Q

American Computer Science League

2019-2020 **Intermediate and Classroom Shorts Solutions** **ACSL Finals**

---

1. Boolean Algebra

$$\overline{A + \overline{B}\,C} \; + \; \overline{B + \overline{A\,C}} \; + \; \overline{\overline{C} + A\,B}$$

$$= \overline{A}\,\overline{\overline{B}\,C} \; + \overline{B}\,\overline{\overline{A\,C}} \; + \overline{\overline{C}}\,\overline{A\,B}$$

$$= \overline{A}(B + \overline{C}) \; + \overline{B}(A\,C) \; + C(\overline{A} + \overline{B})$$

$$= \overline{A}\,B + \overline{A}\,\overline{C} + A\,\overline{B}\,C + \overline{A}\,C + \overline{B}\,C$$

$$= \overline{A}\,B + \overline{A}(\overline{C} + C) \; + \overline{B}\,C\,(A + 1)$$

$$= \overline{A}\,B + \overline{A} + \overline{B}\,C$$

$$= \overline{A}(B + 1) \; + \overline{B}\,C$$

$$= \overline{A} + \overline{B}\,C$$

There is 1 OR operator.

1. 1 (B)

---

2. Boolean Algebra

First simplify the new operation:

$$A \;\$\; B = \overline{A\,\overline{B} + \overline{A}}$$

$$= \overline{A\,\overline{B}}\,A$$

$$= (\overline{A} + B)\,A$$

$$= \overline{A}\,A + A\,B$$

$$= A\,B$$

$$A \;\$\; B \; + \; B \;\$\; C \; + \; \overline{A} \;\$\; \overline{C}$$

$$= A\,B \; + B\,C + \overline{A}\,\overline{C}$$

$$\mathit{If}\;\; A = 0,\; \mathit{then}\; 0 + B\,C + \overline{C} = 0$$

$$\rightarrow \overline{C} = 0 \rightarrow C = 1 \wedge B = 0 \quad \Rightarrow (0,\ 0,\ 1)$$

$$\mathit{If}\;\; A = 1,\; \mathit{then}\; B + B\,C = 0$$

$$\rightarrow B = 0 \wedge C = 0 \;\mathit{or}\; 1 \;\Rightarrow\; (1,\ 0,\ 1),\ (1,\ 0,\ 0)$$

2. 3 (C)

3. Bit-String Flicking X = 01101 and Y = 10110 (RSHIFT-1 (LCIRC-3 X)) | (NOT (LSHIFT-1 ((RCIRC-2 X) & Y))) = (RSHIFT-1 (LCIRC-3 01101)) OR (NOT (LSHIFT-1 ((RCIRC-2 01101) AND 10110))) = (RSHIFT-1 01011) OR (NOT (LSHIFT-1 (01011 AND 10110))) = 00101 OR (NOT (LSHIFT-1 00010)) = 00101 OR (NOT 00100) = 00101 OR 11011 = 11111	3. 11111 (A)
4. Bit-String Flicking Let X = abcde and NOT X = ABCDE LHS = (LCIRC-2 01010) OR (RSHIFT-1 ((LCIRC-2 abcde) AND 01110)) = 01001 OR (RSHIFT-1 (cdeab AND 01110)) = (01001 OR (RSHIFT-1 0dea0) = 01001 OR 00dea = 01de1 LHS = RHS → 01de1 = 01101 → d = 1, e = 0, a = *, b = *, c = * → b = 1, c = 1, e = 1 → a = *, d = * Therefore X = abcde = ***10 8 solutions	4. 8 (C)
5. Recursive Functions $f(30) = f(30+3)+1 = f(33)+1 = 27+1 = 28$ $f(33) = 2\cdot f\left(\left[\frac{33}{2}\right]\right) - 3 = 2\cdot f(16) - 3 = 2\cdot 15 - 3 = 27$ $f(16) = 16 - 1 = 15$ $f(28) = 28 - 1 = 27$ $f(27) = 2\cdot f\left(\left[\frac{27}{2}\right]\right) - 3 = 2\cdot f(13) - 3 = 2\cdot 12 - 3 = 21$ $f(13) = 13 - 1 = 12$ $f(21) = 2\cdot f\left(\left[\frac{21}{2}\right]\right) - 3 = 2\cdot 9 - 3 = 15$ $f(10) = 10 - 1 = 9$ *So* $f(f(f(f(30))))$ $= f(f(f(28)))$ $= f(f(27))$ $= f(21)$ $= 15$	5. 15 (C)

