American Computer Science League

2021 Finals ● Solutions to Short Problems ● Senior Division

1. Boolean Algebra

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

1. $\overline{A} + B + \overline{C}$ (D)

2. Boolean Algebra

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

If $\overline{C} = 1$, then $C = 0$. 4 ordered triples make it TRUE. (*, *, 0)
If $\overline{C} = 0$, then $C = 1$. $\Rightarrow \overline{A} = 1$, $\overline{B} = 1$. (0, 0, 1)

2. 5 (C)

3. Bit-String Flicking

((LCIRC-25 011001) OR (RCIRC-16 101101) AND (NOT 011100))
= ((LCIRC-1 011001) OR (RCIRC-4 101101) AND 100011)
= (110010 OR (110110 AND 100011))
= (110010 OR 100010)
= 110010

3. 110010 (A)

4. Bit-String Flicking

Let X = abcde.
((LSHIFT-1 X) OR (NOT (RSHIFT-1 10111))) = X
((LSHIFT-1 abcde) OR (NOT (RSHIFT-1 10111))) = abcde
LHS = ((LSHIFT-1 abcde) OR (NOT (RSHIFT-1 10111)))
= (bcde0 OR (NOT 01011))
= (bcde0 OR 10100)
= 1c1e0
LHS = RHS $\Rightarrow$ 1c1e0 = abcde
$\Rightarrow$ a = 1, b = c, c = 1, d = e, e = 0
$\Rightarrow$ a = b = c = 1, d = e = 0
$\Rightarrow$ 11100

4. 11100 (B)

5. Recursive Functions

$f(20) = f(f(20-3)) + 3 = f(f(17)) + 3 = f(5) + 3 = 0$
$f(17) = f(f(17-3)) + 3 = f(f(14)) + 3 = f(3) + 3 = 2$
$f(14) = f(f(14-3)) + 3 = f(f(11)) + 3 = f(1) + 3 = 0$
$f(11) = f(f(11-3)) + 3 = f(f(18)) + 3 = f(-1) + 3 =$
$f(8) = f(8-2) - 2 = f(6) - 2 = 1 - 2 = -1$
$f(6) = f(6-2) + 2 = f(6) - 2 = 3 - 2 = 1$
$f(4) = 4 - 1 = 3$
$f(-1) = -1 - 1 = -2$
$f(1) = 1 - 1 = 0$
$f(3) = 3 - 1 = 2$
$f(5) = f(5-2) - 2 = f(3) - 2 = 2 - 2 = 0$

5. 3 (B)

6. Recursive Functions

Each zig zag curve has twice the length of the segment on which it is constructed. The perimeter will double the length at each stage.

Stage	1	2	3	4	5	6	7
Perimeter	64	128	256	512	1024	2048	4096

6. 7 (C)

7. Digital Electronics

The circuit can be represented by the following boolean expression:

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

A	B	C	$B \oplus C$	$A \oplus (B \oplus C$	$\overline{A \oplus (B \oplus C}$	$A \oplus \overline{A \oplus (B \oplus$
0	0	0	0	0	1	1
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	0	0	1	1
1	0	0	0	1	0	1
1	0	1	1	0	1	0
1	1	0	1	0	1	0
1	1	1	0	1	0	1

7. 4 (C)

8. Digital Electronics

The boolean expression for the circuit is: $\overline{A(\overline{A + B + BC}) + C}$

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

8. $\overline{C}$ (B)

9. Prefix-Infix-Postfix

3 4 5 + 7 2 1 & 6 ! * * 7 8 - 4 3 + 2 & 3 2 ^ ! / &
= 3 (4 5 +) (7 2 1 &)(6 !) * * (7 8 -)(4 3 +) 2 & (3 2 ^) ! / &
= 3 { (4 5 +) [(7 2 1 &)(6 !) *] *} {[(7 8 -)(4 3 +) 2 &][(3 2 ^) !] / }&
= 3{ (+4 5) [(&7 2 1)(! 6) *] *} {[(-7 8)(+4 3) 2 &][(^3 2) !] / }&
= 3{ (+4 5) [*&7 2 1! 6)] *} {[& -7 8 +4 3 2][!^3 2] / }&
= 3{ *+4 5*&7 2 1! 6} {/& -7 8 +4 3 2!^3 2 }&
= &3*+4 5*&7 2 1! 6/& -7 8 +4 3 2!^3 2

9. &3*+4 5*&7 2 1! 6/& -7 8 +4 2!^3 2 (D)

10. Prefix-Infix-Postfix

* - + 4 7 - 1 8 / + ^ 5 2 * 5 - 6 2 ^ 3 2
= * - (+ 4 7) (- 1 8) / + (^ 5 2) * 5 (- 6 2) (^ 3 2)
= (* (- 11 (-7)) / + 25 (* 5 4) 9
= * 18 /(+ 25 20) 9
= * 18 (/ 45 9)
= * 18 5
= 90

10. 90 (C)

11. Computer Number Systems

$A74F2C_{16}$ = 1010 0111 0100 1111 0010 1100_2
4_{16} = 100_2
Dividing 1010 0111 0100 1111 0010 1100_2 by 100_2
shifts the bits 2 spaces to the right. Therefore the answer is:
1010 0111 0100 1111 0010 11_2 = 10 1001 1101 0011 1100 1011_2
= 2 9 D 3 C B_{16}

