Given the command line being:-

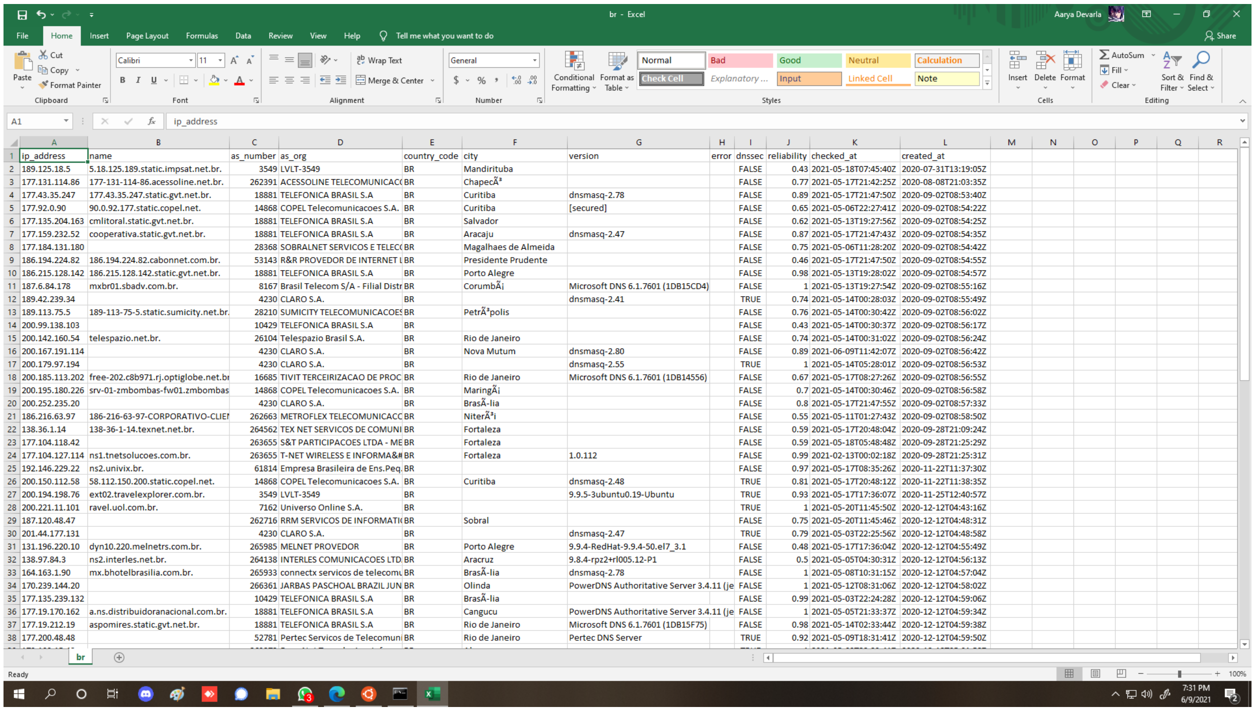
```
curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50 | xargs -i timeout 1 ping -c1 -w 1 {} | grep "time=" | awk '{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}' | sort -n | awk '{print $2 " " $1 "ms"}' | head -n 10
```

Let's go command by command:-

1. Curl :- It basically get the file from the internet from the specified url i.e., http://public-dns.info/nameserver/br.csv

   -s argument is used to do the operation in silent mode or silently without giving output

