# grep Assignment

**Mandatory:**

1. Use any .c file. Using grep command extract and display
   1. all lines beginning with #include with line numbers

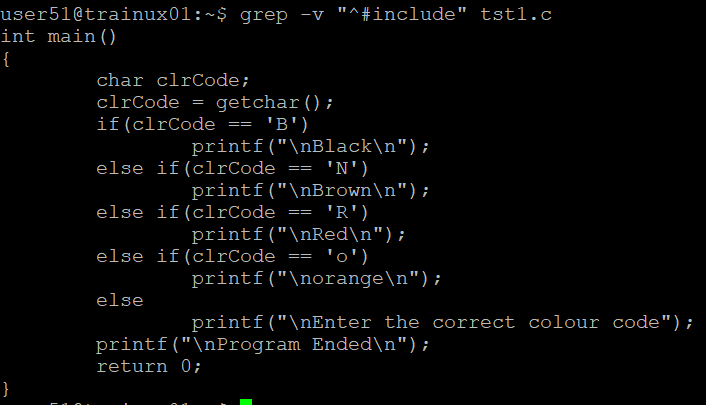
Ans: grep -n “^#include” tst1.c

A black screen with white text

Description automatically generated

* 1. display all lines which do not begin with #include

Ans: grep -v “^#include” tst1.c



* 1. display the line number of main()

Ans: grep -n “^int main(“filename



* 1. extract all lines containing characters of opening and closing parathesis {(,))

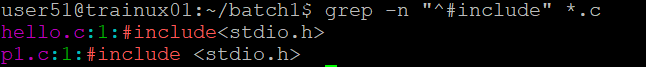
Ans: grep -n “[{} ()]” filename

A screen shot of a computer

Description automatically generated

1. Perform the above operations on a set of \*.c files

Ans: grep -n “^#include” \*.c



grep -n “^int main(“ \*.c

A black screen with white text

Description automatically generated

**Optional Assignments:**

1. Use a file with email id’s . Use regular expression to extract and display the user and domain names.
2. Obtain the file [grepdata.txt](http://evc-cit.info/cit052/grepdata.txt). You can save the file on your local system. Once you have the file, write a series of grep statements that do the following:

* Print all lines that contain a phone number with an extension (the letter x or X followed by four digits).
* Print all lines that begin with three digits followed by a blank. Your answer *must* use the \{ and \} repetition specifier.
* Print all lines that contain a date. Hint: this is a *very* simple pattern. It does not have to work for any year before 2000.
* Print all lines containing a vowel (a, e, i, o, or u) followed by a single character followed by the same vowel again. Thus, it will find “eve” or “adam” but not “vera”. Hint: \( and \)
* Print all lines that do not begin with a capital S.

[Note: Your patterns should work in any generic file of this sort. They should not be dependent upon the data in this particular file; if more lines of the same form to the file are added then, your patterns should still work.]