American Computer Science League

2018-2019 **Contest #2**

SENIOR DIVISION SOLUTIONS

---

1. Pre/Post/Infix Notation

+ / ↑ 2 4 # 2 1 * # 4 5 ↑ / − * 6 3 ↑ 2 3 * 5 2 # 2 2
= + / (↑ 2 4) (# 2 1) * (# 4 5) ↑ / − (* 6 3) (↑ 2 3) (* 5 2) (# 2 2)
= + (/ 16 2) * 5 ↑ / (− 18 8) 10 2
= + 8 * 5 ↑(/ 10 10) 2 = + 8 * 5 (↑ 1 2)
= + 8 (* 5 1) = + 8 5 = 13

1. 13

---

2. Pre/Post/Infix Notation

Prefix: − + / * A + B C ↑ B 2 / * B C A / + ↑ A 2 ↑B 2 C
= − + / * A (+ B C) (↑ B 2) / (* B C) A / + (↑ A 2) (↑B 2) C
= − + / (* A (B + C)) (B ↑ 2) (/ (B * C) A) /(+ (A ↑ 2) (B ↑ 2)) C
= − + (/ (A * (B + C)) (B ↑ 2)) ((B * C) / A) (/ (A ↑ 2 + B ↑ 2) C)
= − (+ ((A * (B + C))/(B ↑ 2)) ((B * C) / A)) ((A ↑ 2 + B ↑ 2) / C)
= − (((A * (B + C)) / (B ↑2)) + ((B * C) / A)) ((A ↑ 2 + B ↑ 2) / C)

Infix : = (((A * (B + C)) / (B ↑2)) + ((B * C) / A)) − ((A ↑ 2 + B ↑ 2) / C)
= (((A B C + *) / (B 2 ↑)) + (B C * A /)) − (A 2 ↑ B 2 ↑ + C /)
= ((A B C + * B 2 ↑ /) + (B C * A /)) − (A 2 ↑ B 2 ↑ + C /)
= (A B C + * B 2 ↑ / B C * A / +) − (A 2 ↑ B 2 ↑ + C /)

Postfix: = A B C + * B 2 ↑ / B C * A / + A 2 ↑ B 2 ↑ + C / −

2. As shown

---

3. Bit-String Flicking

(LSHIFT-2 01101) AND ((RCIRC-3 10010) OR 01100)
= 10100 AND (01010 OR 01100)
= 10100 AND 01110 = 00100

3. 00100

---

4. Bit-String Flicking

Let X = abcde
LHS = (LCIRC-2 (RSHIFT-1 (LCIRC-2 abcde)))
= (LCIRC-2 (RSHIFT-1 cdeab))
= (LCIRC-2 0cdea) = dea0c
RHS = 01001 OR (NOT 10110) = 01001 OR 01001 = 01001
LHS = RHS → dea0c = 01001 → d = 0, e = 1, a = 0, c = 1, b = *
Therefore X = 0*101

4. 0*101

SENIOR DIVISION SOLUTIONS

5. LISP

```
(CAR(CDR(CAR(CAR(REVERSE(CDR(CDR
        (CDR '((1 (2 3)) (2 3) 4 (2 3 (1)) 2 3 ((1 (2 3) 4) 1)))))))))
 = (CAR(CDR(CAR(CAR(REVERSE(CDR
        (CDR '((2 3) 4 (2 3 (1)) 2 3 ((1 (2 3) 4) 1)))))))))
 = (CAR(CDR(CAR(CAR(REVERSE
        (CDR '(4 (2 3 (1)) 2 3 ((1 (2 3) 4) 1)))))))
 = (CAR(CDR(CAR(CAR(REVERSE '((2 3 (1)) 2 3 ((1 (2 3) 4) 1)))))))
 = (CAR(CDR(CAR(CAR '(((1 (2 3) 4) 1) 3 2 (2 3 (1)))))))
 = (CAR(CDR(CAR '((1 (2 3) 4) 1))))
 = (CAR(CDR '(1 (2 3) 4)))
 = (CAR '((2 3) 4))
 = (2 3)
```

5. (2 3)