   It consists of the following data:-



ip_address	name	as_number	as_org	country_code	city	version	error	dnssec	reliability	checked_at	created_at
189.125.18.5	5.18.125.189.static.impsat.net.br.	3549	LVLT-3549	BR	Mandirituba			FALSE	0.43	2021-05-18T07:45:40Z	2020-07-31T13:19:05Z
177.131.114.86	177-131-114-86.acessoline.net.br.	262391	ACESSOLINE TELECOMUNICAC(	BR	ChapecÃ³			FALSE	0.77	2021-05-17T21:42:25Z	2020-08-08T21:03:35Z
177.43.35.247	177.43.35.247.static.gvt.net.br.	18881	TELEFONICA BRASIL S.A	BR	Curitiba	dnsmasq-2.78		FALSE	0.89	2021-05-17T21:47:50Z	2020-09-02T08:53:40Z
177.92.0.90	90.0.92.177.static.copel.net.	14868	COPEL Telecomunicacoes S.A.	BR	Curitiba	[secured]		FALSE	0.65	2021-05-06T22:27:41Z	2020-09-02T08:54:22Z
177.135.204.163	cmlitoral.static.gvt.net.br.	18881	TELEFONICA BRASIL S.A	BR	Salvador			FALSE	0.62	2021-05-13T19:27:56Z	2020-09-02T08:54:25Z
177.159.232.52	cooperativa.static.gvt.net.br.	18881	TELEFONICA BRASIL S.A	BR	Aracaju	dnsmasq-2.47		FALSE	0.87	2021-05-17T21:47:43Z	2020-09-02T08:54:35Z
177.184.131.180		28368	SOBRALNET SERVICOS E TELEC(	BR	Magalhaes de Almeida			FALSE	0.75	2021-05-06T11:28:20Z	2020-09-02T08:54:42Z
186.194.224.82	186.194.224.82.cabonnet.com.br.	53143	R&R PROVEDOR DE INTERNET L	BR	Presidente Prudente			FALSE	0.46	2021-05-17T21:47:50Z	2020-09-02T08:54:55Z
186.215.128.142	186.215.128.142.static.gvt.net.br.	18881	TELEFONICA BRASIL S.A	BR	Porto Alegre			FALSE	0.98	2021-05-13T19:28:02Z	2020-09-02T08:54:57Z
187.6.84.178	mxbr01.sbadv.com.br.	8167	Brasil Telecom S/A - Filial Distr	BR	CorumbÃ¡	Microsoft DNS 6.1.7601 (1DB15CD4)		FALSE	1	2021-05-13T19:27:54Z	2020-09-02T08:55:16Z
189.42.239.34		4230	CLARO S.A.	BR		dnsmasq-2.41		TRUE	0.74	2021-05-14T00:28:03Z	2020-09-02T08:55:49Z
189.113.75.5	189-113-75-5.static.sumicity.net.br.	28210	SUMICITY TELECOMUNICACOES	BR	PetrÃ³polis			FALSE	0.76	2021-05-14T00:30:42Z	2020-09-02T08:56:02Z
200.99.138.103		10429	TELEFONICA BRASIL S.A	BR				FALSE	0.43	2021-05-14T00:30:37Z	2020-09-02T08:56:17Z
200.142.160.54	telespazio.net.br.	26104	Telespazio Brasil S.A.	BR	Rio de Janeiro			FALSE	0.74	2021-05-14T00:31:02Z	2020-09-02T08:56:24Z
200.167.191.114		4230	CLARO S.A.	BR	Nova Mutum	dnsmasq-2.80		FALSE	0.89	2021-06-09T11:42:07Z	2020-09-02T08:56:42Z
200.179.97.194		4230	CLARO S.A.	BR		dnsmasq-2.55		TRUE	1	2021-05-14T05:28:01Z	2020-09-02T08:56:53Z
200.185.113.202	free-202.c8b971.rj.optiglobe.net.br	16685	TIVIT TERCEIRIZACAO DE PROC	BR	Rio de Janeiro	Microsoft DNS 6.1.7601 (1DB14556)		FALSE	0.67	2021-05-17T08:27:26Z	2020-09-02T08:56:55Z
200.195.180.226	srv-01-zmbombas-fw01.zmbombas	14868	COPEL Telecomunicacoes S.A.	BR	MaringÃ¡			FALSE	0.7	2021-05-14T00:30:46Z	2020-09-02T08:56:58Z
