part 2 "Saskatoon": counting IPs.

Command to solve the problem

```
awk '{print $1}' access.log | sort | uniq -c | sort -r | head -n 1 |  awk '{print $2}' > /home/admin/highestip.txt
```

Explained in Video : https://youtu.be/1-u3wPz9bkQ

```
awk '{print $1}' access.log | sort | uniq -c | sort -nr | head -n 1
```

Other commands we can use:

```
cut -d ' ' -f1 access.log | sort | uniq -c | sort -nr | head -n 1
```

cut -d'<delimiter>' -f<field_numbers> <file>

```
grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' access.log | sort | uniq -c | sort -nr | head -n 1
```

'([0-9]{1,3}\.) Only print matched pattern [000 -999].[000 -999].[000 -999].[000-999]

What if we use this [0-9]{1,3}.[0-9]{1,3}

it would match:

- 192a168
- 10-0
- 172_16
- The . will be replace by a single literal so we use .\ which tells regex "Treat the dot as a **literal dot**, not a wildcard."

192\.168\.0\.1

192.168.0.1