1. **What Does This Program Do? (BASIC)**
   What is printed when the following program is run?

   ```bASIc
   10 X = 1 + 9 - ((9 + 3 * 9) - 4)
   20 PRINT X
   30 END
   ```

2. **What Does This Program Do? (BASIC)**
   When the following program is run, how many times is ACSL printed?

   ```bASIc
   10 FOR X = 1 TO 20 STEP 2
   20 FOR Y = 4 TO 20 STEP 4
   30 FOR Z = 7 TO 1 STEP -2
   40 IF (Z>Y) AND (X<Z) THEN PRINT "ACSL"
   50 NEXT Z
   60 NEXT Y
   70 NEXT X
   80 END
   ```

3. **Computer Number Systems**
   What is the length of the longest string of consecutive 1's in the binary representation of 1F2E3D4C5B6A79816?

4. **Computer Number Systems**
   Which of the following is the largest?

<table>
<thead>
<tr>
<th>101010011₂</th>
<th>52₇</th>
<th>155₁₆</th>
<th>34₇₁₀</th>
</tr>
</thead>
</table>

5. **Computer Number Systems**
   Solve for X in the following octal equation.

   \[ X = 37₄₂₁ + 3₄₁₅ \]
1. Evaluation is as follows:

\[
1 + 9 - ((9 + 3 * 9) - 4) = 10 - ((9 + 3 * 9) - 4) \\
= 10 - ((9 + 27) - 4) \\
= 10 - (36 - 4) \\
= 10 - 32 \\
= -22
\]

2. The string ACSL is printed whenever line 40 is true. This happens when \(Z\) is larger than both \(X\) and \(Y\). Let’s consider what values the three variables take on: \(X\) is 1, 3, 5, 7, 9, 11, 13, 15, 17, 19; \(Y\) is 4, 8, 12, 16, 20; and \(Z\) is 7, 5, 3, 1. When \(Z\) is 7, there is only one value of \(Y\) less than 7, the initial value of 4. \(X\) can be 1, 3, or 5. When \(Z\) is 5, again, there is only one value of \(Y\) that is less than 5. But \(X\) can be 1 or 3 (not 5). When \(Z\) is 3 or 1, there are no values of \(Y\) less than it. Thus, the 5 triples, \((x, y, z)\), for which ACSL is printed are \((1, 4, 7)\), \((3, 4, 7)\), \((5, 4, 7)\), \((1, 4, 5)\), and \((3, 4, 5)\).

3. Just convert each hex number to binary, and inspect the resulting string. The largest consecutive run of 1s appears at the beginning:

\[
1F2E3D4C5B6A798_{16} = 0001\ 1111\ 0010\ 1110\ 0011...\]

4. To compare the numbers, we first need to convert them all to a common base. Since one of the numbers is already in base 10, it’s probably just as easy to convert them all to base 10:

\[
101010011_{10} = 339_{10} \\
527_{8} = 343_{10} \\
155_{10} = 341_{10}
\]

5. It’s probably easiest to solve this problem by doing all the additions in base 8, rather than converting each number to decimal, summing, and converting the sum back to octal. The addition proceeds from the right to the left (the base of each number is 8, unless otherwise noted):

\[
1 + 5 = 6 \\
2 + 1 = 3 \\
4 + 5 = 9_{10} = 11 \\
carry + 7 + 3 = 11_{10} = 13 \\
carry + 3 = 4
\]
1. What Does This Program Do? (BASIC)
   When the following program is run, how many times is “ACSL” printed?

   ```basic
   FOR X = 1 TO 7 STEP 4
     FOR Y = 1 TO 7 STEP 3
       FOR Z = 1 TO 7 STEP 2
         PRINT "ACSL"
       NEXT Z
     NEXT Y
   NEXT X
   ```

2. Recursive Functions
   Find $f(20)$ where
   
   $$f(x) = \begin{cases} 
   f(x - 5) + 6 & \text{if } x > 12 \\
   f(x - 3) + 4 & \text{if } 12 \geq x > 1 \\
   -1 & \text{otherwise}
   \end{cases}$$