200.252.235.20		4230	CLARO S.A.	BR	BrasÃ-lia			FALSE	0.8	2021-05-17T21:47:55Z	2020-09-02T08:57:33Z
186.216.63.97	186-216-63-97-CORPORATIVO-CLIEI	262663	METROFLEX TELECOMUNICACC	BR	NiterÃ³i			FALSE	0.55	2021-05-11T01:27:43Z	2020-09-02T08:58:50Z
138.36.1.14	138-36-1-14.texnet.net.br.	264562	TEX NET SERVICOS DE COMUNI	BR	Fortaleza			FALSE	0.59	2021-05-17T20:48:04Z	2020-09-28T21:09:24Z
177.104.118.42		263655	S&T PARTICIPACOES LTDA - ME	BR	Fortaleza			FALSE	0.59	2021-05-18T05:48:48Z	2020-09-28T21:25:29Z
177.104.127.114	ns1.tnetsolucoes.com.br.	263655	T-NET WIRELESS E INFORMA&#	BR	Fortaleza	1.0.112		FALSE	0.99	2021-02-13T00:02:18Z	2020-09-28T21:25:31Z
192.146.229.22	ns2.univix.br.	61814	Empresa Brasileira de Ens.Peq.	BR				FALSE	0.97	2021-05-17T08:35:26Z	2020-11-22T11:37:30Z
200.150.112.58	58.112.150.200.static.copel.net.	14868	COPEL Telecomunicacoes S.A.	BR	Curitiba	dnsmasq-2.48		TRUE	0.81	2021-05-17T20:48:12Z	2020-11-22T11:38:35Z
200.194.198.76	ext02.travelexplorer.com.br.	3549	LVLT-3549	BR		9.9.5-3ubuntu0.19-Ubuntu		TRUE	0.93	2021-05-17T17:36:07Z	2020-11-25T12:40:57Z
200.221.11.101	ravel.uol.com.br.	7162	Universo Online S.A.	BR				TRUE	1	2021-05-20T11:45:50Z	2020-12-12T04:43:16Z
187.120.48.47		262716	RRM SERVICOS DE INFORMATI(	BR	Sobral			FALSE	0.75	2021-05-20T11:45:46Z	2020-12-12T04:48:31Z
201.44.177.131		4230	CLARO S.A.	BR		dnsmasq-2.47		TRUE	0.79	2021-05-03T22:25:56Z	2020-12-12T04:48:58Z
131.196.220.10	dyn10.220.melnetrs.com.br.	265985	MELNET PROVEDOR	BR	Porto Alegre	9.9.4-RedHat-9.9.4-50.el7_3.1		FALSE	0.48	2021-05-17T17:36:04Z	2020-12-12T04:55:49Z
138.97.84.3	ns2.interles.net.br.	264138	INTERLES COMUNICACOES LTD.	BR	Aracruz	9.8.4-rpz2+rl005.12-P1		FALSE	0.5	2021-05-05T04:30:31Z	2020-12-12T04:56:13Z
164.163.1.90	mx.bhotelbrasilia.com.br.	265933	connectx servicos de telecomu	BR	BrasÃ-lia	dnsmasq-2.78		FALSE	1	2021-05-08T10:31:15Z	2020-12-12T04:57:04Z
170.239.144.20		266361	JARBAS PASCHOAL BRAZIL JUN	BR	Olinda	PowerDNS Authoritative Server 3.4.11 (je		FALSE	1	2021-05-12T08:31:06Z	2020-12-12T04:58:02Z
177.135.239.132		10429	TELEFONICA BRASIL S.A	BR	BrasÃ-lia			FALSE	0.99	2021-05-03T22:24:28Z	2020-12-12T04:59:06Z
177.19.170.162	a.ns.distribuidoranacional.com.br.	18881	TELEFONICA BRASIL S.A	BR	Cangucu	PowerDNS Authoritative Server 3.4.11 (je		FALSE	1	2021-05-05T21:33:37Z	2020-12-12T04:59:34Z
177.19.212.19	aspomires.static.gvt.net.br.	18881	TELEFONICA BRASIL S.A	BR	Rio de Janeiro	Microsoft DNS 6.1.7601 (1DB15F75)		FALSE	0.98	2021-05-14T02:33:44Z	2020-12-12T04:59:38Z
177.200.48.48		52781	Pertec Servicos de Telecomuni	BR	Rio de Janeiro	Pertec DNS Server		TRUE	0.92	2021-05-09T18:31:41Z	2020-12-12T04:59:50Z

