Homework 5: String Matching and Empty Line Count in a Text File

Specification:

You need to write a program that will read a file name and a string from the command line. Next, the program will find the number of occurrences of the string in that file and count the number of empty lines too. Please find attached the C code "**file.c**". You need to implement the following two functions:

int countString(const char* fileName, char *str); // counts the number of occurrences of "str" in file "fileName" 6 points

int countEmptyLines(const char* fileName); // counts the number of empty lines in file "filename" 6 points

Let's consider the content of the file named "**poem.txt**"

Do not go gentle into that good night,
Old age should burn and rave at close of day;
Rage, rage against the dying of the light.

Though wise men at their end know dark is right,
Because their words had forked no lightning they
Do not go gentle into that good night.

Good men, the last wave by, crying how bright
Their frail deeds might have danced in a green bay,
Rage, rage against the dying of the light.

Wild men who caught and sang the sun in flight,
And learn, too late, they grieved it on its way,
Do not go gentle into that good night.

Grave men, near death, who see with blinding sight
Blind eyes could blaze like meteors and be gay,
Rage, rage against the dying of the light.

And you, my father, there on the sad height,
Curse, bless, me now with your fierce tears, I pray.
Do not go gentle into that good night.
Rage, rage against the dying of the light.

After you complete both of the above-mentioned functions, you can find the occurrences of the string "**light**" in "**poem.txt**" as follows:

```
syasmin@cscd-linux01:~/CSCD240/HW5$ ./fileTest poem.txt light
The string "light" occurs 6 times in File "poem.txt".
File "poem.txt" has 5 empty lines.
syasmin@cscd-linux01:~/CSCD240/HW5$
```

The program also finds the number of occurrences of empty lines in "**poem.txt**". For empty line, you can check whether the line contains only new line ('\n'), tab ('\t') or space (' ').

Search should not be case sensitive. For example, search for string "Rage" or "rage" should give the same result as follows:

```
syasmin@cscd-linux01:~/CSCD240/HW5$ ./fileTest poem.txt  Rage
The string "Rage" occurs 8 times in File "poem.txt".
File "poem.txt" has 5 empty lines.
syasmin@cscd-linux01:~/CSCD240/HW5$
```

```
syasmin@cscd-linux01:~/CSCD240/HW5$ ./fileTest poem.txt  rage
The string "rage" occurs 8 times in File "poem.txt".
File "poem.txt" has 5 empty lines.
syasmin@cscd-linux01:~/CSCD240/HW5$
```

For string match, you may find the following standard library (string.h) function useful:

```
char * strstr ( const char *st, const char *sub );
```

'**strstr**' finds the first occurrence of the substring '**sub**' in the string '**st**'. An example about how to use '**strstr**' can be found to the following link: https://www.tutorialspoint.com/c_standard_library/c_function_strstr.htm

Makefile: You need to use **Makefile**; split the attached file **file.c** into three files: (a) header file named **fileTest.h** and (b) two separate source files: **fileTest.c** and (c) **file.c**. **fileTest.h** will only contain function declaration. For convenience, you can use additional functions. **fileTest.c** will contain the function body. **file.c** will include the main function only. 8 pts

Submission:

Your final submission will be a **zip** file that will include the following files:

fileTest.h

fileTest.c

file.c and

Makefile

Name your zip file with your last name first letter of your first name HW5.zip (ex: yasminsHW5.zip).

Submission deadline is: 11:59 pm, Tuesday, March 16. No late submission will be considered.