logs

- The `/var` directory is where Linux systems store variable data—files that change frequently during normal system operation.
- `/var` holds runtime data that grows and shrinks as the system runs.
- Common subdirectories include:
  - `/var/log` for log files
  - `/var/cache` for application caches
  - `/var/tmp` for temporary files
  - `/var/spool` for print and mail queues.
- The `/var/log/syslog` file is the system's general-purpose log file on Debian-based systems (like Ubuntu). Red Hat-based systems (like CentOS or Fedora) use `/var/log/messages` instead of syslog.
  It captures messages from the kernel and various system processes—things like boot messages, service startup/shutdown events, authentication attempts, hardware detection, and general system activity. When something goes wrong on a Linux system, `syslog` is often one of the first places to look for clues about what happened.
- Each line in syslog typically includes a timestamp, the hostname, the process name, and the message itself.
- You can view it with commands like:
  - `cat /var/log/syslog`
  - `less /var/log/syslog`
  - `tail -f /var/log/syslog` (to watch new entries as they're written).
- Since it can grow quite large, systems often use log rotation to archive old entries and keep the file manageable.
- Many applications use severity levels like DEBUG, INFO, WARNING, ERROR, and CRITICAL (or FATAL). The syslog protocol itself defines eight standard severity levels (ranging from Emergency down to Debug), so you'll often see those labels in logs. But the exact format—whether it says "ERROR" or "ERR" or "E", where it appears in the line, what punctuation surrounds it—varies widely depending on who wrote the application.

What we'll be writing

- Write a script that analyzes syslog.
- We'll be looking for lines containing IP addresses
- In these lines we'll look for error/warning keywords (make your own list of keywords)
- It then writes all this analysis into a timestamped report file and displays it.
    - We'll Count all lines referring to the same IP address and count them
    - collect all the keywords that appeared for those lines
    - Write your repoert

Here's a report example:

```
file name: report-26-noc-2025.rep
*******************************************************************************
Report created at 11:116
192.168.15.2 address appeared in 38 lines.
  keywords appeared:  ERROR, WARNING
10.5.3.8 address appeared in 4 lines.
  keyword appeared: FATAL
*******************************************************************************
```

some guidelines

- You can use regular expressions to find an IP address inside a line.
  We have not covered it but you can easily find examples a guidance on the Internet.
- If you don't want to use it - you can iterate over the characters of the line and look for the IP pattern yourself.
  you can iterate over the characters of a string using the **parameter substring expansion**.
  Here's an example:
  yuval@comp> myvar=abcdefghijk

```
yuval@comp> echo ${myvar:3:6}
defghi
yuval@comp>
```

- USE FUNCTIONS!!!!!
  for example
  - a function that looks for an ip address in a line
  - a function that looks for a keyword (from you list) in this line
- When you work - each time something works, even if it is very far from what you want to achieve…
  Commit to you repository.
- Send me a link for the repository, or make me a contributor if it is private one.