6. Recursive Functions

$$f(14,20) = f(14+1, 20-2) + f(14,20) + 1$$
$$= f(15,18) + f(20,14) + 1 = 12 + 6 + 1 = 19$$
$$f(15,18) = f(15+1, 18-2) + f(18,15) + 1$$
$$= f(16,16) + f(18,15) + 1 = 8 + 3 + 1 = 12$$
$$f(20,14) = 20 - 14 = 6$$
$$f(16,16) = f\left(f\left(\frac{16}{2}, 16\right), \frac{16}{2}\right) - 3 = f(f(8,16), 8) - 3$$
$$= f(19,8) - 3 = 11 - 3 = 8$$
$$f(18,15) = 18 - 15 = 3$$
$$f(8,16) = f(8+1, 16-2) + f(16,8) + 1$$
$$= f(9,14) + f(16,8) + 1 = 10 + 8 + 1 = 19$$
$$f(9,14) = f(9+1, 14-2) + f(14,9) + 1$$
$$= f(10,12) + f(14,9) + 1 = 4 + 5 + 1 = 10$$
$$f(16,8) = 16 - 8 = 8$$
$$f(14,9) = 14 - 9 = 5$$
$$f(10,12) = f(10+1, 12-2) + f(12,10) + 1$$
$$= f(11,10) + f(12,10) + 1 = 1 + 2 + 1 = 4$$
$$f(11,10) = 11 - 10 = 1$$
$$f(12,10) = 12 - 10 = 2$$
$$f(19,8) = 19 - 8 = 11$$

6. 19 (A)

7. Digital Electronics

The digital circuit translates to:

$$\overline{\left(A + \overline{\overline{(A + B)}\,\overline{(B\,C)}}\right)}\,C$$
$$= \left(\overline{A}\left(\overline{A + B}\right)\left(\overline{B\,C}\right)\right)C$$
$$= (\overline{A}\,\overline{A}\,\overline{B}\,(\overline{B} + \overline{C}))\,C$$
$$= \overline{A}\,\overline{B}\,C\,(\overline{B} + \overline{C})$$
$$= \overline{A}\,\overline{B}\,C\,\overline{B} + \overline{A}\,\overline{B}\,C\,\overline{C}$$
$$= \overline{A}\,\overline{B}\,C + 0$$
$$= \overline{A}\,\overline{B}\,C \textit{ which is TRUE if } A = 0,\ B = 0 \textit{ and } C = 1$$

7. 001 (D)

8. Digital Electronics

The circuit translates to:

(A)(□(A, B, C) C) + ((□(A, B, C) + C)

Let X = □ (A, B, C). The expression is now: A X + (X + C)

A	B	C	X	AX	X + C	AX + (X+C)
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	1	1	1
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Therefore there are 6 triples that make the expression TRUE.

8. 6 (D)

9. Prefix-Infix-Postfix

2 4 # 4 2 $ 5 - & + 8 2 $ 7 3 $ * - &

= (2 4 #) (4 2 $) 5 - & + (8 2 $) (7 3 $) * - &

= 2 (3 5 -) & + (5 5 *) - &

= 2(-2 &) + 25 - &

= (2 2 +) 25 - &

= (4 25 -) &

= -21 &

= 21

9. 21 (D)

10. Prefix-Infix-Postfix

* / + a * b c * a ^ d 3 ^ b - c * 3 a

= * / + 1 (* 3 5) * 1 (^ 2 3) ^ 3 - 5 (* 3 1)

= * / (+ 1 15) (* 1 8) ^ 3 (- 5 3)

= * (/ 16 8) (^ 3 2)

= * 2 9

= 18

10. 18 (D)

11. Computer Number Systems

Change each to its binary representation:

50: 110010	55: 110111	60: 111100
51: 110011	56: 111000	61: 111101
52: 110100	57: 111001	62: 111110
53: 110101	58: 111010	63: 111111
54: 110110	59: 111011	64: 1000000

Therefore there are 60 1's.

