

Write a program in C++ which behaves quite like grep -r.

The program has one obligatory parameter which is <pattern (a string)> and can have four optional parameters (if user does not specify then default values are used).

Additional parameters:

Parameter	Description	default value
-d or --dir	start directory where program needs to look for files (also in subfolders)	current directory
-l or --log_file	name of the log file	<program name>.log
-r or --result_file	name of the file where result is given	<program name>.txt
-t or --threads	number of threads in the pool	4

The program executed without obligatory parameter should return information on console how to use it and stopped. The program should use **threads pool** to process files.

When the program finished his work, it should give following **output on console**:

Searched files: <nr of files found in directory and subdirectories>

Files with pattern: <nr of files where the patten was found>

Patterns number: <Number of patterns found in all files>

Result file: <path to result file>

Log file: <path to log file>

Used threads: <nr of threads>

Elapsed time: <time elapsed from beginning to the end of the program>

Program produces two files; **the result file** should look as below:

<file path>:<line number>: line content

<file path>:<line number>: line content

etc

it should contain list of all files (with line number and line content) where the pattern was found, sorted from the file where most patterns were found, and ending were least.

The log file should contain list of thread ids and file names which its processed, sorted from the thread id with the most files number and ending with the least, (here are only file names, not full paths to files).

<thread id>:<file name>,<file name>, etc

<thread id>:<file name>,<file name>, etc

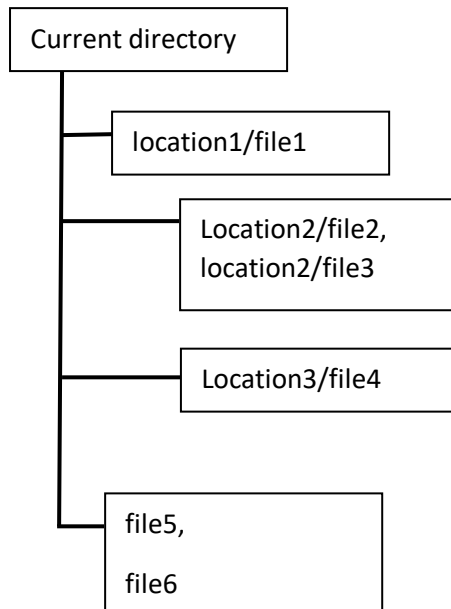
etc

Example:

Program, let it be called **specific_grep** was executed as below:

`./specific_grep "house"`

The program should start searching from current directory, and let it be structure as this:



Let it be files 4 and 6 contain the word "house", and the file2 contains it three times.

The result file would look (absolute paths to files):

`.../location2/file2:5: the house was big`

`.../location2/file2:8: something was done in his house`

`.../location2/file2:20: another house was far from`

`.../file6:30: in his house`

`.../file4:15: nearby his house`

And the log file would look like below

`140035841935104: file6, file 4`

`140035825149696: file2`

`140035833542400:`

`140035855147214:`

Console output :

Searched files: 6

Files with pattern: 3

Patterns number: 5

Result file: specific_grep.txt

Log file: specific_grep.log

Used threads: 4

Elapsed time: 267[ms]

Important

To deliver the implementation pls use <https://github.com/> (with public access)

Its good if you have there some more commits, pls push every time you finish a part of implementation.

Pls also add **files to automatically build the program** like Makefile or similar.