This lab investigates searching a character string for patterns and as an introduction to regular expressions.

1. Make a copy of the executable `regexp6hw` in your own directory and run it (the source `regexp6hw.cpp` is private since it is a solution to this assignment).

2. Make up a line of text and a search pattern to look for and try the program. Note that you can put just characters in the search string and in this case the program behaves just like the case sensitive `strstr()`. However, you can also use the special symbols
   ^ beginning of the line
   . wildcard, matches any character
   $ end of the line
   the \ is used to indicate a literal, for example
   \. a real period
   \' a real single quote
   \" a real double quote
   \\ a real backslash
   As an example you might choose
   line="He said that it’s not at all clear."
   and try to search for one of the following
   search="sa.d"
   search="it\'s not"
   search="clear\.\$"
   search="`He said"
   (you don’t type the quotes, this is what the program displays). You can also invoke the program in debug mode (`regexp6hw -d`) which will give you more information about how it did the search.

**Assignment** Write a program which will accomplish the same thing as `regexp6hw`. Before you start writing any code you should think about what algorithm you want to devise to solve this problem and think about what code we wrote in class to implement `strstr()`. Regardless, your solution should have the algorithm incorporated into a function

```c
char *regexp(char str[], char subst[]);
```

which will either return the NULL pointer for failure or a pointer to the position of the match (same behavior as `strstr()`).

Email me the pathname of your program in plain text, not as an attachment. For example, you might say

My lab8 program is /usr/stu/demo/cs221/lab8.cpp

**Extra Credit** For extra credit implement the search subrange operation which uses square brackets to indicate that any one of the characters is a match, e.g. `[aeiou]` matches any vowel and `[Ss]` matches either upper or lowercase ’s’.