11. 60 (B)

12. Computer Number Systems

$2020_8 - 202_8 - 20_8 + 2_8 = 1600_8$

Convert each bit to binary: 001 110 000 000

Group 4 at a time: 0011 1000 0000

Convert to hex: 3 8 0

12. 380 (C)

13. Data Structures

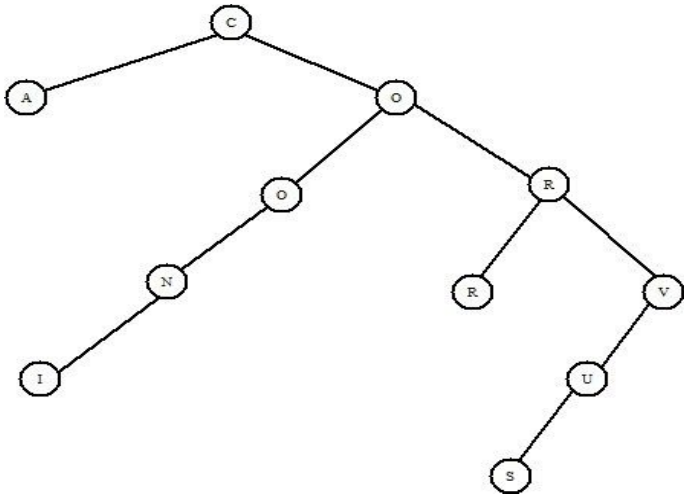
The queue is constructed using FIFO as follows:

R, RH, RHO, RHOD, ~~R~~HOD, HOD, ~~H~~OD, OD, ODO, ~~O~~DO. OD, ODD, ODDE, ODDEN, ~~O~~DDEN, DDEN, DDEND, DDENDR, ~~D~~DENDR, ~~D~~ENDR, ENDR, ENDRO, ENDRON, ~~E~~NDRON, NDRON, ~~N~~DRON, DRON, ~~D~~RON, RON. The next item popped would be R.

13. R (D)

14. Data Structures

The binary search tree is as follows:

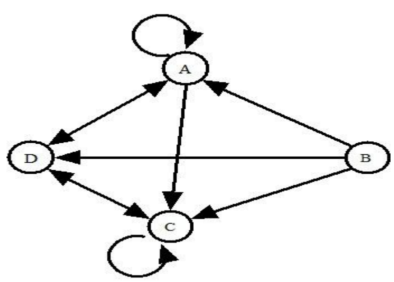


There are 4 nodes with only one left child: O, N, V, U

14. 4 (A)

15. Graph Theory

The graph that the adjacency matrix represents is:



The cycles are: AA, ACDA, ADA, CC, and CDC.

15. 5 (C)

16. Graph Theory

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 0 & 2 & 1 & 1 & 0 \\ 2 & 0 & 2 & 1 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & 1 & 0 \end{bmatrix}$$

The starting and ending vertices with the most paths of length 2 between them are from F to C or FC.

16. FC (C)

17. What Does This Program Do? This program counts the number of increasing factors of 2020 that sum to less than 2020. They are 1, 2, 4, 5, 10, 20, 101, 202, 404 and 505.	17. 10 (C)
18. LISP (SETQ Z ‘(C(O N)(N(E C)T)(I(C(U)T)))) (CADADAR (REVERSE (CDDR Z))) (CDDR Z) = (CDDR ‘(C(O N)(N(E C)T)(I(C(U)T)))) = (CDR ‘((O N)(N(E C)T)(I(C(U)T)))) = ‘((N(E C)T)(I(C(U)T))) (REVERSE ‘((N(E C)T)(I(C(U)T)))) =‘((I(C(U)T))(N(E C)T)) (CADADAR ‘((I(C(U)T))(N(E C)T))) = (CADADR ‘(I(C(U)T))) = (CADAR ‘((C(U)T))) = (CADR ‘(C(U)T)) = (CAR ‘((U)T)) = (U)	18. (U) (B)
19. FSAs and Regular Expressions `[^aeiou]* [aeiou] [fghj-np-t] +. (ing|ful|age|less)?` a. brush|ing - OK b. help/ful - OK c. fractals - fails at C d. java - fails at V e. python! - OK f. shapeless - OK g. igloo - fails at second o h. apple - OK i. striving - fails at v j. image - fails at g Therefore, there are 5 strings that satisfy the regular expression.	19. a, b, e, f, h (C)

20. Assembly Language	

The assembly programs can be converted to ACSL WDTPD code as follows:

```
input n
while n != 0
   b = int(n / 10)
   x = b * 10
   c = n - x
   m = b + c
   y = m - int(m /  3) * 3
   if m == y then
      print n
   end if
   input n
end while
```

This program checks if a given number is divisible by 3 by adding the digits to see if the sum is a multiple of 3. There are 4 such numbers before inputting 0: 24, 45, 51, 60.

20. 4 (A)