11. $29D3CB_{16}$ (B)

12. Computer Number Systems

2001 = 3721_8 2021 = 3745_8

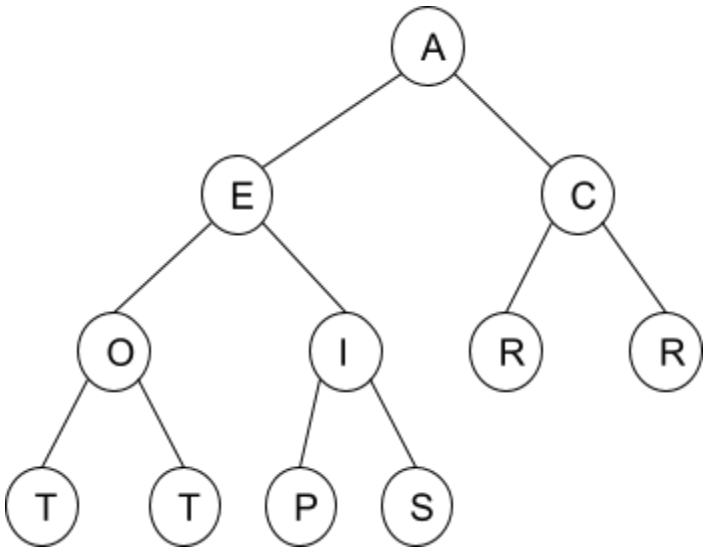
$YEAR_8$	3721	3722	3723	3724	3725	3726	3727
DIGITS	1	1	1	1	2	2	2
$YEAR_8$	3730	3731	3732	3733	3734	3735	3736
DIGITS	1	1	1	1	1	2	2
$YEAR_8$	3737	3740	3741	3742	3743	3744	3745
DIGITS	2	1	1	1	1	1	2

There are 28 digits greater than 4.

12. 28 (D)

13. Data Structures

The min-heap for TRICERATOPS is:



13. O I R R (C)

14. Data Structures

FIFO: July: $20/share (sold)
March: $10/share (bought)
Profit: $10/share
Total profit: $1000 for 100 shares
Tax: 20% of $1000 = $200

LIFO: July: $20/share (sold)
May: $5/share (bought)
Profit: $15/share
Total profit: $1500 for 100 shares
Tax: 20% of $1500 = $300

14. FIFO, saving $100 (A)

15. Graph Theory

Squaring a matrix gives the number of paths of length 2.

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

There are 15 paths that do not have a length of 2, but only 4 unique pairs of vertices do not have a path of length 2 in either direction.

DD, AD/DA, BD/DB, DF/FD

15. 4 (B)

16. Graph Theory

There are 12 cycles: ACA, ABCA, ABEA, ADEA, ACBEA, ACDEA, ABCDEA, ADEBCA, BEB, BCB, BCDEB, and DED

16. None of the above (E)

17. What Does This Program Do?

Pascal's triangle rows start and end with 1. A new entry, pt(i), in a row is found by adding the entry above to the left, pt(i-1), and the one to the right, pt(i). This is choice B.

17.
```
for i = X to 1 step -1
    pt(i) = pt(i) + pt(i-1)
next i
```
(B)

18. LISP Programming

G = (CADDADADDR S)
= (CADDADADDR '(H ((A C) K) (E (R R (A))) (N (K))))
= (CADDADADR '(((A C) K) (E (R R (A))) (N (K))))
= (CADDADAR '((E (R R (A))) (N (K))))
= (CADDADR '(E (R R (A))))
= (CADDAR '((R R (A))))
= (CADDR '(R R (A)))
= (CADR '(R (A)))
= (CAR '((A)))
= (A)

H = (CADADADR Q))
= (CADADADR '((P L) (A (T (F O) R)) M))
= (CADADAR '((A (T (F O) R)) M))
= (CADADR '(A (T (F O) R)))
= (CADAR '((T (F O) R)))
= (CADR '(T (F O) R))
= (CAR '((F O) R))
= (F O)

(CONS G H) = (CONS (A) (F O))
= ((A) F O)

18. ((A) F O) (A)

19. FSAs and Regular Expressions

Given the pattern **`[^aeiou][aeiou][^a-j][p-t]?.(er|s)*`** the following strings fail at the given character:

crayons: r - must be a vowel
erasers: e - cannot start with a vowel
staples: t - must be a vowel
staplers: t - must be a vowel
notebooks: e - must be p-t
computers: t - there is no symbol to use
tablets: b - cannot be a-j
boards: a - cannot be a-j
rulers: e - must have another character before the er or the s

There are 3 left which satisfy the regular expression which are: markers, desks, and compass.

19. None of the above (E)

20. Assembly Language

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

```
input x, y
z = x - y
c = 1: b = 1: a = 1
while z > 0
    c = c * z
    z =  z - 1
end while
while y > 0
    b = b * y
    y = y - 1
end while
while x > 0
    a = a * x
    x = x - 1
 end while
 d = a / b / c
 output d
```

Each while loop calculates a factorial.
d = a!/b!/c! which is the formula to calculate 8 things taken 3 at a time where order doesn't matter.

20. 56 (C)