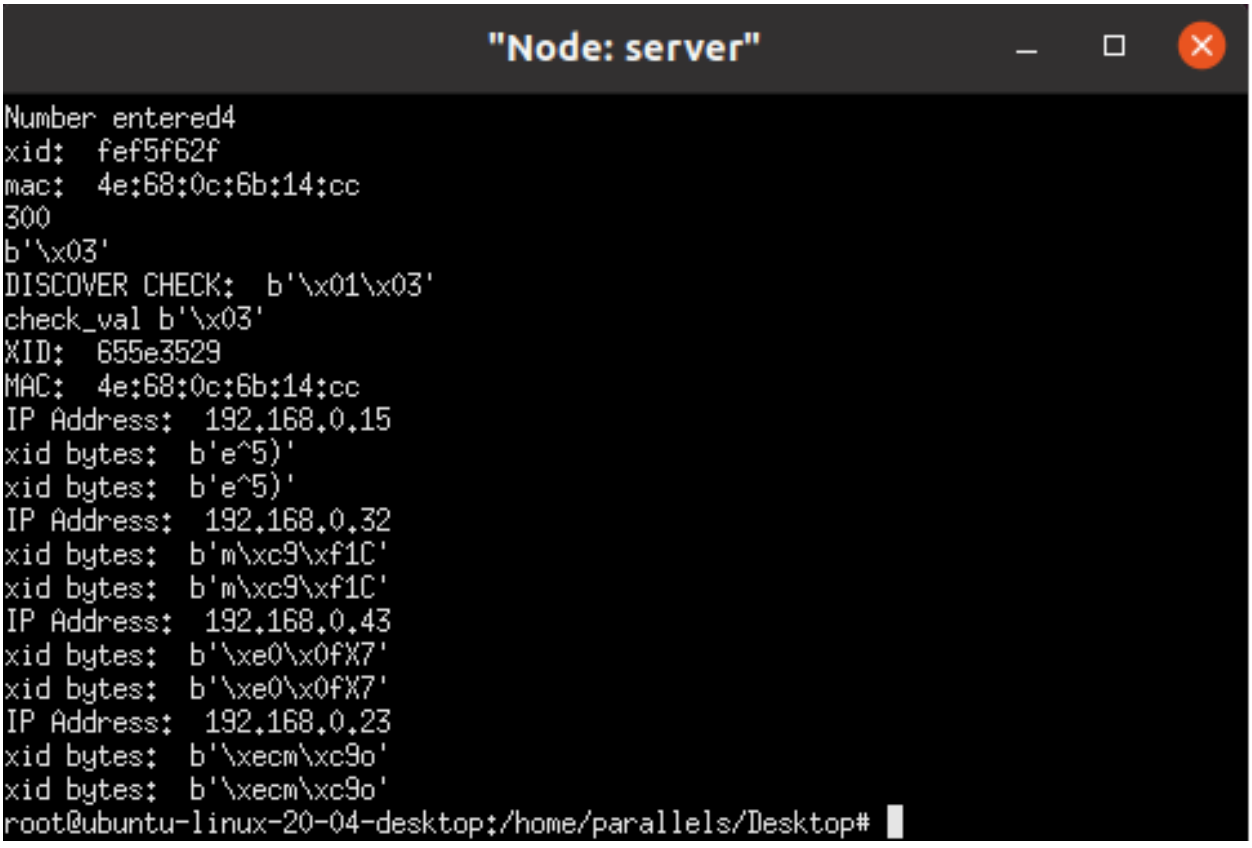
To run the mytopo1.py file.

```
parallels@ubuntu-linux-20-04-desktop:~/Desktop$ sudo mn --custom=mytopo1.py --topo=mytopo
```

Then assign a static IPv4 address to server-eth0 using the following commands: **server ip address add 192.168.0.1/24 dev server-eth0**, **server ip route add default dev server-eht0**. Next open a "Node server" in the mininet environment using the command **xterm server**. Now run the deserver.py file. On the "Node server', with command **python3 dhserver.py**.

```
root@ubuntu-linux-20-04-desktop:/home/parallels/Desktop# python3 dhserver.py
Enter the number of clients 4
Number entered 4
```

Assign each clients IP address using the following command **clientX dhclient -4 -v clientX-eht0**. Where 'X' is the number of clients setup in the 'mytopo1.py' file. Afterward your "Node server" terminal should look similar to the image below.



