Linux administration with bash. Home task

A. Create a script that uses the following keys:

1. When starting without parameters, it will display a list of possible keys and their description.

```
student@chshai:~/bashtasks$ ./scriptA
-a    Display a list of possible keys and their description
-t    Display a list of open system TCP ports
```

2. The --all key displays the IP addresses and symbolic names of all hosts in the current subnet

```
student@chshai:~/bashtasks$ ./scriptA -a
Starting Nmap 7.80 ( https://nmap.org ) at 2023-08-28 17:21 UTC
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0018s latency).
Nmap scan report for 192.168.1.102
Host is up (0.0031s latency).
Nmap scan report for 192.168.1.103
Host is up (0.0041s latency).
Nmap scan report for 192.168.1.104
Host is up (0.043s latency).
Nmap scan report for chshai (192.168.1.105)
Host is up (0.000058s latency).
Nmap scan report for 192.168.1.106
Host is up (0.000080s latency).
Nmap done: 256 IP addresses (6 hosts up) scanned in 3.23 seconds
```

3. The --target key displays a list of open system TCP ports.

```
student@chshai:~/bashtasks$ ./scriptA -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 chshai:ssh             192.168.1.106:60761     ESTABLISHED
tcp        0      0 chshai:ssh             192.168.1.106:60762     ESTABLISHED
```

The code that performs the functionality of each of the subtasks must be placed in a separate function

B. Using Apache log example create a script to answer the following questions:

1. From which ip were the most requests?

```
#!/bin/bash
file_out=out_script1
grep -E -o "([0-9]{1,3}[\.]){3}[0-9]{1,3}" $1 | sort |uniq -c | sort -gr > $fil>
{
read line1
}<$file_out
echo $line1
```

```
student@chshai:~/bashtasks$ ./scriptB1 apche_log
62 157.55.39.250
```

2. What is the most requested page?

```
#!/bin/bash
file_out=out_script2
awk '{print $7}' $1 | sort |uniq -c | sort -gr > $file_out
{
read line1
}<$file_out
echo $line1
```

```
student@chshai:~/bashtasks$ ./scriptB1.save apche_log
8 /sitemap1.xml.gz
```

3. How many requests were there from each ip?

```
#!/bin/bash
file_out=out_script1
grep -E -o "([0-9]{1,3}[\.]){3}[0-9]{1,3}" $1 | sort |uniq -c | sort -gr > $fil

while read -r line1;
 do
  echo "$line1"
done < "$file_out"
```

```
student@chshai:~/bashtasks$ ./scriptB3 apche_log
62 157.55.39.250
61 46.29.2.62
34 207.46.13.48
10 178.76.227.154
7 176.59.119.104
4 157.55.39.174
3 37.140.141.30
2 66.249.78.58
2 217.69.134.29
2 157.55.39.182
1 95.108.158.190
1 93.158.178.129
1 66.249.78.72
1 66.249.78.65
1 66.249.69.39
1 5.255.253.74
1 5.255.253.45
1 217.69.134.39
1 217.69.134.15
1 217.69.134.13
1 217.69.134.12
1 217.69.134.11
1 213.87.151.38
```

4. What non-existent pages were clients referred to?

```
#!/bin/bash
file_out=out_script1
awk '$9 == 302 {print $7}' $1 | sort | uniq  > $file_out
{
 read line1
}< "$file_out"
echo $line1
```

```
student@chshai:~/bashtasks$ ./scriptB4 apche_log
/vote/1279
```

5. What time did site get the most requests?

6. What search bots have accessed the site? (UA + IP)

C. Create a data backup script that takes the following data as parameters:

1. Path to the syncing directory.

2. The path to the directory where the copies of the files will be stored.

In case of adding new or deleting old files, the script must add a corresponding entry to the log file indicating the time, type of operation and file name. [The command to run the script must be added to crontab with a run frequency of one minute]

```
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Usage: $0 <syncing_directory> <backup_directory>"
    exit 1
fi

sync_dir="$1"
backup_dir="$2"
log_file="backup.log"

mkdir -p "$backup_dir"

getTime(){
  date +'%Y-%m-%d %H:%M:%S'
}

changes=$(diff -rq "$sync_dir" "$backup_dir")

echo "$changes"
for line in $changes; do
    change_type=$(echo "$line" | awk '{print $1}')
    file_path=$(echo "$line" | awk '{print $4}')

    if [ "$change_type" = "Only" ]; then
        echo "$(getTime) - ADDED: $file_path" >> "$log_file"
    elif [ "$change_type" = "Files" ]; then
        echo "$(getTime) - DELETED: $file_path" >> "$log_file"
    fi
done

rsync -av --delete "$sync_dir/" "$backup_dir/"
```

```
student@chshai:~/bashtasks/first_dir$ ls -l
total 0
-rw-rw-r-- 1 student student 0 Aug 31 08:49 file1
-rw-rw-r-- 1 student student 0 Aug 31 08:49 file2
-rw-rw-r-- 1 student student 0 Aug 31 08:49 file3
-rw-rw-r-- 1 student student 0 Aug 31 08:49 file4
```

```
student@chshai:~/bashtasks$ ./scriptC
Usage: ./scriptC <syncing_directory> <backup_directory>
```

```
student@chshai:~/bashtasks$ ./scriptC first_dir second_dir
Only in first_dir: file1
Only in first_dir: file2
Only in first_dir: file3
Only in first_dir: file4
sending incremental file list
./
file1
file2
file3
file4

sent 289 bytes  received 95 bytes  768.00 bytes/sec
total size is 0  speedup is 0.00
student@chshai:~/bashtasks$ cat backup.log
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:16:14 - ADDED:
```

```
student@chshai:~/bashtasks$ rm first_dir/file2
student@chshai:~/bashtasks$ ./scriptC first_dir second_dir
Only in second_dir: file2
sending incremental file list
deleting file2
./

sent 121 bytes  received 28 bytes  298.00 bytes/sec
total size is 0  speedup is 0.00
```

```
student@chshai:~/bashtasks$ cat backup.log
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:16:14 - ADDED:
2023-08-31 09:17:15 - DELETED:
```