Q1- Steps:

1. Making the mygrep.sh  
   
2. Making the file to be able to execute:  
   A screenshot of a computer

   AI-generated content may be incorrect.
3. Writing the testfile.txt:  
   A screenshot of a computer

   AI-generated content may be incorrect.
4. Supporting Additional Grep Features Option Handling (-n, -v, -E, -c, -l):

The while getopts "nvEcl" opt loop (line 20) parses flags and sets corresponding flags (e.g., line\_numbers=1 for -n).

Flags are dynamically combined into grep\_opts (lines 50–64), e.g., grep\_opts="${grep\_opts}n" adds -n for line numbers.

Error Handling:

Checks for missing arguments (if [ $# -lt 2 ], line 38) and invalid files (if [ ! -f "$file" ], line 43).

The -v testfile.txt case fails as expected because shift $((OPTIND-1)) (line 36) ensures the search string is required.

Regex and Extended Features:

The -E flag (line 24) enables extended regex via grep -E.

Case-insensitive search (-i) is hardcoded (line 47), while -c and -l are optional (lines 58–64).

1. Test with:

./mygrep.sh hello testfile.txt # Basic search

./mygrep.sh -n hello testfile.txt # Line numbers

./mygrep.sh -vn hello testfile.txt # Inverted match

./mygrep.sh -v testfile.txt # Error: Missing search string

A screen shot of a computer

AI-generated content may be incorrect.

Another view for the -c,-E,-l, and some combination and –help &-h:  
A screen shot of a computer program

AI-generated content may be incorrect.  
A screenshot of a computer program

AI-generated content may be incorrect.  
  
thank you for your time, have a nice day!