

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.

OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!



BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!

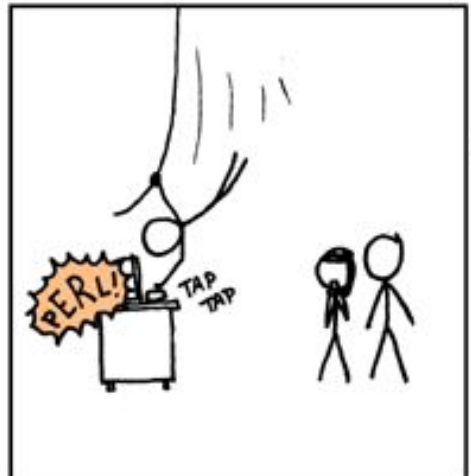


IT'S HOPELESS!

EVERYBODY STAND BACK.



I KNOW REGULAR EXPRESSIONS.



POSIX Regular Expressions

Expression	Matches	Example
<code>c</code>	the one non-operator character <code>c</code>	<code>a</code>
<code>\c</code>	character <code>c</code> literally *	<code>*</code>
<code>"s"</code>	string <code>s</code> literally	<code>"hello"</code>
<code>.</code>	any character except newline	<code>a.b</code>
<code>^</code>	beginning of a line	<code>^abc</code>
<code>\$</code>	end of a line	<code>abc\$</code>
<code>[s]</code>	any one of the characters in string <code>s</code>	<code>[abc]</code>
<code>[^s]</code>	any one character not in string <code>s</code>	<code>[^abc]</code>
<code>r*</code>	≥ 0 occurrences of <code>r</code>	<code>a*</code>
<code>r+</code>	≥ 1 occurrences of <code>r</code>	<code>a+</code>
<code>r?</code>	0 or 1 occurrences of <code>r</code>	<code>a?</code>
<code>r1 r2</code>	<code>r1</code> followed by <code>r2</code>	<code>ab</code>
<code>r1 r2</code>	<code>r1</code> or <code>r2</code>	<code>a b</code>
<code>(r)</code>	same as <code>r</code>	<code>(a b)</code>
<code>r1 / r2</code>	<code>r1</code> when followed by <code>r2</code>	<code>abc / 123</code>

Notes on Regular Expressions

Special matching

- Dash within [] for ranges ([A-Z] [a-z] [A-Za-z] [0-9])
- If blank (space) is inside brackets, will match as a character
- Special characters: \t, \n, \\, \"
- \s matches any whitespace character: [\t\n\r\f]

More info and examples:

<http://marvin.cs.uidaho.edu/Handouts/regex.html>

Flex Example 0:

```
/* Word Counter */
%{
  int chars = 0;
  int words = 0;
  int lines = 0;
}%

%%
[a-zA-Z]+ { words++; chars += strlen(yytext); }
\n       { chars++; lines++; }
.        { chars++; }
%%

main()
{
  yylex();
  printf("%8d%8d%8d\n", lines, words, chars);
}
```

