1. How my script handles arguments and options:

First, the script checks if user wrote --help then it shows usage and exit.

After that, it uses getopts to read options like -n or -v.

If the wrong option is given, it prints error and exit.

After reading options, it moves to the next arguments to get search_word and file_name.

It checks if the word is not a file and checks if the file is .txt file.

Then it reads the file line by line and depending on options, it prints matching or not matching lines, with or without line number.

2. If I add regex or -i, -c, -l options:

If I want to support regex, I will change the [... =~ ...]] part maybe use grep inside script.

For -i (ignore case) it will be easy because now I already use ${line,,} and ${search_word,,} for lowercase.

For -c (count matches), I will add a counter and show how many matches are found.

For -l (only show file name if match), I will stop reading after first match and just print file name.

Maybe I will make more variables like count_only=true and list_only=true and handle them.

3. Hardest part in script and why:

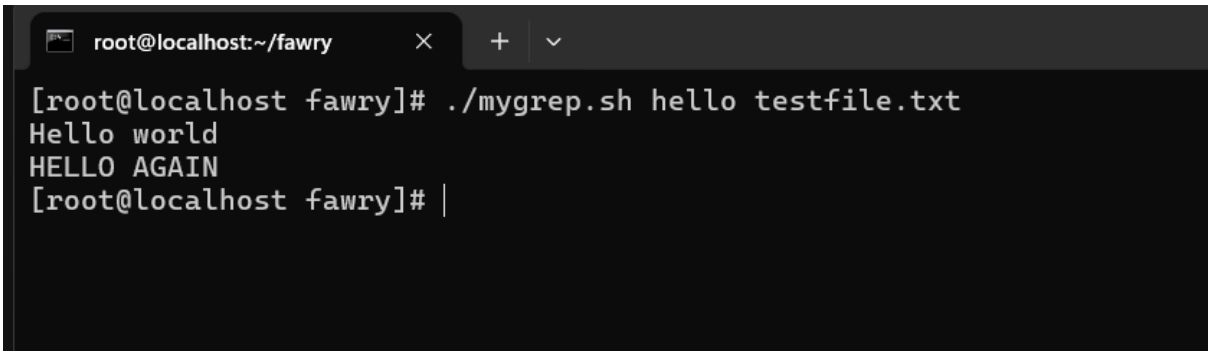
The hardest part was making options work good together.

Like when user write -vn or -nv both must work same.

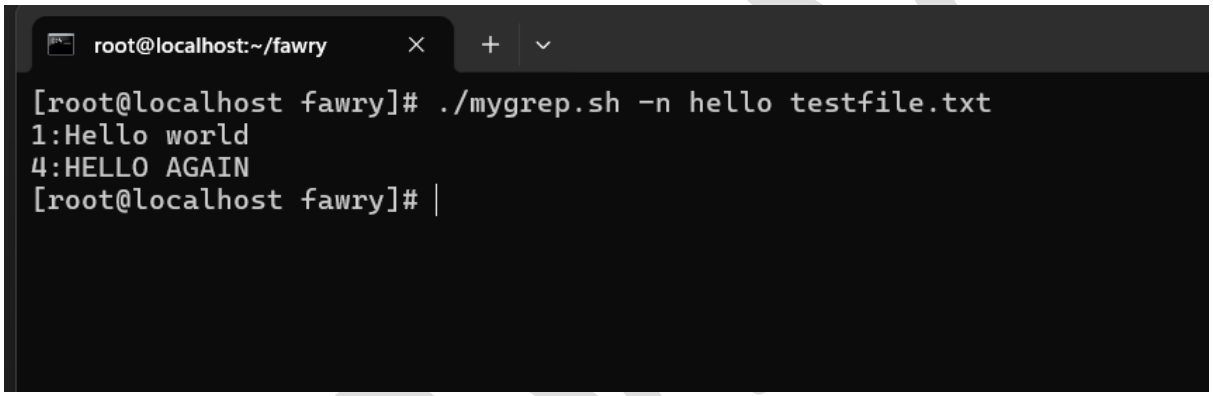
Also, it was a little hard to use getopts first time and understand shift and OPTIND.

But after testing and searching many times, it worked well.

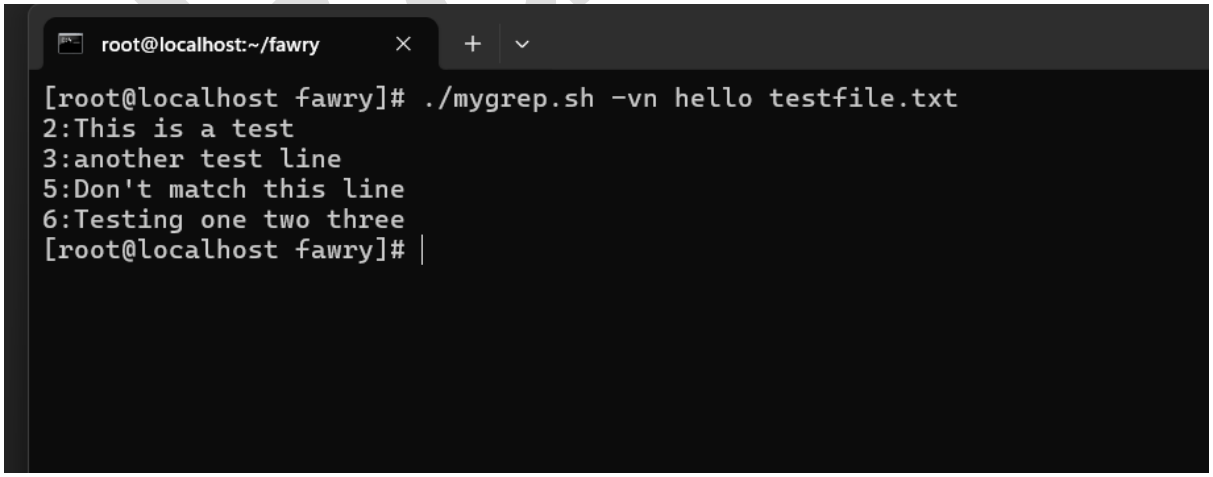
./mygrep.sh hello testfile.txt



./mygrep.sh -n hello testfile.txt



./mygrep.sh -vn hello testfile.txt



./mygrep.sh -v testfile.txt