3. Recursive Functions
   Evaluate $f(10)$ where
   
   $$f(x) = \begin{cases} 
   f(1 + f(x - 3)) - 2 & \text{if } x \geq 7 \\
   1 + f(x - 4) & \text{if } -1 < x < 7 \\
   2x + 3 & \text{otherwise}
   \end{cases}$$

4. Boolean Algebra
   Simplify the following expression so that it uses at most one NOT operator.
   
   $$\overline{A + B} + \overline{B} + \overline{B}$$

5. Boolean Algebra
   List all ordered triples that satisfy the following expression.
   
   $$AB + A(B + \overline{A})C$$
1. The outer loop happens twice, with \( X = 1 \) and 5. Each time through the outer loop, the middle loop executes with \( Y = 1, 4, \) and 7. Each time through the middle loop, the inner loop executes with \( Z = 1, 3, 5, \) and 7. So the \texttt{PRINT} statement is executed \( 2 \cdot 3 \cdot 4 = 24 \) times.

2. The evaluation is as follows:

   \[
   f(20) = f(15) + 6 \\
   = (f(10) + 6) + 6 = f(10) + 12 \\
   = (f(7) + 4) + 12 = f(7) + 16 \\
   = (f(4) + 4) + 16 = f(4) + 20 \\
   = (f(1) + 4) + 20 = f(1) + 24 \\
   = -1 + 24 = 23
   \]

3. The evaluation is as follows:

   \[
   f(10) = f(1 + f(7)) - 2 \\
   f(7) = f(1 + f(4)) - 2 \\
   f(4) = 1 + f(0) \\
   f(0) = 1 + f(-4) \\
   f(-4) = 2 \cdot (-4) + 3 = -5 \\
   \Rightarrow f(0) = 1 + (-5) = -4 \\
   \Rightarrow f(4) = 1 + (-4) = -3 \\
   \Rightarrow f(7) = f(1 + (-3)) - 2 = f(-2) - 2 \\
   = (2 \cdot (-2) + 3) - 2 = (-4 + 3) - 2 = -3 \\
   \Rightarrow f(10) = f(1 + (-3)) - 2 = f(-2) - 2 = -3
   \]

4. The simplification uses DeMorgan’s Law extensively: twice in the first step, and again in the last step:

   \[
   \overline{A + B + AB} + \overline{B} = (\overline{A B}) + (\overline{A} + \overline{B}) + \overline{B} \\
   = \overline{A} \overline{X} + \overline{A} + \overline{B} \\
   = \overline{B} (\overline{A} + 1) + \overline{A} \\
   = \overline{B} + \overline{A} \\
   = \overline{B} A
   \]

   Since the AND operator is commutative, \( \overline{AB} \) is also a correct answer.

5. Because an \( A \) is a factor of both terms, it must be 1. Substitute \( A = 1 \) and simplify:

   \[
   AB + A(B + AC)C = B + (B + C)C \\
   = B + BC + C \overline{C} \\
   = B + BC + 0 \\
   = B(1 + C) = B
   \]

   There are no constraints on the value of \( C \), so the solutions to the expression are \((1, 1, *)\).
1. What Does This Program Do? (BASIC)
   For what values of \( X \) and \( Y \) will the following program print “ACSL”? 
   
   ```basic
   10 IF X+2=10-3*Y AND Y=5*X-8 THEN PRINT "ACSL"
   20 END
   ```
   
   \( X = \) \_
   \( Y = \) \_

2. Bit String Flicking
   Evaluate the following:
   
   \((\text{LCIRC-2 (RSHIFT-1 10110)}) \text{ XOR 01110 AND 11101})\)

3. Bit String Flicking
   Find all values of \( x \) (5 bits long) for which the following expression has a value of 00110.

   \((\text{LSHIFT}-1 \text{ (RCIRC-4 x)})\)

4. Digital Electronics
   How many triples make the following circuit true?

   ![Circuit Diagram]

5. Digital Electronics
   Write a Boolean Algebra expression that represents the following circuit. Do not simplify your expression.