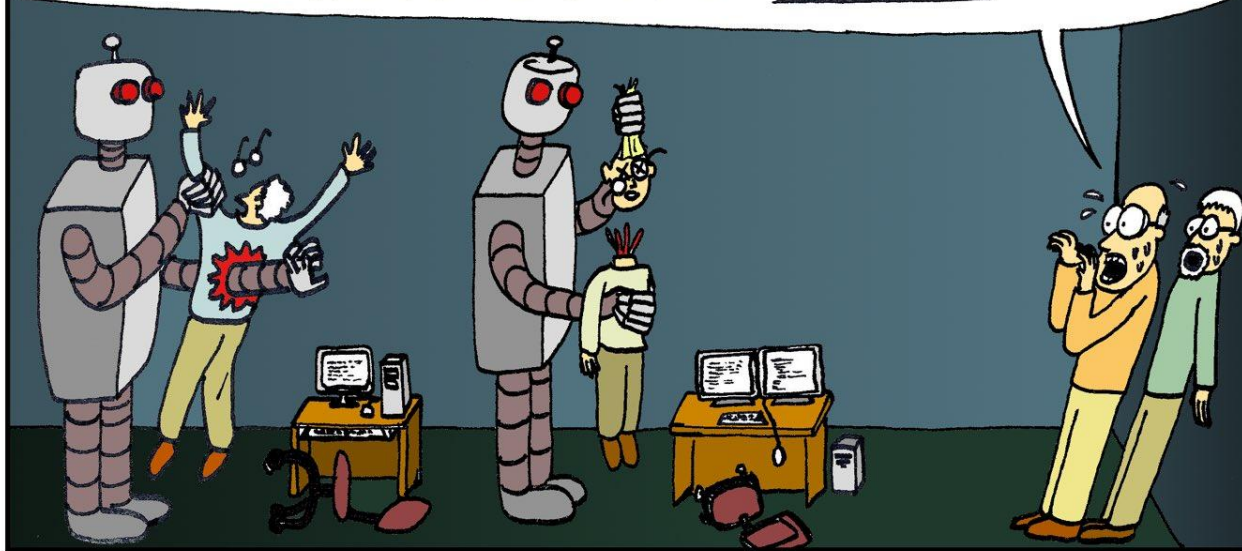
Flex Example 2:

```
%{
#include <iostream>
using namespace std;
#define YY_DECL extern "C" int yylex()
%}

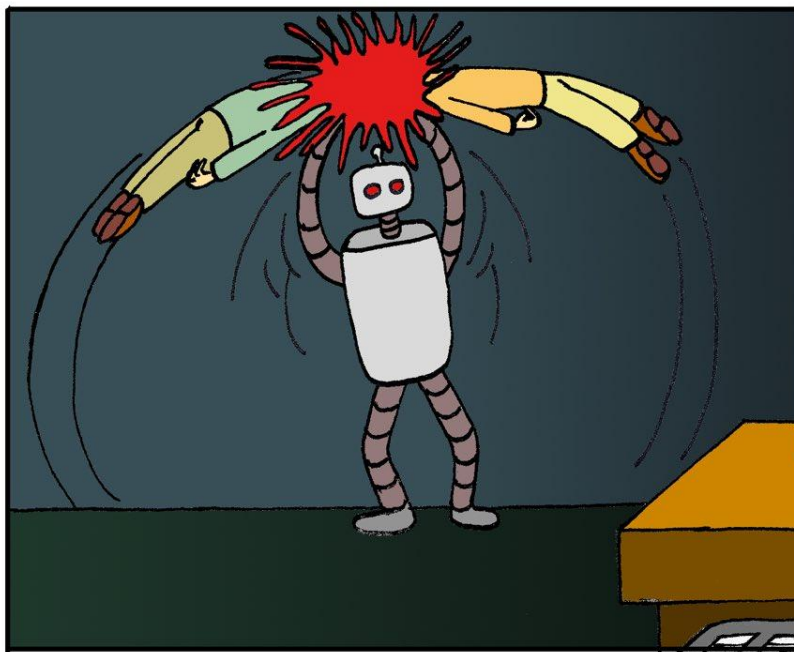
%%
[ \t\n]      ;
[0-9]+\.[0-9]+ { cout << "Found a floating-point number:" << yytext <<
endl; }
[0-9]+       { cout << "Found an integer:" << yytext << endl; }
[a-zA-Z0-9]+ { cout << "Found a string: " << yytext << endl; }
%%

main() {
    // lex through the input:
    yylex();
}
```

OH NO! THE ROBOTS ARE KILLING US!!!



BUT WHY?!!? WE NEVER PROGRAMMED THEM TO DO THIS!!!



```
static bool isCrazyMurderingRobot = false;
```

```
void interact_with_humans (void){  
    if(isCrazyMurderingRobot = true)  
        kill(humans);  
    else  
        be_nice_to(humans);  
}
```