CS 3342 – Homework #2 (100 points)

Due Date: March 5, 2023 (Sunday) - 11:59pm

Converting Regular Expressions to NFA/DFA

Part one: (50 points)

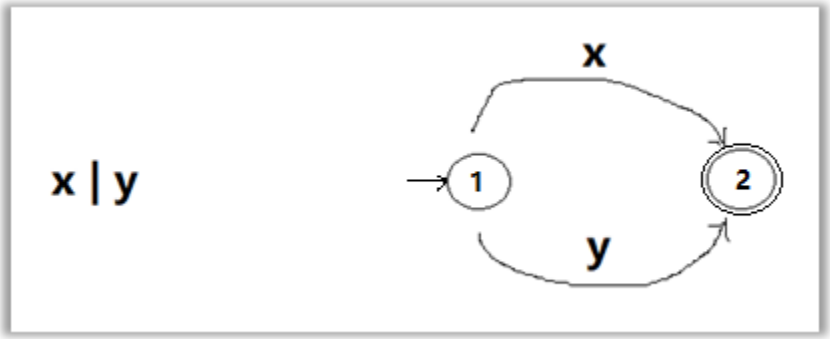
Convert the following regular expressions to NFA/DFA. Draw the new NFA/DFA and place them in a Word Doc.

1. (a|b)cd*
2. a(bc)*d
3. (a|b)*(abb|a+b)

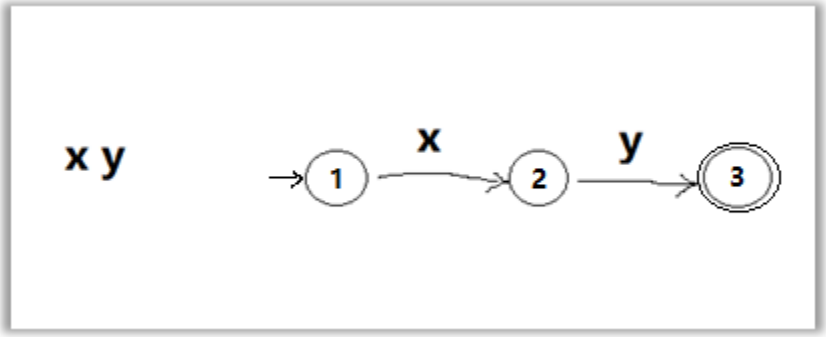
The following are simple examples from the lecture slides.

Regular Expression to NFA/DFA:

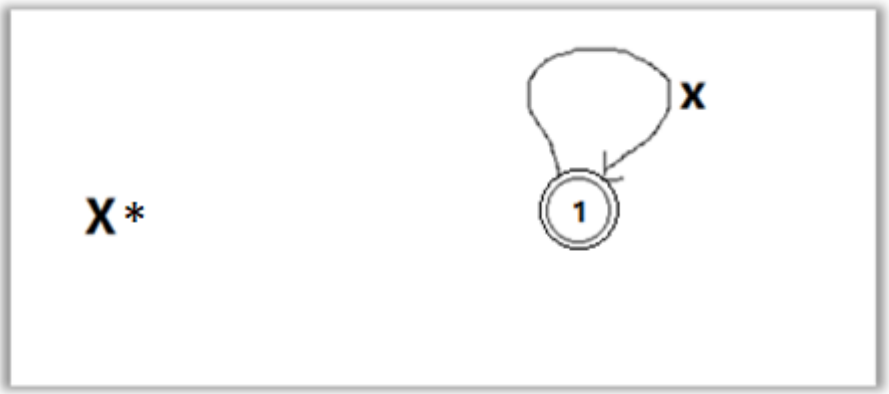
1.



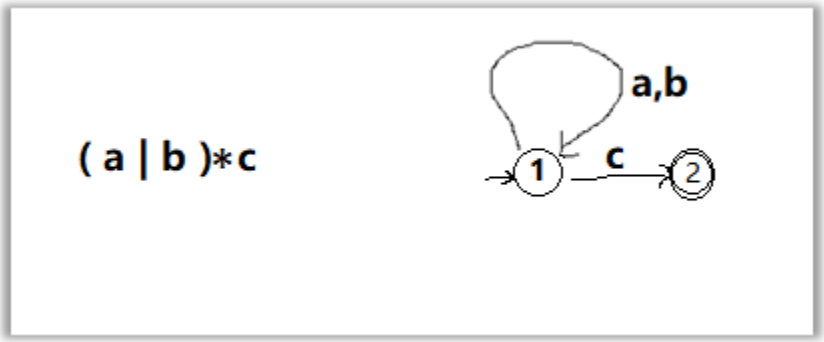
2.



