1. Simplify the following Boolean expression:

$$A\left(A+BC\right)+B\left(B+AC\right)$$

A + B

2. List all ordered triples (A,B,C) that make the following expression FALSE:

$$\overline{AB}+A(\overline{B+C})$$

(1, 1, 0) (1, 1, 1)

3. Simplify the following expression:

$$B(\overline{A}+B)+A(\overline{B}+C)+\overline{C}(A+B)$$

A + B

4. How many ordered triples make the following expression TRUE?

$$\overline{A}(B+\overline{C})+A\overline{B}C$$

4

5. Given that the command SWITCH changes a stack to a queue or vice versa and REVERSE switches the order of a given stack or queue, what is the next item to be POPPED from an initially empty queue?

PUSH(S); PUSH(U); PUSH(N); SWITCH; PUSH(S); PUSH(H);
PUSH(I); REVERSE; PUSH(N); PUSH(E); SWITCH; PUSH(S);
PUSH (T); PUSH (A); PUSH (T); PUSH (E); POP(X); POP(X);
SWITCH; POP(X); POP(X); REVERSE; POP(X); POP(X)

U

6. What is the depth of the binary search tree for **NORTHHOLLYWOODACSL**?

5

7. What is outputted when this program is executed? 11

The data inputted is: 1, 1, 2, 5, 8, 13, 34, 55, 89

```
d = 0
for i = 0 to 2
    for j = 0 to 2
        input a(i, j)
    next j
next i

for i = 1 to 2
    for j = 1 to 2
        if a(i,j) / (i*j) == int(a(i,j) / (i*j) then
            a(i,j) =1
        else
            a(i,j) = int(a(i,j) / (i*j))
        end if
    next j
next i

for i = 0 to 2
    for j = 0 to 2
        if a(i, j) > 25 then
            a(i, j) = a(i, j) − 25
        end if
        a(i, j) = a(i, j) % 5
    next j
next i

for i = 0 to 2
    d = d + a(i, i) + a(i, 2-i)
next i

output d

end
```

RegEx in Practice

Programmers use Regular Expressions (usually referred to as **regex**) extensively for expressing patterns to search for. All modern programming languages have regular expression libraries. Unfortunately, the specific syntax rules vary depending on the specific implementation, programming language, or library in use. Interactive websites for testing regexes are a useful resource for learning regexes by experimentation. An excellent online tool is https://regex101.com/.

Here are the additional syntax rules that we will use. They are pretty universal across all regex packages.

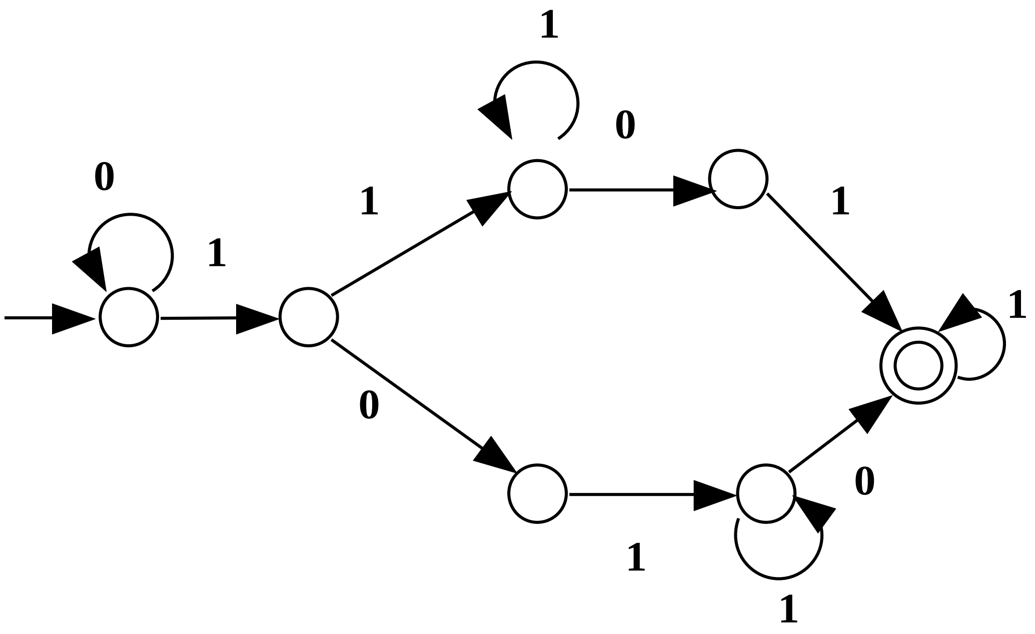
<table>
<tr><th>Pattern</th><th>Description</th></tr>
<tr><td>|</td><td>As described above, a vertical bar separates alternatives. For example, gray|grey can match "gray" or "grey".</td></tr>
<tr><td>*</td><td>As described above, the asterisk indicates zero or more occurrences of the preceding element. For example, ab*c matches "ac", "abc", "abbc", "abbbc", and so on.</td></tr>
<tr><td>?</td><td>The question mark indicates zero or one occurrences of the preceding element. For example, colou?r matches both "color" and "colour".</td></tr>
<tr><td>+</td><td>The plus sign indicates one or more occurrences of the preceding element. For example, ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac".</td></tr>
<tr><td>.</td><td>The wildcard . matches any character. For example, a.b matches any string that contains an "a", then any other character, and then a "b" such as "a7b", "a&b", or "arb", but not "abbb". Therefore, a.*b matches any string that contains an "a" and a "b" with 0 or more characters in between. This includes "ab", "acb", or "a123456789b".</td></tr>
<tr><td>[ ]</td><td>A bracket expression matches a single character that is contained within the brackets. For example, [abc] matches "a", "b", or "c". [a-z] specifies a range which matches any lowercase letter from "a" to "z". These forms can be mixed: [abcx-z] matches "a", "b", "c", "x", "y", or "z", as does [a-cx-z].</td></tr>
<tr><td>[^]</td><td>Matches a single character that is not contained within the brackets. For example, [^abc] matches any character other than "a", "b", or "c". [^a-z] matches any single character that is not a lowercase letter from "a" to "z". Likewise, literal characters and ranges can be mixed.</td></tr>
<tr><td>( )</td><td>As described above, parentheses define a sub-expression. For example, the pattern H(ä|ae?)ndel matches "Handel", "Händel", and "Haendel".</td></tr>
</table>

8. What is the length of the smallest string that can be produced by the following regular expression? 12

ab*ba(ab **U** aa*b)a(b **U** ab*a)a(ab **U** (a U b))b*bab

9. Which regular expressions are accepted by the following FSA? 3

I) 101101101 II) 111100111 III) 1011101111 IV) 1010101 V) 11111111111



10. Given the following regular expression: 1*01(01)*1100* which of the following strings match the pattern? B, E

A. 0010100

B. 101011100

C. 01010101100

D. 1010110

E. 01110