Project 2
CS221: C and Systems Programming – Fall 2018

Deadline: November 12, 2018 at 11:59pm

Restricted grep (rgrep)

`grep` is a UNIX utility that is used to search for patterns in text files. It's a powerful and versatile tool, and in this project you will implement a version that, while simplified, should still be useful.[1]

Your project is to complete the implementation of `rgrep`, our simplified, restricted `grep`. `rgrep` is "restricted" in the sense that the patterns it matches only support a few regular operators (the easier ones). The way `rgrep` is used is that a pattern is specified on the command line. `rgrep` then reads lines from its standard input and prints them out on its standard output if and only if the pattern "matches" the line. For example, we can use `rgrep` to search for lines that contain text file names that are at least 3 characters long (plus the extension) in a file like the following:

`#` so you can see what lines are in the file:

```
$ cat testin

a.out
cs221.txt
cs221.pdf
usf.txt
nope.pdf
.txt

$ ./rgrep '.\.txt' < testin

cs221.txt
usf.txt
```

What's going on here? `rgrep` was given the pattern `".\.txt"`; it printed only the lines from its standard input that matched this pattern. How can you tell if a line matches the pattern? A line matches a pattern if the pattern "appears" somewhere inside the line. In the absence of any special operators, seeing if a line matches a pattern reduces to seeing if the pattern occurs as a substring anywhere in the line. So for most characters, their meaning in a pattern is just to match themselves in the target string. However, there are a few special clauses you must implement:

.(period)	Matches any character
+(plus sign)	The preceding character may appear 1 or more times (in other words, the preceding character can be repeated several times in a row).
?(question mark)	The preceding character may appear between 0 and 1 times (in other words, the preceding character is optional).
\(backslash)	"Escapes" the following character, nullifying any special meaning it has.

So, here are some examples of patterns and the kinds of lines they match.

(	An open parenthesis must appear somewhere in the line.
hey+	Matches a line that contains the string "hey" followed by any number (0 or more) of y's.
str?ing	Matches lines that contain the substrings "string" or "sting", since the "r" is optional..
z.z\.txt	Matches lines that contain the substring "zaz.txt", "zbz.txt", etc., where the character between the z's can be anything, including a period.

[1] Type `man grep` in terminal for more detailed information on how grep works.