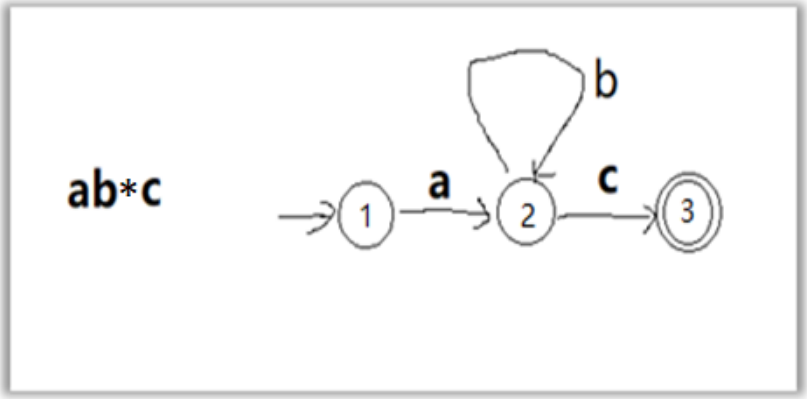
3.



4.



5.



Sample code:
string function1()

```
{
        string str = "";
        c = NextChar();
        if (c == 'a')
        {
           str = str + c;
           c = NextChar();
           while (c == 'b')
           {
                   str = str + c;
                  c = NextChar();
           }
           if (c == 'c')
           {
              str = str + c;
              return str;
            }
        return error;
}
```

Part Two: (50 points)

Write the corresponding code in Java without using 'java.util.regex' to recognize the following regular expressions.

1. (a|b)cd*
2. a(bc)*d

The Java program should read an input file and try to find the lexemes/tokens that match the REs. The content in the input file looks like this:

abc acdd abcbcd ac bcddd ad
abbbbbbc acddddddd abcbcbcbcd

A sample program and a sample input file will be given to you on Canvas. The output of the program should be like the following. The sample program is written for RE: ab*c.

```
This is line 1    :  abc acdd abcbcd ac bcddd ad
 ********* This is for RE - ab*c
Input string is :  abc
    From main :  Pattern found - abc
Input string is :  acdd
    From main :  Reject - acdd
Input string is :  abcbcd
```

```
    From main :  Reject - abcbcd
Input string is :  ac
    From main :  Pattern found - ac
Input string is :  bcddd
    From main :  Reject - bcddd
Input string is :  ad
    From main :  Reject - ad
This is line 2    :  abbbbbbc acddddddd abcbcbcbcd
 ********* This is for RE - ab*c
Input string is :  abbbbbbc
    From main :  Pattern found - abbbbbbc
Input string is :  acddddddd
    From main :  Reject - acddddddd
Input string is :  abcbcbcbcd
    From main :  Reject – abcbcbcbcd
```

You need to modify the program or write your own program to accept or reject the input strings with the two REs below:

1. (a|b)cd*
2. a(bc)*d

Your output should look like the following:

```
This is line 1    :  abc acdd abcbcd ac bcddd ad
 ********* This is for the first RE - (a|b)cd*
  RE is : (a|b)cd* & input string is :  abc
     From main   :  Reject - abc
  RE is : (a|b)cd* & input string is :  acdd
     From main   :  Accept  - acdd
  RE is : (a|b)cd* & input string is :  abcbcd
     From main   :  Reject - abcbcd
  RE is : (a|b)cd* & input string is :  ac
     From main   :  Accept  - ac
  RE is : (a|b)cd* & input string is :  bcddd
     From main   :  Accept  - bcddd
  RE is : (a|b)cd* & input string is :  ad
     From main   :  Reject - ad
 ********* This is for the second RE - a(bc)*d
  RE is : a(bc)*d & input string is :  abc
     From main   :  Reject - abc
  RE is : a(bc)*d & input string is :  acdd
     From main   :  Reject - acdd
  RE is : a(bc)*d & input string is :  abcbcd
```

```
    From main   :  Accept  - abcbcd
  RE is : a(bc)*d & input string is :  ac
    From main   :  Reject - ac
  RE is : a(bc)*d & input string is :  bcddd
    From main   :  Reject - bcddd
  RE is : a(bc)*d & input string is :  ad
    From main   :  Accept  - ad
This is line 2    :  abbbbbbc acddddddd abcbcbcbcd
 ********* This is for the first RE - (a|b)cd*
  RE is : (a|b)cd* & input string is :  abbbbbbc
    From main   :  Reject - abbbbbbc
  RE is : (a|b)cd* & input string is :  acddddddd
    From main   :  Accept  - acddddddd
  RE is : (a|b)cd* & input string is :  abcbcbcbcd
    From main   :  Reject - abcbcbcbcd
 ********* This is for the second RE - a(bc)*d
  RE is : a(bc)*d & input string is :  abbbbbbc
    From main   :  Reject - abbbbbbc
  RE is : a(bc)*d & input string is :  acddddddd
    From main   :  Reject - acddddddd
  RE is : a(bc)*d & input string is :  abcbcbcbcd
    From main   :  Accept  - abcbcbcbcd
```

Take a screen shot of your output and put it in a Word doc. **Please do not zip the Word Doc.** Zip the source file. Submit the word document and the source file separately to Canvas.

You can put the answers for part A and the output for part B in the same word document.