SIT323 Practical Software Development, Trimester 2, 2020

Week 5 – Practical 4

Regular Expressions

Regular Expressions

Regular expressions will be useful to ensure a line of data or multiple lines of data in TAFF and CFF files conform to expectations.

You may use the ***Regex Practical*** program to create a specific ***regular expression pattern*** that matches each line in a set of data. Figure 1 depicts that the regular expression ^\d*$ matches each line in the data. The background colour of the TextBox containing a regular expression changes to green when the pattern matches each data line, otherwise it changes to pink as in Figure 2, even if one data line doesn't match the regular expression.

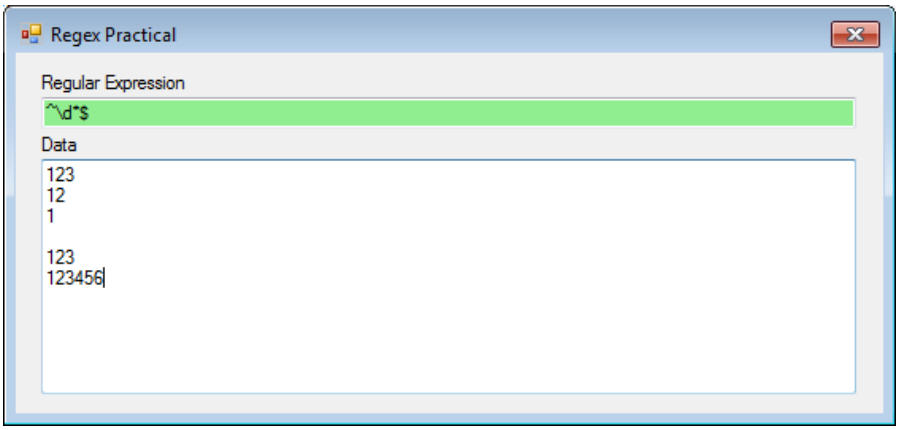

Figure 1. Each line of data matches the regular expression.

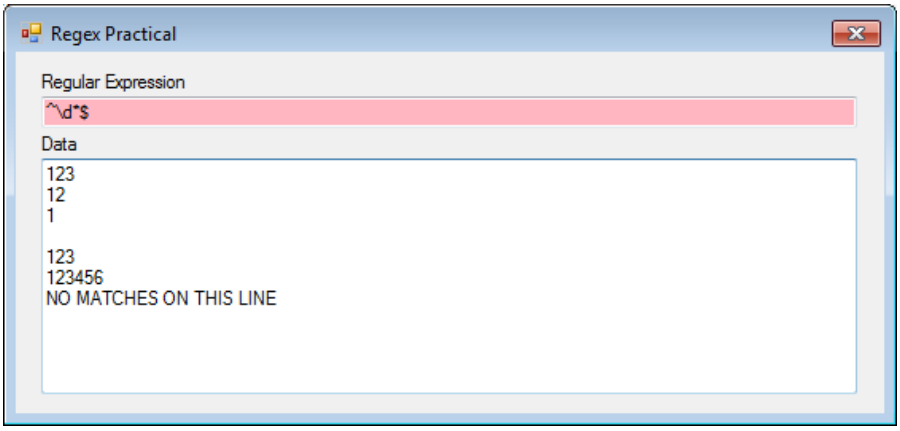

Figure 2. Each of the first 6 data lines match the regular expression, the last line doesn't not match.

Task 1 – Regular Expressions

(a) The regular expression pattern **^\s*$** matches any string containing 0 or more white spaces, i.e., a blank line.

^	represents the start of the string (not the first character)
\s	represents 1 white space character
.	represents 1 character
\d	represent one digit
*	represents 0 or more
+	represents 1 or more
$	represent the end of the string (not the last character)

Test the pattern **^\s*$** by using the Regex program and the following kinds of data.

1. A line containing several spaces.
2. A line containing several tabs.
3. A line containing a mixture of several spaces and several tabs.
4. A line containing no characters.

(b) The regular expression pattern ^\s*//.*$ matches a string containing a // comment such as the following.

```
// This is a comment.
    // This line starts with spaces and tabs, and ends with a comment.
```

Test this pattern ^\s*//.*$ by using the Regex program and the following kinds of data lines.

1. A line containing just //.
2. A line starting with // followed a few words.
3. A line starting with several spaces followed by // and a few words.
4. A line starting with several tabs followed by // and a few words.
5. A line starting with a mixture of several spaces and tabs followed by // and a few words.

(c) The regular expression pattern ^LIMITS-TASKS=\d+,\d+$ matches a string containing LIMITS-TASKS, an equal sign, one or more digits, a comma, and one or more digits such as the following.

LIMITS-TASKS=1,10
LIMITS-TASKS=2,100
LIMITS-TASKS=10,2500

Test this pattern ^LIMITS-TASKS=\d+,\d+$ by using the Regex program and several kinds of valid and invalid data lines.

(d) Create a regular expression pattern to match a string containing PROGRAM-DATA, an equal sign, one or more digits (followed by an optional period and one or more digits), a comma, one or more digits, a comma, one or more digits such as the following.

PROGRAM-DATA=3,5,3
PROGRAM-DATA=4.5,10,20
PROGRAM-DATA=25.667,100,1250

(e) Create a regular expression to match a string containing a non-empty, random sequence of 0s and 1s that are comma separated such as the following.

0
1
0,0,0,0,0
1,1,1,1,1
1,1,0,0,0
0,0,1,1,0,1,0,1
0,0,0,0,0,1,1,1,1,1