Forming a Team in your Middle/High School and joining the American Computer Science League

Mario Correia
Barrington High School
What is the American Computer Science League (ACSL) ?

- Organize and oversee Computer Science and Computer Programming Contests
- 42nd year providing Contests
- Hundreds of teams compete from four continents: North America, Europe, Africa and Asia
- Is on approved list of National Association of Secondary School Principals (NASSP)
- Institutional member of CSTA
What are some of the benefits of joining ACSL?

- Promotes CS
- CS Enrichment
- Develop and deepen Program Solving Skills
- Participate in four Contests Yearly
- Camaraderie and Team building in Academic setting
What support is available for the ACSL Advisor of your school?

- Online Category Rule Book
- Sample Questions
- Sample Programming problems
- Website with rules, schedules, all topics covered
- Wiki page with each category description
- Online videos illustrating solutions to Questions
ACSL Categories

- Assembly Language Programming
- Bit-String Flicking
- Boolean Algebra
- Computer Number Systems
- Data Structures
- Digital Electronics

- Graph Theory
- Lisp Programming
- Prefix/Infix/Postfix Notation
- Recursive Functions
- Regular Expressions
- What Does This Program Do?
1. Assembly Language

- Load
- Store
- Add
- Subtract
- Multiply
- Divide

After the following program is executed, what value is in location TEMP?

<table>
<thead>
<tr>
<th>TEMP</th>
<th>DC</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>DC</td>
<td>8</td>
</tr>
<tr>
<td>B</td>
<td>DC</td>
<td>-2</td>
</tr>
<tr>
<td>C</td>
<td>DC</td>
<td>3</td>
</tr>
<tr>
<td>LOAD</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>MULT</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>SUB</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>STORE</td>
<td>TEMP</td>
<td></td>
</tr>
</tbody>
</table>

The ACC takes on values -2, -6, 2, -1 and -9 in that order. The last value, -9, is stored in TEMP.
2. Bit-String Flicking

- Circulates
- Shifts
- And
- Or
- Xor
- Not

Evaluate the following expression:

\[(\text{RSHIFT-1} (\text{LCIRC-4} (\text{RCIRC-2} 01101)))\]

The expression evaluates as follows:

\[(\text{RSHIFT-1} (\text{LCIRC-4} (\text{RCIRC-2} 01101)))\]
\[= (\text{RSHIFT-1} (\text{LCIRC-4} 01011))\]
\[= (\text{RSHIFT-1} 10101) = 01010\]
3. Boolean Algebra

- Simplify Expression
- Solutions that make expression true? false?
- How many solutions that make expression true? false?

Find all ordered pairs \((A, B)\) that make the following expression TRUE.

\[
(A + B) + AB
\]

| \(A, B\), \(A + B\), \(\sim A + B\), \(AB\), \(\sim A + B + AB\), \(\sim A + B + AB\) |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
4. Computer Number Systems

Number Systems:
- Binary
- Octal
- Decimal
- Hexadecimal

Solve Equations multiple number Systems

Solve for $X$.

$X_{16} = 3676_8$

$3676_8 = 01111011110_8$

$= 01111011110_8$

$= 7BE_{16}$
5. Data Structures

- Stacks
- Queues
- Binary Search Tree
- Priority Queues

Define the function REV to reverse the order of the items in the current stack before continuing.

What would the next popped element be in the following stack?
PUSH(B), PUSH(L), PUSH(E), POP(X), REV, PUSH(A), PUSH(K), REV, POP(X), POP(X), REV, PUSH(H), POP(X), POP(X), PUSH(O), REV, POP(X), PUSH(S), PUSH(E), REV, POP(X)

Data Structures
The stack is constructed as follows:
BLE, BL, LB, LBAK, KABL, KA, AK, AKH, A, AO, OA, O, OSE, ESO, ES. The next popped item is S since stacks are LIFO.
6. Digital Electronics

- How many solutions makeExpression: True? False?
- Simplify an Expression

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>AB</th>
<th>B + C</th>
<th>⊕</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</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>0</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>
<td>1</td>
</tr>
</tbody>
</table>
7. Graph Theory

In a Directed Graph find:
- the number of cycles
- the number of paths of a Specific length
Construct Adjacency Matrix

Find the number of different cycles contained in the directed graph with vertices
\{A, B, C, D, E\}
and edges
\{AB, BA, BC, CD, DC, DB, DE\}.