   ![Circuit Diagram]
1. “ACSL” is printed when two conditions hold: \( x + 2 = 10 - 3y \) and \( y = 5x - 8 \). These two linear equations can be solved as follows:

\[
\begin{align*}
  x + 2 &= 10 - 3y \\
  y &= 5x - 8
\end{align*}
\]

Substituting for \( y \) into the first expression and solve for \( x \):

\[
\begin{align*}
  x + 2 &= 10 - 3(5x - 8) \\
  x + 2 &= 10 - 15x + 24 \\
  16x &= 32 \\
  x &= 2
\end{align*}
\]

Finally, substitute for \( x \) into either equation to solve for \( y \).

\[
X = \boxed{2} \quad \text{and} \quad Y = \boxed{2}
\]

2. The evaluation is as follows:

\[
\text{LCIRC-2 (RSHIFT-1 10110) XOR 01110 AND 11101} = \text{LCIRC-2 01011) XOR 01100 = 01101 XOR 01100 = 00001}
\]

A common mistake is to evaluate the XOR before the AND.

3. Represent \( x \) by \( abcde \) and simplify: \( \text{RCIRC-4 abcde} \) evaluates to \( bcdea \), and \( \text{LSHIFT-1 bcdea} \) to \( cdea0 \). Setting this equal to \( 00110 \), we observe that \( c=0, \ d=0, \ e=1, \) and \( a=1 \). There are no constraints on the value of bit \( b \).

4. The circuit can be represented by the equation

\[
AB \oplus (B + C)
\]

which has the following truth table:

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( C )</th>
<th>( AB )</th>
<th>( B + C )</th>
<th>( AB \oplus (B + C) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

5. It’s OK to commute the operands of the OR and XOR operators. It isn’t OK to simplify the expression in any way. The parentheses are needed as shown.

\[
(x + y) \oplus y
\]
1. **What Does This Program Do?** (Pascal)

   In the following program, the array `a` contains the characters “INTERMEDIATE DIVISION”. That is, `a[1]` contains the letter “I” and `a[20]` contains the letter “N”. What are the final values of `x` and `y` when procedure `test` returns?

   ```pascal
   procedure test;
   var m, x, y: integer;
   begin
     x:=1; m:=0;
     while a[x] < a[x+1] do
     begin
       y := x;
       repeat
         x := x + 1;
       until a[y] > a[x];
       m := y - x;
     end;
   end;
   ```

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Prefix/Infix/Postfix Notation**

   Evaluate the following postfix expression:

   ```latex
   4 3 2 \text滔} 2 3 4 \text滔} -
   ```

3. **Prefix/Infix/Postfix Notation**

   Translate the following into postfix:

   ```latex
   2A^4 + 6B - \frac{5C}{3}
   ```

4. **Data Structures**

   Build a binary search tree with the letters C O N T E S T F O U R, starting with the C and ending with the R. How many nodes have 1 child?

5. **Data Structures**

   Build a binary search tree with the letters H A R D E R. What is the depth of the tree?
1. The while loop terminates when $x$ is the index of the first “D”, whereas the until loop terminates at the rightmost “D”. (What would happen if the last character in a were a “D”?)

2. The expression converts into infix as follows:

$$4 \ 3 \ 2 \ \uparrow \ * \ 2 \ 3 \ 4 \ \uparrow \ * \ -$$

$$4 \ \frac{3^4}{2} \ * \ 2 \ \frac{3^4}{2} \ -$$

$$4 \ \frac{3^4}{2} \ - \ 2 \ \frac{3^4}{2}$$

$-126$

3. Because addition and subtraction have the same level and precedence and because they group from left to right, the fully parenthesized expression to convert into postfix looks as follows:

$$(2A^4 + 6B) - \frac{5C}{3}$$

The expression converts to $\overline{[x \ y]} -$, where $\overline{x}$ is $\overline{p \ q \ +}$, and $\overline{y}$ is $\overline{5 \ c \ * \ 3}$.

And $\overline{p}$ is $\overline{2 \ a \ 4 \ \uparrow \ *}$, and $\overline{q}$ is $\overline{6 \ b \ *}$.

4. The resulting tree looks like this:

```
C
  O
    N
      E
  D
      F
T
  S
U
  R
B
```

5. The resulting tree looks like this:

```
H
  A
    D
      E
  R
```