2. Cut :- After it gets all the data from the specified file it then removes unnecessary data which we won't be using i.e, removing all the data except for the first column

   Here,

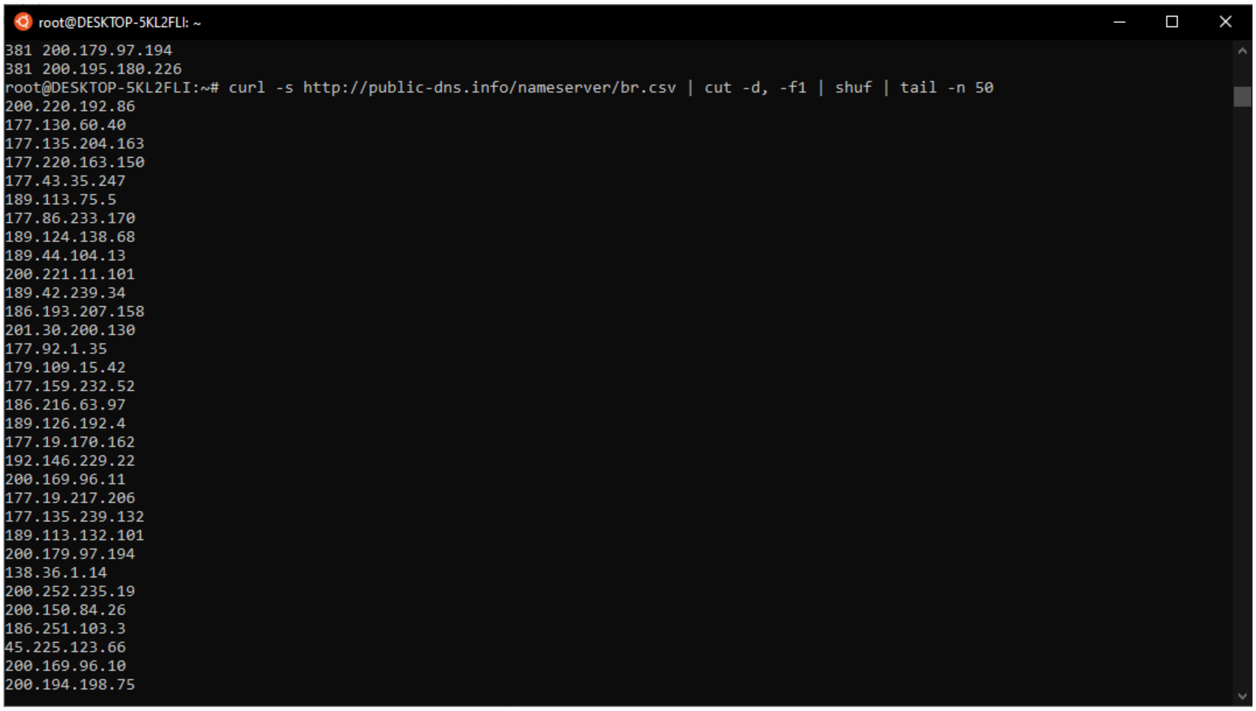
   -d, is used as a delimiter (as this is a csv file we use *comma*)

   -f1 is used to select the fields/data in the which column, here we are selecting the column 1

3. Shuf :- It is to shuffle the data

4. Tail :- It is used to get the data from the end of the file

   We are using *-n* so that it selects last n lines instead of last n characters



```
381 200.179.97.194
381 200.195.180.226
root@DESKTOP-5KL2FLI:~# curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50
200.220.192.86
177.130.60.40
177.135.204.163
177.220.163.150
177.43.35.247
189.113.75.5
177.86.233.170
189.124.138.68
189.44.104.13
200.221.11.101
189.42.239.34
186.193.207.158
201.30.200.130
177.92.1.35
179.109.15.42
177.159.232.52
186.216.63.97
189.126.192.4
177.19.170.162
192.146.229.22
200.169.96.11
177.19.217.206
177.135.239.132
189.113.132.101
200.179.97.194
138.36.1.14
200.252.235.19
200.150.84.26
186.251.103.3
45.225.123.66
200.169.96.10
200.194.198.75
```

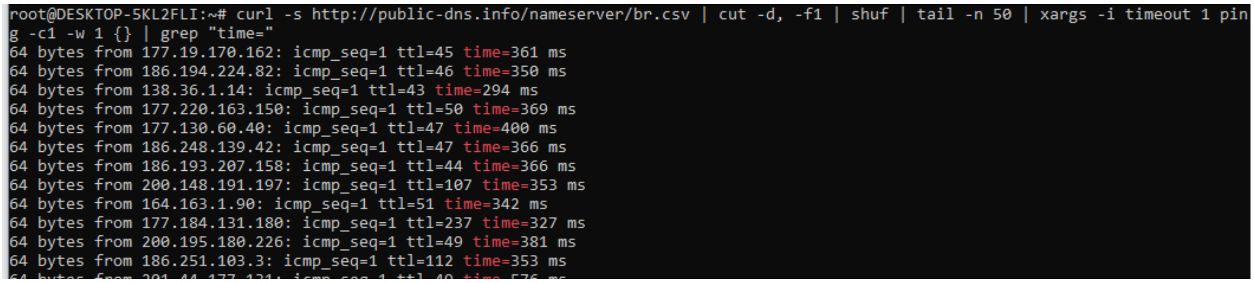
   Here we see that we’ve the last 50 ip addresses after shuffling them so these ip addresses are all randomly shuffled.