The graph is as follows:

By inspection, the cycles are: \{A,B\}, \{B,C,D\} and \{C,D\}. Thus, there are 3 cycles in the graph.
8. Lisp
(List Processing Language)

Arithmetic Operations:
Add, Subtract, Multiply, Divide,
Square, Exponents

Lisp Functions:
Car, Cdr, Reverse, Cons

Evaluate: \( \text{CDR } '((2 (3)) (4 (5 6) 7)) \)

The CDR function takes the first element of its parameter (which is assumed to be a list) and returns the modified list. The first element of the list is \( (2 (3)) \) and the list without this element is \( ((4 (5 6) 7)) \).
9. Prefix/Infix/Postfix Notation

Convert from:
- Infix to Prefix, Infix to Postfix
- Prefix to Postfix
- Postfix to Prefix

Evaluate Expressions in the form of:
- Infix, Postfix, Prefix
10. Recursive Functions

Find the value of a Recursive Function of the form:

\[ f(x), \ f(x,y), \ f(f(x)) \]

From a word problem find a recursive solution.

Find \( f(6) \), given:

\[
f(x) = \begin{cases} 
  f(f(x-2)) + 1 & \text{if } x > 1 \\
  x + 1 & \text{if } x \leq 1 
\end{cases}
\]

Working backwards, we get

\[
\begin{align*}
  f(0) &= 1 \\
  f(1) &= 2 \\
  f(2) &= f(f(0)) + 1 = f(1) + 1 = 2 + 1 = 3 \\
  f(3) &= f(f(1)) + 1 = f(2) + 1 = 3 + 1 = 4 \\
  f(4) &= f(f(2)) + 1 = f(3) + 1 = 4 + 1 = 5 \\
  f(5) &= f(f(3)) + 1 = f(4) + 1 = 5 + 1 = 6 \\
  f(6) &= f(f(4)) + 1 = f(5) + 1 = 6 + 1 = 7
\end{align*}
\]
11. FSA’s and Regular Expressions

- Translate FSA to Regular Expression
- Simplify Regular Expression
- Determine if Regular Expressions are equivalent
- Determine if a String is accepted by a Regular Expression or FSA
12. What Does This Program Do?

Based on Pseudo-code used in APCS Principles course.

Use the following constructs:
- Operators: ABS, SQR, INT, Modulus Division
- If then else Statements
- For and While Loops
- Arrays
- Strings

How many of the entries are not 0 after this program is run?
```
Dim a(100)
For i = 2 To 25
    a(i) = i
Next i
For k = 2 To 25
    If a(k) = 0 Then
        For j = 2*k To 25 Step k
            a(j) = 0
        Next j
    End If
Next k
```

This program counts the number of primes from 2 to 25. They are 2, 3, 5, 7, 11, 13, 17, 19, and 23. There are 9 of them. It uses the method called the Sieve of Eratosthenes.
ACSL Programming Contests

- Programming Problem given for each Contest
- Provided with text files:
  - Sample Input
  - Test Input
Programming Contest #1

Here is a list of typical problems. Students are asked to develop an algorithm and solve:

- card games
- Number patterns
- Board games
- algebraic/geometry problem
- Sports games

Sample Programming Problem. Taken from 2011-12 Senior Contest #1

Draft Picks: See Handout Page 2:

For most of this past summer, there was still a lockout and the potential for not having a 2012 National Football League (NFL) season. Two of the sticking points between owners and players were the high price for first round draft picks and the extension of the season to 18 games. The following statistics are available for the top 10 first-round draft picks in 2008.....
Programming Contest #2

Typically solve a String based question. Examples:
- String manipulation
- Find matching Strings
- Binary Bits

Sample Programming Problem. Taken from 2014-15 Intermediate Contest #2 Lisp Expressions:
See Handout Page 4:

PROBLEM: Given a LISP expression, perform operations on the expression. There will be no list elements that also contain a list such as (A (B (C D))), which has one atom and one list but the list contains a sublist.....
Programming Contest #3

Two-Dimensional Arrays
- Board Games
- Traversals/find a path
- Piece things together

Sample Programming Problem. Taken from 2014-15 Senior Contest #3 ACSL Isola:
See Handout Page 5:

ACSL Isola is a board game played by 2 players with each player having just one marker. The game has 49 grid squares as shown above.
Programming Contest #4

Various types of problems, has consisted of:
- Puzzles
- Stacks
- Binary Trees
- Prefix Expressions
- Matrices

Sample Programming Problem. Taken from 2001-02 Junior Contest #4 Hairy Conic Junior:
See Handout Page 6:

The figure on the left above is an ellipse and the figure on the right above is a circle. They are related mathematically because the equations used to describe each figure are similar. The general form of the equation for both is $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$. ....
ACSL Divisions

- Elementary
- Classroom
- Junior
- Intermediate
- Senior
Elementary Division

- Open to students grades 3-6
- Non programming problems
- Four categories, one each contest
- Categories are: Computer Number Systems, Prefix/Infix/Postfix, Boolean Algebra, Graph Theory
- 30 minute contest, 5 questions
- Top 5 student scores submitted
- Prizes awarded to top scoring students and teams.
- Do not participate in ALL Stars

1. What is the value of $11101001011_2$ in base 16?
   
   $= 0111 0100 1011_2 = 74B_{16}$

2. Translate the following infix expression into prefix.
   
   $(6 + 3) / (7 - 1) * 3 ^ 2$

   The final answer is $* /+ 6 3 - 7 1 ^ 3 2$

3. Simplify $(\text{NOT} A \text{ OR} A \text{ AND} \text{ NOT} B) \text{ AND} (\text{NOT} A \text{ AND} B)$

   $= \text{NOT} A \text{ AND} B \text{ OR} 0 = \text{NOT} A \text{ AND} B$

4. Find the number of different cycles contained in the graph with vertices $\{A,B,C,D\}$ and edges $\{AB, BC, AC, AD, DB\}$

   the cycles are ABDA and ABCA.
Classroom Division

- Open to students from all grades
- Consists of problems from the other three Divisions
- Consists of all 12 Topics
- 50 minute Contest, 10 questions
- Top 5 student scores submitted
- Prizes awarded to top scoring students and teams.
- Do not participate in ALL Stars

Prefix-Infix-Postfix
Rewrite the following infix expression in postfix:

\[ A \ ( B \ + \ D) \ / \ ( C \ - \ E) \]

What Does This Program Do - Looping
When the following program is run, what is the last element of array B that is modified?

```
FOR J = 5 TO 1 STEP -1
    FOR K = 3 TO 10 STEP 2
        B(J,K) = J + K
    NEXT K
NEXT J
```

Digital Electronics
Find all ordered triplets \((A, B, C)\) that make the following circuit TRUE.
Junior Division

- Open to Junior High/Middle School students up to 9th grade
- Covers 9 of the 12 Topics (excluding Lisp, FSA Regular Expressions and Assembly Language)
- 30 minute contest - 5 short answer questions
- Program to solve over 72 hours
- Top 5 student scores submitted
- Prizes awarded to top scoring students and teams.
- Top teams invited to ALL Stars
Intermediate Division

- Geared to high school students with little or no programming experience or advanced junior high students
- Cover all 12 topics
- 30 minute contest - 5 short answer questions
- Program to solve over 72 hours
- Top 3 or 5 student scores submitted (depending on Team size)
- Prizes awarded to top scoring students and teams.
- Top teams invited to ALL Stars
Senior Division

- Geared for high school students with programming experience
- Cover all 12 topics
- 30 minute contest - 5 short answer questions
- Program to solve over 72 hours
- Top 3 or 5 student scores submitted (depending on Team size)
- Prizes awarded to top scoring students and teams.
- Top teams invited to ALL Stars

Computer Number Systems
Evaluate the following expression and express the answer in octal:

\[ 2_{10} \times 416_8 + A23_{16} - 10110_2 + 31_3 / 101_2 \]

Prefix/Infix/Postfix
Define a \# b = (a+b)/2 (arithmetic mean) and a @ b = \sqrt{ab} (geometric mean). Evaluate the following prefix expression:
(Note: all numbers are single digits)

\[-# * 8 6 / 4 2 - @ * @ 9 4 3 * \uparrow 5 2 2 \uparrow \# 8 2 3\]

Boolean Algebra
How many ordered triples make the following Boolean expression TRUE?

\[ \overline{B} (AC + BC) + C (AC + B) + \overline{B} + C \]

Data Structures
Create a binary search tree from the string CINDERELLA. Then add, one at a time, the strings ARIEL, MULAN and BELLE to the original tree. Which one increases the depth of the tree by the most?
What team can I form in the Elementary, Classroom and Junior Divisions?