RESULTS:

After all clients have received their IP address you can verify each one by using the following command **clientX ip address** in the mininet terminal.

```
mininet> client0 ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: client0-eth0@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group de
fault qlen 1000
    link/ether 4e:68:0c:6b:14:cc brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.0.15/24 brd 192.168.0.255 scope global dynamic client0-eth0
       valid_lft 86286sec preferred_lft 86286sec
    inet6 fe80::4c68:cff:fe6b:14cc/64 scope link
       valid_lft forever preferred_lft forever
```

In the end, all four IP address will belong to the same subnet of /24.

CODE:

The code for DHCPOFFER. The offer acknowledgment comes from the following bytes \x35\x01\x02

```
def DCHPOFFER(ip_address,xid,mac):
        pkt = b''
        pkt += b'\x02' #Op code
        pkt += b'\x01' #htype
        pkt += b'\x06' #lenght
        pkt += b'\x00' #hops
        print("xid bytes: ",bytes.fromhex(xid))
        pkt += bytes.fromhex(xid) #xid
        #print("pkt: ", pkt) # b'\x02\x01\x06\x00\xa9\x85\x94\x0b'
        pkt += b'\x00\x00' #secs
        pkt += b'\x00\x00' #flags
        pkt += b'\x00\x00\x00\x00' # Client IP Address ciadder
        pkt += Ip_offer(ip_address) # Yiadder
        pkt += Ip_offer('192.168.0.1') # Server Ip siadder siadder
        pkt += Ip_offer('0.0.0.0') #Relay Ip Address giadder
        pkt += bytes.fromhex(str(mac).replace(':','')) # chaddr

        pkt += b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
        pkt += b'\x00' * 67
        pkt += b'\x00' *125
        pkt += b'\x63\x82\x53\x63'


        pkt += b'\x35\x01\x02'
        pkt += b'\x01\x04\xff\xff\xff\x00'
        pkt += b'\x03\x04'+Ip_offer('192.168.0.1')
        pkt += b'\x33\x04\x00\x01\x51\x80'
        pkt += b'\x51\x04'+Ip_offer('192.168.0.1')
        pkt += b'\xff'
        return pkt
```

The code for DHCPACK. The ack acknowledgment comes from the following bytes \x35\x01\x05

```
def DCHPACK(ip_address,xid,mac):
        pkt = b''
        pkt += b'\x02' #Op code
        pkt += b'\x01' #htype
        pkt += b'\x06' #lenght
        pkt += b'\x00' #hops
        print("xid bytes: ",bytes.fromhex(xid))
        pkt += bytes.fromhex(xid) #xid
        #print("pkt: ", pkt) # b'\x02\x01\x06\x00\xa9\x85\x94\x0b'
        pkt += b'\x00\x00' #secs
        pkt += b'\x00\x00' #flags
        pkt += b'\x00\x00\x00\x00' # Client IP Address ciadder
        pkt += Ip_offer(ip_address) # Yiadder
        pkt += Ip_offer('192.168.0.1') # Server Ip siadder siadder
        pkt += Ip_offer('0.0.0.0') #Relay Ip Address giadder
        pkt += bytes.fromhex(str(mac).replace(':','')) # chaddr

        pkt += b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
        pkt += b'\x00' * 67
        pkt += b'\x00' *125
        pkt += b'\x63\x82\x53\x63'


        pkt += b'\x35\x01\x05'
        pkt += b'\x01\x04\xff\xff\xff\x00'
        pkt += b'\x03\x04'+Ip_offer('192.168.0.1')
        pkt += b'\x33\x04\x00\x01\x51\x80'
        pkt += b'\x51\x04'+Ip_offer('192.168.0.1')
        pkt += b'\xff'
        return pkt
```