5. *xargs -i timeout 1 ping -c1 -w 1 {} :-*
   This was a bit tricky to understand.
   So it has multiple commands in a same line unlike the other where they are all separated by |.

   Let’s go modular:-
   a. Timeout: It basically executes any piece of code for a fixed amount of time.
   b. Ping: It is used to ping the address to see if the computer can communicate with that system or server.
   c. Xargs: This is basically used to send whatever data it received and send it as an arguments to the commands.

   Here, xargs is feeding the *ping* command with the list of ip addresses as an argument at *{}*.

6. Grep: It is basically used to filter or catch the lines which matches the given substring. Here we give the substring as *"time="*. So we get,



```
root@DESKTOP-5KL2FLI:~# curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50 | xargs -i timeout 1 pin
g -c1 -w 1 {} | grep "time="
64 bytes from 177.19.170.162: icmp_seq=1 ttl=45 time=361 ms
64 bytes from 186.194.224.82: icmp_seq=1 ttl=46 time=350 ms
64 bytes from 138.36.1.14: icmp_seq=1 ttl=43 time=294 ms
64 bytes from 177.220.163.150: icmp_seq=1 ttl=50 time=369 ms
64 bytes from 177.130.60.40: icmp_seq=1 ttl=47 time=400 ms
64 bytes from 186.248.139.42: icmp_seq=1 ttl=47 time=366 ms
64 bytes from 186.193.207.158: icmp_seq=1 ttl=44 time=366 ms
64 bytes from 200.148.191.197: icmp_seq=1 ttl=107 time=353 ms
64 bytes from 164.163.1.90: icmp_seq=1 ttl=51 time=342 ms
64 bytes from 177.184.131.180: icmp_seq=1 ttl=237 time=327 ms
64 bytes from 200.195.180.226: icmp_seq=1 ttl=49 time=381 ms
64 bytes from 186.251.103.3: icmp_seq=1 ttl=112 time=353 ms
64 bytes from 201.44.177.131: icmp_seq=1 ttl=49 time=576 ms
```

7. Awk:

   This command is used to scan the patterns and process it accordingly.

   Here we are trying to rewrite the output which we are getting to only ip addresses and the time for the ping.

   So we are using substrings to get the words, we notice that we use $7 and $4. It basically denotes which column or the word of that line is to be considered. $7 refers to the 7th word and $4 refers to the 4th word.

   Here we are first taking the time and then the ip addresses as we will be sorting based on the ping time.

8. Sort:

   This command sorts the given input.

   -n denotes numeric sort

9. Awk:

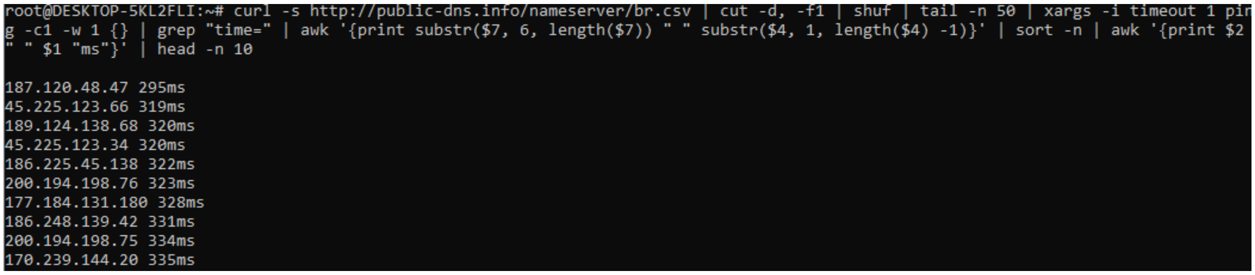
   We are using the same command which we used earlier to rewrite the output as ip address and ping instead of other way around

10. Head:

    We use this command to print the first 10 characters/lines of any file or input given.

    -n denotes number of lines
    10 is the amount of lines or characters

The final output would be:-



```
root@DESKTOP-5KL2FLI:~# curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50 | xargs -i timeout 1 pin
g -c1 -w 1 {} | grep "time=" | awk '{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}' | sort -n | awk '{print $2
" " $1 "ms"}' | head -n 10

187.120.48.47 295ms
45.225.123.66 319ms
189.124.138.68 320ms
45.225.123.34 320ms
186.225.45.138 322ms
200.194.198.76 323ms
177.184.131.180 328ms
186.248.139.42 331ms
200.194.198.75 334ms
170.239.144.20 335ms
```

In simple words, this whole command summarises to finding ip addresses which have the least ping in the given file.