- In each of these Divisions they may consist of as many students as you wish.
- Each contest you submit your Top Five scores.
What teams can I form in the Intermediate and Senior Divisions?

- Team is either considered a Three Person or a Five Person Team
- For a Three Person team you submit your Top Three scores.
- For a Five Person team you submit your Top Five scores.
- Each school may only have either a 3-person or 5-person team per Division.
Role of the Advisor:

- Conduct Meetings to prepare for Contests
- Via email receive Contest material at least one month prior to Contest end date
- Material received will be Short Answer Questions and their Solutions
- For Junior, Intermediate and Senior Teams will receive Programming problem with Sample and Test Input provided in Text files
- Administer Short Answer Rounds following specified time limit
- For Junior, Intermediate and Senior Teams give students copy of programming problem to solve in specified time limit
- Grade short answer round and test programs
- Submit Team Scores
How ACSL Contests Work
Elementary Division:

- Five Questions per contest
- 30 minute time Limit
- Submit top Five scores for each contest
Classroom Division

- Ten Questions per contest
- 50 minute time Limit
- Submit top Five scores for each contest
Junior Division

- Five Questions per contest
- 30 minute time Limit
- Programming Problem - given 72 hours to complete
- Test program for 5 test inputs.
- Each individual student can earn up to 10 points per contest
- Submit top Five scores for each contest
Intermediate Division

- Five Questions per contest
- 30 minute time Limit
- Programming Problem - given 72 hours to complete
- Test program for 5 test inputs.
- Each individual student can earn up to 10 points per contest
- Submit top Five scores for 5-Person Team or Top Three scores for 3-Person Team
Senior Division

- Five Questions per contest
- 30 minute time Limit
- Programming Problem - given 72 hours to complete
- Test program for 5 test inputs.
- Each individual student can earn up to 10 points per contest
- Submit top Five scores for 5-Person Team or Top Three scores for 3-Person Team
Schedule
Contest 1 End Date: Dec. 20, 2019

Elementary Division:
- Computer Number System

Classroom Division
- Computer Number Systems
- Recursive Functions
- What Does This Program Do?

Junior, Intermediate, Senior Divisions:
- Computer Number Systems
- Recursive Functions
- What Does This Program Do?
Contest 2 End Date: Feb. 7, 2020

Elementary Division:
- Prefix/Infix/Postfix

Classroom Division
- Prefix/Infix/Postfix
- Bit String
- What Does This Program Do?
- Lisp

Junior Division:
- Prefix/Infix/Postfix
- Bit String
- What Does This Program Do?

Intermediate and Senior Divisions:
- Prefix/Infix/Postfix
- Bit String
- Lisp
Contest 3 End Date: March 13, 2020

Elementary Division:
- Boolean Algebra

Classroom Division
- Boolean Algebra
- Data Structures
- What Does This Program Do?
- FSA/Regular Expressions

Junior Division:
- Boolean Algebra

Intermediate and Senior Divisions:
- Data Structures
- FSA/Regular Expressions
- What Does This Program Do?
Contest 4 End Date: April 17, 2020

Elementary Division:
- Graph Theory

Classroom Division
- Graph Theory
- Digital Electronics
- What Does This Program Do?
- Assembly Language

Junior Division:
- Graph Theory
- Digital Electronics
- What Does This Program Do?

Intermediate and Senior Divisions:
- Graph Theory
- Digital Electronics
- Assembly Language
Sample Category Questions
(found in Handout at Page Indicated)  (Solutions to questions start on Page 15)

- Junior - Bit String Flicking  Pg.9
- Intermediate - Data Structures  Pg.9
- Senior - Computer Number Systems Pg. 10
- Intermediate - Recursive Functions  Pg. 11
- Junior - Graph Theory  Pg. 12
- Senior - Prefix/Infix/Postfix  Pg. 12
- Junior - Digital Electronics Pg. 13
- Intermediate - Lisp  Pg. 13
Registration Form

See Last Page in Handout